Automotive applications of explicit non-linear model predictive control

Mathias Metzler

Supervisors: Prof. A. Sorniotti
Dr. P. Gruber

Centre for Automotive Engineering
Department of Mechanical Engineering Sciences
University of Surrey

This dissertation is submitted for the degree of
Doctor of Philosophy

August 2019
This thesis is dedicated to my loving parents, who have always given me the foundation and the freedom to fulfil my dreams. I am very grateful for their endless support.
“I have no special talent. I am only passionately curious.”

— Albert Einstein
Declaration

This thesis and the work to which it refers are the results of my own efforts. Any ideas, data, images, or text resulting from the work of others (whether published or unpublished) are fully identified as such within the work and attributed to their originator in the text, bibliography, or in footnotes. This thesis has not been submitted in whole or in part for any other academic degree or professional qualification. I agree that the University has the right to submit my work to the plagiarism detection service TurnitinUK for originality checks. Whether or not drafts have been so-assessed, the University reserves the right to require an electronic version of the final document (as submitted) for assessment as above.

Mathias Metzler
August 2019
Acknowledgements

I would like to express my deep gratitude to my research supervisors Professor Aldo Sorniotti and Doctor Patrick Gruber for their guidance, critical feedback, and valuable suggestions to improve my work during the entire PhD.

I would like to acknowledge the financial support provided by MTS Systems Ltd and the facilities made available by the University of Surrey’s Faculty of Engineering and Physical Sciences.

I would also like to thank all the colleagues from the Centre of Automotive Engineering for interesting discussions, fruitful collaborations, and making the time at the University enjoyable.

Finally, I must express my very profound gratitude to my fiancée Irina Dimitrova for providing me with unfailing support and continuous encouragement throughout the process of researching and writing this thesis. This accomplishment would not have been possible without her. Thank you.
Abstract

This thesis presents automotive applications of explicit non-linear model predictive control. Model predictive control is a state-of-the-art control methodology that allows for systematic incorporation of physical and operational constraints in the control system design. Multiple-input multiple-output systems can be easily employed in the optimal control problem. The receding horizon method combines optimisation and feedback adjustment. However, for systems with non-linear and sufficiently fast dynamics, the online optimisation can be a challenging task to accomplish in real-time on embedded hardware. The non-linear optimisation problem is formulated in parametric form and a sub-optimal solution thereof is computed offline. Local accurate multi-parametric quadratic approximations are used together with iterative and recursive orthogonal partitioning of the parameter exploration space. A piecewise affine state feedback law defined on polyhedral regions is obtained. The reduction in online software complexity leads to computing times lower than the required sampling times. The explicit solution, with guaranteed levels of sub-optimality, allows for a priori verification and functional safety validation being viable for safety-critical applications.

Three representative automotive control problems are selected. Each case study includes a detailed analysis of the control system based on explicit non-linear model predictive control (NMPC). The design and performance assessment of an explicit NMPC-based vehicle stability controller with an electro-hydraulic braking system is presented in the first case study. A systematic simulation-based investigation on the prediction model complexity shows the necessity of a non-linear lateral tyre force model for acceptable controller behaviour. A load transfer model considering the side-slip angle rate is important for an accurate prediction of lateral tyre forces and their yaw moment contributions. The modelling of longitudinal and lateral tyre force coupling significantly influences the front-to-rear braking force distribution. The second case study proposes traction controllers (TC) based on explicit NMPC for electric vehicles with in-wheel motors and compares these with more conventional TC strategies based on proportional-integral control. The NMPC allows for longitudinal slip tracking improvement during variable tyre-road friction scenarios simulated with a high fidelity vehicle model. Experimental validation on a fully electric vehicle prototype demonstrates real-time operation. An explicit NMPC-based TC for combined driving and cornering conditions is presented.
for a front-wheel-drive electric vehicle with in-wheel motors. The controller based on a combined slip tyre force model shows enhanced slip tracking performance compared to a pure longitudinal slip tyre force model in a TC scenario. A proof-of-concept design of an explicit NMPC for anti-lock braking systems (ABS) is demonstrated. The study includes an implementation on an industrial electro-hydraulic braking unit. The experimental comparison on a hardware-in-the-loop test-rig shows that the NMPC ABS consistently outperforms a proportional-integral-derivative ABS. In the third case study, the design, optimisation-based tuning, and systematic comparison of six different explicit NMPC anti-jerk controllers for a front-wheel-drive electric vehicle with on-board motors are presented. The inclusion of a non-linear backlash model improves in all cases the controller behaviour. The modelling of the wheel dynamics and the tyre dynamics with constant relaxation length does not bring significant performance enhancements.

The detailed analysis of the explicit solutions shows the best suitability of the approach for systems with a low number of states, minimal number of additional parameters, limited number of control inputs, strong non-linearities, and fast dynamics for applications that are safety-critical, subjected to strict performance requirements, and characterised by low computational power but sufficient memory capacity. The influence on the controller complexity in terms of polyhedral critical regions and memory requirements of various aspects, e.g. number of parameters (states and varying parameters), number of control inputs, definition of the parameter exploration space, formulation and number of constraints, etc. is investigated. The saturation of parameters, degrading closed-loop performance, and the exploitation of symmetry properties and state transformations, reducing complexity, are analysed. The role of soft constraints, promoting feasibility, is discussed. The influence of approximation tolerances, requiring a compromise between acceptable performance degradations and controller complexity, is shown. Methods to reduce complexity are embedded in the proposed post-processing algorithm that achieves good performance. These include memory-optimised binary search tree generation, reducing the memory requirements and online complexity, disjoint optimal and sub-optimal merging procedures and clipping-based complexity reduction, both leading to lower numbers of polyhedral regions in the explicit approximate receding horizon feedback law. An analysis on the memory requirements shows great variations depended on the specific application. The empirically assessed execution times of the explicit controllers on a rapid control prototyping unit are in the sub-millisecond range and prove real-time feasibility. The applications of the explicit NMPC demonstrate the incorporation of non-linear system dynamics in the controller design, reduction of online computing times, the systematic constraint satisfaction, and the possibility of a priori verification for safety-critical applications. The analysis also shows increased memory requirements with the suitability of the explicit NMPC approach depending on the specific system including design choices and the number of states and parameters, respectively.
# Table of contents

List of figures .............................. xvii
List of tables ................................ xix
Nomenclature ................................ xxi

1 Introduction ................................ 1
   1.1 Motivation ................................... 1
   1.2 Research contributions ......................... 3
   1.3 List of publications ............................ 5
   1.4 Structure of the thesis ......................... 6
   1.5 Personal contribution ......................... 8

I Methodology ................................ 9

2 Introduction and literature review ............. 11

3 Non-linear model predictive control ........... 21
   3.1 Continuous-time optimal control problem ....... 21
      3.1.1 Dimensional NMPC problem ................ 21
      3.1.2 Regulation problem .......................... 23
      3.1.3 Reference tracking problem ................. 24
   3.2 Non-dimensionalisation and scaling ............. 25
      3.2.1 Scaling factors in the cost function ....... 25
      3.2.2 Non-dimensionalisation of the optimal control problem ...... 26
      3.2.3 Dimensionless NMPC problem ............... 30
   3.3 Numerical optimal control ..................... 31
      3.3.1 Finite parametrisation and discretisation ... 31
      3.3.2 Direct methods .............................. 34
      3.3.3 Formulation as an mp-NLP problem ........... 43
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.4</td>
<td>Tuning and stability</td>
<td>46</td>
</tr>
<tr>
<td>3.4</td>
<td>Numerical optimisation</td>
<td>50</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Non-linear programming methods</td>
<td>50</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Jacobian and Hessian matrices</td>
<td>52</td>
</tr>
<tr>
<td>4</td>
<td>Explicit non-linear model predictive control</td>
<td>53</td>
</tr>
<tr>
<td>4.1</td>
<td>Sub-optimal solution using mp-QP approximations of mp-NLP</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Local mp-QP approximation to mp-NLP</td>
<td>55</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Local mp-QP approximation</td>
<td>55</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Reformulation into MPT 3.0 format</td>
<td>60</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Eigen-decomposition of Hessian matrix</td>
<td>61</td>
</tr>
<tr>
<td>4.3</td>
<td>mp-QA algorithm for explicit approximate solution of mp-NLP</td>
<td>62</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Close-to-global solution of non-convex mp-NLP</td>
<td>62</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Computation of an explicit approximate solution</td>
<td>64</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Estimation of errors bounds</td>
<td>64</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Structure of the algorithm</td>
<td>66</td>
</tr>
<tr>
<td>4.3.5</td>
<td>Heuristic splitting rules</td>
<td>69</td>
</tr>
<tr>
<td>4.4</td>
<td>Post-processing of the explicit approximate solution of mp-NLP</td>
<td>71</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Complexity reduction</td>
<td>74</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Data handling and point location</td>
<td>78</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Post-processing algorithm</td>
<td>82</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Online evaluation</td>
<td>86</td>
</tr>
</tbody>
</table>

**II Application**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Vehicle stability control</td>
<td>91</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction and literature review</td>
<td>91</td>
</tr>
<tr>
<td>5.2</td>
<td>Control system design</td>
<td>107</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Prediction models</td>
<td>107</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Optimal control problem</td>
<td>116</td>
</tr>
<tr>
<td>5.3</td>
<td>Performance analysis</td>
<td>125</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Simulation model</td>
<td>126</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Test manoeuvre</td>
<td>127</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Performance indicators</td>
<td>129</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Controlled and uncontrolled vehicle</td>
<td>133</td>
</tr>
<tr>
<td>5.3.5</td>
<td>Controller tuning and constraints</td>
<td>139</td>
</tr>
<tr>
<td>5.3.6</td>
<td>Model complexity</td>
<td>143</td>
</tr>
<tr>
<td>5.3.7</td>
<td>Sensitivity analysis and robustness</td>
<td>159</td>
</tr>
<tr>
<td>5.3.8</td>
<td>Sampling time and execution time</td>
<td>161</td>
</tr>
</tbody>
</table>
## Table of contents

5.3.9 Explicit solution ........................................... 164
5.4 Conclusion and future work .................................. 171

### 6 Wheel slip control

6.1 Traction control .............................................. 173
   6.1.1 Introduction and literature review ...................... 173
   6.1.2 Contribution ........................................... 175
   6.1.3 Conclusion and future work ............................. 175
6.2 Traction control in cornering ................................. 177
   6.2.1 Introduction ........................................... 177
   6.2.2 Control system design .................................. 177
   6.2.3 Performance analysis ................................... 184
   6.2.4 Conclusion and future work ............................. 192
6.3 Anti-lock braking control .................................... 193
   6.3.1 Introduction and literature review ...................... 193
   6.3.2 Contribution ........................................... 195
   6.3.3 Conclusion ............................................. 195

### 7 Anti-jerk control

7.1 Introduction and literature review .......................... 197
7.2 Control system design ....................................... 200
   7.2.1 Prediction models ..................................... 200
   7.2.2 Optimal control problem ............................... 206
7.3 Performance analysis ....................................... 209
   7.3.1 Simulation model ..................................... 209
   7.3.2 Performance indicators and controller tuning ........... 210
   7.3.3 Performance assessment ................................. 213
   7.3.4 Sensitivity analysis ................................... 216
   7.3.5 Explicit solution ..................................... 219
7.4 Conclusion and future work .................................. 225

### 8 Conclusion

8.1 Summary and conclusion ..................................... 227
8.2 Outlook and future work ..................................... 235

### References

| References | 237 |
List of figures

5.1 Notations and sign conventions of the proposed vehicle model formulation. 108
5.2 Illustration of the tyre models in pure lateral slip. ............... 114
5.3 Simplified model of the Pacejka Magic Formula in pure lateral slip. . 115
5.4 Simplified model of the Pacejka Magic Formula in combined slip. . 115
5.5 Steady-state yaw velocity in varying tyre-road friction conditions. . . 119
5.6 Reference yaw velocity as a function of vehicle speed and steering angle. . 120
5.7 Comparison of the high-fidelity simulation model and the real vehicle demonstrator in a step-steer manoeuvre. . . . . . . . . . . . . . . . . . . 126
5.8 Sine with Dwell test. ........................................... 127
5.9 Under-steer characteristics of the uncontrolled vehicle. ............... 128
5.10 Dynamic response of the uncontrolled vehicle in run 12. ............. 133
5.11 Dynamic response of the uncontrolled vehicle in the state-space. . . 134
5.12 Selected performance indicators of the uncontrolled vehicle. ........ 135
5.13 Dynamic response of the vehicle with controller (f) in run 12. ......... 136
5.14 Dynamic response of the vehicle with controller (f) in the state-space. . 137
5.15 Selected performance indicators of the vehicle with controller (f). ..... 138
5.16 Directional stability and responsiveness criteria for the controlled and uncontrolled vehicle. ............................................. 139
5.17 Selected performance indicators showing the influence of different weights. 140
5.18 Influence of soft constraints for controller (f) in run 12. ............... 142
5.19 Comparison of the direct yaw moment reference in controller (f) with Setting A and B in run 12. ........................................... 143
5.20 Dynamic response of the vehicle with controllers (e) and (d) in run 12. . 149
5.21 Selected performance indicators of the vehicle with controllers (e) and (d). 150
5.22 Dynamic response of the vehicle with controllers (c) and (f) in run 12. . 153
5.23 Selected performance indicators of the vehicle with controllers (c) and (f). 154
5.24 Dynamic response of the vehicle with controllers (g) and (f) in run 12. . 156
5.25 Selected performance indicators of the vehicle with controller (f) with and without delay. ................................................ 163
5.26 Dynamic response of the vehicle with controller (f) for symmetry detection and parameter space definition. .......................... 165
5.27 Orthogonal partitions and polyhedral regions of explicit controller (f)-(5). 169
5.28 Feedback law of explicit controller (f)-(5). .......................... 170

6.1 Longitudinal tyre force characteristics. ............................... 180
6.2 Time histories for controllers (a) and (b) in the traction control scenario. 185
6.3 Orthogonal partitions and polyhedral regions of explicit controller (a)-(4). 190
6.4 Feedback law of explicit controller (a)-(4). ............................... 191

7.1 Graphical representation of the prediction models. ...................... 200
7.2 Non-linear backlash model. ........................................... 203
7.3 Block diagram of the explicit non-linear MPC. .......................... 205
7.4 Performance index $J$ for the nominal tip-in test. ...................... 214
7.5 Backlash influence for controller $C_3$ with settings $N_p = 2$ and $H = 10$ ms. 215
7.6 Backlash influence for controller $C_4$ with settings $N_p = 2$ and $H = 1$ ms. 216
7.7 Performance index $J$ for different initial speeds and torque requests. ... 218
7.8 Orthogonal partitions and polyhedral regions of explicit controller $C_3^{BL}$ with reduced constraints. .......................... 223
7.9 Feedback law of explicit controller $C_3^{BL}$ with reduced constraints. 224
List of tables

5.1 Main parameters of the prediction model .............................. 113
5.2 Settings of the optimal control problem ............................... 123
5.3 Definition of test cases ................................................. 144
5.4 Part I of average performance indicators for a series of SwD test runs ............................... 145
5.5 Part II of average performance indicators for a series of SwD test runs ............................... 146
5.6 Part III of average performance indicators for a series of SwD test runs ............................... 147
5.7 Selected average performance indicators for a series of SwD test runs with parameter variations ................................................. 160
5.8 Definition of post-processing settings ..................................... 166
5.9 Partitioning algorithm and post-processing data for explicit controller (f) ............................... 167
6.1 Prediction models ................................................... 181
6.2 Settings of the optimal control problem ................................ 183
6.3 Performance indicators in the traction control scenario ......................... 187
6.4 Partitioning algorithm and post-processing data for explicit controller (a) ............................... 189
7.1 Prediction models ................................................... 205
7.2 Settings of the optimal control problem ................................ 207
7.3 Main vehicle parameters................................................ 209
7.4 Performance indicators for the best performing controllers in the nominal tip-in test ................................................. 216
7.5 Partitioning algorithm and post-processing data for the best performing explicit controllers ................................................. 220
7.6 Partitioning algorithm and post-processing data for the best performing explicit controllers with reduced constraints ................................................. 222
Nomenclature

Roman Symbols

$A$ Matrix for linear term of decision variables in approximate inequality constraint functions

$\mathcal{A}$ Optimal active set

$A$ Vehicle frontal area

$A_{\delta_{sw}}$ Steering wheel angle amplitude at $0.3 \text{ g}$ lateral acceleration

$A_e$ Matrix for linear term of decision variables in approximate equality constraint functions

$a_k$ Vector for linear term of hyperplane definition

$A_{th}$ Matrix for linear term of definition of parameter space partition

$a_x$ Longitudinal vehicle acceleration

$a_{x,\text{ref}}$ Reference longitudinal vehicle acceleration

$a_x^*$ High frequency component of the longitudinal vehicle acceleration

$a_y$ Lateral vehicle acceleration

$a_{y,\text{max}}$ Maximum lateral vehicle acceleration

$B$ Matrix for linear term of parameters in approximate inequality constraint functions

$b$ Vector for constant term in approximate inequality constraint functions

$b$ Front-to-total braking ratio of the longitudinal tyre forces

$b_e$ Vector for constant term in approximate equality constraint functions

$b_k$ Constant term of hyperplane definition
\( \mathbf{B}_\nu \) Vector of bound constraints on the slack variables

\( B_r \) Stiffness factor of reference yaw velocity

\( B_{s_{ij}} \) Stiffness factor of lateral tyre force on corner \( ij \)

\( B_{s_i} \) Stiffness factor of lateral tyre forces on the front and rear axle

\( \mathbf{b}_{th} \) Vector for constant term of definition of parameter space partition

\( \mathbf{B}_u \) Vector of bound constraints on the inputs

\( \mathbf{B}_x \) Vector of bound constraints on the states

\( B_x \) Stiffness factor of longitudinal tyre force

\( B_{xa} \) Stiffness factor for combined slip weighting function

\( \mathbf{C} \) Vector for linear term of parameters in approximate cost function

\( c \) Constant term in approximate cost function

\( C_d \) Equivalent torsional drivetrain damping

\( C_r \) Shape factor of reference yaw velocity

\( C_{s_{ij}} \) Shape factor of lateral tyre force on corner \( ij \)

\( C_{s_i} \) Shape factor of lateral tyre forces on the front and rear axle

\( C_U \) Indicator for sanity checks to the union \( U \) of polyhedra

\( C_x \) Aerodynamic drag coefficient

\( C_x \) Shape factor of longitudinal tyre force

\( C_{xa} \) Shape factor for combined slip weighting function

\( C_z \) Vertical tyre stiffness

\( \mathbf{D} \) Canonical form of Hessian matrix \( \mathbf{H}_0 \)

\( d \) Number of system and controller parameters

\( d \) Track width

\( \mathbf{D}_0 \) Jacobian of cost function w.r.t. decision variables at linearisation point
Nomenclature

$d_c$ Number of controller parameters

$\Delta F_{z_i}$ Change in vertical tyre forces on the front and rear axle due to load transfer effects

$\Delta T$ Time to normalise performance indicators of each SwD test run

$\Delta T$ Torque reduction imposed by the traction controller

$\Delta t_{a_x}$ Time delay between motor torque input variation and achievement of longitudinal reference acceleration

$\Delta v_x$ Difference between passive and active vehicle speed at the end of the tip-in

$df_z$ Normalised change in vertical tyre force

$df_{z_{ij}}$ Normalised change in vertical tyre force on corner $ij$

$D_{l_{ij}}$ Peak factor of longitudinal tyre force on corner $ij$

$D_r$ Peak factor of reference yaw velocity

$d_s$ Number of system parameters

$D_{s_{ij}}$ Peak factor of lateral tyre force on corner $ij$

$D_{s_i}$ Peak factor of lateral tyre forces on the front and rear axle

$D_{ux}$ Vector of parametrised dynamic equality constraints

$D_x$ Peak factor of longitudinal tyre force

$E$ Matrix for linear term of parameters in approximate equality constraint functions

$E_{e,0}$ Jacobian of equality constraint functions w.r.t. parameters at linearisation point

$E_{i,0}$ Jacobian of inequality constraint functions w.r.t. parameters at linearisation point

$e_{\kappa_x}$ Longitudinal slip violation

$e_{\dot{\psi}}$ Vehicle yaw velocity error

$e_{\dot{\psi},\text{max}}$ Maximum bound on yaw velocity error

$e_{\dot{\psi},\text{min}}$ Minimum bound on yaw velocity error

$E_{ux}$ Vector of arbitrary equality constraints

$F$ Matrix for linear term of decision variables and parameters in approx. cost function
\( F \)  Numerical integration scheme  
\( f \)  Vector for linear term of decision variables in approximate cost function  
\( F, f \)  Vector of dimensional/dimensionless state functions  
\( F_0 \)  Matrix defined by Jacobians of cost function w.r.t. decision variables and parameters at linearisation point  
\( f_0, f_2 \)  Tyre rolling resistance coefficients  
\( F_{ij} \)  Tyre force on corner \( ij \) in the road plane  
\( F_j \)  Matrix for linear term of piecewise affine feedback law in critical region \( R_j \)  
\( F_{t_{ij}} \)  Longitudinal tyre force on corner \( ij \)  
\( F_{s_{ij}} \)  Lateral tyre force on corner \( ij \)  
\( F_{s_{0_{ij}}} \)  Lateral tyre force on corner \( ij \) in pure lateral slip conditions  
\( F_{s_{0_{F}}} \)  Lateral forces on front axle in pure lateral slip conditions  
\( F_{s_{0_{R}}} \)  Lateral forces on rear axle in pure lateral slip conditions  
\( F_x \)  Longitudinal tyre force  
\( F_{z} \)  Vehicle traction or braking force  
\( F_{z_{0}} \)  Longitudinal tyre force in pure longitudinal slip conditions  
\( F_{z} \)  Vertical tyre force  
\( F_{z_{ij}} \)  Vertical tyre force on corner \( ij \)  
\( F_{z_{0_{ij}}} \)  Nominal vertical tyre force on corner \( ij \)  
\( F_{z_{0}} \)  Nominal vertical tyre force  
\( F_{z_{ij,stat}} \)  Static vertical tyre force on corner \( ij \)  
\( G \)  Vector of (inequality and equality) constraint functions  
\( g \)  Gravitational acceleration  
\( G_{B_v} \)  Vector of bound constraint functions on the slack variables  
\( G_{B_u} \)  Vector of bound constraint functions on the inputs
Nomenclature

$G_{B_x}$ Vector of bound constraint functions on the states

$G_{D_{ux}}$ Vector of parametrised dynamic equality constraint functions

$G_e$ Vector with violated equality constraint functions

$G_{e}$ Vector of equality constraint functions

$G_{e,0}$ Jacobian of equality constraint functions w.r.t. decision variables at linearisation point

$G_{E_{ux}}$ Vector of arbitrary equality constraint functions

$\hat{G}$ Linear approximate constraint functions of local mp-QP problem

$G_i$ Vector with violated inequality constraint functions

$G_{i}$ Vector of inequality constraint functions

$G_{i,0}$ Jacobian of inequality constraint functions w.r.t. decision variables at linearisation point

$g_j$ Vector for constant term of piecewise affine feedback law in critical region $R_j$

$G_{P_u}$ Vector of path constraint functions on the inputs

$G_{P_{ux}}$ Vector of path constraint functions on the inputs and states

$G_{P_x}$ Vector of path constraint functions on the states

$G_{xa}$ Weighting function for longitudinal tyre force model in combined slip conditions

$H$ Matrix for quadratic term of decision variables in approximate cost function

$H, h$ Vector of dimensional/dimensionless output functions

$h$ Height of vehicle centre of gravity

$H, h$ Dimensional/dimensionless discretisation time step

$H_0$ Hessian of cost function w.r.t. decision variables at linearisation point

$\bar{H}_0$ Modified positive definite Hessian matrix

$h_F$ Roll centre height of front suspension

$H_j$ Matrix defining the hyperplanes of polyhedral region $R_j$
\( h_k \)  Function for hyperplane definition

\( h' \)  Distance between roll axis and the vehicle centre of gravity

\( h_R \)  Roll centre height of rear suspension

\( I_n \)  Identity matrix of dimension \( n \)

\( \bar{I} \)  Average value of performance indicators over the series of 22 SwD test runs

\( I_{\text{est}} \)  Performance indicator for the prediction model of the NMPC controller in each run of the SwD test series

\( i_g \)  Gear ratio

\( \mathcal{I}_j \)  Index set of hyperplanes defining polyhedral critical region \( R_j \)

\( I_{\text{sim}} \)  Performance indicator for the high-fidelity simulation model in each run of the SwD test series

\( I_z \)  Vehicle yaw mass moment of inertia

\( J \)  Cost function for controller tuning and performance assessment

\( J_1 \)  Directional stability criterion for the yaw velocity at \( T_{0+1} \) in the SwD test

\( J_1 \)  Equivalent mass moment of inertia for electric motor, gearbox, and half-shaft

\( J_1 \)  Integral of the absolute value of the control error

\( J_2 \)  Directional stability criterion for the yaw velocity at \( T_{0+1.75} \) in the SwD test

\( J_2 \)  Equivalent mass moment of inertia for half-shaft and wheel

\( J_2 \)  Integral of the absolute value of the control action

\( J'_2 \)  Equivalent mass moment of inertia for half-shaft, wheels, and vehicle mass

\( J_3 \)  Equivalent mass moment of inertia for vehicle mass and wheel

\( J_3 \)  Responsiveness criterion in the SwD test

\( J_{\text{EM}} \)  Mass moment of inertia of the rotor of the electric motor

\( J_{g1}, J_{g2} \)  Mass moment of inertia of the primary and secondary gearbox shaft

\( J_{hs} \)  Mass moment of inertia of the half-shaft
\( J_w \)  Mass moment of inertia of the wheel

\( k \)  Piecewise affine feedback law (computing the sub-optimal solution)

\( K_1, K_2 \)  Tuning parameters of the backlash model

\( K_{\alpha R} \)  Direct yaw moment reference gain based on the rear slip angle

\( \bar{k} \)  Maximum saturation limits of feedback law

\( \hat{k} \)  Complexity reduced piecewise affine receding horizon feedback law

\( K_d \)  Equivalent torsional drivetrain stiffness

\( K_{e_y} \)  Direct yaw moment reference gain based on the yaw velocity error

\( K_F \)  Roll stiffness value of the front suspension

\( K_{pT,i} \)  Pressure-to-torque coefficient on axle \( i \)

\( K_R \)  Roll stiffness value of the rear suspension

\( k_s \)  Tuning factor for soft constraints on the longitudinal tyre forces

\( K_{s_{ij}} \)  Cornering stiffness of tyre on corner \( ij \)

\( \hat{k} \)  Piecewise affine receding horizon feedback law

\( k \)  Minimum saturation limits of feedback law

\( K_{us} \)  Reference under-steer gradient

\( K_{\kappa} \)  Longitudinal slip stiffness

\( L \)  Lagrange term

\( l_F \)  Front semi-wheelbase

\( l_R \)  Rear semi-wheelbase

\( M \)  Mayer term

\( m \)  Number of input variables

\( m \)  Vehicle mass

\( m_{H/Q} \)  Relevant mass representing half or quarter of the vehicle
\[ M_v \] Vehicle mass

\[ M_z \] Overall yaw moment of longitudinal and lateral tyre forces

\[ M_{z,F_{ij}} \] Yaw moment contribution of longitudinal and lateral tyre forces on corner \( ij \)

\[ M_{z,F_l} \] Overall direct yaw moment of longitudinal tyre forces

\[ M_{z,F_{l,ref}(\alpha_R)} \] Direct yaw moment reference contribution based on the rear slip angle

\[ M_{z,F_{l,ref}(\dot{\psi})} \] Direct yaw moment reference contribution based on the yaw velocity error

\[ M_{z,F_s} \] Overall yaw moment contribution of lateral tyre forces

\[ M_{z,F_{s,F}} \] Yaw moment contribution of lateral tyre forces on the front axle

\[ M_{z,F_{s,R}} \] Yaw moment contribution of lateral tyre forces on the rear axle

\[ N \] Number of generated points

\[ n \] Number of items in search data structure

\[ n \] Number of state variables

\[ N_0 \] Number of additional interior hyper-rectangles

\[ N_c \] Number of control steps

\[ N_e \] Number of edge centres

\[ N_{H,i} \] Number of unique hyperplanes defining polyhedral critical regions in partition \( X_i \)

\[ N_k \] Node of binary search tree

\[ N_{p} \] Number of prediction steps

\[ n_p \] Number of parameters

\[ N_{\dot{\psi}} \] Number of facet centres

\[ N_{R,i} \] Number of polyhedral critical regions in partition \( X_i \)

\[ N_s \] Number of partitions marked to be split

\[ N_{\theta} \] Number of vertices

\[ N_{unsat} \] Number of polyhedral regions with unsaturated feedback law
Nomenclature

\( N_X \)  Number of orthogonal partitions

\( P, p \)  Vector of dimensional/dimensionless system and controller parameters

\( p \)  Number of reference signals

\( P_c, p_c \)  Vector of dimensional/dimensionless controller parameters

\( P_{ch} \)  Characteristic quantities for the system and controller parameters

\( P_{EM,max}, P_{EM,min} \)  Maximum electric motor power in traction and regeneration

\( P_s, p_s \)  Vector of dimensional/dimensionless system parameters

\( P_u \)  Vector of path constraints on the inputs

\( p_u \)  Number of input reference signals

\( P_{ux} \)  Vector of path constraints on the inputs and states

\( P_x \)  Vector of path constraints on the states

\( p_x \)  Weighting matrix for the states in the Mayer term

\( p_y \)  Number of output reference signals

\( q \)  Number of output variables

\( q_{\nu} \)  Weighting matrix for the slack variables in the Lagrange term

\( q_x \)  Weighting matrix for the states in the Lagrange term

\( R, r \)  Vector of dimensional/dimensionless reference signals

\( r \)  Index of polyhedral critical region \( R_j \) that contains the parameter vector \( x_p \)

\( R_0 \)  Unloaded wheel radius

\( R_e \)  Effective radius in free rolling conditions

\( R_{I^{\text{max}}} \)  Sub-set of critical regions with feedback law jointly saturated at maximum

\( R_{I^{\text{min}}} \)  Sub-set of critical regions with feedback law jointly saturated at minimum

\( R_{I^{\text{unsat}}} \)  Sub-set of polyhedral critical regions with unsaturated feedback law

\( R_j \)  Polyhedral critical region in partition \( X_i \)
\( R_l \)  Loaded wheel radius

\( RMS_{a_z} \)  Root-mean-square value of the filtered longitudinal vehicle acceleration

\( RMS_{T_{corr}} \)  Root-mean-square value of the anti-jerk torque correction

\( RMS_{T_{corr,ss}} \)  Root-mean-square value of the steady-state anti-jerk torque correction

\( r_u \)  Weighting matrix for the inputs in the Lagrange term

\( R_{u}, r_u \)  Vector of dimensional/dimensionless input reference signals

\( R_w \)  Wheel radius

\( R_{X_i} \)  Set of polyhedral critical regions in partition \( X_i \)

\( R_{y}, r_y \)  Vector of dimensional/dimensionless output reference signals

\( s \)  Number of decision variables

\( s_{ij} \)  Combined tyre slip on corner \( ij \)

\( S_{min} \)  Minimal allowed volume of any orthogonal partition \( X_0 \) in hyper-rectangle \( X \)

\( S_{X_0} \)  Volume of orthogonal partition \( X_0 \)

\( T \)  Simulation time

\( T_0 \)  Time at the completion of the steering input in the SwD test

\( T_1 \)  Motor torque at the gearbox output shaft

\( T_c \)  Start of the manoeuvre

\( T_1 \)  Time at the beginning of the simulation

\( T_2 \)  Time at the end of the simulation

\( T_2 \)  Time at the start of the calculation of the steady-state indicators

\( T_3 \)  Time at the end of the simulation

\( T_a \)  Time at the start of the normalisation interval for performance indicators

\( T_{aero} \)  Torque contribution related to aerodynamic drag

\( T_{c}, t_c \)  Dimensional/dimensionless control horizon
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{ch}$</td>
<td>Characteristic time</td>
</tr>
<tr>
<td>$T_{corr}$</td>
<td>Anti-jerk torque correction</td>
</tr>
<tr>
<td>$T_d$</td>
<td>Drivetrain torque</td>
</tr>
<tr>
<td>$T_d$</td>
<td>Time at the end of the normalisation interval for performance indicators</td>
</tr>
<tr>
<td>$t_d$</td>
<td>Set of discrete time instants</td>
</tr>
<tr>
<td>$T_{dwell}$</td>
<td>Time of the beginning of the dwell in the steering input in the SwD test</td>
</tr>
<tr>
<td>$T_{e,0}$</td>
<td>Equality constraint functions at linearisation point</td>
</tr>
<tr>
<td>$T_{EM,max}, T_{EM,min}$</td>
<td>Maximum electric motor torque in traction and regeneration</td>
</tr>
<tr>
<td>$T_{EM}$</td>
<td>Electric motor torque</td>
</tr>
<tr>
<td>$T_f, t_f$</td>
<td>Dimensional/dimensionless time at the end of the prediction horizon</td>
</tr>
<tr>
<td>$T_{hs}$</td>
<td>Half-shaft torque</td>
</tr>
<tr>
<td>$T_{i,0}$</td>
<td>Inequality constraint functions at linearisation point</td>
</tr>
<tr>
<td>$T_{k}, t_k$</td>
<td>Current or instantaneous dimensional/dimensionless time</td>
</tr>
<tr>
<td>$T_{p, t_p}$</td>
<td>Dimensional/dimensionless prediction horizon</td>
</tr>
<tr>
<td>$T_{plant}$</td>
<td>Torque request to the plant</td>
</tr>
<tr>
<td>$T_{req}$</td>
<td>Torque request (from a higher level controller)</td>
</tr>
<tr>
<td>$T_{req,tip-in}$</td>
<td>Torque request after tip-in application</td>
</tr>
<tr>
<td>$T_{req,tip-out}$</td>
<td>Torque request before tip-out application</td>
</tr>
<tr>
<td>$T_{roll}$</td>
<td>Torque contribution related to rolling resistance</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Time at the start of the steering signal in the SwD test</td>
</tr>
<tr>
<td>$T_s, t_s$</td>
<td>Dimensional/dimensionless sampling time interval</td>
</tr>
<tr>
<td>$T_w$</td>
<td>Wheel torque</td>
</tr>
<tr>
<td>$U$</td>
<td>Vector of input variable trajectory parameters</td>
</tr>
<tr>
<td>$U, u$</td>
<td>Vector of dimensional/dimensionless input variables</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$U_{ch}$</td>
<td>Characteristic quantities for the inputs</td>
</tr>
<tr>
<td>$\dot{u}$</td>
<td>Sub-optimal control action computed by the receding horizon feedback law</td>
</tr>
<tr>
<td>$U_{sc}$</td>
<td>Scaling factor for the inputs in the cost function</td>
</tr>
<tr>
<td>$V$</td>
<td>Modal matrix of Hessian matrix $H_0$</td>
</tr>
<tr>
<td>$v$</td>
<td>Right eigenvector</td>
</tr>
<tr>
<td>$V$</td>
<td>Cost function or objective function</td>
</tr>
<tr>
<td>$v$</td>
<td>Vehicle speed</td>
</tr>
<tr>
<td>$V_0$</td>
<td>Cost function at linearisation point</td>
</tr>
<tr>
<td>$v_c$</td>
<td>Number of continuous slack variables</td>
</tr>
<tr>
<td>$v_d$</td>
<td>Number of discrete slack variables</td>
</tr>
<tr>
<td>$\dot{v}$</td>
<td>Vehicle tangential acceleration</td>
</tr>
<tr>
<td>$VDV_{ax}^*$</td>
<td>Fourth power vibration dose value of the filtered longitudinal vehicle acceleration</td>
</tr>
<tr>
<td>$\hat{V}$</td>
<td>Quadratic approximate cost function of local mp-QP problem</td>
</tr>
<tr>
<td>$v_r$</td>
<td>Linear speed of rolling</td>
</tr>
<tr>
<td>$v_{sx}$</td>
<td>Longitudinal slip speed</td>
</tr>
<tr>
<td>$v_x$</td>
<td>Longitudinal component of the vehicle speed in the wheel reference system</td>
</tr>
<tr>
<td>$v_x$</td>
<td>Longitudinal vehicle speed</td>
</tr>
<tr>
<td>$V_{x0}$</td>
<td>Jacobian of cost function w.r.t. parameters at linearisation point</td>
</tr>
<tr>
<td>$v_{x,0}$</td>
<td>Initial longitudinal vehicle speed</td>
</tr>
<tr>
<td>$V_{xx0}$</td>
<td>Hessian of cost function w.r.t. parameters at linearisation point</td>
</tr>
<tr>
<td>$w$</td>
<td>Vector with coefficients for cost function discretisation with trapezoidal rule</td>
</tr>
<tr>
<td>$w_{1-6}$</td>
<td>Weights on the performance indicators</td>
</tr>
<tr>
<td>$w_0$</td>
<td>Chebyshev centre of polyhedron</td>
</tr>
<tr>
<td>$W_0$</td>
<td>Set of generated points of hyper-rectangle $X_0$</td>
</tr>
</tbody>
</table>
Nomenclature

\( W_c \)  
Weight on the comfort oriented performance indicators

\( w_c \)  
Number of (inequality and equality) constraint functions

\( w_{c,e} \)  
Number of equality constraint functions

\( w_{c,i} \)  
Number of inequality constraint functions

\( w_d \)  
Number of discrete (inequality and equality) constraint functions

\( w_{d,b_v} \)  
Number of discrete bound constraints on the slack variables

\( w_{d,b_u} \)  
Number of discrete bound constraints on the inputs

\( w_{d,b_s} \)  
Number of discrete bound constraints on the states

\( w_{d,d_{ux}} \)  
Number of discrete dynamic equality constraints

\( w_{d,e} \)  
Number of discrete equality constraint functions

\( w_{d,i} \)  
Number of discrete inequality constraint functions

\( w_{d,p_u} \)  
Number of discrete path constraints on the inputs

\( w_{d,p_{ux}} \)  
Number of discrete path constraints on the inputs and states

\( w_{d,p_s} \)  
Number of discrete path constraints on the states

\( W^f \)  
Sub-set of points at which a feasible solution of the corresponding NLP was found

\( w_i \)  
Generated point included in the set of points \( W_0 \) of hyper-rectangle \( X_0 \)

\( w_\rho \)  
Weighting vector for solution error bound

\( W_s \)  
Weight on the sport oriented performance indicators

\( \dot{X} \)  
Vector of state variable trajectory parameters

\( X, x \)  
Vector of dimensional/dimensionless state variables

\( \bar{X} \)  
Closed poly-topic set of parameters
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{X}$</td>
<td>Hyper-rectangle of the parameter space to be explored</td>
</tr>
<tr>
<td>$X$</td>
<td>Orthogonal partition of the parameter space</td>
</tr>
<tr>
<td>$x$</td>
<td>Index of orthogonal partition $X_i$ that contains the parameter vector $x_p$</td>
</tr>
<tr>
<td>$X_0$</td>
<td>Selected orthogonal partition of the parameter space</td>
</tr>
<tr>
<td>$X_0'$</td>
<td>Additional interior hyper-rectangle of hyper-rectangle $X_0$</td>
</tr>
<tr>
<td>$X_{ch}$</td>
<td>Characteristic quantities for the states</td>
</tr>
<tr>
<td>$X_f$</td>
<td>Set of feasible parameters</td>
</tr>
<tr>
<td>$X_p, x_p$</td>
<td>Vector of dimensional/dimensionless parameters</td>
</tr>
<tr>
<td>$x_{p,0}$</td>
<td>Parameter vector at the linearisation/centre point</td>
</tr>
<tr>
<td>$\bar{x}_p$</td>
<td>Upper bound defining the orthogonal parameter space to be explored</td>
</tr>
<tr>
<td>$X_{p,ch}$</td>
<td>Transformation vector of characteristic quantities for the vector of parameters</td>
</tr>
<tr>
<td>$\tilde{x}_p$</td>
<td>Parameter vector clipped to the bounds of hyper-rectangle $X$</td>
</tr>
<tr>
<td>$x_p$</td>
<td>Lower bound defining the orthogonal parameter space to be explored</td>
</tr>
<tr>
<td>$X_{sc}$</td>
<td>Scaling factor for the states in the cost function</td>
</tr>
<tr>
<td>$Y$</td>
<td>Matrix for quadratic term of parameters in approximate cost function</td>
</tr>
<tr>
<td>$Y, y$</td>
<td>Vector of dimensional/dimensionless output variables</td>
</tr>
<tr>
<td>$Y_0$</td>
<td>Constant term in cost function of local mp-QP problem for fixed parameters</td>
</tr>
<tr>
<td>$Z, z$</td>
<td>Vector of dimensional/dimensionless decision variables</td>
</tr>
<tr>
<td>$z_0$</td>
<td>Close-to-global solution of NLP problem at the linearisation/centre point</td>
</tr>
<tr>
<td>$Z_{ch}$</td>
<td>Transformation vector of characteristic quantities for the vector of decision variables</td>
</tr>
<tr>
<td>$\hat{z}$</td>
<td>Piecewise affine solution of approximating mp-QP problem</td>
</tr>
<tr>
<td>$\hat{z}_{X_i}$</td>
<td>Piecewise affine solution of approximating mp-QP problem in partition $X_i$</td>
</tr>
<tr>
<td>$z^*$</td>
<td>Global solution of NLP problem</td>
</tr>
</tbody>
</table>
Greek Symbols

\( \alpha \)  Tyre slip angle

\( \alpha_{ij} \)  Tyre slip angle on corner \( ij \)

\( \alpha_F \)  Front axle slip angle

\( \alpha_R \)  Rear axle slip angle

\( \alpha_{R,\text{max}} \)  Maximum bound on rear slip angle

\( \alpha_{R,\text{min}} \)  Minimum bound on rear slip angle

\( \beta \)  Vehicle side-slip angle

\( \dot{\beta} \)  Vehicle side-slip angle rate

\( \beta_F \)  Vehicle side-slip angle on the front axle

\( \beta_R \)  Vehicle side-slip angle on the rear axle

\( \delta \)  Terminal set

\( \Delta \)  Finite difference

\( \delta \)  Maximum constraint violation

\( \bar{\delta} \)  Steering angle

\( \bar{\delta} \)  Arithmetic mean of constraint violations at all points within considered partitions

\( \tilde{\delta} \)  Penalty term for the constraint violation

\( \delta_e \)  Maximum equality constraint violation

\( \hat{\delta}_e \)  Estimation of maximum equality constraint violation

\( \hat{\delta} \)  Estimation of maximum constraint violation

\( \delta_i \)  Maximum inequality constraint violation

\( \hat{\delta}_i \)  Estimation of maximum inequality constraint violation

\( \delta_{sw} \)  Steering wheel angle

\( \hat{\delta}_{sw} \)  Steering wheel angle rate
\( \hat{\delta}_{sw} \) Steering wheel angle amplitude
\( \Delta \theta \) Drivetrain torsion angle
\( \Delta \dot{\theta} \) Drivetrain torsion rate
\( \hat{\delta} \) Tolerance for the maximum constraint violation
\( \epsilon^j_i \) Edge centre included in the set of edge centres \( \mathcal{E}^j \) of hyper-rectangle \( X_j \)
\( \epsilon \) Weighted sum of mean approximation errors of two newly created hyper-rectangles
\( \mathcal{E}^j \) Set of edge centres of hyper-rectangle \( X_j \)
\( \epsilon \) Cost error bound
\( \bar{\epsilon} \) Arithmetic mean of cost errors at all points within considered partitions
\( \hat{\epsilon} \) Penalty term for the cost error
\( \hat{\epsilon} \) Estimation of cost error bound
\( \hat{\epsilon} \) Tolerance for the approximation error of the cost function
\( \eta \) Vector of parametrised slack variable trajectory functions
\( \eta \) Drivetrain efficiency
\( \kappa_{ij} \) Longitudinal tyre slip on corner \( ij \)
\( \kappa'_x \) Delayed longitudinal slip
\( \kappa_{ref} \) Longitudinal slip reference
\( \kappa_x \) Longitudinal slip
\( \lambda \) Eigenvalue
\( \lambda_{\mu} \) Scaling factor for tyre-road friction coefficient
\( \mu \) Vector of parametrised input variable trajectory functions
\( \mu \) Tyre-road friction coefficient
\( \mu_{cij} \) Combined friction utilisation on corner \( ij \)
\( \mu_{lij} \) Longitudinal friction utilisation on corner \( ij \)
\( \mu_{lp_{ij}} \)  Longitudinal peak friction coefficient on corner \( ij \)

\( \mu_{tm} \)  Tyre-road friction coefficient in tyre testing conditions

\( \mu_{s_{ij}} \)  Lateral friction utilisation on corner \( ij \)

\( \mu_{sp_{ij}} \)  Lateral peak friction coefficient on corner \( ij \)

\( \mathbf{N} \)  Vector of slack variable trajectory parameters

\( \mathbf{N}, \mathbf{\nu} \)  Vector of dimensional/dimensionless slack variables

\( N_{\alpha_{R}} \)  Slack variable for soft constraints on the rear slip angle

\( \mathbf{N}_{ch} \)  Characteristic quantities for the slack variables

\( \dot{N}_{\psi_{\phi}} \)  Slack variable for soft constraints on the yaw velocity error

\( N_{F_{t_{l}}} \)  Slack variable for soft constraints on the longitudinal tyre forces

\( N_{F_{e}} \)  Slack variable for soft constraint on the vehicle braking force

\( N_{s_{x}} \)  Slack variable for soft constraints on the longitudinal slip

\( N_{sc} \)  Scaling factor for the slack variables in the cost function

\( \Omega \)  Terminal set

\( \Omega \)  Angular wheel speed

\( \Phi \)  Vector of parametrised reference signal functions

\( \phi \)  Clipping function

\( \phi \)  Vector of parametrised numerical ODE solution functions

\( \Pi \)  Set of orthogonal partitions

\( \psi_{i}^{j} \)  Facet centre included in the set of facet centres \( \Psi^{j} \) of hyper-rectangle \( X_{j} \)

\( \Psi^{j} \)  Set of facet centres of hyper-rectangle \( X_{j} \)

\( \dot{\psi} \)  Vehicle yaw velocity

\( \dot{\psi}_{\max} \)  Maximum achievable yaw velocity

\( \dot{\psi}_{\text{peak}} \)  First peak yaw velocity in the SwD test
Nomenclature

\( \dot{\psi}_{\text{ref}} \) Yaw velocity reference

\( \dot{\psi}_{\text{ref,lin}} \) Linear yaw velocity reference

\( \rho \) Solution error bound

\( \rho \) Air density

\( \rho \) Vertical tyre deflection

\( \rho_0 \) Vertical tyre deflection at nominal wheel load

\( \bar{\rho} \) Arithmetic mean of solution errors at all points within considered partitions

\( \hat{\rho} \) Penalty terms for the solution errors

\( \hat{\rho} \) Estimation of solution error bounds

\( \tilde{\rho} \) Tolerance for the approximation errors of the solution

\( \sigma_\kappa \) Relaxation length

\( \tan(\alpha_{ij}) \) Lateral tyre slip on corner \( ij \)

\( \tau \) Frequency of the sine in the SwD steering signal

\( \theta_i \) Vertex \( i \) included in the set of vertices \( \Theta_i \) of hyper-rectangle \( X_j \)

\( \Theta_j \) Set of vertices of hyper-rectangle \( X_j \)

\( \theta_1 \) Angular speed of the gearbox output

\( \theta_2 \) Average angular speed of the front wheels

\( \theta_{BL} \) Nominal backlash at the wheel

\( \dot{\theta}_{EM} \) Electric motor speed

\( \dot{\theta}_{EM,\text{vib}} \) Oscillating component of the electric motor speed

\( \dot{\theta}_w \) Driven wheel speed

Superscripts

- Average or mean value of a quantity
- Upper bound of quantity
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Clipping-based complexity reduction</td>
</tr>
<tr>
<td>\cdot</td>
<td>Parameters clipped to some bounds</td>
</tr>
<tr>
<td>\cdot</td>
<td>Quantity after complexity reduction</td>
</tr>
<tr>
<td>.</td>
<td>Time derivative</td>
</tr>
<tr>
<td>est</td>
<td>Quantity based on the prediction model of the NMPC controller</td>
</tr>
<tr>
<td>exp</td>
<td>Quantity based on experimental test results with the real vehicle demonstrator</td>
</tr>
<tr>
<td>g</td>
<td>Greedy merging complexity reduction</td>
</tr>
<tr>
<td>\hat{}</td>
<td>Estimated quantity</td>
</tr>
<tr>
<td>o</td>
<td>Optimal merging complexity reduction</td>
</tr>
<tr>
<td>\hat{}</td>
<td>Approximate or sub-optimal value</td>
</tr>
<tr>
<td>sim</td>
<td>Quantity based on the high-fidelity simulation model</td>
</tr>
<tr>
<td>*</td>
<td>Optimal value of a quantity</td>
</tr>
<tr>
<td>\tilde{}</td>
<td>Receding horizon equivalent</td>
</tr>
</tbody>
</table>

**Subscripts**

<table>
<thead>
<tr>
<th>Subscript</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Linearisation point for Taylor series expansion</td>
</tr>
<tr>
<td>ch</td>
<td>Characteristic quantity</td>
</tr>
<tr>
<td>i = F,R</td>
<td>Front and rear axle</td>
</tr>
<tr>
<td>j = L,R</td>
<td>Left and right side</td>
</tr>
<tr>
<td>\nu</td>
<td>Slack variable of the optimal control problem</td>
</tr>
<tr>
<td>ref</td>
<td>Reference value</td>
</tr>
<tr>
<td>sc</td>
<td>Scaling factor of the cost function</td>
</tr>
<tr>
<td>.</td>
<td>Lower bound of quantity</td>
</tr>
</tbody>
</table>

**Other Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Bold symbols represent matrices</td>
</tr>
</tbody>
</table>
**Nomenclature**

**Bold symbols represent vectors**

⊗ Dyadic product

\[ \nabla \] Gradient

\[ \frac{\partial}{\partial x} \] Partial derivative with respect to variable x

Q Capital letters for dimensional quantities

q Regular letters for dimensionless quantities

S Regular symbols represent scalars

**Acronyms / Abbreviations**

ABS Anti-Lock Braking System

AJC Anti-Jerk Control

ASIL Automotive Safety Integrity Level

BFGS Broyden-Fletcher-Goldfarb-Shanno

CPU Central Processing Unit

DOCR Disjoint Optimal Complexity Reduction

DoF Degree of Freedom

ECU Electronic Control Unit

EHB Electro-Hydraulic Braking

EM Electric Motor

ESC Electronic Stability Control

EU European Union

FPGA Field Programmable Gate Array

FWD Front-Wheel Drive

GVS Geometric Vertex Search

HIL Hardware-in-the-Loop
Nomenclature

ICE Internal Combustion Engine
IP Interior Point
IPOPT Interior Point OPTimizer
k-d K-Dimensional
KKT Karush-Kuhn-Tucker
LICQ Linear Independence Constraint Qualification
LP Linear Programming
LPV-MPC Linear Parameter-Varying Model Predictive Control
LPV Linear Parameter-Varying
LQG Linear-Quadratic Gaussian
LQR Linear-Quadratic Regulator
LTI Linear Time-Invariant
LTV-MPC Linear Time-Varying Model Predictive Control
LTV Linear Time-Varying
MF Magic Formula
MILP Mixed-Integer Linear Programming
MIMO Multiple-Input Multiple-Output
MIQP Mixed-Integer Quadratic Programming
mp-LP Multi-Parametric Linear Programming
mp-MINLP Multi-Parametric Mixed-Integer Non-linear Programming
mp-MIQP Multi-Parametric Mixed-Integer Quadratic Programming
mp-NLP Multi-Parametric Non-Linear Programming
mp-OA Multi-Parametric Outer Approximation
mp-Q/OA Multi-Parametric Quadratic/Outer Approximation
mp-QA Multi-Parametric Quadratic Approximation
mp-QP Multi-Parametric Quadratic Programming
mp-B&B Multi-Parametric Branch and Bound
MPC Model Predictive Control
MPT Multi-Parametric Toolbox
NLP Non-linear Programming
NLSENS Non-linear Sensitivity
NMPC Non-linear Model Predictive Control
NOCR Non-disjoint Optimal Complexity Reduction
ODE Ordinary Differential Equation
PDIP Primal-Dual Interior Point
PID Proportional-Integral-Derivative
PI Proportional-Integral
PLCP Parametric Linear-Complementary Problem
POP Parametric Optimisation
P Proportional
PSO Particle Swarm Optimisation
PWA Piecewise Affine
QP Quadratic Programming
RAM Random-Access Memory
RCP Rapid Control Prototyping
RMS Root-Mean-Square
RTI Real-Time Iteration
SIL Software-in-the-Loop
SISO  Single-Input Single-Output
SQP  Sequential Quadratic Programming
SUV  Sport-Utility Vehicle
SwD  Sine with Dwell
TCS  Traction Control System
TC  Traction Control
TV  Torque Vectoring
UKF  Unscented Kalman Filter
VDC  Vehicle Dynamics Control
VDV  Vibration Dose Value
VSC  Vehicle Stability Control
Chapter 1

Introduction

1.1 Motivation

Model predictive control (MPC) is a widely accepted and state-of-the-art control methodology. It is an optimisation-based method for controlling multiple-input multiple-output (MIMO) systems and allows to incorporate physical and operational constraints in a systematic manner. With an increasing number of active automotive systems, MPC helps to tackle new challenges for the integration and coordination of conceivably over-actuated systems requiring MIMO control methods. Moreover, MPC allows for the design of multi-variable feedback controllers with similar procedural complexity as of single-variable ones, thus, contributing to a reduction of the design and tuning effort for shorter development times. Constraints on the system inputs, states, and outputs can be systematically accounted for in the design phase, which are then enforced by the controller. The model predictive control strategy in a receding horizon fashion involves the on-line solution of a finite horizon optimal control problem at each discrete sampling instant. There is, therefore, also some philosophical attractiveness to MPC mimicking many processes in nature that seem to inherently operate in a similar way embodying both optimisation and feedback adjustment.

Non-linear model predictive control (NMPC) allows to include crucial non-linearities of the system dynamics in the formulation of the optimal control problem. This is particularly important for systems where a linear prediction model cannot model the non-linear system dynamics with sufficient accuracy over the required operating range of the controller. Apart from NMPC, there are other control design techniques for non-linear systems which are characterised by different hurdles. Feedback linearisation introduces auxiliary non-linear feedback employing a non-linear coordinate transformation such that linear control methods can be applied. However, certain properties of the system must hold and the difficulty of this approach lies in finding a global diffeomorphism of the
system to be controlled. An adaptive control method for non-linear systems is recursive backstepping by designing a sequential control for each state of the non-linear differential equations of the system. Recursive backstepping has limitations and can yield several different designs based on the experience of the control engineer and the insight into the system. Sliding control is a variable structure control for non-linear systems. The method does not need a precise model of the process and generally leads to good performance and robustness. However, the method can cause high-frequency control chattering which may excite structural modes thereby degrading system performance and compromising comfort. A further method for non-linear systems is the control Lyapunov function approach which is, however, characterised by the difficulty of finding the underlying control Lyapunov function.

In the context of MPC, the incorporation of non-linearities leads to a non-linear optimisation problem based on a cost function subject to the non-linear system dynamics and possibly non-linear constraints on the inputs, states, and outputs of the system. However, the online solution of a non-linear optimisation problem can be computationally demanding and time-consuming. It is, therefore, usually limited to rather slow processes, e.g. industrial applications for chemical process control, where the sampling interval is sufficiently large. The online computational burden can be considerably reduced by using an explicit approach to NMPC, in which the optimisation problem is solved off-line, and the resulting explicit approximate solution, with a guaranteed level of sub-optimality, is evaluated online. To this end, the optimal control problem is reformulated as multi-parametric non-linear programming (mp-NLP) problem. As the exact solution of an mp-NLP problem is in general not possible, a local quadratic approximation of the cost function and a linear approximation of the constraint functions lead to a multi-parametric quadratic program (mp-QP), with the accuracy of the approximation controlled via iterative and recursive partitioning of the parameter space. The sub-optimal solution provides locally accurate explicit state feedback laws avoiding the need for the real-time execution of a numerical optimisation routine.

The explicit representation of the solution can be expressed as piecewise affine state feedback defined on polyhedral regions of the state-space. The efficient online evaluations of this formulation enable the real-time execution of the controller with the required sampling frequency. The reduction of the online software complexity, compared to non-linear numerical optimisation, leads to a reduction in the requirements on the embedded automotive control hardware. Moreover, the explicit nature of the control law allows for verification of the implementation including infeasibility handling and functional safety validation, posing an essential issue for safety-critical applications. The complete representation of the solution facilitates rigorous and a priori analysis of performance, sub-optimality, and stability.
To demonstrate the multiple benefits of explicit non-linear model predictive control, three representative case studies are presented. The selected automotive applications are characterised by the presence of non-linear and relatively fast system dynamics and both physical and operational constraints in a safety-critical context. The longitudinal, lateral, and yaw dynamics of a vehicle are strongly influenced by the highly non-linear behaviour of the tyre forces depending on the slip conditions. Thus, the consideration of the non-linear system dynamics in the controller design is expected to bring significant benefits for the control system performance compared to linear methods. In particular, in safety-critical applications, every performance enhancement has an important impact.

In the proposed vehicle stability control studies, the friction brakes are controlled to stabilise the lateral and yaw dynamics of a vehicle at the limits of handling. Each of the four electro-hydraulically actuated friction brakes of the case study vehicle can generate only braking torques. This can be systematically modelled in a model predictive control approach formulating appropriate constraints. Moreover, the inherent possibility to deal with MIMO structures makes MPC an attractive choice for the control of the over-actuated system.

The relevant dynamics in the frequency range of 10 to 20 Hz in the wheel slip control studies require sampling times typically of few milliseconds to ensure adequate performance and stability. An explicit approach facilitates the real-time implementation of the non-linear MPC modelling the highly non-linear behaviour of the tyre forces. Moreover, the increased control bandwidth and precision in torque modulation adopted by electric drivetrains can be fully exploited by the explicit NMPC to significantly enhance the performance of wheel slip control systems. The MPC approach allows a further appealing possibility of straight-forward handling and compensation of delays in the control system.

Apart from the non-linear tyre dynamics, the drivetrains of electric vehicles with on-board motors are typically characterised by torsional compliance and inevitable backlash inducing oscillations in the 5 to 50 Hz frequency range. In the anti-jerk control studies, the non-linearity of the backlash can be formally incorporated in the control system design besides physical limitations of the electric motor.

### 1.2 Research contributions

The main focus of the PhD research presented in this thesis is on the development of an explicit non-linear model predictive control framework for automotive applications. The research contributions can be summarised as follows:

- The development of an algorithm for the approximate solution of multi-parametric non-linear programming problems using multi-parametric quadratic programming
approximations (mp-QA) incorporating a combination of well-performing elements of established algorithms.

• A novel approach for robust post-processing of the explicit approximate solution of mp-NLP problems using mp-QP approximations employing readily available and tested tools for a straightforward implementation.

• The development of an explicit non-linear MPC-based vehicle stability controller including the effects of load transfer in braking and cornering and the interaction of longitudinal and lateral tyre forces. The study includes a systematic simulation-based comparison of different prediction models investigating the influences of varying model fidelity, e.g. tyre non-linearities, the coupling of tyre forces, and incorporation of load transfer effects, on the controller performance. Moreover, the influences of cost function weights, formulation of rear slip angle constraints, the controller sampling time, and additional delays caused by non-trivial execution times on the performance of the controller are analysed and the robustness assessed in various conditions.

• The development of explicit NMPC-based wheel slip controllers for traction control and anti-lock braking scenarios. The study on traction control includes a simulation-based analysis of the performance benefits of the explicit NMPC compared to a benchmark PI (proportional-integral) controller and a sensitivity analysis investigating the influence of the sampling time on the controller performance. Furthermore, the feasibility of hardware implementation and experimental validation is demonstrated with test results on a fully electric vehicle prototype. In the study on traction control in cornering, the performance enhancements in terms of slip tracking as a result of a combined slip non-linear tyre model is demonstrated compared to a pure longitudinal slip tyre model. Finally, the study on anti-lock braking presents the design of an explicit NMPC-based continuous wheel slip controller including a delay compensation strategy and its experimental implementation on a hardware-in-the-loop test-rig with an electro-hydraulic braking system. The performance is analysed compared to a benchmark PI controller and the robustness is assessed in various conditions.

• The development of an explicit NMPC-based anti-jerk controller including a non-linear backlash model in the controller formulation. A comparison of different prediction models with varying fidelity, e.g. backlash non-linearity, wheel dynamics, and tyre dynamics, and a sensitivity analysis with variations in the vehicle speed and torque requests demonstrate the influences of the model complexity and controller settings on the controller performance. Objective performance indicators are introduced to quantify the performance and establish an optimisation-based tuning procedure. Moreover, a comparison of the explicit NMPC-based anti-jerk controller with other
control methodologies shows the differences in terms of control system performance, robustness and design, and utilised input signals.

- The analysis of the explicit solutions. The three case studies allow to investigate the influences of various aspects in the design process on the complexity of the explicit controller in terms of polyhedral critical regions and subsequently memory requirements. This includes the influences of the definition of the parameter exploration space, the number of parameters (number of system states and reference parameters), the number of system inputs, the definition of constraints, and the formulation of non-linearities. Moreover, the effectiveness of the post-processing procedure, based on binary search tree generation, clipping-based complexity reduction, and merging procedures, is demonstrated.

- The development and implementation of an explicit non-linear model predictive control toolbox was also in the scope of the research activity. The optimal control problem formulation and the non-dimensionalisation procedure, with all its necessary symbolic manipulations, were implemented in the computing environment Maple. A versatile framework was implemented in MATLAB, managing different definitions of prediction models and controller set-ups by storing them in a library. The implementation in MATLAB includes the mp-NLP algorithm with different heuristic splitting rules. For the solution of the NLP problems, the software package IPOPT (Wächter and Biegler, 2006) is embedded in the MATLAB framework. Furthermore, the MPT 3.0 toolbox (Herceg et al., 2013) is employed for the solution of the mp-QP problems and the post-processing algorithm including complexity reduction and search data handling.

1.3 List of publications

The publications that arose from the PhD research work can be summarised as follows:

Vehicle stability control


**Wheel slip control**


**Anti-jerk control**


1.4 Structure of the thesis

This thesis is divided into two parts introducing the methodology on explicit non-linear model predictive control in Part I and presenting the application of explicit NMPC to three representative automotive case studies in Part II.

Chapter 2 of Part I gives an introduction and an overview of the relevant literature on the topics of multi-parametric non-linear and quadratic programming, data handling, and complexity reduction methods.
In Chapter 3, important concepts of non-linear model predictive control are introduced. Starting with the definition of a continuous-time optimal control problem, the use of non-dimensionalisation and scaling, and the application of finite parametrisation and discretisation leads to an mp-NLP problem in general form. Subsequently, the numerical solution of the optimisation problem is addressed, employing appropriate mathematical programming methods, and some comments on the tuning and stability of model predictive controllers are given.

Chapter 4 introduces the explicit non-linear model predictive control approach based on the sub-optimal solution of mp-NLP using mp-QP approximations. The local mp-QP approximations to mp-NLP are defined, the structure of the mp-QA algorithm is explained, and heuristic splitting rules for the recursive sub-partitioning of the parameter space are presented. Then, a post-processing algorithm including complexity reduction methods and search data handling is suggested.

In Chapter 5 of Part II, explicit NMPC for vehicle stability control is presented. A detailed review of the relevant literature is followed by the control system design introducing different prediction models. The performance analysis includes the definition of objective indicators besides various studies on the influences of prediction model complexity, tuning weights, controller sampling and execution time, and parameter variations. The explicit solution is analysed and the effectiveness of the post-processing is investigated.

Chapter 6 presents three studies on explicit NMPC-based wheel slip control. For the traction control scenario, a review of the literature, and the main contributions and conclusions are presented. For the traction control scenario in cornering, the control system design is explained and the performance of the controller is analysed. An analysis of the explicit solution and the post-processing results is carried out. For the anti-lock braking scenarios, a brief literature review is given and followed by the main contributions and conclusions of the study.

In Chapter 7, an explicit NMPC-based approach to anti-jerk control is presented. After a concise literature review, the optimal control problem based on various prediction models is introduced. Then, the performance and robustness of different controllers are assessed with objective indicators and the explicit solution and post-processing results are analysed.

Summarising the findings, Chapter 8 formulates the most important conclusions of the research work. Concluding remarks and comments on future developments complete the thesis.
1.5 Personal contribution

Chapter 3 and Chapter 4 are strongly related to the work of Alexandra Grancharova and Tor Arne Johansen. Substantial parts in the two chapters are adapted and included for readability and ease of understanding of the thesis and are, therefore, not the author’s own work. Those parts are, however, clearly indicated in the text. Moreover, the entire Part I should be seen as necessary foundation for Part II which presents the actual contribution of this thesis.

Chapter 5 is based on Metzler et al. (2019c), Metzler et al. (2018), and Metzler et al. (2019b). All three publications are entirely the author’s work besides the support of the co-authors in the revision of the manuscripts.

Chapter 6 is based on Tavernini et al. (2019a), Metzler et al. (2019a), and Tavernini et al. (2019b). In particular, Sections 6.1 and 6.3 are subjected to ©2018-2019 IEEE copyright. The publication Tavernini et al. (2019a) emerged from a collaboration with the co-authors with personal contributions in the control system design and the process of writing and reviewing the paper. The publication Metzler et al. (2019a) is the author’s work apart from the support of the co-authors concerning the simulation model implementation and the revision of the manuscript. The publication Tavernini et al. (2019b) comes from a collaboration with the co-authors with personal contributions in the control system design and the process of writing and reviewing the paper.

Chapter 7 is based on Scamarcio et al. (2019a) and Scamarcio et al. (2019b). Both publications emerged from a collaboration with the co-authors with shared and equally apportioned contributions to the entire development of the NMPC-based controllers and the process of writing and reviewing the papers.
Part I

Methodology
Chapter 2

Introduction and literature review

Non-linear model predictive control (NMPC) requires the formulation of an optimal control problem involving a cost function and constraint functions incorporating the system dynamics. Starting from the continuous formulation, the application of discretisation and finite parametrisation procedures, also known as numerical optimal control, leads to a parametrised and discrete formulation of the optimal control problem. This formulation can also be described more generally as a multi-parametric non-linear mathematical program (mp-NLP) dependent on a parameter vector, $x_p$, appearing in the objective function and the constraint functions. The mp-NLP problem can be generally defined as

$$V^*(x_p) = \min_{z} (z, x_p),$$

subject to

$$G(z, x_p) \leq 0,$$

where $z \in \mathbb{R}^s$ is the vector of decision variables, $x_p \in \mathbb{R}^{n_p}$ is the vector of parameters, $V: \mathbb{R}^s \times \mathbb{R}^{n_p} \mapsto \mathbb{R}$ is the objective function, and $G: \mathbb{R}^s \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{w_d}$ are the constraint functions. Unlike the non-parametric case where only one static solution is computed, the solution of a multi-parametric program comprises the following triple of information:

- Exact or approximate expression of the optimal value function $V^*(x_p)$ as a function of the parameters.
- Exact or approximate expression of the optimisation variables $z^*(x_p)$ as a function of the parameters.
- Exact or approximate characterisation of the set of feasible parameters $X_f$, i.e. the critical regions, where the above expressions are valid and admit a solution to the problem (2.1)-(2.2).
For multi-parametric non-linear programming problems it is a very challenging or even impossible task to derive an exact solution. For this reason, all major efforts in the literature have focused on approximate solutions. In the following, various algorithmic developments are critically discussed. In Domínguez et al. (2010), recent developments of approximate mp-NLP are summarised and four different approaches are compared to each other. Similarly, in Domínguez and Pistikopoulos (2011), three different algorithms are presented and their performance is shown through an illustrative example.

Convex multi-parametric non-linear programming problems

For convex mp-NLP problems, a recursive approximating algorithm is proposed in Bemporad and Filippi (2006) and further addressed in Domínguez et al. (2010). The partitioning of the parameter space is done with a set of simplices. Inside each simplex, the mp-NLP problem is approximated via linear interpolation of the NLP solution at the vertices of the simplex.

Another possible approach is the multi-parametric outer approximation (mp-OA) algorithm, as proposed in Dua and Pistikopoulos (1999), where the mp-NLP problem is transformed into a multi-parametric linear programming (mp-LP) problem using continuous first-order linearisation of the cost and constraint functions. As pointed out in Domínguez et al. (2010), even for a small demonstration example with mild non-linearities, the algorithm requires many outer approximations in order to obtain an optimal value function meeting the required tolerances. A similar approach can be found in Acevedo and Salgueiro (2003) proposing an algorithm based on outer approximations involving heuristics using the solution of NLP sub-problems and master mp-LPs.

The approximation of convex mp-NLP problems can also be achieved via a geometric vertex search (GVS) algorithm, as proposed in Narciso (2009), where the parameter space is partitioned into a set of hypercubes and simplices, including the identification of vertices of the critical region boundaries, where a combination of critical sets occurs. According to Domínguez et al. (2010), the problem of correctly identifying the active sets can become challenging with problem formulations involving higher non-linearities of the system dynamics and non-linear constraints, respectively.

Convex mp-NLP problems can also be approximated with an approximate mp-NLP algorithm on orthogonal partitions, as proposed in Johansen (2004), solving only NLP sub-problems at the vertices of the hypercubes and incorporating heuristic splitting rules, as similarly used in Grancharova and Johansen (2002). The approximating function is piecewise affine on orthogonal state space partitions that can be represented by an efficient binary search tree called k-dimensional (k-d) tree. A detailed description of the algorithm can also be found in Grancharova and Johansen (2012). A modification of the approximate mp-NLP algorithm in Johansen (2004) was proposed in Domínguez and
Pistikopoulos (2011) using simplicial partitions instead of orthogonal partitions of the state-space in order to reduce the number of partitions being required to guarantee the feasibility of the linear approximation. This adaptation leads to a similar concept as already mentioned and proposed in Bemporad and Filippi (2006).

Another algorithm using orthogonal partitions is shown in Johansen (2002) representing a multi-parametric quadratic approximation (mp-QA) algorithm locally approximating the mp-NLP problem with multi-parametric quadratic programming (mp-QP) sub-problems. The approximating mp-QP problems involve a quadratic approximation of the cost function and linear approximation of the constraint functions and can be solved exactly using the method proposed in Tøndel et al. (2003a) on each orthogonal partition. The solution leads to an explicit affine feedback corresponding to a fixed active set within each polyhedral critical region. The approximation error can be determined by solving NLPs at the vertices of the orthogonal partition and using heuristic splitting rules when the tolerances are violated. Similar to above, a detailed description can also be found in Grancharova and Johansen (2012). The same algorithm is revisited in Domínguez and Pistikopoulos (2011) and shown through a demonstration example. The application of the mp-QA algorithm leads to the least amount of polyhedral regions with associated approximate solutions within their corresponding error bounds. As pointed out in Domínguez and Pistikopoulos (2011), the exploration based on hypercubes allows the use of an orthogonal search tree enabling very efficient online search.

A modification of the multi-parametric quadratic approximation algorithm above is presented in Domínguez et al. (2010) locally approximating the cost function with a quadratic approximation and the constraints with linear approximations being accumulated in case of infeasibility of the parameter space for subsequent iterations. This can be summarised as a multi-parametric quadratic/outer approximation (mp-Q/OA) algorithm. The demonstration example in Domínguez et al. (2010) is not very representative as the algorithm stops after one iteration not performing any partitioning of the parameter space. A related algorithm can be found in Domínguez and Pistikopoulos (2013) for multi-parametric mixed-integer non-linear programming (mp-MINLP) problems.

Another way to approximate mp-NLP problems is the proposed non-linear sensitivity (NLSENS) based algorithm in Domínguez and Pistikopoulos (2010) partitioning the parameter space into a set of poly-topical approximate critical regions representing the active set changes after perturbing the parameters. The algorithm is based on local sensitivity analysis of the mp-NLP problem and successive linearisation of the non-linear dynamic system and the non-linear constraints at different points of the parameter space. Domínguez and Pistikopoulos (2011) show through a demonstration example that a post-processing step cannot form convex unions out of most of the critical regions despite having the same feedback law.
Non-convex multi-parametric non-linear programming problems

Many practical NMPC applications involve non-convex expressions, which can be identified as such by the violation of positive definiteness of the second order partial derivatives. However, the approximation of non-convex mp-NLP problems is another major challenge. Explicit NMPC offers the possibility to move offline the computational demand to derive the global solution arising with the non-convex expressions, being a critical issue for the online solution of NMPC. The use of global optimisation techniques embedded in mp-NLP algorithms or efficient initialisation procedures combined with local solvers leading to close-to-global solutions as a practical approach can be employed. Workable computational methods proposed for the explicit approximate solution of mp-NLP do not necessarily lead to guaranteed properties of the explicit solution or have some other kind of limitations. However, when combined with verification and analysis methods they may give a viable tool for an explicit approximate solution of non-convex mp-NLPs.

For non-convex mp-NLP problems, a multi-parametric branch and bound (mp-B&B) algorithm is proposed in Dua et al. (2004) obtaining a parametric upper and lower bound via underestimating and overestimating mp-LP sub-problems. It can be applied for the solution of multi-parametric continuous and mixed-integer optimisation problems.

Another possible approach for non-convex mp-NLP problems is a moving-front algorithm, as proposed in Thompson Hale (2005), transforming the mp-NLP problem into a set of non-linear equations reformulating the Fritz-John conditions. The solution is approximated via local parametrisations inside simplicial complexes. According to Thompson Hale (2005), the algorithm only gives satisfactory results for problems with up to three or four parameters. This circumstance is quite restrictive for an application of the controller.

A similar approach as in Johansen (2004) can be found in Grancharova et al. (2007) and Grancharova and Johansen (2006) for non-convex mp-NLP problems representing an approximating mp-NLP algorithm. The parameter space partitioning is based on orthogonal partitions with associated linear or non-linear approximation functions. In order to avoid global optimisation techniques for the non-convex case with multiple local minima, the use of efficient initialisation procedures using local NLP solvers combined with heuristic strategies is proposed. As pointed out in Domínguez et al. (2010), the use of heuristics for partitioning the parameter space combined with efficient methods for the handling of data structures, such as orthogonal and binary search trees, is a promising approach. More details on the approximating mp-NLP algorithm can also be found in Grancharova and Johansen (2012).
Suggested approach

The suggested approach combines performing elements of established algorithms for approximating mp-NLP problems, also in the non-convex case. The algorithm is composed by blending the approximate mp-NLP algorithm incorporating global optimisation tools, as proposed in Grancharova et al. (2007) and Grancharova and Johansen (2012), with the mp-QA algorithm, as proposed in Johansen (2002) and Grancharova and Johansen (2012).

The mp-NLP problem is locally approximated with mp-QP sub-problems on orthogonal partitions involving a quadratic approximation of the cost function and a linear approximation of the constraint functions, as suggested in Johansen (2002). The Hessian matrix, incorporating the second derivatives with respect to the optimisation variables, is manipulated in case of negative eigenvalues. The use of mp-QP sub-problems allows an approximation of the critical regions with polyhedral regions within each orthogonal partition.

Non-convex problems require the use of global optimisation techniques. As a practical approach obtaining a close-to-global solution of single NLPs, an effective initialisation procedure, as proposed in Grancharova et al. (2007), is incorporated in the algorithm comparing local minima corresponding to several initial guesses.

For the computation of the error bound approximation, the proposed heuristic method in Grancharova et al. (2007), including interior points in addition to the vertices, is applied to the cost and solution error, as well as the maximum constraint violation, as shown in Johansen (2002).

The heuristic splitting rules, as proposed in Grancharova et al. (2007) and Grancharova and Johansen (2002), are revisited and extended with the computation of new hyperrectangles. The best of all possible combinations minimising a weighted average of the considered errors is kept and the remaining combinations are discarded.

A post-processing step is established, as proposed in Johansen (2002) and demonstrated in Domínguez and Pistikopoulos (2011), combining regions with the same solution associated with the first sample to convex sets. Without loss of accuracy, the complexity of the controller can be reduced, especially because the combination of orthogonal partitions with polyhedral regions is likely to form convex sets.

The definition of approximating functions on polyhedral regions allows the generation of a binary search tree, as proposed in Tøndel et al. (2003b), facilitating efficient data handling. The method using a binary search tree enables evaluation times being logarithmic in the number of regions and storage requirements being polynomial in the number of regions.

For convex mp-NLP problems, guaranteed properties can be established. However, when convexity is violated, partitioning and termination criteria based on convexity
theory combined with heuristics are used in favour of a computationally efficient approach handling non-convex problems. Despite not having guaranteed properties a priori, an explicit representation of the approximating solution is available, making the application of rigorous validation and verification procedures to the properties of the explicit approximate solution of non-convex mp-NLPs possible. The use of heuristics will likely reduce non-convexity related errors in the computed bounds and approximation of the constraints. For example, the introduction of interior points in addition to the set of vertices serves to verify the accuracy of the piecewise affine approximate solution, not leading to additional complexity of the solution itself.

Multi-parametric quadratic programming problems

In the suggested approach, the approximating mp-QP sub-problems have to be solved on each corresponding orthogonal partition. The two main avenues for the solution of an mp-QP problem are the exact solution and the approximate solution.

The mp-QA algorithm, as proposed in Johansen (2002) and Grancharova and Johansen (2012), employs the algorithm in Tøndel et al. (2003a) leading to an exact solution of the mp-QP problem. The method in Tøndel et al. (2003a) is a modification of Bemporad et al. (2000) and Bemporad et al. (2002) establishing an exact solution to multi-parametric linear and quadratic programming problems. The proposed approach significantly improves the efficiency by analysing several characteristics of the geometry of the polyhedral partitions and its relation to the combination of active constraints at the optimiser of the quadratic program. The algorithms allow the offline computation of the explicit and piecewise affine state feedback law associated with each corresponding polyhedral critical region. A comprehensive overview can also be found in Grancharova and Johansen (2012).

A combination of the algorithms presented in Bemporad et al. (2002) and Tøndel et al. (2003a), using the strategy of Tøndel et al. (2003a) to step over the facets between neighbouring regions, in combination with the QP solution of Bemporad et al. (2002) identifying the optimal active set, is presented in Baotić (2002). Similar to Tøndel et al. (2003a), it depends on the facet-to-facet properties and including modifications, as described in Spjøtvold et al. (2007), gives further improvements. The approach proposed in Baotić (2002) is the primary mp-QP algorithm of the commonly used Multi-Parametric Toolbox (MPT), introduced in Kvasnica et al. (2004).

In the suggested approach, the revised version 3.0 of the Multi-Parametric Toolbox (MPT 3.0), as introduced in Herceg et al. (2013), is employed. The mp-QP formulation is converted into a parametric linear-complementary problem (PLCP) form, providing automatic routines using affine transformations for the variables. The PLCP approach is numerically robust and provides superior efficiency compared to other methods. In the
MPT 3.0 toolbox, the algorithm as proposed in Jones and Morrari (2006) is used for the solution of the PLCP problem.

For both the mp-Q/OA algorithm, shown in Domínguez et al. (2010), and the revisited mp-QA algorithm Johansen (2002) in Domínguez and Pistikopoulos (2011), the Parametric OPtimisation (POP) toolbox, introduced in Pistikopoulos et al. (1999), is employed for the solution of the mp-QP sub-problems.

Possible other approaches solving the mp-QP problem include a combinatorial method, as proposed in Seron et al. (2008), considering combinations of potentially optimal active constraints tending to lead, however, to critical regions that are not fully dimensional and must be, therefore, disregarded. Another approach, as suggested in Olaru and Dumur (2004), exploits the double representation with vertices and hyperplanes of polyhedrons. A non-geometric combinatorial approach, as shown in Gupta et al. (2011), can be implemented trimming infeasible candidates of active sets.

Instead of searching for an exact solution of the mp-QP problem, the approximate and sub-optimal solution with mp-QP algorithms has been proposed leading to approximations of lower complexity. Such a method for the approximate solution of mp-QP problems is proposed in Johansen and Grancharova (2003) and Johansen and Grancharova (2002). The algorithms rely on the approximation with affine functions on an orthogonal partitioning of the parameter space that is recursively build to achieve the specified approximation tolerances. A similar approach is proposed in Grancharova and Johansen (2002) incorporating the use of heuristics in the algorithm. Other approaches can be found in Bemporad and Filippi (2001) using simplicial partitions of the parameter space, and in Scibilia et al. (2009) employing Delaunay tessellation for the partitioning.

There are various other approaches, such as Grieder and Morari (2003), proposing the reformulation of the MPC problem solving a sequence of simpler one-step explicit MPC problems and a nested sequence of terminal sets. Another sub-optimal approach based on short horizons is proposed in Johansen et al. (2002). Reducing the complexity of the piecewise affine solution to mp-QP problems is investigated in Bemporad and Filippi (2003). In Nguyen et al. (2011), interpolation techniques and the use of nested invariant sets are employed, and in Canale et al. (2009a), an approximation using set membership approximations is introduced. While all the above-mentioned approximations lead to piecewise affine function representations, in Kvasnica et al. (2011a), polynomial approximations are considered. A comprehensive overview can also be found in Alessio and Bemporad (2009) including mp-LP algorithms.

**Post-processing and function evaluation**

All the above-presented mp-QP and mp-NLP algorithms provide piecewise function representations of the solution defined over polyhedral regions. Except for the one
approporah presented in Kvasnica et al. (2011a), all mp-QP algorithms and most of the
mp-NLP algorithms return affine function pieces. The performance of the explicit non-
linear MPC strongly depends on efficient methods evaluating the piecewise functions,
since the complexity of the representations may be large even for relatively small systems.
A comprehensive overview of different approaches can be found in Grancharova and
Johansen (2012).

Data handling  Efficient methods for function evaluation are relying on the appropriate
handling of the defining data. One possible approach is the direct approach, which
evaluates for each region if the present parameter belongs to that region. This simple
method is computationally demanding but may be implemented on parallel computer
architecture.

A computationally more efficient method is proposed in Tøndel and Johansen (2002)
and Tøndel et al. (2003b) relying on a representation of the polyhedral partitions with a
binary search tree. It is a data structure excluding a significant fraction of the remaining
potential regions at each level of the search tree leading to a logarithmic complexity in
the number of partitions. The search tree can build an integral part of multi-parametric
programming approximation on orthogonal partitions, such as the approximate mp-NLP
algorithm in Johansen (2004) or the approximate mp-QP algorithm in Johansen and
Grancharova (2003). The approach proposed in Tøndel et al. (2003b) is employed in
the MPT 3.0 toolbox Herceg et al. (2013) for the generation of a binary search tree
for the representation of the explicit solution. The construction of a balanced binary
search tree minimising the worst-case computational burden can require extensive offline
computations and still be in the need of significant online computer memory. This also
applies for approaches, such as Fuchs et al. (2010), exploiting optimal algorithms for the
selection of hyperplanes for decisions.

Modifications of the binary search tree method to address this issue are proposed in
Bayat et al. (2011a) using a truncated binary search tree combined with direct search or
in Bayat et al. (2011b) using an orthogonal truncated binary search tree combined with
a lattice representation of the piecewise affine (PWA) functions.

Other data structures, such as bounding-boxes in Christophersen et al. (2007) or
hash-tables in Bayat et al. (2011c), can be used as supporting structures to efficiently
narrow down the localisation of the optimal region.

Another method, as proposed in Spjøtvold et al. (2006) and Wang et al. (2007),
represents the topology of polyhedral regions for the identification of neighbouring
partitions along the path from one parameter to the subsequent one. In the context of
MPC, the parameter at one time instant is likely to be close to the parameter at the
previous time instant, considering the continuity of the trajectory of the dynamic system.
Alternatively, the collection and evaluation of a Boolean vector based on the hyperplane arrangement together with Boolean functions relating to the feedback law are proposed in Geyer et al. (2008). This method reducing the memory requirements and online computation times is particularly suitable for hardware implementation.

**Complexity reduction** Another approach, significantly improving the performance of explicit NMPC, tries to reduce the complexity of the explicit representation of the approximate solution.

One concept is based on joining regions, which contain the same control law associated with the first sample of the control trajectory to partitions, such that the union remains convex. Merging of regions can significantly reduce the complexity of the solution generated by the algorithm and potentially have a major impact on the performance of the explicit NMPC. As mentioned in Domínguez and Pistikopoulos (2011), the operation is restricted to cases where the union preserves convexity. The partitioning of the parameter space with orthogonal partitions combined with polyhedral regions as occurring in the suggested approach is, however, likely to form convex sets and, therefore, advantageous in the post-processing step.

A sub-optimal method of merging polyhedra by cycling through the regions and consecutively determining if a pair of neighbouring polyhedra forms a convex union is proposed in Geyer et al. (2008) using algorithms described in Bemporad et al. (2001). As mentioned in Geyer et al. (2008), trying all polyhedral combinations using standard techniques, based on linear programming, can be prohibitive if the number of polyhedra with the same function is large since the number of possible combinations explodes.

Two other methods for merging regions sharing the same control law are suggested in Geyer et al. (2008) using algorithms based on the cells and the markings of the hyperplane arrangement and possibly employing a divide and conquer strategy. The proposed disjoint optimal complexity reduction, based on branch and bound search, leads to an optimal merging problem yielding to a non-overlapping set of polyhedra, and the non-disjoint optimal complexity reduction, based on logical minimisation, leads to an optimal set covering problem obtaining polyhedra that are in general overlapping. According to Huba et al. (2011), such an approach is computationally demanding due to the computation of the cells in the hyperplane arrangement. All three of the mentioned methods are implemented in the MPT 3.0 toolbox, Herceg et al. (2013).

The concept suggested in Kvasnica and Fikar (2012), Kvasnica and Fikar (2010), and Huba et al. (2011) exploits the fact that control saturation will occur in a large number of regions. This number is reduced by eliminating those regions over which the PWA function attains a saturated value and by applying clipping functions. This method is also implemented in an extending package for the MPT 3.0 toolbox, Herceg et al. (2013).
A related method is proposed in Kvasnica et al. (2013) and Kvasnica et al. (2011b) simplifying the controller by employing separating functions, where only the unconstrained regions and the separator needs to be stored.

Another concept for practical implementation is shown in Huba et al. (2011), Kvasnica et al. (2011a), Kvasnica et al. (2010), and Kvasnica et al. (2008) approximating the piecewise affine control law defined on polyhedral regions by a single polynomial valid on the entire parameter space.

A similar method is proposed in Takacs et al. (2013) by optimising offline the parameters of an approximation of the PWA feedback law such that an integrated square error between the optimal but complex controller and its simpler replacement is minimised.

Further methods for additional compression of representation and efficient function evaluation are mentioned in Szucs et al. (2011) and Nielsen and Axehill (2016).

**Suggested approach** The proposed post-processing of the approximate solution, generated by the suggested mp-NLP algorithm, includes both complexity reduction and efficient data handling. The complexity reduction incorporates the clipping-based method, as proposed in Kvasnica and Fikar (2012), and the merging procedures based on the sub-optimal and the disjoint optimal merging methods, proposed in Geyer et al. (2008), employing the implemented tools of the MPT 3.0 toolbox, Herceg et al. (2013). Similarly, for the data processing and effective evaluation of the function representation, the tools of the MPT 3.0 toolbox, Herceg et al. (2013), are employed, providing an implementation of a binary search tree generation, based on Tøndel et al. (2003b).
Chapter 3

Non-linear model predictive control

In the following, the problem of controlling a multi-variable non-linear dynamic system is considered that is subject to physical and operational constraints on the states and inputs. In the context of non-linear model predictive control, the formulation of an optimisation problem including the constraints on the control and state variables is required. Key elements leading to a finite-dimensional numerical optimisation problem are discretisation and parametrisation procedures. In this chapter, the formulation of the optimal control problem is considered, which is an essential part of the control design and involves numerous decisions. The influence on the control performance, feasibility, stability, and robustness, as well as aspects such as computational complexity and numerical challenges are investigated. The following sections will focus on the formulation of an NMPC optimisation problem, leading to a non-linear mathematical programming problem, such as (2.1)-(2.2).

3.1 Continuous-time optimal control problem

3.1.1 Dimensional NMPC problem

In this section, a general non-linear optimal control problem formulation based on a continuous-time model and dependent on dimensional variables is introduced. The objective is to minimise the cost function, $V$, over a finite horizon,

$$V\left(\mathbf{X}[T_k, T_f], \mathbf{U}[T_k, T_f], \mathbf{P}(T_k), \mathbf{N}[T_k, T_f]\right) = \int_{T_k}^{T_f} L\left(\mathbf{X}(T), \mathbf{U}(T), \mathbf{P}(T_k), \mathbf{N}(T), T\right) dT + M\left(\mathbf{X}(T_f), \mathbf{P}(T_k), T_f\right), \quad (3.1)$$
subject to the general inequality and equality constraints for all $T \in [T_k, T_f]$,

$$G\left(X(T), U(T), P(T_k), N(T), T\right) \leq 0,$$

and the ordinary differential equations (ODE) describing the evolution of the system dynamics defined by

$$\frac{d}{dT}X(T) = F\left(X(T), U(T), P_s(T_k), T\right),$$

with given initial values of the states $X(T_k) \in \mathbb{R}^n$. The output equations are defined by

$$Y(T) = H\left(X(T), U(T), P_s(T_k), T\right).$$

The function $L$ is known as the stage cost or Lagrange term, $M$ is the terminal cost or Mayer term, $T_f > T_k$ is the horizon, and together they define the cost function $V: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \times \mathbb{R}^v \rightarrow \mathbb{R}$, as introduced in (3.1). The evolution of the states $X(T) \in \mathbb{R}^n$ is given by the state equations $F: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \rightarrow \mathbb{R}^n$, defined by (3.3), and depends on the input trajectory $U(T) \in \mathbb{R}^m$, the system parameters $P_s(T_k) \in \mathbb{R}^d$, and the time $T$. The state vector $X(T)$ contains the state variables of the system, which are the smallest possible subset of system variables that can represent the entire state of the system at any given time. It is supposed that a full measurement or estimation of the state vector $X(T_k)$ is available at the current time $T_k$. The vector of system parameters $P_s(T_k)$ contains physical properties that are assumed to be constant for an optimisation problem at a given time. However, considering a receding horizon philosophy, they can change for each optimisation problem with changing time. The ordinary differential equations impose infinite-dimensional equality constraints to the solution in the optimal control problem formulation. For a general non-linear system, the ordinary differential equations of arbitrary order describing the system dynamics can be transformed into a system of first order differential equations leading to the state equations, as introduced in (3.3). The algebraic output equations $H: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \rightarrow \mathbb{R}^q$ in (3.4) describe the evolution of the outputs $Y(T) \in \mathbb{R}^q$.

Additionally, hard constraints on the inputs representing saturation with minimum and maximum thresholds $U_{\text{min}}$ and $U_{\text{max}}$, respectively, can be formulated. Also possible are soft constraints typically on the states including slack variables $N(T) \in \mathbb{R}^v$ and general inequality constraints jointly on the states and inputs, for $T \in [T_k, T_f]$. The general formulation of the constraints includes \textit{bound constraints} and \textit{path constraints}. All the inequality constraints can be summarised and defined by the functions $G_i: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \times \mathbb{R}^v \rightarrow \mathbb{R}^{w_{c,i}}$. Analogously, general equality constraints can be introduced and defined by the functions $G_e: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \times \mathbb{R}^v \rightarrow \mathbb{R}^{w_{c,e}}$. All constraint functions are summarised
in the vector of (inequality and equality) constraint functions \( G = [G_i^T, G_e^T]^T, G \in \mathbb{R}^{wc} \). Both physical and operational constraints of the control system can be taken into account and formulated as constraint equations, as well as stability-preserving terminal sets.

The definition and the properties of the Lagrange term \( L \) and the Mayer term \( M \) have a major influence on the control performance, including stability, and must be well understood and carefully tuned. Time-varying reference trajectories, e.g. output reference signals \( R_y \), exogenous input signals, e.g. input reference signals \( R_u \), and possibly known disturbances can be accounted for in the optimal control problem formulation by the explicit time-dependence in \( F, G, L, \) and \( M \). The solving multi-parametric programs, the time, or some parametrisation of these exogenous signals, should then be included in the parameter vector \( P_c(T_k) \in \mathbb{R}^{d_c} \). The vector \( P_c(T_k) \) contains quantities that are assumed constant for an optimisation problem at a given time. However, they can change with time considering a receding horizon philosophy. By combining the vector of system parameters \( P_s(T_k) \) and the vector of controller parameters \( P_c(T_k) \), one can define the vector of system and controller parameters \( P(T_k) \in \mathbb{R}^{d} \), where \( d = d_s + d_c \). It has to be noted, however, that this parametrisation will enter the parameter vector of the later introduced multi-parametric non-linear programming problem, and may significantly contribute to the complexity of the representation of the explicit solution approximation. In the following, all necessary assumptions, such as continuity and smoothness of the functions involved, are implicitly assumed. Below, the optimal control problem formulation is described using two common examples, as introduced in Grancharova and Johansen (2012), namely a regulation and a reference tracking problem.

### 3.1.2 Regulation problem

Firstly, the optimal control problem is introduced as regulation problem where the goal is to steer the system states to the origin. For the current \( X(T_k) \), the regulation NMPC solves the continuous optimisation problem:

\[
V^*\left( X[T_k, T_f], P_s(T_k) \right) = \min_{U[T_k, T_f], N[T_k, T_f]} V\left( X[T_k, T_f], U[T_k, T_f], P_s(T_k), N[T_k, T_f] \right), \quad (3.5)
\]

subject to

\[
Y_{\text{min}} \leq Y(T) \leq Y_{\text{max}}, \quad (3.6)
\]

\[
U_{\text{min}} \leq U(T) \leq U_{\text{max}}, \quad (3.7)
\]

\[
N(T) \geq 0, \quad (3.8)
\]

\[
X(T_f) \in \Omega, \quad (3.9)
\]
Non-linear model predictive control

\[
\frac{d}{dT} X(T) = F(X(T), U(T), P_s(T_k), T),
\]

(3.10)

\[
Y(T) = H(X(T), U(T), P_s(T_k), T),
\]

(3.11)

with \(T \in [T_k, T_f]\) and the cost function typically defined as

\[
V = \int_{T_k}^{T_f} \left( \|X(T)\|^2_{q_x} + \|U(T)\|^2_{r_u} + \|N(T)\|^2_{q_v} \right) dT + \|X(T_f)\|^2_{p_x}.
\]

(3.12)

Here, \(T_f\) is a finite horizon and \(N(T) \in \mathbb{R}^{v_c}\) is the vector of slack variables. Further, \(q_x, r_u, q_v,\) and \(p_x\) are weighting matrices, and \(\Omega \subset \mathbb{R}^n\) is a terminal set.

### 3.1.3 Reference tracking problem

Secondly, the optimal control problem is formulated as *reference tracking* problem which is relevant for the practical implementation of the controller. The goal of a reference tracking controller is to have the output vector \(Y(T)\) track the output reference signal \(R_y(T)\) and possibly make the input vector \(U(T)\) track the input reference signal \(R_u(T)\). By combining the vector of output reference signals \(R_y(T) \in \mathbb{R}^{p_y}\) and the vector of input reference signals \(R_u(T) \in \mathbb{R}^{p_u}\), one can define the vector of reference signals \(R(T) \in \mathbb{R}^p\), where \(p = p_y + p_u\). For the current \(X(T_k)\), the reference tracking NMPC solves the optimisation problem:

\[
V^*(X[T_k, T_f], P_s(T_k), R[T_k, T_f]) = \min_{U[T_k, T_f], N[T_k, T_f]} V(X[T_k, T_f], U[T_k, T_f], P_s(T_k), R[T_k, T_f], N[T_k, T_f]),
\]

(3.13)

subject to

\[
Y_{\min} \leq Y(T) \leq Y_{\max},
\]

(3.14)

\[
U_{\min} \leq U(T) \leq U_{\max},
\]

(3.15)

\[
N(T) \geq 0,
\]

(3.16)

\[
\|Y(T_f) - R_y(T_f)\| \leq \delta,
\]

(3.17)

\[
\frac{d}{dT} X(T) = F(X(T), U(T), P_s(T_k), T),
\]

(3.18)

\[
Y(T) = H(X(T), U(T), P_s(T_k), T),
\]

(3.19)
with $T \in [T_k, T_f]$ and the cost function typically defined as

$$
V = \int_{T_k}^{T_f} \left( \| \mathbf{Y}(T) - \mathbf{R}_y(T) \|^2_{q_x} + \| \mathbf{U}(T) - \mathbf{R}_u(T) \|^2_{r_u} + \| \mathbf{N}(T) \|^2_{q_v} dT + \| \mathbf{Y}(T_f) - \mathbf{R}_y(T_f) \|^2_{p_x} \right) .
$$

(3.20)

Similar to above, $T_f$ is a finite horizon and $q_x$, $r_u$, $q_v$, and $p_x$ are weighting matrices. The explicit time dependence in the vector $\mathbf{R}(T)$ for the reference signals allows to include time-varying reference trajectories in the optimal control problem formulation. One can define a parametrisation of these exogenous signals included in the vector $\mathbf{R}(T)$ parametrised by the vector of controller parameters $\mathbf{P}_c(T_k)$. For instance, the parametrised reference trajectory $\mathbf{R}(T) = \Phi \left( \mathbf{X}(T_k), \mathbf{P}_c(T_k), T \right)$ can be uniquely described by the vector $\mathbf{P}_c(T_k)$ and the initial states $\mathbf{X}(T_k)$ of the system. Using the definition of the vector of system and controller parameters, the optimisation problem can be written as

$$
V^* \left( \mathbf{X}[T_k, T_f], \mathbf{P}(T_k) \right) = \min_{\mathbf{U}[T_k, T_f], \mathbf{N}[T_k, T_f]} V \left( \mathbf{X}[T_k, T_f], \mathbf{U}[T_k, T_f], \mathbf{P}(T_k), \mathbf{N}[T_k, T_f] \right),
$$

(3.21)

subject to the constraints (3.14)-(3.19), leading to the formulation of the cost function as defined in (3.1).

### 3.2 Non-dimensionalisation and scaling

#### 3.2.1 Scaling factors in the cost function

In the formulation defining the Lagrange term and the Mayer term of the general cost function (3.1), the use of scaling factors has proven to be beneficial. The main reason for the introduction of scaling factors is to ensure that all terms in the cost function have the same order of magnitude and that the influence of the different weights on the corresponding terms is comparable.

For the numeric evaluation of the cost function, the contributions of the individual terms within the formulation should not be too different. A priori, the formulation of the cost function includes dimensional variables with different units possibly leading to significant differences in the values of the variables for typical operating conditions of the system. By introducing dimensional scaling factors and properly choosing suitable values, it is possible to achieve a comparable contribution of the individual terms leading to a similar influence of the chosen weights. Therefore, setting all weights to the same value would result in a similar effect of each term. Furthermore, the dimensional scaling factors allow a definition of dimensionless weights and result in a dimensionless formulation of the cost function.
The introduction of scaling factors is not mandatory in the formulation of the cost function. However, the use of properly chosen scaling factors facilitates the choice of appropriate weights, tuning of the controller, and interpretation of the results.

### 3.2.2 Non-dimensionalisation of the optimal control problem

**Motivation**

Non-dimensionalisation is a procedure applying a suitable substitution of variables in order to partially or fully remove units from an equation involving physical quantities. It can parametrise and simplify problems involving measured units by relating certain quantities to some appropriate units of measure, referring to characteristic quantities intrinsic to the system.

The procedure of non-dimensionalisation involves the identification of dependent and independent variables in the describing differential and algebraic equations. Both variables are replaced with quantities scaled to a characteristic unit of measure with appropriate dimension. By judiciously choosing the definition of the characteristic quantity involved in the substitution of each variable, the coefficients of as many terms as possible in the equations become 1 or close to 1. This leads to the main motivation of non-dimensionalisation, namely the good numerical condition of the dimensionless optimal control problem. It is also worth to mention, that the non-dimensionalisation of all variables involved in the optimisation problem and, therefore, scaling to the same order of magnitude is often a pre-requisite for numerical non-linear optimisation methods to work satisfactorily. However, the success of the non-dimensionalisation procedure obviously depends on the choice of the characteristic quantities and should be carried out conscientiously for that reason.

An additional benefit of non-dimensionalisation is the standardisation of the optimisation problem. By choosing the bounds on the states and inputs, characterising the maximum allowed or possible values of the corresponding variables, as the magnitude of the characteristic quantities, the typical range of numerical values for the dependent dimensionless variables is between $-1$ and 1. This circumstance is in particular beneficial for the state-space exploration performed during the execution of the explicit NMPC algorithm 1 approximating the mp-NLP problem. Regardless of the optimal control problem formulation and the underlying system, it is, therefore, possible to derive a uniform optimisation problem and standardise the state-space exploration, as described in Chapter 4.

Non-dimensionalisation can also facilitate the identification of characteristic properties inherent to the system, such as an immanent resonance frequency or time constant. Measurements made in relation to an intrinsic property of a system also apply to other
systems having the same intrinsic property. Additionally, the comparison of common properties of different systems is possible. More specifically, non-dimensionalisation systematically determines characteristic units of a system to use or can even identify which parameters should be used for analysing a system without having detailed prior knowledge about the system’s intrinsic properties.

Finally, non-dimensionalised equations help to gain a greater insight into the relative size of the various terms present in the equations. Following an appropriate selection of the characteristic quantities for the non-dimensionalisation process, this leads to the identification of negligible terms in the equations.

**Non-dimensionalisation procedure**

Starting from the describing differential and algebraic equations, the following steps are necessary to perform the non-dimensionalisation:

1. All independent and dependent variables are identified.

2. Each variable is replaced according to a transformation law with a dimensionless variable and a related characteristic quantity consisting of a characteristic magnitude and a characteristic unit of measure of appropriate dimension.

3. The differential or algebraic equation is divided by the coefficient of the highest derivative or polynomial term.

4. The magnitude of the characteristic quantity of each variable is judiciously defined such that the coefficients of most of the terms become 1 or close to 1.

5. The equations are rewritten and expressed in terms of the new dimensionless variables.

**Nomenclature** In order to easily distinguish between dimensional and dimensionless quantities in the text of Part I, the following conventions on the nomenclature are introduced:

- *Dimensional quantities* All quantities related to a fundamental dimension and a unit of measure, are denoted with *capital letters*. This definition applies regardless of the type of quantity such as variable, parameter, etc. and also the size such as scalar, vector, or matrix. There are only a few exceptions where quantities denoted with capital letters contain dimensionless quantities. These cases are explicitly indicated in the text.

- *Dimensionless quantities* All quantities not related to a fundamental dimension or a unit of measure, are denoted with *lower-case letters*. Similar to above, this definition
applies regardless of the type of quantity such as variable, parameter, etc. and also the size such as scalar, vector, or matrix.

**Demonstration example**  As an illustrative example for the demonstration of the non-dimensionalisation procedure, a first-order differential equation with constant coefficients is considered:

\[
\frac{d}{dT} X(T) = F(X(T), U(T), T) := AX(T) + BU(T).
\]

This equation is a *dimensional equation dependent on dimensional variables*.

1. In equation (3.22), the independent variable is \(T\), and the dependent variables are \(X(T)\) and \(U(T)\). All variables are dimension- and unit-related, and, therefore, defined with capital letters. According to the theory of ordinary differential equations, the input \(U(T)\) actually represents a forcing function. However, for simplicity it is treated as a dependent variable in the procedure of non-dimensionalisation.

2. Replace each of them by applying the transformation laws \(T = T_{ch} t\), \(X(T) = X_{ch} x(t)\), and \(U(T) = U_{ch} u(t)\). This results in the *dimensional equation depended on dimensionless variables*,

\[
\frac{d}{dT} X(T) = \frac{X_{ch}}{T_{ch}} \frac{d}{dt} x(t) = F(x(t), u(t), t) := A X_{ch} x(t) + B U_{ch} u(t).
\]

After the transformation, all variables are dimensionless and, therefore, denoted with lower-case letters.

3. Dividing by the coefficient of the highest order term, i.e. the coefficient in front of the first derivative term, gives a *dimensionless equation dependent on dimensionless variables*,

\[
\frac{d}{dt} x(t) = \frac{T_{ch}}{X_{ch}} F(x(t), u(t), t) = f(x(t), u(t), t) := A T_{ch} x(t) + B T_{ch} \frac{U_{ch}}{X_{ch}} u(t).
\]

4. In the following, the characteristic quantities are defined using bounds or equivalent quantities incorporated in the optimal control problem. However, it is also possible to choose the characteristic quantities such that the coefficients of the differential equation systematically become 1,

\[
A T_{ch} = 1 \Rightarrow T_{ch} = \frac{1}{A}, \quad B T_{ch} \frac{U_{ch}}{X_{ch}} = \frac{B}{A} \frac{U_{ch}}{X_{ch}} = 1 \Rightarrow X_{ch} = \frac{B}{A} U_{ch}.
\]
5. Depending on the definition of the characteristic quantities, the final dimensionless equation can become completely independent of any units,

\[
\frac{d}{dt} x(t) = x(t) + u(t). \tag{3.26}
\]

However, the application of the non-dimensionalisation procedure to the describing differential equations of the optimal control problem and the typical choice of the characteristic quantities will most likely lead to equations in the form (3.24) and coefficients of the individual terms in the equations close to 1.

The application of the non-dimensionalisation procedure to algebraic equations is equivalent to the differential equation shown above. It will affect all the constraint equations as they are algebraic equations.

**Transformation law** In the non-dimensionalisation procedure, the independent and dependent variables are replaced with their scaled quantities according to the following transformation laws. The subscript “\(ch\)” added to a quantity’s variable-name is used to denote the characteristic quantity used to scale that variable. If differential operators are needed to describe the original system, their scaled counterparts become dimensionless differential operators.

- **Independent variables** The independent dimension-related variable \(T\) will be substituted according to the following transformation law,

\[
T = T_{ch} t, \tag{3.27}
\]

involving the characteristic dimension-related quantity \(T_{ch}\) and the new dimensionless independent variable \(t\).

- **Dependent variables** Suppose \(X(T)\) is the dependent variable and \(T\) is the independent variable, where \(X(T)\) is a function of \(T\). Both \(X(T)\) and \(T\) represent variables with dimensions and units, respectively. To scale the dependent variable, assume there is an intrinsic unit of measure, the characteristic quantity \(X_{ch}\) with the same dimension as \(X\), such that

\[
X(T) = X_{ch} x(t), \tag{3.28}
\]

more specifically,

\[
X(T) = X_{ch} \hat{x}(T) = X_{ch} \hat{x}(T_{ch} t) = X_{ch} \hat{x}(T(t)) = X_{ch} x(t). \tag{3.29}
\]
Therefore, the dependent dimension-related variable $X(T)$ will be substituted by the
dimension-related characteristic quantity $X_{ch}$ and the new dependent dimensionless
variable $x(t)$ being a function of the independent dimensionless variable $t$.

### 3.2.3 Dimensionless NMPC problem

Starting from the formulation of the NMPC problem (3.1)-(3.4) dependent on dimensional
variables, the application of the non-dimensionalisation procedure leads to a formulation
dependent on dimensionless variables. In analogy to (3.1), the general formulation of the
continuous cost function dependent on dimensionless variables is given by

$$
V\left(\mathbf{x}[t_k, t_f], \mathbf{u}[t_k, t_f], \mathbf{p}(t_k), \mathbf{\nu}[t_k, t_f]\right) = \\
\int_{t_k}^{t_f} L\left(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t_k), \mathbf{\nu}(t), t\right) dt + M\left(\mathbf{x}(t_f), \mathbf{p}(t_k), t_f\right),
$$

(3.30)

subject to the continuous inequality and equality constraints dependent on dimensionless
variables for all $t \in [t_k, t_f]$,

$$
G\left(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t_k), \mathbf{\nu}(t), t\right) \leq 0,
$$

(3.31)

and the evolution of the continuous ordinary differential equation dependent on dimen-

$$
\frac{d}{dt} \mathbf{x}(t) = f\left(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}_s(t_k), t\right),
$$

(3.32)

with given initial values of the states $\mathbf{x}(t_k) \in \mathbb{R}^n$. For a reference tracking problem, the
application of the non-dimensionalisation procedure to the output equations (3.4) leads
to the continuous equation,

$$
\mathbf{y}(t) = h\left(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}_s(t_k), t\right),
$$

(3.33)

dependent on dimensionless variables.

In the equations above, the dimensionless vectors $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, and
$\mathbf{y}(t) \in \mathbb{R}^q$ are for the state, input, and output variables, respectively, and $\mathbf{\nu}(t) \in \mathbb{R}^{v_c}$ for
the slack variables. Moreover, $\mathbf{p}_s(t_k) \in \mathbb{R}^{d_s}$ are the system parameters and together with
the controller parameters $\mathbf{p}_c(t_k) \in \mathbb{R}^{d_c}$ they form the vector of system and controller
parameters $\mathbf{p}(t) \in \mathbb{R}^d$.

The nomenclature in the formulation of the cost function in (3.30) and the constraint
functions in (3.31) differs from the convention introduced in Subsection 3.2.2. As defined
beforehand, dimensional and unit-related quantities are denoted with capital letters.
However, the cost function will always be denoted with the capital letter $V$, regardless of the dimension of the cost function and its dependent variables. The same applies to the vector of constraint functions $G$, which is also denoted with a capital letter independent of the dimension of the functions and their dependent variables.

By introducing dimension-related scaling factors for the definition of the cost function, the cost function $V$ has become dimensionless already before applying the non-dimensionalisation procedure (refer to Subsection 3.1.1, where the equations generally are depending on dimensional variables). However, the cost function $V$ will be latest dimensionless after applying the non-dimensionalisation procedure, regardless of the introduction of scaling factors.

### 3.3 Numerical optimal control

The formulations of the NMPC problem introduced so far in Sections 3.1 and 3.2, represent an infinite-dimensional optimal control problem. The solution of such problems can be characterised using classical tools such as Pontryagin’s maximum principle, calculus of variations, and dynamic programming. These are so-called *indirect methods* and an exact analytic representation of the solution can only be found in a very limited number of special cases. Although numerical solutions can be found based on the characterisations of the indirect methods, in the suggested approach and in the context of NMPC, we restrict our attention to the most promising and popular so-called *direct methods*.

#### 3.3.1 Finite parametrisation and discretisation

Direct methods are characterised by finite parametrisation and discretisation applied to the optimal control problem formulation which is then solved directly with numerical methods. The concept of NMPC is to solve the finite-horizon optimisation problem repeatedly at each time instant. This approach is also known as receding horizon approach. The initial values of the states $x(t_k)$ to (3.32) combined with a given vector of system and controller parameters $p(t_k)$ are considered as the current parameter vector $x_p(t_k)$, based on the most recent measurements initialising the optimisation problem. The solution of this problem leads to the optimal control trajectory $u[t_k, t_f]$ and slack variable trajectory $\nu[t_k, t_f]$. The optimal control trajectory is implemented for a short period of time, commonly one sampling interval $t_s$ and typically much smaller than the horizon $t_p = t_f - t_k$. The procedure is repeated and an updated optimal control trajectory will be available. As presented in Huba et al. (2011), the continuous formulation of the optimal control problem requires reformulation, in order to numerically solve the problem, for the following reasons.
The infinite-dimensional equality constraints imposed by the differential equations (3.32) must be represented by a finite approximation. The solution to the ordinary differential equations with given initial conditions is approximated applying a numerical integration scheme since an exact closed-form solution of the ODEs is usually not possible to derive in the non-linear case.

The infinite-dimensional unknown trajectory \( \mathbf{u}[t_k, t_f] \) and possibly \( \mathbf{v}[t_k, t_f] \) must be approximated by a finite number of decision variables in order to define a finite-dimensional optimisation problem that can be solved using numerical optimisation.

The measurements of the states \( \mathbf{x}(t) \) are typically sampled data and, therefore, only available at the sampling instants. Subsequently, the updated initial conditions \( \mathbf{x}(t_k) \) are only available at the defined sampling instants.

For the implementation of the control signal, it is not possible to implement arbitrary input trajectories since the control command can typically be changed only at defined sampling instants and is assumed to be constant (or following a similar sample-and-hold function such as a linear first-order hold) between those instants.

For the reformulation of the problem into a finite-dimensional optimisation problem and practical setting, the following assumptions are necessary to approximate the integral and differential operators with numerical integration methods, Huba et al. (2011).

The horizon, or also prediction horizon, \( t_p = t_f - t_k \) is finite and given. It is an integer multiple of the regular sampling interval \( t_s \) such that it fulfils the relationship \( t_p = N_p t_s \) with \( N_p \) as the number of prediction steps.

The input signal \( \mathbf{u}[t_k, t_f] \) is assumed to be piecewise constant over the horizon \( t_p \) with the regular sampling interval \( t_s \), and parametrised by a vector of control trajectory parameters \( \mathbf{U} \in \mathbb{R}^{mN} \) such that \( \mathbf{u}(t) = \mathbf{\mu}(t, \mathbf{U}) \in \mathbb{R}^m \) for \( t \in [t_k, t_f] \) is piecewise continuous.

Likewise, the slack variables \( \mathbf{v}[t_k, t_f] \) are assumed to be constant over the horizon \( t_p \) or piecewise constant with the regular sampling interval \( t_s \). Assuming piecewise constant slack variables, the signal can be parametrised by a vector of slack variable trajectory parameters \( \mathbf{N} = [\mathbf{v}(t_k)^T, \ldots, \mathbf{v}(t_{k+N_p})]^T \in \mathbb{R}^{vd} \), where \( vd = v_c \cdot (N_p + 1) \), such that \( \mathbf{v}(t) = \mathbf{\eta}(t, \mathbf{N}) \in \mathbb{R}^{vc} \) is piecewise continuous. Assuming the slack variables to be constant over the entire horizon, the signal can be parametrised by a vector containing the slack variables at the beginning of the horizon \( \mathbf{N} = \mathbf{v}(t_k) \in \mathbb{R}^{vc} \), where \( vd = v_c \), such that \( \mathbf{v}(t) = \mathbf{\eta}(t, \mathbf{N}) \in \mathbb{R}^{vc} \) for \( t \in [t_k, t_f] \) is continuous and constant.

An approximate solution to the ordinary differential equations can be defined in the form \( \mathbf{x}(t) = \phi(\mathbf{x}(t_k), \mathbf{U}, p_s(t_k), t) \) for the sequential approach, or in the form \( \mathbf{x}(t) = \phi(\mathbf{x}(t_k), \mathbf{X}, \mathbf{U}, p_s(t_k), t) \) for the simultaneous approach, at \( N_p \) discrete time instants \( t_d = \{t_{k+1}, t_{k+2}, \ldots, t_{k+N_p}\} \subset [t_k, t_f] \) using a solution function \( \phi(\cdot) \). The intermediate states in the simultaneous approach are summarised as parameters \( \mathbf{X} \in \mathbb{R}^{nN_p} \) for the
state trajectory. The discretisation of the ODEs leads to the discrete set of time instants \( t_d \) not necessarily being equidistant. The numerical integration of the ODEs embedded in the solution function \( \phi(\cdot) \) can include additional intermediate time steps not included in \( t_d \). The primary purpose of the discrete set \( t_d \) is to discretise the inequality constraints (3.31) at a finite number of points in time and approximate the integral in (3.30) with a finite sum. Although the time instants \( t_d \) need not coincide with the sampling instants in general, they are assumed in the following to be equidistant with the sampling interval \( t_d = t_s \) and coincide with the sampling instants.

As the parametrisation of the input trajectory \( \mu(t, U) \) on the horizon \( t \in [t_k, t_f] \) is important and will have influence on both the control performance and computational performance, it should satisfy in general the following objectives, according to Huba et al. (2011).

The parametrised trajectory should be flexible enough in order to allow for a solution of the reformulated optimal control problem close to the original problem (3.1)-(3.4). The parametrisation should not contain unnecessary parameters leading to an increased complexity of the explicit approximate representation, numerical sensitivity, and computational complexity.

The trajectory should also be implementable considering the capabilities of the hardware and software of the control system and the actuators, especially in the case using a parametrisation beyond a piecewise constant input trajectory that is restricted to change its value only at the sampling instants.

Based on the last two practical points, a general choice is the definition of a piecewise constant control input \( \mu(t, U) = u(t_{k+j}) \) for \( t_{k+j} \leq t < t_{k+j+1} \) over the horizon \( j = 0, \ldots, N_p - 1 \), parametrised by the vector of control trajectory parameters \( U = [u(t_0)^T, \ldots, u(t_{k+N_c-1})]^T \in \mathbb{R}^{mN_c} \), where \( N_c = N_p \). Practical experience shows that the receding horizon implementation offers considerable flexibility for an NMPC to recover performance due to sub-optimality at each step. Consequently, it is common practice to implement move-blocking strategies restricting the input from changing at every sampling instant over the horizon, especially towards the end of the prediction horizon, leading to a reduced number of required parameters. For that reason, the control horizon \( t_c \) is introduced, satisfying the relationship \( t_c = N_c t_s \) with \( N_c \) as the number of control steps. It is an integer multiple of the sampling interval \( t_s \), typically being \( t_c \leq t_p \), i.e. shorter or equal than the prediction horizon. Assuming a control horizon shorter than the prediction horizon \( t_c < t_p \), the piecewise constant control input would be defined as \( \mu(t, U) = u(t_{k+j}) \) for \( t_{k+j} \leq t < t_{k+j+1} \) over the control horizon \( j = 0, \ldots, N_c - 1 \) and \( \mu(t, U) = u(t_{k+N_c-1}) \) until the end of the prediction horizon \( t_{k+N_c} \leq t < t_{k+N_p} \), parametrised by the vector of control trajectory parameters \( U = [u(t_0)^T, \ldots, u(t_{k+N_c-1})]^T \in \mathbb{R}^{mN_c} \).
3.3.2 Direct methods

In this section, the general optimal control problem (3.30)-(3.33) is reformulated into a form suitable for the numeric solution by a non-linear optimisation solver. There are two main avenues to direct numerical optimal control, Grancharova and Johansen (2012):

- **Sequential approach** The ODE constraints (3.32) are solved via numeric simulation when evaluating the cost and constraint functions. This means that the intermediate states \( x(t_{k+1}), \ldots, x(t_{k+N_p}) \) disappear from the problem formulation by substitution into the cost function and the constraint functions, while the parameters \( U \) for the control trajectory and \( N \) for the slack variable trajectory are treated as unknowns. This method is known as Direct Single Shooting.

- **Simultaneous approach** The ODE constraints (3.32) are discretised in time and the resulting finite set of non-linear algebraic equations are treated as non-linear equality constraints. In addition to the parameters \( U \) for the control trajectory and \( N \) for the slack variable trajectory, the intermediate states \( x(t_{k+1}), \ldots, x(t_{k+N_p}) \), summarised as parameters \( X \) for the state trajectory, are treated as unknown variables. The integral in the cost function (3.30) is replaced by a finite sum. The solution of the resulting optimisation problem incorporating a larger number of constraints and variables leads to a simultaneous solution of the ordinary differential equation. The most well-known methods of this type are Direct Multiple Shooting and the Collocation Method.

All the above-mentioned methods have advantages and disadvantages depending on the application, that could make them the method of choice when considering a specific problem. For the practical implementation, the direct single shooting and the direct multiple shooting approach were the preferred choices of the presented methods.

The direct single shooting approach leads to a small and dense problem with a computationally complex cost function usually without non-linear equality constraints. The optimisation problem derived from the direct single shooting approach involves a smaller number of constraints and variables, and therefore leads to a “smaller” problem. The formulation only requires the control trajectory initialisation.

The direct multiple shooting method and the direct collocation method are in this sense fundamentally different. They not only require an initial control trajectory guess, but also one for the state trajectory. The availability of an appropriate initial guess for the state trajectory can be exploited by the simultaneous approach. However, for non-convex optimisation problems, e.g. in the presence of non-linear equality constraints, the initial guess provided for the solution has a major impact on the convergence and success of the chosen numerical optimisation algorithm. The simultaneous approach leads to a “bigger” problem involving a larger number of constraints and variables with, however, considerably simpler function evaluations for the cost and constraint functions.
As implemented in the developed toolbox for the control system design, the use of symbolic formulations for the cost function and the constraint functions, in terms of the parameter vector \( x_p \) and the vector of decision variables \( z \), allows an exact derivation of the defining matrices of a local mp-QP approximation with expressions in closed-form. However, with the increasing complexity of the underlying definitions of the optimal control problem and specifically of the continuous-time model, the complexity of the resulting expressions for the cost and constraint functions increases considerably. In particular, the direct single shooting method leads to a complex expression of the cost function growing significantly with an increasing number of prediction steps over the horizon. For this reason, the direct single shooting method can only be used for the anti-jerk control problems in Chapter 7. However, for the vehicle stability control problems in Chapter 5 and the wheel slip control problems in Chapter 6, the direct multiple shooting approach is employed exploiting the reduced complexity of the expressions for the cost and constraint functions, while having an increased number of constraints and decision variables.

**Ordinary differential equations**

**Direct single shooting** The idea of the direct single shooting method is to eliminate the ODE constraints (3.32) by substituting the discretised numerical solution \( \phi \) into the discrete formulation of the cost function and the constraint functions.

Starting from the continuous constraint equations, as defined in (3.32),

\[
\frac{d}{dt} x(t) = f(x(t), u(t), p_s(t_k), t),
\]

(3.34)

describing the evolution of the ordinary differential equations, the numerical solution,

\[
x(t_{k+j}) = \phi(x(t_k), U, p_s(t_k), t_{k+j}),
\]

(3.35)

for \( j = 1, \ldots, N_p \) is derived by discretisation and integration of the equations.

The ODE solution functions \( \phi \) are the result of a numerical integration scheme. In its simplest form, an explicit integration scheme may be used,

\[
x(t_{k+j+1}) = F(x(t_{k+j}), \mu(t_{k+j}, U), p_s(t_k), t_{k+j}),
\]

(3.36)

with the initial condition \( x(t_k) \) given. This integration scheme will lead to the numerical solution of the ODE constraints for \( t = t_{k+1} \) after one discrete time instant,

\[
x(t_{k+1}) = F(x(t_k), \mu(t_k, U), p_s(t_k), t_k),
\]

(3.37)
and after two discrete time instants for \( t = t_{k+2} \), substituting \( \mathbf{x}(t_{k+1}) \) with equation (3.37),

\[
\mathbf{x}(t_{k+2}) = F\left( \mathbf{x}(t_{k+1}), \mu(t_{k+1}, U), \mathbf{p}_s(t_k), t_{k+1} \right) \\
= F\left( F\left( \mathbf{x}(t_k), \mu(t_k, U), \mathbf{p}_s(t_k), t_k \right), \mu(t_{k+1}, U), \mathbf{p}_s(t_k), t_{k+1} \right),
\]

(3.38)

and generally for an arbitrary number of time instants \( t = t_{k+j} \) for \( j = 1, \ldots, N_p \),

\[
\mathbf{x}(t_{k+j}) = F\left( \mathbf{x}(t_{k+j-1}), \mu(t_{k+j-1}, U), \mathbf{p}_s(t_k), t_{k+j-1} \right) \\
= F\left( F\left( \mathbf{x}(t_{k+j-2}), \mu(t_{k+j-2}, U), \mathbf{p}_s(t_k), t_{k+j-2} \right), \mu(t_{k+j-1}, U), \mathbf{p}_s(t_k), t_{k+j-1} \right), \\
\vdots \\
= F\left( \ldots F\left( \mathbf{x}(t_{k+1}), \mu(t_{k+1}, U), \mathbf{p}_s(t_k), t_{k+1} \right), \ldots, t_{k+j-2} \right), \mu(t_{k+j-1}, U), \mathbf{p}_s(t_k), t_{k+j-1} \right),
\]

(3.39)

leading to the relationship (3.35).

**Direct multiple shooting** The idea of the direct multiple shooting method is to discretise the ODE constraints (3.32) in time and formulate the resulting finite set of non-linear algebraic equations as non-linear equality constraints in addition to the discrete formulation of the cost function and the constraint functions.

Analogously to (3.35), the numerical solution,

\[
\mathbf{x}(t_{k+j}) = \phi\left( \mathbf{x}(t_k), \mathbf{X}, U, \mathbf{p}_s(t_k), t_{k+j} \right),
\]

(3.40)

for \( j = 1, \ldots, N_p \) is derived by discretisation and integration of the equations.

Again, the ODE solution functions \( \phi \) are the result of a numerical integration scheme. An explicit integration scheme will lead to the numerical solution of the ODE constraints for an arbitrary number of time instants \( t = t_{k+j} \) for \( j = 0, \ldots, N_p - 1 \),

\[
\mathbf{x}(t_{k+j+1}) = F\left( \mathbf{x}(t_{k+j}), \mu(t_{k+j}, U), \mathbf{p}_s(t_k), t_{k+j} \right),
\]

(3.41)

with the initial condition \( \mathbf{x}(t_k) \) given. Equivalently, an implicit integration scheme may be used.
The finite set of non-linear algebraic equations resulting from the discretisation of the ODE constraints are formulated as non-linear equality constraints. In addition to the parameters $U$ for the control trajectory and $N$ for the slack variable trajectory, the intermediate states $x(t_{k+1}), \ldots, x(t_{k+N_p})$, that are summarised in the vector $X$ as parameters for the state trajectory, are treated as unknown variables. In order to enforce continuity at the interface of each subinterval arising from the discretisation, the following general conditions are formulated,

$$x(t_{k+j+1}) = F(x(t_{k+j+1}), x(t_{k+j}), \mu(t_{k+j}, U), p_s(t_k), t_{k+j}), \quad (3.42)$$

for $j = 0, \ldots, N_p - 1$. This expression describes the condition that the decision variables describing the intermediate states $x(t_{k+j+1})$, on the left hand side of the equation, have to be equal to the approximate numerical solution, on the right hand side, at that time instant $t = t_{k+j+1}$. Reformulating equation (3.42) leads to the following equality constraint function,

$$g_{dux}(x(t_k), X, U, p(t_k), N, t_{k+j+1}) = x(t_{k+j+1}) - F(x(t_{k+j+1}), x(t_{k+j}), \mu(t_{k+j}, U), p_s(t_k), t_{k+j}), \quad (3.43)$$

for $j = 0, \ldots, N_p - 1$. Evaluating the dynamic equality constraints at each time instant within the prediction horizon ($j = 0, \ldots, N_p - 1$) leads to $D_{ux} \in \mathbb{R}^{w,dux}$,

$$D_{ux} = G_{dux}(x(t_k), X, U, p(t_k), N) = \begin{bmatrix} g_{dux}(x(t_k), X, U, p(t_k), N, t_{k+1}) \\ \vdots \\ g_{dux}(x(t_k), X, U, p(t_k), N, t_{k+N_p}) \end{bmatrix}. \quad (3.44)$$

**Explicit integration schemes** Generally, the use of any numerical integration scheme is possible. However, for a practical implementation the two following methods are advantageous representing an explicit discretisation scheme:

- **Euler’s method** The forward Euler’s method is an explicit first-order Runge-Kutta method with one stage. It is the most basic form of explicit methods for numerical integration of ordinary differential equations and it is also the simplest Runge-Kutta method,

$$x(t_{k+j+1}) = x(t_{k+j}) + f(x(t_{k+j})), \quad (3.45)$$

for $j = 0, \ldots, N_p - 1$. Applying the numerical integration scheme to the ordinary differential equations (3.34), one can derive the following difference equations,

$$x(t_{k+j+1}) = x(t_{k+j}) + h f(x(t_{k+j}), u(t_{k+j}), p_s(t_k), t_{k+j}), \quad (3.46)$$
for \( j = 0, \ldots, N_p - 1 \) and with initial conditions \( x(t_k) \) known. The step size \( h \) is chosen to be equidistant and equal to the sampling interval \( t_s \). One step of the Euler’s method is then \( t_{k+j+1} = t_{k+j} + h \) or \( t_{k+j} = t_k + j h \).

- **Heun’s method** The Heun’s method is an explicit second-order Runge-Kutta method with two stages. More specifically, the Heun’s method is a predictor-corrector method using the forward Euler’s method for the prediction step in (3.47) and the trapezoidal method for the correction step in (3.48),

\[
\begin{align*}
\tilde{x}(t_{k+j+1}) &= x(t_{k+j}) + h f(x(t_{k+j})), \\
x(t_{k+j+1}) &= x(t_{k+j}) + \frac{h}{2} \left[ f(x(t_{k+j})) + f(x(t_{k+j+1})) \right],
\end{align*}
\]

for \( j = 0, \ldots, N_p - 1 \). It can be seen as an extension of the Euler’s method into a two-stage second-order Runge-Kutta method. Applying the numerical integration scheme to the ordinary differential equations (3.34), the first calculation represents the prediction step incorporating the forward Euler’s method,

\[
\tilde{x}(t_{k+j+1}) = x(t_{k+j}) + h f(x(t_{k+j}), u(t_{k+j}), p_s(t_k), t_{k+j}),
\]

followed by the correction step incorporating the trapezoidal method,

\[
\begin{align*}
x(t_{k+j+1}) &= x(t_{k+j}) + \frac{h}{2} \left[ f(x(t_{k+j}), u(t_{k+j}), p_s(t_k), t_{k+j}) + \ldots \\
&\quad + f(\tilde{x}(t_{k+j+1}), u(t_{k+j+1}), p_s(t_k), t_{k+j+1}) \right],
\end{align*}
\]

for \( j = 0, \ldots, N_p - 1 \) and with initial conditions \( x(t_k) \) known. Corresponding to the order of the methods, the accuracy of the Euler’s method improves only linearly with the step size being decreased, whereas the Heun’s method improves the accuracy quadratically. However, it is computationally also more expensive.

The ODE solution function \( \phi(x(t_k), U, p_s(t_k), t) \) for the direct single shooting approach and \( \phi(x(t_k), X, U, p_s(t_k), t) \) for the direct multiple shooting approach may also be computed using any other discretisation scheme in the simulation.

### Cost function

Starting from the continuous formulation of the cost function, as defined in (3.30),

\[
V(x[t_k, t_f], u[t_k, t_f], p(t_k), \nu[t_k, t_f]) \triangleq \\
\int_{t_k}^{t_f} L(x(t), u(t), p(t), \nu(t), t) \, dt + M(x(t_f), p(t_k), t_f),
\]

\[ (3.51) \]
and substituting

1. the state trajectories $x[t_k, t_f]$ of the system with the approximate numerical solution $x(t) = \phi(x(t_k), U, p_s(t_k), t) \in \mathbb{R}^n$ for $t \in [t_k, t_f]$, as defined in (3.35) for the single shooting method, in order to eliminate the ODE constraints, (For the multiple shooting method, the substitution with $x(t) = \phi(x(t_k), X, U, p_s(t_k), t) \in \mathbb{R}^n$ for $t \in [t_k, t_f]$, as defined in (3.40), corresponds to the substitution of the intermediate states $x(t_{k+1}), \ldots, x(t_{k+N_p})$ with the decision variables collected in the vector $X$.)

2. the input signals $u[t_k, t_f]$ with the function $u(t) = \mu(t, U) \in \mathbb{R}^m$ for $t \in [t_k, t_f]$, parametrised by the control trajectory parameters $U = [u(t_k)^T, \ldots, u(t_{k+N_c-1})^T]^T \in \mathbb{R}^{m\cdot N_c}$, and

3. the slack variable signals $\nu[t_k, t_f]$ with the function $\nu(t) = \eta(t, N) \in \mathbb{R}^{v_c}$ for $t \in [t_k, t_f]$, parametrised by the vector of slack variable trajectory parameters $N = [\nu(t_k)^T, \ldots, \nu(t_{k+N_p})^T]^T \in \mathbb{R}^{v_d}$,

one can derive the following continuous and parametrised formulation of the cost function,

$$V(x(t_k), \{X\}, U, p(t_k), N) \cong \int_{t_k}^{t_f} L(\phi(x(t_k), \{X\}, U, p_s(t_k), t), \mu(t, U), p(t_k), \eta(t, N), t) \, dt + \ldots$$

$$\ldots + M(\phi(x(t_k), \{X\}, U, p_s(t_k), t_f), p(t_k), t_f). \quad (3.52)$$

For brevity of the explanations, the additional state variable trajectory parameters $X$ in the argument of the expressions using the direct multiple shooting approach are indicated with braces. For the direct single shooting approach they should be omitted.

The integral in the formulation (3.52) has to be approximated with a finite sum in order to minimise the cost function subject to the constraints with numerical optimisation tools. Two discretisation methods for approximating a finite integral are briefly explained:

- **Rectangle method** The rectangle method approximates a definite integral by finding the area of a collection of rectangles weighted by the values of the function,

$$\int_{t_k}^{t_f} f(x(t)) \, dt \approx h \sum_{j=0}^{N_p-1} f(x(t_{k+j})). \quad (3.53)$$

Specifically, the interval $[t_k, t_f]$, over which the function has to be integrated, is divided into $N_p$ equidistant subintervals of length $t_s$, equal to the sampling interval. The approximation of the integral is then calculated by adding up the areas of the $N_p$ rectangles. Applying the rectangle method to the formulation (3.52) of the cost...
function, one gets the following discrete and parametrised expression,

\[
V(x(t_k), \{X\}, U, p(t_k), N) \doteq h \sum_{j=0}^{N_p-1} L(\phi(x(t_k), \{X\}, U), p(t_k), \eta(t_{k+j}, N), t_{k+j}) + \ldots
\]

\[
\ldots + M(\phi(x(t_k), \{X\}, U, p_s(t_k), t_f), p(t_k), t_f). \tag{3.54}
\]

- **Trapezoidal rule** The trapezoidal rule is a technique for approximating a definite integral and can be attributed to the family of formulas for numerical integration called Newton-Cotes formulas. Assuming a uniform grid, the interval \([t_k, t_f]\) is divided into \(N_p\) equally spaced intervals with length \(t_s\) and the approximation of the integral becomes

\[
\int_{t_k}^{t_f} f(x(t)) \, dt \approx \frac{h}{2} \sum_{j=0}^{N_p-1} f(x(t_{k+j})) + f(x(t_{k+j+1}))
\]

\[
= \frac{h}{2} \left( f(x(t_k)) + 2 f(x(t_{k+1})) + \ldots + 2 f(x(t_{k+N_p-1})) + f(x(t_{k+N_p})) \right). \tag{3.55}
\]

The application of the trapezoidal rule to the formulation (3.52) of the cost function leads to the following discrete and parametrised expression,

\[
V(x(t_k), \{X\}, U, p(t_k), N) \doteq h \sum_{j=0}^{N_p-1} w_{j+1} L(\phi(x(t_k), \{X\}, U, p_s(t_k), t_{k+j}), \mu(t_{k+j}, \eta(t_{k+j}, N), t_{k+j}) + \ldots
\]

\[
\ldots + M(\phi(x(t_k), \{X\}, U, p_s(t_k), t_f), p(t_k), t_f). \tag{3.56}
\]

where \(w = [1, 2, \ldots, 2, 1]\) is a weighting vector to obtain the correct coefficients according to (3.55).

**Constraint functions**

Starting from the continuous formulation (3.31), combining all the constraint equations into the vector of constraint functions \(G \in \mathbb{R}^w\),

\[
G = G(x(t), u(t), p(t_k), \nu(t), t), \tag{3.57}
\]

for all \(t \in [t_k, t_f]\), and analogously to the discretisation of the cost function, substituting
1. the state trajectories $x[t_k, t_f]$ of the system with the approximate numerical solution 
$x(t) = \phi(x(t_k), U, p_s(t_k), t) \in \mathbb{R}^n$ for the single shooting method and with $x(t) = \phi(x(t_k), X, U, p_s(t_k), t) \in \mathbb{R}^n$ for the multiple shooting method,

2. the input signals $u[t_k, t_f]$ with the function $u(t) = \mu(t, U) \in \mathbb{R}^m$, and

3. the slack variable signals $\nu[t_k, t_f]$ with the function $\nu(t) = \eta(t, N) \in \mathbb{R}^n$,

one can derive the following continuous and parametrised formulation of the constraint functions,

$$G(x(t_k), \{X\}, U, p(t_k), N, t) = G(\phi(x(t_k), \{X\}, U, p_s(t_k), t), \mu(t, U), p(t_k), \eta(t, N), t). \quad (3.58)$$

Relaxing the constraints to hold only at the time instants $\{t_k, t_{k+1}, \ldots, t_{k+N_p}\} \subset [t_k, t_f]$, the continuous formulation (3.58) can be discretised leading to a finite number of discrete and parametrised constraint functions. As the constraint equations can be classified into various categories and undergo different discretisation procedures, the appropriate time instants for the discretisation vary between the different constraint equations in order to avoid redundant discrete constraint equations.

For the dimensionless bound constraints on the states $x(t)$, the continuous equations are parametrised and evaluated at each time instant within the prediction horizon $(j = 0, \ldots, N_p)$ leading to $B_x \in \mathbb{R}^{w_{dx}, b_x}$,

$$B_x = G_{B_x}(x(t_k), \{X\}, U, p(t_k), N) = \begin{bmatrix} g_{b_x}(x(t_k), \{X\}, U, p(t_k), N, t_k) \\
\vdots \\
g_{b_x}(x(t_k), \{X\}, U, p(t_k), N, t_{k+N_p}) \end{bmatrix}. \quad (3.59)$$

For the dimensionless bound constraints on the inputs $u(t)$, the continuous equations are parametrised and evaluated at each time instant within the control horizon $(j = 0, \ldots, N_c - 1)$ leading to $B_u \in \mathbb{R}^{w_{du}, b_u}$,

$$B_u = G_{B_u}(x(t_k), \{X\}, U, p(t_k), N) = \begin{bmatrix} g_{b_u}(x(t_k), \{X\}, U, p(t_k), N, t_k) \\
\vdots \\
g_{b_u}(x(t_k), \{X\}, U, p(t_k), N, t_{k+N_c-1}) \end{bmatrix}. \quad (3.60)$$

For the dimensionless bound constraints on the slack variables $\nu(t)$, the continuous equations are parametrised and evaluated at each time instant within the prediction
horizon \((j = 0, \ldots, N_p)\) leading to \(B_\nu \in \mathbb{R}^{w_{d,b\nu}}\),

\[
B_\nu = G_{B_\nu}(x(t_k), \{X\}, U, p(t_k), N) = \begin{bmatrix} g_{b_\nu}(x(t_k), \{X\}, U, p(t_k), N, t_k) \\ \vdots \\ g_{b_\nu}(x(t_k), \{X\}, U, p(t_k), N, t_{k+N_p}) \end{bmatrix}.
\]

However, when assuming constant slack variables over the prediction horizon, leading to a vector of slack variable trajectory parameters \(N = \nu(t_k)\), the evaluation of the constraints is only necessary at the “current” time instant \(t = t_k\) \((j = 0)\).

For the dimensionless path constraints on the states \(x(t)\), the continuous equations are parametrised and evaluated at each time instant within the prediction horizon \((j = 0, \ldots, N_p)\) leading to \(P_x \in \mathbb{R}^{w_{d,p_x}}\),

\[
P_x = G_{P_x}(x(t_k), \{X\}, U, p(t_k), N) = \begin{bmatrix} g_{p_x}(x(t_k), \{X\}, U, p(t_k), N, t_k) \\ \vdots \\ g_{p_x}(x(t_k), \{X\}, U, p(t_k), N, t_{k+N_p}) \end{bmatrix}.
\]

For the dimensionless path constraints on the inputs \(u(t)\), the continuous equations are parametrised and evaluated at each time instant within the control horizon \((j = 0, \ldots, N_c - 1)\) leading to \(P_u \in \mathbb{R}^{w_{d,p_u}}\),

\[
P_u = G_{P_u}(x(t_k), \{X\}, U, p(t_k), N) = \begin{bmatrix} g_{p_u}(x(t_k), \{X\}, U, p(t_k), N, t_k) \\ \vdots \\ g_{p_u}(x(t_k), \{X\}, U, p(t_k), N, t_{k+N_c-1}) \end{bmatrix}.
\]

For the dimensionless path constraints on both the inputs \(u(t)\) and the states \(x(t)\), the continuous equations are parametrised and evaluated at each time instant within the prediction horizon \((j = 0, \ldots, N_p)\) leading to \(P_{ux} \in \mathbb{R}^{w_{d,p_{ux}}}\),

\[
P_{ux} = G_{P_{ux}}(x(t_k), \{X\}, U, p(t_k), N) = \begin{bmatrix} g_{p_{ux}}(x(t_k), \{X\}, U, p(t_k), N, t_k) \\ \vdots \\ g_{p_{ux}}(x(t_k), \{X\}, U, p(t_k), N, t_{k+N_p}) \end{bmatrix}.
\]

All these definitions for the inequality constraints can be summarised in the vector \(G_i \in \mathbb{R}^{w_{d,i}}\) of discrete and parametrised inequality constraint functions,

\[
G_i = G_i(x(t_k), \{X\}, U, p(t_k), N),
\]

where \(w_{d,i} = w_{d,bx} + w_{d, bu} + w_{d, b\nu} + w_{d, p_x} + w_{d, p_u} + w_{d, p_{ux}}\).
For arbitrary dimensionless equality constraints, the continuous equations are parametrised and evaluated at each appropriate time instant within the prediction horizon leading to $E_{ux} \in \mathbb{R}^{w_{d,e}ux}$.

In the direct multiple shooting approach, the discrete dynamic equality constraints $D_{ux} \in \mathbb{R}^{w_{d,d}ux}$, as introduced in (3.44), need to be included in the formulation.

All these definitions for the equality constraints can be summarised in the vector $G_e \in \mathbb{R}^{w_{d,e}}$ of discrete and parametrised equality constraint functions,

$$G_e = G_e(x(t_k), \{X\}, U, p(t_k), N),$$

(3.66)

where $w_{d,e} = w_{d,e}ux + w_{d,d}ux$.

Similarly, all constraint functions can be summarised in the vector of discrete and parametrised (inequality and equality) constraint functions,

$$G = [G_i \ G_e], \quad G \in \mathbb{R}^{w_d},$$

(3.67)

where $w_d = w_{d,i} + w_{d,e}$.

### 3.3.3 Formulation as an mp-NLP problem

The finite parametrisation and discretisation is applied to the cost function (3.30) and the constraint functions (3.31). Moreover, the integration scheme is applied to the ordinary differential equations (3.32) and they are eliminated by employing the direct single shooting method or replaced with non-linear equality constraints by employing the direct multiple shooting approach. Then, the optimisation problem can be concluded as follows,

$$V^*(x(t_k), p(t_k)) = \min_{U,N,\{X\}} V(x(t_k), \{X\}, U, p(t_k), N),$$

(3.68)

subject to

$$G(x(t_k), \{X\}, U, p(t_k), N) \leq 0.$$  

(3.69)

Depending on the choice of the discretisation method, the formulation (3.54) or (3.56) is employed for the cost function and the formulation (3.46) or (3.49)-(3.50) is used for the numerical integration of the ODE constraints.

The problem (3.68)-(3.69) is a non-linear program in $U$ and $N$ for the single shooting method and $U, N,$ and $X$ for the multiple shooting method, parametrised by the initial state vector $x(t_k)$ and the vector of system and controller parameters $p(t_k)$. Hence, it is in the multi-parametric non-linear programming form (2.1)-(2.2). The receding horizon MPC strategy will, therefore, re-optimise $U$ and $N$ in the single shooting approach.
and $U$, $N$, and $X$ in the multiple shooting approach, respectively, when new state or external input information appears, typically periodically at each sample. We assume the solution exists, and let it be denoted $U^*$ and $N^*$ for the single shooting method, and $U^*$, $N^*$, and $X^*$ for the multiple shooting method.

**Vector of parameters** The vector of parameters $x_p(t_k) \in \mathbb{R}^{n_p}$ is introduced, where $n_p = n + d$, combining the vector of initial states $x(t_k) \in \mathbb{R}^n$ and the vector of system and controller parameters $p(t_k) \in \mathbb{R}^d$, leading to

$$x_p(t_k) = \begin{bmatrix} x(t_k) \\ p(t_k) \end{bmatrix}. \quad (3.70)$$

**Vector of decision variables** Similarly, the vector of decision variables $z \in \mathbb{R}^s$ is introduced, combining the vector of input trajectory parameters $U$ and the vector of slack variable trajectory parameters $N$, leading to

$$z = \begin{bmatrix} U \\ N \end{bmatrix}, \quad (3.71)$$

in the direct single shooting approach, and additionally combining the vector of state variable trajectory parameters $X$, leading to

$$z = \begin{bmatrix} U \\ N \\ X \end{bmatrix}, \quad (3.72)$$

in the direct multiple shooting approach.

Assuming a piecewise constant input signal being parametrised by the vector $U$, such that $u(t) = \mu(t, U) \in \mathbb{R}^m$ is continuous on every sampling interval $t_s$, the vector of control trajectory parameters $U$ is defined as

$$U = \begin{bmatrix} u(t_k) \\ \vdots \\ u(t_k+N_c-1) \end{bmatrix}, \quad U \in \mathbb{R}^{mN_c}. \quad (3.73)$$

Assuming piecewise constant slack variables being parametrised by the vector $N$, such that $\nu(t) = \eta(t, N) \in \mathbb{R}^{v_c} is continuous on every sampling interval $t_s$, the vector of slack
variable parameters $N$ is defined as

$$N = \begin{bmatrix} \mathbf{v}(t_k) \\ \vdots \\ \mathbf{v}(t_{k+N_p}) \end{bmatrix}, \quad N \in \mathbb{R}^{v_d}, \quad (3.74)$$

where $v_d = v_c \cdot (N_p + 1)$. Assuming a state variable trajectory being parametrised by the vector $X$, such that $x(t) = \phi(x(t_k), X, U, p_s(t_k), t) \in \mathbb{R}^n$ is continuous over the entire horizon, the vector of state trajectory parameters $X$ is defined as

$$X = \begin{bmatrix} x(t_{k+1}) \\ \vdots \\ x(t_{k+N_p}) \end{bmatrix}, \quad X \in \mathbb{R}^{nN_p}. \quad (3.75)$$

Considering the assumptions $(3.73)-(3.75)$, the vector of decision variables $z \in \mathbb{R}^s$ is of dimension $s = m N_c + v_d$ for the direct single shooting method and of dimension $s = m N_c + v_d + n N_p$ for the direct multiple shooting method.

**Transformation law** For the conversion between the dimension-related and the dimensionless vector of decision variables and vector of parameters, the characteristic quantities of the individual transformation laws are combined to a transformation vector.

By applying the non-dimensionalisation procedure to each element of the parameter vector, one can derive the following relation,

$$X_p(T_k) = X_{p,ch} x_p(t_k), \quad (3.76)$$

with the dimensional transformation vector of characteristic quantities $X_{p,ch}$ for the parameter vector. Employing (3.76), the transformation from the dimension-related vector $X_p(T_k)$, combining the initial states $X(T_k)$ and the vector system and controller parameters $P(T_k)$, to the dimensionless vector $x_p(t_k)$ and vice versa is uniquely defined.

The vector of decision variables was only introduced in the dimensionless form so far. However, equivalently to (3.76), the dimensionless vector $z$ can also be transformed to a dimension-related vector $Z$ and vice versa. By applying the individual transformation laws of the non-dimensionalisation procedure to each element of the vector, one can derive,

$$Z = Z_{ch} z, \quad (3.77)$$

with the dimensional transformation vector of characteristic quantities $Z_{ch}$ for the decision variable vector.
Multi-parametric non-linear program

By employing the definition (3.70) for the vector of parameters $x_p \in \mathbb{R}^{n_p}$, and (3.71) and (3.72), respectively, for the vector of decision variables $z \in \mathbb{R}^s$, one can reformulate the optimisation problem (3.68)-(3.69), leading to the multi-parametric non-linear mathematical programming (mp-NLP) problem,

$$V^*(x_p(t_k)) = \min_z V(z, x_p(t_k)),$$

subject to

$$G(z, x_p(t_k)) \leq 0,$$

minimised with respect to the optimisation variables $z$ and parametrised by the vector of parameters $x_p(t_k)$ with the objective function $V: \mathbb{R}^s \times \mathbb{R}^{n_p} \mapsto \mathbb{R}$ and the constraint functions $G: \mathbb{R}^s \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^w$.

### 3.3.4 Tuning and stability

#### Feasibility

The formulation of the optimal control problem leads to the non-linear mathematical programming problem (3.78)-(3.79). The existence of a solution to (3.78)-(3.79) corresponds to the feasibility of the optimisation problem, with the feasible set of parameters, $X_f$, defining the set that contains all parameters, $x_p$, for which the optimisation problem has a solution $z^*$. One wants to make this set as large as possible, while still fulfilling the physical and operational constraints.

Inappropriate choices in the design of the optimal control problem or controllability issues may lead to infeasibility, i.e. no solution exists. Moreover, it can also lead to poor conditioning or indefiniteness of the Hessian matrix. Since it is in general impossible to verify the optimality conditions a priori, these important issues need to be analysed in the algorithms for the approximate solution of the multi-parametric programming problems. This is a considerable advantage of the explicit approach to assess already in the design phase the quality of the optimal control problem formulation including the tunings choices.

In a practical approach, the feasibility should be ensured as far as possible by relaxing the constraints when needed and when possible. Generally, operational constraints can be relaxed according to the design specifications. Moreover, stability-enforcing terminal constraints may also be relaxed or even omitted completely in practice. They may tend to be conservative and are often not needed in particular for open-loop stable systems. However, physical constraints, on the contrary, can never be relaxed.

A straightforward approach to guarantee feasibility is the use of slack variables, as introduced in Section 3.1. The weights on the slack variables are usually chosen such that their penalty terms in the cost function dominate the remaining terms. This ensures that
the soft constraints enhancing feasibility are not relaxed when not needed. Additional constraints on the slack variables guarantee that they are positive semi-definite to only relax and not tighten the soft constraints. The use of slack variables increases the number of decision variables, which has generally no implications in the explicit approach as opposed to implicit NMPC. Finally, the feasibility of the optimisation problem is also important for the generation of the approximate solution to avoid fine partitioning based on the heuristic splitting rules leading to an unnecessary increase in the number of parameter space partitions.

**Tuning**

The dynamic performance and control functionality of the non-linear MPC depends on the cost function and the constraint functions. The cost function is typically $l_2$ type, as introduced in (3.12) and (3.20), or $l_1$ type. The properties of the weight matrices, in particular of $q_x > 0$ and $r_u > 0$, are essential for the performance, and in some cases also stability. All states and inputs are observable through the cost function and its minimisation will influence all the states that are controllable. For the stabilisation of unstable dynamics, it is, in fact, sufficient that only the unstable modes of the dynamic system are observable through the cost function. To ensure uniqueness of the control variable trajectory, it is generally recommended that $r_u > 0$. In summary, conventional LQR and MPC tuning guidelines are useful as a starting point also for the explicit non-linear MPC approach. In the following, some general practice is summarised (MathWorks, 2019a,b):

- As a first step, it is recommended to choose, if possible and not specified by hardware or other requirements, the sampling time $T_s$ and hold it constant during the tuning of the remaining parameters. In case the initial choice was poor, the value needs to be reviewed which also implies retuning of the other parameters. In terms of unknown disturbance rejection, the performance typically improves and then plateaus for decreasing values of $T_s$. However, since the sampling time is assumed to be equal to the discretisation time $H$ of the dynamic system, the choice considerably depends on the dynamic characteristics of the system as well as on the choice of the numerical integration scheme. As the sampling time decreases, the computational effort increases with the growing number of prediction steps, assuming a constant prediction horizon $T_p = N_p T_s$. As a rough guideline, the sampling time $T_s$ can be chosen between 10\% and 25\% of the minimum desired closed-loop response time. The unmeasured disturbance rejection should be assessed for significant improvements when the sampling time is halved.
With the sampling time $T_s$ chosen, in the next step the prediction horizon $T_p$ and the number of prediction steps $N_p$, respectively, have to be determined. It is recommended to increase the number of prediction steps until a further increase only has a minor effect on the performance. As a rough guideline, the prediction horizon $T_p$ should be close to the desired closed-loop response time. However, since the complexity of the explicit controller in terms of polyhedral regions significantly grows with an increasing number of prediction steps, which goes along with an increasing number of constraints, it is important to keep the number of $N_p$ as low as possible. Here, the optimal choice is a balance of explicit controller complexity and performance.

Further on, the control horizon needs to be chosen by defining the number of control steps $N_c$. Low numbers for the control steps result in fewer decision variables to compute in the mathematical program and generally promote internal stability. Equivalently to $N_p$, the number of control steps $N_c$ also has a significant influence on the complexity of the explicit controller and needs to be chosen small for that reason.

Before choosing the cost function weights, it is recommended to specify scaling factors, as introduced in Section 3.2.1. The scaling factors are chosen for each dependent variable, i.e. input, state, and slack variables, and potentially also for the time as an independent variable. The factors are held constant during the tuning of the remaining parameters.

Finally, the weights on the output and the manipulated variables can be determined. For ease of interpretation, the tuning procedure can be started with a plant that is identical to the prediction model to avoid any prediction errors followed by refinements on the actual plant. Moreover, for the preliminary tests, the constraints can be disabled. For reference tracking, satisfactory tracking behaviour can be achieved by increasing the respective weights. To avoid tracking error at steady-state, the degree-of-freedom should be considered, i.e. the number of manipulated variables with respect to the number of output variables to be tracked. Also, depending on the number of input constraints respective measures needs to be taken if necessary. Lastly, the weights on the manipulated variables are chosen. The objectives for output and manipulated variables tracking are likely to conflict. Therefore, the weights can be chosen to not compromise the tracking performance of the output variables with, however, non-zero values to ensure uniqueness of the control variable trajectory.

In the last step, the sensitivity of the controller with respect to prediction errors as well as the robustness of the controller against parameter variations of the plant needs to be assessed. For these simulations, the actual plant is employed and evaluated for different operating conditions. In case the controller performance degrades significantly
due to the prediction errors, retuning of the controller to make it less aggressive leading to more conservative control moves might be considered for an open-loop stable plant. For the robustness, relevant parameters of the plant may be varied in a reasonable range and the controller performance assessed in different operating conditions.

**Stability**

The non-linear model predictive controller is based on the receding horizon philosophy in which an open-loop finite horizon optimal control solution is applied until a new optimised control trajectory is available at the next sampling instant. Computing the new optimised control trajectory based on the most recent state information leads to closed-loop control. However, the stability of the closed-loop is a priori not guaranteed by the MPC law based on the optimal open-loop trajectory. With unfortunate choices of the design parameters, the closed-loop MPC may be unstable.

There are some general principles that are useful to ensure the stability of the non-linear MPC, as summarised in Mayne et al. (2000). The control trajectory should be for instance sufficiently rich, with most of the theoretical work assuming piecewise constant trajectories. Generally, an infinite horizon cost does have stabilising properties, however, leading to an infinite dimensional control problem. More practical approaches assume sufficiently large horizons $T_p$. Grüne (2009) and Reble and Allgöwer (2012) present a stability analysis for unconstrained, in the sense of stability preserving constraints, implicit NMPC schemes. Another practical approach is to choose the Mayer term in such a way that it approximates the cost-to-go, i.e. $M(\mathbf{X}(T_f), \mathbf{P}(T_k), T_f) \approx \int_{T_f}^{\infty} L(\mathbf{X}(T), \mathbf{U}(T), \mathbf{P}(T_k), \mathbf{N}(T), T) \, dt$, such that the total cost function approximates an infinite horizon cost. Unfortunately, the cost-to-go is hard to compute in general and simple approximations can be chosen instead. Moreover, terminal set constraints, as introduced in (3.9) and (3.17), can be employed. The terminal set is designed in such a way that the system is subject to an actual or hypothetical feasible and stabilising controller once the states are inside the set making the system asymptotically stable. Similarly, terminal equality constraints can be employed ensuring convergence in finite time. Those constraints, however, are demanding and may, therefore, compromise feasibility. The idea of a stabilising terminal constrained set and a terminal cost approximating the cost-to-go may be combined, as elaborated in more detail in Chen and Allgöwer (1998).

However, the mentioned approaches are for implicit NMPC schemes. To the best of the author’s knowledge, there is no comparable practical NMPC theory in the literature addressing the stability of the approximate solution to explicit NMPC. Moreover, most of the approaches that ensure the closed-loop stability of the NMPC law are implemented successfully only after restrictive assumptions, such as regulation to a constant set-point.
However, with time-varying set-points in transient manoeuvres, the recursive feasibility, as one of the stability conditions, may not be fulfilled.

**Robustness**

General experience shows that model predictive control tends to be robust, even without specific considerations in the design phase. However, an adequate identification of the system dynamics and realistic formulation of the constraints and the cost function is a requisite. There are mechanisms to handle steady-state modelling errors. One of such is implemented in Chapter 6 as an integral action. Moreover, in this work, the robustness and stability of the explicit NMPC controller are empirically assessed through a sensitivity analysis varying some parameters of the simulation model. Starting from the nominal conditions relevant parameters of the plant are varied in a reasonable range.

### 3.4 Numerical optimisation

The main motivation of the explicit approach to NMPC is to avoid numerical optimisation online and move the computational load offline. Therefore, the choice of the non-linear programming method is important for the generation of the explicit approximate solution.

For the solution of a non-linear mathematical programming problem to be optimal, some optimality conditions need to be satisfied. For a given parameter vector $x_{p,0} \in X$, a local minimum $z_0 = z^*(x_{p,0})$ of the optimisation problem (3.78)-(3.79) has to satisfy the well-known Karush-Kuhn-Tucker (KKT) necessary first-order conditions comprising stationarity, primal feasibility, dual feasibility, and complementary slackness. Assuming sufficient regularity and smoothness, the necessary first-order KKT conditions are sufficient provided the second-order conditions hold true. They examine the Hessian of the Langrangian for positive semi-definiteness in the directions for which the first order conditions cannot specify whether the objective function will increase or decrease.

#### 3.4.1 Non-linear programming methods

There are numerous methods to numerically compute a local minimum $z_0$ of the problem (3.78)-(3.79) for a given parameter $x_{p,0}$. Here, only a brief overview is given of the various methods. Comprehensive overviews can be found in Grancharova and Johansen (2012) or Leyffer and Mahajan (2010).

**Newton-type methods**

Newton-type methods seem the most widely used optimisation methods. Based on successive linearisations of the KKT system and using standard numeric linear algebra
tools, they try to find a point satisfying the KKT conditions. The following two main groups can be differentiated:

**Sequential quadratic programming methods** The sequential quadratic programming (SQP) methods iteratively solve the KKT system by linearising the included non-linear functions. In each iteration, a quadratic program (QP), defined by the Hessian of the Lagrangian, is considered and a unique solution can be found in case the QP problem is convex with the Hessian matrix being positive semi-definite. In the quasi-Newton methods, an approximation of the Hessian matrix and the constraints Jacobian matrix is employed leading to typically slower convergence but computationally less expensive iterations. The most popular method to update the Hessian matrix is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula.

**Interior-point methods** An alternative method to solve the KKT system are interior-point (IP) methods replacing the non-smooth KKT conditions of complementary slackness with a smooth non-linear approximation employing slack variables. The problem is, therefore, transformed to a problem with a so-called barrier function, which is positive semi-definite and continuous over the region, setting a barrier against leaving the feasible region by growing to infinity as the boundary is approached from the interior. By reducing the slack variable once a solution was found for a given value, an accurate solution to the original problem is obtained by a limited number of Newton iterations.

In the implementation of the algorithm, an interior-point method is employed to solve the non-linear programming problems. The chosen solver is the interior-point solver IPOPT, a line-search filter interior-point method based on Wächter and Biegler (2005b), Wächter and Biegler (2005a), and Wächter and Biegler (2006). More details and a comprehensive overview including further mathematical programming methods can be found in Leyffer and Mahajan (2010). In the implementation, function and gradient information is provided, as well as the Hessian of the Lagrangian.

**Penalty function methods**

Penalty function methods transform the constrained problem into an unconstrained problem by incorporating the constraints into the cost function with a parameter penalising any violations of the constraints. A penalty is only desired if the point \( z \) is not feasible and, therefore, the methods are also referred to exterior penalty function methods. To avoid difficulties due to numerical ill-conditioning associated with large penalty parameters, common schemes solve a sequence of problems with increasing penalty parameters. For a given parameter, the problem can be solved with the steepest descent method.
Direct search methods

Direct search methods do not rely on the gradient of the objective function and, thus, represent derivative-free methods for optimisation. They are based on the values of the objective function and the constraints on a set of sample points, the comparison of those trial solutions with previous trial solutions, and a strategy for the next trial solutions.

The most popular direct search methods for constrained optimisation are the Complex method of Box, an extension of the Simplex method, and the DIRECT (DIViding RECTangles) method. The DIRECT algorithm is a direct search method for global optimisation and is based on partitioning with hyper-rectangles and an algorithm for the selection and trisection of the partitions.

3.4.2 Jacobian and Hessian matrices

The computation of Jacobian and Hessian matrices usually requires the main computational effort in numerical optimisation methods. The matrices are also essential for the multi-parametric quadratic approximations of mp-NLP, introduced in Chapter 4. Moreover, even small inaccuracies in the calculations of Jacobian matrices may lead to convergence problems. Here, it is worthwhile to mention that the scaling and the non-dimensionalisation of all variables is in many cases a prerequisite for numerical optimisation methods to work satisfactorily.

Typically, the matrices are obtained by the application of the finite difference method. Numerical errors and convergence need to be considered in this approach. However, the most accurate results together with considerable reductions of the necessary computational effort can be obtained by symbolic differentiation of the cost and constraints functions. Since the formulation of the optimal control problem is performed via symbolic manipulation in Maple in the implementation of the toolbox, the Jacobian and Hessian matrices are obtained via symbolic differentiation and numeric evaluation. This approach is employed for both the solution of the non-linear programming problems and for the definition of the approximating multi-parametric quadratic programming problems. However, symbolic differentiation may easily become intractable for problems with complex expressions or a large number of decision variables, equations, and inequalities. In this respect, the simultaneous approach, introduced in Section 3.3.2, is beneficial for the generation of the symbolic matrices since the complexity of expressions for the cost function and the constraint functions is considerably lower and the symbolic differentiation can be more easily exploited.
Chapter 4

Explicit non-linear model predictive control

In this chapter, the suggested numerical algorithm for approximate multi-parametric non-linear programming is presented. The multi-parametric non-linear program (mp-NLP) is locally approximated with a multi-parametric quadratic program (mp-QP), leading to an approximate solution to the mp-NLP being composed of the solution of local mp-QP sub-problems. The approximate solution can be represented as an explicit piecewise affine function only dependent on the problem parameters. In the context of non-linear model predictive control, the piecewise affine state feedback law allows an efficient on-line implementation without any real-time optimisation avoiding, therefore, the solution of the non-linear programming problem numerically at every sampling interval as required in the implicit approach to NMPC.

4.1 Sub-optimal solution using mp-QP approximations of mp-NLP

The exact solution of a multi-parametric non-linear program is only possible in a very limited number of cases or it is even an impossible task. Here, an approximate mp-NLP algorithm is suggested using NLP and mp-QP algorithms. The algorithm is a complementary combination of the approximate mp-NLP algorithm incorporating global optimisation tools, as proposed in Grancharova et al. (2007) and Grancharova and Johansen (2012), together with the mp-QA algorithm, as proposed in Johansen (2002) and Grancharova and Johansen (2012).

The main idea in the approximate mp-NLP algorithm is to locally approximate the mp-NLP with mp-QPs, similar to the use of QPs in the Sequential Quadratic Programming (SQP) approach for the numerical solution of non-linear programming.
Explicit non-linear model predictive control (NLP) problems. The accuracy of the approximation is controlled via an iterative and recursive partitioning of the parameter space. In order to meet the selected tolerances and accuracy specifications in terms of sub-optimality bounds on the cost, the solution, and the maximum constraint violation, the partitions are refined in parts of the parameter space to improve the accuracy of the local mp-QP approximation.

The mp-NLP problem is formulated as follows,

\[
V^*(x_p) = \min_z V(z, x_p), \tag{4.1}
\]

subject to

\[
G(z, x_p) \leq 0, \tag{4.2}
\]

for all \( x_p \in \mathbb{X} \), where \( \mathbb{X} \) is a set of parameters. Equations (4.1)-(4.2) define the mp-NLP problem, since it is an NLP problem in the optimisation variables \( z \), parametrised by the parameter vector \( x_p \). Assume that the exact solution to the mp-NLP problem exists and let it be denoted \( z^*(x_p) \).

In the special case when the cost function \( V \) is quadratic and the constraint functions \( G \) are linear, in both \( z \) and \( x_p \), the problem changes to an mp-QP problem and an exact explicit solution can be found leading to a continuous piecewise affine mapping \( z^*(x_p) \). The mp-QP algorithm builds polyhedral partitions of the parameter-space in an iterative manner with associated exact solutions corresponding to a fixed active set within the polyhedral critical regions. This leads to a piecewise affine solution \( z^*(x_p) \), due to the fact that a fixed active set of constraints leads to a solution that is affine in \( x_p \).

In the general non-linear case, described by the mp-NLP problem (4.1)-(4.2), the original mp-NLP problem is locally approximated by mp-QP sub-problems on orthogonal partitions. On each orthogonal partition, the exact non-linear solution is approximated by a piecewise affine solution found by solving the corresponding mp-QP sub-problem, constructed as a locally accurate quadratic approximation to \( V \) and linear approximation to \( G \). The solution of the mp-QP sub-problems is conducted by an mp-QP algorithm in the same manner as described beforehand. In order to keep the accuracy specifications such as cost error, solution error, and constraint violation within the specified bounds, sub-partitioning of the orthogonal partitions is introduced when needed. Under regularity and smoothness assumptions on \( V \) and \( G \), the approximation error is expected to be small, if the corresponding partitions are sufficiently small. The approximation error is for this reason analysed within each of the partitions. In the mp-NLP case, the approximate solution keeps the piecewise affine structure with associated polyhedral critical regions being continuous on each partition and, however, being in general discontinuous across the partitions, Grancharova and Johansen (2012).
4.2 Local mp-QP approximation to mp-NLP

In this section, the local approximation of the mp-NLP problem with an mp-QP problem is derived from Taylor series expansion of the cost function $V$ and the constraint functions $G$ about the point $(z_0, x_{p,0})$. Let the point $x_{p,0} \in \mathbb{X}$ be arbitrary and denote the corresponding optimal solution $z_0 = z^*(x_{p,0})$.

4.2.1 Local mp-QP approximation

In order to unambiguously specify the derivations of the equations, the following definitions for the dyadic product and the gradient are introduced.

Dyadic product

The dyadic product specifies one way to multiply two Euclidean vectors. With the two vectors, $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$, it returns a second-order tensor, called a dyadic in this context,

$$\otimes : \mathbb{R}^m \times \mathbb{R}^n \mapsto \mathbb{R}^{m \times n}, \quad (a, b) \mapsto a \otimes b. \quad (4.3)$$

Consider the two vectors,

$$a = \sum_{i=1}^{m} a_i \mathbf{e}_i = a_1 \mathbf{e}_1 + a_2 \mathbf{e}_2 + \ldots + a_m \mathbf{e}_m, \quad (4.4)$$

$$b = \sum_{j=1}^{n} b_j \mathbf{e}_j = b_1 \mathbf{e}_1 + b_2 \mathbf{e}_2 + \ldots + b_n \mathbf{e}_n, \quad (4.5)$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$, respectively, are the standard basis vectors in the Euclidean space. Then, the dyadic product of $a$ and $b$ can be represented as a sum in algebraic form,

$$a \otimes b = \sum_{i=1}^{m} \sum_{j=1}^{n} a_i b_j \mathbf{e}_i \mathbf{e}_j, \quad (4.6)$$

or in matrix form,

$$a \otimes b = a \cdot b^T = \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} \cdot \begin{bmatrix} b_1 & \ldots & b_n \end{bmatrix} = \begin{bmatrix} a_1 b_1 & \ldots & a_1 b_n \\ \vdots & \ddots & \vdots \\ a_m b_1 & \ldots & a_m b_n \end{bmatrix}. \quad (4.7)$$

Gradient

With the definition of the dyadic product, the (left) gradient of a multivariate function can be introduced, representing a multi-variable generalisation of the derivative. If
Explicit non-linear model predictive control

\[ f(x_1, x_2, \ldots, x_l) \] is a differentiable, real-valued function of several variables, its gradient is the vector whose components are the \( l \) partial derivatives of \( f \). For a scalar-valued function \( f : \mathbb{R}^l \mapsto \mathbb{R} \), the gradient can be defined in algebraic and matrix form,

\[
\text{grad}_x(f) = \nabla_x(f) = \frac{\partial(f)}{\partial x_i} e_i, \quad \text{and} \quad \left[ \nabla_x(f) \right] = \begin{bmatrix} \frac{\partial(f)}{\partial x_1} \\ \vdots \\ \frac{\partial(f)}{\partial x_l} \end{bmatrix},
\]

respectively. For a vector-valued function \( f : \mathbb{R}^l \mapsto \mathbb{R}^n \), the gradient can be defined in algebraic and matrix form,

\[
\text{grad}_x(f) = \nabla_x(f) = \frac{\partial(f_i)}{\partial x_j} e_j \otimes e_i, \quad \text{and} \quad \left[ \nabla_x(f) \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_l} & \cdots & \frac{\partial f_n}{\partial x_l} \end{bmatrix},
\]

respectively. Finally, for a matrix-valued function \( F : \mathbb{R}^l \mapsto \mathbb{R}^{m \times n} \), the gradient can be defined in algebraic form,

\[
\text{grad}_x(F) = \nabla_x(F) = \frac{\partial(F_{ij})}{\partial x_k} e_k \otimes e_i \otimes e_j.
\]

Taylor series expansion of the cost function

The cost function \( V : \mathbb{R}^s \times \mathbb{R}^{np} \mapsto \mathbb{R} \) can be approximated by using a finite number of terms of its Taylor series. The second-order Taylor series expansion around the point \( \mathbf{a}_0 \) can be computed,

\[
V(\mathbf{a}_0 + \Delta \mathbf{a}) = V(\mathbf{a}_0) + \left( \nabla_{\mathbf{a}} V(\mathbf{a}_0) \right)^T \Delta \mathbf{a} + \frac{1}{2!} \Delta \mathbf{a}^T H(\mathbf{a}_0) \Delta \mathbf{a} + \ldots \quad (4.11)
\]

By substituting the auxiliary vector \( \mathbf{a} \) with the vector of decision variables \( \mathbf{z} \in \mathbb{R}^s \) and the parameter vector \( \mathbf{x}_p \in \mathbb{R}^{np} \), such that \( \mathbf{a} = [\mathbf{z}^T, \mathbf{x}_p^T]^T \), one can derive in matrix form,

\[
V(\mathbf{z}_0 + \Delta \mathbf{z}, \mathbf{x}_{p,0} + \Delta \mathbf{x}_p) \approx V(\mathbf{z}_0, \mathbf{x}_{p,0}) + \ldots \\
\ldots + \left[ \begin{array}{c} \nabla_{\mathbf{z}} V(\mathbf{z}_0, \mathbf{x}_{p,0})^T \\ \nabla_{\mathbf{x}_p} V(\mathbf{z}_0, \mathbf{x}_{p,0})^T \end{array} \right] \left[ \begin{array}{c} \Delta \mathbf{z} \\ \Delta \mathbf{x}_p \end{array} \right] + \ldots \\
\ldots + \frac{1}{2} \left[ \begin{array}{c} \nabla_{\mathbf{z}}^T \\ \nabla_{\mathbf{x}_p}^T \end{array} \right] \left[ \begin{array}{c} \nabla_{\mathbf{z}} \left( \nabla_{\mathbf{z}} V(\mathbf{z}_0, \mathbf{x}_{p,0}) \right) \\ \nabla_{\mathbf{x}_p} \left( \nabla_{\mathbf{z}} V(\mathbf{z}_0, \mathbf{x}_{p,0}) \right) \end{array} \right] \left[ \begin{array}{c} \Delta \mathbf{z} \\ \Delta \mathbf{x}_p \end{array} \right]. \quad (4.12)
\]
Taylor series expansion of the constraints functions

The constraint functions $G: \mathbb{R}^n \times \mathbb{R}^{n_p} \to \mathbb{R}^{w_d}$ can be approximated using a first-order Taylor series expansion,

$$G(a_0 + \Delta a) = G(a_0) + \left( \nabla_a G(a_0) \right)^T \Delta a + \ldots$$  \hspace{1cm} (4.13)

By substituting the vector $a$ with the vector of decision variables $z$ and the vector of parameters $x_p$, as defined above, one can derive in matrix form,

$$G(z_0 + \Delta z, x_{p,0} + \Delta x_p) \approx G(z_0, x_{p,0}) + \ldots$$

$$\ldots + \left[ \left( \nabla_z G(z_{0,0}) \right)^T \left( \nabla_{x_p} G(z_{0,0}) \right)^T \right] \left[ \Delta z \Delta x_p \right].$$  \hspace{1cm} (4.14)

Full set of equations

Using the Taylor series expansion (4.12) for the cost function and (4.14) for the constraint functions, the following local approximate mp-QP problem can be obtained,

$$\hat{V}(z, x_p) \triangleq \frac{1}{2} (z - z_0)^T H_0 (z - z_0) + \left( D_0 + (x_p - x_{p,0})^T F_0 \right) (z - z_0) + Y_0(x_p),$$  \hspace{1cm} (4.15)

subject to

$$G_{i,0} (z - z_0) \leq E_{i,0} (x_p - x_{p,0}) + T_{i,0},$$  \hspace{1cm} (4.16)

$$G_{e,0} (z - z_0) = E_{e,0} (x_p - x_{p,0}) + T_{e,0}.$$  \hspace{1cm} (4.17)

In this formulation, the following matrices are used,

$$H_0 \triangleq \nabla_z \left( \nabla_z V(z_0, x_{p,0}) \right)^T,$$

$$D_0 \triangleq \left( \nabla_z V(z_0, x_{p,0}) \right)^T,$$

$$F_0 \triangleq \frac{1}{2} \left\{ \left[ \nabla_z \left( \nabla_{x_p} V(z_0, x_{p,0}) \right) \right]^T + \nabla_{x_p} \left( \nabla_z V(z_0, x_{p,0}) \right) \right\},$$

$$G_{i,0} \triangleq \left( \nabla_z G_i(z_0, x_{p,0}) \right)^T,$$

$$E_{i,0} \triangleq -\left( \nabla_{x_p} G_i(z_0, x_{p,0}) \right)^T,$$

$$T_{i,0} \triangleq -G_i(z_0, x_{p,0}),$$

$$G_{e,0} \triangleq \left( \nabla_z G_e(z_0, x_{p,0}) \right)^T,$$

$$E_{e,0} \triangleq -\left( \nabla_{x_p} G_e(z_0, x_{p,0}) \right)^T,$$

$$T_{e,0} \triangleq -G_e(z_0, x_{p,0}),$$

together with the definition,

$$Y_0(x_p) = \frac{1}{2} (x_p - x_{p,0})^T V_{xx0} (x_p - x_{p,0}) + V_{z0} (x_p - x_{p,0}) + V_0.$$  \hspace{1cm} (4.18)
and the matrices,
\[ V_{x0} = \nabla_{x_0} \left( \nabla_{x_0} V(z_0, x_{p,0}) \right), \quad V_{z0} = \left( \nabla_{x_0} V(z_0, x_{p,0}) \right)^T, \quad V_0 = V(z_0, x_{p,0}). \]  \hspace{1cm} (4.19)

With the vector of optimisation variables \( z \in \mathbb{R}^s \), the vector of parameters \( x_p \in \mathbb{R}^{n_p} \), the cost function \( V: \mathbb{R}^s \times \mathbb{R}^{n_p} \to \mathbb{R} \), the inequality constraint functions \( G_i: \mathbb{R}^s \times \mathbb{R}^{n_p} \to \mathbb{R}^{m_{i,n}} \), and the equality constraint functions \( G_e: \mathbb{R}^s \times \mathbb{R}^{n_p} \to \mathbb{R}^{m_{e,n}} \), one can specify the matrices and vectors defining the approximation of the cost function and the constraint functions:

- Matrix \( H_0 \in \mathbb{R}^{s \times s} \), given by
  \[ H_0 \triangleq \nabla_z \left( \nabla_z V(z_0, x_{p,0}) \right) = \begin{bmatrix} \frac{\partial^2 V}{\partial z_1^2} & \cdots & \frac{\partial^2 V}{\partial z_1 \partial z_s} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 V}{\partial z_s \partial z_1} & \cdots & \frac{\partial^2 V}{\partial z_s^2} \end{bmatrix} \bigg|_{(z_0,x_{p,0})} \]  \hspace{1cm} (4.20)

- Vector \( D_0 \in \mathbb{R}^{1 \times s} \), given by
  \[ D_0 \triangleq \left( \nabla_z V(z_0, x_{p,0}) \right)^T = \begin{bmatrix} \frac{\partial V}{\partial z_1} \\ \vdots \\ \frac{\partial V}{\partial z_s} \end{bmatrix} \bigg|_{(z_0,x_{p,0})}. \]  \hspace{1cm} (4.21)

- Matrix \( F_0 \in \mathbb{R}^{n_p \times s} \), given by
  \[ F_0 = \frac{1}{2} \left\{ \left( \nabla_z \left( \nabla_{x_p} V(z_0, x_{p,0}) \right) \right)^T + \nabla_{x_p} \left( \nabla_z V(z_0, x_{p,0}) \right) \right\} = \ldots \\
  \frac{1}{2} \left\{ \begin{bmatrix} \frac{\partial^2 V}{\partial x_{p,1} \partial z_1} & \cdots & \frac{\partial^2 V}{\partial x_{p,1} \partial z_s} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 V}{\partial x_{p,s} \partial z_1} & \cdots & \frac{\partial^2 V}{\partial x_{p,s} \partial z_s} \end{bmatrix}^T \bigg|_{(z_0,x_{p,0})} + \begin{bmatrix} \frac{\partial^2 V}{\partial x_{p,1} \partial z_1} & \cdots & \frac{\partial^2 V}{\partial x_{p,1} \partial z_s} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 V}{\partial x_{p,s} \partial z_1} & \cdots & \frac{\partial^2 V}{\partial x_{p,s} \partial z_s} \end{bmatrix} \bigg|_{(z_0,x_{p,0})} \right\} \]  \hspace{1cm} (4.22)

- Matrix \( V_{x0} \in \mathbb{R}^{n_p \times n_p} \), given by
  \[ V_{x0} \triangleq \nabla_{x_p} \left( \nabla_{x_p} V(z_0, x_{p,0}) \right) = \begin{bmatrix} \frac{\partial^2 V}{\partial x_{p,1}^2} & \cdots & \frac{\partial^2 V}{\partial x_{p,1} \partial x_{p,n_p}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 V}{\partial x_{p,n_p} \partial x_{p,1}} & \cdots & \frac{\partial^2 V}{\partial x_{p,n_p}^2} \end{bmatrix} \bigg|_{(z_0,x_{p,0})} \]  \hspace{1cm} (4.23)

- Vector \( V_{z0} \in \mathbb{R}^{1 \times n_p} \), given by
  \[ V_{z0} \triangleq \left( \nabla_{x_p} V(z_0, x_{p,0}) \right)^T = \begin{bmatrix} \frac{\partial V}{\partial x_{p,1}} \\ \vdots \\ \frac{\partial V}{\partial x_{p,n_p}} \end{bmatrix} \bigg|_{(z_0,x_{p,0})}. \]  \hspace{1cm} (4.24)
4.2 Local mp-QP approximation to mp-NLP

- Scalar \( V_0 \in \mathbb{R} \), given by
  \[
  V_0 \triangleq V(z_0, x_{p,0})
  \]  \hfill (4.25)

- Matrix \( G_{i,0} \in \mathbb{R}^{w_{d,i} \times s} \), given by
  \[
  G_{i,0} \triangleq \left( \nabla_z G_i(z_0, x_{p,0}) \right)^T = 
  \begin{bmatrix}
  \frac{\partial G_{i,1}}{\partial z_1} & \cdots & \frac{\partial G_{i,1}}{\partial z_s} \\
  \vdots & \ddots & \vdots \\
  \frac{\partial G_{i,w_{d,i}}}{\partial z_1} & \cdots & \frac{\partial G_{i,w_{d,i}}}{\partial z_s}
  \end{bmatrix}
  \]  \hfill (4.26)

- Matrix \( E_{i,0} \in \mathbb{R}^{w_{d,i} \times n_p} \), given by
  \[
  E_{i,0} \triangleq -\left( \nabla_{x_p} G_i(z_0, x_{p,0}) \right)^T = -
  \begin{bmatrix}
  \frac{\partial G_{i,1}}{\partial x_{p,1}} & \cdots & \frac{\partial G_{i,1}}{\partial x_{p,n_p}} \\
  \vdots & \ddots & \vdots \\
  \frac{\partial G_{i,w_{d,i}}}{\partial x_{p,1}} & \cdots & \frac{\partial G_{i,w_{d,i}}}{\partial x_{p,n_p}}
  \end{bmatrix}
  \]  \hfill (4.27)

- Vector \( T_{i,0} \in \mathbb{R}^{w_{d,i} \times 1} \), given by
  \[
  T_{i,0} \triangleq -G_i(z_0, x_{p,0})
  \]  \hfill (4.28)

- Matrix \( G_{e,0} \in \mathbb{R}^{w_{d,e} \times s} \), given by
  \[
  G_{e,0} \triangleq \left( \nabla_z G_e(z_0, x_{p,0}) \right)^T = 
  \begin{bmatrix}
  \frac{\partial G_{e,1}}{\partial z_1} & \cdots & \frac{\partial G_{e,1}}{\partial z_s} \\
  \vdots & \ddots & \vdots \\
  \frac{\partial G_{e,w_{d,e}}}{\partial z_1} & \cdots & \frac{\partial G_{e,w_{d,e}}}{\partial z_s}
  \end{bmatrix}
  \]  \hfill (4.29)

- Matrix \( E_{e,0} \in \mathbb{R}^{w_{d,e} \times n_p} \), given by
  \[
  E_{e,0} \triangleq -\left( \nabla_{x_p} G_e(z_0, x_{p,0}) \right)^T = -
  \begin{bmatrix}
  \frac{\partial G_{e,1}}{\partial x_{p,1}} & \cdots & \frac{\partial G_{e,1}}{\partial x_{p,n_p}} \\
  \vdots & \ddots & \vdots \\
  \frac{\partial G_{e,w_{d,e}}}{\partial x_{p,1}} & \cdots & \frac{\partial G_{e,w_{d,e}}}{\partial x_{p,n_p}}
  \end{bmatrix}
  \]  \hfill (4.30)

- Vector \( T_{e,0} \in \mathbb{R}^{w_{d,e} \times 1} \), given by
  \[
  T_{e,0} \triangleq -G_e(z_0, x_{p,0})
  \]  \hfill (4.31)
4.2.2 Reformulation into MPT 3.0 format

For the solution of the mp-QP problems, the MPT 3.0 toolbox (Herceg et al., 2013) is employed. The following general formulation for an mp-QP problem is considered by the Opt class, interfacing the solver of the optimisation problems. When formulating the optimisation problem, the user is supposed to provide the problem data to the Opt class that accepts the following format,

\[
\begin{align*}
\min & \quad \frac{1}{2} z^T H z + (F x_p + f)^T z + x_p^T Y x_p + C x_p + c, \\
\text{s.t.} & \quad A z \leq b + B x_p, \\
& \quad A_e z = b_e + E x_p, \\
& \quad A_{th} x_p \leq b_{th},
\end{align*}
\]  

(4.32)

which contains the vector of optimisation variables \( z \), the vector of parameters \( x_p \), inequality constraints, equality constraints, and constraints on the parameter. Employing the notation introduced beforehand and similarly defined as in Grancharova and Johansen (2012), one can reformulate the optimisation problem and define the matrices of the cost function,

\[
H = H_0, 
\]

(4.36)

\[
F = F_0^T, 
\]

(4.37)

\[
f = -\frac{1}{2} (H_0 + H_0^T) z_0 - F_0^T x_{p,0} + D_0^T, 
\]

(4.38)

\[
Y = \frac{1}{2} V_{xx0}, 
\]

(4.39)

\[
C = -z_0^T F_0^T - \frac{1}{2} x_{p,0}^T (V_{xx0} + V_{xx0}^T) + V_{x0}, 
\]

(4.40)

\[
c = \frac{1}{2} z_0^T H_0 z_0 - D_0 z_0 + x_{p,0}^T F_0 z_0 + \frac{1}{2} x_{p,0}^T V_{xx0} x_{p,0} + V_{x0} x_{p,0} + V_0, 
\]

(4.41)

and the matrices of the constraint functions,

\[
A = G_{i,0}, 
\]

(4.42)

\[
b = G_{i,0} z_0 - E_{i,0} x_{p,0} + T_{i,0}, 
\]

(4.43)

\[
B = E_{i,0}, 
\]

(4.44)

\[
A_e = G_{e,0}, 
\]

(4.45)

\[
b_e = G_{e,0} z_0 - E_{e,0} x_{p,0} + T_{e,0}, 
\]

(4.46)

\[
E = E_{e,0}. 
\]

(4.47)
4.2 Local mp-QP approximation to mp-NLP

The matrices \( A_{ih} \) and \( b_{ih} \) define the partition \( X \) of the parameter space, given by the polyhedral set \( X \triangleq \{ x_p \in X \mid A_{ih} x_p \leq b_{ih} \} \), in which the mp-QP approximation is valid. The matrices can be defined as

\[
A_{ih} = \begin{bmatrix} I_{n_p} & -I_{n_p} \end{bmatrix},
\]

\[
b_{ih} = \begin{bmatrix} \bar{x}_p \\ -\bar{x}_p \end{bmatrix},
\]

with \( \bar{x}_p \) and \( x_p \) for the upper and lower bounds, respectively, of the corresponding parameter space.

4.2.3 Eigen-decomposition of Hessian matrix

The uniqueness of the piecewise affine solution \( \hat{z}(x_p) \) to the mp-QP problem (4.15)-(4.17) can be established introducing some assumptions, as demonstrated in Johansen (2002) and Grancharova and Johansen (2012) in detail. It follows that the Hessian matrix \( H_0 \) must be positive definite and the linear independence constraint qualification (LICQ) and the mentioned regularity assumptions hold true to guarantee uniqueness of the solution.

If the matrix \( H_0 \) is not positive definite at the point \( (z_0, x_{p,0}) \), it is modified in such a way that the matrix becomes positive definite and the mp-QP problem is convex. In order to extract the negative eigenvalues of the matrix \( H_0 \), it is decomposed by means of an eigenvalue/eigenvector decomposition and fulfills the following eigenvalue and eigenvector problem,

\[
H_0 V = V D,
\]

where \( D \) denotes the canonical form of \( H_0 \), a diagonal matrix with the eigenvalues of \( H_0 \) on the main diagonal, and \( V \) denotes the modal matrix, a matrix whose columns are the right eigenvectors of \( H_0 \). The eigenvalue problem determines the non-trivial solutions to the equation \( H_0 v = \lambda v \), with the matrix \( H_0 \in \mathbb{R}^{s \times s} \), the vector \( v \in \mathbb{R}^{s \times 1} \), and the scalar \( \lambda \in \mathbb{R} \). The values of \( \lambda \), that satisfy the equation, are the eigenvalues, and the corresponding values of \( v \), that satisfy the equation, are the right eigenvectors.

For the mp-QP problem, non-positive eigenvalues of the matrix \( H_0 \) are replaced by small positive numbers leading to the modified matrix \( \tilde{H}_0 \). The positive definite Hessian matrix \( \tilde{H}_0 \) can be defined by the following equation,

\[
\tilde{H}_0 = V \tilde{D} V^{-1}.
\]
Therefore, and together with the remaining assumptions, the system of linear equations, defined by the stationarity and complementary slackness conditions of the KKT system for an optimal active set $\mathcal{A}$ at a given $x_p \in X$, has a unique solution.

### 4.3 mp-QA algorithm for explicit approximate solution of mp-NLP

In this section, the suggested approach of the multi-parametric quadratic approximation algorithm for multi-parametric non-linear programming problems is discussed in detail. The algorithm combines performing elements of established algorithms and is composed by blending the approximate mp-NLP algorithm incorporating global optimisation tools, as proposed in Grancharova et al. (2007) and Grancharova and Johansen (2012), together with the mp-QA algorithm, as proposed in Johansen (2002) and Grancharova and Johansen (2012).

#### 4.3.1 Close-to-global solution of non-convex mp-NLP

For a general problem, the cost function $V$ can be non-convex having multiple local minima. Non-convex problems require the use of global optimisation techniques. As a practical approach, an effective initialisation method, as proposed in Grancharova et al. (2007) and Grancharova and Johansen (2012), is incorporated in the algorithm. One possible way to find a close-to-global solution to an NLP program at the point $w_0 \in X_0$ includes the computation of local minima, corresponding to several initial guesses, and selection of the best solution. To propagate the solution, the close-to-global solution at $w_0$ is used as an initial guess at the neighbouring points $w_i \in X_0$, $i = 1, 2, \ldots, N$.

**Chebyshev centre of a polyhedron**

In geometry, the Chebyshev centre $w_0$ of a bounded set $X_0$, having non-empty interior, is the centre of the minimal-radius ball enclosing the entire set, or alternatively and non-equivalently the centre of the largest inscribed ball of $X_0$. In this context, we consider the problem of finding the largest Euclidean ball that lies inside a polyhedron described by linear inequalities $X_0 = \{ x_p \in \mathbb{R} \mid A_{th} x_p \leq b_{th} \}$. The centre $w_0$ of the optimal ball is called the *Chebyshev centre* of the polyhedron. It is also the point deepest inside the polyhedron, in the sense that it is farthest from the boundary, or complement, of $X_0$. More details can be found in Boyd (2004). For the implementation, the embedded tools of the MPT 3.0 toolbox (Herceg et al., 2013) can be employed or the solution of an LP using integrated functions in MATLAB can be considered.
4.3 mp-QA algorithm for explicit approximate solution of mp-NLP

Generation of set of points

Following the procedure, introduced in Grancharova et al. (2007) and Grancharova and Johansen (2012) and extended, for the close-to-global solution of NLPs and the estimation of error bounds, a set of points $W_0 = \{w_0, w_1, \ldots, w_N\} \subset X_0$ is generated. For any hyper-rectangle $X_0 \subseteq X$, the centre point $\mathbf{w}_0$, the set of vertices $\Theta^0 = \{\theta^0_1, \theta^0_2, \ldots, \theta^0_{N_p}\}$, the set of facet centres $\Psi^0 = \{\psi^0_1, \psi^0_2, \ldots, \psi^{0}_{N_v}\}$, and additionally the set of edge centres $\mathcal{E}^0 = \{\varepsilon^0_1, \varepsilon^0_2, \ldots, \varepsilon^0_{N_e}\}$ are defined. For the generation of interior points, further hyper-rectangles $X^j_0 \subset X_0$ with $j = 1, 2, \ldots, N_0$ are considered including a corresponding set of vertices $\Theta^j = \{\theta^j_1, \theta^j_2, \ldots, \theta^j_{N_p}\}$, set of facet centres $\Psi^j = \{\psi^j_1, \psi^j_2, \ldots, \psi^j_{N_v}\}$, and set of edge centres $\mathcal{E}^j = \{\varepsilon^j_1, \varepsilon^j_2, \ldots, \varepsilon^j_{N_e}\}$. It is assumed that $X^j_0 \subset X^j_0 \subset \ldots \subset X^{N_0}_0 \subset X_0$. In summary, the set of all points is defined as $W_0 = \{w_0, w_1, w_2, \ldots, w_N\}$, where $w_i \in \bigcup_{j=0}^{N_0} \Theta^j \cup \bigcup_{j=0}^{N_0} \Psi^j \cup \bigcup_{j=0}^{N_0} \mathcal{E}^j, i = 1, 2, \ldots, N$.

For an $n_p$-dimensional parameter space, the number of vertices of a hyper-rectangle is $N_\theta = 2^{n_p}$, the number of centres of the facets is $N_\psi = n_p (n_p - 1) 2^{(n_p-3)}$, and the number of centres of the edges is $N_\varepsilon = n_p 2^{(n_p-1)}$. The number of all points defined with the generation procedure is $N = 1 + (N_0 + 1) (N_\theta + N_\psi + N_\varepsilon)$ with $N_0$ as the number of interior hyper-rectangles.

Close-to-global solution of NLPs

In order to derive a close-to-global solution of the NLPs at the points $\mathbf{w}_i \in X_0, i = 0, 1, \ldots, N$, the initialisation procedure, as proposed in Grancharova et al. (2007) and Grancharova and Johansen (2012), is applied. Based on the computations of several local minima corresponding to different initial guesses, the close-to-global solution is determined at the centre point $\mathbf{w}_0$ of $X_0$ by choosing the local minimum with smallest cost function value. The close-to-global solution of the NLP at the remaining points $\mathbf{w}_i \in W_0, i = 1, 2, \ldots, N$ is determined by some propagation rules. For each point $\mathbf{w}_i$, the closest point $\tilde{\mathbf{w}} \in W^j$ is determined via minimisation, where $W^j = \{\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{N_j}\}$ is the sub-set of points at which a feasible solution to the corresponding NLP was already found. At the current point $\mathbf{w}_i$, the NLP is then solved with the variables set to $\mathbf{z}^* (\tilde{\mathbf{w}})$ as initial guess for the optimisation. In case a feasible solution was found at the point $\mathbf{w}_i$ to the NLP, it is marked as feasible and added to the set of feasible points $W^j$. Otherwise, the point $\mathbf{w}_i$ is marked as infeasible. This procedure is performed for each point $\mathbf{w}_i \in X_0$.

For the solution of the NLPs at the points $\mathbf{w}_i \in W_0, i = 0, 1, \ldots, N$, an appropriate mathematical programming method has to be selected. As mentioned in Chapter 3, the interior-point solver IPOPT is employed to solve the non-linear programming problems in the implementation of the algorithm.
4.3.2 Computation of an explicit approximate solution

We restrict our attention to a hyper-rectangle $X_0 \subseteq \mathbb{X}$, where $\mathbb{X} \subset \mathbb{R}^{n_x}$ is the hyper-rectangle in the parameter space that has to be explored. On the hyper-rectangle $X_0$, we seek to approximate the solution $z^*(x_p)$ to the mp-NLP problem (4.1)-(4.2).

If a feasible close-to-global solution was found to the NLP problem at the centre point $w_0$, and a feasible close-to-global solution was found at each of the points $w_i$, $i = 1, 2, \ldots, N$, as described in Section 4.3.1, then the local mp-QP approximation (4.15)-(4.17) to the mp-NLP is constructed with the information of the close-to-global solution $z_0 = z^*(w_0)$ and the parameter vector $x_{p,0} = w_0$ at the centre point $w_0$. The local accurate quadratic approximation to the cost function is defined by the matrices (4.20)-(4.25) and the local accurate linear approximation to the constraint functions by the matrices (4.26)-(4.31).

The approximate solution to the mp-NLP is locally defined by the solution of the mp-QP problem (4.15)-(4.17) on the orthogonal partition $X_0$. The solution of the mp-QP problem is leading to a continuous and piecewise affine function defined on associated polyhedral regions, corresponding to a fixed set of active constraints, within the orthogonal partition $X_0$.

For the solution of the mp-QP problem in the implementation of the algorithm, the MPT 3.0 toolbox (Herceg et al., 2013) is employed. For the case of non-positive eigenvalues, the Hessian matrix incorporating the second derivatives with respect to the optimisation variables is manipulated by means of an eigen-decomposition of the matrix, as described in Section 4.2.3.

4.3.3 Estimation of errors bounds

As demonstrated in Johansen (2002) and Grancharova and Johansen (2012), it can be qualitatively shown that the mp-QP approximation of the mp-NLP is locally accurate under some assumptions. For a quantitative measure, an estimation of the error bounds on the cost function and the solution as well as the maximum constraint violation can be derived.

- The solution error bound is defined as

$$
\rho \triangleq \max_{x_p \in X_0} \left| w^T_\rho \left( \hat{z}(x_p) - z^*(x_p) \right) \right|, \tag{4.52}
$$

for an arbitrary hyper-rectangle $X_0 \subset \mathbb{X}$. The weighting vector $w_\rho$ is a vector with positive weights and the elements corresponding to the first sample of the control trajectory will typically be one. As the primary interest for the concept of receding
horizon control is the first sample of the trajectory, the remaining elements will usually be zero.

- The **cost error bound** is defined as
  \[
  \varepsilon \triangleq \max_{x_p \in \mathcal{X}_0} | \hat{V}(\hat{z}(x_p), x_p) - V(z^*(x_p), x_p) |,
  \]
  with \( z^*(x_p) \) denoting a close-to-global solution.

- Additionally, the **maximum constraint violation** can be defined as
  \[
  \delta \triangleq \max(\delta_i, \delta_e),
  \]
  with the maximum constraint violation of the inequality and equality constraint functions, respectively,
  \[
  \delta_i \triangleq \max_{x_p \in \mathcal{X}_0} w^T \delta_i \mathcal{G}_i(\hat{z}(x_p), x_p),
  \]
  \[
  \delta_e \triangleq \max_{x_p \in \mathcal{X}_0} w^T \delta_e | \mathcal{G}_e(\hat{z}(x_p), x_p) |,
  \]
  with
  \[
  \mathcal{G}_i(\hat{z}(x_p), x_p) = \left\{ \forall j \in \{1, \ldots, w_d, i\} \mid G_{i,j}(\hat{z}(x_p), x_p) > 0 \right\},
  \]
  \[
  \mathcal{G}_e(\hat{z}(x_p), x_p) = \left\{ \forall j \in \{1, \ldots, w_d, e\} \mid G_{e,j}(\hat{z}(x_p), x_p) \neq 0 \right\},
  \]
  and the weighting vectors \( w_{\delta_i} \) and \( w_{\delta_e} \) with non-negative weights, typically containing only ones.

Suppose that a solution \( \hat{z}(x_p) \) to the mp-QP problem has been computed for the partition \( \mathcal{X}_0 \). The maximum constraint violation (4.54) can be then computed by solving an NLP program. However, the computation of the cost error bound (4.53) and the solution error bound (4.52) is not easily carried out without introducing additional assumptions or allowing underestimation.

In the case of non-convexity of the cost function or the constraint functions, it would not be sufficient to evaluate the errors only at the vertices of the hyper-rectangle \( \mathcal{X}_0 \). For that reason, additional interior points are included, leading to the set of points \( W_0 = \{ w_0, w_1, \ldots, w_N \} \subset \mathcal{X}_0 \), generated by applying the procedure as described in Section 4.3.1, and representing the vertices, and facet and edge centres of hyper-rectangles contained in the interior of \( \mathcal{X}_0 \). If the convexity assumption does not hold, this seems to be a fairly robust strategy. An additional problem is that the computation of the cost and solution error bound relies on the computation of the exact solution \( z^*(x_p) \).
In order to derive a close-to-global solution, the initialisation procedure as described in Section 4.3.1 is used.

Techniques for the estimation of the cost error bound $\hat{\varepsilon}$ and the solution error bound $\hat{\rho}$ as well as the maximum constraint violation $\hat{\delta}$ take the maximum over a finite number of points $W_0 \subset X_0$. The following procedures can be used to compute an error bound approximation through a maximisation over the set of points $W = \{w_0, w_1, \ldots, w_N\}$.

- An estimation for the solution error bound is computed as
  \[
  \hat{\rho} \doteq \max_{w_i \in \{w_0, w_1, \ldots, w_N\}} \left| w_i^T (\hat{z}(w_i) - z^*(w_i)) \right|. \tag{4.59}
  \]
- An estimation for the cost error bound is computed as
  \[
  \hat{\varepsilon} \doteq \max_{w_i \in \{w_0, w_1, \ldots, w_N\}} \left| \hat{V}(\hat{z}(w_i), w_i) - V(z^*(w_i), w_i) \right|. \tag{4.60}
  \]
- An estimation for the maximum constraint violation is computed as
  \[
  \hat{\delta} \doteq \max(\hat{\delta}_i, \hat{\delta}_e), \tag{4.61}
  \]
  with
  \[
  \hat{\delta}_i \doteq \max_{w_i \in \{w_0, w_1, \ldots, w_N\}} w_i^T G_i \left( \hat{z}(w_i), w_i \right), \tag{4.62}
  \]
  \[
  \hat{\delta}_e \doteq \max_{w_i \in \{w_0, w_1, \ldots, w_N\}} w_i^T |G_e \left( \hat{z}(w_i), w_i \right)|. \tag{4.63}
  \]

The estimations are only valid, if feasible solutions $\hat{z}(w_i)$ and $z^*(w_i)$ are found for all $w_i \in W_0$. Further techniques could include points generated by Monte Carlo methods. It should be emphasised, that these methods can underestimate the bounds in general.

### 4.3.4 Structure of the algorithm

Under some regularity conditions, it can be established that the local mp-QP solution gives an accurate approximation to the solution of the mp-NLP problem when restricted to a sufficiently small region, a subset $X_0 \subset X$. What remains to be determined, is an orthogonal partitioning of the hyper-rectangle $X$ in which the solutions of the local mp-QP problems, associated with each partition, are sufficiently accurate.

The following algorithm is suggested to approximate the mp-NLP solution based on local approximate mp-QP sub-problems and recursive sub-partitioning, guided by the approximation errors incorporating the cost and solution errors as well as the maximum constraint violation.
Assume that the tolerances $\hat{\varepsilon} > 0$ for the approximation error of the cost function, $\hat{\rho} > 0$ for the approximation errors of the solution, and $\hat{\delta} > 0$ for the maximum constraint violation are defined. The weighting vectors $w_\rho$ for the solution error bound, and $w_\delta$, and $w_\delta'$, respectively, for the maximum constraint violation are specified. The volume of a given orthogonal partition, i.e. a hyper-rectangular region, $X_0 \subseteq \mathbb{X} \subset \mathbb{R}^{n_p}$ is denoted with $S_{X_0}$ and computed as $S_{X_0} = \prod_{i=1}^{n_p} \Delta x_{p,i}$, where $\Delta x_{p,i}$ is the size of $X_0$ along the parameter $x_{p,i}$. The minimal allowed volume of the orthogonal partitions in the hyper-rectangle $\mathbb{X}$ is denoted with $S_{\text{min}}$. Let $\mathbb{X} = \{ x_p \in \mathbb{R}^{n_p} \mid A_{th} x_p \leq b_{th} \}$ be a closed polytopic set of parameters, defining the hyper-rectangle of the state-space to be explored. The definition of the matrices $A_{th}$ and $b_{th}$ correspond to an orthogonal partition with the bounds $x_p \leq x_p \leq \bar{x}_p$.

The following algorithm 1 is suggested to compute an explicit approximate solution of non-convex mp-NLP:

**Algorithm 1 Explicit approximate solution of non-convex mp-NLP**

**Input:** Formulation of the mp-NLP problem (4.1)-(4.2); Number of internal hyper-rectangles $N_0$; Approximation tolerances $\hat{\varepsilon}$, $\hat{\rho}$, and $\hat{\delta}$; Weighting vectors $w_\rho$, $w_\delta$, and $w_\delta'$; Minimal allowed volume $S_{\text{min}}$; Definition of hyper-rectangle $\mathbb{X}$ to be explored.

**Output:** Set of orthogonal partitions $\Pi = \{ X_1, X_2, \ldots, X_{N_X} \}$; Solutions to the associated mp-QP problems $\hat{z} = \{ \hat{z}_{X_1}, \hat{z}_{X_2}, \ldots, \hat{z}_{X_{N_X}} \}$, i.e. the piecewise affine solution functions $\hat{z}_{X_i} = \{ \hat{z}_{X_i}^1, \hat{z}_{X_i}^2, \ldots, \hat{z}_{X_i}^{N_{R_i}} \}$ and the set of corresponding polyhedral critical regions $R_{X_i} = \{ R_1, R_2, \ldots, R_{N_{R_i}} \}$ for each partition $X_i$, $i = 1, 2, \ldots, N_X$.

1: Initialise the partition $\Pi$ to the whole hyper-rectangle $\mathbb{X}$, i.e. $\Pi := \mathbb{X}$.
2: Mark the hyper-rectangle $\mathbb{X}$ as unexplored and set $\text{flag} := \text{true}$.
3: while $\text{flag} = \text{true}$ do
4: while $\exists$ unexplored hyper-rectangles in $\Pi$ do
5: Select any unexplored hyper-rectangle $X_0 \in \Pi$.
6: Compute the volume $S_{X_0}$ of $X_0$.
7: Compute the Chebyshev centre $w_0$ of $X_0$ by solving an LP and define a set of points $W_0 = \{ w_0, w_1, \ldots, w_N \}$, as described in Section 4.3.1.
8: Search for a close-to-global solution $z_0 = z^*(x_{p,0})$ to the mp-NLP problem (4.1)-(4.2) with $x_{p,0} = w_0$ at the centre point $w_0$ of $X_0$ by solving an NLP and applying the initialisation procedure, as introduced in Section 4.3.1.
9: if a feasible solution was found to the mp-NLP problem (4.1)-(4.2) at the centre point $w_0$ then
10: Search for a close-to-global solution to the mp-NLP problem (4.1)-(4.2) for $x_p$ fixed to each of the points $w_i$, $i = 1, 2, \ldots, N$ by solving an NLP and applying the initialisation procedure, as introduced in Section 4.3.1.
11: if the mp-NLP problem (4.1)-(4.2) has a feasible solution at all points
Explicit non-linear model predictive control

18: \( w, i = 1, 2, \ldots, N \) then
19: Compute the local mp-QP problem (4.15)-(4.17) at \((z_0, x_{p,0})\).
20: if \( H_0 \) is not positive definite then
21: Modify \( H_0 \) such that it becomes a positive definite matrix \( \tilde{H}_0 \) and
22: the mp-QP problem is convex, as described in Section 4.2.3.
23: end if
24: if a solution \( \hat{z}(x_p) \) was found to the mp-QP problem then
25: Compute an estimation \( \hat{\varepsilon} \) of the cost error bound \( \varepsilon \), an estimation
26: \( \hat{\rho} \) of the solution error bound \( \rho \), and an estimation \( \hat{\delta} \) of the maximum
27: constraint violation \( \delta \) in \( X_0 \), according to (4.59)-(4.63).
28: if \( \hat{\varepsilon} > \bar{\varepsilon}, \hat{\rho} > \bar{\rho}, \) or \( \hat{\delta} > \bar{\delta} \), and \( S_{X_0} > S_{\text{min}} \) then
29: Mark \( X_0 \) to be split with splitting rule \texttt{best} and explored.
30: else
31: Mark \( X_0 \) as feasible and explored.
32: end if
33: else
34: if \( S_{X_0} \leq S_{\text{min}} \) then
35: Mark \( X_0 \) as infeasible and explored.
36: else
37: Mark \( X_0 \) to be split with splitting rule \texttt{longest} and explored.
38: end if
39: end if
40: else
41: if \( S_{X_0} \leq S_{\text{min}} \) then
42: Mark \( X_0 \) as infeasible and explored.
43: else
44: Mark \( X_0 \) to be split with splitting rule \texttt{longest} and explored.
45: end if
46: end if
47: else
48: if \( S_{X_0} \leq S_{\text{min}} \) then
49: Mark \( X_0 \) as infeasible and explored.
50: else
51: Mark \( X_0 \) to be split with splitting rule \texttt{bruteforce} and explored.
52: end if
53: end if
54: end while
55: flag := false
56: if \( \exists \) hyper-rectangles in \( \Pi \) that are marked to be split then
flag := true

while ∃ hyper-rectangles in Π that are marked to be split do

Select any hyper-rectangle $X_0 \in \Pi$ marked to be split.
Split $X_0$ into new hyper-rectangles $X_1, X_2, \ldots, X_{N_s}$ by applying the specified heuristic splitting rule, as defined in Section 4.3.5, and mark the new hyper-rectangles $X_1, X_2, \ldots, X_{N_s}$ unexplored.
Remove $X_0$ from Π and add $X_1, X_2, \ldots, X_{N_s}$ to Π.

end while

end if

end while

4.3.5 Heuristic splitting rules

The sub-partitioning in the suggested algorithm is carried out by applying splitting rules based on heuristic criteria. The purpose is to select one or multiple axis-orthogonal hyperplanes through the Chebyshev centre to split the hyper-rectangle $X_0$, if one or more approximation errors are not fulfilled or in case of infeasibility. The following splitting rules are based on the rules introduced in Grancharova et al. (2007) and Grancharova and Johansen (2012). However, the proposed rules are extended by taking into consideration the maximum possible information on the approximation errors incorporating cost and solution errors as well as maximum constraint violations, based on the combination of new hyper-rectangles.

Determination of the best split of a partition

The following procedure is denoted as splitting rule "best". It is applied to determine the split of a region $X_0$ with volume $S_{X_0} > S_{\min}$ that is big enough to be split and for which the solution $\hat{z}(x_p)$ to the mp-QP problem was found. Suppose that the required accuracy $\tilde{\varepsilon}$ on the cost, $\tilde{\rho}$ on the solution, or $\tilde{\delta}$ on the maximum constraint violation is not achieved. Define penalty terms $\tilde{\varepsilon} \gg 1$ for the cost error, $\tilde{\rho} \gg 1$ for the solution errors, and $\tilde{\delta} \gg 1$ for the constraint violation. The split of the partition $X_0$ is determined in the following way:

1. Let $j = 1$.

2. Split the hyper-rectangle $X_0$ by a hyperplane through its Chebyshev centre and orthogonal to the axis $x_{p,j}$. Denote the two newly created hyper-rectangles with $X_1^j$ and $X_2^j$ and mark the hyper-rectangles as unexplored.
3. Compute the solution $\hat{z}_1^i(\mathbf{x}_p)$ for $\mathbf{x}_p \in X_1^j$ and $\hat{z}_2^i(\mathbf{x}_p)$ for $\mathbf{x}_p \in X_2^j$ to the local mp-QP problem on the hyper-rectangle $X_1^j$ and $X_2^j$, respectively. Mark the hyper-rectangles $X_1^j$ and $X_2^j$ as explored.

4. Compute the arithmetic mean $\bar{\epsilon}_1^j = 1/N_1^j \sum_{i=1}^{N_1^j} \bar{\epsilon}_{1,i}$ of the cost errors at all points $\mathbf{w}_{1,i}^j \in W_1^j$ in $X_1^j$, and $\bar{\epsilon}_2^j = 1/N_2^j \sum_{i=1}^{N_2^j} \bar{\epsilon}_{2,i}$ of the cost errors at all points $\mathbf{w}_{2,i}^j \in W_2^j$ in $X_2^j$, respectively. In the implementation, the number of points for all hyper-rectangles is assumed the same, $N_1^j = N_2^j = N$. Let the average cost error over the newly created hyper-rectangles be $\bar{\epsilon}^j = 1/2 (\bar{\epsilon}_1^j + \bar{\epsilon}_2^j)$.

5. Compute the arithmetic mean $\bar{\rho}_1^j = 1/N_1^j \sum_{i=1}^{N_1^j} \bar{\rho}_{1,i}$ and $\bar{\rho}_2^j = 1/N_2^j \sum_{i=1}^{N_2^j} \bar{\rho}_{2,i}$, respectively, of the solution errors at all points $\mathbf{w}_{1,i}^j \in W_1^j$ in $X_1^j$ and $\mathbf{w}_{2,i}^j \in W_2^j$ in $X_2^j$. Let the average solution errors over the newly created hyper-rectangles be $\bar{\rho}^j = 1/2 (\bar{\rho}_1^j + \bar{\rho}_2^j)$.

6. Compute the arithmetic mean $\bar{\delta}_1^j = 1/N_1^j \sum_{i=1}^{N_1^j} \bar{\delta}_{1,i}$ and $\bar{\delta}_2^j = 1/N_2^j \sum_{i=1}^{N_2^j} \bar{\delta}_{2,i}$, respectively, of the constraint violations at all points $\mathbf{w}_{1,i}^j \in W_1^j$ in $X_1^j$ and $\mathbf{w}_{2,i}^j \in W_2^j$ in $X_2^j$. Let the average constraint violation over the newly created hyper-rectangles be $\bar{\delta}^j = 1/2 (\bar{\delta}_1^j + \bar{\delta}_2^j)$.

7. If no feasible solution $\hat{z}_1^j(\mathbf{x}_p)$ or $\hat{z}_2^j(\mathbf{x}_p)$ was found to the local mp-QP problem on the hyper-rectangle $X_1^j$ and $X_2^j$, respectively, set the average cost error equal to the penalty term $\bar{\epsilon}^j = \bar{\epsilon}$, the average solution errors equal to the penalty terms $\bar{\rho}^j = \bar{\rho}$, and the average constraint violation equal to the penalty term $\bar{\delta}^j = \bar{\delta}$.

8. Let the weighted sum of average errors be $\bar{\epsilon}^j = \bar{\epsilon}^j / \bar{\epsilon} + 1/m \sum_{k=1}^{m} \bar{\rho}_k / \bar{\rho} + \bar{\delta}^j / \bar{\delta}$, where the individual average error terms are scaled to their corresponding approximation tolerances.

9. Let $j = j + 1$. If $j \leq n_p$, go to step 2.

10. Determine the combination of hyper-rectangles $X_1^j$ and $X_2^j$ for which the weighted sum of average errors $\bar{\epsilon}^j$ is minimal. Keep the two hyper-rectangles $X_1^j$ and $X_2^j$ together with the associated information and discard all the other hyper-rectangles.

The splitting rule can be summarised as follows: The best combination of hyper-rectangles, after an axis-orthogonal split by a hyperplane through the Chebyshev centre, minimising a weighted sum of the considered average errors, is kept and the remaining combinations of new hyper-rectangles are discarded. The suggested splitting rule incorporating heuristics is expected to give a significant reduction in the approximation errors in both newly created hyper-rectangles after splitting. The effectiveness has been observed in a number of examples.
4.4 Post-processing of the explicit approximate solution of mp-NLP

Split of partition in case of infeasibility

The following splitting rules are applied when no feasible solution was found to the NLP problem (4.1)-(4.2) at some of the points $\mathbf{w}_i \in W_0 = \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_N\}$ or when no solution $\hat{z}(x_p)$ was found to the local mp-QP problem (4.15)-(4.17) at $(\mathbf{z}_0, x_{p,0})$:

- The splitting rule denoted as "bruteforce" is applied if no feasible solution was found to the NLP problem (4.1)-(4.2) at the centre point $\mathbf{w}_0$ of $X_0$ and the volume $S_{X_0}$ of the hyper-rectangle is big enough to be split $S_{X_0} > S_{\text{min}}$.

  The hyper-rectangle $X_0$ is then split by hyperplanes through its Chebyshev centre and orthogonal to every parameter axis $x_{p,i}$, $i = 1, \ldots, n_p$.

- The splitting rule denoted as "longest" is applied in two different cases:
  
  a) If no solution $\hat{z}(x_p)$ was found to the local mp-QP problem (4.15)-(4.17) at $(\mathbf{z}_0, x_{p,0})$ and the volume $S_{X_0}$ of the hyper-rectangle is big enough to be split $S_{X_0} > S_{\text{min}}$.

  b) If a feasible solution was not found at all points $\mathbf{w}_i$, $i = 1, 2, \ldots, N$ and the volume $S_{X_0}$ of the hyper-rectangle is big enough to be split $S_{X_0} > S_{\text{min}}$.

  The hyper-rectangle $X_0$ is then split by a hyperplane through its Chebyshev centre and orthogonal to the parameter axis $x_{p,i}$ along which the size $\Delta x_{p,i}$ of the hyper-rectangle $X_0$ has its maximum.

These splitting rules are presented in order to handle infeasibility. However, it is worthwhile to mention that the inner approximation of infeasible regions with orthogonal partitions leads in general to a very significant number of partitions. Therefore, it is strongly recommended to use techniques to avoid infeasibility of the underlying mathematical program. One effective method, as explained in Section 3.3.4, is the use of slack variables allowing soft constraints. For all presented applications in Chapters 5-7, soft constraints are implemented to successfully avoid infeasibility related issues for the partitioning algorithm.

4.4 Post-processing of the explicit approximate solution of mp-NLP

In this section, an approach for the post-processing of the output of Algorithm 1 is introduced including complexity reduction and appropriate data handling methods for the sub-optimal solution.
Explicit non-linear model predictive control

**mp-QA algorithm output**  The application of the suggested mp-QA algorithm 1 to the mp-NLP problem (4.1)-(4.2) leads to a piecewise affine function representation defined on polyhedral regions. More specifically, the output from the algorithm is composed of the solution to the local mp-QP sub-problems defined on orthogonal partitions and includes the following information:

- Set of orthogonal partitions $\Pi = \{X_1, X_2, \ldots, X_{N_X}\}$.
- Solutions to the associated local mp-QP problems $\hat{\mathbf{z}} = \{\hat{\mathbf{z}}_{X_1}, \hat{\mathbf{z}}_{X_2}, \ldots, \hat{\mathbf{z}}_{X_{N_X}}\}$, i.e. piecewise affine solution functions $\hat{\mathbf{z}}_{X_i} = \{\hat{\mathbf{z}}_{X_i}^1, \hat{\mathbf{z}}_{X_i}^2, \ldots, \hat{\mathbf{z}}_{X_i}^{N_{R,i}}\}$.
- Set of corresponding polyhedral critical regions $R_{X_i} = \{R_1, R_2, \ldots, R_{N_{R,i}}\}$ for each partition $X_i, i = 1, 2, \ldots, N_X$.

**Post-processing objectives**  The performance of the explicit NMPC considerably depends on efficient methods for evaluating the piecewise affine functions, since the complexity of the representations may be large even for relatively small systems. The post-processing algorithm should, therefore, include the assessment of a possible reduction of the necessary memory-requirements without losing essential information for the online evaluation. Since in a receding horizon approach the solution associated to the first sample only is applied, also a modification of the feedback law $k(\mathbf{x}_p)$, a priori including all decision variables, should be incorporated. For the online evaluation, the identification of the polyhedral critical region containing a given parameter vector and the associated feedback law leads to a point-location problem. With a suitable method tackling this problem, what further needs to determined are the methods of storing the piecewise affine receding horizon feedback law $\hat{k}(\mathbf{x}_p)$ with its associated polyhedral regions for an efficient online evaluation in a real-time environment. The suitable post-processing of the algorithm output can, therefore, significantly reduce the complexity of the necessary data and increase the performance of an explicit NMPC.

**Post-processing implementation**  With the defined objectives for the post-processing of the approximate solution, generated by the suggested mp-NLP algorithm, the post-processing algorithm 4 includes both complexity reduction and efficient data handling. The following tasks and concepts are combined in the algorithm, designed to provide a robust implementation for general practical applications:

- Create a binary search algorithm and data structure to identify the corresponding orthogonal partition for a given parameter vector.
- Modify the piecewise affine feedback law $k(\mathbf{x}_p)$ in such a way that it computes the sub-optimal solution associated to the first sample only, i.e. the receding horizon feedback law $\hat{k}(\mathbf{x}_p)$.
4.4 Post-processing of the explicit approximate solution of mp-NLP

- Incorporate available tools to reduce the complexity of the piecewise affine function representations defined on polyhedral regions. The embedded methods in the algorithm include a clipping-based complexity reduction approach, and a disjoint optimal and a sub-optimal merging procedure.

- Employ an efficient search algorithm, combined with appropriate data handling, to identify the polyhedral critical region for a given parameter within a specified orthogonal partition. The search algorithm for this point location problem makes use of two different approaches including sequential and binary search.

- Generate robust and real-time executable code including checks for evaluation feasibility and safety features.

The complexity reduction incorporates the clipping-based complexity reduction, as proposed in Kvasnica and Fikar (2012), and the merging procedures based on the sub-optimal and the disjoint optimal merging methods, proposed in Geyer et al. (2008), employing the implemented tools of the MPT 3.0 toolbox (Herceg et al., 2013). Similarly, for the data processing and effective evaluation of the function representation, the tools of the MPT 3.0 toolbox (Herceg et al., 2013) are employed, providing an implementation of a memory-optimised binary search tree generation, based on Tøndel et al. (2003b). These carefully developed tools are easily accessible and facilitate a straightforward implementation. However, for a practical application, the following aspects should be considered. The application of the procedures for complexity reduction can lead to non-fully covering, overlapping, or non-convex unions of sets. Likewise, the generation of a binary search tree can fail, e.g. due to bad numerical conditioning of the underlying solution of the local mp-QP problem. For this reason, the post-processing algorithm 4 is designed to compute the best possible option, considering potential failures due to numerical issues. The algorithm provides a robust implementation being able to recover from these failures in the complexity reduction procedures and binary search tree generation. The suggested approach combines well performing state-of-the-art elements proposed in the literature to a robust and straightforward implementable algorithm for the post-processing of piecewise affine functions defined on polyhedral regions derived from the suggested mp-QA algorithm.

**Post-processing output** The output of the post-processing algorithm 4 is based on a topology of the executable files with separate layers. With a given parameter vector, the following three layers perform the real-time capable executions to derive a sub-optimal control action to be applied to the plant:

- Top layer for the identification of the corresponding orthogonal partition and associated local mp-QP sub-problem.
• Bottom layer for the identification of the polyhedral critical region and evaluation of the corresponding affine receding horizon feedback law $\tilde{k}(x_p)$.

• Check of the feasibility of the evaluation to ensure that an orthogonal partition, polyhedral critical region, and corresponding receding horizon feedback law $\tilde{k}(x_p)$ was found. In the case of evaluation infeasibility, the last control action applied to the plant is provided as reasonable alternative control action.

4.4.1 Complexity reduction

The explicit NMPC approach leads to the sub-optimal control input as an explicit piecewise affine function of the parameter vector. However, the function is typically complex even for relatively small systems. The complexity of the real-time implementation of the function representation is determined by the amount of memory required to describe the function and by the amount of CPU time needed to evaluate the expressions for a particular value of $x_p$. In the simplest form, the explicit solution can be stored in a table with rows, representing the individual regions. The size of the table increases typically exponentially on the number of constraints. As the memory requirements and similarly the CPU time grow generally speaking proportionally with the table size, the aim is to find methods to keep the complexity of the receding horizon feedback law $\tilde{k}(x_p)$ as low as possible. For a successful real-time implementation, it is, therefore, of imminent importance to reduce the number of polyhedral critical regions and find a replacement function $\hat{k}(x_p)$ by preserving the characteristics of $\tilde{k}(x_p)$ and its influence on the closed-loop performance, stability, and constraint satisfaction at the same time.

As discussed in Chapter 2, the issue of complexity reduction is addressed from various perspectives in the literature. The most simple methods consider the reduction of the number of constraints and the number of optimisation variables by employing move blocking of the inputs or reduction of the prediction horizon. Similarly, adjusting the weights in the cost function and the formulation of soft constraints can help to reduce the complexity. Also, the limitation of the exploration space can be considered. In this section, however, we will concentrate on two approaches for the post-processing of the sub-optimal solution: A clipping-based complexity reduction and merging of regions.

For a successful clipping-based complexity reduction, the explicit receding horizon feedback $\hat{k}(x_p)$, i.e. the reduced feedback law, should contain at least one region over which the function attains a saturated value. For a vector-valued function, the regions are considered saturated if all its elements are saturated jointly at maximum or at minimum. The concept is based on eliminating a significant portion of the regions of the piecewise affine function by discarding all saturated regions. Subsequently, the neighbouring unsaturated regions, with their associated functions, are expanded covering the discarded
areas. This replacement function \( \hat{k}(x_p) \) is passed through a so-called clipping function \( \phi(\hat{k}(x_p)) \) during evaluation such that the equivalence \( \hat{k}(x_p) = \phi(\hat{k}(x_p)) \) is established for all feasible parameter vectors \( x_p \) and the individual values do not exceed their respective saturation limits.

Likewise, for a potential merging of regions, the explicit receding horizon feedback law \( \hat{k}(x_p) \) is considered, extracting the sub-optimal inputs \( \hat{u}(t_k) \) at the first sample only. Due to the reduction of the feedback law, it is likely that some of the polyhedral regions contain the same expressions defining the function. Subsequently, neighbouring regions with the same affine feedback law can be merged in case their union builds a convex set. The result is a new piecewise affine function \( \tilde{k}(x_p) \), which is an equivalent replacement of \( \hat{k}(x_p) \) in the sense that \( \tilde{k}(x_p) = \hat{k}(x_p) \), for all feasible parameter vectors \( x_p \). However, the employed disjoint optimal merging method is computationally demanding and, thus, limited to small-scale problems only, despite possible modifications incorporating divide and conquer strategies that can handle larger problems. Alternatively, the application of the sub-optimal merging method can be considered.

Both approaches reduce the complexity of the algorithm output provided the feedback law is modified and contains those elements only that are associated with the first sample. In general, the potential for complexity reduction depends on the system and the dimension of the input vector. The clipping-based complexity reduction removes those regions in which all control inputs are saturated. The likelihood that this would happen naturally decreases with an increasing dimension of the input vector. Similarly, it is less likely that all inputs have the same expressions for the feedback law, facilitating the merging of both saturated and unsaturated regions, with an increasing number of inputs.

In the suggested post-processing algorithm 4, clipping and region merging are combined and applied in sequence. In Huba et al. (2011), several supporting aspects are mentioned to apply clipping first and region merging afterwards. Especially, the disjoint optimal merging method cannot handle problems with a number of regions having the same control law larger than about 50. Thus, for a majority of practical problems, the optimal method does not converge. The modified optimal region merging method based on a divide and conquer strategy may be used with relative success. Additionally, experience shows that the majority of polyhedral regions with the same control law are saturated regions. The different methods for merging of regions have to join these into larger convex polytopes, whereas clipping needs not to take care of convexity. The application of clipping keeps resulting regions convex and removes a large fraction of situations that would lead to non-convex unions of polytopes with the disjoint optimal merging method. Finally, clipping cannot handle regions with unconstrained control laws. In a very limited
number of cases, it could happen that the application of region merging methods followed by clipping could produce a smaller number of regions.

Embedded in the post-processing algorithm 4, the presented procedures allow to significantly decrease the memory and runtime requirements of the explicit NMPC.

**Clipping-based complexity reduction**

For the approximate mp-NLP algorithm, the explicit receding horizon feedback \( \tilde{k}(x_p) \) is a continuous piecewise affine function defined over \( N_{R_i} \) polyhedral critical regions on orthogonal partitions \( X_i \) derived from local mp-QP problems. Using the clipping-based complexity reduction approach, suggested in Kvasnica and Fikar (2012), Huba et al. (2011), and Kvasnica and Fikar (2010), the primary idea is to replace \( \tilde{k}(x_p) \) by a simpler function \( \tilde{\gamma}(x_p) \), which requires less memory for its description and is faster to evaluate.

The approach is based on the premise that the receding horizon controller operates at the limits of the admissible control freedom for some parameters vectors \( x_p \). Regions, where the control action attains a saturated value, are removed and replaced by extensions of unsaturated regions. Doing so, the procedure constructs a replacement function \( \tilde{\gamma}(x_p) \) for the optimal control law \( \tilde{k}(x_p) \) which is guaranteed to contain less regions than the original one using basic polyhedral operations.

Given a saturated continuous piecewise affine function \( \tilde{k}(x_p) \), defined over the partition \( X_i \), for a suitable augmentation \( \hat{k}(x_p) \) the following properties hold true: \( \hat{k}(x_p) \) is defined over \( X_i \), \( \hat{k}(x_p) = \tilde{k}(x_p) \) for all \( x_p \in R_{I_{unsat}} \), \( \hat{k}(x_p) \geq \bar{k} \) for all \( x_p \in R_{I_{max}} \), and \( \hat{k}(x_p) \leq \bar{k} \) for all \( x_p \in R_{I_{min}} \) with \( R_I \) as the corresponding sub-set of regions and \( \bar{k} \) and \( k \) as the maximum and minimum saturation limits. With a suitable augmentation, the complexity in terms of number of regions can be reduced usually leading to \( \bar{N}_{R,i} << N_{R,i} \). However, the augmentation cannot be readily applied, since generally \( \hat{k}(x_p) \neq \tilde{k}(x_p) \) for some \( x_p \in X_i \).

For that reason, the replacement function \( \tilde{k}(x_p) \) is then passed through a so-called clipping function \( \phi(\cdot) \) such that the equivalence \( \hat{k}(x_p) = \phi(\tilde{k}(x_p)) \) is established for all \( x_p \in X_i \). Consider a saturated continuous piecewise affine function \( \hat{k}(x_p) \) and its suitable augmentation \( \tilde{k}(x_p) \). Then the equivalence is preserved by employing the clipping function,

\[
\phi(\tilde{k}(x_p)) := \min \left( \max \left( \hat{k}(x_p), \bar{k} \right), \bar{k} \right),
\]

which leads to the following three different cases,

\[
\phi(\tilde{k}(x_p)) = \begin{cases} 
\bar{k} & \text{if } \hat{k}(x_p) \geq \bar{k}, \quad (\forall x_p \in R_{I_{max}}), \\
k & \text{if } \hat{k}(x_p) \leq \bar{k}, \quad (\forall x_p \in R_{I_{min}}), \\
\tilde{k}(x_p) & \text{otherwise}, \quad (\forall x_p \in R_{I_{unsat}}).
\end{cases}
\]
This compact encoding is embedded in the generated layer for the check of evaluation feasibility. Having established the equivalence $\tilde{k}(x_p) = \phi(\hat{k}(x_p))$, the replacement $\hat{k}(x_p)$ is, therefore, a performance-lossless replacement of $\tilde{k}(x_p)$.

It can be shown that $\hat{k}(x_p)$ is a piecewise affine function defined over $\tilde{N}_{R,i}$ polyhedral regions such that $N_{unsat,i} \leq \tilde{N}_{R,i} \leq N_{R,i}$, typically with $\tilde{N}_{R,i} = N_{unsat,i}$ and $\tilde{N}_{R,i} << N_{R,i}$, where $N_{unsat}$ is the number of unsaturated regions of the original function $\tilde{k}(x_p)$. Therefore, replacing the explicit receding horizon feedback $\tilde{k}(x_p)$ by $\phi(\hat{k}(x_p))$ usually leads to a significant reduction of the memory consumption and to an increased on-line evaluation speed. Evaluation speed can be further increased by employing advanced region traversal strategies, as described in Section 4.4.2.

The main benefit, compared to the optimal region merging method, is that the construction of the replacement function scales significantly better with the growing complexity of the original function $\tilde{k}(x_p)$.

**Disjoint optimal merging method**

This approach for merging regions, sharing the same receding horizon control law $\tilde{k}(x_p)$, as proposed in Geyer et al. (2008), is based on the cells and the markings of the hyperplane arrangement. For a given piecewise affine function representation on polyhedral regions, the aim of the algorithm is to derive an equivalent polyhedral piecewise representation that is minimal and, thus, optimal in the number of polyhedra. The cells are the polyhedra generated by the hyperplane arrangement and can uniquely be identified by their markings, i.e. their relative positions with respect to the hyperplanes.

The disjoint optimal complexity reduction (DOCR) algorithm, based on executing branch and bound techniques on the markings of the cells, yields a set of disjoint (non-overlapping) merged polyhedra, being unions of the original polyhedra. The approach leads to an optimal merging problem, which can be also considered as a specific optimal set partitioning problem. The concept of markings in a hyperplane arrangement allows evaluating the convexity of polyhedra using their associated set of markings without solving LPs. Therefore, the proposed algorithm is computationally efficient in the sense that the convexity recognition is performed only by comparing the markings. Additional heuristics on the branching strategy can also be used to reduce the computation time.

However, the main bottleneck is the computation of the cells in the hyperplane arrangement. The main problem here is that the number of polyhedra significantly increases when deriving the hyperplane arrangement. Furthermore, the computation of the full hyperplane arrangement involves solving LPs. Such an algorithm is, therefore, computationally only tractable for problems with limited complexity. However, employing a divide and conquer strategy, the original large problem can be sequentially divided into
Explicit non-linear model predictive control

smaller sub-problems, solved efficiently by the DOCR algorithm, and the solutions of the sub-problems are recombined.

The non-disjoint optimal complexity reduction (NOCR), proposed in Geyer et al. (2008), based on logical minimisation, is an optimal set covering problem and leads, therefore, in general to non-disjoint (overlapping) polyhedra. Compared to the disjoint OCR approach, it is generally faster and also scales better as the problem size increases. However, because of possible overlaps of the merged regions, the generation of a binary search tree, as implemented in the MPT 3.0 toolbox (Herceg et al., 2013) and employed in the post-processing algorithm 4, is not well defined. As the binary search is the desired search algorithm for the point-location, only the DOCR algorithm is further employed.

As the disjoint optimal merging method has been implemented in MATLAB and is included in the MPT 3.0 toolbox (Herceg et al., 2013), it can easily be embedded in the post-processing algorithm 4.

Sub-optimal merging method

Another approach for the merging of regions is proposed in Geyer et al. (2008). It represents a sub-optimal method as optimality is not pursued and no additional facets are introduced. The so-called greedy merging algorithm cycles through the regions and consecutively determines if a pair of neighbouring polyhedra, sharing the same receding horizon feedback law, forms a convex union. To speed up computations, a sparse matrix is built in a first step indicating whether two polyhedra are neighbouring using algorithms defined in Bemporad et al. (2001). In a second step, the algorithm determines based on this list if a pair of neighbouring polyhedra forms a convex union by solving LPs. In case of convexity, the pair is replaced by its union and the matrix is updated accordingly. This procedure is executed sequentially until no pair is left for merging. As mentioned in Geyer et al. (2008), trying all polyhedral combinations using standard techniques based on linear programming can be prohibitive if the number of polyhedra with the same function is large since the number of possible combinations explodes. For this reason, the sub-optimal method is only performed in case the optimal merging method fails.

4.4.2 Data handling and point location

Problem definition When implementing a piecewise affine state-feedback control law, one faces the problem of finding the corresponding control law for a given parameter $x_p$ in an efficient way. This may seem a trivial task at first sight, but when the complexity of the function representation is high, a straightforward evaluation is computationally expensive.
The evaluation of the receding horizon control law \( \hat{k}(x_p) \) for a given value of \( x_p \) is typically a three stage process: First, the index \( r \) of the region which contains \( x_p \) is identified. This leads to a point-location problem, a fundamental topic of computational geometry. Once the index \( r \) is found, the \( r \)-th elements \( \hat{F}_r \) and \( \hat{g}_r \) are extracted from the memory and \( \hat{k}(x_p) = \hat{F}_r x_p + \hat{g}_r \) is computed. Finally, the function value is clipped by passing it through the clipping function \( \hat{k}(x_p) = \phi(\hat{k}(x_p)) \). However, without applying the clipping-based complexity reduction, only the first two stages are necessary.

**Search data structures** Efficient methods for the function evaluation are relying on appropriate search data structures for point-location. An efficient data structure for the representation of piecewise affine functions allows the retrieval of a specific region from the set of regions in an effort to minimise the required time. Concurrently, the storage needed by this data structure is supposed to be kept small.

The most general and least efficient search structure is merely an unordered list of all items. Locating the desired item in such an unsorted list requires a query time \( \mathcal{O}(n) \) using \( \mathcal{O}(n) \) space, with \( n \) as the number of items. In the context of the piecewise affine function representation, the most direct way of doing this is to perform a sequential search through the polyhedral regions of the partition. It is carried out by cycling through the critical regions \( \hat{R}_j \) and checking if the corresponding inequalities hold true. In the worst case, every region and every hyperplane is checked in the partition. Even though these operations only involve matrix multiplications, this approach becomes prohibitive for a large number of polyhedra and small sampling times, respectively. The aim is to find a method to determine a region \( \hat{R}_r \) to which a given parameter \( x_p \) belongs to by evaluating as few hyperplanes as possible.

Useful alternative search data structures allow faster retrieval, such as a balanced binary search tree with query time \( \mathcal{O}(\log n) \) and required space \( \mathcal{O}(n) \). The binary search tree allows, therefore, to efficiently determine the polyhedron and associated control law for a given state. It is a data structure, where the state-feedback control law is found by traversing from the root node to the leaves. Branching at the nodes, that are associated with a hyperplane, is done according to the half-space induced by the hyperplane, where the state lies in. Each leaf of the binary search tree is associated with a polyhedron and the corresponding control law. A balanced search tree minimises the worst-case computational burden, which is logarithmic in the maximal number of polyhedra to be evaluated. However, the cost of building alternative search data structures is at least proportional to \( n \). Indeed, it is, for instance, a non-trivial task to generate a search tree of minimal depth. Therefore, they only pay off if several queries are to be performed on the same database. Since this is always the case for an explicit NMPC, one has to find a reasonable trade-off between evaluation times and memory requirements by choosing an appropriate data structure.
In the suggested post-processing algorithm 4, an attempt is made to generate a binary search tree at first for each individual orthogonal partition \( X_i \subseteq X \). If the generation of the search tree fails, due to the bad numerical condition of the underlying function representation, a sequential search algorithm is provided as an alternative.

**Sequential search**

Sequential search or linear search is the simplest method for finding a target value within a list. It sequentially checks each element of the list for the target value until a match is found or until the end of the list is reached, terminating the search unsuccessfully. The sequential search runs in at the worst linear time and makes at most \( n \) comparisons, where \( n \) is the length of the list. The performance of linear search could improve if values that are more likely to be searched than others are placed at the beginning of the list. However, this fact is not easy to exploit for the application of explicit NMPC. Although in theory, other search algorithms may be faster than linear search, it might be indeed a practicable solution for small-sized arrays.

Sequential search is usually simple to implement, as shown in Algorithm 2. Consider the set of polyhedral critical regions \( \hat{R}_{X_i} = \{ \hat{R}_1, \hat{R}_2, \ldots, \hat{R}_{\hat{N}_{R,i}} \} \) that form an orthogonal partition \( X_i \subseteq X \) of the hyper-rectangle \( X \subset \mathbb{R}^n \) of the state-space to be explored. Let all hyperplanes defining the polyhedra in the partition be denoted \( \hat{a}_k \hat{x}_p = \hat{b}_k \) for \( k = 1, 2, \ldots, \hat{N}_{H,i} \). With the definition \( \hat{h}_k(x_p) = \hat{a}_k \hat{x}_p - \hat{b}_k \), the polyhedral critical region \( \hat{R}_j \) can be represented through its index set \( I_j \) such that

\[
\hat{R}_j = \{ x_p \in X_i \mid \hat{h}_k(x_p) \leq 0 \text{ for all } k \in I_j \}. \tag{4.66}
\]

By summarising \( \hat{h}_k(x_p) \) for all \( k \in I_j \) into \( \hat{H}_j(x_p) \) for each region \( \hat{R}_j \), where \( j = 1, 2, \ldots, \hat{N}_{R,i} \), the algorithm for sequential search can then be defined as follows:

**Algorithm 2** Sequential search algorithm

**Input:** Parameter vector \( x_p \); Index \( x \) of orthogonal partition; Number of polyhedral regions \( \hat{N}_{R,x} \); Matrices \( \hat{H}_j \) defining the set of polyhedral critical regions \( \hat{R}_{X,x} \); Matrices \( \hat{F}_j \) and \( \hat{g}_j \) defining the receding horizon feedback law \( \hat{k}(x_p) \) for \( x_p \in \hat{R}_j \).

**Output:** Index \( r \) of polyhedral critical region; Value \( \hat{k} \) of receding horizon feedback law.

1: for \( j \in \{1, 2, \ldots, \hat{N}_{R,x}\} \) do
2:  if \( \hat{H}_j(x_p) \leq 0 \) then
3:    \( r = j \)
4:  \( \hat{k} = \hat{F}_r \hat{x}_p + \hat{g}_r \)
5: end if
6: end for
The index $x$ of the orthogonal partition is calculated by the top layer by traversing a binary search tree for a given parameter vector $x_p$. The generation of the sequential search algorithm is implemented in the MPT 3.0 toolbox (Herceg et al., 2013).

**Binary search**

Binary search is a search algorithm that finds the position of a target value within a sorted list. It compares the target value to the middle element of the list. In case, the target value is less or greater than the middle element, the search continues in the lower or upper half of the array, respectively, eliminating the other half from the search. Otherwise, the search terminates and returns the middle element. The binary search runs in at the worst logarithmic time, making $O(\log n)$ comparisons.

An appropriate variation of binary search is the binary search tree. The internal nodes of this rooted tree store a key and each has two distinguished sub-trees commonly denoted left and right. In case the target value is greater than the key the right sub-tree is selected, if it is less than or equal the key the left sub-tree is selected. The leaves of the tree, also denoted as final nodes, contain no key but the relevant information. For a search, the tree is traversed from the root to leaf, making a comparison to the keys stored in the nodes. Based on that, the search is continued in the left or right sub-trees, discarding on average about half of the tree. Usually, a binary tree will most likely be imperfectly balanced, resulting in a worse performance than a binary search.

For the evaluation of a piecewise affine function representation defined on polyhedral regions, the aim is to minimise the number of linear inequalities to evaluate in order to determine which critical region $x_p$ belongs to. An efficient way to exploit the polyhedral structure of the regions and the convexity of the polyhedral sets is to build a binary search tree where at each level of the tree only one linear inequality is evaluated. The idea is to construct the binary search tree such that for a given $x_p \in X_i$, at each node the affine function $\hat{h}_i(x_p)$ is evaluated and the sign is tested. Based on that sign, one can determine on which side of the hyperplane the polyhedral region $\hat{R}_r$ is located and select the left or right sub-tree. Traversing the tree from the root to a leaf node, one will end up with a leaf node giving a unique affine control law defined by $\hat{F}_r$ and $\hat{g}_r$.

The complexity of the binary search tree depends on the depth and the number of nodes in the tree. In the best case, we would be able to select a hyperplane at each node traversing the tree having half of the remaining regions on each side. This would lead to a depth of the tree $O(\log n)$ and each hyperplane would be stored once in the tree. However, this best-case estimate will typically not be possible for a general problem. It is desirable and also the main challenge to design a tree of minimum depth such that the number of hyperplanes to be evaluated to determine the solution is minimised. The construction of a binary search tree is computationally demanding, which can become
prohibitive if the number of regions is large. The information on the solutions and the hyperplanes can be stored in a table. Each leaf node in the tree contains then one pointer to the table of control laws, while each non-leaf node contains one pointer to the table of hyperplanes and two additional pointers to its child nodes.

An algorithm for binary search using a binary search tree can be defined as follows:

**Algorithm 3 Binary search algorithm**

**Input:** Parameter vector $x_p$; Index $x$ of orthogonal partition; Binary search tree data structure including all nodes of the search tree, affine functions $\tilde{h}_k(x_p)$ defining the hyperplanes, and matrices $\tilde{F}_j$ and $\tilde{g}_j$ defining the receding horizon feedback law $\tilde{k}(x_p)$ for all $x_p \in \tilde{R}_j$.

**Output:** Index $r$ of polyhedral critical region; Value $\tilde{k}$ of receding horizon feedback law.

1. Let the current node $N_k$ be the root node of the tree.
2. while $N_k$ is not a leaf node do
3. Evaluate the hyperplane $\tilde{h}_k(x_p)$ corresponding to $N_k$.
4. Let $N_k$ be the child node according to the sign of $\tilde{h}_k$.
5. end while
6. Evaluate the control feedback $\tilde{k} = \tilde{F}_r x_p + \tilde{g}_r$ corresponding to $N_k$.

Equivalently to the sequential search algorithm, the index $x$ of the orthogonal partition is calculated by the top layer also by traversing a binary search tree for the given parameter vector $x_p$. The generation of the binary search tree, based on Tøndel et al. (2003b), is implemented in the MPT 3.0 toolbox (Herceg et al., 2013). This allows a straightforward incorporation of a memory-optimised binary search tree employing an efficient evaluation with computation times logarithmic and storage requirements polynomial in the number of polyhedral critical regions.

### 4.4.3 Post-processing algorithm

In this section, the suggested post-processing algorithm 4, incorporating complexity reduction and efficient data handling methods, is explained in detail. The post-processing of the mp-NLP algorithm output consists mainly of three parts:

1. The complexity reduction of the sub-optimal solution employing a clipping-based complexity reduction approach and merging procedures based on a disjoint optimal and a sub-optimal approach.

2. The generation of an efficient search data structure based on a binary search employing a binary search tree or alternatively based on sequential search.

3. The generation of real-time executable code for the practical application of the explicit NMPC.
4.4 Post-processing of the explicit approximate solution of mp-NLP

For straightforward implementation of the post-processing algorithm, the following available tools implemented in the MPT 3.0 toolbox (Herceg et al., 2013) are incorporated in the implementation:

- Clipping-based complexity reduction "clipping", based on Kvasnica and Fikar (2012), Kvasnica and Fikar (2010), and Huba et al. (2011), and implemented in an extending package of the MPT 3.0 toolbox (Herceg et al., 2013).

- Disjoint optimal complexity reduction "optimal" with region merging employing branch and bound techniques, based on Geyer et al. (2008).

- Sub-optimal complexity reduction "greedy" with region merging based on Geyer et al. (2008).

- Generation of a memory-optimised binary search tree data structure "bst", based on Tøndel et al. (2003b).

- Generation of a sequential search data structure "sq".

Both, the procedures for complexity reduction of the sub-optimal solution and the generation of a binary search tree can be disabled in order to work out the best settings for a specific problem and guarantee the generation of executable code.

Sanity checks

To verify within the algorithm if the application of the clipping-based complexity reduction and the merging procedures was successful and leads to plausible results, a testing procedure is introduced and performed on the union of polyhedral critical regions \(\tilde{R}_{X,i} = \{\tilde{R}_{1}, \tilde{R}_{2}, \ldots, \tilde{R}_{N_{R,i}}\}\) in every partition \(X_i\). The following sanity checks are conducted:

- Test if the union of polyhedra \(\tilde{R}_{X,i}\) is convex. If the union of polyhedra \(\tilde{R}_{X,i}\) is convex, it implies that the union is connected.

- Test if the union \(\tilde{R}_{X,i}\) is build from full-dimensional polyhedra \(\tilde{R}_{1}, \ldots, \tilde{R}_{N_{R,i}}\).

- Test if the union of polyhedra \(\tilde{R}_{X,i}\) contains no overlaps. This is detected in case the intersection of two full-dimensional polyhedra is full-dimensional.

- Test if the convex set of polyhedra \(\tilde{R}_{X,i}\) is bounded.

- Test if the set covered by the union \(\tilde{R}_{X,i}\) is the same as the set covered by the polyhedron \(X_i\) defining the orthogonal partition by testing if both \(\tilde{R}_{X,i} \subseteq X_i\) and \(X_i \subseteq \tilde{R}_{X,i}\).
Explicit non-linear model predictive control

If all of the tests applied to the union of polyhedra $U$ give positive results, the variable $C_U$ is set to true, otherwise to false. Some of these functions are computationally demanding and, therefore, only suitable for unions with a limited number of polyhedra. For this reason, the sanity checks can be disabled.

Structure of the algorithm

The algorithm in this section is suggested for the post-processing of the approximate mp-NLP solution based on local approximate mp-QP sub-problems defined on orthogonal partitions.

The various procedures are applied consecutively to the solution of the mp-QP sub-problem in each partition. To enhance the robustness of the complexity reduction including merging procedures and the generation of a binary search tree, the post-processing algorithm is restricted to the solution of the mp-QP problem in each orthogonal partition. If, therefore, the complexity reduction or generation of a binary search tree fails in one of the partitions, the algorithm can carry on and still lead to better results in the remaining partitions. However, the post-processing procedures could also be applied to the union of all polyhedra in case the complexity reduction and binary search tree generation were successful for each partition, leading to a single binary search tree.

Assume, the flags clipping for clipping-based complexity reduction, optimal for the disjoint optimal merging procedure, greedy for the sub-optimal merging procedure, and bst for binary search tree generation are defined.

The following algorithm 4 is suggested as a practical approach for post-processing of the explicit approximate solution of mp-NLP:

### Algorithm 4 Post-processing of the explicit approximate solution of mp-NLP

**Input:** Set of orthogonal partitions $\Pi = \{X_1, X_2, \ldots, X_{N_X}\}$; Solutions to the associated mp-QP problems $\tilde{z} = \{\hat{z}_{X_1}, \hat{z}_{X_2}, \ldots, \hat{z}_{X_{N_X}}\}$ defined by piecewise affine solution functions $\hat{z}_{X_i} = \{\hat{z}_{X_{i1}}, \hat{z}_{X_{i2}}, \ldots, \hat{z}_{X_{iN_{R,i}}}\}$ and the set of corresponding polyhedral critical regions $R_{X_i} = \{R_{1}, R_{2}, \ldots, R_{N_{R,i}}\}$ for each partition $X_i$, $i = 1, 2, \ldots, N_X$;

**Output:** Top layer including binary search tree data structure for the identification of an orthogonal partition $X_i$ for a given parameter vector $x_p$; bottom layer including search data structure for the identification of a polyhedral region $\tilde{R}_j \in \tilde{R}_{X_i}$ and defining matrices $\tilde{F}_j$ and $\tilde{g}_j$ of complexity reduced piecewise affine receding horizon feedback law $\tilde{k}(x_p)$ for a given partition $X_i$ and parameter vector $x_p$; Evaluation feasibility check including clipping function $\phi$ for evaluation of $\tilde{k}(x_p(t_k))$ and buffer storage for $\tilde{k}(x_p(t_{k-1}))$.

1: **procedure** TopLayer($\Pi$)
2: Perform sanity checks $C_{\Pi}$ to union of orthogonal partitions $\Pi = \{X_1, X_2, \ldots, X_{N_X}\}$.
3: **if** $C_{\Pi} = \text{true}$ **then**
4.4 Post-processing of the explicit approximate solution of mp-NLP

4: Generate memory-optimised binary search tree from $\Pi$.
5: Export binary search tree data structure to top layer.
6: else
7: throw Exception(“Partitions give unexpected results”).
8: end if
9: end procedure

procedure BottomLayer($\Pi$, $\hat{z}_{X_i}$, $R_{X_i}$)

for $i \in \{1, 2, \ldots, N_X\}$ do

Reduce $k(x_p) = \hat{z}_{X_i}(x_p)$ for all $x_p \in X_i$ to the receding horizon feedback law $\hat{k}(x_p)$ extracting the elements associated to the first sample only.

if clipping = true then

try

Perform clipping-based complexity reduction to union $R_{X_i}$.

Perform sanity checks $C^c_{X_i}$ to clipped union $\hat{R}_{X_i}$.

catch Exception

Display error message.

end try

end if

if clipping = true & $C^c_{X_i}$ = true then

Select clipped union of polyhedra, i.e. $S_0 := \hat{R}_{X_i}$.

else

Select original union of polyhedra, i.e. $S_0 := R_{X_i}$.

end if

try

if optimal = true then

Perform optimal merging procedure to selected union $S_0$.

else

throw Exception(“Optimal merging disabled”).

end if

Perform sanity checks $C^o_{X_i}$ to optimal merged union $\hat{R}^o_{X_i}$.

if $C^o_{X_i}$ = true then

Select optimal merged union of polyhedra, i.e. $S_0 := \hat{R}^o_{X_i}$.

else

throw Exception(“Optimal merging gives unexpected results”).

end if

catch Exception

try

if greedy = true then

Perform greedy merging procedure to selected union $S_0$.
43:     else
44:         throw Exception("Greedy merging disabled").
45:     end if
46:     Perform sanity checks $C^g_{X_i}$ to greedy merged union $\tilde{R}^g_{X_i}$.
47:     if $C^g_{X_i} = \text{true}$ then
48:         Select greedy merged union of polyhedra, i.e. $S_0 := \tilde{R}^g_{X_i}$.
49:     else
50:         throw Exception("Greedy merging gives unexpected results").
51:     end if
52: catch Exception
53:     Perform sanity checks $C_{X_i}$ to original union, i.e. $S_0 := R_{X_i}$.
54:     if $C_{X_i} = \text{false}$ then
55:         Display message("Original union gives unexpected results").
56:     end if
57: end try
58: end try
59: if bst = true then
60:     try
61:         Generate memory-optimised binary search tree from $S_0$.
62:         catch Exception
63:             Generate sequential search from $S_0$.
64:         end try
65:     else
66:         Generate sequential search from $S_0$.
67:     end if
68:     Export binary search tree and sequential search data structures.
69: end for
70: Combine search data structures and export to bottom layer.
71: Export check of feasibility and clipping function.
72: end procedure

4.4.4 Online evaluation

The post-processing algorithm 4 generates and exports real-time executable code for the online evaluation of the explicit NMPC. The actual evaluation is a three-stage process and works as follows:

1. For a given parameter vector $x_p(t_k)$, the top layer identifies the corresponding orthogonal partition $X_x$ and associated local mp-QP sub-problem by traversing the memory-optimised binary search tree. In case no orthogonal partition can be identified,
the index $x$ is set to zero. Moreover, in case the parameter vector lies outside the hyper-rectangle $X = \{ x_p \in \mathbb{R}^{n_p} \mid A_{th} x_p \leq b_{th} \}$ of the state-space to be explored, the parameter is set to $\tilde{x}_p = \min(\bar{x}_p, \max(x_p, \bar{x}_p))$ with $x_p$ and $\bar{x}_p$ as the lower and upper bounds defining the closed polytopic set of parameters. The clipped parameter $\tilde{x}_p$ is then used for further evaluations and at the same time, a flag indicating the clipping of the parameter vector is set to true.

2. Having determined the index $x$ of the orthogonal partition for the parameter vector $x_p$, the bottom layer identifies the corresponding polyhedral critical region $\tilde{R}_r$ of the complexity reduced receding horizon state-feedback law $\tilde{k}(x_p)$ by traversing the binary search tree or alternatively the sequential search data structure. It also evaluates the feedback law, defined by the matrices $\tilde{F}_r$ and $\tilde{g}_r$, and returns the value. In case the index cannot be found or the evaluation fails, the returned value of the index $r$ is set to zero and the value of the feedback law $\tilde{k}$ is set to a vector of appropriate size composed of NaN.

3. The evaluated value is passed through a clipping function to ensure that the equivalence $\tilde{k}(x_p) = \phi(\tilde{k}(x_p))$ is established. Subsequently, the value $\tilde{k}$ can be applied as suboptimal control action $\hat{u}(t_k)$ to the plant. However, in case of evaluation infeasibility, indicated by NaN as value of the feedback law or zero as one of the indices, the last control action $\hat{u}(t_{k-1})$, that is stored in a buffer, is provided as reasonable alternative control action.
Part II

Application
Chapter 5

Vehicle stability control

5.1 Introduction and literature review

Vehicle stability control

Drivers with average experience rarely know when they are driving a car at the physical limits and, therefore, have only a vague idea of the vehicle’s lateral stability margin. If the limit of adhesion between the tyres and the road is reached, the driver is caught by surprise and typically reacts in the wrong way by steering too much. The vehicle handling becomes critical for the driver as soon as the vehicle behaviour differs too much from the normally linear relationship. The primary objective of a vehicle stability controller (VSC) is, therefore, to provide vehicle stability and handling predictability within the physical operating limits. This can be achieved by preventing excessive deviations between the lateral response intended by the driver and the actual vehicle response. Since the nominal trajectory desired by the driver is not known, the driver’s inputs such as steering wheel angle, the accelerator pedal position, and the brake pressure are utilised. They are used to obtain nominal state variables, i.e. the vehicle’s yaw velocity and side-slip angle, describing the intended vehicle motion instead. The main task of the vehicle stability controller is to limit the side-slip angle to prevent vehicle spin. Moreover, the side-slip angle is kept below a characteristic value to preserve some yaw moment gain, as explained by the contribution of Shibahata et al. (1993). Since it is difficult to always obtain reliable estimates of the side-slip angle, a yaw velocity control is introduced exploiting the coupled lateral and yaw dynamics. A common approach is a model following control, typically based on the linear bicycle model of the car. If the states of the car, in terms of yaw velocity and side-slip angle, differ from the nominal state variables, limited to values corresponding to the coefficient of friction, the vehicle stability controller checks if this difference is within a tolerable dead-zone. Otherwise, an additional yaw moment needs to be generated by control of brake and traction slip at selected wheels to bring
this difference to within the tolerable zone. Appropriate action in the braking system and engine management is then taken.

Hereinafter, various approaches to the control methods and control system design in terms of reference generation, vehicle motion control, and yaw moment generation are addressed in the scope of the existing literature on vehicle stability control.

**Solutions in production vehicles**

Practical solutions integrate the vehicle dynamics control (VDC) with the anti-lock braking system (ABS), the traction control system (TCS), and the engine management. The VDC system is typically based on a hierarchical approach with cascade control using simple control methods based on classical control, e.g. proportional-integral-derivative (PID) control, and optimal control, e.g. linear-quadratic regulator (LQR). The main objective is to track nominal values of side-slip angle and yaw velocity with a cascaded control scheme, using a target yaw velocity based on the bicycle model with appropriate limitations within physical constraints. The allocation of slip is generally determined by design rules (front wheel on the outside of the turn and rear wheel on the inside of the turn) to control the longitudinal and lateral forces, and the yaw moment on the vehicle. Those rules are based on simulation studies investigating the gains of each tyre to a change in yaw moment considering the property of decreasing lateral force with increasing magnitude of the tyre slip for a given slip angle.

The first production system was introduced in 1995 to the market and is presented in van Zanten et al. (1995) describing the VDC system of Bosch. It is based on a state feedback controller using the Riccati method regulating the yaw velocity and side-slip angle based on the definition of a nominal vehicle behaviour. The weighting of the side-slip angle contribution increases as its value increases. The control of the brake slip is performed using a simple robust PID controller and of the control of the drive slip using a non-linear PID controller. In van Zanten (2000), extended work is presented giving more insights into the state estimation methods. The cascade control with a yaw velocity model following control in the inner loop and side-slip angle control in the outer loop shows good performance compared to a vehicle equipped with ABS only. More details on estimation algorithms and the model following control are given in van Zanten (2002) using an electro-hydraulic brake system, also giving an overview of possibilities for yaw velocity control based on other active systems such as active steering or active suspension. The simulations in these publications show the effectiveness of the proposed VDC, yielding a considerably reduced stopping distance with significantly improved handling performance compared to a vehicle equipped with ABS only and a passive vehicle. The suggested method leads to improved vehicle handling behaviour in limit situations with respect to the experience of normal drivers, making the vehicle motion more predictable by
following the yaw velocity and side-slip angle the driver’s input. The work in Tseng et al. (1999) addresses practical concerns in the design and development of stability control systems. Different aspects like the driver intent recognition and control development philosophy are discussed. Concepts for event anticipation, control smoothness, and system transparency in the application of the control action are developed. Moreover, algorithms for vehicle side-slip and road bank angle estimation are proposed.

For the developers of VDC systems, it is important to specify some performance requirements for the control system design to ensure progressive and non-intrusive action without annoying or disturbing the driver. Moreover, it is also essential to reduce the development time and the typical extensive tuning and testing effort for such systems. Besides these main challenges, there is a mandate requiring all new passenger cars and commercial vehicle models seeking type approval according to UN/ECE-R 13H (1958) in the EU to be equipped with an ESC system since 2011 and compulsory fitting of all newly registered vehicle with ESC since 2014. This requires also new approaches to the development and homologation of VDC systems, as described in Lutz et al. (2017) with focus on vehicle variants.

**Motivation for non-linear model predictive control**

With increasing dissemination of VSC and growing experience in the development of such systems, the interest in advanced control methods is growing. Special attention is given to model predictive control (MPC) due to multiple reasons. Suppliers of VSC systems strive for a reduction of the design and tuning effort for shorter development times. Moreover, the increasing number of active systems brings new challenges for the integration and coordination of such over-actuated systems requiring multiple-input multiple-output (MIMO) control methods. MPC allows for the design of multi-variable feedback controllers with similar procedural complexity as of single-variable ones. Moreover, constraints on the system inputs, states, and outputs can be systematically accounted for in the design phase which are then enforced by the controller. For VSC applications those can be physical constraints, e.g. friction brakes can only generate braking torques, and operational constraints, e.g. limits on yaw velocity and side-slip angle. Inherent to the MPC method is the specification of an objective function, which is optimised in a receding horizon fashion, increasing the performance of the controller. Other advantageous features are the capability of dealing with time delays and rejecting measured and unmeasured disturbances. Moreover, there is some philosophical attractiveness to MPC mimicking many processes in nature that seem to inherently operate in a similar way embodying both optimisation and feedback adjustment. This has lead to several investigations for the application of MPC in the automotive industry, e.g. the overviews in Hrovat et al. (2012) and del Re et al. (2010).
Of utmost importance is to incorporate the highly non-linear behaviour of the tyres at the limit of adhesion into the control method. This leads to non-linear model predictive control (NMPC). The general increase of computational complexity with the use of MPC increases even further by incorporating the non-linear system dynamics within NMPC. Therefore, the real-time implementation of NMPC or approximations thereof is a serious challenge and popular research question, especially considering the limited computing and memory specifications of industrial automotive embedded systems. Moreover, functional safety aspects need to be properly addressed by verification and feasibility of the control method.

Objectives of the review

Along with the growing interest in MPC methods for VSC, or more general the control of lateral and yaw dynamics with any available actuator, there is also an increasing number of scientific publications with different aims. Besides the classification of different MPC methods, the following points shall be addressed in more detail throughout the survey:

- The formulation of a constrained optimal control problem for vehicle stability control gives better performance than an unconstrained solution and additional constraints can bring performance benefits, as shown in Siampis et al. (2015c), Siampis et al. (2015a), Falcone et al. (2010), Falcone et al. (2007b), and Keviczky et al. (2006).

- The non-linear MPC has superior performance and gives better results than a linear MPC or approximate approaches such as linear time-varying MPC (LTV-MPC), as shown in Borrelli et al. (2005), Falcone et al. (2007b), Siampis et al. (2018), Falcone et al. (2008c), Falcone et al. (2006), and Besselmann and Morari (2009).

- The complexity of the prediction model and formulation of non-linear relations has a significant influence on the performance of the controller, as shown in Berntorp et al. (2014), Liu et al. (2016), Siampis et al. (2015c), Quirynen et al. (2018), Siampis et al. (2018), Li et al. (2019), Beal and Gerdes (2013), Beal (2011), Besselmann and Morari (2009), Falcone et al. (2008c), and Falcone et al. (2010).

- Existing studies, systematically investigating the influence of model complexity, only address offline MPC in simulation, such as Berntorp et al. (2014) and Liu et al. (2016).

- The modelling of wheel dynamics can be neglected for the application of MPC to vehicle stability control. However, it is necessary for an integrated approach to vehicle stability and traction control, as shown in Siampis et al. (2015c), Falcone et al. (2010), Jalali et al. (2016), Jalaliyazdi (2016), Jalali et al. (2017a), Bächle et al. (2014b), and Quirynen et al. (2018).
• The real-time implementation of non-linear MPC requiring online optimisation is challenging due to the computational complexity, as shown in Borrelli et al. (2005), Falcone et al. (2007b), Falcone et al. (2006), Falcone et al. (2008c), Falcone et al. (2010), Keviczky et al. (2006), Siampis et al. (2018), Zarkadis et al. (2018), Quirynen et al. (2018), and Bächle et al. (2014b).

• An explicit approach to MPC facilitates real-time implementation, as shown in Tøndel and Johansen (2003), Tøndel and Johansen (2005), Besselmann and Morari (2009), Di Cairano and Tseng (2010), Di Cairano et al. (2010), Di Cairano et al. (2013), Hrovat et al. (2012), and Gupta and Falcone (2018).

• The verification of an embedded MPC implementation for reliability and correctness is possible through an explicit approach enabling offline verification of the solution, as shown in Gupta and Falcone (2018), Tøndel and Johansen (2003), and Tøndel and Johansen (2005).

• It is important to reduce delays and minimise any additional delays coming from the control system. Moreover, it is possible to handle delays in a systematic way with an MPC approach, as shown in Jalaliyazdi et al. (2015), Jalaliyazdi (2016), Jalali et al. (2017b), Tseng et al. (1999), Hrovat et al. (2012), Beal and Gerdes (2013), Beal (2011), Siampis et al. (2018), Canale et al. (2010), Canale et al. (2009b), and Canale and Fagiano (2008).

The aim of this literature study is to show the beneficial influence of non-linear modelling of the relevant vehicle dynamics. The need for a systematic study investigating the influence of different prediction model complexities on the controller performance of real-time capable NMPC approaches is demonstrated. Moreover, the motivation for an explicit solution is shown facilitating real-time implementation and giving the possibility of verification of the solution being of significant importance for safety-critical applications.

Classification of MPC methods

Non-linear MPC Several variations of MPC have been proposed for vehicle stability control. Employing the non-linear system dynamics in the prediction model leads to non-linear MPC. This results in a non-linear programming (NLP) problem, which is computationally expensive to solve. The implicit solution involving NLP problems is, therefore, not attractive for practical applications, despite the typically excellent performance of NMPC.

In Borrelli et al. (2005), an implicit NMPC approach to path following based on autonomous steering is presented. A bicycle model with a non-linear tyre model is used as a prediction model, which, however, neglects load transfer effects. The influence
of increasing entry speeds on the minimum required prediction and control horizon to stabilise the vehicle in a double-lane-change manoeuvre is investigated in simulations. The NMPC controller shows good performance, but the resulting increase in computational complexity, and subsequently computational time, would limit a possible experimental validation to low speeds, typically below $10 \text{ m/s}$. Building on this analysis, Keviczky et al. (2006) demonstrate the capability of the NMPC to reject wind gusts and changes in tyre-road friction. In Falcone et al. (2007b), the implicit NMPC scheme is implemented on a dSPACE rapid control prototyping (RCP) unit and experimental tests on icy roads demonstrate the real-time capability. The NMPC controller is able to stabilise the vehicle in the experiments only up to a speed of $7 \text{ m/s}$ since the solver does not converge for higher speeds to a feasible solution due to a sub-optimal approach limiting the number of iterations. To facilitate the real-time implementation, a linear time-varying MPC is proposed employing a linear model derived by successively linearising the non-linear model online. The LTV-MPC stabilises the vehicle up to much higher speeds but needs additional constraints. However, the NMPC shows better performance compared to the approximate LTV-MPC approach. Extending Borrelli et al. (2005) and Falcone et al. (2007b), two implicit combined steering and braking NMPC path following controllers with different model complexities are presented in Falcone et al. (2008c). The controller with the more complex model, i.e. a double-track vehicle model including wheel dynamics with the steering angle and individual wheel torques as inputs, has improved performance compared to the controller with the simpler model, i.e. a single-track model with steering angle and direct yaw moment as input distributed by a simple control allocation scheme. Moreover, it has the ability to stabilise the vehicle for higher speeds, despite being difficult to tune and not being capable of real-time execution. Here again, a non-linear tyre model with combined slip modelling is employed, but no load transfer effects were considered. In Falcone et al. (2008a), the performance benefit of an NMPC for replanning of the reference trajectory for autonomous path following is demonstrated. Similarly, in Falcone et al. (2010) a comparison of two design paradigms for predictive control of autonomous vehicles is given. The two-level approach consists of an over-simplified model in the trajectory re-planner and a linear model in the low-level. The single-level approach consists of a more complex non-linear vehicle model in the low-level only. Another approach for path following with active steering is presented in Quirynen et al. (2018) employing the real-time iteration (RTI) scheme for implicit NMPC. Simulation results demonstrate a significant influence of the tyre-road friction on the tracking performance. However, interestingly there is no significant difference between a single-track and a double-track vehicle as a prediction model for this particular application.

Apart from applications for autonomous driving, there is also extensive work on active safety and the improvement of vehicle handling. In Bächle et al. (2014b), an implicit NMPC approach to control allocation is presented for a torque vectoring (TV)
controller. The predictive approach is combined with a static preallocation algorithm and yaw moment tracking with feed-forward and feedback contribution. Extending the work of Siampis et al. (2015c), in Siampis et al. (2018) two real-time implementable NMPC approaches to combined longitudinal and lateral control using rear-axle TV are discussed. The comparison of a linear time-varying MPC, a non-linear MPC using the sub-optimal RTI scheme, and a non-linear MPC using the primal-dual interior point (PDIP) method shows an acceptable performance of the PDIP method, while unstable behaviour of the RTI scheme in simulation scenarios. Even though the linear time-varying MPC yields the shortest execution times, the non-linear MPC using the PDIP method can achieve much better performance, while still being implementable online, as demonstrated on a dSPACE RCP unit using soft constraints and a limited number of iterations. Rather than tracking a velocity reference, as in Siampis et al. (2018) and previous work thereof, the controller design in Zarkadis et al. (2018) allows for the intervention of the driver’s torque demand to regulate vehicle speed by introducing constraints to a region where the requested lateral acceleration becomes feasible. The proposed torque vectoring controller is based on an implicit NMPC approach also using the PDIP method. The concept of not relying on exact optimisation is also incorporated in Tjønnås and Johansen (2010) proposing a modular hierarchical control scheme for vehicle stability control based on differential braking. In particular, the intermediate control allocation scheme utilises sub-optimal dynamic optimisation converging to the optimum in a stable manner. Together with a simple PI controller in the high-level vehicle motion control and a low-level Lyapunov-based slip control, the method is suitable for low cost automotive electronic control units (ECU), due to its low memory and computational requirements. Motivated by real-time implementation, Guo et al. (2019) propose an implicit NMPC approach to vehicle stability control and its hardware implementation on a field programmable gate array (FPGA) unit employing particle swarm optimisation (PSO) techniques exploiting parallel search capabilities. Simulation and experimental hardware-in-the-loop (HIL) test results show improved performance of the proposed NMPC approach with active steering and braking compared to a controller based on braking only. As a prediction model, a single-track model neglecting load transfer effects with a simple rational function as a tyre model is employed in this study. Instead of a physical modelling approach, Canale et al. (2012) propose an empirical modelling approach based on a non-linear set membership identification methodology using previously collected input-output data. The implicit NMPC-based on the identified model for improvement of the vehicle handling using active front wheel steering has allegedly improved robustness compared to an NMPC approach based on a physical model.

As seen from the literature discussed so far, there are some studies comparing different models. However, there are only a few works with a systematic analysis on the influence of model complexity. Liu et al. (2016) investigate in simulations the level of model fidelity
Vehicle stability control

needed to successfully navigate an autonomous truck in obstacle avoidance scenarios. The importance of a non-linear tyre model and modelling of longitudinal load transfer effects for a single-track model is demonstrated in open-loop simulations. Closed-loop simulations of low-speed scenarios with an implicit NMPC approach based on quantised inputs show results of the single-track model with a linear tyre model and no load transfer effects comparable to the results of a 14 degree-of-freedom (DoF) benchmark model. However, for higher speeds, a non-linear tyre model and modelling of load transfer effects are necessary to successfully complete the obstacle avoidance manoeuvre. It is interesting to observe that only the effect of longitudinal load transfer is investigated, which is, given the assumption of constant speed, typically only significant for high values of the vehicle side-slip angle. Another investigation can be found in Berntorp et al. (2014) for a minimum-time optimal control problem for trajectory generation using front wheel steering, braking, and rear wheel drive. The simulation study gives insights into the balance between vehicle model detail, in terms of tyre and chassis models of different complexities, and computational complexity. It is shown that the tyre model has a significant influence without giving, however, a general conclusion for the best model. Despite the context of trajectory planning and the lack of possible real-time implementation, this study contains probably the most comprehensive analysis of the modelling influence.

Linear time-varying MPC Non-linear MPC has very good performance but requires the online solution of a non-convex constrained non-linear optimisation problem for the implicit approach. Another frequently used method is the successive online linearisation of the non-linear model around the current vehicle states leading to a linear prediction model at each sample time. Based on the norms used in the objective function, the linear time-varying MPC approach requires solving a linear programming (LP) or quadratic programming (QP) problem, which is considerably easier to solve than a non-linear programming problem.

Based on the work of Borrelli et al. (2005), Falcone et al. (2006) propose an LTV-MPC scheme for possible real-time implementation of a path following controller. The reduction of computational complexity with respect to the NMPC approach is accompanied by a slight decrease in performance. In Falcone et al. (2007b), the LTV-MPC scheme is implemented on a dSPACE RCP unit and stabilises the vehicle up to much higher speeds of 21.5 m/s compared to the proposed NMPC scheme due to the reduced computational complexity. However, to obtain stable behaviour and acceptable performance, additional constraints on the tyre slip angle, implicitly keeping the front wheel steering angle in range, and a further reference, i.e. for the yaw velocity, need to be included. Moreover, a low order LTV-MPC with acceptable performance and possible implementation on low-cost hardware is proposed. Extending the work of Borrelli et al. (2005) and Falcone
et al. (2007b), an implicit LTV-MPC based path following control using active steering and differential braking shows better performance in Falcone et al. (2007a) compared to a controller with active steering only. Falcone et al. (2007c) extend the formulation by actuation of active differentials in addition to steering and braking. It is demonstrated that the controller can, therefore, achieve better tracking performance without excessively slowing down the vehicle. In Falcone et al. (2008b), a generalisation of the stability conditions on the LTV-MPC scheme is proposed, using additional convex constraints instead of the ad hoc constraints on the tyre slip angle, as proposed in Falcone et al. (2007b). While the sufficient condition guarantees uniform asymptotic stability, the ad hoc constraints designed by expert knowledge lead to similar performance and much less computing time. The prediction models employed in these studies are a single-track model in Falcone et al. (2006), Falcone et al. (2007b), and Falcone et al. (2008b) and a double-track model in Falcone et al. (2007a) and Falcone et al. (2007c) with a non-linear tyre model neglecting wheel dynamics and effects of load transfer.

With regards to active safety, an implicit LTV-MPC approach to vehicle stability control based on independent braking of the wheels is proposed in Barbarisi et al. (2009a) and Barbarisi et al. (2009b). The optimal braking forces are calculated in the stability controller and are then passed to a separate slip control module to generate the actual braking forces. The yaw velocity and side-slip angle tracking controller is combined with a supervisory controller with activation logic based on thresholds for yaw velocity and side-slip angle. The performance of the controller is evaluated in simulations of a series of Sine with Dwell (SwD) tests fulfilling the requirements of the respective standards. Complementary work giving more details on the slip control scheme is presented in Palmieri et al. (2009) also using a double-track vehicle model considering the effects of longitudinal and lateral load transfer. Moreover, a comparison in Palmieri et al. (2010) of the MIMO LTV-MPC approach with a parallel single-input single-output (SISO) PI/P controller approach using a simple control allocation shows some performance benefits of the proposed LTV-MPC controller. In Beal and Gerdes (2013) and Beal (2011), the development of a safe handling envelope in the yaw velocity and side-slip angle domain and the formulation of an MPC strategy for a driver assistance system is presented. Based on a rear-wheel driven vehicle demonstrator with active front wheel steering, an implicit LTV-MPC envelope controller is proposed using a linearised tyre model on the rear axle and an affine force input on the front axle. A comparison with a linear formulation shows significant performance enhancements of the LTV-MPC approach considering tyre saturation and longitudinal and lateral tyre force coupling. A similar approach is proposed in Jalali et al. (2016) and Jalaliyazdi (2016) for integrated vehicle stability and slip control using LTV-MPC. A double-track vehicle model with wheel dynamics is employed as a prediction model. Simulation and experimental results show the performance in varying road conditions using vehicle demonstrators with different
drive-line configurations, i.e. four electric motors and torque vectoring or two electric motors on the rear axle and differential braking as actuation. Related work is presented in Jalali et al. (2017a) extended by a vehicle state estimation scheme. Moreover, the vehicle side-slip angle is indirectly controlled by adjusting the reference yaw velocity, thus, reducing the complexity of the prediction model. Even though the integration of vehicle and wheel dynamics makes a separate traction control module obsolete, Siampis et al. (2015c) show, however, that the inclusion of wheel dynamics in the optimisation problem does not bring noticeable performance benefits. Two implicit LTV-MPC strategies for combined longitudinal, lateral, and yaw control using rear-axle TV are compared in simulation, i.e. i) a prediction model that includes the vehicle dynamics and much faster wheel dynamics with the wheel torques as inputs and ii) a prediction model including the vehicle dynamics but neglecting the wheel dynamics with the rear longitudinal wheel slips as inputs. It is shown that the modelling of wheel dynamics requires for much higher sampling times, results in a more complex optimisation problem, does not increase the performance noticeably, and is significantly more sensitive against reductions of the control horizon. Previous studies in Siampis et al. (2015a) show the performance benefits and importance of input and state constraints in stabilising the vehicle in a comparison of an LQR and an LTV-MPC approach. Moreover, the influence of sampling time and prediction horizon on the computational complexity and closed-loop performance is investigated. Based on the work in Siampis et al. (2015a), the LTV-MPC controller is coupled with a continuous-discrete unscented Kalman filter (UKF) in Siampis et al. (2015b) for estimation of the full state feedback. The employed prediction model in Siampis et al. (2015c), Siampis et al. (2015a), and Siampis et al. (2015b) is a double-track vehicle considering the effects of longitudinal and lateral load transfer with a non-linear combined slip tyre model. An LTV-MPC approach for a control allocation method is proposed in Chang and Gordon (2009) as part of a hierarchical vehicle stability control including longitudinal motion control. In Li et al. (2019), a comparison of an implicit NMPC and two LTV-MPC controllers, i.e. with constant and varying cornering stiffness over the prediction horizon, is presented. The LTV-MPC controller with constant matrices and no additional constraints, such as in Falcone et al. (2007b), cannot stabilise the vehicle in low friction conditions.

As a concluding remark, in the addressed literature the model is linearised around an operating point, which is generally not an equilibrium point. Given the current state as an initial condition, a linear time-invariant (LTI) model is derived to predict the deviations from the operating point. However, in case a nominal state and input trajectory are known, the non-linear vehicle model can be linearised around the trajectories leading to a linear time-varying model to predict the deviations from the nominal trajectories over the prediction horizon.
Hybrid and switched MPC  As an alternative to successively linearising the non-linear prediction model, the important non-linearities of the dynamic system can be approximated by several piecewise affine (PWA) functions. This leads to a hybrid or mixed-integer dynamic system as a prediction model, where based on the states one of the affine functions is active at each time instant. This is indicated by the index of the functions forming one of the variables of the system. Solving the optimal control problem in a hybrid MPC approach requires solving a mixed-integer linear programming (MILP) or mixed-integer quadratic programming (MIQP) problem dependent on the norms used in the objective function.

In Bernardini et al. (2009), an implicit hybrid MPC approach to vehicle stability control is presented using active front wheel steering and differential braking. In the hybrid dynamical system, the tyre force characteristics are approximated by PWA functions. An analysis shows the stability region of the open-loop and closed-loop dynamics using phase-plane plots in the tyre slip angle domain. Further work is presented in Di Cairano et al. (2013) demonstrating the beneficial coordination of the actuators, i.e. effective differential braking, however, perturbing longitudinal dynamics together with active front steering having reduced authority but being less intrusive, compared to braking as actuation only. To improve the robustness with respect to uncertainties such as modelling errors, Palmieri et al. (2012) propose a robust control approach based on reachability analysis and invariant set theory. The non-linear vehicle dynamics are modelled as PWA system for lateral stabilisation based on active front steering and differential braking.

Since the complexity of the hybrid MPC approach makes it unsuitable for online implementation, a switched MPC approach can be used as an alternative. In the hybrid MPC approach, the PWA mode indicated by the index can vary along the prediction horizon. Since the closed control loop exhibits relatively few mode switches for the hybrid MPC approach in Di Cairano et al. (2013), a prediction model is considered in the switched MPC approach, where the PWA mode is kept constant over the prediction horizon. To further reduce complexity for real-time implementation on automotive-grade ECUs an explicit approach to switched MPC is developed.

Linear MPC  The simplest but least accurate way of modelling the system dynamics is to use a linear prediction model. In Liu et al. (2018), a linear implicit MPC approach to path following is presented considering the effect of known road topography including curvature and bank angle. The controller with roll-over prevention is based on a single-track model with roll dynamics. A comparison of different controllers in simulation shows that the consideration of both aspects of the road topography gives an increased performance with the influence of road curvature being more significant. In Siampis et al. (2013), a linear vehicle model is employed for a rear-axle-based TV approach using scheduled LQR for combined yaw stabilisation and velocity regulation at the limits of
Vehicle stability control

handling. A further LQR approach to TV is proposed in Esmailzadeh et al. (2003) investigating the influence of vehicle speed on the control law. Anwar (2005) proposes a simple unconstrained predictive approach to vehicle stability control. The transfer function from yaw moment as input to yaw velocity as output, assuming zero steering wheel angle, is used as a prediction model. The controller using a rule-based control allocation scheme is tested in experiments on a vehicle with electromagnetic actuators. In Mirzaeinejad et al. (2018), an implicit unconstrained MPC using a one-step prediction based on Taylor series expansion of the prediction model is proposed for vehicle stability control using active front wheel steering and differential braking. Using a stability index based on phase plane analysis, a fuzzy logic is designed to determine the weight of each control input in various conditions designed for a minimal intervention of direct yaw moment. Simulations show better results in comparison with a controller based on braking only, however, exhibiting significant oscillations in both cases.

Explicit approaches to MPC  All the presented MPC methods so far are characterised by an implicit approach that requires the solution of the respective mathematical program online. Considering the computational complexity of the different approaches this can be a very challenging task to accomplish in real-time. Moreover, given the fact that the usual application of the suggested control problems is safety-critical, real-time optimisation seems not to be a technique applicable for practical implementation. A viable alternative is an explicit approach to MPC where the underlying mathematical program is formulated as a parametric optimisation problem, e.g. a multi-parametric non-linear program (mp-NLP) in lieu of a non-linear program. For a special class of parametric programs, it is possible to derive an exact solution offline over a defined parameter space that can be then stored in the memory of the ECU. In the online implementation of the controller, the respective PWA solution needs to be located and evaluated.

In Tøndel and Johansen (2003), an explicit non-linear MPC method using multi-parametric quadratic program (mp-QP) approximations of mp-NLP is proposed as real-time scheme for vehicle stability control actuated by brakes. A double-track vehicle model is employed as prediction model neglecting the effects of load transfer. An analogous prediction model is used in Tøndel and Johansen (2005) for a control allocation problem based on explicit NMPC together with a simple proportional yaw moment reference generation in a hierarchical control scheme. In both approaches, an additional brake request from the driver is not taken into consideration in the control system design. Another approximate approach to mp-NLP is presented in Canale et al. (2008) proposing the fast non-linear MPC method. It employs set membership function estimation methodologies under the nearest point approach based on the offline computation of a finite number of exact NMPC solutions. In Canale and Fagiano (2008), this method is
applied to the improvement of vehicle handling by torque vectoring using a rear active differential. Canale et al. (2009b) and Canale et al. (2010) demonstrate the real-time implementation in software-in-the-loop (SIL) experiments on a low performance embedded processor with relatively high memory specifications. A comparison in simulations shows good agreement with the nominal implicit NMPC approach. However, the approximate solution leads to chattering in the results, which could be potentially corrected increasing the number of offline computed points at the expense of increased memory and computational requirements. In Canale and Fagiano (2008), Canale et al. (2009b), and Canale et al. (2010), a single-track vehicle neglecting the effects of load transfer is employed as prediction model and the actuator dynamics are modelled as transportation delay.

In Besselmann and Morari (2009), a linear parameter-varying (LPV) single-track model with the vehicle speed as scheduling variable is used for a path following controller with active steering. An explicit LPV-MPC approach integrated into a cascade control is proposed to reduce complexity. A comparison with an implicit non-linear MPC and explicit linear MPC demonstrates good performance of the NMPC and significant oscillations of the other controllers for high speeds, but no noticeable difference for low speeds.

The proposed implicit hybrid MPC approaches in Bernardini et al. (2009) and Di Cairano et al. (2013) can also be synthesised in an explicit form by multi-parametric mixed-integer quadratic programming (mp-MIQP). However, the explicit hybrid MPC controller is complex in terms of memory occupancy and worst case computational complexity even if symmetry and mode selection properties are exploited. This is due to a large number of possible PWA mode sequences along the prediction horizon.

Di Cairano et al. (2010), Di Cairano and Tseng (2010), and Di Cairano et al. (2013) suggest an explicit switched MPC method as alternative neglecting the effects of mode switches, i.e. linear and saturated tyre condition, during the prediction horizon. This requires the solution of a limited number of mp-QP problems instead of the generally more complex solution of an mp-MIQP problem. In Di Cairano et al. (2010), a vehicle stability controller using active front wheel steering and differential braking is proposed stabilising the vehicle dynamics and tracking the yaw velocity requested by the driver. Taking into account the driver’s interaction physically affecting the vehicle steering angle, a driver-assist system is proposed in Di Cairano and Tseng (2010) and Di Cairano et al. (2013). Simulation and experimental results show an acceptable performance of the controller despite limit cycle behaviour in steady-state conditions with unachievable reference values. The reason lies in the switched MPC method alternating between yaw velocity tracking and tyre slip angle regulation. Experimental results demonstrate the real-time implementation on a dSPACE RCP unit successfully stabilising the vehicle on a slippery road by constraining the tyre slip angles within their limits. A single-track model neglecting the load transfer effects is employed as prediction model in those studies.
Motivated by the Automotive Safety Integrity Level (ASIL) D type of requirements mentioned in ISO 26262-2 (2018) for safety and performance criteria in terms of recursive feasibility and stability, Gupta and Falcone (2018) propose an explicit linear MPC approach for path following based on front wheel steering. The method incorporates terminal constraints using a low complexity robust control invariant set to guarantee bounds on the lateral deviation. An analysis demonstrates the influence of the prediction horizon on the complexity of the explicit solution enabling extensive verification and validation offline before implementation.

**Delay handling**

Independently of the underlying control methodology used, all controllers are subject to performance degradation in the presence of delays in the control loop. Delays occur in any practical control system originating from various sources such as the sensory and measurement system, filters exhibiting phase shifts, communication, dynamics of actuators, etc. If the delays are minimal, then the controller is often designed by ignoring the delay. However, if the amount of delays increases it can potentially have a severe impact on the performance of the control loop, even inducing oscillations or eventually causing instability of the controller. It is, therefore, essential to reduce general delays in the control loop and furthermore minimise any additional delays caused by the execution of the controller. Especially in the context of the implicit approach to MPC, the extensive computational demand at each sampling time requires non-trivial processing time contributing to the overall delay in the control loop. The explicit approach generally results in significantly reduced online computational complexity, thus, potentially minimising the effects of additional delays. Moreover, the reduction of computing times allows for smaller sampling times having a significant influence on the controller in terms of closed-loop performance and disturbance rejection.

The MPC approach, in general, enables the systematic handling of delays. In Jalaliyazdi et al. (2015) for instance, a delay handling technique is proposed for a TV controller of an electric vehicle based on an implicit LTV-MPC scheme. Using the internal prediction model, based on a double-track vehicle model, the predicted states at the end of the delay period is controlled instead of the currently measured states. This is the earliest time that the effect of any control action is visible on the system. The controller with the delay handling scheme has significantly better performance and stable behaviour compared to the conventional MPC scheme without delay handling. Further work can be found in Jalali et al. (2017b), and Jalaliyazdi (2016), including first-order delays in the systematic delay handling method in addition to the transportation delay. Simulation results show good performance of the proposed easy implementable technique and experimental results confirm the findings leading to reduced values of the vehicle
side-slip angle and yaw velocity overshoot of the rear-wheel driven vehicle. A very similar approach can be found in Beal (2011), Beal and Gerdes (2013), and Siampis et al. (2018) compensating for the non-trivial execution time of the controller by employing the states predicted for the first time step within the prediction horizon. This is expressed directly in the problem formulation assuming that these states are only dependent on the initial conditions. In Canale et al. (2010), Canale et al. (2009b), and Canale and Fagiano (2008), the MPC accounts for the actuator dynamics in the optimal control problem formulation by approximating them as a transportation delay on the input.

To conclude, MPC offers the possibility of systematic and straightforward handling of delays reducing the influence on the control loop to a minimum. However, it should be noted that the effects of delays cannot be fully compensated since the performance in terms of disturbance rejection will always suffer in the presence of delays. It is, therefore, important to minimise any additional delays where possible.

Contribution

In the review of the relevant literature on predictive approaches to vehicle stability control, one important aspect was to demonstrate the beneficial influence of non-linear modelling of the vehicle dynamics. The tyre force characteristics have the most significant influence and for that reason, a tyre model describing the important non-linearities of the dynamical system is chosen considering the interaction of longitudinal and lateral tyre forces. Moreover, a double-track vehicle model is selected capturing the significant effects of lateral and longitudinal load transfer in the context of vehicle stability control. The proposed non-linear MPC approach is focused on the vehicle motion control with the wheel slip control as a separate module. The wheel dynamics are neglected in the formulation since they are much faster than the vehicle dynamics and, thus, are not contributing to additional performance. The modelling choices are motivated by the reduction of complexity to allow for real-time implementation while still capturing all significant physical phenomenons. An appropriate definition of constraints complements the prediction model for a practical formulation of the optimal control problem.

Another relevant aspect in the literature review was the influence of varying prediction model complexity on the performance of predictive controllers. The common justification for the choice of the models is the low computational load required. In fact, some studies do compare MPC approaches with different model complexities. However, the available systematic studies do not focus on real-time implementation and, moreover, not on vehicle stability control. Vice versa, those studies targeting real-time implementation do not include a systematic analysis of the various effects. Since the answer to this research question depends on the context, the effects of varying complexity of the prediction model are investigated in the scope of a real-time non-linear MPC approach to vehicle
stability control. The objective is to determine the level of model fidelity required in the MPC formulation to obtain the best controller performance.

The final relevant aspect is the motivation for an explicit approach to non-linear MPC. The required real-time optimisation for an implicit approach to MPC is not a technique applicable for practical implementation in a real vehicle for a safety-critical application such as vehicle stability control. The explicit approach guarantees a level of sub-optimality and allows for offline verification and validation of the solution beforehand. Since the online computational complexity is generally significantly reduced, the explicit approach facilitates real-time implementation. The decreased execution times minimise the influence of additional delays in the control loop and even allow for a reduction of the sampling time increasing the performance compared to an implicit approach.

The different research gaps identified in the review are addressed in the following publications.

In Metzler et al. (2019c), an explicit non-linear MPC approach to vehicle stability control is proposed. The formulation of the optimal control problem is motivated by practical application. The flexibility of the NMPC cost function formulation adopted will allow ease of implementation of the controller on real vehicles, with different rather complex performance requirements for the stability control function. Compared to existing explicit NMPC approaches in the literature, the effects of lateral load transfer are included in the non-linear vehicle model for control system design. Together with the coupling of longitudinal and lateral tyre forces incorporated in the formulation, both effects are crucial to the exploitation of the benefits of NMPC for vehicle control at the limits of handling. The explicit approach to non-linear MPC is motivated by real-time implementation and the possibility of verification for safety-critical applications while having the benefits of modelling the non-linear dynamics.

Extending this work, in Metzler et al. (2018), the influence of constraints and the effects of the weights in the cost function on the controller performance is investigated.

Based on the previous work, in Metzler et al. (2019b), a systematic comparison of different prediction models suitable for real-time implementation is proposed analysing the influence of varying model complexity on the performance of the non-linear MPC vehicle stability controller. In particular, the importance of modelling the non-linear tyre characteristics, the interaction of lateral and longitudinal tyre forces, and the effects of lateral and longitudinal load transfer are demonstrated in a detailed study. Moreover, the robustness of the explicit non-linear MPC-based vehicle stability controller against parameter variations of the controlled vehicle is demonstrated. Compared to the previous work, the formulation is extended considering the braking request from the driver and includes the effects of longitudinal load transfer.
5.2 Control system design

In the following sections, all necessary definitions for the optimal control problem of the non-linear model predictive controller are introduced. This includes the design of the cost function and the definition of reference targets. Furthermore, functions describing the system dynamics as well as physical and operational constraints are introduced.

5.2.1 Prediction models

This section deals with the prediction models of the non-linear model predictive controller including variations in the model fidelity.

Lateral force and yaw moment balance equations

A double-track vehicle model (Fig. 5.1), with the yaw velocity, \( \dot{\psi}(t) \), and vehicle side-slip angle, \( \beta(t) \), as state variables, is used as a prediction model for the formulation of the optimal control problem. The longitudinal dynamics are neglected, as a constant speed, \( v(t_k) \), is assumed over the prediction horizon. However, the longitudinal load transfer is considered in some of the models, see Section Vertical tyre forces. The Newton-Euler lateral force and yaw moment balance equations of the vehicle rigid body are

\[
\dot{\beta}(t) = \frac{1}{mv(t_k)} \left[ \left( F_{FL}(t) + F_{FR}(t) \right) \sin(\delta(t_k) - \beta(t)) \\
+ \left( F_{sFL}(t) + F_{sFR}(t) \right) \cos(\delta(t_k) - \beta(t)) \\
- \left( F_{iRL}(t) + F_{iRR}(t) \right) \sin \beta(t) + \left( F_{sRL}(t) + F_{sRR}(t) \right) \cos \beta(t) \right] - \dot{\psi}(t),
\]

(5.1)

\[
\ddot{\psi}(t) = \frac{1}{I_z} \left[ \left( F_{FL}(t) + F_{FR}(t) \right) \sin \delta(t_k) l_F + \left( F_{sFL}(t) + F_{sFR}(t) \right) \cos \delta(t_k) l_F \\
- \left( F_{sRL}(t) + F_{sRR}(t) \right) l_R - \left( F_{iFL}(t) - F_{iFR}(t) \right) \cos \delta(t_k) \frac{d}{2} \\
+ \left( F_{sFL}(t) - F_{sFR}(t) \right) \sin \delta(t_k) \frac{d}{2} - \left( F_{iRL}(t) - F_{iRR}(t) \right) \frac{d}{2} \right],
\]

(5.2)

where \( m \) is the vehicle mass; \( I_z \) is the yaw mass moment of inertia; \( l_F \) and \( l_R \) are the front and rear semi-wheelbases; \( d \) is the track width; \( \delta(t_k) \) is the steering angle, which is assumed to be equal on the left and right front wheels; \( F_{ij} \) and \( F_{sij} \) are the longitudinal and lateral tyre forces, respectively, with the subscripts \( i = F,R \) referring to the front and rear axles, and \( j = L,R \) to the left and right sides.

As the vehicle stability controller ensures small side-slip angle values in any condition, and the relevant manoeuvres imply rather limited values of steering angle, (5.1) and (5.2) are linearised with respect to the arguments \( \beta(t) \) and \( \delta(t_k) - \beta(t) \).
Vertical tyre forces

For the estimation of the vertical tyre forces, \( F_{zij} \), three different models are considered:

A.i The static vertical tyre loads, \( F_{zij,\text{stat}} \), on the individual tyres are estimated as

\[
F_{z\text{FL,stat}} = F_{z\text{FR,stat}} = \frac{1}{2} \frac{l_R}{l_F + l_R} mg, \quad (5.3)
\]

\[
F_{z\text{RL,stat}} = F_{z\text{RR,stat}} = \frac{1}{2} \frac{l_F}{l_F + l_R} mg, \quad (5.4)
\]

where \( g \) is the gravitational acceleration.

A.ii The estimation of varying vertical tyre forces considering the load transfer associated with the normal acceleration in steady-state cornering conditions and an estimation, \( \hat{\gamma} \), of the tangential acceleration, yields

\[
F_{z\text{FL}}(t) = F_{z\text{FL,stat}} - C_x \hat{v}(t_k) - C_{yF} v(t_k) \hat{\psi}(t_k), \quad (5.5)
\]

\[
F_{z\text{FR}}(t) = F_{z\text{FR,stat}} - C_x \hat{v}(t_k) + C_{yF} v(t_k) \hat{\psi}(t_k), \quad (5.6)
\]

\[
F_{z\text{RL}}(t) = F_{z\text{RL,stat}} + C_x \hat{v}(t_k) - C_{yR} v(t_k) \hat{\psi}(t_k), \quad (5.7)
\]

\[
F_{z\text{RR}}(t) = F_{z\text{RR,stat}} + C_x \hat{v}(t_k) + C_{yR} v(t_k) \hat{\psi}(t_k). \quad (5.8)
\]

The formulations (5.5)-(5.8) neglect the influence of the side-slip angle \( \beta \) and, therefore, assume that the tangential and normal acceleration are equal to the longitudinal and lateral acceleration, respectively.

A.iii The estimation of varying vertical tyre forces considering the load transfer associated with the normal acceleration including an estimation, \( \hat{\beta} \), of the side-slip angle rate
and an estimation of the tangential acceleration, follows to

$$F_{zFL}(t) = F_{zFL,stat} - C_x \hat{v}(t_k) - C_{yFL} v \left( \dot{\psi}(t_k) + \hat{\beta}(t_k) \right),$$  \hspace{1cm} (5.9)

$$F_{zFR}(t) = F_{zFR,stat} - C_x \hat{v}(t_k) + C_{yFR} v \left( \dot{\psi}(t_k) + \hat{\beta}(t_k) \right),$$  \hspace{1cm} (5.10)

$$F_{zRL}(t) = F_{zRL,stat} + C_x \hat{v}(t_k) - C_{yFR} v \left( \dot{\psi}(t_k) + \hat{\beta}(t_k) \right),$$  \hspace{1cm} (5.11)

$$F_{zRR}(t) = F_{zRR,stat} + C_x \hat{v}(t_k) + C_{yRR} v \left( \dot{\psi}(t_k) + \hat{\beta}(t_k) \right),$$  \hspace{1cm} (5.12)

neglecting again the influence of the side-slip angle $\beta$.

The calculations are based on the constants $C_x$, $C_{yFL}$, and $C_{yRR}$, defined as

$$C_x = \frac{h}{2} \left( \frac{l_F}{l_F + l_R} m \right),$$  \hspace{1cm} (5.13)

$$C_{yFL} = \left( \frac{l_R}{l_F + l_R} \frac{h_F}{d} + \frac{K_F}{K_F + K_R} \frac{h'}{d} \right) m,$$  \hspace{1cm} (5.14)

$$C_{yRR} = \left( \frac{l_F}{l_F + l_R} \frac{h_R}{d} + \frac{K_R}{K_F + K_R} \frac{h'}{d} \right) m,$$  \hspace{1cm} (5.15)

where $h$ is the height of the vehicle centre of gravity; $h_F$ and $h_R$ are the roll centre heights of the front and rear suspensions; $h'$ is the distance between the roll axis and the vehicle centre of gravity; and $K_F$ and $K_R$ are the roll stiffness values of the front and rear suspensions.

The tangential acceleration is estimated as

$$\hat{\dot{v}}(t) = \frac{1}{m} \left( F_{zFL}(t_k) + F_{zFR}(t_k) + F_{zRL}(t_k) + F_{zRR}(t_k) \right),$$  \hspace{1cm} (5.16)

and the vehicle side-slip angle rate is estimated employing a single-track model,

$$\hat{\dot{\beta}}(t) = \frac{1}{m v(t_k)} \left( F_{a0F}(t_k) + F_{a0R}(t_k) \right) - \dot{\psi}(t_k).$$  \hspace{1cm} (5.17)

The lateral axle forces, $F_{a0F}$ and $F_{a0R}$, on the front and the rear, respectively, are calculated based on non-linear functions of the front and rear axle slip angles, $\alpha_F$ and $\alpha_R$,

$$F_{a0F} = D_{aF} \sin \left( C_{sF} \arctan(B_{aF} \alpha_F) \right),$$  \hspace{1cm} (5.18)

$$F_{a0R} = D_{aR} \sin \left( C_{sR} \arctan(B_{aR} \alpha_R) \right),$$  \hspace{1cm} (5.19)

with lateral peak factors, $D_{a_i}$, and stiffness factors, $B_{a_i}$, linearly dependent on the estimated tyre-road friction coefficient, $\hat{\mu}$, and constants for the shape factors, $C_{s_i}$. 
Vehicle stability control

Pure slip lateral tyre force model

The lateral tyre force characteristics in pure lateral slip conditions are modelled with three different approaches:

B.i The linear tyre model approximates the lateral forces in pure lateral slip conditions, $F_{s0,ij}$, with a simple linear function of the tyre slip angles, $\alpha_{ij}$,

$$F_{s0,ij} = B_{sij} C_{sij} D_{sij} \alpha_{ij}. \quad (5.20)$$

B.ii The non-linear simplified model of the Pacejka Magic Formula (MF), Pacejka (2012), approximates the non-linear tyre behaviour in pure lateral slip conditions,

$$F_{s0,ij} = D_{sij} \sin(C_{sij} \arctan(B_{sij} \alpha_{ij})). \quad (5.21)$$

B.iii The non-linear saturation model approximates the non-linear tyre behaviour in the low slip regions and saturates for higher slips,

$$F_{s0,ij} = D_{sij} \tanh(C_{sij} B_{sij} \alpha_{ij}). \quad (5.22)$$

Tyre model peak, stiffness, and shape factor

The lateral peak factors, $D_{sij}$, are defined as

$$D_{sij} = \mu_{sp,ij} F_{zij}, \quad (5.23)$$

and the lateral peak friction coefficients, $\mu_{sp,ij}$, are modelled in two ways:

C.i Constant peak friction coefficients for the linear peak factors,

$$\mu_{sp,ij} = p_{Dy1} \lambda_{\mu}, \quad (5.24)$$

C.ii and varying peak friction coefficients for the non-linear peak factors,

$$\mu_{sp,ij} = (p_{Dy1} + p_{Dy2} df_{zij}) \lambda_{\mu}, \quad (5.25)$$

with constants, $p_{Dy1}$ and $p_{Dy2}$, and scaling factor, $\lambda_{\mu}$,

$$\lambda_{\mu} = \frac{\hat{\mu}}{\mu_m}, \quad (5.26)$$

defined by the estimated tyre-road friction coefficient, $\hat{\mu}$, and the tyre-road friction coefficient in tyre testing conditions, $\mu_m$. The normalised changes in vertical load,


\[ df_{zij} = \frac{F_{zij} - F_{z0ij}}{F_{z0ij}}, \quad (5.27) \]

with nominal tyre loads, \( F_{z0ij} \).

The lateral shape factors, \( C_{sij} \), are constant and defined by the parameter, \( p_{Cy1} \):

\[ C_{sij} = p_{Cy1}. \quad (5.28) \]

The lateral stiffness factors, \( B_{sij} \), are defined by

\[ B_{sij} = \frac{K_{s\alpha ij}}{C_{sij} D_{sij}}, \quad (5.29) \]

with two different formulations of the cornering stiffnesses, \( K_{s\alpha ij} \):

C.iii Linear cornering stiffnesses,

\[ K_{s\alpha ij} = p_{Ky1} F_{zij}, \quad (5.30) \]

C.iv and non-linear cornering stiffnesses, based on a quadratic approximation of the formulation in the Magic Formula,

\[ K_{s\alpha ij} = \frac{p_{Ky1}}{p_{Ky2}} \left( 2 - \frac{F_{zij}}{p_{Ky2} F_{z0ij}} \right) F_{zij}, \quad (5.31) \]

using the constants, \( p_{Ky1} \) and \( p_{Ky2} \).

**Lateral and longitudinal tyre force coupling**

For the coupling of lateral and longitudinal tyre forces, three different approaches are defined:

D.i No coupling of the tyre forces,

\[ F_{sij} = F_{s0ij}, \quad (5.32) \]

means that the tyre forces in combined slip conditions are equivalent to the tyre forces in pure lateral slip conditions.

D.ii The linear coupling of the tyre forces,

\[ F_{sij} = F_{s0ij} \left( 1 + C_{lsij} \frac{F_{lij}}{D_{lij}} \right), \quad (5.33) \]

approximates the interaction between longitudinal and lateral tyre forces in combined slip conditions with a linear approximation of the tyre friction envelope. This means
that each lateral tyre force in combined slip, $F_{sij}$, is reduced by a factor, which depends on the current estimated longitudinal tyre force, $F_{lij}$ ($\leq 0$ in braking), divided by its maximum value, $D_{lij}$, in pure longitudinal slip conditions.

D.iii The non-linear coupling of the tyre forces,

$$F_{sij} = F_{s0ij} \cos \left( C_{lsij} \frac{F_{lij}}{D_{lij}} \frac{\pi}{2} \right), \quad (5.34)$$

gives a more accurate approximation of the tyre friction envelope for higher values of estimated longitudinal tyre forces.

The maximum longitudinal tyre forces, $D_{lij}$,

$$D_{lij} = \mu_{lpij} F_{zij}, \quad (5.35)$$

are based on the vertical tyre loads and constant longitudinal peak friction coefficients, $\mu_{lpij}$, with constant, $p_{Dx1}$,

$$\mu_{lpij} = p_{Dx1} \lambda_{\mu}. \quad (5.36)$$

The constants $C_{lsij}$, in (5.33) and (5.34), are used as tuning parameters.

**Tyre slip angles**

The computation of the tyre slip angles, $\alpha_{ij}$, neglects the track width of the vehicle leading to $\alpha_F = \alpha_{FL} = \alpha_{FR}$ and $\alpha_R = \alpha_{RL} = \alpha_{RR}$, respectively, and assumes small vehicle side-slip angles,

$$\alpha_F(t) = \delta(t_k) + \beta_F(t) = \delta(t_k) - \beta(t) - \frac{\dot{\psi}(t)}{v(t_k)} l_F, \quad (5.37)$$

$$\alpha_R(t) = -\beta_R(t) = -\beta(t) + \frac{\dot{\psi}(t)}{v(t_k)} l_R. \quad (5.38)$$

In (5.37) and (5.38), $\beta_F$ and $\beta_R$ are the vehicle side-slip angles on the front and rear axle.

**Model parameters**

For the identification of the different prediction model parameters least squares regressions are performed. Table 5.1 includes a selection of the prediction model parameters, representative of a sport-utility vehicle (SUV).

In the formulations C.i - C.iv, the identified parameters are $p_{Dy1}$ and $p_{Dy2}$ for $D_{sij}$, $p_{Cy1}$ for $C_{sij}$, and $p_{Ky1}$ and $p_{Ky2}$ for $B_{sij}$. The lateral tyre forces defined by the different tyre models B.i - B.iii are fitted against the full Pacejka Magic Formula in pure lateral
Table 5.1: Main parameters of the prediction model.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>Vehicle mass</td>
<td>1962 kg</td>
</tr>
<tr>
<td>$I_z$</td>
<td>Vehicle yaw mass moment of inertia</td>
<td>3382 kg m$^2$</td>
</tr>
<tr>
<td>$l_F$</td>
<td>Front semi-wheelbase</td>
<td>1.10 m</td>
</tr>
<tr>
<td>$l_R$</td>
<td>Rear semi-wheelbase</td>
<td>1.57 m</td>
</tr>
<tr>
<td>$d$</td>
<td>Front and rear track width</td>
<td>1.62 m</td>
</tr>
<tr>
<td>$h$</td>
<td>Height of vehicle centre of gravity</td>
<td>0.56 m</td>
</tr>
<tr>
<td>$h_F$</td>
<td>Front suspension roll centre height</td>
<td>0.26 m</td>
</tr>
<tr>
<td>$h_R$</td>
<td>Rear suspension roll centre height</td>
<td>0.29 m</td>
</tr>
<tr>
<td>$K_F$</td>
<td>Front suspension roll stiffness</td>
<td>44 750 N m/rad</td>
</tr>
<tr>
<td>$K_R$</td>
<td>Rear suspension roll stiffness</td>
<td>39 290 N m/rad</td>
</tr>
<tr>
<td>$\mu_m$</td>
<td>Friction coefficient in tyre testing conditions</td>
<td>1.0</td>
</tr>
</tbody>
</table>

slip conditions. The tyre slip angle $\alpha$ is varied in the range $-25\text{deg}$ to $25\text{deg}$ with an increment of $0.1\text{deg}$ and the vertical tyre force $F_z$ in the range 1 kN to 9 kN with an increment of 0.5 kN. The lateral force error is weighted based on the probability density function of the tyre slip angles in a series of Sine with Dwell tests, defined in Section 5.3.2.

Based on these identified values, the parameters $p_{D_{x1}}$ for $D_{ij}$, and $C_{ls_{ij}}$ are determined for the formulations D.ii and D.iii. For the regression analysis, the Magic Formula is employed in combined slip conditions. The vertical tyre force $F_z$ is varied in the range 2 kN to 8 kN with an increment of 2 kN, the tyre slip angle $\alpha$ in the range $-8\text{deg}$ to $8\text{deg}$ with an increment of 1 deg, the longitudinal tyre force in the range $-6.5\text{kN}$ to 0 kN with an increment of 50 N for the simple models, and the longitudinal slip in the range $-20\%$ to 0 % for the Pacejka Magic Formula with an increment of 0.25 % up to the approximate peak of 10 %, followed by an increment of 1 % for the remaining interval.

As an illustration, the tyre models in pure lateral slip conditions are demonstrated in Fig. 5.2, i.e. the linear tyre model B.i in sub-plot (a), the simplified Magic Formula B.ii in sub-plot (b), and the saturation model B.iii in sub-plot (c), both using formulations C.ii and C.iv.

For the influence of the peak and stiffness factors, the simplified model of the Pacejka Magic Formula is plotted in Fig. 5.3: In sub-plot (a), the peak and stiffness factors are linear; in sub-plot (b), the peak factor is non-linear and the stiffness factor is linear; in sub-plot (c), the peak factor is linear and the stiffness factor is non-linear; and in sub-plot (d), the peak and stiffness factors are non-linear.

The simple models for the coupling of longitudinal and lateral tyre forces are shown in Fig. 5.4. The comparison of the linear formulation D.ii in sub-plot (a) with the non-linear formulation D.iii in sub-plot (b) is demonstrated. The dashed lines indicate
Fig. 5.2: Illustration of the different tyre models in pure lateral slip conditions. The coloured lines represent the simplified tyre models and the black lines the full Pacejka Magic Formula.

the maximum lateral and longitudinal forces in pure slip conditions calculated by the factors $D_{lij}$ according to (5.35) and $D_{sij}$ according to (5.23) and (5.25), respectively.

In the formulations A.ii and A.iii for the estimation of the vertical tyre forces, the parameters $D_{s1}, C_{s1},$ and $B_{s1}$ are fitted against the full Pacejka Magic Formula tyre model in pure lateral slip conditions.

Based on these parameters, $K_F, K_R,$ and $h$ are fitted in a manoeuvre with ramp steering and straight-line braking minimising the vertical tyre force error. The test is carried out in conditions with a tyre-road friction coefficient $\mu = 0.8$, a speed of $v = 80$ km/h, and a steering wheel angle rate $\dot{\delta}_{sw} = 3$ deg/s.

In the same manoeuvre without braking, the relationship between steering wheel angle $\delta_{sw}$ and steering angle $\delta$ on the road is determined based on the average steering angle on the front axle. This approach accounts for the varying steering ratio and the compliance of the steering system in a practical manner.
5.2 Control system design

Fig. 5.3: Simplified model of the Pacejka Magic Formula in pure lateral slip conditions with different formulations of the peak and stiffness factors. The vertical tyre forces and the representation of the lines are equivalent to Fig. 5.2.

Fig. 5.4: Simplified model of the Magic Formula with vertical tyre force $F_z = 6 \text{kN}$ in combined slip conditions with different formulations of the longitudinal and lateral tyre force coupling.
5.2.2 Optimal control problem

The formulation of the non-linear model predictive vehicle stability controller is based on an optimal control problem defined by a cost function and constraints, in addition to the vehicle model equations in Section 5.2.1.

Cost function

The cost function, \( V(t_k) \), of the optimal control problem is defined as the integral of optimality criteria over the prediction horizon. More specifically, the longitudinal tyre force distribution imposed by the stability controller minimises the weighted deviation from reference targets for i) the overall vehicle traction or braking force, \( F_x(t) \), which tracks \( F_{x,\text{ref}}(t_k) \) demanded by the driver; ii) the overall vehicle direct yaw moment, \( M_{z,F_l}(t) \), which tracks \( M_{z,F_l,\text{ref}}(t_k) \), i.e. the reference direct yaw moment to be generated by the longitudinal tyre forces on the left and right sides of the vehicle; iii) the braking ratio, \( b(t) \), which tracks \( b_{\text{ref}} \), i.e. the desired ratio between the front and total longitudinal tyre forces within the considered vehicle side; iv) the yaw velocity error, \( \dot{\psi}(t) \), with violation of its bounds minimised by the slack variable \( N_{\dot{\psi}}(t) \); v) the average slip angle on the rear axle, \( \alpha_R(t) \), with violation of its bounds minimised by the slack variable \( N_{\alpha_R}(t) \); vi) the violation of the vehicle braking force bounds minimised by the slack variable \( N_{F_l}(t_k) \); and vii) the violation of the longitudinal tyre force bounds minimised by the slack variable \( N_{F_x}(t_k) \). The slack variables are discussed in more detail in the Section Constraints. The objective function consists of a least-squares type Lagrange term and a Mayer term is not considered:

\[
V(t_k) = \int_{t_k}^{t_f} \left[ \frac{r_{u,F_x}}{U_{sc,F_x}} \left( F_{x,\text{ref}}(t_k) - F_x(t) \right)^2 \\
+ \frac{r_{u,M_z}}{U_{sc,M_z}} \left( M_{z,F_l,\text{ref}}(t_k) - M_{z,F_l}(t) \right)^2 \\
+ \frac{r_{u,b}}{U_{sc,b}} \left( (1 - b_{\text{ref}}) F_{l,F_l}(t) - b_{\text{ref}} F_{l,RL}(t) \right)^2 \\
+ \frac{r_{u,b}}{U_{sc,b}} \left( (1 - b_{\text{ref}}) F_{l,F_R}(t) - b_{\text{ref}} F_{l,RR}(t) \right)^2 \\
+ \frac{q_{\nu,e_{\psi}}}{N_{sc,e_{\psi}}} N_{e_{\psi}}(t)^2 + \frac{q_{\nu,\alpha_R}}{N_{sc,\alpha_R}} N_{\alpha_R}(t)^2 \\
+ \frac{q_{\nu,F_x}}{N_{sc,F_x}} N_{F_x}(t_k)^2 + \frac{q_{\nu,F_l}}{N_{sc,F_l}} N_{F_l}(t_k)^2 \right] dt.
\]

(5.39)

In the cost function, the overall braking force is estimated as

\[
F_x(t) = F_{l,F_L}(t) + F_{l,F_R}(t) + F_{l,RL}(t) + F_{l,RR}(t),
\]

(5.40)
and the direct yaw moment generated by the longitudinal tyre forces is approximated assuming small steering angles,

\[ M_{z,F_l}(t) = \left( F_{FL}(t) + F_{FR}(t) \right) \frac{d}{2} - \left( F_{FL}(t) + F_{RL}(t) \right) \frac{d}{2}. \]

(5.41)

The prediction horizon is the interval between the current time, \( t_k \), and the time at the end of the prediction horizon, \( t_f = t_k + N_p h \), defined by the number of prediction steps, \( N_p \), and the discretisation time, \( h \). The longitudinal tyre forces \( F_{FL}(t) \), \( F_{FR}(t) \), \( F_{RL}(t) \), and \( F_{RR}(t) \) are the four control inputs to the prediction model. They can vary \( N_c \) times over the prediction horizon, where \( N_c \) is the number of control steps, and then are kept constant from \( t_k + (N_c - 1) h \) until \( t_f \).

The contribution of the different terms of \( V(t_k) \) are weighted with the coefficients \( r_{u,F_x}, r_{u,M_z}, r_{u,b}, r_{q,v,e}, r_{q,v,F_\psi}, \) and \( q_{v,F_l} \). Appropriate scaling factors, i.e. \( U_{sc,F_x}, U_{sc,M_z}, U_{sc,b}, N_{sc,e}, N_{sc,ar}, N_{sc,F_x}, \) and \( N_{sc,F_l} \), allow an equivalent influence of the various weights. The settings of the optimal control problem are discussed in the Section Settings. The cost function formulation in (5.39) permits very different operating principles of the stability controller determined by the tuning choices, e.g. focused on both yaw velocity tracking and slip angle constraints, or the yaw velocity tracking only, or the control allocation of the individual braking forces. The flexibility of this novel cost function formulation meets the diversified performance requirements of real vehicle implementations.

**Reference generation**

The reference vehicle braking force, \( F_{x,ref} \), is obtained from the demanded deceleration, \( a_{x,ref} \), of the vehicle. This is calculated from the driver input on the accelerator and brake pedals,

\[ F_{x,ref}(t) = m a_{x,ref}(t_k). \]

(5.42)

The reference direct yaw moment, \( M_{z,F_l,ref} \), is calculated based on two different approaches:

E.i Minimise the direct yaw moment generated by the longitudinal tyre forces with a zero reference,

\[ M_{z,F_l,ref}(t) = 0. \]

(5.43)

E.ii Track the direct yaw moment reference based on proportional contributions, \( M_{z,F_l,ref(e_\psi)} \) and \( M_{z,F_l,ref(\alpha_R)} \), dependent on the yaw velocity error and the rear slip angle, respectively, with gains \( K_{e_\psi} \) and \( K_{\alpha_R} \),

\[ M_{z,F_l,ref}(t) = K_{e_\psi} M_{z,F_l,ref(e_\psi)}(t_k) - K_{\alpha_R} M_{z,F_l,ref(\alpha_R)}(t_k). \]

(5.44)
The contributions are calculated based on the violation of the bounds on the yaw velocity error and the rear slip angle at $t_k$, described by the slack variables $N_{e_\psi}$ and $N_{\alpha_R}$, respectively. For appropriate signs, the hyperbolic tangent terms with the yaw velocity error and rear slip angle related to their bounds, $e_{\psi,max}$ and $\alpha_{R,max}$, and the factor 5 to ensure saturation, approximate the signum function,

$$M_{z,F,ref(e_\psi)} = U_{sc,M} \tanh\left(5 \frac{e_{\psi}(t_k)}{e_{\psi,max}} \right) \frac{N_{e_\psi}(t_k)}{N_{sc,e_\psi}}. \quad (5.45)$$

$$M_{z,F,ref(\alpha_R)} = U_{sc,M} \tanh\left(5 \frac{\alpha_R(t_k)}{\alpha_{R,max}} \right) \frac{N_{\alpha_R}(t_k)}{N_{sc,\alpha_R}}. \quad (5.46)$$

The influence of the direct yaw moment reference is discussed in more detail in Section 5.3.5.

The reference braking ratio, $b_{ref}$, is considered constant in the implementation of the controller.

For the desired handling of the controlled vehicle, a non-linear under-steer characteristic is defined. It is based on the expression,

$$\dot{\psi}_{ref,lin}(t) = \frac{v(t_k)}{l_F + l_R + \frac{K_{us}}{g} v(t_k)^2} \delta(t_k), \quad (5.47)$$

for the linear yaw velocity reference, $\dot{\psi}_{ref,lin}$, with gradient $K_{us}$ related to gravity. Both vehicle speed and steering angle are assumed constant over the prediction horizon. Taking into account the physical limitations of the vehicle, the maximum achievable yaw velocity, $\dot{\psi}_{max}$, can be approximated in steady-state,

$$\dot{\psi}_{max} = \frac{a_{y,max}}{v}, \quad (5.48)$$

based on the maximum lateral acceleration, $a_{y,max}$, defined with constants, $p_{DY1}$ and $p_{DY2}$, as

$$a_{y,max} = \left( p_{DY1} \lambda + p_{DY2} \right) g, \quad (5.49)$$

dependent on the tyre-road friction condition. For a smooth transition between the linear and maximum yaw velocity, the use of an Ansatz comparable to the Pacejka Magic Formula (Pacejka, 2012) leads to a non-linear reference yaw velocity, $\dot{\psi}_{ref}$,

$$\dot{\psi}_{ref}(t) = D_r \dot{\psi}_{max}(t_k) \sin\left(C_r \arctan\left(B_r \frac{\dot{\psi}_{ref,lin}(t_k)}{\dot{\psi}_{max}(t_k)}\right)\right), \quad (5.50)$$
very similar to the yaw velocity associated with the under-steer characteristics of the passive vehicle. The Ansatz with tuning parameters $K_{us}$, $D_r$, $C_r$, and $B_r$ allows for a degressive yaw velocity characteristics beyond the yaw velocity peak for high values of the steering angle.

Based on the identified value of the linear reference under-steer gradient $K_{us}$, the parameters $p_{DY1}$, $p_{DY2}$, $D_r$, $C_r$, and $B_r$ are calculated minimising the yaw velocity error in a least squares fashion in ramp steer manoeuvres with different tyre-road friction coefficients. The tests are performed with a speed of $v = 80$ km/h and a steering wheel angle rate $\dot{\delta}_{sw} = 3$ deg/s. Starting from the nominal value $\mu = 0.8$, the tyre-road friction coefficient is varied with values $\mu = 0.4$, $0.6$, $0.8$, $1.0$, and $1.2$.

In Fig 5.5, the yaw velocity of the passive vehicle is shown in red lines for the ramp steer manoeuvre in different friction conditions. In blue solid lines, the non-linear yaw velocity reference, as defined in (5.50), is plotted and in blue dashed lines the linear yaw velocity reference, as defined in (5.47), is shown.

![Fig. 5.5: Steady-state yaw velocity in varying tyre-road friction conditions.](image)

In Fig 5.6, the yaw velocity reference is shown over the vehicle speed $v$ and steering angle $\delta$ for a tyre-road friction coefficient $\mu = 0.8$.

As a final note, it should be said that the tyre-road friction coefficient in the formulation of the reference yaw velocity and also in the prediction model is assumed to be a constant. However, the formulations are chosen such that an extension of the optimal control problem with an estimation of the tyre-road friction coefficient would be straight-forward.
Fig. 5.6: Reference yaw velocity as a function of the vehicle speed $v$ and the steering angle $\delta$.

**Constraints**

Soft constraints are used on the yaw velocity error, $e_\psi = \dot{\psi}_{\text{ref}} - \dot{\psi}$, with maximum and minimum bounds, $e_{\psi,\text{max}} = -e_{\psi,\text{min}}$,

$$e_{\psi,\text{min}} - N_{e_\psi}(t) \leq \dot{\psi}_{\text{ref}}(t_k) - \dot{\psi}(t) \leq e_{\psi,\text{max}} + N_{e_\psi}(t).$$  \hspace{1cm} (5.51)

Additional constraints are beneficial to vehicle stability for a wide range of driving conditions. For example, soft constraints are adopted on the rear slip angle, as in Beal and Gerdes (2013), with maximum and minimum bounds, $\alpha_{R,\text{max}} = -\alpha_{R,\text{min}}$,

$$\alpha_{R,\text{min}} - N_{\alpha_R}(t) \leq \alpha_R(t) \leq \alpha_{R,\text{max}} + N_{\alpha_R}(t).$$  \hspace{1cm} (5.52)

In contrary to constraints on the vehicle side-slip angle, commonly used in the literature, the formulation based on the rear slip angle does not include a kinematic contribution, as elaborated in Lenzo et al. (2017), and scales naturally with the vehicle speed, as discussed in Beal and Gerdes (2013).

As a consequence of these constraints, the controller generates a direct yaw moment only when the yaw velocity error or the rear slip angle over the prediction horizon exceeds the thresholds defined by the bounds. This is indicated by a positive value of the slack variables $N_{e_\psi}(t)$ and $N_{\alpha_R}(t)$, respectively.
Additionally, the longitudinal tyre forces are constrained to be negative, since only braking torques can be applied by the friction brakes,

\[-k_s D_{l_{ij}}(t) - N_{F_i}(t_k) \leq F_{l_{ij}}(t) \leq 0.\] (5.53)

The lower bounds on the longitudinal tyre forces are determined by the estimated maximum values in pure longitudinal slip conditions, as defined in (5.35), reduced by a tuning factor \(k_s \leq 1\). The soft constraints with the constant slack variable, \(N_{F_i}(t_k)\), for all tyre forces allow a certain violation of the lower bounds.

Together with the soft constraint on the overall braking force,

\[F_x(t) \leq F_{x,ref}(t_k) + N_{F_x}(t_k),\] (5.54)

with constant slack variable, \(N_{F_x}(t_k)\), the constraints (5.53) and (5.54) allow to vary the importance of direct yaw moment generation and braking force generation in limit cornering conditions.

The slack variables represent the violation of the respective bounds and are constrained to be positive semi-definite,

\[N_{e_{\psi}}(t) \geq 0,\] (5.55)
\[N_{\alpha_R}(t) \geq 0,\] (5.56)
\[N_{F_x}(t_k) \geq 0,\] (5.57)
\[N_{F_l}(t_k) \geq 0.\] (5.58)

**State transformation**

For the optimisation problem, the vehicle side-slip angle, \(\beta\), and the yaw velocity, \(\dot{\psi}\), at the current time \(t_k\), are required as initial conditions for the dynamic equality constraints (5.1) and (5.2). The vehicle speed, \(v(t_k)\), the steering angle, \(\delta(t_k)\), and the requested deceleration, \(a_{x,ref}(t_k)\), are considered as varying parameters.

In the optimal control problem, the yaw velocity error, \(e_{\psi} = \dot{\psi}_{ref} - \dot{\psi}\), and the rear slip angle, \(\alpha_R\), are constrained in (5.51) and (5.52), respectively. Therefore, a formulation of the optimal control problem with the yaw velocity error, \(e_{\psi}(t)\), and the rear slip angle, \(\alpha_R(t)\), instead of the side-slip angle, \(\beta(t)\), and the yaw velocity, \(\dot{\psi}(t)\), turned out to be beneficial. This is due to the fact that the state-space exploration and splitting strategy in the chosen explicit approach to non-linear model predictive control is based on orthogonal partitions. With the description based on the new variables, employing
the transformation laws,
\[ e_{\dot{\psi}}(t) = \dot{\psi}_{\text{ref}}(t_k) - \dot{\psi}(t), \]
\[ \alpha_R(t) = -\beta(t) + \frac{\dot{\psi}(t) l_R}{v(t_k)}, \]
using (5.50) and (5.38), respectively, the constrained variables are aligned with the coordinate axes of the orthogonal exploration space.

**Settings**

Table 5.2 reports the settings for the optimal control problem of this study.

The main objective of the controller is to reduce the violation of the yaw velocity error and rear slip angle bounds by providing the demanded vehicle deceleration and limiting the individual longitudinal tyre forces at the same time. The weights \( q_{\nu,e} \dot{\psi} \) and \( q_{\nu,\alpha_R} \) are chosen to give priority to the rear slip angle compared to the yaw velocity error, i.e. the vehicle stability is more important than the yaw velocity tracking. Moreover, the weights \( q_{\nu,F_x} \) and \( q_{\nu,F_l} \) are chosen to allow direct yaw moment generation compromising the overall braking demand in limit cornering conditions. The choice of the prediction and control horizon in Table 5.2 is motivated by the explicit approach where the complexity of the controller grows with an increasing number of prediction and control steps.

**Multi-parametric non-linear program**

The optimal control problem consists of the cost function (5.39), the equality constraints (5.1) and (5.2), and the inequality constraints (5.51)-(5.58) including all necessary definitions in Section 5.2.1 and 5.2.2. The problem has the longitudinal tyre forces \( F_{iFL}(t), F_{iFR}(t), F_{iRL}(t), \) and \( F_{iRR}(t) \) and the slack variables \( N_{e_{\dot{\psi}}}(t), N_{\alpha_R}(t), N_{F_x}(t_k), \) and \( N_{F_l}(t_k) \) as optimisation variables in its continuous formulation. The general optimal control problem is reformulated into a form suitable for the numerical solution using the so-called direct methods.

According to the simultaneous approach, as introduced in Section 3.3.2 in detail, the ordinary differential equation constraints (5.1) and (5.2) are parametrised and discretised in time. The resulting algebraic equations are treated as additional non-linear equality constraints. The forward Euler method, i.e. an explicit first-order Runge Kutta method with one stage, is applied as a numerical integration scheme. The trajectories of the control inputs \( u(t) = [F_{iFL}(t)/U_{ch,F_l}, F_{iFR}(t)/U_{ch,F_l}, F_{iRL}(t)/U_{ch,F_l}, F_{iRR}(t)/U_{ch,F_l}]^T \) and slack variables \( \nu(t) = [N_{e_{\dot{\psi}}}(t)/N_{ch,e_{\dot{\psi}}}, N_{\alpha_R}(t)/N_{ch,\alpha_R}, N_{F_x}(t_k)/N_{ch,F_x}, N_{F_l}(t_k)/N_{ch,F_l}]^T \) are parametrised by the vectors \( U = [u(t_k)^T, \ldots, u(t_k+N_p-1)^T]^T \) and \( \nu = [\nu(t_k)^T, \ldots, \nu(t_k+N_p)^T]^T, \) respectively, which are considered as unknown variables. In addition,
Table 5.2: Settings of the optimal control problem.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Setting A</th>
<th>Setting B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>Discretisation time</td>
<td>20 ms</td>
<td></td>
</tr>
<tr>
<td>$T_s$</td>
<td>Sampling time</td>
<td>20 ms</td>
<td></td>
</tr>
<tr>
<td>$N_p$</td>
<td>Prediction steps</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$N_c$</td>
<td>Control steps</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$K_{\psi}$</td>
<td>Yaw velocity error direct yaw moment gain</td>
<td>0</td>
<td>0.30</td>
</tr>
<tr>
<td>$K_{\alpha R}$</td>
<td>Rear slip angle direct yaw moment gain</td>
<td>0</td>
<td>0.32</td>
</tr>
<tr>
<td>$b_{\text{ref}}$</td>
<td>Reference braking ratio</td>
<td></td>
<td>0.75</td>
</tr>
<tr>
<td>$r_{u,F_x}$</td>
<td>Weight on traction force tracking</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$r_{u,M_z}$</td>
<td>Weight on direct yaw moment tracking</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$r_{u,b}$</td>
<td>Weight on braking ratio tracking</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>$q_{\nu,e,\psi}$</td>
<td>Weight on yaw velocity error violation</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>$q_{\nu,\alpha R}$</td>
<td>Weight on rear slip angle violation</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>$q_{\nu,F_x}$</td>
<td>Weight on traction force violation</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>$q_{\nu,F_l}$</td>
<td>Weight on tyre force violation</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$U_{sc,F_x}$</td>
<td>Scaling factor for traction force</td>
<td>6000 N</td>
<td></td>
</tr>
<tr>
<td>$U_{sc,M_z}$</td>
<td>Scaling factor for direct yaw moment</td>
<td>6000 N m</td>
<td></td>
</tr>
<tr>
<td>$U_{sc,b}$</td>
<td>Scaling factor for braking ratio</td>
<td>1000 N</td>
<td></td>
</tr>
<tr>
<td>$N_{sc,e,\psi}$</td>
<td>Scaling factor for yaw velocity error</td>
<td>10 deg/s</td>
<td></td>
</tr>
<tr>
<td>$N_{sc,\alpha R}$</td>
<td>Scaling factor for rear slip angle</td>
<td>1 deg</td>
<td></td>
</tr>
<tr>
<td>$N_{sc,F_x}$</td>
<td>Scaling factor for traction force violation</td>
<td>300 N</td>
<td></td>
</tr>
<tr>
<td>$N_{sc,F_l}$</td>
<td>Scaling factor for tyre force violation</td>
<td>300 N</td>
<td></td>
</tr>
<tr>
<td>$e_{\psi,max}$</td>
<td>Bound on yaw velocity error</td>
<td>5.0 deg/s</td>
<td></td>
</tr>
<tr>
<td>$\alpha_{\text{R, max}}$</td>
<td>Bound on rear slip angle</td>
<td>5.5 deg</td>
<td></td>
</tr>
<tr>
<td>$k_s$</td>
<td>Safety factor for tyre force bounds</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>$X_{ch,e,\psi}$</td>
<td>Characteristic yaw velocity error</td>
<td>15 deg/s</td>
<td></td>
</tr>
<tr>
<td>$X_{ch,\alpha R}$</td>
<td>Characteristic rear slip angle</td>
<td>5 deg</td>
<td></td>
</tr>
<tr>
<td>$P_{ch,v}$</td>
<td>Characteristic vehicle speed</td>
<td>80 km/h</td>
<td></td>
</tr>
<tr>
<td>$P_{ch,\delta}$</td>
<td>Characteristic steering angle</td>
<td>8 deg</td>
<td></td>
</tr>
<tr>
<td>$P_{ch,\alpha_{\text{ref}}}$</td>
<td>Characteristic vehicle acceleration</td>
<td>3 m/s²</td>
<td></td>
</tr>
<tr>
<td>$U_{ch,F_l}$</td>
<td>Characteristic longitudinal tyre force</td>
<td>4800 N</td>
<td></td>
</tr>
<tr>
<td>$N_{ch,e,\psi}$</td>
<td>Characteristic yaw velocity error violation</td>
<td>10 deg/s</td>
<td></td>
</tr>
<tr>
<td>$N_{ch,\alpha R}$</td>
<td>Characteristic rear slip angle violation</td>
<td>1 deg</td>
<td></td>
</tr>
<tr>
<td>$N_{ch,F_x}$</td>
<td>Characteristic traction force violation</td>
<td>300 N</td>
<td></td>
</tr>
<tr>
<td>$N_{ch,F_l}$</td>
<td>Characteristic tyre force violation</td>
<td>300 N</td>
<td></td>
</tr>
</tbody>
</table>
parameters of the state trajectory, described by \( \mathbf{x}(t) = [e_\dot{\psi}(t)/X_{ch,e}, \alpha_R(t)/X_{ch,\alpha R}]^T \), are summarised in the vector \( \mathbf{X} = [\mathbf{x}(t_{k+1})^T, \ldots, \mathbf{x}(t_{k+N_p})^T]^T \), representing the intermediate predicted states, and are treated as additional unknown variables. For a numerically well-conditioned optimisation problem, a non-dimensionalisation procedure, as introduced in Section 3.2.2, is adopted for all variables with appropriate choices for the characteristic quantities, e.g. \( U_{ch,F}, N_{ch,e}, N_{ch,\dot{\psi}, R}, N_{ch,Fz}, X_{ch,e}, X_{ch,\alpha R}, \) etc. The integral in the cost function (5.39) is approximated by a finite sum by applying a numerical integration method. The continuous inequality constraints (5.51)-(5.58) are parametrised and relaxed to hold only at some discrete time instants, \( \{t_k, \ldots, t_{k+N_p}\} \subset [t_k, t_f] \), leading to a finite number of discrete and parametrised constraint functions.

By implementing the finite parametrisation and discretisation on the continuous cost function and constraint functions, and by applying the integration scheme to the ordinary differential equations (ODEs), the following optimisation problem, defined by the parametrised and discretised cost function, \( V \), and constraint functions, \( G \), is obtained as derived in Section 3.4,

\[
V^*(\mathbf{x}_p(t_k)) = \min_z V(z, \mathbf{x}_p(t_k)),
\]

subject to,

\[
G(z, \mathbf{x}_p(t_k)) \leq 0.
\]

The vector of parameters, \( \mathbf{x}_p(t_k) \), combines the initial states of the prediction model, \( \mathbf{x}(t_k) \), and the vector of varying parameters, \( \mathbf{p}(t_k) \). This leads to \( \mathbf{x}_p(t_k) = [e_\dot{\psi}(t_k)/X_{ch,e}, \alpha_R(t_k)/X_{ch,\alpha R}, v(t_k)/P_{ch,v}, \delta(t_k)/P_{ch,\delta}, a_{x,ref}(t_k)/P_{ch,a_{x,ref}}]^T \) as the vector of parameters. Similarly, the vector of optimisation variables, \( \mathbf{z} \), combines the vector of control trajectory parameters, \( \mathbf{U} \), the slack variable trajectory parameters, \( \mathbf{N} \), and the state trajectory parameters, \( \mathbf{X} \), leading to \( \mathbf{z} = [\mathbf{U}^T, \mathbf{N}^T, \mathbf{X}^T]^T \).

Assuming that it exists, the optimal solution of the receding horizon MPC strategy at \( t_k \), denoted as \( \mathbf{z}^* \), represents the optimal open-loop trajectory over the prediction horizon. The longitudinal tyre forces, \( F_{l_{ij}}(t_k) \), are extracted from the solution and converted to reference hydraulic pressures for the electro-hydraulic braking system. At the next time instant, after the sampling time, \( t_s \), of the controller has elapsed, the optimisation problem is solved with the updated parameter vector, making the MPC strategy a closed-loop approach.

**Explicit non-linear MPC**

The general formulation (5.61)-(5.62) can be considered as a non-linear programming problem for a fixed parameter \( \mathbf{x}_p(t_k) \). Since the online solution of an NLP poses significant challenges for real-time implementation due to the involved non-linearities,
5.3 Performance analysis

an explicit solution is investigated in this application for vehicle stability control. The optimisation problem (5.61)-(5.62) is, therefore, considered as a multi-parametric non-linear programming problem with $x_p(t_k)$ as the parameter, i.e. for which the optimal solution, $z^*$, has to be found over a range of values of the parameter $x_p$. However, for mp-NLP problems, it is in general not possible to derive an explicit solution in an exact form. Therefore, the explicit NMPC algorithm 1, as proposed in Section 4.3, using multi-parametric quadratic programming approximations of the mp-NLP, is employed to derive a sub-optimal solution with guaranteed levels of sub-optimality.

The algorithm locally approximates the multi-parametric non-linear program with a multi-parametric quadratic program, leading to an explicit approximate solution of the mp-NLP (5.61)-(5.62), consisting of the solution of the mp-QP sub-problems over each orthogonal partition including piecewise affine solution functions and corresponding polyhedral critical regions. Therefore, the main computational effort is carried out offline. The online computation reduces to the identification of the orthogonal partition and polyhedral region for a given parameter vector, $x_p$, and the evaluation of the associated multi-variable affine feedback law.

**Complexity reduction and online implementation**

With an appropriate post-processing procedure, it is possible to considerably reduce the complexity of the explicit solution. The selected methods include a clipping-based complexity reduction, and disjoint optimal and sub-optimal merging procedures embedded in the proposed post-processing algorithm 4 in Section 4.4. Since the dynamic system (5.1) and (5.2) is point-symmetric with respect to the origin for a given speed, $v(t_k)$, and reference deceleration, $a_{x,ref}(t_k)$, the exploration space and, therefore, the complexity of the controller can be reduced by bisection discarding redundant information. Moreover, the state transformation in Section State transformation helps to further reduce the complexity of the explicit solution.

The post-processing generates and exports real-time executable code for the online evaluation of the explicit NMPC. The actual evaluation is a three-stage process and is explained in more detail in Section 4.4.4.

**5.3 Performance analysis**

In the following sections, the performance of the non-linear model predictive controller for vehicle stability is assessed in various conditions. The influence of the controller tuning, the variation in the prediction model complexity, and the formulation of constraints on the performance is investigated. Moreover, the influence of the sampling time and delays on the controller performance and the robustness of the controller against parameter
variations is assessed. Finally, the influence of the number of parameters and the definition of the parameter space on the complexity of the explicit controller is analysed.

5.3.1 Simulation model

The results are based on an experimentally validated high-fidelity IPG Automotive CarMaker simulation model of a case study SUV. In Fig. 5.7, the good agreement of the simulation results of the high-fidelity model with the experimental test results of the real vehicle demonstrator is shown for a step-steer manoeuvre in high friction conditions. The vehicle sideslip angle and speed is measured with an optical sensor positioned at the front end of the test vehicle.

![Graphs showing dynamic response](image)

Fig. 5.7: Comparison of the dynamic response of the uncontrolled vehicle in a step-steer manoeuvre with the high-fidelity simulation model and the real vehicle demonstrator. The superscripts $\text{exp}$ and $\text{sim}$ indicate if the respective quantity is measured in the experimental test with the vehicle demonstrator or calculated based on the high-fidelity simulation model.

The simulation model includes consideration of the electro-hydraulic braking system pressure dynamics at the callipers, based on variable transportation delays and first-order dynamics. The parameter identification is performed through experimental tests on a test bed for the electro-hydraulic braking system, described in Tavernini et al. (2019b) and Chapter 6 in more detail. With the knowledge of the conversion from calliper pressure
5.3 Performance analysis

to braking torque, the requested longitudinal tyre forces, $F_{ij}$, from the vehicle stability controller can be converted to reference brake pressures, $p_{b_i,ref}$. The pressure-to-torque coefficients on the front and rear axle, $K_{pT,F}$ and $K_{pT,R}$, respectively, are employed neglecting the wheel dynamics.

5.3.2 Test manoeuvre

The behaviour of the uncontrolled vehicle and the controlled vehicle with the NMPC-based stability controller is simulated along the Sine with Dwell test, as shown in Fig. 5.8.

The test conditions are explained in the regulation UN/ECE-R 13H (1958) and can be summarised as follows:

- Vehicle coasting in high gear at $80\pm2$ km/h.
- The quantity $A_{\delta_{sw}}$, defining the steering wheel angle amplitude, $\hat{\delta}_{sw}$, is the steering wheel angle, $\delta_{sw}$, at the lateral acceleration $a_y = 0.3\, g$, determined in a slowly increasing steering manoeuvre with a steering wheel angle rate $\dot{\delta}_{sw} = 13.5\, \text{deg/s}$ at $80\pm2\, \text{km/h}$.
- In each run of the Sine with Dwell test series, the steering wheel angle amplitude is increased from run to run by $0.5\, A_{\delta_{sw}}$, starting with $\dot{\delta}_{sw} = 1.5\, A_{\delta_{sw}}$ for the initial run.
- The steering wheel angle amplitude of the final run in each series is greater than $6.5\, A_{\delta_{sw}}$ or 270 deg with a maximum of 300 deg.

Fig. 5.8: Sine with Dwell test; figure adapted from Lutz et al. (2017).
• The steering wheel input is based on a sine wave with a frequency $\tau = 0.7\,\text{Hz}$ and a dwell of 500 ms beginning at the second peak amplitude, as shown in Fig. 5.8.

The quantity, $A_{\delta_{sw}} = 23.1\,\text{deg}$, for the SUV simulation model was determined according to UN/ECE-R 13H (1958). The under-steer characteristics of the uncontrolled vehicle determined in the ramp steer manoeuvre is shown in Fig. 5.9. The test series are performed on a surface with tyre-road friction coefficient $\mu = 0.8$, different from the coefficient $\mu = 0.9$ specified in the regulation to avoid roll-over of the simulation model without outriggers.

![Graph showing under-steer characteristics](image)

Fig. 5.9: Under-steer characteristics of the uncontrolled vehicle.

The directional or yaw stability criteria to be fulfilled are:

• The yaw velocity at time $T_{0+1}$, i.e. $1\,\text{s}$ after $T_0$, shall be $\leq 35\%$ of the yaw velocity peak, $\dot{\psi}_{\text{peak}}$.

• The yaw velocity at time $T_{0+1.75}$, i.e. $1.75\,\text{s}$ after $T_0$, shall be $\leq 20\%$ of the yaw velocity peak, $\dot{\psi}_{\text{peak}}$.

$\dot{\psi}_{\text{peak}}$ is defined as the first peak value of yaw velocity, recorded after the steering wheel angle changes sign, and $T_0$ is defined at the completion of the steering wheel input.

Moreover, the following responsiveness criterion must be met by vehicles with a mass $\leq 3500\,\text{kg}$, such as the one of this study:

• The lateral displacement of the vehicle centre of gravity with respect to its initial straight path must be $\geq 1.83\,\text{m}$ when computed at the beginning of the dwell in the steering wheel input for runs with a steering wheel angle amplitude $\hat{\delta}_{sw} \geq 5\,A_{\delta_{sw}}$. 
The performance indices $J_1$ and $J_2$ are introduced to quantify the directional stability criteria:

$$J_1 = \frac{\dot{\psi}(T_{0+1})}{\psi_{peak}} \times 100,$$

$$J_2 = \frac{\dot{\psi}(T_{0+1.75})}{\psi_{peak}} \times 100.$$

The performance index $J_3$ is introduced, as defined in UN/ECE-R 13H (1958), by double integration of the lateral acceleration at the vehicle centre of gravity,

$$J_3 = \int_0^{T_{dwell}} \int_0^{T_{dwell}} a_y \, dt \, dt.$$

$T_{dwell} = T_s + \frac{3}{4} \tau$ is the beginning of the dwell in the steering input and $T_s = 0.2$ s indicates the start of the steering action.

## 5.3.3 Performance indicators

To quantify the influence of different prediction models and tunings of the cost function on the vehicle behaviour with the NMPC controller, some performance indicators are introduced, which are defined over the simulation interval $[0, T]$ with $T = 4$ s. The time difference $\Delta T = T_d - T_a$ is used to normalise the indices, calculated for each individual test run of the Sine with Dwell test series. $T_a$ and $T_d$ are the first and last time instant, respectively, at which the change in vertical tyre forces $F_{z_{ij}}$ of the simulation model exceeds a user-defined threshold during the simulation interval. The superscripts $\text{sim}$ and $\text{est}$ indicate if the performance index is calculated based on the high-fidelity simulation model or based on the prediction model of the NMPC controller, respectively.

$$I_{1,ij}^{\text{sim}} = \frac{1}{\Delta T} \int_0^T |M_{z,F_{ij}}^{\text{sim}}(t)| \, dt,$$

$$I_{1,ij}^{\text{est}} = \frac{1}{\Delta T} \int_0^T |M_{z,F_{ij}}^{\text{est}}(t)| \, dt.$$

The indices $I_{1,ij}^{\text{sim}}$ and $I_{1,ij}^{\text{est}}$ are the average of absolute values of the yaw moment contribution on each corner of the simulation and prediction model, respectively. The yaw moment contribution, $M_{z,F_{ij}}$, is the yaw moment that is generated by the tyre on corner $ij$, i.e. by the lateral tyre force $F_{s_{ij}}$ and the longitudinal tyre force $F_{l_{ij}}$.

$$I_{2,ij} = \frac{1}{\Delta T} \int_0^T |M_{z,F_{ij}}^{\text{sim}}(t) - M_{z,F_{ij}}^{\text{est}}(t)| \, dt.$$
The index $I_{2,ij}$ is the average of absolute values of the yaw moment contribution error on corner $ij$ of the simulation model and the prediction model.

$$I_{3}^{\text{sim}} = \frac{1}{\Delta T} \int_{0}^{T} \left| M_{z}^{\text{sim}}(t) \right| dt,$$  \hfill (5.69)

$$I_{3}^{\text{est}} = \frac{1}{\Delta T} \int_{0}^{T} \left| M_{z}^{\text{est}}(t) \right| dt. \hfill (5.70)$$

The indices $I_{3}^{\text{sim}}$ and $I_{3}^{\text{est}}$ are the average of absolute values of the overall yaw moment, $M_{z}$, generated by the lateral and longitudinal tyre forces on every corner of the simulation and prediction model, respectively.

$$I_{4,a} = \sum_{ij} I_{2,ij} = \sum_{ij} \left( \frac{1}{\Delta T} \int_{0}^{T} \left| M_{z,F_{ij}}^{\text{sim}}(t) - M_{z,F_{ij}}^{\text{est}}(t) \right| dt \right), \hfill (5.71)$$

$$I_{4,b} = \frac{1}{\Delta T} \int_{0}^{T} \left| M_{z}^{\text{sim}}(t) - M_{z}^{\text{est}}(t) \right| dt. \hfill (5.72)$$

The index $I_{4,a}$ is the sum of the average of absolute yaw moment contribution errors, as defined by the indices $I_{2,ij}$ in (5.68). The index $I_{4,b}$ is the average of the absolute overall yaw moment error, i.e. simulated through the high-fidelity simulation model and estimated through the prediction model of the NMPC controller.

$$I_{5}^{\text{sim}} = \max_{t \in [0,T]} \left| M_{z,F_{i}}^{\text{sim}}(t) \right|, \hfill (5.73)$$

$$I_{5}^{\text{est}} = \max_{t \in [0,T]} \left| M_{z,F_{i}}^{\text{est}}(t) \right|. \hfill (5.74)$$

The indices $I_{5}^{\text{sim}}$ and $I_{5}^{\text{est}}$ are the maximum values of the absolute overall direct yaw moment, $M_{z,F_{i}}$, generated by the longitudinal tyre forces, $F_{i}$, on every corner of the simulation and prediction model, respectively. The average of the absolute overall direct yaw moment is defined by the indices $I_{6}^{\text{sim}}$ and $I_{6}^{\text{est}}$.

$$I_{6}^{\text{sim}} = \frac{1}{\Delta T} \int_{0}^{T} \left| M_{z,F_{i}}^{\text{sim}}(t) \right| dt, \hfill (5.75)$$

$$I_{6}^{\text{est}} = \frac{1}{\Delta T} \int_{0}^{T} \left| M_{z,F_{i}}^{\text{est}}(t) \right| dt. \hfill (5.76)$$

The index $I_{7}^{\text{sim}}$ (equal to $I_{7}^{\text{est}}$, since $e_{\dot{\psi}}^{\text{sim}} = e_{\dot{\psi}}^{\text{est}}$) is the average of the absolute value of the violation of the yaw velocity error bounds, described by the slack variable $N_{e_{\dot{\psi}}}(t_{k})$, as defined in (5.51),

$$I_{7}^{\text{sim}} = \frac{1}{\Delta T} \int_{0}^{T} \left( \left| e_{\dot{\psi}}(t) \right| - e_{\dot{\psi},\text{max}} \right) dt = \frac{1}{\Delta T} \int_{0}^{T} \left| N_{e_{\dot{\psi}}}(t_{k}) \right| dt. \hfill (5.77)$$
5.3 Performance analysis

Analogously, the indices $I_{8,a}^{\text{sim}}$, $I_{8,a}^{\text{est}}$, and $I_{8,b}^{\text{sim}}$ are defined as the average of the absolute value of the violation of the rear slip angle bounds. In (5.78) and (5.80), $\alpha_{R}^{\text{sim}}$ is the average slip angle and $\beta_{R}^{\text{sim}}$ the side-slip angle on the rear axle of the simulation model. In (5.79), $\alpha_{R}^{\text{est}}$ is based on (5.60) for the prediction model, and the violation can be described by the slack variable $N_{\alpha_{R}(t_{k})}$, as defined in (5.52),

$$I_{8,a}^{\text{sim}} = \frac{1}{\Delta T} \int_{0}^{T} \left( |\alpha_{R}^{\text{sim}}(t)| - \alpha_{R,max} \right) dt,$$

$$I_{8,a}^{\text{est}} = \frac{1}{\Delta T} \int_{0}^{T} \left( |\alpha_{R}^{\text{est}}(t)| - \alpha_{R,max} \right) dt = \frac{1}{\Delta T} \int_{0}^{T} \left| N_{\alpha_{R}(t_{k})} \right| dt,$$

$$I_{8,b}^{\text{sim}} = \frac{1}{\Delta T} \int_{0}^{T} \left( |\beta_{R}^{\text{sim}}(t)| - \alpha_{R,max} \right) dt.$$

The indices $I_{9,e}^{\text{sim}}$, $I_{11,e}^{\text{sim}}$, and $I_{13,e}^{\text{sim}}$ are the average sums of the average of absolute values of the lateral slips, $\tan(\alpha_{ij}^{\text{sim}})$, longitudinal slips, $\kappa_{ij}^{\text{sim}}$, and combined slips, $s_{ij}^{\text{sim}} = \sqrt{\kappa_{ij}^{\text{sim}}^2 + \tan(\alpha_{ij}^{\text{sim}})^2}$, on every corner,

$$I_{9,e}^{\text{sim}} = \frac{1}{4} \sum_{ij} \left( \frac{1}{\Delta T} \int_{0}^{T} |\tan(\alpha_{ij}^{\text{sim}}(t))| dt \right),$$

$$I_{11,e}^{\text{sim}} = \frac{1}{4} \sum_{ij} \left( \frac{1}{\Delta T} \int_{0}^{T} |\kappa_{ij}^{\text{sim}}(t)| dt \right),$$

$$I_{13,e}^{\text{sim}} = \frac{1}{4} \sum_{ij} \left( \frac{1}{\Delta T} \int_{0}^{T} |s_{ij}^{\text{sim}}(t)| dt \right).$$

The indices $I_{10,e}^{\text{sim}}$, $I_{12,e}^{\text{sim}}$, and $I_{14,e}^{\text{sim}}$ are the average sums of the maximum of absolute values of the lateral slips, longitudinal slips, and combined slips on every corner,

$$I_{10,e}^{\text{sim}} = \frac{1}{4} \sum_{ij} \left( \max_{t \in [0,T]} \left| \tan(\alpha_{ij}^{\text{sim}}(t)) \right| \right),$$

$$I_{12,e}^{\text{sim}} = \frac{1}{4} \sum_{ij} \left( \max_{t \in [0,T]} \left| \kappa_{ij}^{\text{sim}}(t) \right| \right),$$

$$I_{14,e}^{\text{sim}} = \frac{1}{4} \sum_{ij} \left( \max_{t \in [0,T]} \left| s_{ij}^{\text{sim}}(t) \right| \right).$$

The index $I_{15,a}^{\text{sim}}$ indicates the absolute value of the lateral displacement, $Y$, of the vehicle centre of gravity with respect to its initial straight path at the beginning of the dwell, $T_{dwell}$, in the steering input. The index $I_{15,b}^{\text{sim}}$ indicates the maximum of the absolute value
of the lateral displacement,

\[ I_{15,a}^{\text{sim}} = |Y(T_{\text{dwell}})|, \quad (5.87) \]

\[ I_{15,b}^{\text{sim}} = \max_{t \in [0,T]} |Y(t)|. \quad (5.88) \]

The indices \( I_{16,e}^{\text{sim}}, I_{16,e}^{\text{est}}, I_{18,e}^{\text{sim}}, I_{18,e}^{\text{est}}, \) and \( I_{20,e}^{\text{est}} \) are the average sums of the average of absolute values of the lateral friction utilisation, \( \mu_{s_{ij}} = \frac{|F_{s_{ij}}|}{F_{z_{ij}}}, \) longitudinal friction utilisation, \( \mu_{l_{ij}} = \frac{|F_{l_{ij}}|}{F_{z_{ij}}}, \) and combined friction utilisation, \( \mu_{c_{ij}} = \frac{|F_{c_{ij}}|}{F_{z_{ij}}}, \) on every corner of the simulation and prediction model, respectively, with \( F_{ij} = \sqrt{F_{l_{ij}}^2 + F_{s_{ij}}^2} \),

\[ I_{16,e}^{\text{sim}} = \frac{1}{4} \sum_{ij} \left( \frac{1}{\Delta T} \int_0^T |\mu_{s_{ij}}(t)| \, dt \right), \quad (5.89) \]

\[ I_{18,e}^{\text{sim}} = \frac{1}{4} \sum_{ij} \left( \frac{1}{\Delta T} \int_0^T |\mu_{l_{ij}}(t)| \, dt \right), \quad (5.90) \]

\[ I_{20,e}^{\text{sim}} = \frac{1}{4} \sum_{ij} \left( \frac{1}{\Delta T} \int_0^T |\mu_{c_{ij}}(t)| \, dt \right), \quad (5.91) \]

\[ I_{16,e}^{\text{est}} = \frac{1}{4} \sum_{ij} \left( \frac{1}{\Delta T} \int_0^T |\mu_{s_{ij}}(t)| \, dt \right), \quad (5.92) \]

\[ I_{18,e}^{\text{est}} = \frac{1}{4} \sum_{ij} \left( \frac{1}{\Delta T} \int_0^T |\mu_{l_{ij}}(t)| \, dt \right), \quad (5.93) \]

\[ I_{20,e}^{\text{est}} = \frac{1}{4} \sum_{ij} \left( \frac{1}{\Delta T} \int_0^T |\mu_{c_{ij}}(t)| \, dt \right). \quad (5.94) \]

The indices \( I_{17,e}, I_{19,e}, \) and \( I_{21,e} \) are the sums of the average of absolute values of the errors of lateral friction utilisation, longitudinal friction utilisation, and combined friction utilisation on every corner,

\[ I_{17,e} = \sum_{ij} \left( \frac{1}{\Delta T} \int_0^T |\mu_{s_{ij}}(t) - \mu_{s_{ij}}^{\text{est}}(t)| \, dt \right), \quad (5.95) \]

\[ I_{19,e} = \sum_{ij} \left( \frac{1}{\Delta T} \int_0^T |\mu_{l_{ij}}(t) - \mu_{l_{ij}}^{\text{est}}(t)| \, dt \right), \quad (5.96) \]

\[ I_{21,e} = \sum_{ij} \left( \frac{1}{\Delta T} \int_0^T |\mu_{c_{ij}}(t) - \mu_{c_{ij}}^{\text{est}}(t)| \, dt \right). \quad (5.97) \]
5.3.4 Controlled and uncontrolled vehicle

Uncontrolled vehicle

The uncontrolled vehicle is simulated along the series of Sine with Dwell tests, starting from run 1 with a steering wheel angle amplitude $\hat{\delta}_{sw} = 34.7\,\text{deg}$ up to the final run 22 with an amplitude $\hat{\delta}_{sw} = 277.2\,\text{deg}$.

![Graphs of uncontrolled vehicle responses](image)

Fig. 5.10: Dynamic response of the uncontrolled vehicle in run 12 of the Sine with Dwell test series with a steering wheel angle amplitude $\hat{\delta}_{sw} = 161.7\,\text{deg}$.

Figure 5.10 reports the response of the passive vehicle, i.e. the vehicle without the stability controller, during test run 12 with a steering wheel angle amplitude $\hat{\delta}_{sw} = 161.7\,\text{deg}$.
The large values of rear slip angle and vehicle side-slip angle show that the vehicle manoeuvrability is compromised. Similar behaviour can be observed for the following test runs of the uncontrolled vehicle. In Fig. 5.11, the trajectory of the passive vehicle in the rear slip angle and yaw velocity error domain is shown for the entire series of SwD test runs, demonstrating the excessive values of the rear slip angle.

![Fig. 5.11: Dynamic response of the uncontrolled vehicle in the entire Sine with Dwell test series in the rear slip angle and yaw velocity error domain. The yaw velocity error is introduced to obtain results comparable to the controlled vehicle and is calculated based on (5.59).](image)

Similarly, the performance indicators for the yaw velocity error and rear slip angle, as demonstrated in Fig. 5.12 for the entire test series, reflect the dynamic response. The average values of the performance indicators are summarised in Tables 5.4-5.6. In particular, very high values of the yaw velocity and rear slip angle indicators, i.e. $\bar{I}_7^{\text{sim}}$, $\bar{I}_{8,a}^{\text{sim}}$, and $\bar{I}_{8,b}^{\text{sim}}$, can be observed. Subsequently, also the lateral slip indicators, $\bar{I}_{9e}^{\text{sim}}$ and $\bar{I}_{10,e}^{\text{sim}}$, and the lateral friction utilisation indicator, $\bar{I}_{16,e}^{\text{sim}}$, have high values.

**Controlled vehicle**

Analogously to the passive vehicle, the controlled vehicle is simulated along the series of Sine with Dwell tests. Figure 5.13 shows the vehicle response of the controlled vehicle with the non-linear MPC-based stability controller. The formulation including the estimation A.iii of the varying vertical tyre forces, the simplified model B.ii of the Pacejka Magic Formula with linear peak factor C.i and linear stiffness factor C.iii, and linear tyre force
5.3 Performance analysis

Fig. 5.12: Selected performance indicators of the uncontrolled vehicle in the entire Sine with Dwell test series.

Coupling D.ii is later in Section 5.3.6 introduced as controller (f). Unlike the simulations of the uncontrolled vehicle, the results of the controlled vehicle are shown only for the interval [0s, 3s] since a steady-state is reached for the remaining interval.

In contrast to the uncontrolled vehicle, the rear slip angle and vehicle side-slip angle have moderate values, which demonstrates the effectiveness of the proposed control system. The violation of the yaw velocity error bound (indicated by the dashed lines in the relevant sub-plot of Fig. 5.13 and 5.14) is significantly reduced by the control action. Consequently, the yaw velocity profile proximately follows the steering wheel input profile. Moreover, the decrease of the vehicle speed due to differential braking is at an acceptable level, and the final value of the vehicle speed is higher than for the passive vehicle.

In Fig. 5.15, a very significant reduction of the performance indicators for the yaw velocity error and the rear slip angle can be observed compared to Fig. 5.12. This can also be identified by the average performance indicators $\bar{I}_{7,\text{sim}}$, $\bar{I}_{8,\text{a,sim}}$, and $\bar{I}_{8,\text{b,sim}}$, reported in Tables 5.4-5.6. At the same time the lateral slip indicators, $I_{9,\text{c,sim}}$ and $I_{10,\text{c,sim}}$, and lateral friction utilisation indicator, $I_{16,\text{c,sim}}$, significantly reduce and an acceptable reduction of the lateral displacement indicators, $\bar{I}_{15,\text{a,sim}}$ and $\bar{I}_{15,\text{b,sim}}$, occurs.
Fig. 5.13: Dynamic response of the controlled vehicle with controller (f) in run 12 of the Sine with Dwell test series with a steering wheel angle amplitude $\hat{\delta}_{sw} = 161.7$ deg. Solid lines indicate the vehicle states from the high-fidelity simulation model and dash-dotted lines indicate the vehicle states estimated from the prediction model.
5.3 Performance analysis

Fig. 5.14: Dynamic response of the controlled vehicle with controller (f) in the entire Sine with Dwell test series in the rear slip angle and yaw velocity error domain. The dashed lines indicate the bounds on the yaw velocity error and rear slip angle.

Figure 5.16 shows the performance indices for the directional stability and responsiveness criteria, defined in equations (5.63)-(5.65), for the passive vehicle, without the stability controller, and the controlled vehicle, with the non-linear MPC stability controller active, simulated for the entire SwD test series. The dotted lines in the figure represent the threshold values for $J_1$, $J_2$, and $J_3$ defined by the regulation UN/ECE-R 13H (1958) with the shaded areas indicating the inadmissible zones. A very significant reduction of the performance indices is achieved by the proposed control system, for the entire set of values of the steering wheel angle amplitude. At the same time, the responsiveness criterion remains almost unchanged by the stability controller. These results demonstrate the fulfilment of the performance requirements of the regulation UN/ECE-R 13H (1958). It is interesting to observe that the index $J_3$ defined in the regulation by double integration is more conservative than the indicator $I_{15}$ obtained from the actual displacement of the vehicle.
Fig. 5.15: Selected performance indicators of the controlled vehicle with controller (f) in the entire Sine with Dwell test series. Solid lines refer to the indicators from the high-fidelity simulation model and dash-dotted lines refer to the indicators from the prediction model.
5.3 Performance analysis

Fig. 5.16: Directional stability and responsiveness criteria for the uncontrolled vehicle and controlled vehicle with controller (f) in the entire Sine with Dwell test series.

5.3.5 Controller tuning and constraints

Weights in the cost function

The interdependence of the two main objectives, namely reducing the violation of the bound on the yaw velocity error and rear slip angle, can be influenced by the appropriate selection of the weights in the cost function (5.39). To demonstrate the influence of the weights, three different tunings of the controller with Setting A are defined:

- Tuning (I): Controller (f) with Setting A - Priority is given to the rear slip angle contribution, with moderate weights on the yaw velocity error contribution.

- Tuning (II): The rear slip angle contribution is set to zero, and the yaw velocity error contribution remains unchanged with respect to tuning (I).

- Tuning (III): The rear slip angle contribution is set to zero, and the yaw velocity error weight, \( q_{\nu,\psi} \), is increased by a factor of 1.5 with respect to tuning (I).

Figure 5.17 shows the performance indicators for the series of 22 test runs. Despite having the same yaw velocity error contribution, tuning (II) brings significantly larger values of violation of the yaw velocity error bounds, represented by \( I_{\text{sim}} \), compared to tuning (I). Moreover, much larger values of the rear slip angle, indicated by \( I_{8a}^{\text{sim}} \) and \( I_{8b}^{\text{sim}} \), can be observed in particular in medium runs of the test series, due to the missing rear slip angle contribution. This also leads to a reduced average of the control action,
Vehicle stability control

Fig. 5.17: Selected performance indicators demonstrating the influence of different weights by means of controller (f) with the different tunings (I)-(III). The solid lines refer to the indicators from the high-fidelity simulation model over the entire Sine with Dwell test series.

captured by lower values of $I_{6}^{\text{sim}}$. These trends can also be identified in the average values of the performance indicators over the test series in Tables 5.4-5.6.

In order to achieve comparable yaw velocity tracking, in tuning (III) the yaw velocity error weight, $q_{\nu,e}^{\dot{\psi}}$, was increased by a factor of 1.5 with respect to tunings (I) and (II). As a result, much larger values of $I_{5}^{\text{sim}}$, i.e. the maximum direct yaw moment, and an increase of $I_{6}^{\text{sim}}$, i.e. the average direct yaw moment, are observed. The values of $I_{8,a}^{\text{sim}}$ and $I_{8,b}^{\text{sim}}$ are at an acceptable level, but they are still higher compared to tuning (I) in medium runs, because of the missing rear slip angle contribution. Again, these relations can be identified in the average values of the indicators in Tables 5.4-5.6.

The conclusion of this analysis is that the inclusion of rear slip angle constraints with relatively large weights in the optimal control problem formulation leads to small maximum values of the direct yaw moment. At the same time, the yaw velocity tracking performance is not compromised while the rear slip angle value is limited.

As shown in more detail in Lenzo et al. (2017), the side-slip angle at the centre of gravity consists of kinematic and dynamic contributions. In particular, the kinematic contribution could bring undesired interventions of the controller if this was based on
the side-slip angle at the centre of gravity, especially in conditions of small instantaneous radius of curvature. On the other hand, the rear slip angle is only a function of the system dynamics and does not include any kinematic contribution. This justifies the selection of rear slip angle constraints, rather than constraints on the side-slip angle at the centre of gravity. The interplay of the violations on yaw velocity error and rear slip angle, as observed from the comparison of tunings (I) and (III), is described in Beal and Gerdes (2013) for a vehicle actuated by a front-wheel steering system using a so-called envelope control.

**Longitudinal tyre force constraints**

The lower bounds on the longitudinal tyre forces in (5.53) are formulated as soft constraints to allow a certain violation. Together with the soft constraints in (5.54), the importance of direct yaw moment generation and braking force generation in limit conditions can be specified. In Fig. 5.18, a comparison of the controller (f) is demonstrated with the lower bounds in (5.53) inactive in the left sub-plots and active in the right sub-plots. Since there is no separate slip control module in the implementation of the vehicle stability controller, the longitudinal slips can grow to high values. In the constrained case, the longitudinal slips are significantly reduced and, moreover, the performance in terms of yaw velocity error and rear slip angle response is improved.

In the simulations shown in Fig. 5.18, a reference trajectory for the longitudinal deceleration is imposed on the controller. A longitudinal deceleration demand generally facilitates the direct yaw moment generation. This is due to the fact that both vehicle sides can contribute by means of the friction brakes and allow for higher values of direct yaw moment. This leads to reduced violations of the bound on the yaw velocity error and rear slip angle compared to the case with zero deceleration demand.

**Direct yaw moment reference**

Figure 5.19 shows a comparison of the controller (f) with different direct yaw moment reference trajectories. In the left sub-plot, the controller is based on the zero reference E.i for the direct yaw moment, as in Setting A. Whereas in the right sub-plot, the controller has the direct yaw moment reference E.ii with proportional contributions based on the violations of the bounds on the yaw velocity error and rear slip angle, as in Setting B.

The direct yaw moment reference E.ii leads to slightly increased average direct yaw moments and, therefore, to reduced values of the yaw velocity error and rear slip angle. At the same time, the maximum direct yaw moment is marginally reduced. However, including the direct yaw moment reference E.ii does not bring any considerable performance benefits neither in nominal conditions nor in conditions with parameter variations of the plant. In fact, even a performance decay can be observed for increasing
Fig. 5.18: Influence of soft constraints on longitudinal tyres forces on the performance of controller (f) in run 12 of the Sine with Dwell test series with a trajectory for the longitudinal deceleration demand. Solid lines indicate the vehicle states from the high-fidelity simulation model and dash-dotted lines indicate the vehicle states estimated from the prediction model.
values of the weight, \( r_{u,Mz} \), in Setting B. For this reason, the reference E.ii can be replaced by the simple zero direct yaw moment reference E.i.

### 5.3.6 Model complexity

The simulation of controllers with different formulations of the prediction model, as introduced in Section 5.2.1, allows to investigate the influence of varying prediction model fidelity on the performance of the non-linear MPC controller.

#### Test scenarios

For the controlled vehicle, different test cases are defined in Table 5.3. Tables 5.4-5.6 report the average values, \( \bar{I} \), of the performance indicators, introduced in Section 5.3.3, over the entire series of 22 Sine with Dwell test runs. To investigate the influence of the prediction model on the performance, Setting A as defined in Table 5.2 is chosen for the NMPC. Moreover, the lower bounds in (5.53) and consequently also (5.58) are disabled, avoiding the influence of significant estimation errors of the maximum longitudinal forces in case of constant vertical tyre loads. Also, an implicit solution of the controller is considered to avoid any influence of the sub-optimal solution.

The following sections investigate the influence of the different prediction models and are sorted according to the decreasing significance of each contribution.
Table 5.3: Definition of test cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>Vertical tyre force $F_{z_{ij}}$</th>
<th>Tyre model $F_{\theta_{ij}}$</th>
<th>Peak factor $D_{s_{ij}}$</th>
<th>Stiffness factor $B_{s_{ij}}$</th>
<th>Coupling $F_{s_{ij}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>A.i const.</td>
<td>B.i lin.</td>
<td>C.i lin.</td>
<td>C.iii lin.</td>
<td>D.i pure</td>
</tr>
<tr>
<td>(b)</td>
<td>A.i const.</td>
<td>B.ii sMF</td>
<td>C.i lin.</td>
<td>C.iii lin.</td>
<td>D.i pure</td>
</tr>
<tr>
<td>(c)</td>
<td>A.i const.</td>
<td>B.ii sMF</td>
<td>C.i lin.</td>
<td>C.iii lin.</td>
<td>D.ii lin.</td>
</tr>
<tr>
<td>(d)</td>
<td>A.ii vary. $\dot{\psi}$</td>
<td>B.ii sMF</td>
<td>C.i lin.</td>
<td>C.iii lin.</td>
<td>D.ii lin.</td>
</tr>
<tr>
<td>(f)</td>
<td>A.iii vary. $\dot{\psi}, \dot{\beta}$</td>
<td>B.ii sMF</td>
<td>C.i lin.</td>
<td>C.iii lin.</td>
<td>D.ii lin.</td>
</tr>
<tr>
<td>(g)</td>
<td>A.iii vary. $\dot{\psi}, \dot{\beta}$</td>
<td>B.ii sMF</td>
<td>C.i lin.</td>
<td>C.iii lin.</td>
<td>D.i pure</td>
</tr>
<tr>
<td>(h)</td>
<td>A.iii vary. $\dot{\psi}, \dot{\beta}$</td>
<td>B.ii sMF</td>
<td>C.ii non-lin.</td>
<td>C.iii lin.</td>
<td>D.ii lin.</td>
</tr>
<tr>
<td>(n)</td>
<td>A.iii vary. $\dot{\psi}, \dot{\beta}$</td>
<td>B.ii sMF</td>
<td>C.i lin.</td>
<td>C.iii lin.</td>
<td>D.iii cos.</td>
</tr>
</tbody>
</table>

Non-linear lateral tyre force model

Investigating the influence of the non-linearity in the pure slip lateral tyre force model, controllers (a) and (b) with constant vertical loads and without tyre force coupling and controllers (e) and (d) with varying vertical loads and linear tyre force coupling are compared. The predicted lateral forces for the linear tyre model in the controllers (a) and (e) are slightly underestimated for small tyre slip angles. However, they are significantly overestimated for increasing values of the tyre slip angles, as shown in Fig. 5.2. The tyre slip angles are in general considerably higher on the front axle, in particular, due to the steering angle, causing a significant overestimate of the lateral forces on the front axle. Consequently, this leads to a considerable overestimate of the predicted yaw moment contribution, $M_{z,F_{\theta}}$, of the front axle lateral forces, as seen in Fig. 5.20 comparing controllers (e) and (d). Since this phenomenon is dependent on the tyre slip angles, its significance increases with higher test runs. To compensate for higher values of predicted yaw velocity, caused by the overestimated yaw moment of lateral forces, controllers (a) and (e) generate significantly higher stabilising direct yaw moments compared to controllers (b) and (d), even if the yaw velocity error is still within the tolerable dead-zone, as seen in Fig. 5.20 at around 0.7 s. Controllers (a) and (e) act, therefore, significantly more aggressive generating bigger direct yaw moments, $M_{z,F_{\theta}}$, by
Table 5.4: Part I of average performance indicators for a series of 22 Sine with Dwell test runs with NMPC controller using Setting A.

<table>
<thead>
<tr>
<th>Case</th>
<th>$\bar{I}_{1,FL}^{\text{sim}}$ (N m)</th>
<th>$\bar{I}_{1,FL}^{\text{est}}$ (N m)</th>
<th>$\bar{I}_{1,FR}^{\text{sim}}$ (N m)</th>
<th>$\bar{I}_{1,FR}^{\text{est}}$ (N m)</th>
<th>$\bar{I}_{1,RL}^{\text{sim}}$ (N m)</th>
<th>$\bar{I}_{1,RL}^{\text{est}}$ (N m)</th>
<th>$\bar{I}_{2,FL}^{\text{sim}}$ (N m)</th>
<th>$\bar{I}_{2,FL}^{\text{est}}$ (N m)</th>
<th>$\bar{I}_{2,FR}^{\text{sim}}$ (N m)</th>
<th>$\bar{I}_{2,FR}^{\text{est}}$ (N m)</th>
<th>$\bar{I}_{2,RL}^{\text{sim}}$ (N m)</th>
<th>$\bar{I}_{2,RL}^{\text{est}}$ (N m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>passive</td>
<td>4300</td>
<td>-</td>
<td>2971</td>
<td>-</td>
<td>4656</td>
<td>-</td>
<td>2692</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(a)</td>
<td>2021</td>
<td>4349</td>
<td>2124</td>
<td>5373</td>
<td>2446</td>
<td>1700</td>
<td>1812</td>
<td>1647</td>
<td>2959</td>
<td>3703</td>
<td>1284</td>
<td>1164</td>
</tr>
<tr>
<td>(b)</td>
<td>2287</td>
<td>2124</td>
<td>2247</td>
<td>2537</td>
<td>2983</td>
<td>2379</td>
<td>2161</td>
<td>2370</td>
<td>913</td>
<td>946</td>
<td>1588</td>
<td>1442</td>
</tr>
<tr>
<td>(c)</td>
<td>2249</td>
<td>2049</td>
<td>2223</td>
<td>2481</td>
<td>2936</td>
<td>2314</td>
<td>2129</td>
<td>2314</td>
<td>1012</td>
<td>995</td>
<td>1602</td>
<td>1459</td>
</tr>
<tr>
<td>(d)</td>
<td>2217</td>
<td>2139</td>
<td>2263</td>
<td>2085</td>
<td>2974</td>
<td>2650</td>
<td>2142</td>
<td>1968</td>
<td>574</td>
<td>611</td>
<td>843</td>
<td>689</td>
</tr>
<tr>
<td>(e)</td>
<td>2021</td>
<td>4400</td>
<td>2117</td>
<td>4457</td>
<td>2393</td>
<td>1895</td>
<td>1781</td>
<td>1331</td>
<td>3057</td>
<td>2871</td>
<td>902</td>
<td>660</td>
</tr>
<tr>
<td>(f)-(I)</td>
<td>2222</td>
<td>2173</td>
<td>2260</td>
<td>2136</td>
<td>2973</td>
<td>2705</td>
<td>2141</td>
<td>1891</td>
<td>384</td>
<td>469</td>
<td>578</td>
<td>494</td>
</tr>
<tr>
<td>(f)-(II)</td>
<td>2373</td>
<td>2354</td>
<td>2283</td>
<td>2148</td>
<td>3113</td>
<td>2884</td>
<td>2132</td>
<td>1894</td>
<td>389</td>
<td>474</td>
<td>561</td>
<td>479</td>
</tr>
<tr>
<td>(f)-(III)</td>
<td>2308</td>
<td>2280</td>
<td>2241</td>
<td>2116</td>
<td>2980</td>
<td>2704</td>
<td>2117</td>
<td>1858</td>
<td>401</td>
<td>464</td>
<td>560</td>
<td>486</td>
</tr>
<tr>
<td>(g)</td>
<td>2259</td>
<td>2324</td>
<td>2311</td>
<td>2232</td>
<td>3036</td>
<td>2775</td>
<td>2185</td>
<td>1937</td>
<td>445</td>
<td>463</td>
<td>589</td>
<td>490</td>
</tr>
<tr>
<td>(h)</td>
<td>2224</td>
<td>2159</td>
<td>2264</td>
<td>2226</td>
<td>2977</td>
<td>2807</td>
<td>2142</td>
<td>2001</td>
<td>343</td>
<td>374</td>
<td>613</td>
<td>523</td>
</tr>
<tr>
<td>(i)</td>
<td>2210</td>
<td>2203</td>
<td>2268</td>
<td>2179</td>
<td>2975</td>
<td>2879</td>
<td>2145</td>
<td>2060</td>
<td>400</td>
<td>472</td>
<td>606</td>
<td>488</td>
</tr>
<tr>
<td>(j)</td>
<td>2216</td>
<td>2185</td>
<td>2275</td>
<td>2252</td>
<td>2985</td>
<td>2981</td>
<td>2146</td>
<td>2174</td>
<td>331</td>
<td>393</td>
<td>676</td>
<td>561</td>
</tr>
<tr>
<td>(k)</td>
<td>2273</td>
<td>2303</td>
<td>2284</td>
<td>2299</td>
<td>3015</td>
<td>3001</td>
<td>2171</td>
<td>2197</td>
<td>330</td>
<td>371</td>
<td>664</td>
<td>562</td>
</tr>
<tr>
<td>(l)</td>
<td>2208</td>
<td>2229</td>
<td>2247</td>
<td>2175</td>
<td>2948</td>
<td>2619</td>
<td>2140</td>
<td>1863</td>
<td>423</td>
<td>409</td>
<td>594</td>
<td>504</td>
</tr>
<tr>
<td>(m)</td>
<td>2199</td>
<td>2259</td>
<td>2254</td>
<td>2253</td>
<td>2953</td>
<td>2834</td>
<td>2143</td>
<td>2064</td>
<td>420</td>
<td>403</td>
<td>610</td>
<td>510</td>
</tr>
<tr>
<td>(n)</td>
<td>2281</td>
<td>2293</td>
<td>2270</td>
<td>2187</td>
<td>3003</td>
<td>2725</td>
<td>2166</td>
<td>1912</td>
<td>412</td>
<td>466</td>
<td>572</td>
<td>492</td>
</tr>
</tbody>
</table>
Table 5.5: Part II of average performance indicators for a series of 22 Sine with Dwell test runs with NMPC controller using Setting A.

<table>
<thead>
<tr>
<th>Case</th>
<th>Total yaw moment</th>
<th>Yaw moment errors</th>
<th>Maximum direct yaw moment</th>
<th>Average direct yaw moment</th>
<th>Yaw velocity error</th>
<th>Rear slip angle</th>
<th>Lateral displacement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$I_{3}^{\text{sim}}$ (N m)</td>
<td>$I_{3}^{\text{est}}$ (N m)</td>
<td>$I_{4,a}$ (N m)</td>
<td>$I_{4,b}$ (N m)</td>
<td>$I_{5}^{\text{sim}}$ (N m)</td>
<td>$I_{5}^{\text{est}}$ (N m)</td>
<td>$I_{6}^{\text{sim}}$ (N m)</td>
</tr>
<tr>
<td>passive</td>
<td>1756</td>
<td>-</td>
<td>-</td>
<td>348</td>
<td>-</td>
<td>81</td>
<td>-</td>
</tr>
<tr>
<td>(a)</td>
<td>2257</td>
<td>7669</td>
<td>9110</td>
<td>6467</td>
<td>4612</td>
<td>5011</td>
<td>1358</td>
</tr>
<tr>
<td>(b)</td>
<td>2035</td>
<td>2083</td>
<td>4889</td>
<td>670</td>
<td>4059</td>
<td>4185</td>
<td>747</td>
</tr>
<tr>
<td>(c)</td>
<td>2009</td>
<td>2101</td>
<td>5068</td>
<td>759</td>
<td>4085</td>
<td>4230</td>
<td>720</td>
</tr>
<tr>
<td>(d)</td>
<td>2063</td>
<td>2165</td>
<td>2718</td>
<td>830</td>
<td>3793</td>
<td>3909</td>
<td>679</td>
</tr>
<tr>
<td>(e)</td>
<td>2252</td>
<td>6906</td>
<td>7490</td>
<td>5798</td>
<td>4044</td>
<td>4504</td>
<td>1168</td>
</tr>
<tr>
<td>(f)-(I)</td>
<td>2053</td>
<td>2122</td>
<td>1925</td>
<td>749</td>
<td>3845</td>
<td>3964</td>
<td>678</td>
</tr>
<tr>
<td>(f)-(II)</td>
<td>1990</td>
<td>2064</td>
<td>1902</td>
<td>733</td>
<td>3911</td>
<td>4009</td>
<td>647</td>
</tr>
<tr>
<td>(f)-(III)</td>
<td>1957</td>
<td>2001</td>
<td>1911</td>
<td>688</td>
<td>4427</td>
<td>4579</td>
<td>702</td>
</tr>
<tr>
<td>(g)</td>
<td>2112</td>
<td>2100</td>
<td>1988</td>
<td>642</td>
<td>3565</td>
<td>3663</td>
<td>724</td>
</tr>
<tr>
<td>(h)</td>
<td>2048</td>
<td>2273</td>
<td>1854</td>
<td>854</td>
<td>3908</td>
<td>4029</td>
<td>674</td>
</tr>
<tr>
<td>(i)</td>
<td>2067</td>
<td>2275</td>
<td>1966</td>
<td>833</td>
<td>3766</td>
<td>3874</td>
<td>674</td>
</tr>
<tr>
<td>(j)</td>
<td>2059</td>
<td>2479</td>
<td>1961</td>
<td>996</td>
<td>3855</td>
<td>3966</td>
<td>671</td>
</tr>
<tr>
<td>(k)</td>
<td>2033</td>
<td>2381</td>
<td>1928</td>
<td>898</td>
<td>4202</td>
<td>4434</td>
<td>692</td>
</tr>
<tr>
<td>(l)</td>
<td>2065</td>
<td>1883</td>
<td>1930</td>
<td>662</td>
<td>3750</td>
<td>3865</td>
<td>686</td>
</tr>
<tr>
<td>(m)</td>
<td>2074</td>
<td>2071</td>
<td>1943</td>
<td>738</td>
<td>3685</td>
<td>3785</td>
<td>680</td>
</tr>
<tr>
<td>(n)</td>
<td>2028</td>
<td>2031</td>
<td>1942</td>
<td>663</td>
<td>4226</td>
<td>4385</td>
<td>697</td>
</tr>
</tbody>
</table>
Table 5.6: Part III of average performance indicators for a series of 22 Sine with Dwell test runs with NMPC controller using Setting A.

<table>
<thead>
<tr>
<th>Case</th>
<th>Average lateral, longitudinal, and combined slip</th>
<th>Maximum lateral, longitudinal, and combined slip</th>
<th>Lateral, longitudinal, and combined friction utilisation</th>
<th>Lateral, longitudinal, and combined friction utilisation error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( I_{g,e}^{sim} ) (%), ( I_{11,e}^{sim} ) (%)</td>
<td>( I_{10,e}^{sim} ) (%), ( I_{12,e}^{sim} ) (%), ( I_{14,e}^{sim} ) (%)</td>
<td>( I_{15,e}^{sim} ) (%), ( I_{16,e}^{sim} ) (%), ( I_{17,e}^{sim} ) (%), ( I_{18,e}^{sim} ) (%), ( I_{19,e}^{sim} ) (%), ( I_{20,e}^{sim} ) (%), ( I_{21,e}^{sim} ) (%), ( I_{22,e}^{sim} ) (%)</td>
<td></td>
</tr>
<tr>
<td>passive</td>
<td>38.08 0.32 38.09</td>
<td>81.06 1.40 81.06</td>
<td>62.63 - 2.22</td>
<td>- 62.76</td>
</tr>
<tr>
<td>(a)</td>
<td>5.36 0.95 5.59</td>
<td>14.16 7.03 15.38</td>
<td>37.77 55.38 8.04 6.86</td>
<td>40.18 56.97</td>
</tr>
<tr>
<td>(b)</td>
<td>6.33 0.37 6.37</td>
<td>16.81 2.84 16.85</td>
<td>43.62 39.02 4.93</td>
<td>3.84 44.48 40.05</td>
</tr>
<tr>
<td>(c)</td>
<td>6.31 0.37 6.35</td>
<td>16.75 3.00 16.80</td>
<td>43.54 38.06 4.72</td>
<td>3.47 44.40 39.26</td>
</tr>
<tr>
<td>(d)</td>
<td>6.54 0.40 6.58</td>
<td>17.77 2.87 17.85</td>
<td>44.05 - 4.61</td>
<td>- 44.85 -</td>
</tr>
<tr>
<td>(e)</td>
<td>5.51 1.04 5.79</td>
<td>14.68 8.12 16.69</td>
<td>38.15 - 7.01</td>
<td>- 40.17 -</td>
</tr>
<tr>
<td>(f)-(I)</td>
<td>6.52 0.39 6.57</td>
<td>17.68 2.86 17.75</td>
<td>44.04 39.13 4.60</td>
<td>2.28 44.83 39.73</td>
</tr>
<tr>
<td>(f)-(II)</td>
<td>6.77 0.35 6.81</td>
<td>18.23 2.80 18.25</td>
<td>45.20 40.50 4.36</td>
<td>2.18 45.96 41.08</td>
</tr>
<tr>
<td>(f)-(III)</td>
<td>6.40 0.36 6.45</td>
<td>16.92 3.36 16.93</td>
<td>44.09 39.12 4.61</td>
<td>2.40 45.03 39.89</td>
</tr>
<tr>
<td>(g)</td>
<td>6.59 0.43 6.63</td>
<td>18.02 2.89 18.12</td>
<td>44.12 39.82 5.00</td>
<td>2.76 44.91 40.35</td>
</tr>
<tr>
<td>(h)</td>
<td>6.53 0.40 6.57</td>
<td>17.73 2.91 17.80</td>
<td>44.07 41.35 4.58</td>
<td>2.26 44.87 41.97</td>
</tr>
<tr>
<td>(i)</td>
<td>6.55 0.40 6.59</td>
<td>17.80 2.90 17.88</td>
<td>44.09 41.30 4.60</td>
<td>2.27 44.88 41.89</td>
</tr>
<tr>
<td>(j)</td>
<td>6.57 0.40 6.61</td>
<td>17.89 2.97 17.97</td>
<td>44.18 43.55 4.59</td>
<td>2.26 44.97 44.16</td>
</tr>
<tr>
<td>(k)</td>
<td>6.50 0.38 6.55</td>
<td>17.52 3.34 17.71</td>
<td>44.17 43.81 4.69</td>
<td>2.50 44.97 44.43</td>
</tr>
<tr>
<td>(l)</td>
<td>6.47 0.40 6.51</td>
<td>17.46 2.85 17.53</td>
<td>43.80 39.10 4.63</td>
<td>2.31 44.60 39.70</td>
</tr>
<tr>
<td>(m)</td>
<td>6.49 0.41 6.54</td>
<td>17.58 2.89 17.66</td>
<td>43.88 41.89 4.63</td>
<td>2.29 44.67 42.48</td>
</tr>
<tr>
<td>(n)</td>
<td>6.46 0.37 6.50</td>
<td>17.30 3.06 17.35</td>
<td>44.05 39.41 4.68</td>
<td>2.47 44.85 39.99</td>
</tr>
</tbody>
</table>
applies higher longitudinal tyre forces. This results in significantly reduced yaw velocity errors and rear slip angles, especially for test runs with higher steering wheel angle amplitudes and, thus, higher tyre slip angles. There is actually no intervention due to the rear slip angle constraints for controller (a) and reduced intervention for controller (e). The increase in longitudinal tyre forces and direct yaw moments, respectively, leads to significantly reduced speed and lateral displacements. Controllers (a) and (e) result in much higher longitudinal slip ratios on the front axle caused by the yaw velocity error constraints, also leading to a significant decrease of lateral forces on the braked wheels due to the tyre force coupling. Controllers (b) and (d) result in slightly higher slip ratios on the front axle due to the rear slip angle constraints. The overestimation of the lateral tyre forces also results in an overestimation of the lateral friction utilisation.

These findings can also be identified by the performance indicators reported in Tables 5.4-5.6. In particular, the indicators based on the estimation of yaw moment contributions, such as \( \tilde{I}_{1,ij} \), \( \tilde{I}_{2,ij} \), \( \tilde{I}_{3}^{est} \), \( \tilde{I}_{4} \), and \( \tilde{I}_{4}^{b} \), considerably increase for controllers (a) and (e). In fact, a more significant influence can be observed for the indicators of the front axle. Likewise, the direct yaw moment indicators \( \tilde{I}_{5}^{sim} \), \( \tilde{I}_{5}^{est} \), \( \tilde{I}_{6}^{sim} \), and \( \tilde{I}_{6}^{est} \) increase, whereas the yaw velocity, rear slip angle, and lateral displacement indicators, i.e. \( \tilde{I}_{7} \), \( \tilde{I}_{8,a} \), \( \tilde{I}_{8,b} \), \( \tilde{I}_{15,a} \), and \( \tilde{I}_{15,b} \), decrease. The more aggressive behaviour causes increased longitudinal slips, \( \tilde{I}_{11,e}^{sim} \) and \( \tilde{I}_{12,e}^{sim} \), and reduced lateral slips, \( \tilde{I}_{9,e}^{sim} \) and \( \tilde{I}_{10,e}^{sim} \). Similar behaviour can be observed for the friction utilisation. In Fig. 5.21, the performance indicators \( I_{3} \), \( I_{4,a} \), \( I_{4,b} \), \( I_{7,a} \), \( I_{8,a} \), and \( I_{8,b} \) are reported over the series of 22 Sine with Dwell test runs. A significant influence can be observed for higher values of steering wheel angle amplitudes. The overestimation of the yaw moment contributions can also be identified in the indicators \( I_{3}^{est} \), \( I_{4,a} \), and \( I_{4,b} \) leading to a reduction of the indicators \( I_{7,a} \), \( I_{8,a} \), and \( I_{8,b} \). In the case (e), the controller is so aggressive that eventually \( I_{7} \) increases for high steering wheel angle amplitudes.

The conclusion of this comparison is that the non-linear lateral tyre force model significantly better estimates the yaw moment contributions of the lateral tyre forces. The linear tyre model considerably overestimates the lateral tyre forces with increasing influence dependent on the tyre slip angle. The overestimation of the corresponding yaw moment contributions causes aggressive behaviour in terms of yaw velocity and rear slip angle response. It is fair to say that an MPC with a linear lateral tyre force model as a prediction model does not give useful results for manoeuvres at the limits.

**Load transfer effects**

For the investigation of load transfer effects and the influence of the side-slip angle rate on the performance, controllers (b) and (g) with a non-linear lateral tyre force model without tyre force coupling and controllers (c), (f), and (d) with a non-linear
5.3 Performance analysis

Fig. 5.20: Comparison of the dynamic response of controllers (e) and (d) in run 12 of the Sine with Dwell test series demonstrating the influence of a non-linear lateral tyre force model. Solid lines indicate the vehicle states from the high-fidelity simulation model and dash-dotted lines indicate the vehicle states estimated from the prediction model.
Fig. 5.21: Comparison of selected performance indicators of controllers (e) and (d) in the entire Sine with Dwell test series demonstrating the influence of a non-linear lateral tyre force model. Solid lines refer to the indicators from the high-fidelity simulation model and dash-dotted lines refer to the indicators from the prediction model.
lateral tyre force model and linear tyre force coupling are compared. The predicted yaw moment contributions, in particular of the lateral tyre forces on the front axle, $M_{z,F_p}$, are influenced by lateral load transfer effects. Given the definitions for the prediction model, it is calculated as

$$M_{z,F_p} = \left( F_{sFL} + F_{sFR} \right) \cos \delta_F + \left( F_{sFL} - F_{sFR} \right) \sin \delta \frac{d}{2}. \quad (5.98)$$

Assuming a linear influence of the vertical load on the lateral tyre forces, e.g. as in C.i and C.iii, the lateral forces are computed, for constant vertical loads $F_{zF,stat} = F_{zF,stat} = F_{zF,stat}$ as defined in A.i, as

$$F_{sFL} = \mu_s(\alpha_F) F_{zF,stat} \quad \text{and} \quad F_{sFR} = \mu_s(\alpha_F) F_{zF,stat}. \quad (5.99)$$

leading to the yaw moment contribution,

$$M_{z,F_p}^{Fz,\text{const}} = 2 \mu_s(\alpha_F) F_{zF,stat} \cos \delta_F. \quad (5.100)$$

However, assuming varying vertical loads considering the effects of lateral load transfer, $\Delta F_{zF}$, on the front axle, the lateral tyre forces are computed as

$$F_{sFL} = \mu_s(\alpha_F) \left( F_{zF,stat} - \Delta F_{zF} \right) \quad \text{and} \quad F_{sFR} = \mu_s(\alpha_F) \left( F_{zF,stat} + \Delta F_{zF} \right), \quad (5.101)$$

leading to the yaw moment contribution,

$$M_{z,F_p}^{Fz,\text{var}} = M_{z,F_p}^{Fz,\text{const}} - 2 \mu_s(\alpha_F) \Delta F_{zF} \sin \delta \frac{d}{2}. \quad (5.102)$$

The predicted yaw moment contribution, $M_{z,F_p}^{Fz,\text{const}}$, of the lateral tyre forces on the front axle assuming constant vertical loads is, therefore, bigger compared to the predicted yaw moment contribution, $M_{z,F_p}^{Fz,\text{var}}$, with varying vertical loads. Based on (5.102), this phenomenon is dependent on the load transfer and geometric properties changing with the steering angle. To compensate for higher values of predicted yaw velocity due to the overestimation of the yaw moment, the controllers (b) and (c) with constant loads generate bigger stabilising and, vice versa, smaller destabilising direct yaw moments. The controllers (b) and (c) act, therefore, more aggressive, in particular for higher test runs implying higher values of steering wheel angle amplitudes and increasing significance of load transfer effects. This leads to reduced yaw velocity errors, rear slip angles, and lateral displacements. It goes along with less intervention due to the rear slip angle constraints. The comparison of controllers (f) and (d) shows that the controller (d) is the least aggressive due to the overestimation of load transfer effects based on the yaw velocity only. In particular in high transients, such as around 1.2 s in Fig. 5.22, the side-slip angle
rate grows to significant values and cannot be neglected. A comparison of the indicator $I_{4,a}$ in Fig. 5.23 and the indicators $I_{2,ij}$ and $I_{4,a}$ in Tables 5.4-5.6 for the controllers (c), (f), and (d), shows best results for the estimation of vertical loads based on yaw velocity and side-slip angle rate and acceptable results for the estimation based on the yaw velocity only, with the influence of the yaw velocity being more significant than the side-slip angle rate. The controllers (b) and (c) in Fig. 5.22 under- and overestimate the vertical, lateral, and maximum longitudinal tyre forces and the lateral force yaw moment contributions on the outer and inner wheels, respectively, due to the constant vertical loads. This also leads to an over- and underestimation of the longitudinal friction utilizations on the outer and inner wheels, respectively. The controller (d) has the opposite behaviour due to the overestimation of the lateral load transfer effects. The controllers (g) and (f) in Fig. 5.22 show a satisfactory estimation of the vertical, lateral, and maximum longitudinal tyre forces and the lateral force yaw moment contributions. The controllers (b) and (c) result in slightly increased longitudinal slip ratios due to the yaw velocity error constraints and reduced slips due to the rear slip angle constraints. The significant influence of geometric properties can be observed in Fig. 5.22. Although the estimated lateral tyre forces on the front axle in controller (c) are only slightly different caused by the linear tyre force coupling, the resulting yaw moment contributions are considerably different with increasing significance for test runs with higher steering wheel angle amplitudes.

The relations can also be identified by having a closer look at the performance indicators in Tables 5.4-5.6. The controllers (b) and (c) have the highest indicators $I_{2,ij}$ and $I_{4,a}$, followed by (d), and with the lowest indicators the controllers (g) and (f) due to the better estimation of the vertical tyre loads. The direct yaw moment indicators $I_{6}^{\text{sim}}$ and $I_{6}^{\text{est}}$, and in particular $I_{5}^{\text{sim}}$ and $I_{5}^{\text{est}}$, increase for (b) and (c) and, therefore, the yaw velocity, rear slip angle, and lateral displacement indicators, i.e. $I_{7}^{\text{sim}}$, $I_{8,a}^{\text{sim}}$, $I_{8,a}^{\text{est}}$, $I_{8,b}^{\text{sim}}$, $I_{15,a}^{\text{sim}}$, and $I_{15,b}^{\text{sim}}$, decrease. Due to less intervention of the rear slip angle constraints, indicators $I_{9,\epsilon}^{\text{sim}}$ to $I_{14,\epsilon}^{\text{sim}}$ are decreased. The indicators $I_{17,\epsilon}$, $I_{19,\epsilon}$, and $I_{21,\epsilon}$ are increased due to the estimation of the vertical forces. In Fig. 5.23, the performance indicators $I_{4,a}$, $I_{4,b}$, $I_{7}$, $I_{8,a}$, and $I_{8,b}$ are reported over the series of 22 Sine with Dwell test runs. The significant influence of load transfer effects on the estimation quality of the yaw moment contributions, shown by indicator $I_{4,a}$, can be identified especially in higher test runs.

Concluding this comparison, an accurate estimation of the vertical tyre loads is important for a reasonable estimation of yaw moment contributions with significant influence of the steering angle, lateral acceleration, and side-slip angle rate. An estimation of the side-slip angle rate has proven to have a beneficial influence in transient manoeuvres. The controllers with constant vertical loads overestimate the yaw moment contributions of the lateral tyre forces and are, therefore, too aggressive. For the vehicle stability control application of the MPC, a prediction model with load transfer effects and consideration of the side-slip angle rate is recommended.
Fig. 5.22: Comparison of the dynamic response of controllers (c) and (f) in run 12 of the Sine with Dwell test series demonstrating the influence of load transfer effects. Solid lines indicate the vehicle states from the high-fidelity simulation model and dash-dotted lines indicate the vehicle states estimated from the prediction model.
Fig. 5.23: Comparison of selected performance indicators of controllers (c) and (f) in the entire Sine with Dwell test series demonstrating the influence of load transfer effects. Solid lines refer to the indicators from the high-fidelity simulation model and dash-dotted lines refer to the indicators from the prediction model.
Tyre force coupling

For the investigation of the influence of longitudinal and lateral tyre force coupling, seven controllers with a non-linear tyre force model are compared, i.e. (b) and (c) with constant vertical loads; (g), (f), and (n) with varying vertical loads; and (j) and (k) with varying vertical loads and non-linear influence on the peak and stiffness factor. The tyre force coupling results in a decrease of the lateral tyre forces with increasing longitudinal tyre forces. The predicted yaw moment contributions of the lateral tyre forces are, therefore, influenced by the longitudinal tyre forces, e.g. a predicted destabilising yaw moment contribution of the lateral tyre forces on the front axle of a controller with consideration of the tyre force coupling is reduced by a stabilising direct yaw moment generated by front axle longitudinal forces and is, therefore, smaller compared to a controller without tyre force coupling. The reduction of the predicted lateral force yaw moment contribution depends on the tyre force coupling approximation, i.e. higher influence in the linear model D.ii for low values of longitudinal tyre forces and vice versa in the cosine model D.iii for higher values of longitudinal tyre forces, as also shown in Fig. 5.4. For both models, the longitudinal braking force distribution of the non-linear MPC controller is significantly influenced. The controllers (c), (f), (n), (j), and (k) generate higher longitudinal forces on the front axle for stabilising direct yaw moments to reduce a destabilising lateral force yaw moment contribution, and vice versa smaller front tyre forces for destabilising direct yaw moments to not reduce a destabilising lateral force yaw moment contribution. This can also be observed in Fig. 5.24 comparing controllers (g) and (f) with significant differences in the longitudinal tyre forces. The controllers (f) and (j) are braking most on the front, with highest force peaks for lower test runs, since the linear model yields highest reductions for small and medium longitudinal tyre forces. The controllers (n) and (k) are more aggressive and result in highest force peaks and maximum direct yaw moments for higher runs, due to the cosine model yielding highest reductions for high longitudinal tyre forces. The controllers (c), (f), (n), (j), and (k) lead, therefore, to a reduction in yaw velocity errors and rear slip angles causing less intervention due to the rear slip angle constraints, most significant for (n) and (k). The controllers (n) and (k), followed by (f) and (j), lead to more braking on the front and, thus, to increased longitudinal slip ratios and friction utilizations on the front due to the yaw velocity error constraints and to a reduction due to the rear slip angle constraints. The controllers (f) and (j) have the biggest reduction of the visually well estimated lateral tyre forces and lateral force yaw moment contributions on the front axle for medium runs and moderate longitudinal tyre forces. For higher runs and bigger longitudinal tyre forces, the controllers (f) and (n) have similar reductions as the controllers (j) and (k).

For the performance indicators in Tables 5.4-5.6, the maximum direct yaw moment indicators $\bar{I}_5^{\text{sim}}$ and $\bar{I}_5^{\text{est}}$ increase switching from no coupling in (b) and (g) to linear
Fig. 5.24: Comparison of the dynamic response of controllers (g) and (f) in run 12 of the Sine with Dwell test series demonstrating the influence of tyre force coupling. Solid lines indicate the vehicle states from the high-fidelity simulation model and dash-dotted lines indicate the vehicle states estimated from the prediction model.

coupling in (c) and (f). However, the average direct yaw moment indicators $\bar{I}_{6}^{\text{sim}}$ and $\bar{I}_{6}^{\text{est}}$ decrease together with the yaw velocity indicator $\bar{I}_{7}^{\text{sim}}$ and rear slip angle indicators $\bar{I}_{8,a}^{\text{sim}}$, $\bar{I}_{8,a}^{\text{est}}$, and $\bar{I}_{8,b}^{\text{sim}}$. For the cosine coupling in (n) and (k), $\bar{I}_{5}^{\text{sim}}$ and $\bar{I}_{5}^{\text{est}}$ increase, and $\bar{I}_{7}^{\text{sim}}$, $\bar{I}_{8,a}^{\text{sim}}$, $\bar{I}_{8,a}^{\text{est}}$, and $\bar{I}_{8,b}^{\text{sim}}$ decrease even further, but also $\bar{I}_{6}^{\text{sim}}$ and $\bar{I}_{6}^{\text{est}}$ increase. The indicators $\bar{I}_{2,ij}$ and $\bar{I}_{4,a}$ increase for the controller (c) since the lateral forces are even more over- and underestimated with tyre force coupling, and decrease for controllers (f), (n), and (k). The indicators $\bar{I}_{9,e}$, $\bar{I}_{10,e}$, $\bar{I}_{11,e}$, $\bar{I}_{13,e}$, and $\bar{I}_{14,e}$ decrease for controllers (c) and (f) and controllers (n) and (k) although the longitudinal slips on the front axle increase, i.e. $\bar{I}_{12,e}$. Switching from no coupling in (b) and (g) to linear coupling in (c) and (f) decreases the indicators $\bar{I}_{17,e}$ and $\bar{I}_{21,e}$, and increases the indicator $\bar{I}_{19,e}$. Opposite behaviour occurs switching to cosine coupling.

In conclusion to this comparison, the model of longitudinal and lateral tyre force coupling significantly influences the front-to-rear braking distribution of the MPC vehicle stability controller. The approximation of the tyre force coupling takes into account for the decrease of the lateral tyre force yaw moment contribution due to the direct yaw moment generated by the longitudinal tyre forces. A controller with a model of the tyre
force coupling generates direct yaw moments that actively influence the lateral force yaw moment contributions. It is, therefore, better performing, leading to reduced yaw velocity errors and rear slip angles even with reduced average control effort.

**Non-linear peak and stiffness factors**

For the influence of non-linear peak and stiffness in the lateral tyre force modelling of the prediction model, six controllers with linear tyre force coupling and varying vertical tyre loads are compared, i.e., the controllers (f), (h), (i), and (j) with non-linear lateral tyre force model and controllers (l) and (m) with saturation tyre force model. The lateral tyre forces non-linearly depend on the vertical loads and the influence can be described by the peak and stiffness factors in the tyre model. The predicted lateral tyre forces and their yaw moment contributions are higher with a linear peak and stiffness model compared to a non-linear peak and stiffness model. To compensate for this effect, the controllers with the linear peak and stiffness model are, therefore, more aggressive. The controllers (h), (i), (j), and (m) with non-linear formulations of the peak and stiffness, lead to increased yaw velocity errors and rear slip angles. The effects on the lateral tyre forces, lateral force yaw moment contributions, and in particular the lateral and combined friction utilizations are best visible in lower test runs for the non-linear stiffness in (i) and in medium and higher test runs for the non-linear peak in (h). The combination of both in (j) gives visually the best results, analogously to Fig. 5.3. For controller (m), the better estimation is apparent only for lower test runs due to the saturation tyre model. For controllers (h), (i), (j), and (m), the longitudinal slip ratios and friction utilizations on the front slightly increase due to the rear slip angle constraints.

For the performance indicators in Tables 5.4-5.6, the average direct yaw moment indicators \( \bar{I}_6^\text{sim} \) and \( \bar{I}_6^\text{est} \) decrease for all four controllers (h), (i), (j), and (m) with non-linear models, and the yaw velocity indicator, \( \bar{I}_7^\text{sim} \), and the rear slip angle indicators, \( \bar{I}_{8,a}^\text{sim} \), \( \bar{I}_{8,a}^\text{est} \), and \( \bar{I}_{8,b}^\text{sim} \), increase. For the indicators \( \bar{I}_{8,a}^\text{sim} \), \( \bar{I}_{8,a}^\text{est} \), and \( \bar{I}_{8,b}^\text{sim} \), there is more influence from the controller (h) compared to the controller (i). Similar behaviour can be observed for the maximum direct yaw moment indicators, \( \bar{I}_5^\text{sim} \) and \( \bar{I}_5^\text{est} \). The indicator \( I_{4,a} \) reduces for (h) in all test runs and for (i) in lower test runs. Indicators \( \bar{I}_{15,a}^\text{sim} \) and \( \bar{I}_{15,b}^\text{sim} \) increase with no significant changes in the indicators \( \bar{I}_{9,e}^\text{sim} \) to \( \bar{I}_{14,e}^\text{sim} \). Indicator \( \bar{I}_{19,e}^\text{sim} \) remains almost unchanged, while indicators \( \bar{I}_{17,e}^\text{sim} \) and \( \bar{I}_{21,e}^\text{sim} \) reduce. The indicators \( I_{17,e}^\text{sim} \) and \( I_{21,e}^\text{sim} \) reduce for (h) in all test runs, for (i) only in lower test runs, and in (j) and (m) even more in all test runs representing a good combination.

Concluding this comparison, the non-linear models of stiffness and peak factors estimate the lateral tyre forces and their yaw moment contributions slightly more accurately. The influence of the non-linear stiffness model is more significant for lower test runs implying lower values of tyre slip angles. Whereas the influence of the non-linear
peak model is more significant in higher test runs implying bigger load transfer effects. The controller with the non-linear stiffness and peak model has a reduced average control effort due to improved estimation of the lateral tyre forces. However, the performance enhancement is visible, but not significant and can, therefore, be omitted to reduce the complexity of the prediction model.

**Saturation in lateral tyre force model**

For the influence of saturation in the non-linear lateral tyre force model, four controllers with linear tyre force coupling and varying vertical loads are compared, i.e. (f) and (l) with linear peak and stiffness model, and (i) and (m) with linear peak and non-linear stiffness model. The lateral tyre forces and their yaw moment contributions are well or slightly underestimated for low values of the tyre slip angles, as shown in Fig. 5.2, but, however, considerably overestimated for higher values of the tyre slip angles using the saturation model. A controller with the simple non-linear saturation model is, therefore, more aggressive in particular in higher test runs to compensate for the overestimation of the lateral force yaw moment contributions. The controllers (l) and (m) result in reduced yaw velocity errors and rear slip angles for higher test runs. For lower test runs implying small tyre slip angles, the controller (m) has less influence than (l) compared to (f) and (i), due to the non-linear stiffness model. The controllers (l) and (m) have, compared to (f) and (i), higher estimates of lateral tyre forces, lateral force yaw moment contributions, and lateral and combined friction utilizations in high test runs implying higher values of tyre slip angles. The opposite behaviour occurs for lower test runs.

Considering the performance indicators in Tables 5.4-5.6, the maximum direct yaw moment indicators, $\vec{I}_5^{sim}$ and $\vec{I}_5^{est}$, decrease for controllers (l) and (m), together with the yaw velocity indicator, $\vec{I}_7^{sim}$, and the rear slip angle indicators, $\vec{I}_{8,a}^{sim}$, $\vec{I}_{8,a}^{est}$, and $\vec{I}_{8,b}^{sim}$. The average direct yaw moment indicators, $\bar{I}_6^{sim}$ and $\bar{I}_6^{est}$, increase, while the indicators $\bar{I}_6^{sim}$ and $\bar{I}_6^{est}$ increase especially for high test runs. The indicators $\bar{I}_{9,e}^{sim}$, $\bar{I}_{10,e}^{sim}$, $\bar{I}_{12,e}^{sim}$, $\bar{I}_{13,e}^{sim}$, and $\bar{I}_{14,e}^{sim}$ decrease for (l) and (m), while $\bar{I}_{11,e}^{sim}$ slightly increases. The lateral displacements, $\bar{I}_{15,a}^{sim}$ and $\bar{I}_{15,b}^{sim}$, decrease and indicators $\bar{I}_{17,e}$, $\bar{I}_{19,e}$, and $\bar{I}_{21,e}$ increase with an increase of $I_{17,e}$ and $I_{21,e}$ for the controller (m) in lower runs and for the controller (l) in medium and higher runs.

Concluding this comparison, the non-linear lateral tyre force model with saturation based on the hyperbolic tangent overestimates the lateral tyre forces for high tyre slip angles. Whereas the non-linear lateral tyre force model based on the simplified version of the Pacejka Magic Formula gives a better estimation of the lateral tyre forces over the whole range of tyre slip angles. Since the controller with the saturation tyre model overestimates the lateral tyre forces and their yaw moment contributions with increasing tyre slip angles, it acts, therefore, more aggressive to compensate. It is possible to
5.3 Performance analysis

identify this effect, but the influence is, however, not significant. The simpler non-linear saturation model could, therefore, be used as an alternative reducing the complexity of the prediction model.

5.3.7 Sensitivity analysis and robustness

With respect to the closed-loop stability of the non-linear MPC law defined by (5.61)-(5.62), common schemes in the literature for implicit methods include a stabilising terminal constrained set and a terminal cost, which needs to satisfy Lyapunov function-type conditions, as elaborated in more detail in Mayne et al. (2000) and Chen and Allgöwer (1998). Alternatively, Grüne (2009) and Reble and Allgöwer (2012) present a stability and performance analysis for unconstrained, in the sense of stability preserving constraints, implicit non-linear MPC schemes. However, the mentioned approaches are for implicit NMPC schemes. To the best of the author’s knowledge, there is no comparable practical NMPC theory in the literature addressing the stability of explicit NMPC.

In this study, the stability and robustness of the explicit NMPC controller are, therefore, empirically assessed through a sensitivity analysis varying some parameters of the simulation model. Starting from the nominal parameters for the vehicle mass \( m = 1962 \text{ kg} \), the vehicle yaw mass moment of inertia \( I_z = 3382 \text{ kg m}^2 \), and the tyre-road friction coefficient \( \mu = 0.8 \), each parameter is increased and decreased by 15\% and 20\%, respectively. In Table 5.7, some selected performance indicators are shown over the series of 22 Sine with Dwell test runs of the explicit controller (f) with Setting A, as defined in Table 5.2, with the lower bounds in (5.53) and constraints (5.58) enabled (even if they are not active).

With regards to the variation of the tyre-road friction coefficient, \( \mu \), the yaw velocity errors and rear slip angles decrease with increasing friction coefficient in low runs and increase for medium and high runs. The yaw velocity error and rear slip angle indicators, i.e. \( \bar{I}_{7}\text{sim}, \bar{I}_{7}\text{est}, \bar{I}_{8,a}\text{sim}, \bar{I}_{8,a}\text{est}, \bar{I}_{8,b}\text{sim}, \bar{I}_{8,b}\text{est} \) shift to higher runs, in particular, due to the reaction related to the rear slip angle constraints. In medium and high runs, the direct yaw moment generation and longitudinal tyre forces, respectively, considerably increase for higher friction coefficients, and, therefore, speed is reduced. In low runs, an opposite behaviour can be observed. This is also reflected in an increase of the direct yaw moment indicators, \( \bar{I}_{5}\text{sim}, \bar{I}_{5}\text{est}, \bar{I}_{6}\text{sim}, \bar{I}_{6}\text{est} \), and \( \bar{I}_{6}\text{est} \), and the longitudinal slip indicator, \( \bar{I}_{11,e}\text{sim} \). Moreover, the overall yaw moment significantly increases with higher friction coefficients, indicated by \( \bar{I}_{3}\text{sim} \) and \( \bar{I}_{3}\text{est} \). An apparent influence on the lateral displacement, \( \bar{I}_{15,a}\text{sim} \) and \( \bar{I}_{15,a}\text{est} \), can also be observed. The lateral and maximum longitudinal tyre forces, lateral and combined friction utilisations, and lateral force yaw moment contributions are over- and underestimated for low and high friction, respectively. This leads to an increase of indicators such as \( \bar{I}_{4,a}, \bar{I}_{17,e} \), and
Table 5.7: Selected average performance indicators for a series of 22 SwD test runs with explicit NMPC controller (f) using Setting A.

<table>
<thead>
<tr>
<th>Parameter variation</th>
<th>Total yaw moment</th>
<th>Yaw moment errors</th>
<th>Maximum direct yaw moment</th>
<th>Average direct yaw moment</th>
<th>Yaw velocity error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$I_{3}^{\text{sim}}$ (N m)</td>
<td>$I_{3}^{\text{est}}$ (N m)</td>
<td>$I_{4,a}$ (N m)</td>
<td>$I_{4,b}$ (N m)</td>
<td>$I_{5}^{\text{sim}}$ (N m)</td>
</tr>
<tr>
<td>nominal</td>
<td>2053</td>
<td>2122</td>
<td>1925</td>
<td>749</td>
<td>3845</td>
</tr>
<tr>
<td>$\mu$ +20%</td>
<td>2317</td>
<td>2470</td>
<td>3027</td>
<td>1234</td>
<td>4248</td>
</tr>
<tr>
<td></td>
<td>1748</td>
<td>1936</td>
<td>2451</td>
<td>630</td>
<td>2964</td>
</tr>
<tr>
<td>$I_{z}$ +15%</td>
<td>2230</td>
<td>2255</td>
<td>1869</td>
<td>739</td>
<td>3760</td>
</tr>
<tr>
<td></td>
<td>1872</td>
<td>1989</td>
<td>1990</td>
<td>759</td>
<td>3938</td>
</tr>
<tr>
<td>$m$ +15%</td>
<td>2189</td>
<td>2079</td>
<td>2961</td>
<td>917</td>
<td>3996</td>
</tr>
<tr>
<td></td>
<td>1911</td>
<td>2242</td>
<td>1920</td>
<td>629</td>
<td>3634</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter variation</th>
<th>Rear slip angle</th>
<th>Average lateral, longitudinal, and combined slip displacement</th>
<th>Lateral, longitudinal, and combined friction utilisation error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$I_{8,a}^{\text{sim}}$ (deg)</td>
<td>$I_{8,a}^{\text{est}}$ (deg)</td>
<td>$I_{8,b}^{\text{sim}}$ (deg)</td>
</tr>
<tr>
<td>nominal</td>
<td>0.076</td>
<td>0.056</td>
<td>0.052</td>
</tr>
<tr>
<td>$\mu$ +20%</td>
<td>0.141</td>
<td>0.100</td>
<td>0.090</td>
</tr>
<tr>
<td></td>
<td>0.067</td>
<td>0.054</td>
<td>0.051</td>
</tr>
<tr>
<td>$I_{z}$ +15%</td>
<td>0.090</td>
<td>0.069</td>
<td>0.065</td>
</tr>
<tr>
<td></td>
<td>0.064</td>
<td>0.045</td>
<td>0.041</td>
</tr>
<tr>
<td>$m$ +15%</td>
<td>0.096</td>
<td>0.069</td>
<td>0.062</td>
</tr>
<tr>
<td></td>
<td>0.059</td>
<td>0.044</td>
<td>0.041</td>
</tr>
</tbody>
</table>
In non-nominal conditions. Concluding these results, the controller (f) is robust against variations in the tyre-road friction coefficient with significant but acceptable influence on the yaw velocity and rear slip angle response.

For the variation of the yaw mass moment of inertia, \(I_z\), the yaw velocity response gets slower with increasing values of \(I_z\) leading to less yaw velocity overshoot but slightly increased rear slip angles. The indicators \(\bar{I}^\text{sim}_{7}, \bar{I}^\text{sim}_{8,a}, \bar{I}^\text{est}_{8,a}, \text{ and } \bar{I}^\text{sim}_{8,b}\) for the yaw velocity errors and rear slip angles increase. Due to the rear slip angle constraints, the direct yaw moment generation and longitudinal tyre forces, respectively, are increased and lead to reduced speed. The average direct yaw moment indicators, \(\bar{I}^\text{sim}_{6}\) and \(\bar{I}^\text{est}_{6}\), as well as the overall yaw moment indicators, \(\bar{I}^\text{sim}_{3}\) and \(\bar{I}^\text{est}_{3}\), increase. However, the maximum direct yaw moment indicators, \(\bar{I}^\text{sim}_{5}\) and \(\bar{I}^\text{est}_{5}\), decrease. Moreover, the indicators for the slips, \(\bar{I}^\text{sim}_{9,e}\) and \(\bar{I}^\text{sim}_{13,e}\), except of \(\bar{I}^\text{sim}_{11,e}\), the indicators for the friction utilisations, \(\bar{I}_{17,e}, \bar{I}_{19,e}, \text{ and } \bar{I}_{21,e}\), and the indicators for the lateral displacement, \(\bar{I}^\text{sim}_{15,a}\) and \(\bar{I}^\text{sim}_{15,b}\), decrease. In summary, the controller (f) is robust against variations in the vehicle yaw mass moment of inertia with visible albeit not significant influence on the performance.

By varying the vehicle mass, \(m\), the yaw velocity response is slightly faster with increasing values of the mass. The yaw velocity errors and rear slip angles decrease with increasing mass in low runs and increase in medium and high runs. The rear slip angle indicators, \(\bar{I}^\text{sim}_{8,a}, \bar{I}^\text{est}_{8,a}, \text{ and } \bar{I}^\text{sim}_{8,b}\), increase, while the yaw velocity error indicator, \(\bar{I}^\text{sim}_{7}\), slightly decreases. The indicators \(\bar{I}^\text{sim}_{5}, \bar{I}^\text{est}_{5}\), and \(\bar{I}^\text{sim}_{8,a}\) shift to higher runs with increasing vehicle mass. In medium and high runs, the direct yaw moment generation and longitudinal tyre forces, respectively, increase as reflected by the indicators \(\bar{I}^\text{sim}_{5}\) and \(\bar{I}^\text{est}_{5}\) are shifted to higher runs. However, the vehicle speed is also increased. For the overall yaw moment indicators, \(\bar{I}^\text{sim}_{3}\) increases and \(\bar{I}^\text{est}_{3}\) decreases. Moreover, a decrease in the lateral displacement indicators, \(\bar{I}^\text{sim}_{15,a}\) and \(\bar{I}^\text{sim}_{15,b}\), can be observed. The vertical, lateral, and maximum longitudinal tyre forces, lateral and combined friction utilisations, and lateral force yaw moment contributions are over- and underestimated for low and high vehicle mass, respectively. This is also reflected by an increase of \(\bar{I}_{17,e}\) and \(\bar{I}_{21,e}\) in non-nominal conditions. Indicators \(\bar{I}^\text{sim}_{9,e}\) and \(\bar{I}^\text{sim}_{13,e}\) increase, whereas \(\bar{I}^\text{sim}_{11,e}\) and \(\bar{I}_{19,e}\) decrease. In conclusion, the controller (f) is robust against variations in the vehicle mass with visible albeit not significant influence on the performance.

5.3.8 Sampling time and execution time

To investigate the influence of the sampling time on the controller performance, the controller (f) is simulated in test run 12 over a range of different sampling times with a fixed discretisation time \(H = 20\) ms. The smallest sampling time is \(T_s = 2\) ms with an increment of 2 ms up to a sampling time \(T_s = 10\) ms, followed by an increment of
10 ms up to a sampling time $T_s = 150$ ms. Moreover, to investigate the influence of the execution time on the controller performance, the NMPC solution is subjected to a transportation delay equal to the sampling time. For controller (f) with delay, the solution is, therefore, assumed to be available only at the end of the sampling interval at the next sampling instant. However, for the controller (f) without delay, the solution is assumed to be available at the same sampling instant of the controller. This study is motivated by the fact that an implicit solution of the NMPC leads to non-trivial execution time and, thus, can only be applied at the next time instant. On the contrary, the explicit solution requires significantly lower execution times and can, therefore, be assumed to be instantaneously available. An implementation of the explicit solution of the wheel slip controllers in Chapter 6 on a dSPACE rapid control prototyping unit shows execution times in the range of 5 $\mu$s to 25 $\mu$s for the controller of Section 6.1 and execution times $< 95$ $\mu$s for the controller of Section 6.3 justifying these assumptions with respect to this study.

For controller (f) without delay, only a minimal increase of the yaw velocity error can be observed for higher sampling times. This can also be identified from performance indicator $I^{\text{sim}}_7$ in Fig. 5.25. Generally, there is only a small influence of the sampling time on the performance of controller (f) without delay in nominal conditions. In contrast, the controller (f) with delay exhibits significantly increased yaw velocity errors, both in terms of average and peak values, especially for sampling times higher than 60 ms, as shown in Fig. 5.25. Likewise, the rear slip angles, indicated by $I^{\text{sim}}_{\delta,a}$, $I^{\text{est}}_{\delta,a}$, and $I^{\text{sim}}_{\delta,b}$, significantly increase for higher sampling times with a slight reduction of rear slip angle peaks for medium sampling times. For controller (f) with delay, higher sampling times lead to significantly increased longitudinal tyre forces and, therefore, increased longitudinal slip ratios and friction utilisations, and increased maximum and average direct yaw moments. The average direct yaw moment indicators, $I^{\text{sim}}_6$ and $I^{\text{est}}_6$, are shown in Fig. 5.25 together with the indicators $I^{\text{sim}}_3$, $I^{\text{est}}_3$, $I_{4,a}$, and $I_{4,b}$. For all performance indicators, a similar trend can be observed, such as the moderate influence of the sampling time $T_s$ up to 60 ms to 80 ms and a very significant influence for higher sampling times. Generally, there is a considerable influence of the sampling time on the performance of controller (f) with delay in nominal conditions, notably also on the trajectory of the vehicle. For the selected sampling time $T_s = 20$ ms, reduced yaw velocity errors, longitudinal forces, slip ratios and friction utilisations, and maximum direct yaw moments for the controller (f) without delay can be observed.

Concluding this comparison, the influence of additional time delays is more significant than the influence of the sampling time on the controller performance in nominal conditions, especially for high sampling times. However, the performance regarding disturbance rejection generally considerably increases with smaller sampling times. It is important to reduce any additional delays coming from the non-trivial execution times
5.3 Performance analysis

Fig. 5.25: Comparison of selected performance indicators of controller (f) in run 12 over a range of different sampling times demonstrating the influence of delays due to non-trivial execution times. Solid lines refer to the indicators from the high-fidelity simulation model and dash-dotted lines refer to the indicators from the prediction model.
of the NMPC controller and in this regard, an explicit solution is the preferred method compared to an implicit approach.

5.3.9 Explicit solution

Parameter exploration space

For the execution of the Algorithm 1, the hyper-rectangle $X = \{ x_p \in \mathbb{R}^{n_p} \mid A_{th} x_p \leq b_{th} \}$ of the parameter space to be explored needs to be defined. Moreover, as explained in more detail in Section 4.4.4, the parameter vector $x_p(t_k)$ is clipped yielding the vector $\bar{x}_p(t_k) = \min(\bar{x}_p, \max(x_p, x_p(t_k)))$ in case it lies outside the hyper-rectangle $X$ for the parameter space exploration with $\bar{x}_p$ and $\bar{x}_p$ as the lower and upper bounds, respectively, defining the closed poly-topic set of parameters. Therefore, the choice of the bounds has significant influence not only on the complexity of the controller, but also on the dynamic behaviour.

For example, reduced parameter bounds on the yaw velocity error lead to increased yaw velocity errors and consequently to increased rear slip angles in medium and high runs. The direct yaw moment generation is decreased due to the yaw velocity error constraints and increased due to the rear slip angle constraints. In conclusion, the saturation of the yaw velocity error generally makes the controller less aggressive.

Further, a reduction of the parameter bounds on the rear slip angle leads to increased rear slip angles and consequently to increased yaw velocity errors in particular in medium runs. The direct yaw moment generation is decreased due to the rear slip angle constraints and increased due to the yaw velocity error constraints. Concluding, the saturation of the rear slip angle generally makes the controller less aggressive.

Finally, reduced parameter bounds on the steering angle lead to decreased yaw velocity errors and rear slip angles for runs with medium and high steering wheel angle amplitudes (refer to Fig. 5.5). The direct yaw moment generation is increased due to the yaw velocity error constraints and decreased due to the rear slip angle constraints. In conclusion, the saturation of the steering angle generally makes the controller more aggressive.

However, the effects of yaw velocity error and rear slip angle saturation and the effects of steering angle saturation have the tendency to compensate.

For the case without symmetry detection, as shown in Fig. 5.26 in the left sub-plot, only symmetric bounds on the parameters in the yaw velocity error and rear slip angle domain are an appropriate choice. However, with symmetry detection, as explained in Section 5.2.2, it is reasonable to define asymmetric bounds on the parameters. This allows different values for the lower and upper bounds, respectively. In addition to the complexity reduction due to bisection based on the symmetry exploitation, asymmetric bounds help to further reduce the complexity of the explicit controller. Based on the
dynamic response, as shown with symmetry detection in Fig. 5.26 in the right sub-plot, reasonable values for the parameter bounds are obtained: Lower bound on the yaw velocity error $\bar{\epsilon}_\psi = -18 \text{deg/s}$, upper bound on the yaw velocity error $\bar{e}_\psi = 24 \text{deg/s}$, lower bound on the rear slip angle $\bar{\alpha}_R = -5.5 \text{deg}$, and upper bound on the rear slip angle $\bar{\alpha}_R = 7.0 \text{deg}$. The steering angle with symmetry detection is lower bounded with $\bar{\delta} = 0 \text{deg}$ and upper bounded with $\bar{\delta} = 8 \text{deg}$.

In addition to the effects of parameter saturation described above, a discontinuity in the control action due to the asymmetric bounds on the yaw velocity error can be observed. However, with the suitable choice of asymmetric bounds, the influence of the yaw velocity error saturation is significantly reduced and no influence of the rear slip angle saturation is visible. The effects of steering angle saturation are visible and slightly more significant in higher runs compared to the yaw velocity error saturation being more significant in medium runs. The controller with saturated mp-NLP parameters and asymmetric bounds is in general marginally more aggressive with, however, minimal observable influence of the parameter saturation.

**Algorithm and post-processing**

In Table 5.9, the explicit controller (f) with Setting A, as defined in Table 5.2, is shown with different parameter vectors and varying bounds on the parameter exploration space.
The lower bounds in (5.53) and the constraints (5.58) are disabled, and for the parameter vector \( x_p = [\dot{\psi}, \alpha_R, v, \delta]^T \) also constraints (5.54) and (5.57) are deactivated.

The approximation tolerances for the partitioning algorithm 1 are chosen \( \tilde{\varepsilon} = 5 \) for the cost error, \( \tilde{\rho} = [0.1, 0.1, 0.1, 0.1]^T \) for the solution errors, and \( \delta = 0.15 \) for the maximum constraint violation. The weighting vector \( w_\rho \) extracts only the first elements associated with the receding horizon control law of the four inputs and the weighting vectors \( w_\delta \) and \( w_\deltae \), respectively, are chosen to equally weight all constraint violations. Moreover, an appropriate minimal allowed volume \( S_{min} \) for the partitions in the hyper-rectangle \( \mathbb{X} \) is defined as \( S_{min} = \prod_{i=1}^{n_p} (1/32) (\bar{x}_{p,i} - \bar{x}_{p,i}) \). This volume would result after five splits of the hyper-rectangle \( \mathbb{X} \) by hyperplanes through the Chebyshev centre and orthogonal to every axis. The number of internal hyper-rectangles is chosen \( N_0 = 1 \). Therefore, for the generation of the set of test points, the vertices of the hyper-rectangles \( X_0 \) and \( X_0^1 \), respectively, are included in the set of vertices. For the hyper-rectangle \( X_0 \), the centres of the edges are additionally considered in the set of generated points. The hyper-rectangles \( \mathbb{X} \) to be explored are defined in Table 5.9.

### Table 5.8: Definition of post-processing settings.

<table>
<thead>
<tr>
<th>Case</th>
<th>Clipping-based complexity reduct.</th>
<th>Optimal disjoint merging procedure</th>
<th>Sub-optimal merging procedure</th>
<th>Binary search tree generation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>clipping</td>
<td>optimal</td>
<td>greedy</td>
<td>bst</td>
</tr>
<tr>
<td>(a)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>(b)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>(c)</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(d)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

For the post-processing algorithm 4, different settings are defined in Table 5.8 with, however, the sanity checks (indicated by the flag check) active for all cases. To demonstrate the post-processing without complexity reduction and efficient search data structures, the flags clipping, optimal, greedy, and bst are disabled for case (a). These settings will lead to sequential search algorithms in every partition. To see the influence of efficient search data structures, the generation of memory-optimised binary search trees with flag bst is enabled in case (b). In case (c), the merging procedures based on the disjoint optimal approach with flag optimal and the sub-optimal approach with flag greedy are enabled in addition to the binary search tree generation. Finally, the influence of the clipping-based complexity reduction can be observed in case (d) with flag clipping active. The output of the post-processing algorithm 4 in terms of the number of polyhedral regions, the size of the bottom layer file, and required execution time for the algorithm is indicated in Table 5.9 for the different settings of the cases (a)-(d).
Table 5.9: Partitioning algorithm and post-processing data for the explicit NMPC controller (f) using Setting A. The approximation tolerances are $\tilde{\varepsilon} = 5$ for the cost error, $\tilde{\rho} = [0.1, 0.1, 0.1, 0.1]^T$ for the solution error, and $\tilde{\delta} = 0.15$ for the maximum constraint violation.

<table>
<thead>
<tr>
<th>Case</th>
<th>Parameter vector</th>
<th>Yaw velocity error bounds</th>
<th>Rear slip angle bounds</th>
<th>Speed bounds</th>
<th>Steering angle bounds</th>
<th>Deceleration bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_p$</td>
<td>$\bar{e}_\psi$ (deg/s)</td>
<td>$\bar{e}_\psi$ (deg/s)</td>
<td>$\alpha_R$ (deg)</td>
<td>$\bar{v}$ (km/h)</td>
<td>$\bar{v}$ (km/h)</td>
</tr>
<tr>
<td>(1)</td>
<td>$e_\psi, \alpha_R, v, \delta$</td>
<td>-24</td>
<td>24</td>
<td>-7.0</td>
<td>7.0</td>
<td>65</td>
</tr>
<tr>
<td>(2)</td>
<td>$e_\psi, \alpha_R, v, \delta$</td>
<td>-24</td>
<td>-24</td>
<td>-5.5</td>
<td>7.0</td>
<td>65</td>
</tr>
<tr>
<td>(3)</td>
<td>$e_\psi, \alpha_R, v, \delta$</td>
<td>-18</td>
<td>24</td>
<td>-5.5</td>
<td>7.0</td>
<td>65</td>
</tr>
<tr>
<td>(4)</td>
<td>$e_\psi, \alpha_R, v, \delta$</td>
<td>-18</td>
<td>-24</td>
<td>-5.5</td>
<td>7.0</td>
<td>65</td>
</tr>
<tr>
<td>(5)</td>
<td>$e_\psi, \alpha_R, v, \delta$</td>
<td>-18</td>
<td>-24</td>
<td>-5.5</td>
<td>7.0</td>
<td>55</td>
</tr>
<tr>
<td>(6)</td>
<td>$e_\psi, \alpha_R, v, \delta$</td>
<td>-18</td>
<td>24</td>
<td>-5.5</td>
<td>7.0</td>
<td>65</td>
</tr>
<tr>
<td>(7)</td>
<td>$e_\psi, \alpha_R, v, \delta, a_{x,ref}$</td>
<td>-18</td>
<td>24</td>
<td>-5.5</td>
<td>7.0</td>
<td>65</td>
</tr>
<tr>
<td>(8)</td>
<td>$e_\psi, \alpha_R, v, \delta, a_{x,ref}$</td>
<td>-18</td>
<td>24</td>
<td>-5.5</td>
<td>7.0</td>
<td>65</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Orthogonal partitions</th>
<th>Polyhedral regions</th>
<th>Size top layer</th>
<th>Size bottom layer</th>
<th>Algorithm execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a) / (b) / (c) / (d)</td>
<td>(-) / (-)</td>
<td>(kB) / (kB)</td>
<td>(1) / 4 (b) / 4 (c) / 4 (d)</td>
<td>(h) / (h) / (h) / (h)</td>
</tr>
<tr>
<td>(1)</td>
<td>233</td>
<td>2621 / 2621 / 2128 / 2180</td>
<td>9 / 1420 / 1044 / 858 / 858</td>
<td>1.58 / 0.24 / 0.68 / 0.93</td>
<td></td>
</tr>
<tr>
<td>(2)</td>
<td>206</td>
<td>2163 / 2163 / 1779 / 1829</td>
<td>8 / 1176 / 846 / 804 / 815</td>
<td>1.20 / 0.21 / 0.47 / 0.62</td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td>104</td>
<td>2386 / 2386 / 2060 / 2070</td>
<td>5 / 1380 / 1049 / 974 / 977</td>
<td>2.09 / 0.41 / 0.67 / 0.94</td>
<td></td>
</tr>
<tr>
<td>(4)</td>
<td>296</td>
<td>4169 / 4169 / 3438 / 3333</td>
<td>11 / 2322 / 1723 / 1523 / 1470</td>
<td>3.47 / 0.70 / 0.87 / 1.35</td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td>134</td>
<td>2502 / 2502 / 2160 / x</td>
<td>6 / 1419 / 1021 / 964 / x</td>
<td>1.67 / 0.43 / 0.70 / x</td>
<td></td>
</tr>
<tr>
<td>(6)</td>
<td>195</td>
<td>3762 / 3762 / 3074 / x</td>
<td>8 / 2108 / 1611 / 1441 / x</td>
<td>3.43 / 0.93 / 1.29 / x</td>
<td></td>
</tr>
<tr>
<td>(7)</td>
<td>174</td>
<td>20 943 / 20 943 / x / x</td>
<td>8 / 19 111 / 21 527 / x / x</td>
<td>7.86 / 15.13 / x / x</td>
<td></td>
</tr>
</tbody>
</table>
For the explicit controllers in Table 5.9, the maximum estimated error bounds on the cost function and the solution, as well as the maximum constraint violation, fulfil the specified approximation tolerances in all hyper-rectangles $X$. It is interesting to observe that the average approximation errors are considerably smaller than the respective maximum values. Moreover, the minimum volumes are far from the specified minimum allowed volume. In Table 5.9, the influence of the parameter bounds on the complexity of the explicit controller in terms of polyhedral critical regions and memory requirements is shown. For example, comparing the controllers (1), (2), and (3), the beneficial influence of appropriately chosen asymmetric parameter bounds on the yaw velocity error and the rear slip angle, identified as in Section Parameter exploration space, can be observed. Moreover, the significant benefits of the symmetry detection becomes evident by comparing controllers (3) and (4). Exploiting the symmetry allows bisection of the parameter exploration space consequently leading to a reduction of the number of polyhedral critical regions by almost a half. Comparing controllers (3), (5), and (6), a considerable influence of the parameter bounds corresponding to the speed on the complexity of the controller becomes apparent. However, the most significant influence can be observed considering an additional parameter in the parameter vector $x_p$ by comparing the controllers (1)-(6) with the controllers (7)-(8). With increasing complexity of the controller, the size of the top and bottom layer increases. An implementation of the explicit solution on a rapid control prototyping unit, such as the dSPACE MicroAutoBox II with 16 MB main memory used for the deployment of the wheel slip controller in Chapter 6, would be, therefore, only possible for controllers (1)-(6). Also, an increase of the algorithm execution times, obtained on an Intel Core i7-4770 CPU at 3.40 GHz, can be observed.

For the post-processing algorithm 4, the generation of memory-optimised binary search trees in case (b) considerably reduces the memory requirements of controllers (1)-(6) compared to case (a) by up to 28%. For controllers (7) and (8), the memory requirements increase, however, for case (b). This is due to a number of polyhedral regions in the individual orthogonal partitions in the hundreds to thousands, compared to a number in the tens to hundreds for controllers (1)-(6), increasing the memory requirements and in particular the algorithm execution times significantly. For controllers (7) and (8), the computation of the sanity checks also contribute to a considerable increase of the algorithm execution times. A reduction of complexity in terms of polyhedral regions can be achieved by the disjoint optimal and the sub-optimal merging procedure active in case (c) by up to 19% compared to case (a). However, as seen from the numbers in Table 5.9, the potential of the merging produce is generally limited with an increasing number of inputs since it becomes less likely that neighbouring polyhedra have the same receding horizon feedback law and, moreover, form convex unions. A similar effect can be observed in case (d), were joint saturation becomes less likely with an increasing
number of system inputs reducing the effectiveness of the clipping-based complexity reduction. The number of polyhedral regions is reduced by up to 20% compared to case (a). Case (d) yields, however, less reductions for controllers (1)-(3) compared to case (c). For controller (7), the large number of polyhedral regions in some orthogonal partitions cause the termination of the algorithm, marked with x, due to convergence problems in case (c). For the clipping-based complexity reduction in case (d), also controllers (5) and (6) are affected.

In conclusion, the number of parameters has the most significant influence on the complexity of the explicit solution. With the exploitation of symmetry properties and an appropriate choice of the bounds on the parameters, in particular, using asymmetric bounds, the complexity can be considerably reduced. Moreover, the post-processing allows to further simplify the explicit solution (by up to 40% in terms of memory requirements and up to 20% in terms of polyhedral regions) employing complexity reduction approaches and efficient search data structures.

In Fig. 5.27, the orthogonal partitions of the top layer and polyhedral critical regions of the bottom layer of the explicit controller (f)-(5) with parameter \( x_{p,3} = \frac{v}{P_{ch,v}} \) fixed to 1 corresponding to a speed \( v = 80 \text{ km/h} \).

Fig. 5.27: Orthogonal partitions of the top layer and polyhedral critical regions of the bottom layer of the explicit controller (f)-(5) with parameter \( x_{p,3} = \frac{v}{P_{ch,v}} \) fixed to 1 corresponding to a speed \( v = 80 \text{ km/h} \).
Fig. 5.27, the pattern of the polyhedral critical regions reflects the constraints on the rear slip angle and yaw velocity error.

In Fig. 5.28, the feedback law of explicit controller (f)-(5) over the normalised yaw velocity error and rear slip angle parameter space is shown. The parameters $x_{p,3} = v / P_{ch,v}$ are fixed to 1 corresponding to a speed $v = 80\,\text{km/h}$ and $x_{p,4} = \delta / P_{ch,\delta}$ fixed to 0 corresponding to a steering angle $\delta = 0\,\text{deg}$. The explicit receding horizon feedback law is continuous on each orthogonal partition, defined by the piecewise affine solution of the local mp-QP sub-problem. However, it may be discontinuous across the
orthogonal partitions. This can also be observed in Fig. 5.28 showing the dimensionless inputs $\tilde{u}(t_k)$, which need to be converted to dimension-related quantities by applying the corresponding transformation laws of the non-dimensionalisation procedure.

A comparison of the implicit and explicit implementation of the NMPC-based vehicle stability controller shows only marginal differences in the control action. More importantly, there is no observable difference in the vehicle response. The difference in results between implicit NMPC and explicit NMPC can be influenced by choosing the tolerances on the respective error bounds of the partitioning algorithm 1, to find a balance between reasonable levels of sub-optimality and moderate complexity of the explicit controller.

5.4 Conclusion and future work

In this chapter, the design, implementation, and performance assessment of an explicit non-linear MPC for yaw and lateral stability control of a vehicle with an electro-hydraulic braking system is presented. Moreover, the influence of varying prediction model fidelity on the performance of the controller is systematically investigated. Finally, the robustness of the controller against parameter variations and the influence of the controller sampling time and additional delays in the solution is analysed.

Conclusion  The simulation results demonstrate the effectiveness of the explicit NMPC approach to vehicle stability control, which brings significant improvements in the vehicle response in Sine with Dwell tests. The controlled vehicle remains within the specified thresholds and fulfils the performance requirements of regulation UN/ECE-R 13H (1958).

An analysis of different tuning settings and weights demonstrates the influence on the controller performance and the importance of the inclusion of rear slip angle constraints. Moreover, the beneficial influence of longitudinal tyre force constraints in limit conditions is shown and, further, the influence of the direct yaw moment reference on the performance of the controller is analysed.

The systematic analysis of the prediction model complexity concludes that a non-linear lateral tyre force model is necessary for an acceptable behaviour of an MPC-based vehicle stability controller operating at the limits of handling. Moreover, the modelling of load transfer effects and the incorporation of the side-slip angle rate in the estimation of the vertical tyre forces is important for an accurate prediction of the lateral tyre forces and their yaw moment contributions. The modelling of longitudinal and lateral tyre force coupling, with a simple linear model yielding acceptable results, significantly influences the front-to-rear distribution of the longitudinal tyre forces leading to improved vehicle response with reduced average control effort. The non-linear modelling of stiffness and peak factors in the lateral tyre force model brings performance benefits that are, however,
not significant and can, therefore, be omitted to reduce the complexity of the prediction model. Finally, a simple non-linear model with saturation based on the hyperbolic tangent gives reasonable results and can be used as an alternative to a more complex non-linear model based on a simplified version of the Pacejka Magic Formula.

A sensitivity analysis demonstrates the robustness of the explicit NMPC stability controller against parameter variations of the controlled vehicle. Variations in the tyre-road friction coefficient lead to significant but acceptable influence in the vehicle response, while variations in the vehicle mass properties, such as mass and mass moment of inertia, lead to an insignificant influence on the performance.

Further analysis shows that the influence of additional delays in the solution of the non-linear MPC stability controller in nominal conditions is more significant than the influence of the sampling time on the controller performance. In this respect, an explicit approach to non-linear MPC is, therefore, the preferred choice compared to an implicit approach due to reduced online execution times.

The influence of an appropriate choice of parameter bounds and subsequent saturation of parameters such as yaw velocity error, rear slip angle, and steering angle on the explicit solution is investigated. Moreover, the beneficial influence of asymmetric parameter bounds and the significant influence of symmetry exploitation reducing the complexity of the explicit solution is demonstrated. Further analysis shows a very significant influence of the number of parameters on the controller complexity in terms of polyhedral critical regions and subsequently memory requirements. However, the proposed post-processing with complexity reduction approaches and memory-optimised search data structures can simplify the explicit solution, although the potential of the merging and clipping-based complexity reduction methods is limited mainly due to the high number of inputs.

The explicit approach facilitates the verification of the solution prior to the safety-critical application on the vehicle. Moreover, the explicit NMPC approach does not imply any performance decay with respect to the corresponding implicit NMPC in terms of vehicle response. However, the explicit approach to non-linear MPC for vehicle stability control leads to considerable complexity of the controller depending on the formulation in terms of parameters. A feasible real-time implementation can, therefore, generally only be considered on embedded hardware with suitable memory specifications.

**Future work** Future work will focus on practical concerns on the deployment of the explicit controller on a vehicle demonstrator. This includes the development of estimation techniques for the vehicle side-slip angle, vehicle speed, and the tyre-road friction coefficient to meet the practical requirements for the implementation of the explicit NMPC-based stability controller. A further interesting aspect is a comparison with stability control systems based on classical control structures.
Chapter 6

Wheel slip control

6.1 Traction control

6.1.1 Introduction and literature review

The adoption of electric drivetrains, and in particular of in-wheel motor layouts, has the potential of significantly enhancing the performance of wheel slip control systems, i.e. anti-lock braking systems (ABS) and traction control (TC) systems, Fujimoto et al. (2012). This is caused by the higher control bandwidth and precision in torque modulation that electric drivetrains can offer, with respect to more conventional internal combustion engines and hydraulic/electro-hydraulic braking units. Murata (2012) and Ivanov et al. (2015a) include experimentally measured reductions in stopping distances and acceleration times, achieved through the continuous modulation of the electric drivetrain torques. However, further work can be done in terms of control design to enhance the slip ratio tracking performance and the seamless blending of the regenerative and dissipative braking contributions.

In parallel to sliding mode control (Amodeo et al., 2010) and maximum transmissible torque estimation (Yin and Hori, 2010) algorithms, the recent literature on the topic of ABS and TC shows growing interest in model-based control, with focus on model predictive control. For example, Johansen et al. (2003) discuss a gain scheduled linear-quadratic regulator (LQR) approach for ABS control, with experimental results on an internal-combustion-engine-driven vehicle with electro-mechanical brakes. Boisvert et al. (2014) and Anwar and Ashrafi (2002) include different approaches to ABS control, i.e. linear-quadratic Gaussian (LQG) regulation and generalised predictive control, which is re-proposed in Anwar (2003) for a TC implementation. A linear MPC strategy is developed in de Castro et al. (2012), where the ABS slip regulation is achieved through torque blending between the friction brakes and in-wheel motors. Similarly, Satzger et al. (2016), Satzger and de Castro (2014), and Satzger and De Castro (2018) combine
ABS control and torque blending, by using linear MPC formulations. Yoo and Wang (2007) present an MPC-based ABS, with test results on a hardware-in-the-loop (HIL) rig. The internal model includes a tyre force dynamics formulation; however, its effect on the controller performance is not discussed in the study, nor, to the author’s best knowledge, in any other study in the literature. Yuan et al. (2016) present a non-linear model predictive controller (NMPC) for ABS and TC. The formulation considers all four wheels in the same internal model. Reference tracking is not used since the slip ratio is solely controlled through the constraints of the NMPC formulation. Moreover, the tyre-road friction coefficient is considered to be known a priori, which introduces further challenges for a real vehicle implementation. For an internal-combustion-engine-driven vehicle, Borrelli et al. (2006) introduce four linear MPC TC strategies that are compared with a hybrid explicit MPC. The hybrid design adopts a piecewise affine approximation of the non-linear longitudinal tyre force characteristics as a function of the slip ratio. Simulation and experimental results show the performance enhancement of the hybrid strategy with respect to the linear approaches.

In the case of implicit NMPC, a non-linear programming (NLP) problem is solved online at each sampling time. The resulting computational load makes implicit NMPC difficult to implement in real automotive applications if the required sampling frequency is high. In this respect, Basrah et al. (2017) provide an example of real-time capable NMPC for an ABS with torque blending, including a comparison with a linear MPC approach. The results show that the computational time of the implicit NMPC, i.e. 3-4 ms on a desktop personal computer, is within the selected sampling interval of 5 ms. In Yuan et al. (2016), the implicit NMPC strategy is run on a rapid control prototyping unit, with a computational time of 4-5 ms and an implemented sampling time of 10 ms.

The studies of Sections 6.1 and 6.2 present an explicit NMPC (eNMPC in the remainder) for TC on electric vehicles with in-wheel drivetrains. The explicit solution is computed offline by using a multi-parametric quadratic programming approximation of the mp-NLP problem. The control action is evaluated online at each sampling time starting from the current values of the system states and parameters, and the offline explicit solution, stored in the memory of the control unit. This drastically reduces the required computational power. The other advantage is that the complete feedback law is available beforehand in its explicit form, which allows its analysis for the range of state and reference parameters.

Another important aspect is the performance comparison and critical analysis of different TC implementations. In this respect, Borrelli et al. (2006) claim that the performance of the proposed MPC “is comparable with that of a well-tuned PID” (proportional-integral-derivative) controller. The same authors state that “the simulation and test results demonstrated that the $l_1$-optimal hybrid controller used in this problem can lead to about 20% reduction in peak slip amplitudes and corresponding spin duration
when compared to best case linear MPC counterparts”. Similarly, Basrah et al. (2017) show the superiority of NMPC over linear MPC in terms of slip control performance. The necessity of “objective benchmarking technologies” in the field of ABS/TC was pointed out in the survey study in Ivanov et al. (2015c). In order to understand where the strategies of the different papers stand with respect to each other, a comparison is well needed. De Pinto et al. (2017) partially cover this knowledge gap, but limit the analysis to on-board electric drivetrains, characterised by significant torsional dynamics. Satzger and De Castro (2018) include also an MPC-PI experimental comparison, but for an ABS application combined with torque blending.

### 6.1.2 Contribution

Based on the previous discussion, the points of novelty of the study presented in Tavernini et al. (2019a) are:

- The design of TC systems based on explicit NMPC, implementable at the same sampling interval as a typical PI controller for TC, but with better tracking performance.

- The observation of the explicit feedback control law, and its dependency on the vector of parameters from the plant.

- The simulation-based analysis of the performance advantages of the proposed explicit NMPC compared to a well-tuned benchmark PI TC system with gain scheduling and anti-windup features.

- The sensitivity analysis of the performance of TC algorithms with respect to their sampling interval.

- The discussion of the benefit of considering transient tyre response and vertical load transfers in the internal model for the NMPC formulation.

- The presentation of experimental test results based on explicit non-linear model predictive control applied to a fully electric vehicle prototype with in-wheel drivetrains.

### 6.1.3 Conclusion and future work

The paper Tavernini et al. (2019a) presents traction controllers for electric vehicles with in-wheel motors, based on explicit non-linear model predictive control of the wheel slip velocity. These are compared with more conventional TC strategies based on PI control. The interested reader is referred to the publication for details on the results. Here, however, the novel conclusions are summarised:
• The implementation time step of the TC has a more significant impact on the control system performance than the selection of the control system technology. Employing non-linear MPC is not enough to provide better performance than that of a PI running at an appropriate time step. To achieve a performance enhancement, for the case study TC application, time steps of approximately 2 ms are recommended, rather than of 4 ms or 8 ms. Both for the PI TC and non-linear model predictive control TC, the control system settings have to be fine-tuned through tests in the time domain for the selected time step.

• The presented explicit non-linear model predictive control implementations are characterised by online computational times in the range of 5-25 μs on the adopted dSPACE MicroAutoBox rapid control prototyping unit. This means that the strategies could be potentially implemented at any reasonable frequency for automotive TC applications. On the contrary, based on the literature it would not be possible to run an equivalent implicit non-linear model predictive controller at the required time step of 2 ms.

• The non-linear model predictive controller allows a 9.2% tracking performance improvement with respect to a PI controller during the variable tyre-road friction scenario, simulated with a high fidelity vehicle model.

• The local multi-parametric quadratic approximation of the non-linear problem, typical of the selected explicit non-linear model predictive control method, does not bring any perceivable performance difference with respect to the corresponding implicit non-linear model predictive controller.

• The consideration of tyre force dynamics and vertical load transfers in the internal model for model predictive control system design has negligible effects on the TC performance during the simulated scenario.

• The interpretation of the non-linear model predictive control law provides useful information on the effect of the different input parameters on the control action. The piecewise affine control law can be approximated with only three planes.

• An explicit non-linear model predictive control strategy for TC has been successfully implemented on a fully electric prototype vehicle for the first time in the literature, to the best of the author’s knowledge.

Future developments of the research will evaluate: i) the increase of the number of parameters of the explicit non-linear model predictive control problem, and the implications in terms of memory requirements and performance benefits; and ii) the possibility of simpler strategies able to replicate a similar control pattern with reduced memory requirements for the online implementation of the controller.
6.2 Traction control in cornering

6.2.1 Introduction

As appeared in outlines in Section 6.1.1, extensive literature is available on the topic of wheel slip control. Some of the published work includes adaptations for combined driving/braking and cornering conditions. For example, the TC in Aligia et al. (2018) applies a dynamic saturation of the electric motor torque, by using a friction circle formulation and the estimated lateral tyre force. In Park and Kim (1999), the longitudinal slip reference is varied as a function of the estimated slip angle. However, the wheel slip control implementations based on model predictive control, such as Borrelli et al. (2006), Satzger and de Castro (2014), or Tavernini et al. (2019a), do not include adaptations for combined slip conditions. To the best of the author’s knowledge, the only contribution in this direction is in Bächle et al. (2014a), which proposes an MPC control allocation scheme adopting a linear tyre model with varying slip stiffness as a function of slip angle.

Contribution

The paper Metzler et al. (2019a) presents a traction controller for combined driving and cornering conditions, based on explicit non-linear model predictive control. The prediction model includes a non-linear tyre force model using a simplified version of the Pacejka Magic Formula, incorporating the effect of combined longitudinal and lateral slips. The benefits of the proposed formulation with respect to a controller with a tyre model for pure longitudinal slip are highlighted through simulations of a front-wheel-drive electric vehicle with in-wheel motors. Objective performance indicators provide a performance assessment in traction control scenarios.

In summary, the novelties of the study are:

- The explicit NMPC-based traction controller with a non-linear tyre model for combined slip.
- The objective assessment of the benefits of the combined slip formulation within the prediction model of the traction controller, with respect to a tyre model for pure longitudinal slip.

6.2.2 Control system design

In the following sections, all necessary definitions for the optimal control problem of a model predictive traction controller are introduced.
Prediction models

The prediction model is a half-car model or a quarter-car model, depending on whether the specific vehicle is two-wheel-drive or four-wheel-drive, with a non-linear formulation of the steady-state tyre force of the driven wheel. The states are the longitudinal component of the vehicle speed in the wheel reference system, \( v_x(t) \), and the angular wheel speed, \( \Omega(t) \). The vehicle body dynamics are described by,

\[
\dot{v}_x(t) = \frac{1}{m_{H/Q}} F_x(t),
\]

(6.1)

where \( m_{H/Q} \) is the relevant mass (representing half or a quarter of the vehicle), \( t \) is time, and \( F_x \) is the longitudinal tyre force. By neglecting the rolling resistance, the wheel dynamics are given by,

\[
\dot{\Omega}(t) = \frac{1}{J_w} \left( T_w(t) - R_l(t_k) F_x(t) \right),
\]

(6.2)

where \( J_w \) is the wheel mass moment of inertia, and \( R_l \) is the loaded wheel radius. The wheel torque, \( T_w \), is calculated as,

\[
T_w(t) = T_{req}(t_k) - \Delta T(t),
\]

(6.3)

where \( T_{req}(t_k) \) is the torque request from a higher level controller (e.g. a drivability controller or a torque-vectoring controller), and \( \Delta T(t) \) is the torque reduction imposed by the traction controller.

The longitudinal tyre force in pure longitudinal slip conditions, \( F_{x0} \), is approximated with a simplified version of the Pacejka Magic Formula, with peak factor, \( D_x \), constant shape factor, \( C_x = p_{Cz1} \), and stiffness factor, \( B_x \),

\[
F_{x0} = D_x(F_z) \sin \left( C_x \arctan \left( B_x(F_z) \kappa_x \right) \right),
\]

(6.4)

where \( \kappa_x(t) \) is the longitudinal slip in traction and \( F_z(t_k) \) is the vertical tyre load. The peak factor is defined as,

\[
D_x(t) = \left( p_{Dz1} + p_{Dz2} df_z(t_k) \right) \lambda_\mu F_z(t_k),
\]

(6.5)

where \( p_{Dz1} \) and \( p_{Dz2} \) are constants, and the scaling factor \( \lambda_\mu \) depends on the estimated tyre-road friction coefficient. The normalised change in vertical load is,

\[
df_z(t) = \frac{F_z(t_k) - F_{z0}}{F_{z0}},
\]

(6.6)
where $F_{z0}$ is the nominal wheel load. The stiffness factor is defined as,

$$B_x(t) = \frac{K_{x\kappa}(t_k)}{C_x D_x(t_k)},$$  \hspace{1cm} (6.7)

with $K_{x\kappa}$ expressed as,

$$K_{x\kappa}(t) = \left(p_{Kx1} + p_{Kx2} df_z(t_k)\right) \exp\left(p_{Kx3} df_z(t_k)\right) F_z(t_k),$$ \hspace{1cm} (6.8)

where $p_{Kx1}$, $p_{Kx2}$, and $p_{Kx3}$ are constants.

The influence of combined slip conditions on the longitudinal tyre force, $F_x$, is modelled through the weighting function $G_{xa}$ with constant shape factor, $C_{xa} = r_{Cx1}$, and variable stiffness factor, $B_{xa}$,

$$G_{xa} = \cos\left(C_{xa} \arctan\left(B_{xa}(\kappa_x(t))\alpha\right)\right),$$ \hspace{1cm} (6.9)

where $\alpha(t_k)$ is the slip angle. $B_{xa}$ is defined as,

$$B_{xa}(t) = r_{Bx1} \cos\left(\arctan\left(r_{Bx2} \kappa_x(t)\right)\right),$$ \hspace{1cm} (6.10)

with constants $r_{Bx1}$ and $r_{Bx2}$. The longitudinal tyre force in combined slip conditions is given by,

$$F_x(t) = F_{x0}(\kappa_x(t)) G_{xa}(\kappa_x(t),\alpha(t_k)).$$ \hspace{1cm} (6.11)

In traction, the longitudinal slip is,

$$\kappa_x(t) = -\frac{v_{sx}(t)}{v_r(t)},$$ \hspace{1cm} (6.12)

with the slip speed, $v_{sx}$, defined as,

$$v_{sx}(t) = v_x(t) - v_r(t),$$ \hspace{1cm} (6.13)

and the linear speed of rolling, $v_r$, given by,

$$v_r(t) = \Omega(t) R_e(t_k).$$ \hspace{1cm} (6.14)

In (6.14), the effective radius, $R_e$, in free rolling conditions is estimated as,

$$R_e(t) = R_0 - \rho_0 \left(D_{Re} \arctan\left(B_{Re} \frac{F_z(t_k)}{F_{z0}}\right) + F_{Re} \frac{F_z(t_k)}{F_{z0}}\right),$$ \hspace{1cm} (6.15)

where $R_0$ is the unloaded wheel radius, $\rho_0$ is the tyre deflection at the nominal wheel load, and $D_{Re}$, $B_{Re}$, and $F_{Re}$ are constant coefficients. The vertical tyre deflection, $\rho$, is
defined as the ratio between the vertical tyre load, $F_z$, and vertical tyre stiffness, $C_z$,
\[
\rho(t) = \frac{F_z(t)k}{C_z}. \tag{6.16}
\]
Hence, the loaded wheel radius in (6.2) is calculated as,
\[
R_l(t) = R_0 - \rho(t_k). \tag{6.17}
\]

Fig. 6.1: Longitudinal tyre force characteristics as functions of longitudinal slip, for six vertical loads and two slip angles. The black lines indicate the tyre forces calculated with the full version of the Pacejka Magic Formula (version 5.2); the coloured lines correspond to the tyre forces according to the simplified Magic Formula-based combined slip tyre force model.

Figure 6.1 reports the longitudinal tyre force characteristics as functions of longitudinal slip in high tyre-road friction conditions, at six vertical loads (ranging from 1.0 kN to 8.5 kN), for $\alpha = 0$ deg in sub-plot (a) and $\alpha = 8$ deg in sub-plot (b). The graphs are calculated with i) the full version 5.2 of the Pacejka Magic Formula used in the vehicle simulation model for control system assessment, parametrised with real world tyre data (black lines); and ii) the simplified Pacejka Magic Formula of the TC prediction model, according to (6.4)-(6.11) (coloured lines). The good match of the results for both slip angle values confirms the accuracy of the simplified tyre model formulation.

**Model parameters** For the identification of the prediction model parameters, least squares regressions are performed. In the defining equations of formulation (6.4), the identified parameters are $p_{Dx1}$ and $p_{Dx2}$ for $D_x$, $p_{Cx1}$ for $C_x$, and $p_{Kx1}$, $p_{Kx2}$, and $p_{Kx3}$ for $B_x$. The longitudinal tyre forces defined by (6.4) are fitted against the full Pacejka Magic Formula in pure longitudinal slip conditions. The longitudinal slip $\kappa_x$ is varied in...
the range 0% to 12% with an increment of 0.2% and the vertical tyre force $F_z$ in the range 1 kN to 12 kN with an increment of 0.5 kN.

Based on these identified values, the parameters $r_{Bx1}$ and $r_{Bx2}$ for $B_{za}$ and $r_{Cz1}$ for $C_{xa}$ are determined for the defining equations of formulation (6.11). For the regression analysis, the Magic Formula is employed in combined slip conditions. The longitudinal slip and vertical tyre force are varied in the same range as previously defined. The tyre slip angle $\alpha$ is varied in the range 0 deg to 15 deg with an increment of 1 deg.

The parameters for the vertical tyre characteristics and effective radius, respectively, are adopted from the Pacejka Magic Formula definition.

Overview In Table 6.1, an overview of the prediction models is presented. In the remainder, the two controllers are compared, namely, i) controller (a), i.e. the NMPC-based TC with the combined longitudinal and lateral tyre force model in (6.11); and ii) controller (b), i.e. the non-linear model predictive traction controller with the tyre force model in (6.4), purely based on longitudinal slip.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Tyre force model</th>
<th>Slip formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>$F_x(t)$ - (6.11)</td>
<td>combined slip</td>
</tr>
<tr>
<td>(b)</td>
<td>$F_{x0}(t)$ - (6.4)</td>
<td>pure longitudinal slip</td>
</tr>
</tbody>
</table>

Optimal control problem

The formulation of the non-linear model predictive traction controller is based on an optimal control problem defined by a cost function and constraints, in addition to the prediction model equations in Section Prediction models.

Cost function The objective of the traction controller is to prevent excessive growth of the longitudinal slip, $\kappa_x(t)$, while tracking the torque request, $T_{req}(t_k)$, and, therefore, minimising the wheel torque reduction, $\Delta T(t)$, imposed by the TC. Hence, the cost function, $V$, of the optimal control problem of this study is defined as,

$$V(t_k) = \int_{t_k}^{t_f} \left[ \frac{r_u}{U^{2}_{sc}} \Delta T(t)^2 + \frac{q_{\nu}}{N^{2}_{sc}} N_{\kappa_x}(t)^2 \right] dt,$$

(6.18)

where $t_k$ is the current time and $t_f = t_k + N_p h$ is the time at the end of the prediction horizon, defined by the number of prediction steps, $N_p$, and the discretisation time, $h$. The tuning weights $r_u$ and $q_{\nu}$ are related to appropriate scaling factors $U_{sc}$ and $N_{sc}$ to
allow equivalent influence in the tuning procedure. The torque reduction is defined as,

\[
\Delta T(t) = T_{\text{req}}(t_k) - T_w(t),
\]

(6.19)

where the torque request, \(T_{\text{req}}\), generated by a higher level controller, is assumed constant over the prediction horizon.

**Constraints**  The slack variable, \(N_{\kappa_x}(t)\), is positive if the actual or predicted longitudinal slip, \(\kappa_x(t)\), exceeds the reference value, \(\kappa_{x,\text{ref}}\), in the soft constraint,

\[
\kappa_x(t) \leq \kappa_{x,\text{ref}} + N_{\kappa_x}(t).
\]

(6.20)

Based on equation (6.18), an increase of \(q_\nu\) improves, therefore, the slip tracking performance, while an increase of \(r_u\) reduces the TC torque corrections. Appropriate additional constraints are applied, e.g. to ensure that the corrected wheel torque does not exceed the request from the higher level controller or brakes the wheel,

\[
0 \leq \Delta T(t) \leq T_{\text{req}}(t_k).
\]

(6.21)

Additionally, the slack variable must be positive semi-definite to only relax and not tighten the soft constraint (6.20) on the slip,

\[
N_{\kappa_x}(t) \geq 0.
\]

(6.22)

**State transformation**  For the optimisation problem, the vehicle speed, \(v_x(t)\), and angular wheel speed, \(\Omega(t)\), are required as initial conditions for the dynamic equality constraints (6.1) and (6.2). The vertical tyre force, \(F_z(t_k)\), the tyre slip angle, \(\alpha(t_k)\), and the torque request, \(T_{\text{req}}(t_k)\), are considered as varying parameters.

In the optimal control problem, the longitudinal slip violation is constrained in (6.20). Therefore, a formulation with the longitudinal slip, \(\kappa_x(t)\), and the vehicle speed, \(v_x(t)\), instead of angular wheel speed, \(\Omega(t)\), and vehicle speed, \(v_x(t)\), turned out to be beneficial. With the description based on the new variables, employing the transformation law,

\[
\kappa_x(t) = \frac{\Omega(t) R_e(t_k) - v_x(t)}{\Omega(t) R_e(t_k)},
\]

(6.23)

the constrained variables are aligned with the coordinate axes of the orthogonal exploration space. Therefore, the exploration space can be reduced to hyper-rectangles avoiding parameters that are not likely to occur in the closed-loop operation of the controller.
Table 6.2 reports the main settings for the optimal control problem of this study.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>Discretisation time</td>
<td>3 ms</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Sampling time</td>
<td>6 ms</td>
</tr>
<tr>
<td>$N_p$</td>
<td>Prediction steps</td>
<td>5</td>
</tr>
<tr>
<td>$N_c$</td>
<td>Control steps</td>
<td>1</td>
</tr>
<tr>
<td>$\kappa_{x,ref}$</td>
<td>Reference longitudinal slip</td>
<td>8%</td>
</tr>
<tr>
<td>$r_u$</td>
<td>Weight on torque reduction</td>
<td>1</td>
</tr>
<tr>
<td>$q_v$</td>
<td>Weight on slip violation</td>
<td>80</td>
</tr>
<tr>
<td>$U_{sc}$</td>
<td>Scaling factor for torque reduction</td>
<td>300 N m</td>
</tr>
<tr>
<td>$N_{sc}$</td>
<td>Scaling factor for slip violation</td>
<td>2%</td>
</tr>
<tr>
<td>$X_{ch,v_x}$</td>
<td>Characteristic vehicle speed</td>
<td>70 km/h</td>
</tr>
<tr>
<td>$X_{ch,\kappa_x}$</td>
<td>Characteristic longitudinal slip</td>
<td>10%</td>
</tr>
<tr>
<td>$P_{ch,F_z}$</td>
<td>Characteristic vertical load</td>
<td>10 000 N</td>
</tr>
<tr>
<td>$P_{ch,\alpha}$</td>
<td>Characteristic slip angle</td>
<td>17 deg</td>
</tr>
<tr>
<td>$P_{ch,T_{req}}$</td>
<td>Characteristic torque request</td>
<td>1850 N m</td>
</tr>
<tr>
<td>$N_{ch,\kappa_x}$</td>
<td>Characteristic slip violation</td>
<td>5%</td>
</tr>
<tr>
<td>$U_{ch,\Delta T}$</td>
<td>Characteristic torque reduction</td>
<td>1850 N m</td>
</tr>
</tbody>
</table>

Multi-parametric non-linear program  The optimal control problem consists of the cost function (6.18), the equality constraints (6.1) and (6.2), and the inequality constraints (6.20)-(6.22) including all necessary definitions. It has the torque reduction $\Delta T(t)$ and the slack variable $N_{\kappa_x}(t)$ as optimisation variables in its continuous formulation. The optimal control problem is reformulated into a form suitable for the numerical solution by parametrisation and discretisation, and expressed in the general form,

$$V^*(x_p(t_k)) = \min_z V(z, x_p(t_k)), \quad (6.24)$$

subject to

$$G(z, x_p(t_k)) \leq 0. \quad (6.25)$$

which represents a multi-parametric non-linear programming problem, as derived in Section 3.4. The vector of parameters, $x_p(t_k) = [x^T(t_k), p^T(t_k)]^T$, combines the initial conditions, $x(t_k) = [v_{x}(t_k)/X_{ch,v_x}, \kappa_x(t_k)/X_{ch,\kappa_x}]^T$, of the ordinary differential equations of the prediction model, and the vector of varying parameters, $p(t_k) = [F_z(t_k)/P_{ch,F_z}, \alpha(t_k)/P_{ch,\alpha}, T_{req}(t_k)/P_{ch,T_{req}}]^T$. All variables in the optimal control problem are non-dimensionalised according to the procedure introduced in Section 3.2.2 with appropriate
choices for the characteristic quantities, e.g. $U_{ch,ΔT}$, $N_{ch,κx}$, $X_{ch,κz}$, $X_{ch,κz}$, etc. In the simultaneous approach, the vector of decision variables, $z = [U^T, N^T, X^T]^T$, contains the vector of input trajectory parameters, $U$, the vector of slack variable trajectory parameters, $N$, and the vector of state trajectory parameters, $X$. $U = [ΔT(t_k)/U_{ch,ΔT}, \ldots, ΔT(t_{k+N_c-1})/U_{ch,ΔT}]^T$ includes the prediction model inputs, which can vary $N_c$ times over the horizon, where $N_c$ is the number of control steps, and then are kept constant. $N = [N_{κx}(t_k)/N_{ch,κx}, \ldots, N_{κx}(t_{k+N_p})/N_{ch,κx}]^T$ combines the slip violations over the prediction horizon and $X = [x(t_{k+1})^T, \ldots, x(t_{k+N_p})^T]^T$ combines the intermediate predicted states, which are treated as additional optimisation variables.

In the receding horizon approach, the optimal solution, $z^*$, of (6.24)-(6.25) is calculated at each time step. The optimal torque reduction, $ΔT^*(t_k)$, is extracted from the first element of $U$ and applied to the plant. The iteration of the process makes the control system a closed-loop approach.

6.2.3 Performance analysis

In the following sections, the performance of the non-linear model predictive traction controller is assessed in traction control scenarios adopting objective performance indicators. Moreover, an explicit solution is generated and the influence of the parameter space definition on the complexity of the controller is analysed.

Traction control scenario

The results are obtained with a high-fidelity MATLAB/Simulink simulation model of a case study front-wheel drive (FWD) electric vehicle with in-wheel motors. The main vehicle parameters are equivalent to the simulation model of the vehicle stability controller, as defined in Table 5.1. However, the drivetrain configuration with in-wheel motors substantially differs from the case study vehicle of Chapter 5 which has an internal combustion engine. Each driven wheel of the case study FWD electric vehicle is controlled with a separate NMPC-based TC. The tyre model of the simulator is based on the version 5.2 of the Pacejka Magic Formula (see Fig. 6.1). The tyre force transients are represented with the realistic relaxation length formulation proposed by Giangiulio and Arosio (2006), which has been modified to give plausible results at low vehicle speeds. More details on the formulation can be found in Section 7.3.1. All MPCs are simulated with a prediction model discretisation time of $H = 3$ ms and a controller sampling time of 6 ms. The implementation time step size is chosen to be close to the range of practical TC applications ($\sim 10$ ms).

Figure 6.2 reports an example of TC simulation results. The vehicle is initially travelling at a speed of 50 km/h. At the beginning of the relevant part of the manoeuvre,
6.2 Traction control in cornering

Fig. 6.2: Time histories of wheel torques, longitudinal slips, tyre slip angles, and vertical tyre loads for controllers (a) and (b) in the traction control scenario. The subscripts “FL” and “FR” indicate the front left and front right corner.
a ramp steer is initiated with a steering wheel angle rate of 60 deg/s. During the steering application, a sequence of torque request steps is imposed by the higher level controller. The reference wheel torque is the same on the two driven wheels. The figure compares the results of i) controller (a), i.e. the eNMPC-based traction controller with the combined longitudinal and lateral tyre force model in (6.11); and ii) controller (b), i.e. the non-linear model predictive traction controller with the tyre force model in (6.4), purely based on longitudinal slip. Controller (b) is implemented implicitly only, i.e. the solution of the non-linear model predictive control problem is calculated online at each time step.

Controller (a) provides significantly better longitudinal slip tracking performance, especially in conditions of large tyre slip angles. This can be observed in Fig. 6.2 for the last three torque request steps, where the TCs need to apply a torque reduction on both driven wheels. For the first three torque request steps, both traction controllers intervene only on the inner wheel, i.e. the left wheel, because of its lower values of vertical load, which facilitate wheel spinning. The lower levels of longitudinal tyre slip with traction controller (a) lead to a smaller reduction of the lateral tyre forces, and, thus, results in higher values of vehicle yaw velocity, in comparison with controller (b).

Performance indicators

To objectively assess the performance of the controllers, the following indicators are introduced:

- $J_1$, i.e. the integral of the absolute value of the control error, normalised with time,

  \[
  J_1 = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} \left| e_{x, \kappa}(t) \right| dt = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} \left| N_{\kappa_x(t_k)}(t) \right| dt, \tag{6.26}
  \]

  where the slip violation, $e_{x, \kappa}$, is defined as $e_{x, \kappa}(t) = \kappa_x(t) - \kappa_{x,\text{ref}}$ if the longitudinal slip exceeds the slip reference $\kappa_{x,\text{ref}} = 8\%$; otherwise $e_{x, \kappa}$ is set to zero. This definition is equivalent to the slack variable $N_{\kappa_x(t_k)}$, as defined in (6.20).

- $J_2$, i.e. the integral of the absolute value of the control action, normalised with time,

  \[
  J_2 = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} \left| \Delta T(t) \right| dt. \tag{6.27}
  \]

$T_1$ and $T_2$ are the times at the beginning and the end of the relevant part of the simulation, i.e. $T_1 = 0\,s$ and $T_2 = 4\,s$ in the specific test. $J_1$ and $J_2$ are calculated for each front wheel, and then the respective values for the front left and front right corners are summed up to obtain the overall TC indicator.
Table 6.3 reports the performance indicators for the manoeuvre of Fig. 6.2 for i) controller (a) implemented implicitly and, therefore, indicated as iNMPC (a); and ii) the implicit version of controller (b), indicated as iNMPC (b).

Table 6.3: Values of the controller performance indicators for the manoeuvre in Fig. 6.2.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Slip violation</th>
<th>Torque reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iNMPC (a)</td>
<td>0.265, 0.063, 0.328</td>
<td>852.0, 194.1, 1046.1</td>
</tr>
<tr>
<td>iNMPC (b)</td>
<td>0.925, 0.849, 1.774</td>
<td>824.7, 120.9, 945.6</td>
</tr>
</tbody>
</table>

The enhanced longitudinal slip tracking performance of controller (a) is evident. In fact, for controller (a) the indicator $J_1$ ($J_{1,FL} + J_{1,FR}$) has significantly lower values, by a factor > 5, with respect to iNMPC (b). In particular, on the front right corner with high values of vertical load, the slip tracking performance is compromised for controller iNMPC (b) compared to controller iNMPC (a). This is also reflected by the indicator $J_{1,FR}$, which has much higher values for controller (b), by a factor > 13 compared to controller (a). The significant influence of the tyre slip angle in conditions with high vertical tyre forces can also be observed in Fig. 6.1. The superior slip tracking performance of controller (a) is associated with larger torque reductions, which are reflected in the higher values of $J_2$. Again, comparing the two controllers, a more significant influence can be observed on the front right corner, with higher values of the vertical load, investigating the wheel torques in Fig. 6.2 and the performance indicators in Table 6.3.

### Explicit solution

#### Parameter exploration space

For the parameter space exploration of the Algorithm 1, the hyper-rectangle $X$ to be explored needs to be defined. The choice of the bounds defining the hyper-rectangle has a significant influence on the complexity of the explicit controller. Moreover, in case the parameter vector lies outside the hyper-rectangle $X$, the clipping to its bounds affects the dynamic behaviour of the controller.

The bounds were chosen to cover the closed-loop response of controller (a) in the manoeuvre of Fig 6.2. However, a variation of the hyper-rectangle bounds on the vehicle speed, $v_x$, the longitudinal slip, $\kappa_x$, the tyre slip angle, $\alpha$, and the torque request, $T_{req}$, shows the influence of the parameter space definition on the complexity of the explicit controller.

#### Algorithm and post-processing

In Table 6.4, the explicit solution of controller (a), indicated as eNMPC (a), is shown for varying bounds on the parameter exploration space.
The approximation tolerances for the partitioning algorithm 1 are chosen $\tilde{\varepsilon} = 0.2$ for the cost error, $\tilde{\rho} = 0.05$ for the solution error, and $\tilde{\delta} = 0.1$ for the maximum constraint violation. The weighting vector $w_\rho$ extracts only the first element associated with the receding horizon control law of the input and the weighting vectors $w_\delta_i$ and $w_\delta_e$, respectively, are chosen to equally weight all constraint violations. The remaining settings for the partitioning algorithm, i.e. the minimal allowed volume $S_{\text{min}}$, the number of internal hyper-rectangles $N_0$, and the specifications for the generation of the set of test points, are chosen equivalent to the settings for the generation of the explicit solution in Chapter 5. The hyper-rectangles $X$ to be explored are defined in Table 6.4.

For the explicit controllers in Table 6.4, the maximum estimated error bounds on the cost function and the solution, as well as the maximum constraint violation, fulfil the specified tolerances in all hyper-rectangles. Indeed, the average approximation errors are notably smaller than the defined tolerances. Moreover, the minimum volumes are far from the specified minimum allowed volume. In Table 6.4, the influence of the parameter bounds on the complexity of the explicit controller in terms of polyhedral regions and memory requirements is shown. Moreover, the performance indicators in the manoeuvre of Fig. 6.2 are calculated for the implicit controllers, demonstrating the effects of parameter saturation on the controller performance due to different bounds, and for the explicit controllers, showing the influence of the explicit approximate solution. For example, comparing the controllers (1) and (2), a considerable influence of the parameter bound corresponding to the vehicle speed on the controller complexity can be observed. Both performance indicators of the explicit solutions are close to the ones from the implicit controllers demonstrating appropriately chosen approximation tolerances for Algorithm 1. Moreover, the influence of the longitudinal slip bounds can be observed by comparing controllers (1), (3), and (4), showing a higher increase in polyhedral regions by expanding the hyper-rectangle to higher slip values, as it is the case for controller (4), compared to expanding to lower slip values, as for controller (3). This is caused by the slip controller leading to increased control action, i.e. higher torque reductions, in the high slip region. Again, a good agreement of the performance indicators for the implicit and the explicit controllers can be noted. By comparing controllers (1) and (5), the influence of the tyre slip angle range can be analysed. The complexity of the controller in terms of polyhedral regions decreases with reduced bounds on the slip angle. However, the slip tracking performance is compromised for controller (5) even for the implicit approach showing once more the significance of the tyre slip angle. Different parameter bounds on the torque request are chosen for controllers (1), (6), and (7). The complexity of the controllers naturally increases with larger parameter exploration space, i.e. higher bounds on the torque request. A very significant influence of inappropriately chosen torque request bounds can be observed for controller (6), exhibiting very high slip tracking errors. Due to the saturation of the parameters, the controller (6) does not apply
Table 6.4: Partitioning algorithm and post-processing data for the explicit NMPC controller (a). The approximation tolerances are \( \tilde{\varepsilon} = 0.2 \) for the cost error, \( \tilde{\rho} = 0.05 \) for the solution error, and \( \tilde{\delta} = 0.1 \) for the maximum constraint violation.

<table>
<thead>
<tr>
<th>Case</th>
<th>Parameter vector</th>
<th>Vehicle speed bounds</th>
<th>Longitudinal slip bounds</th>
<th>Vertical tyre force bounds</th>
<th>Tyre slip angle bounds</th>
<th>Torque request bounds</th>
<th>Indices iNMPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xp</td>
<td></td>
<td>( \bar{v}_x ) (km/h)</td>
<td>( \bar{v}_x ) (km/h)</td>
<td>( \bar{\kappa}_x ) (%)</td>
<td>( \bar{F}_x ) (N)</td>
<td>( \bar{\alpha} ) (deg)</td>
<td>( \bar{\bar{\alpha}} ) (deg)</td>
</tr>
<tr>
<td>(1)</td>
<td>( v_x, \kappa_x, F_x, \alpha, T_{req} )</td>
<td>50</td>
<td>75</td>
<td>6</td>
<td>9</td>
<td>1500</td>
<td>9500</td>
</tr>
<tr>
<td>(2)</td>
<td>( v_x, \kappa_x, F_x, \alpha, T_{req} )</td>
<td>35</td>
<td>75</td>
<td>6</td>
<td>9</td>
<td>1500</td>
<td>9500</td>
</tr>
<tr>
<td>(3)</td>
<td>( v_x, \kappa_x, F_x, \alpha, T_{req} )</td>
<td>50</td>
<td>75</td>
<td>3</td>
<td>9</td>
<td>1500</td>
<td>9500</td>
</tr>
<tr>
<td>(4)</td>
<td>( v_x, \kappa_x, F_x, \alpha, T_{req} )</td>
<td>50</td>
<td>75</td>
<td>6</td>
<td>12</td>
<td>1500</td>
<td>9500</td>
</tr>
<tr>
<td>(5)</td>
<td>( v_x, \kappa_x, F_x, \alpha, T_{req} )</td>
<td>50</td>
<td>75</td>
<td>6</td>
<td>9</td>
<td>1500</td>
<td>9500</td>
</tr>
<tr>
<td>(6)</td>
<td>( v_x, \kappa_x, F_x, \alpha, T_{req} )</td>
<td>50</td>
<td>75</td>
<td>6</td>
<td>9</td>
<td>1500</td>
<td>9500</td>
</tr>
<tr>
<td>(7)</td>
<td>( v_x, \kappa_x, F_x, \alpha, T_{req} )</td>
<td>50</td>
<td>75</td>
<td>6</td>
<td>9</td>
<td>1500</td>
<td>9500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Orthogonal partitions</th>
<th>Polyhedral regions</th>
<th>Size top layer</th>
<th>Size bottom layer</th>
<th>Algorithm execution time</th>
<th>Indices eNMPC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(-)</td>
<td>(a) / (b) / (c) / (d)</td>
<td>(kB)</td>
<td>(kB)</td>
<td>(h)</td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>54</td>
<td>1713/1713/1288/1295</td>
<td>3</td>
<td>1069/721/605/605</td>
<td>0.45/0.28/0.32/0.63</td>
<td>0.375</td>
</tr>
<tr>
<td>(2)</td>
<td>108</td>
<td>3001/3001/2176/2167</td>
<td>6</td>
<td>1860/1182/979/979</td>
<td>0.94/0.26/0.46/1.02</td>
<td>0.372</td>
</tr>
<tr>
<td>(3)</td>
<td>134</td>
<td>1982/1982/1562/1522</td>
<td>7</td>
<td>1207/805/675/678</td>
<td>1.11/0.31/0.32/0.89</td>
<td>0.379</td>
</tr>
<tr>
<td>(4)</td>
<td>128</td>
<td>2414/2414/1934/1870</td>
<td>6</td>
<td>1404/921/911/886</td>
<td>1.02/0.31/0.97/1.47</td>
<td>0.374</td>
</tr>
<tr>
<td>(5)</td>
<td>52</td>
<td>1329/1329/1029/1010</td>
<td>3</td>
<td>846/501/412/410</td>
<td>0.43/0.08/0.17/0.60</td>
<td>0.874</td>
</tr>
<tr>
<td>(6)</td>
<td>44</td>
<td>1380/1380/1021/1015</td>
<td>3</td>
<td>869/493/450/449</td>
<td>0.36/0.09/0.24/0.52</td>
<td>2.452</td>
</tr>
<tr>
<td>(7)</td>
<td>56</td>
<td>1795/1795/1336/1318</td>
<td>4</td>
<td>1134/712/590/590</td>
<td>0.49/0.32/0.38/0.74</td>
<td>0.376</td>
</tr>
</tbody>
</table>
the necessary torque reductions in case the torque request of 1800 N m in the traction control scenario in Fig. 6.2 exceeds the upper parameter bound of 1600 N m. However, the indicators of the implicit and explicit solution are reasonably close demonstrating a sufficiently accurate approximation of the explicit solution. As a final note, it is interesting to observe that the parameter saturation degrades the performance of all controllers at least slightly, even when implicitly solved, compared to the unsaturated controller in Table 6.3.

![Fig. 6.3: Orthogonal partitions of the top layer and polyhedral critical regions of the bottom layer of the explicit controller (a)-(4) with parameters](image-url)

For the post-processing algorithm 4, the settings as defined in Table 5.8 are employed. For case (b), a significant reduction of up to 43% of the bottom layer file size can be observed compared to case (a). This is due to the generation of binary search trees reducing the memory requirements of the explicit controllers. A further reduction can be achieved by the disjoint optimal and the sub-optimal merging procedure active in case (c). For the optimal control problem of this section with only one control input, the merging procedures are effective and reduce the number of polyhedral regions by up to 27% compared to case (a). Subsequently, also the memory requirements are further reduced compared to case (b). Finally, the clipping-based complexity reduction of case (d) does bring a further reduction in the polyhedral regions (up to 28% compared to case (a)). Similarly to the merging procedures, the single control variable facilitates the clipping-based complexity reduction increasing the likelihood for saturated regions in the receding horizon feedback law compared to multi-variable systems. Summarising,
the post-processing algorithm 4 has a significant beneficial impact on the complexity reduction of the explicit solution with each of the settings giving a contribution.

In conclusion, an apparent albeit not excessive influence of the parameter bound definition on the complexity of the controller in terms of polyhedral regions can be observed. Moreover, the post-processing algorithm proved to be effective in decreasing the complexity of the explicit solution of the controllers by up to 52 % in terms of memory requirements and up to 28 % in terms of polyhedral regions.

Fig. 6.4: Feedback law of the explicit controller (a)-(4) with parameters $x_{p,1} = v_x / X_{ch,v_x}$ fixed to 1 corresponding to a speed $v_x = 70 \text{ km/h}$, $x_{p,3} = F_z / P_{ch,F_z}$ fixed to 0.5 corresponding to a vertical tyre force $F_z = 5000 \text{ N}$, and $x_{p,4} = \alpha / P_{ch,\alpha}$ fixed to 1 corresponding to a tyre slip angle $\alpha = 17 \text{ deg}$.

In Fig. 6.3, the orthogonal partitions of the top layer and the polyhedral critical regions of the bottom layer of the explicit controller (a)-(4) are shown in the normalised longitudinal slip, tyre slip angle, and torque request parameter space. The parameters $x_{p,1} = v_x / X_{ch,v_x}$ are fixed to 1 corresponding to a speed $v_x = 70 \text{ km/h}$ and $x_{p,3} = F_z / P_{ch,F_z}$ fixed to 0.5 corresponding to a vertical tyre force $F_z = 5000 \text{ N}$. Scaling the parameter vector $X_p = [v_x, \kappa_x, F_z, \alpha, T_{req}]^T$ with the characteristic quantities $X_{ch,v_x}$, $X_{ch,\kappa_x}$, $P_{ch,F_z}$, $P_{ch,\alpha}$, and $P_{ch,T_{req}}$ in the non-dimensionalisation procedure helps to explore a numerically well conditioned mp-NLP with the Algorithm 1. In Fig. 6.3, the polyhedral critical regions reflect the soft constraints on the longitudinal slip and, moreover, the non-linear combined slip tyre model. Planes with normal vector in the approximate direction of the parameter axis $x_{p,2}$ are mainly due to the slip tracking objective. However, planes with normal vector in the approximate direction of the parameter axis $x_{p,4}$ and different inclinations in the direction of parameter axes $x_{p,4}$ and $x_{p,5}$ are because of the
non-linear tyre force model dependent on the tyre slip angle. In each polyhedral critical region, an affine feedback law is associated.

In Fig. 6.4, the receding horizon feedback law of the explicit controller (a)-(4) is shown over the normalised longitudinal slip and torque request parameter space. The parameters $x_{p,1} = v_x / X_{ch,v_x}$ are fixed to 1 corresponding to a speed $v_x = 70 \text{ km/h}$, $x_{p,3} = F_z / P_{ch,F_z}$ fixed to 0.5 corresponding to a vertical tyre force $F_z = 5000 \text{ N}$, and $x_{p,4} = \alpha / P_{ch,\alpha}$ fixed to 1 corresponding to a tyre slip angle $\alpha = 17^\circ$. The figure shows the dimensionless input $\hat{u}(t_k)$ which needs to be converted to a dimension-related quantity by applying the corresponding transformation law of the non-dimensionalisation procedure. From Fig. 6.4, the lower bounds on the torque reduction in the constraints (6.21) can be identified, i.e. the zero torque reduction $\Delta T$ for low values of torque request in the low slip region. Also the upper bound in (6.21) on the torque reduction is visible from the inclined plane parallel to the $x_{p,2}$ axis.

6.2.4 Conclusion and future work

In this section, the design, implementation, and performance assessment of an explicit NMPC-based traction controller for combined slip conditions is presented.

The simulation results demonstrate the influence of tyre slip angle on the longitudinal slip tracking performance, which improves by a factor $> 5$ on the vehicle level and by a factor $> 13$ on the wheel level, according to the defined indicator, for the controllers including a combined slip formulation, with respect to the case of a prediction model considering only longitudinal slip.

For the explicit solution, the influence of the parameter exploration space definition varying the vehicle speed, longitudinal slip, tyre slip angle, and torque request bounds on the complexity of the controller in terms of polyhedral regions and subsequently memory requirements is investigated showing an apparent, however, not significant effect. The complexity of the explicit controllers can be effectively reduced by the application of the post-processing algorithm including binary search tree generation, disjoint optimal and sub-optimal merging procedures, and clipping-based complexity reduction. The resulting memory requirements make a real-time implementation on embedded hardware with suitable memory specifications, e.g. a rapid control prototyping unit, feasible.

Future steps will focus on the revision and extension of the proposed explicit non-linear model predictive traction controller to meet the practical requirements for its experimental implementation on a vehicle demonstrator with multiple electric motors.
6.3 Anti-lock braking control

6.3.1 Introduction and literature review

Electro-hydraulic braking (EHB) systems are becoming viable solutions for conventional, hybrid electric, and fully electric vehicles, as demonstrated by the first successful implementations on series passenger cars (Zhu and Zong, 2009, Nakamura et al., 2002, and Ivanov et al., 2015c). In EHB systems, the brake pedal and wheel callipers are decoupled to allow pedal force feedback and, thus, pedal feel that is independent of the operating conditions of the braking system. In addition to this driver comfort benefit, EHB units permit continuous control of each calliper pressure. This is ideal for brake blending, i.e. the seamless and variable braking torque distribution between friction brakes and electric drivetrains. Also, compared to standard braking systems with vacuum booster, EHB systems allow relatively smaller packaging and faster response times, Savitski et al. (2016). The quicker response is beneficial to the performance of active safety functions such as electronic stability control (ESC), D’Alfio et al. (2006).

Anti-lock braking systems for passenger cars were developed to increase road safety by keeping the vehicle steerable and stable during intense braking events, especially on slippery roads surfaces (Savaresi and Tanelli, 2010 and Aly et al., 2011). The first implementations used rule-based algorithms considering the estimated slip ratio and wheel deceleration. These controllers were suitable for hydraulic units capable of generating sequences of pressure increase, decrease, and hold phases, but without the capability of continuous feedback control. Such algorithms were rather robust with respect to the possible operating conditions of the vehicle but provided sub-optimal performance in terms of wheel slip tracking. Since then, improvements have been gradually implemented, at the cost of increased tuning complexity, Choi (2008). Nevertheless, today’s industrial ABS control strategies are still based on a complex set of rules. EHB technology, similarly to electro-mechanical brake technology, permits more refined wheel slip controllers with continuous brake torque modulation, Savitski et al. (2018). Algorithms based on proportional-integral-derivative formulations, such as De Pinto et al. (2017) and Aparow et al. (2013), second-order sliding mode, as in Amodeo et al. (2010), as well as maximum transmissible torque estimation, as in Yin and Hori (2010), have been proposed for ABS or traction control. Fujimoto et al. (2012) discuss a selection of wheel slip controllers for electric vehicles, not requiring vehicle speed detection, while Wang et al. (2017) present slip controllers for split-$\mu$ conditions.

Recent literature shows increasing interest in model-based state feedback controllers, and especially in model predictive control. In Johansen et al. (2003), an ABS based on a gain scheduled linear quadratic regulator was tested on a vehicle with electro-mechanical brake callipers. Linear MPCs are discussed in Satzger et al. (2016), Satzger and de Castro
(2014), and de Castro et al. (2012), in the context of ABS including torque blending between friction brakes and in-wheel motors. In Satzger and De Castro (2018), an MPC strategy is compared with a PI controller and is assessed on an electric vehicle prototype. A linear MPC is also presented in Yoo and Wang (2007), with results from a hardware-in-the-loop rig including a brake-by-wire system. The study shows the deterioration of the longitudinal slip tracking performance during transitions from high to low tyre-road friction levels.

Borrelli et al. (2006) discuss a traction controller for an internal-combustion-engine-driven vehicle, and compares four linear MPC strategies with a hybrid explicit MPC. The performance of the hybrid strategy is comparable with that of a well-tuned PID controller. Basrah et al. (2017) present a real-time capable implicit non-linear model predictive controller for an ABS with torque blending and compares the NMPC simulation results with those from a linear MPC. In addition to the better performance of the non-linear solution, Basrah et al. (2017) report that the computational time for the NMPC is of 3-4 ms on a desktop personal computer, whereas the linear MPC requires approximately 1 ms. In Yuan et al. (2016), an implicit NMPC slip control strategy is assessed in simulation and implemented on a quad-core 2.8 GHz dSPACE unit yielding a computational time of 4-5 ms. Tavernini et al. (2019a) show that the implementation time step is more influential than the selected control technology (NMPC or PID) on the performance of a traction controller for an electric vehicle with in-wheel motors. Hence, controllers with tracking performance and low computing times are required for effective wheel slip control.

In this context, the paper Tavernini et al. (2019b) presents an explicit NMPC algorithm for ABS and its implementation on an EHB system. Explicit NMPC is selected as:

- According to many practitioners, MPC represents the future of automotive control, since this technology i) requires a lower number of calibration parameters than more conventional controllers, and, thus, reduces development times, as stated in Borrelli et al. (2006), Bemporad et al. (2018), and Siampis et al. (2018); ii) permits formal consideration of system constraints; and iii) allows preview control, which is of the essence in the future context of connected and autonomous vehicles. In such implementations, the tyre-road friction estimation could be enhanced by the information from the vehicles located in front of the ego vehicle, and in general, the characteristics of the road ahead are likely to be better known than in existing controllers. The eNMPC ABS of this research prepares the ground for these developments, as the tyre-road friction coefficient can be included as an input parameter varying in real-time.

- NMPC for ABS control offers benefits with respect to alternative control technologies, such as $H_{\infty}$ and sliding mode control, De Pinto et al. (2017). The main issue of $H_{\infty}$ control is that the range of variation of the longitudinal tyre slip stiffness is too wide
to be captured by a single controller based on a linear model with a fixed longitudinal slip stiffness. Moreover, wheel slip control interventions are becoming more frequent in modern stability controllers actuating the friction brakes, which tend to operate to improve the cornering response also in sub-limit conditions, as seamlessly as possible. Hence, it is desirable to have a controller capable of a rather smooth wheel slip control action, without the typical chattering issues of sliding mode controllers.

- Because of its explicit nature, the specific ABS eNMPC of this study requires only a fraction of the computing time (i.e. $< 0.1 \text{ ms}$ on a 900 MHz dSPACE automotive platform) of the implicit solutions in Basrah et al. (2017) and Yuan et al. (2016), and, therefore, can run at much smaller time steps than an equivalent implicit NMPC.

- In eNMPC, the explicit solution is known in advance, which allows carrying out a systematic a priori analysis of the control system performance, with benefits in terms of functional safety of the automotive system, with respect to the implicit NMPC.

### 6.3.2 Contribution

The points of novelty of the study presented in Tavernini et al. (2019b) are:

- The design and experimental implementation of a proof-of-concept explicit NMPC ABS based on continuous wheel slip control. The algorithm takes the experimentally measured dead time of the hydraulic components into account through a compensation strategy.

- The comparison of the explicit NMPC ABS with a benchmark PID controller, including robustness assessment with respect to tyre-road friction conditions and initial vehicle speed.

### 6.3.3 Conclusion

The study in Tavernini et al. (2019b) discusses the proof-of-concept design of an explicit non-linear model predictive controller for anti-lock braking systems, its implementation on an industrial electro-hydraulic braking unit, and the experimental comparison with a PID ABS. The interested reader is referred to the publication for details on the results. The analysis leads, however, to the following conclusions:

- The experimental step response of the case study electro-hydraulic braking system shows significant variations of the pressure dead time and rise time as functions of the final pressure value. As the measured dead times of approximately 20 ms are much longer than the ABS controller time step of 3 ms, the proposed dead time compensation strategy is an important component of the explicit NMPC ABS.
• The experimental braking test results show the explicit NMPC robustness with respect
  to the tyre-road friction conditions and initial speed. Satisfactory performance was
  obtained with a relatively simple internal model formulation that did not consider ac-
  tuation dynamics nor vertical load variation. Conversely, the dead time compensation
  strategy was necessary to ensure the correct performance of the controller.

• The online computation time for the explicit solution was assessed on an automotive
  rapid control prototyping unit to be < 95 μs, which confirms the real-time capability of
  the explicit NMPC ABS for any implementation time step typical of ABS applications.
  The memory requirements are also in line with available automotive micro-controller
  units (up to 16 MB).

• The explicit NMPC ABS consistently outperforms the PID ABS, e.g. it reduces
  the stopping distance in low tyre-road friction conditions by up to 11.4%. These
  results make explicit NMPC a promising technology for automotive wheel slip control
  applications.
Chapter 7

Anti-jerk control

7.1 Introduction and literature review

Anti-jerk control

Abrupt traction torque variations can induce mechanical resonance and oscillations in vehicle drivetrains in the 5 Hz to 50 Hz frequency range, Amann et al. (2004). These oscillations are caused by the torsional compliance of the driveline components, e.g. the half-shafts, Bottiglione et al. (2012), and are emphasised by the inevitable presence of mechanical play, also known as a backlash, in the transmission gears, Pham et al. (2016). In general, the torsional oscillations of automotive drivetrains cause i) oscillations of the longitudinal vehicle acceleration, which compromise passenger comfort, and ii) premature hardware wear, Pham et al. (2016).

Internal combustion engine (ICE) vehicles, hybrid electric vehicles, and electric vehicles with on-board electric motors (EMs) can suffer from torsional drivetrain oscillations. However, ICE drivetrains are usually fitted with additional inertias and mechanical dampers to mitigate the intrinsic unevenness of the ICE torque delivery. The absence of these mechanical devices in on-board electric powertrains, which are the focus of this study, makes them especially prone to torsional oscillations, Böcker et al. (2004). Moreover, EMs have significantly higher bandwidth than ICES, which further facilitates the excitation of the drivetrain torsional dynamics in conditions of swift torque demand variations.

Anti-jerk controllers (AJC) mitigate such behaviour, by applying corrections to the traction torque, $T_{req}$, requested by the driver or a higher level controller. The anti-jerk control actively suppresses the torsional oscillations of the automotive drivetrains, caused by abrupt variations of the traction torque. The main benefits are enhanced passengers' comfort and increased component life. However, this control action may have a counterproductive effect on acceleration performance and/or vehicle responsiveness. These facts are also the main drawbacks of anti-jerk control.
The literature presents a wide range of anti-jerk controllers, which can be divided into five main categories:

- **Feedforward controllers**, i.e. filters on $T_{req}$. For example, the implementations in Millo et al. (2003) and Stewart and Fleming (2004) consist only of the feedforward contribution. However, it is also very common to combine the feedforward controller with a feedback contribution (Kawamura et al. 2011, Orus et al. 2014, and Ivanov et al. 2015b), e.g. to compensate for external disturbances and model mismatches.

- **Feedback controllers based on the oscillating component of the EM speed**, $\dot{\theta}_{EM,vib}$, or ICE speed (Kawamura et al. 2011, Bang 2014, De La Salle et al. 2004, Xu and Farkas 2000). According to the experience of the authors in collaborative projects with industry, this anti-jerk control method is widely adopted in production vehicles.

- **Feedback controllers based on the drivetrain torsion rate**, $\Delta \dot{\theta}$, which is calculated from the measured EM speed, $\dot{\theta}_{EM}$, or ICE speed, and the driven wheel speed, $\dot{\theta}_w$. $\dot{\theta}_w$ is not currently used in production anti-jerk controllers. The simplest control structure of this kind, also known as tachometric controller (Rodriguez et al., 2013), feeds $\Delta \dot{\theta}$ to a proportional controller that calculates the corrective anti-jerk torque, i.e. the controller acts as a virtual linear damper between EM and wheel, as in Orus et al. (2014) and Ivanov et al. (2015b). Yamada et al. (2018) add a switching logic to the tachometric controller, to cope with the drivetrain backlash. In general, the feedback on $\Delta \dot{\theta}$ can be based on any control structure, e.g. proportional-derivative control in Webersinke et al. (2007), $H_\infty$ control in Baumann et al. (2005), and implicit model predictive control in Xiaohui et al. (2011).

- **Feedback controllers based on the drivetrain torsion angle**, $\Delta \theta$, or drivetrain torque, $T_d$. Due to the usually negligible damping of the electric drivetrain components, in a first approximation it is $T_d \approx K_d \Delta \theta$, where $K_d$ is the equivalent torsional stiffness of the drivetrain, Amann et al. (2004). Therefore, the feedback controllers based on $\Delta \theta$ and $T_d$ can be grouped together. For example, Angeringer et al. (2012) propose a sliding mode controller on $\Delta \theta$, Amann et al. (2004) and Böcker et al. (2004) present pole placement controllers on $T_d$, and Lv et al. (2015) discuss a proportional-integral-derivative controller on $T_d$.

- **Feedback controllers based on the longitudinal vehicle acceleration**, $a_x$. Only a few studies adopt $a_x$ as control variable. The difficulty of dealing with $a_x$ is primarily due to the considerable uncertainty and noise associated with its measurement, which is conducted by inertial measurement units and is affected by the vehicle sprung mass resonance (Murray, 2013) and road slope (Bisoffi et al., 2017). To the best of the author’s knowledge, the only experimentally validated anti-jerk controller of this
category is described in Stewart et al. (2005), and calculates $a_x$ from an unspecified estimation of vehicle speed, $v_x$.

Focusing on model predictive anti-jerk controllers in the literature, the prediction models range from two-inertia linear models as in Xiaohui et al. (2011) and Zhang et al. (2017), to more complex formulations including backlash in Lagerberg and Egardt (2005), tyre relaxation in Batra et al. (2018b) and Batra et al. (2018a), and non-linear tyre behaviour in Batra et al. (2018a). For the introduced categories, MPC controllers can be classified differently depending on the control objectives and the inputs to the controller. From the viewpoint of the inputs, all mentioned controllers are feedback controllers based on the drivetrain torsion angle and the torsion rate, additionally with the longitudinal vehicle speed in Batra et al. (2018b) and Batra et al. (2018a) for wheel slip estimation. Regarding the control objectives, Xiaohui et al. (2011), Zhang et al. (2017), and Lagerberg and Egardt (2005) minimise the torsion rate, and Lagerberg and Egardt (2005), Batra et al. (2018b), and Batra et al. (2018a) the torsion angle among other criteria.

**Contribution**

Despite the vast amount of published work, a structured comparison and an objective and comprehensive assessment of the performance of different anti-jerk controllers is still missing. To cover this gap, the work in Scamarcio et al. (2019a) uses electric vehicle simulations to assess the performance of six anti-jerk controllers. The comparison includes five exemplary anti-jerk controllers from the literature and one novel formulation based on explicit non-linear model predictive control. All proposed control structures have the potential to be implemented on production vehicles. A set of objective performance indicators is defined to assess the controllers, which are tuned through an optimisation-based routine. For the controller design and performance assessment, a frequency domain model and a time domain model are employed.

Model predictive control has been applied to anti-jerk control problems, adopting a variety of prediction models. The prediction models should capture effectively the drivetrain dynamics. At the same time, a prediction model with a low number of states and parameters facilitates controller development and real-time implementation. To the best of the author’s knowledge, an analysis of the influence of the prediction model complexity on the anti-jerk controller performance is still missing in the literature. To cover such a knowledge gap, the work in Scamarcio et al. (2019b) proposes six different anti-jerk MPC formulations for electric vehicles with multiple on-board motors using a unified optimisation-based tuning method for the control system design. The study adopts objective performance indicators to assess the influence of different prediction models and the performance of the model predictive anti-jerk controllers in different tip-in and tip-out manoeuvres.
7.2 Control system design

In the following sections, all necessary definitions for the optimal control problem of the model predictive anti-jerk controller are introduced.

7.2.1 Prediction models

Different simplified models of the power-train with two on-board motors of the front-wheel-drive electric vehicle are introduced as the prediction model. The main characteristics of the varying model complexity lie in the number of equivalent inertias representing the motor, gearbox, half-shaft, wheel, and vehicle inertia. Moreover, the modelling of backlash and the formulation of tire slip behaviour with relaxation are further differentiating features.

Two-inertia model with three states

The prediction model $M_3$, adopted in Xiaohui et al. (2011) and Zhang et al. (2017), has two inertias: The first mass moment of inertia, $J_1$, includes the EM rotor inertia, $J_{EM}$, the inertia of the primary and secondary gearbox shafts, $J_{g1}$ and $J_{g2}$, respectively, and half of the half-shaft inertia, $J_{hs}$,

$$J_1 = J_{EM} i_g^2 + J_{g1} i_g^2 + J_{g2} + \frac{1}{2} J_{hs},$$

where $i_g$ is the gear ratio. The second inertia, $J'_2$, includes the inertia of half of the half-shaft, $J_{hs}$, the driven wheel, $J_w$, half of the vehicle mass, $M_v$, and the non-driven wheel, $J'_w$,

$$J'_2 = \frac{1}{2} J_{hs} + 2 J_w + \frac{1}{2} M_v R_w^2,$$

where $R_w$ is the wheel radius.
The states of the model $M_3$ are the angular speed of the gearbox output, $\dot{\theta}_1(t)$, the average angular speed of the front wheels, $\dot{\theta}_2(t)$, and the drivetrain torsion angle, $\Delta \theta(t) = \theta_1(t) - \theta_2(t)$. The angular speeds of the two inertias, $\dot{\theta}_1(t)$ and $\dot{\theta}_2(t)$, are, therefore, related to the measured motor and wheel speeds,

$$\dot{\theta}_1(t) = \frac{\dot{\theta}_{EM}(t)}{i_g},$$
$$\dot{\theta}_2(t) = \dot{\theta}_w(t).$$

The equations of motion are described by (7.5)-(7.7),

$$\ddot{\theta}_1(t) = \frac{1}{J_1} \left( T_1(t) - T_{hs}(t) \right),$$
$$\ddot{\theta}_2(t) = \frac{1}{J'_2} \left( T_{hs}(t) - \frac{1}{2} T_{aero}(t_k) - \frac{1}{2} T_{roll}(t_k) \right),$$
$$\Delta \dot{\theta}(t) = \dot{\theta}_1(t) - \dot{\theta}_2(t),$$

where $T_1(t)$ is the motor torque at the gearbox output shaft, defined as

$$T_1(t) = i_g T_{EM}(t),$$

with the electric motor torque, $T_{EM}(t)$. $T_{hs}(t)$ is the half-shaft torque, and $T_{aero}(t_k)$ and $T_{roll}(t_k)$ are the torque contributions related to aerodynamic drag and rolling resistance.

### Three-inertia model with four states

The prediction model $M_4$ includes a steady-state longitudinal tyre force model (without relaxation) and has, therefore, three inertias: The first mass moment of inertia, $J_1$, which is equivalent to (7.1). The second mass moment of inertia, $J_2$, includes the inertia of half of the half-shaft, $J_{hs}$, and the driven wheel, $J_w$,

$$J_2 = \frac{1}{2} J_{hs} + J_w,$$

and the third mass moment of inertia, $J_3$, includes the inertia of half of the vehicle mass, $M_v$, and the non-driven wheel, $J_w$,

$$J_3 = \frac{1}{2} M_v R_w^2 + J_w.$$

The states of the model $M_4$ are the angular speed of the gearbox output, $\dot{\theta}_1(t)$, the average angular speed of the front wheels, $\dot{\theta}_2(t)$, the drivetrain torsion angle, $\Delta \theta(t)$, and the longitudinal vehicle speed, $v_x(t)$.
The equations of motion are described by (7.11)-(7.14),

\[ \ddot{\theta}_1(t) = \frac{1}{J_1} \left( T_1(t) - T_{hs}(t) \right), \quad (7.11) \]

\[ \ddot{\theta}_2(t) = \frac{1}{J_2} \left( T_{hs}(t) - F_x(t) R_w - \frac{1}{4} T_{roll}(t_k) \right), \quad (7.12) \]

\[ \Delta \dot{\theta}(t) = \dot{\theta}_1(t) - \dot{\theta}_2(t), \quad (7.13) \]

\[ \dot{v}_x(t) = \frac{R_w}{J_3} \left( F_x(t) R_w - \frac{1}{2} T_{aero}(t_k) - \frac{1}{4} T_{roll}(t_k) \right), \quad (7.14) \]

The longitudinal tyre force, \( F_x \), is defined by a simple linear model,

\[ F_x(t) = K_{xx} \kappa_x(t), \quad (7.15) \]

where \( K_{xx} \) is the slip stiffness, which is considered constant, and \( \kappa_x(t) \) is the longitudinal slip in traction, defined as

\[ \kappa_x(t) = \frac{\dot{\theta}_2(t) R_e - v_x(t)}{\theta_2(t) R_e}, \quad (7.16) \]

with constant effective radius, \( R_e \), in free rolling conditions.

**Three-inertia model with five states**

The prediction model \( M_5 \), employed by Batra et al. (2018b), includes a transient longitudinal tyre force model with relaxation. Analogously to \( M_4 \), the prediction model \( M_5 \) has three inertias: \( J_1 \) as defined in (7.1), \( J_2 \) equivalent to (7.9), and \( J_3 \) equivalent to (7.10).

The states of the model \( M_5 \) are the angular speed of the gearbox output, \( \dot{\theta}_1(t) \), the average angular speed of the front wheels, \( \dot{\theta}_2(t) \), the drivetrain torsion angle, \( \Delta \dot{\theta}(t) \), the longitudinal vehicle speed, \( v_x(t) \), and the delayed longitudinal slip, \( \kappa_x'(t) \).

The equations of motion are defined by (7.17)-(7.21),

\[ \ddot{\theta}_1(t) = \frac{1}{J_1} \left( T_1(t) - T_{hs}(t) \right), \quad (7.17) \]

\[ \ddot{\theta}_2(t) = \frac{1}{J_2} \left( T_{hs}(t) - F_x(t) R_w - \frac{1}{4} T_{roll}(t_k) \right), \quad (7.18) \]

\[ \Delta \dot{\theta}(t) = \dot{\theta}_1(t) - \dot{\theta}_2(t), \quad (7.19) \]

\[ \dot{v}_x(t) = \frac{R_w}{J_3} \left( F_x(t) R_w - \frac{1}{2} T_{aero}(t_k) - \frac{1}{4} T_{roll}(t_k) \right), \quad (7.20) \]

\[ \dot{\kappa}_x'(t) = \frac{v_x(t)}{\sigma_\kappa} \left( \kappa_x(t) - \kappa_x'(t) \right), \quad (7.21) \]
where (7.17)-(7.20) are analogous to (7.11)-(7.14). However, the longitudinal tyre force, $F_x$, is defined by a linear transient model,

$$F_x(t) = K_{xx} \kappa_x'(t),$$  \hspace{1cm} (7.22)

where the delayed longitudinal slip, $\kappa_x'(t)$, is calculated according to (7.21) assuming constant relaxation length, $\sigma_\kappa$.

**Drivetrain torque**

For the estimation of the half-shaft torque, $T_{hs}$, given by the sum of a stiffness contribution and a small damping contribution, two different models are considered:

A.i Assuming no backlash, the half-shaft torque is defined as

$$T_{hs}(t) = K_d \Delta \theta(t) + C_d \left( \dot{\theta}_1(t) - \dot{\theta}_2(t) \right),$$  \hspace{1cm} (7.23)

where $K_d$ is the equivalent torsional stiffness at the wheel, and $C_d$ the damping coefficient of the drivetrain.

A.ii Including backlash in the formulation, the half-shaft torque becomes

$$T_{hs}(t) = \frac{1}{2} K_d \left( \Delta \theta(t) - \frac{1}{2} \theta_{BL} \right) \left( \tanh \left( K_1 (\Delta \theta(t) - K_2) \right) + 1 \right) + \frac{1}{2} K_d \left( \Delta \theta(t) + \frac{1}{2} \theta_{BL} \right) \left( \tanh \left( -K_1 (\Delta \theta(t) + K_2) \right) + 1 \right) + C_d \left( \dot{\theta}_1(t) - \dot{\theta}_2(t) \right),$$  \hspace{1cm} (7.24)
with tuning parameters $K_1$ and $K_2$ of the backlash model, and $\theta_{BL}$ for the nominal backlash, measured at the wheel. The drivetrain torque with backlash model (7.24) is demonstrated in Fig. 7.2.

**Aerodynamic and rolling resistance**

The wheel torque contribution related to the aerodynamic drag is estimated as

$$T_{aero}(t) = \frac{1}{2} \rho A C_x v(t_k)^2 R_w,$$

where $\rho$ is the air density, $A$ is the frontal area of the vehicle, and $C_x$ is the aerodynamic drag coefficient.

The wheel torque contribution related to the rolling resistance is calculated as

$$T_{roll}(t) = \left( f_0 M_v g + f_2 M_v g v(t_k)^2 \right) R_w,$$

where $f_0$ and $f_2$ are the tyre rolling resistance coefficients, and $g$ is the gravitational acceleration.

In Eq. (7.6) of the two-inertia model with three states, the vehicle speed is estimated as $v(t_k) \approx \dot{\theta}_2(t_k) R_w$ for both resistance terms (7.25) and (7.26), respectively.

**Model parameters**

For the identification of the prediction model parameters, least squares regressions are performed. With the parameters $K_1$ and $K_2$, the backlash model is fitted against the simulation model, as shown in Fig. 7.2. The parameters $K_d$ for the torsional drivetrain stiffness and $C_d$ for the drivetrain damping are identified for each prediction model separately, i.e. $M_3, M_3^{BL}, M_4, M_4^{BL}, M_5, \text{ and } M_5^{BL}$ as summarised in Table 7.1, by means of a regression analysis employing the respective state trajectories. In the prediction models $M_3$ and $M_3^{BL}$, the angular speed of the gearbox output, $\dot{\theta}_1(t)$, the average angular speed of the front wheels, $\dot{\theta}_2(t)$, and the drivetrain torsion angle, $\Delta \theta(t)$, are fitted against the simulation model. The errors in the states are scaled to obtain numerical values in the same order of magnitude. Additionally, for the prediction models $M_4, M_4^{BL}, M_5, \text{ and } M_5^{BL}$, the longitudinal vehicle speed, $v_x(t)$, and longitudinal slip, $\kappa_x(t)$, are fitted against the simulation model to identify the parameters $K_{\kappa x}$ for the slip stiffness and $\sigma_{\kappa}$ for the relaxation length.
7.2 Control system design

Overview

In Table 7.1, an overview of the prediction models with the different state vectors, $X(t)$, is presented. In the remainder, the controllers are named after their prediction models, e.g. controller $C_3$ refers to the MPC implementation using the prediction model $M_3$.

Table 7.1: Prediction models.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Prediction model</th>
<th>State vector $X$</th>
<th>Drivetrain torque $T_{hs}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_3$</td>
<td>$M_3$</td>
<td>$\dot{\theta}_1, \dot{\theta}_2, \Delta \theta$</td>
<td>A.i (no backlash)</td>
</tr>
<tr>
<td>$C_{3BL}$</td>
<td>$M_{3BL}$</td>
<td>$\dot{\theta}_1, \dot{\theta}_2, \Delta \theta$</td>
<td>A.ii (with backlash)</td>
</tr>
<tr>
<td>$C_4$</td>
<td>$M_4$</td>
<td>$\dot{\theta}_1, \dot{\theta}_2, \Delta \theta, v_x$</td>
<td>A.i (no backlash)</td>
</tr>
<tr>
<td>$C_{4BL}$</td>
<td>$M_{4BL}$</td>
<td>$\dot{\theta}_1, \dot{\theta}_2, \Delta \theta, v_x$</td>
<td>A.ii (with backlash)</td>
</tr>
<tr>
<td>$C_5$</td>
<td>$M_5$</td>
<td>$\dot{\theta}_1, \dot{\theta}_2, \Delta \theta, v_x, \kappa'_x$</td>
<td>A.i (no backlash)</td>
</tr>
<tr>
<td>$C_{5BL}$</td>
<td>$M_{5BL}$</td>
<td>$\dot{\theta}_1, \dot{\theta}_2, \Delta \theta, v_x, \kappa'_x$</td>
<td>A.ii (with backlash)</td>
</tr>
</tbody>
</table>

Fig. 7.3 shows the topology of the explicit non-linear model predictive anti-jerk controller. The input quantities are estimated by a Kalman filter, that is specified in more detail in Scamarcio et al. (2019a).

![Block diagram of the explicit non-linear MPC](image)

Fig. 7.3: Block diagram of the explicit non-linear MPC.

The motor torque request, $T_{req}$, is modified by the controller by applying the anti-jerk torque correction, $T_{corr}$, resulting, therefore, in the torque request to the plant, $T_{plant}$.

The measured signals required by the control system must be easily available on production vehicles. Hence, in this study the inputs for the state estimator and the controller are the torque request, $T_{req}$, the electric motor torque estimated by the inverter, $T_{EM}$, electric motor speed, $\dot{\theta}_{EM}$, wheel speed, $\dot{\theta}_w$, and potentially longitudinal vehicle acceleration, $a_x$. 
7.2.2 Optimal control problem

The formulation of the non-linear model predictive anti-jerk controller is based on an optimal control problem defined by a cost function and constraints, in addition to the prediction model equations in Section 7.2.1.

Cost function

The contradicting performance requirements of maximising comfort, i.e. reducing the drivetrain oscillation which is the root cause of vehicle jerk (De Novellis et al., 2015), and closely tracking the torque demand, i.e. maintaining high vehicle responsiveness, are incorporated in the cost function, $V$, which is equivalent to the formulations in Xiaohui et al. (2011) and Zhang et al. (2017). $V$ includes a term based on the drivetrain torsion rate, $\Delta \dot{\theta}(t) = \dot{\theta}_1(t) - \dot{\theta}_2(t)$, and a term based on the deviation of the anti-jerk torque output from the reference torque of the high-level controller:

$$V(t_k) = \int_{t_k}^{t_f} \left[ \frac{q_x}{X_{sc}^2} (\dot{\theta}_1(t) - \dot{\theta}_2(t))^2 + \frac{r_u}{U_{sc}^2} (T_{req}(t_k) - T_{plant}(t))^2 \right] dt,$$  \hspace{1cm} (7.27)

where $q_x$ and $r_u$ are weights, which are optimised according to the routine in Section 7.3.2, and $X_{sc}$ and $U_{sc}$ are scaling factors. The integral in (7.27) is defined over the prediction horizon $[t_k, t_f]$. $t_k$ is the current time, and $t_f = t_k + N_p h$ is the final time, where $N_p$ is the number of prediction steps and $h$ is the discretisation time.

Constraints

To account for the actual EM limitations, appropriate constraints are incorporated in the optimal control problem formulation,

$$T_{EM,min} \leq T_{plant}(t) \leq T_{EM,max},$$  \hspace{1cm} (7.28)

$$P_{EM,min} \leq T_{plant}(t) \dot{\theta}_{EM}(t_k) \leq P_{EM,max},$$  \hspace{1cm} (7.29)

where $T_{EM,max}$, $P_{EM,max}$, $T_{EM,min}$, and $P_{EM,min}$ are the maximum values of torque and power in traction and regeneration, respectively. With (7.3), the electric motor speed $\dot{\theta}_{EM}(t)$ can be related to the angular speed $\dot{\theta}_1(t)$. Moreover, in (7.28) and (7.29), $T_{plant}(t)$ is assumed equal to $T_{EM}(t)$.

State transformation

For the optimisation problem, the angular speed of the gearbox output, $\dot{\theta}_1(t)$, angular speed of the wheels, $\dot{\theta}_2(t)$, drivetrain torsion angle, $\Delta \theta(t)$, (for all controllers), longitudinal vehicle speed, $v_x(t)$, (additionally for controllers $C_4$-$C_{BL}$), and relaxed slip, $\kappa'_x(t)$,
(additionally for controllers \( C_5 \) and \( C_5^{BL} \)) are required as initial conditions for the dynamic equality constraints (7.5)-(7.7), (7.11)-(7.14), and (7.17)-(7.21), respectively. The torque request, \( T_{req}(t_k) \), is considered as varying parameter and, therefore, constant over the prediction horizon.

The application of state transformations turned out to be beneficial for the state-space exploration in the chosen explicit approach. With the description based on the new variables, employing the transformation laws,

\[
\begin{align*}
X_1(t) &= \dot{\theta}_2(t), \\
X_2(t) &= \Delta \dot{\theta}(t) = \dot{\theta}_1(t) - \dot{\theta}_2(t), \\
X_3(t) &= \Delta \theta(t), \\
X_4(t) &= \kappa_x(t) = \frac{\dot{\theta}_2(t) R_e - v_x(t)}{\theta_2(t) R_e}, \\
X_5(t) &= \kappa'_x(t),
\end{align*}
\]

the exploration space can be reduced to hyper-rectangles avoiding parameters in the exploration space that are not likely to occur in the closed-loop operation of the controller.

**Settings**

Table 7.2 reports the settings for the optimal control problem of this study.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H )</td>
<td>Discretisation time</td>
<td>1 ms, 5 ms, 10 ms</td>
</tr>
<tr>
<td>( T_s )</td>
<td>Sampling time</td>
<td>1 ms, 5 ms, 10 ms</td>
</tr>
<tr>
<td>( N_p )</td>
<td>Prediction steps</td>
<td>2, 4, 6, 8, 10</td>
</tr>
<tr>
<td>( N_c )</td>
<td>Control steps</td>
<td>2</td>
</tr>
<tr>
<td>( q_x )</td>
<td>Weight on drivetrain torsion rate</td>
<td>Tuning routine 7.3.2</td>
</tr>
<tr>
<td>( r_u )</td>
<td>Weight on torque tracking</td>
<td>10</td>
</tr>
<tr>
<td>( X_{sc} )</td>
<td>Scaling factor for drivetrain torsion rate</td>
<td>20 deg/s</td>
</tr>
<tr>
<td>( U_{sc} )</td>
<td>Scaling factor for torque tracking error</td>
<td>75 Nm</td>
</tr>
<tr>
<td>( X_{ch,\dot{\theta}} )</td>
<td>Characteristic angular wheel speed</td>
<td>1250 deg/s</td>
</tr>
<tr>
<td>( X_{ch,\Delta\theta} )</td>
<td>Characteristic drivetrain torsion rate</td>
<td>115 deg/s</td>
</tr>
<tr>
<td>( X_{ch,\Delta\theta} )</td>
<td>Characteristic drivetrain torsion angle</td>
<td>3.4 deg</td>
</tr>
<tr>
<td>( X_{ch,\kappa_x} )</td>
<td>Characteristic steady-state longitudinal slip</td>
<td>13%</td>
</tr>
<tr>
<td>( X_{ch,\kappa'_x} )</td>
<td>Characteristic delayed longitudinal slip</td>
<td>13%</td>
</tr>
<tr>
<td>( P_{ch,T_{req}} )</td>
<td>Characteristic torque request</td>
<td>100 N m</td>
</tr>
<tr>
<td>( U_{ch,TEM} )</td>
<td>Characteristic electric motor torque</td>
<td>150 N m</td>
</tr>
</tbody>
</table>
The weight $r_u$ on the torque tracking term in the cost function (7.27) is fixed and the weight $q_z$ on the torsion rate term is optimised according to the tuning routine presented in Section 7.3.2.

**Multi-parametric non-linear program**

The continuous formulation of the optimal control problem with dynamic constraints is parametrised and discretised, and expressed in the general form,

$$V^*(x_p(t_k)) = \min_z V(z, x_p(t_k)),$$

subject to

$$G(z, x_p(t_k)) \leq 0.$$ (7.36)

which represents a multi-parametric non-linear programming problem, as introduced in Section 3.4. The parameter vector, $x_p(t_k) = [x^T(t_k), p^T(t_k)]^T$, combines the initial conditions, $x(t_k)$, of the ordinary differential equations of the system model, and the tracking reference, $p(t_k) = T_{req}(t_k)/P_{ch,T_{req}}$. The initial conditions vary depending on the prediction model and are $x(t_k) = \begin{bmatrix} \dot{\theta}_2(t_k)/X_{ch,\dot{\theta}_2}; \Delta \dot{\theta}(t_k)/X_{ch,\Delta \dot{\theta}}; \Delta \theta(t_k)/X_{ch,\Delta \theta} \end{bmatrix}^T$ for controllers $C_3$ and $C_{BL}^3$, $x(t_k) = \begin{bmatrix} \dot{\theta}_2(t_k)/X_{ch,\dot{\theta}_2}; \Delta \dot{\theta}(t_k)/X_{ch,\Delta \dot{\theta}}; \Delta \theta(t_k)/X_{ch,\Delta \theta}; \kappa_x(t)/X_{ch,\kappa_x} \end{bmatrix}^T$ for controllers $C_4$ and $C_{BL}^4$, and $x(t_k) = \begin{bmatrix} \dot{\theta}_2(t_k)/X_{ch,\dot{\theta}_2}; \Delta \dot{\theta}(t_k)/X_{ch,\Delta \dot{\theta}}; \Delta \theta(t_k)/X_{ch,\Delta \theta}; \kappa_x(t)/X_{ch,\kappa_x}; \kappa'_x(t)/X_{ch,\kappa'_x} \end{bmatrix}^T$ for controllers $C_5$ and $C_{BL}^5$. All variables in the optimal control problem are non-dimensionalised according to the procedure introduced in Section 3.2.2 with appropriate choices for the characteristic quantities, e.g. $U_{ch,T_{EM}}$, $X_{ch,\dot{\theta}_2}$, $X_{ch,\Delta \dot{\theta}}$, $X_{ch,\kappa_x}$, $X_{ch,\kappa'_x}$, etc. In the sequential approach, the vector of decision variables, $z = U$, contains the vector of input trajectory parameters. $U = [T_{plant}(t_k)/U_{ch,T_{EM}}; \ldots; T_{plant}(t_{k+N_c-1})/U_{ch,T_{EM}}]^T$ includes the prediction model inputs, which can vary $N_c$ times over the horizon, where $N_c$ is the number of control steps, and then are kept constant. In this application, the simultaneous approach can equally be employed for the discretisation of the optimal control problem. Then, the vector of decision variables, $z = [U^T, X^T]^T$, combines the vector of input trajectory parameters, $U$, and the vector of state trajectory parameters, $X$. $X = [x(t_{k+1})^T, \ldots, x(t_{k+N_p})^T]^T$ combines the intermediate predicted states, which are treated as additional optimisation variables.

In the receding horizon approach, the optimal solution, $z^*$, of (7.35)-(7.36) is calculated at each time step. The optimal torque request to the plant, $T^*_{plant}(t_k)$, is extracted from the first element of $U$ and applied to the plant. The iteration of the process makes the control system a closed-loop approach.
7.3 Performance analysis

In the following sections, the performance of the non-linear model predictive anti-jerk controller is assessed in various conditions. To this end, a set of performance indicators and a tuning procedure is introduced. The variation in the prediction model complexity, and the influence of the prediction horizon and sampling time on the performance is investigated. Finally, an explicit solution is generated and the complexity of the controller is analysed with different choices for the prediction model and parameter vector.

7.3.1 Simulation model

The case study plant is a front-wheel-drive electric vehicle with two on-board motors, each one connected to the respective wheel through a single-gear transmission and a flexible half-shaft. Table 7.3 shows the main vehicle parameters.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_v$</td>
<td>Vehicle mass</td>
<td>2140 kg</td>
</tr>
<tr>
<td>$P_{EM,max}$</td>
<td>Maximum electric motor power</td>
<td>37.5 kW</td>
</tr>
<tr>
<td>$T_{EM,max}$</td>
<td>Maximum electric motor torque</td>
<td>150 N m</td>
</tr>
<tr>
<td>$i_g$</td>
<td>Gear ratio</td>
<td>9.73</td>
</tr>
<tr>
<td>$J_{EM}$</td>
<td>Electric motor inertia</td>
<td>0.1 kg m$^2$</td>
</tr>
<tr>
<td>$J_w$</td>
<td>Wheel inertia</td>
<td>2.72 kg m$^2$</td>
</tr>
<tr>
<td>$R_w$</td>
<td>Wheel radius</td>
<td>0.36 m</td>
</tr>
<tr>
<td>$K_d$</td>
<td>Drivetrain stiffness</td>
<td>22 600 N m/rad</td>
</tr>
<tr>
<td>$C_d$</td>
<td>Drivetrain damping</td>
<td>90 N m s/rad</td>
</tr>
<tr>
<td>$\theta_{BL}$</td>
<td>Drivetrain play measured at the wheels</td>
<td>2 deg</td>
</tr>
</tbody>
</table>

Tyre dynamics are modelled with the Pacejka Magic Formula, coupled with the relaxation length formulation proposed by Giangiulio and Arosio (2006), which has been modified to give plausible results at low vehicle speeds. The relaxation length, $\sigma_\kappa$, is a function of the vertical tyre load, $F_z$, and the longitudinal slip ratio, $\kappa_x$:

$$
\sigma_\kappa(F_z, \kappa_x) = \left(\sigma_{\kappa 0}(F_z) - X_1\right) \exp\left(-X_2|\kappa_x|\right) + X_1,
$$

$$
X_1 = a_{x,1} + a_{x,2} \, df_z,
$$

$$
X_2 = a_{x,3} + a_{x,4} \, df_z,
$$

$$
\sigma_{\kappa 0}(F_z) = F_z \left(p_{Tx1} + p_{Tx2} \, df_z\right) \frac{R_0}{F_{z0}},
$$

$$
df_z = \frac{F_z - F_{z0}}{F_{z0}},
$$
where $F_{z0}$ is the nominal wheel load, $R_0$ the unloaded radius, and $a_{x,1}$, $a_{x,2}$, $a_{x,3}$, $a_{x,4}$, $p_{Tx1}$, and $p_{Tx2}$ are constants tuned to match the experimental results provided by Giangiulio and Arosio (2006). The delayed slip, $\kappa'_{x}$, is then defined by the differential equation as

$$\dot{\kappa'}_{x} = -\max\left(\frac{\sigma_{\kappa} + |v_{x}|}{\sigma_{\kappa}}, \epsilon\right) \kappa'_{x} + \frac{v_{x}}{\sigma_{\kappa}} \text{sgn}(v_{x}) \kappa_{x}.$$  (7.42)

Eq. (7.42) differs from the model proposed in Giangiulio and Arosio (2006) as the maximum function, $\max(\cdot)$, and a small positive threshold, $\epsilon$, is introduced to ensure positive definiteness of the time constant and, therefore, the causality of the first-order system.

### 7.3.2 Performance indicators and controller tuning

#### Performance indicators

For an objective assessment of the performance and a unified tuning procedure of the controllers, the following performance indicators are introduced (see also the ISO standards ISO 2631-1, 1997 and ISO 8041-1, 2017):

- The fourth power vibration dose value, $VDV_{a_{x}^{*}}$, evaluating the vehicle comfort level,

$$VDV_{a_{x}^{*}} = \sqrt[4]{\int_{T_1}^{T_3} a_{x}^{*}(t)^4 dt},$$  (7.43)

where $T_1 = 0.1$ s indicates the beginning of the manoeuvre and $T_3 = 0.6$ s the end of the simulation interval.

- The root-mean-square (RMS) value of the longitudinal acceleration, $RMS_{a_{x}^{*}}$, which also accounts for the comfort level,

$$RMS_{a_{x}^{*}} = \sqrt{\frac{1}{T_3 - T_1} \int_{T_1}^{T_3} a_{x}^{*}(t)^2 dt},$$  (7.44)

where $a_{x}^{*}(t)$ is the high-frequency component of the longitudinal vehicle acceleration, filtered with a high-pass Butterworth filter with a cut-off frequency of 8 Hz, chosen to capture the drivetrain oscillations and to avoid low frequency content due to the variation in the torque request.

- $\Delta v_{x}$, which quantifies the degradation of the longitudinal acceleration performance caused by the anti-jerk controller,

$$\Delta v_{x} = \left| v_{x,\text{passive}}(T_3) - v_{x,\text{active}}(T_3) \right|,$$  (7.45)
where \( v_{x,\text{passive}}(T_3) \) and \( v_{x,\text{active}}(T_3) \), respectively, are the passive (without anti-jerk control) and active (with anti-jerk control) vehicle speed values at the end of the tip-in test.

- \( \Delta t_{ax} \), which evaluates the vehicle responsiveness to motor torque requests. \( \Delta t_{ax} \) is the time delay between the application of EM torque request variation, and the achievement of a reference longitudinal vehicle acceleration, \( a_{x,\text{ref}} \), arbitrarily defined as 90% of the ideal steady-state vehicle acceleration,

\[
a_{x,\text{ref}} = 0.9 \frac{2 T_{\text{req,tip-in}}}{M_v R_w},
\]  

(7.46)

where \( T_{\text{req,tip-in}} \) is the torque request after the tip-in application, e.g. 50 Nm in the nominal test of this study, and \( \eta \) is the drivetrain efficiency.

- The root-mean-square value of the anti-jerk torque correction, \( RMS_{T_{\text{corr}}} \), which measures the control effort,

\[
RMS_{T_{\text{corr}}} = \sqrt{\frac{1}{T_3 - T_1} \int_{T_1}^{T_3} \left( 2 T_{\text{req}}(t) - 2 T_{\text{plant}}(t) \right)^2 dt}.
\]  

(7.47)

- The root-mean-square value of the steady-state anti-jerk torque correction, \( RMS_{T_{\text{corr}},ss} \), in the last 0.1 s,

\[
RMS_{T_{\text{corr}},ss} = \sqrt{\frac{1}{T_3 - T_2} \int_{T_2}^{T_3} \left( 2 T_{\text{req}}(t) - 2 T_{\text{plant}}(t) \right)^2 dt}.
\]  

(7.48)

\( T_2 = 0.5 \text{ s} \) is chosen such that the controllers reach a steady-state in the tip-in test. \( RMS_{T_{\text{corr}},ss} \) indicates, therefore, the steady-state offset of the anti-jerk controller. The factor 2 in (7.47) and (7.48) is introduced because of the symmetry of the front electric power-trains and tyre-road friction conditions in the simulated scenarios.

The cost function, \( J \), which combines in a weighted sum all the previous performance indicators, is defined as

\[
J = W_c \left( w_1 VDV_{a_x^*} + w_2 RMS_{a_x^*} \right) + W_s \left( w_3 \Delta v_x + w_4 \Delta t_{ax} + w_5 RMS_{T_{\text{corr}}} + w_6 RMS_{T_{\text{corr}},ss} \right),
\]  

(7.49)

where \( w_1-w_6 \) are the weights for the individual performance indicators, while \( W_c \) and \( W_s \) define the relative significance of the comfort indicators (\( VDV_{a_x^*} \) and \( RMS_{a_x^*} \)) and acceleration performance indicators (\( \Delta v_x \), \( \Delta t_{ax} \), \( RMS_{T_{\text{corr}}} \), and \( RMS_{T_{\text{corr}},ss} \)).

\( J \) in (7.49) is used as the cost function for the optimisation-based tuning of the anti-jerk controllers during tip-in tests. An increase of \( W_c \) reduces the longitudinal
acceleration oscillations, at the price of decay in the acceleration performance. An increase of $W_s$ has the opposite effect.

**Tuning routine**

The values of $q_x$ and $r_u$ in the cost function (7.27) have significant influence on the anti-jerk control performance. The controller tuning routine finds appropriate values of the weights $q_x$ and $r_u$, while providing desirable stability properties.

An optimisation using a non-linear programming solver outputs the values of the controller tuning parameters. The optimisation minimises $J$ during a nominal tip-in test with the torque profile as in Fig. 7.5. The nominal test is run with an initial speed of $v_0 = 10\, \text{km/h}$ with the non-linear simulation model. In the first 0.1 s of the manoeuvre, a negative torque of $-2.5\, \text{N m}$ is applied to each front wheel. Then, between 0.1 s and 0.2 s, a positive ramp with a final value $T_{\text{req,tip-in}} = 50\, \text{N m}$ is requested on each front corner, which is the actual tip-in. The change of the input torque sign allows the assessment of the backlash compensation capability of the controller. The simulations are launched with the values of the tuning parameters that are iteratively imposed by the optimisation method. Hence, this is a sequential approach. Initially, optimisations are run for different values of the weights $w_1$-$w_6$ of $J$ in (7.49), with $W_c = W_s = 1$. In this condition, $w_1$-$w_6$ are defined to bring a comfortable vehicle response for all controllers in the nominal tip-in test. Once $w_1$-$w_6$ are fixed, the optimisation-based tuning of the parameters of all controllers is carried out for two set-ups of $W_c$ and $W_s$: i) Comfort mode, with $W_c = W_s = 1$; and ii) Sport mode, with $W_c = 0.5$ and $W_s = 1$. However, in the remainder only results from the comfort mode are presented. The interested reader is referred to Scamarcio et al. (2019a) for more details. The weights $q_x$ and $r_u$ are optimised for each controller varying the number of prediction steps, $N_p$, and the discretisation time, $H$, of the controller, which is chosen to be equal to the sampling time, $T_s$. The time step size is varied in the range $H = 1\, \text{ms}$, 5 ms, and 10 ms, and the prediction steps in the range $N_p = 2$, 4, 6, 8, and 10. Therefore, the control tunings are optimised for 15 different settings of each controller in the tip-in manoeuvre, and a total of 90 anti-jerk controller configurations are evaluated in the analysis.

To assess the stability of a feedback controller it is common practice to employ phase margin and gain margin obtained from the open-loop transfer function of the system, Ogata (2010). For the comparison of different controllers in Scamarcio et al. (2019a), the parameters were constrained to guarantee a certain gain and phase margin, respectively. For an explicit approach to non-linear MPC, it is, however, not straightforward to formulate such constraints for the tuning routine. Therefore, the stability of the model predictive anti-jerk controller was assessed empirically and a-posteriori, through 1000 simulated scenarios adopting a Monte Carlo approach. Three parameters were randomly
selected: i) Initial vehicle speed, $v_{x,0}$, according to a uniform distribution between 2 km/h and 50 km/h; ii) Backlash, according to a Gaussian distribution with mean value of 1.7 deg (referred to the wheels), and standard deviation of 0.28 deg. The mean backlash of that analysis was purposely selected to be significantly higher than its value (0.5 deg) for the nominal vehicle; and iii) Height of the power spectral density of the white noise on the measured EM and wheel speeds according to a Gaussian distribution with mean value $0.5 \times 10^{-8}$ and standard deviation $0.125 \times 10^{-8}$. The controller was considered stable if, between 1.5 s and 2.0 s after the step torque input of the tip-in, the vehicle acceleration was bounded within a ±5% tolerance of its mean value. In the remainder, however, only a sensitivity analysis assessing the robustness of the controller against variations in the initial vehicle speed and torque requests is carried out and a Monte Carlo analysis is omitted.

### 7.3.3 Performance assessment

In this section, all controllers are simulated in nominal conditions in the tip-in test. Employing the performance indicators introduced in Section 7.3.2, the influence of the following aspects on the controller performance are investigated:

1. Level of prediction model complexity, i.e. the varying model fidelity of the prediction models of controllers $C_3$, $C_{3}^{BL}$, $C_4$, $C_4^{BL}$, $C_5$, and $C_5^{BL}$. In particular, the influence of the number of states, including the modelling of wheel and tyre dynamics, and the non-linear backlash model can be analysed.

2. Variation of the controller sampling time, $T_s$, and discretisation time, $H$.

3. Change in the number of prediction steps, $N_p$. Together with 2, the influence of the prediction horizon can be analysed.

The values of the cost function $J$ reflecting the controllers’ performance are reported in Fig. 7.4 for the nominal case. Each bar in the plot refers to a single controller, optimised for different values of the time step size, $H$, and number of prediction steps, $N_p$. For all controllers, the number of control steps is chosen equally ($N_c = 2$). The cases that show persistent control action oscillations are marked by a grey spot, instead of a white bar, and their $J$ value is omitted.

The controllers become unstable for high values of the discretisation time, $H$. In particular the controllers including the wheel (and tyre) dynamics, i.e. controllers $C_4$, $C_4^{BL}$, $C_5$, and $C_5^{BL}$, are affected by the increase of the time step size, $H$. This is because the inclusion of the fast wheel slip dynamics (compared to the vehicle dynamics) in their prediction models makes the system of differential equations stiff, Batra et al. (2018a). All controller formulations achieve their best performance with the shortest
Fig. 7.4: Performance index $J$ calculated for the nominal tip-in test with different controller formulations, number of prediction steps, $N_p$, and time step sizes, $H$. 
discretisation time \((H = 1 \text{ ms})\). Nonetheless, higher values of the sampling time facilitate the implementation on a real vehicle. This requires, therefore, a trade-off between controller performance and practicability.

A similar trend can be observed for the influence of the number of prediction steps, confirming qualitatively the results from Batra et al. (2018a). The performance increases with a decreasing number of prediction steps. Interestingly, the best performance is achieved by all controllers with the smallest number of prediction steps \((N_p = 2)\). This fact poses questions on the real benefit of the MPC prediction.

In Fig. 7.4, all formulations including the non-linear backlash model achieve lower \(J\) values than the corresponding formulations without backlash model. The benefits of backlash modelling are in particular visible for higher sampling times, e.g. the comparison of controllers \(C_3\) and \(C^{BL}_3\) with \(H = 10 \text{ ms}\) brings a reduction of \(J\) up to 13\%, or the comparison of controllers \(C_4\) and \(C^{BL}_4\) with \(H = 5 \text{ ms}\) brings a reduction of \(J\) up to 22\%.

Two examples are analysed in more detail. In Fig. 7.5, a comparison of the torque and acceleration profiles for i) the passive vehicle, ii) the vehicle with controller \(C_3\) \((J = 9.53)\), and iii) the vehicle with controller \(C^{BL}_3\) \((J = 8.83)\) is shown for \(N_p = 2\) and \(H = 10 \text{ ms}\). The plot highlights the benefits brought by the backlash formulation in terms of vehicle responsiveness, especially for higher sampling times, not affecting comfort.

![Fig. 7.5: Backlash influence on controller \(C_3\) in a tip-in test with settings \(N_p = 2\) and \(H = 10 \text{ ms}\).](image)

The second example in Fig. 7.6 shows a smaller overshoot of the vehicle acceleration, \(a_x\), achieved by the controller \(C^{BL}_4\) \((J = 7.53)\) with respect to \(C_4\) \((J = 8.93)\) with settings \(N_p = 2\) and \(H = 1 \text{ ms}\). In both figures, the green lines refer to the passive vehicle, i.e. longitudinal acceleration, \(a_x\), of the passive vehicle and torque requested to the plant, \(T_{\text{plant}}\), being equal to the torque request, \(T_{\text{req}}\).

The introduction of the tyre dynamics with relaxation in the formulation of controllers \(C_5\) and \(C^{BL}_5\) does not bring any performance benefit in terms of \(J\) values, with respect to controllers \(C_4\) and \(C^{BL}_4\), respectively. However, the tyre relaxation formulation
allows stable controller operation also for a high number of prediction steps with a sampling time \( T_s = 1 \text{ ms} \).

Table 7.4: Performance indicators for the best performing controllers in the nominal tip-in test.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Performance indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_p = 2 )</td>
<td>( H = 1 \text{ ms} )</td>
</tr>
<tr>
<td>( VDV_{a^*_x} ) (m/s(^2))</td>
<td>( RMS_{a^*_x} ) (m/s(^2))</td>
</tr>
<tr>
<td>( C_3 )</td>
<td>0.102</td>
</tr>
<tr>
<td>( C_{3BL} )</td>
<td>0.119</td>
</tr>
<tr>
<td>( C_4 )</td>
<td>0.232</td>
</tr>
<tr>
<td>( C_{4BL} )</td>
<td>0.186</td>
</tr>
<tr>
<td>( C_5 )</td>
<td>0.227</td>
</tr>
<tr>
<td>( C_{5BL} )</td>
<td>0.214</td>
</tr>
</tbody>
</table>

In Table 7.4, the values of \( J \) and the individual performance indicators contributing to the cost function are summarised for the best performing controllers (all with \( N_p = 2 \) and \( H = 1 \text{ ms} \)) in the nominal tip-in test. Surprisingly, controllers \( C_3 \) and \( C_{3BL} \) (without wheel dynamics) show the best performance in terms of \( J \). In particular, they achieve low values for the comfort indicators, \( VDV_{a^*_x} \) and \( RMS_{a^*_x} \), and for the steady-state torque reduction, \( RMS_{T_{corr,ss}} \). However, they are characterised by reduced responsiveness, i.e. higher values of \( \Delta t_{a_x} \) and \( \Delta v_x \), and increased control effort, i.e. higher values of \( RMS_{T_{corr}} \), compared to the controllers including wheel (and tyre) dynamics.

7.3.4 Sensitivity analysis

To investigate the robustness of the controllers, a sensitivity analysis varying the initial speed, \( v_{x,0} \), is carried out. Moreover, the performance of the controllers is assessed
with different torque demands, \(T_{\text{req,tip-in}}\) and \(T_{\text{req,tip-out}}\), respectively, during tip-in and tip-out manoeuvres. All controllers are tested with their best performing configuration in nominal conditions, i.e. the setting \(N_p = 2\) and \(H = 1\) ms.

Figure 7.7 (a) shows the results obtained for different initial speeds \(v_{x,0} = 5\) km/h, 10 km/h, 15 km/h, and 20 km/h. For \(v_{x,0} = 5\) km/h, all controllers achieve higher \(J\) values than in the nominal case, i.e. \(v_{x,0} = 10\) km/h. This is due to the fact that low vehicle speeds induce larger drivetrain oscillations. A frequency response analysis in Scamarcio et al. (2019a) shows the influence of the vehicle speed on the drivetrain dynamics, in particular, the equivalent torsional damping between the tyres and the vehicle inertia. For higher initial speeds, \(v_{x,0}\), the controllers including wheel (and tyre) dynamics, i.e. controllers \(C_4\), \(C_{4BL}\), \(C_5\), and \(C_{5BL}\), achieve lower values of \(J\) with respect to the nominal case; whilst controllers \(C_3\) and \(C_{3BL}\), i.e. without modelling of the wheel and tyre dynamics, experience a marginal increase of \(J\). This highlights the limits of the simple prediction models of controllers \(C_3\) and \(C_{3BL}\) and demonstrates the increased robustness of controllers \(C_4\), \(C_{4BL}\), \(C_5\), and \(C_{5BL}\) with respect to vehicle speed being able to model the related change in drivetrain dynamics.

The results for different torque requests during a tip-in manoeuvre, i.e. \(T_{\text{req,tip-in}} = 25\) N\(\text{m}\), \(50\) N\(\text{m}\), and \(75\) N\(\text{m}\), are shown in Fig. 7.7 (b). Higher torque requests naturally lead to higher oscillations in the passive vehicle drivetrain. For this reason also an increase of \(J\) can be observed for all controllers with the increase of \(T_{\text{req,tip-in}}\). Higher values of the performance indices are also due to linear modelling of the tyre forces becoming less accurate for higher torque and, therefore, longitudinal slip values. It is not possible to calculate \(J\) for controller \(C_3\) and \(T_{\text{req,tip-in}} = 75\) N\(\text{m}\), as it does not reach the reference vehicle acceleration, \(a_{x,\text{ref}}\), due to a steady-state offset of the torque demand. The steady-state offset is particularly noticeable for controller \(C_3\) but is common to all controllers in non-nominal conditions, i.e. \(T_{\text{req,tip-in}} \neq 50\) N\(\text{m}\), not including a backlash model in their formulation. Future research could evaluate the effect of an integral action, see Tavernini et al. (2019a), to reduce the steady-state offset, requiring, however, an additional parameter.

For the tip-out manoeuvres, the torque request has the same profile as in the tip-in test until 0.6 s with, however, the controller deactivated. At 0.6 s the controller is activated and the torque demand is decreased from \(T_{\text{req,tip-out}}\), i.e. \(75\) N\(\text{m}\), \(50\) N\(\text{m}\), and \(25\) N\(\text{m}\), to \(-2.5\) N\(\text{m}\) within 0.1 s, and then kept constant. In the tip-out, \(T_1\), \(T_2\), and \(T_3\) are set to 0.6 s, 1.0 s, and 1.1 s. \(a_{x,\text{ref}}\), i.e. the reference acceleration to calculate \(\Delta t_{a_x}\), is set to zero. As the speed at the end of the tip-out test and \(\Delta v_x\) are not relevant for this manoeuvre, \(w_3\) is set to zero in the computation of \(J\). For the tip-out tests, a uniform trend can be observed for the different torque requests, \(T_{\text{req,tip-out}}\), from Fig. 7.7. The cost function value \(J\) decreases with an increase of the model complexity and a decrease of the torque demand, \(T_{\text{req,tip-out}}\). The simple prediction models of controllers \(C_3\) and \(C_{3BL}\), tuned in a
Fig. 7.7: Performance index $J$ calculated for different initial speeds $v_{x,0}$ in (a), and torque requests $T_{\text{req,tip-in}}$ in (b) and $T_{\text{req,tip-out}}$ in (c), respectively.
tip-in test, show an increased steady-state off-set due to model-plant mismatches and, therefore, higher $J$ values. The controllers with higher prediction model complexity, i.e. controllers $C_4$, $C_{4}^{BL}$, $C_5$, and $C_{5}^{BL}$, show better performance in manoeuvres that deviate from the nominal conditions. The best performance during the tip-out tests is achieved by controller $C_{5}^{BL}$.

7.3.5 Explicit solution

Parameter exploration space

For the execution of the Algorithm 1, the hyper-rectangle $X = \{ x_p \in \mathbb{R}^{n_p} \mid A_{th} x_p \leq b_{th} \}$ of the parameter space to be explored needs to be defined. The bounds defining the hyper-rectangle must be chosen carefully to reduce the influence on the dynamic behaviour due to clipping of the parameter vector in case it lies outside the hyper-rectangle $X$ and to keep the complexity of the explicit controller low.

The bounds are chosen to cover the closed-loop response of the best performing controllers of Table 7.4 in a tip-in manoeuvre with $T_{req,tip-in} = 100 \text{ N m}$ and a tip-out manoeuvre with $T_{req,tip-out} = 50 \text{ N m}$ starting from an initial speed $v_{x,0} = 10 \text{ km/h}$. The exploration space corresponds to a range of the vehicle speed from roughly $5 \text{ km/h}$ to $50 \text{ km/h}$. The bounds for the torque request, $T_{req}$, are chosen to satisfy the typical operating region of an anti-jerk controller. With torque values exceeding the defined bounds, the longitudinal slip values would exhibit large values and require the activation of a traction controller for wheel slip control. It is interesting to note that with this definition of the exploration space, the maximum values for the electric motor torque and power, as defined in Table 7.3, are not achieved by the explicit controllers in Table 7.5.

Algorithm and post-processing

Table 7.5 shows the explicit solution of the best performing controllers, as summarised in Table 7.4. The approximation tolerances for the partitioning algorithm 1 are chosen $\tilde{\varepsilon} = 0.02$ for the cost error, $\tilde{\rho} = 0.01$ for the solution error, and $\tilde{\delta} = 0.01$ for the maximum constraint violation. The weighting vector $w_{\rho}$ extracts only the first element associated with the receding horizon control law of the input and the weighting vectors $w_{\delta_i}$ and $w_{\delta_e}$, respectively, are chosen to equally weight all constraint violations. The remaining settings for the partitioning algorithm, i.e. the minimal allowed volume $S_{min}$, the number of internal hyper-rectangles $N_0$, and the specifications for the generation of the set of test points, are chosen equivalent to the settings for the generation of the explicit solution in Chapters 5 and 6. The hyper-rectangles $X$ to be explored are defined in Table 7.5 for the different parameter vectors.
Table 7.5: Partitioning algorithm and post-processing data for the best performing explicit controllers in nominal conditions. The approximation tolerances are $\tilde{\varepsilon} = 0.02$ for the cost error, $\tilde{\varrho} = 0.01$ for the solution error, and $\tilde{\delta} = 0.01$ for the maximum constraint violation.

<table>
<thead>
<tr>
<th>Case</th>
<th>Parameter vector $X_p$</th>
<th>Wheel speed bounds</th>
<th>Torsion rate bounds</th>
<th>Torsion angle bounds</th>
<th>Steady-state slip bounds</th>
<th>Delayed slip bounds</th>
<th>Torque requ. bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_3$</td>
<td>$\dot{\theta}<em>2, \Delta \dot{\theta}, \Delta \theta, T</em>{req}$</td>
<td>$\dot{\theta}_2$ (deg/s)</td>
<td>$\Delta \dot{\theta}$ (deg/s)</td>
<td>$\Delta \theta$ (deg)</td>
<td>$\kappa_x$ (%)</td>
<td>$\tilde{\kappa}_x$ (%)</td>
<td>$T_{req}$ (N m)</td>
</tr>
<tr>
<td>$C_3^{BL}$</td>
<td>$\dot{\theta}<em>2, \Delta \dot{\theta}, \Delta \theta, T</em>{req}$</td>
<td>200</td>
<td>2200</td>
<td>-30</td>
<td>115</td>
<td>-1.2</td>
<td>3.4</td>
</tr>
<tr>
<td>$C_4$</td>
<td>$\dot{\theta}<em>2, \Delta \dot{\theta}, \Delta \theta, \kappa_x, T</em>{req}$</td>
<td>200</td>
<td>2200</td>
<td>-30</td>
<td>115</td>
<td>-1.2</td>
<td>3.4</td>
</tr>
<tr>
<td>$C_4^{BL}$</td>
<td>$\dot{\theta}<em>2, \Delta \dot{\theta}, \Delta \theta, \kappa_x, T</em>{req}$</td>
<td>200</td>
<td>2200</td>
<td>-30</td>
<td>115</td>
<td>-1.2</td>
<td>3.4</td>
</tr>
<tr>
<td>$C_5$</td>
<td>$\dot{\theta}_2, \Delta \dot{\theta}, \Delta \theta, \kappa_x, \kappa'<em>x, T</em>{req}$</td>
<td>200</td>
<td>2200</td>
<td>-30</td>
<td>115</td>
<td>-1.2</td>
<td>3.4</td>
</tr>
<tr>
<td>$C_5^{BL}$</td>
<td>$\dot{\theta}_2, \Delta \dot{\theta}, \Delta \theta, \kappa_x, \kappa'<em>x, T</em>{req}$</td>
<td>200</td>
<td>2200</td>
<td>-30</td>
<td>115</td>
<td>-1.2</td>
<td>3.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Orthogonal partitions</th>
<th>Polyhedral regions</th>
<th>Size top layer (kB)</th>
<th>Size bottom layer (kB)</th>
<th>Algorithm 1 execution time (s)</th>
<th>Index $J$ (nom. tip-in) implicit controller</th>
<th>Index $J$ (nom. tip-in) explicit controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_3$</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>7.51</td>
<td>7.51</td>
</tr>
<tr>
<td>$C_3^{BL}$</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>30</td>
<td>7.38</td>
</tr>
<tr>
<td>$C_4$</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>8.93</td>
<td>8.93</td>
</tr>
<tr>
<td>$C_4^{BL}$</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>68</td>
<td>7.51</td>
</tr>
<tr>
<td>$C_5$</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>8.87</td>
<td>8.87</td>
</tr>
<tr>
<td>$C_5^{BL}$</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>195</td>
<td>8.72</td>
</tr>
</tbody>
</table>
For the explicit controllers in Table 7.5, the maximum estimated error bounds on the cost function and the solution, as well as the maximum constraint violation, fulfil the specified approximation tolerances. Indeed, the actual approximation errors are significantly smaller than the defined tolerances for all controllers. Moreover, the maximum constraint violations are zero due to the fact that the constraints are not active within the exploration space as defined in Table 7.5. For controllers $C_3$, $C_4$, and $C_5$, the explicit solution consists of one single polyhedral region leading to an affine feedback law for the entire hyper-rectangle $X$. It is interesting to observe that the only non-linear contributions in prediction models $M_3$, $M_4$, and $M_5$ (without backlash formulation) stemming from the aerodynamic and rolling resistance do have a negligible influence on the explicit solution. Further investigations for the tip-in test with $T_{\text{req,tip-in}} = 50 \text{ N m}$ up to a speed of $50 \text{ km/h}$ showed an aerodynamic resistance torque, $T_{\text{aero}}$, one order of magnitude and a rolling resistance torque, $T_{\text{roll}}$, two orders of magnitude smaller than the torque request, $T_{\text{plant}}$, to the plant. The non-linear terms for the resistance forces can be, therefore, neglected. The simple explicit solutions give also good results in terms of the performance index $J$ in nominal conditions compared to the implicit solution.

Including the non-linearity of the backlash formulation in the prediction model of controllers $C_{BL}^3$, $C_{BL}^4$, and $C_{BL}^5$ leads to an increase of the complexity of the explicit controller in terms of polyhedral regions with, however, no significant influence. Nevertheless, the inclusion of backlash gives improved performance, as demonstrated in Section 7.3.3, making the explicit controllers $C_{BL}^3$, $C_{BL}^4$, and $C_{BL}^5$, therefore, attractive for a practical implementation due to their simplicity. The comparison of the performance indicators $J$ of the implicit and explicit controllers in nominal conditions also shows the good approximation of the sub-optimal explicit solution. Due to the explicit solution being composed of orthogonal partitions with only one polyhedral region, the complexity reduction and binary search tree generation become obsolete in the post-processing.

To investigate the influence of the constraints (7.28) and (7.29) on the explicit solution, the maximum values of EM torque and power are reduced to 60% of their nominal values, i.e. $T_{EM,\text{max}} = 90 \text{ N m}$ and $P_{EM,\text{max}} = 22.5 \text{ kW}$, respectively. In Table 7.6, the explicit solution of the best performing controllers with reduced constraints are shown. The approximation tolerances for the partitioning algorithm 1 are relaxed and chosen $\tilde{\varepsilon} = 0.2$ for the cost error, $\tilde{\rho} = 0.05$ for the solution error, and $\tilde{\delta} = 0.1$ for the maximum constraint violation. The remaining settings are equivalent to the settings of the explicit controllers presented in Table 7.5. The explicit solutions in Table 7.6 meet the requirements in terms of specified approximation tolerances. It is interesting to observe that in those orthogonal partitions where no constraints are active the estimated solution errors are very small. However, in partitions with the constraints active, the estimated solution errors and maximum constraint violations naturally increase. A comparison with the results in Table 7.5 demonstrates a significant influence of the constraints on the complexity of
Table 7.6: Partitioning algorithm and post-processing data for the best performing explicit controllers with the constraints on electric motor torque and power reduced to 60% of their nominal values. The approximation tolerances are $\bar{\varepsilon} = 0.2$ for the cost error, $\bar{\rho} = 0.05$ for the solution error, and $\bar{\delta} = 0.1$ for the maximum constraint violation.

<table>
<thead>
<tr>
<th>Case</th>
<th>Parameter vector</th>
<th>Wheel speed bounds</th>
<th>Torsion rate bounds</th>
<th>Torsion angle bounds</th>
<th>Steady-state slip bounds</th>
<th>Delayed slip bounds</th>
<th>Torque req. bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_p$</td>
<td>$\hat{\theta}_2$ / $\hat{\theta}_2$ (deg/s)</td>
<td>$\Delta \hat{\theta}$ / $\Delta \theta$ (deg)</td>
<td>$\kappa_x$ / $\bar{\kappa}_x$ (%)</td>
<td>$\kappa'_x$ / $\bar{\kappa}'_x$ (%)</td>
<td>$T_{req}$ / $\bar{T}_{req}$ (Nm)</td>
<td></td>
</tr>
<tr>
<td>$C_3$</td>
<td>$\theta_2, \Delta \hat{\theta}, \Delta \theta, T_{req}$</td>
<td>200 / 2200</td>
<td>-30 / 115</td>
<td>-1.2 / 3.4</td>
<td>- / -</td>
<td>- / -</td>
<td>-5 / 100</td>
</tr>
<tr>
<td>$C_3^{BL}$</td>
<td>$\theta_2, \Delta \hat{\theta}, \Delta \theta, T_{req}$</td>
<td>200 / 2200</td>
<td>-30 / 115</td>
<td>-1.2 / 3.4</td>
<td>- / -</td>
<td>- / -</td>
<td>-5 / 100</td>
</tr>
<tr>
<td>$C_4$</td>
<td>$\theta_2, \Delta \hat{\theta}, \Delta \theta, \kappa_x, T_{req}$</td>
<td>200 / 2200</td>
<td>-30 / 115</td>
<td>-1.2 / 3.4</td>
<td>-4 / 13</td>
<td>- / -</td>
<td>-5 / 100</td>
</tr>
<tr>
<td>$C_4^{BL}$</td>
<td>$\theta_2, \Delta \hat{\theta}, \Delta \theta, \kappa_x, T_{req}$</td>
<td>200 / 2200</td>
<td>-30 / 115</td>
<td>-1.2 / 3.4</td>
<td>-4 / 13</td>
<td>- / -</td>
<td>-5 / 100</td>
</tr>
<tr>
<td>$C_5$</td>
<td>$\theta_2, \Delta \hat{\theta}, \Delta \theta, \kappa_x, \kappa'<em>x, T</em>{req}$</td>
<td>200 / 2200</td>
<td>-30 / 115</td>
<td>-1.2 / 3.4</td>
<td>-4 / 13</td>
<td>- / -</td>
<td>-5 / 100</td>
</tr>
<tr>
<td>$C_5^{BL}$</td>
<td>$\theta_2, \Delta \hat{\theta}, \Delta \theta, \kappa_x, \kappa'<em>x, T</em>{req}$</td>
<td>200 / 2200</td>
<td>-30 / 115</td>
<td>-1.2 / 3.4</td>
<td>-4 / 13</td>
<td>- / -</td>
<td>-5 / 100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Orthogonal partitions</th>
<th>Polyhedral regions</th>
<th>Size top layer</th>
<th>Size bottom layer</th>
<th>Algorithm execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a) / (b) / (c) / (d)</td>
<td>(a) / (b) / (c) / (d)</td>
<td>(kB)</td>
<td>(kB)</td>
<td>(s)</td>
</tr>
<tr>
<td>$C_3$</td>
<td>10</td>
<td>32 / 32 / 20 / 15</td>
<td>2</td>
<td>13 / 7 / 5 / 5</td>
<td>77 / 10 / 9 / 11</td>
</tr>
<tr>
<td>$C_3^{BL}$</td>
<td>8</td>
<td>37 / 37 / 19 / 13</td>
<td>2</td>
<td>15 / 7 / 5 / 5</td>
<td>52 / 7 / 8 / 10</td>
</tr>
<tr>
<td>$C_4$</td>
<td>3</td>
<td>13 / 13 / 6 / 4</td>
<td>2</td>
<td>8 / 4 / 3 / 3</td>
<td>47 / 5 / 7 / 7</td>
</tr>
<tr>
<td>$C_4^{BL}$</td>
<td>22</td>
<td>58 / 58 / 45 / 37</td>
<td>3</td>
<td>26 / 13 / 11 / 10</td>
<td>339 / 11 / 12 / 15</td>
</tr>
<tr>
<td>$C_5$</td>
<td>6</td>
<td>22 / 22 / 16 / 16</td>
<td>2</td>
<td>14 / 6 / 5 / 5</td>
<td>314 / 7 / 32 / 38</td>
</tr>
<tr>
<td>$C_5^{BL}$</td>
<td>34</td>
<td>68 / 68 / 50 / 54</td>
<td>3</td>
<td>37 / 20 / 17 / 21</td>
<td>2566 / 24 / 25 / 30</td>
</tr>
</tbody>
</table>
all controllers in terms of polyhedral regions. Moreover, the inclusion of the non-linear backlash formulation also contributes to increased complexity analogously to the explicit controllers in Table 7.5 with, however, more significant effect as seen for controllers $C_{4}^{BL}$ and $C_{5}^{BL}$. As anticipated, also the number of parameters increases the complexity of the explicit solution. This trend can be observed for controllers $C_{3}^{BL}$, $C_{4}^{BL}$, and $C_{5}^{BL}$. Interestingly, for controllers $C_{3}$, $C_{4}$, and $C_{5}$, the number of polyhedral regions is the highest for controller $C_{3}$ with the lowest number of parameters.

For the post-processing algorithm 4, the settings as defined in Table 5.8 are employed. For case (b), a significant reduction of up to 57% compared to case (a) of the bottom layer file size of all controllers can be observed due to the generation of memory-optimised binary search trees. A further reduction can be achieved by the disjoint optimal and the sub-optimal merging procedures active in case (c). The smaller size of the bottom layer is a result of a smaller number of polyhedral regions that is reduced by up to 54% compared to case (a). Finally, the clipping-based complexity reduction of case (d) does bring a further reduction in the polyhedral regions by up to 69% compared to case (a). The effectiveness of the merging procedures and clipping-based complexity reduction is mainly based on the fact that there is only one control variable. For merging, this increases the likelihood to form convex unions with the same receding horizon feedback law. Summarising, the post-processing has a significant beneficial impact on the complexity of the explicit solution reducing the memory requirements by up to 64% and the number of polyhedral regions by up to 69% with each of the settings giving a contribution.

Fig. 7.8: Orthogonal partitions of the top layer and polyhedral critical regions of the bottom layer of the explicit controller $C_{3}^{BL}$ with reduced constraints and parameter $x_{p,4} = T_{req} / P_{ch,T_{req}}$ fixed to 1 corresponding to a torque request $T_{req} = 100 \text{ N m}$. 
In conclusion, the explicit solution of the anti-jerk controllers is generally quite simple with the backlash model as only non-linear formulation having a visible albeit not significant influence on the complexity of the solution. Moreover, the post-processing algorithm proved to be effective in decreasing the complexity of the explicit solution of the controllers with reduced constraints, which, however, significantly contribute to the number of polyhedral critical regions.

In Fig. 7.8, the orthogonal partitions of the top layer and the polyhedral critical regions of the bottom layer of the explicit controller $C_{3}^{BL}$ with reduced constraints are shown in the normalised wheel speed, torsion rate, and torsion angle parameter space. The parameter $x_{p,4} = T_{req} / P_{ch,T_{req}}$ is fixed to 1 corresponding to a torque request $T_{req} = 100 \text{ Nm}$. For the explicit approach to NMPC, the non-dimensionalisation procedure scaling the parameter vector $X_p = [\dot{\theta}_2, \Delta \dot{\theta}, \Delta \theta, T_{req}]^T$ (and extensions thereof with $\kappa_x$ and $\kappa_x'$, respectively) with the characteristic quantities $X_{ch, \dot{\theta}_2}$, $X_{ch, \Delta \dot{\theta}}$, $X_{ch, \Delta \theta}$, $(X_{ch, \kappa_x}, X_{ch, \kappa'_x})$ and $P_{ch,T_{req}}$ helps to explore a numerically well-conditioned mp-NLP with the Algorithm 1. In Fig. 7.8, the pattern of the polyhedral critical regions reflects the constraints on the maximum EM torque and the maximum EM power.

![Fig. 7.9: Feedback law of the explicit controller $C_{3}^{BL}$ with reduced constraints and parameters $x_{p,3} = \Delta \dot{\theta} / X_{ch,\Delta \theta}$ fixed to 0 corresponding to a torsion angle $\Delta \theta = 0 \text{ deg}$ and $x_{p,4} = T_{req} / P_{ch,T_{req}}$ fixed to 1 corresponding to a torque request $T_{req} = 100 \text{ Nm}$.](image)

In Fig. 7.9, the receding horizon feedback law of the explicit controller $C_{3}^{BL}$ with reduced constraints is shown over the normalised wheel speed and torsion rate parameter space. The parameters $x_{p,3} = \Delta \dot{\theta} / X_{ch,\Delta \theta}$ are fixed to 0 corresponding to a torsion angle $\Delta \theta = 0 \text{ deg}$ and $x_{p,4} = T_{req} / P_{ch,T_{req}}$ fixed to 1 corresponding to a torque request $T_{req} = 100 \text{ Nm}$. The figure shows the dimensionless input $\hat{u}(t_k)$ which needs to be converted to a dimension-related quantity by applying the corresponding transformation.
law of the non-dimensionalisation procedure. Here again, the constraints (7.28) on the EM motor torque (horizontal plane on the top end of Fig. 7.9) and constraints (7.29) on the EM motor power (inclined planes on the left end of Fig. 7.9) can be identified.

7.4 Conclusion and future work

In this chapter, a comparison of six different formulations of a model predictive anti-jerk controller is presented to assess the influence of the prediction model complexity on the control system performance. An objective performance index, $J$, is designed allowing for a unified tuning procedure. The controllers are tested in several tip-in and tip-out tests analysing the performance and robustness of the different formulations.

Conclusion

For the case study vehicle, i.e. a front-wheel-drive electric vehicle with on-board motors, the results show that the inclusion of a non-linear backlash formulation in the prediction model is in all cases beneficial and increases the performance of the controllers. In particular, $J$ is reduced for the simple two-inertia formulation of controller $C_{3}^{BL}$, with higher sampling times, up to 12% with respect to controller $C_{3}$ and for the prediction model including wheel dynamics and a steady-state tyre model of controller $C_{4}^{BL}$ up to 22% with respect to controller $C_{4}$ without drivetrain backlash.

The modelling of the wheel dynamics and the tyre dynamics with constant relaxation length does not bring significant performance improvements in nominal conditions. Quite the contrary, the inclusion of the faster dynamics makes the system of differential equations stiff and provokes persistent control action oscillations for higher values of the sampling time, $T_s$, and discretisation time, $H$. However, the controllers with wheel (and tyre) dynamics are more robust against variations in the initial vehicle speed, $v_{x,0}$, in tip-in tests and different torque requests, $T_{req,tip-out}$, in tip-out tests. Particularly in tip-out tests, the controllers with wheel (and tyre) dynamics show superior performance compared to the controllers $C_{3}$ and $C_{3}^{BL}$ with the simple two-inertia model. On a minor note, the tyre dynamics formulation in controllers $C_{5}$ and $C_{5}^{BL}$ with relaxation allows stable operation also for a high number of prediction steps with time step size $H = 1 \text{ ms}$. However, the performance is not significantly affected compared to controllers $C_{4}$ and $C_{4}^{BL}$.

The controller time step size, i.e. sampling time, $T_s$, and discretisation time, $H$, has a significant impact on the results. In fact, all controllers perform better with low values of the time step size. However, the choice of the sampling time is in general a compromise between controller performance and control hardware requirements for real vehicle implementation. The controllers $C_{3}$ and $C_{3}^{BL}$ with the simple two-inertia model perform better with higher sampling times compared to the remaining controllers and may facilitate, therefore, the implementation on the vehicle.
As also the most advanced prediction model formulation is characterised by mismatch in comparison with the plant, all controllers perform better with a low number of prediction steps, $N_p$, and, therefore, a short prediction horizon, confirming the results from the literature for similar anti-jerk control configurations.

Concluding these results, the recommended formulation for a model predictive anti-jerk controller includes a simple two-inertia prediction model with backlash formulation, as described by $M^{BL}_3$, as it provides the best results during tip-in tests, also outside nominal conditions.

The generation of the explicit solution shows that the non-linear terms for the aerodynamic and rolling resistance forces can be neglected in the controller formulation. The non-linear backlash model, however, has an influence on the complexity of the explicit solution. Moreover, the size of the parameter vector contributes to an increase of the number of polyhedral regions of the generally quite simple explicit solution of the anti-jerk controllers. Also, a significant influence of constraints on the explicit solution can be observed for controllers with modified values of the bounds. For the formulations with reduced constraints, the post-processing algorithm including binary search tree generation, merging procedures, and clipping-based complexity reduction shows to be very effective and beneficial. Aligning with the good performance of controller $C_{3}^{BL}$, the reasonable complexity of the explicit solution makes this controller an attractive choice for practical implementation even on embedded hardware with more stringent specifications in terms of memory requirements.

Summarising the results from Scamarcio et al. (2019a), the performance of feedback controllers can be improved by the addition of the wheel speed signals, as realised in the model predictive anti-jerk controller. They reduce the $VDV_{a_\ast}$ and $RMS_{a_\ast}$ values by one order of magnitude with respect to the motor speed-based feedback controllers. Moreover, the explicit model predictive anti-jerk controller, i.e. controller $C_{3}^{BL}$, shows consistently good performance both in comfort and sport modes implying a wide range of tuneability. The main drawback is the requirement of accurate and reliable input signals related to half-shaft torque, or relative angular displacement between motor and wheels. Moreover, the explicit NMPC requires specialised development and tuning efforts, with respect to the presented more conventional control structures.

**Future work** Future work will focus on practical concerns on the implementation of the explicit model predictive anti-jerk controller on an experimental vehicle demonstrator. This includes the development of robust estimation techniques for the vehicle speed and drive-train torsion considering measurement noise and delays on the sensor signals. A further interesting aspect is the development of an activation and deactivation logic for the anti-jerk controller.
Chapter 8

Conclusion

8.1 Summary and conclusion

Control problems In this thesis, three illustrative automotive control problems were presented for which explicit non-linear model predictive controllers were designed. The main conclusions of each study can be summarised as follows.

The focus of Chapter 5 was on the design and performance assessment of explicit non-linear MPC-based vehicle stability control (VSC) with an electro-hydraulic braking system. The influence of the prediction model fidelity on the performance of explicit non-linear model predictive stability controllers was systematically investigated in simulations. The controlled vehicle remains within the specified thresholds and fulfils the performance requirements of regulation UN/ECE-R 13H (1958) for the Sine with Dwell test with the exception of formulations with linear tyre model, which do not satisfy the responsiveness criterion in the final run. The conclusions of the systematic analysis on the prediction model fidelity are sorted according to their significance in the remainder of this paragraph. A non-linear lateral tyre force model is necessary for acceptable controller behaviour of an MPC-based vehicle stability controller operating at the limits of handling. Furthermore, a load transfer model considering the side-slip angle rate is important for an accurate prediction of the lateral tyre forces and their yaw moment contributions. The modelling of longitudinal and lateral tyre force coupling, with a simple linear model yielding acceptable results, significantly influences the front-to-rear longitudinal tyre force distribution and leads to improved vehicle response with reduced control effort. Non-linear formulations of the stiffness and peak factors in the lateral tyre force model bring only marginal performance benefits, and can, therefore, be omitted to reduce controller complexity. Finally, a simple non-linear lateral tyre force model with saturation based on the hyperbolic tangent gives reasonable results and can be used as an alternative to a more complex non-linear model based on a simplified version
Conclusion

of the Pacejka Magic Formula. Further analysis shows that the additional delays in the NMPC solution have more influence on the control system performance than the controller sampling time.

In Chapter 6, three wheel slip controllers based on explicit non-linear model predictive control were presented. The study in Section 6.1 proposed traction controllers (TC) based on explicit NMPC for electric vehicles with in-wheel motors and compared these with more conventional TC strategies based on proportional-integral control with gain-scheduling and anti-windup features. The non-linear model predictive controller allows a 9.2\% longitudinal slip tracking performance improvement compared to a PI controller during a variable tyre-road friction scenario, simulated with a high fidelity vehicle model. Moreover, the sampling time interval of the traction controller has a more significant impact on the control system performance than the selection of the control system technology. Finally, an explicit non-linear MPC strategy for TC was successfully implemented on a fully electric prototype vehicle.

In Section 6.2, an explicit NMPC-based TC for combined driving and cornering conditions was presented for a front-wheel-drive electric vehicle with in-wheel motors. The prediction model includes a non-linear tyre force model using a simplified version of the Pacejka Magic Formula, incorporating the effects of combined longitudinal and lateral slips. The benefits of the proposed formulation, considering the influence of tyre slip angle, compared to the case of a prediction model considering only longitudinal slip, are demonstrated through simulations. A traction control scenario shows enhanced longitudinal slip tracking performance, which improves by a factor > 5 on vehicle level and by a factor > 13 on wheel level based on the introduced performance indicators.

The study in Section 6.3 presented the proof-of-concept design of an explicit non-linear model predictive controller for anti-lock braking systems (ABS). Moreover, its implementation on an industrial electro-hydraulic braking unit and the experimental comparison with a PID ABS was carried out. Significant variations of the hydraulic pressure dead time and rise time as functions of the final pressure value were identified in experimental tests on the braking system. A dead time compensation strategy was proposed and found necessary to ensure the correct performance of the controller. The explicit non-linear MPC ABS consistently outperforms the PID ABS, e.g. it reduces the stopping distance in low tyre-road friction conditions by up to 11.4\%.

The subject of Chapter 7 was the design and systematic comparison of different formulations for explicit non-linear model predictive anti-jerk control (AJC) of a case-study front-wheel-drive electric vehicle with on-board motors. A performance index was designed to allow for a unified optimisation-based tuning procedure and objective performance assessment of different controllers with varying prediction model complexity in simulations. The results show that the inclusion of a non-linear backlash formulation
in the prediction model, describing the mechanical play in the transmission gears, is in all cases beneficial and improves the behaviour of the controllers. The modelling of wheel dynamics and tyre dynamics with constant relaxation length does not bring significant performance enhancements in nominal conditions. However, the controllers with wheel (and tyre) dynamics are more robust against variations in the initial speed in tip-in tests and different torque requests in tip-out tests. All controllers perform better with low values of the time step size. The recommended prediction model of this study for an NMPC-based AJC is a simple two-inertia drivetrain model with backlash formulation.

Explicit solution Concerning the explicit approximate solution of the different optimal control problems, the following statements (that have to a great extent also validity for explicit MPC in general) can be formulated from the presented studies:

The number of parameters can strongly contribute to the complexity of the explicit NMPC controller in terms of polyhedral critical regions and memory requirements. The choice of the parameters is, therefore, an important decision in the design process of the controller. This can be seen from the vehicle stability controller in Chapter 5 where the 5-parameter controller has significantly increased complexity compared to the formulation with 4 parameters. A similar trend can be observed for the anti-jerk controllers presented in Chapter 7 where the complexity scales with the number of parameters. In the vector of parameters, the initial conditions of the system states are combined with the system and controller parameters that are essential for the optimal control problem formulation. There is only limited freedom to choose the number of states since they are inherent to the specific dynamic system and hence also based on fundamental decisions on which dynamics are modelled and controlled, e.g. the different prediction models with varying complexity in Chapter 7. However, there is some flexibility in choosing the number of varying parameters. The choice typically depends on the number of required reference signals and generally on the design of the optimal control problem, e.g. an integral action (as employed by the controllers of Section 6.1 and 6.3) typically makes the controller more robust but requires an additional parameter. Also for the VSC in Chapter 5, the reference yaw velocity is for this reason formulated based on the remaining parameters - and not as separate input - to reduce the number of parameters. In conclusion, the explicit NMPC with the proposed algorithm can only be applied to systems and control problems with a relatively low number of states and parameters, respectively.

The number of control inputs increases the complexity of the controller. Despite the lack of a direct comparison or study on this influence in the thesis, the number of manipulated variables is certainly one of the contributing factors to the complexity of the vehicle stability controllers in Chapter 5. Moreover, an increasing number of inputs generally reduces the effectiveness of the post-processing algorithm, to be discussed later.
in more detail. Therefore, the general benefit of model predictive control of dealing with multi-variable systems can not be fully exploited in the scope of an explicit approach.

The definition of the parameter exploration space contributes to the complexity of the explicit controller. The number of polyhedral regions and the required memory increases with the size of the parameter space to be explored and should be kept small for this reason. For the vehicle stability controller in Chapter 5 and the wheel slip controller in Section 6.2 the influence of the exploration space is clearly visible by varying the bounds on individual parameters.

The saturation of parameters, based on the definition of the exploration space, can have a significant influence on the closed-loop performance of the explicit controller. The clipping of the parameter vector to the bounds of the specified hyper-rectangle, in case located outside of it, can degrade the performance of the controllers considerably (even if implemented implicitly), as shown for the traction controller of Section 6.2. In a detailed analysis on the vehicle stability controller in Chapter 5, the effect on the dynamic behaviour is minimised by investigating the influence of each parameter and employing asymmetric bounds. In conclusion, the parameter exploration space of the explicit controller needs to be defined cautiously and big enough to reduce performance degradations bearing in mind the aforementioned influence on the complexity.

The exploitation of symmetry properties and the application of state transformations can help to reduce the complexity of the explicit controller. In case the dynamic system exhibits certain symmetry characteristics, their utilisation can save a substantial amount of information, as demonstrated for the vehicle stability controller of Chapter 5. This is, however, not universally applicable and only limited to certain systems. In contrast, the use of state transformations helps to avoid exploring regions in the parameter space that are unlikely to be reached during the closed-loop operation of the controller. Thus, the exploration space is reduced. For all three control problems, i.e. the VSC of Chapter 5, the TC of Section 6.2, and the AJC of Chapter 7, suitable state transformations are designed and employed. Although a direct comparison is missing, a beneficial effect is to be expected from the transformation.

As one of the main motivations of this thesis, even strongly non-linear system dynamics can be easily incorporated into the design of the explicit controller. However, non-linear formulations increase the complexity of the explicit solution. The comparison of different anti-jerk controllers in Chapter 7 shows a higher number of polyhedral regions for formulations including the non-linear backlash model. The same study demonstrates that an explicit approach to NMPC can help to identify the relevance of non-linearities. In that particular case, the non-linear terms describing the aerodynamic and the rolling resistance can be neglected without compromising the control system performance.
The formulation and the number of constraints increase the complexity of the explicit NMPC solution. The number of polyhedral critical regions, describing the set of active constraints for each affine feedback law, will naturally increase. An illustrative example is the anti-jerk control problem of Chapter 7 where modified values of the bounds on the manipulated variable significantly increase the number of regions and subsequently the memory requirements of the explicit controllers.

The number of prediction and control steps increases the complexity of the controller. This effect is related to the aforementioned influence of constraints. Depending on the optimal control problem formulation, a growing number of prediction and/or control steps will typically increase the number of constraints. Although a detailed analysis is outstanding in this thesis, the numbers were chosen relatively small to keep the complexity low. However, the choice of the prediction horizon, i.e. based on the number of prediction steps and the discretisation time, is a trade-off between controller complexity and stability. Longer horizons generally promote stable controller behaviour, just as low numbers of control steps. Increasing numbers of control steps augment the decision variables, which is, however, not as critical as for an implicit approach. Finally, for the direct single shooting approach, the number of prediction steps has a substantial influence on the complexity of the cost and constraint function formulation. This can limit the maximum possible number of steps, e.g. for the TC of Section 6.1 and the ABS of Section 6.3, and hence is also a motivational fact to use the direct multiple shooting approach.

The use of soft constraints in the optimal control problem helps to enlarge the set of feasible parameters. Infeasibility of the optimisation problem is typically reflected in a significant increase in the number of orthogonal partitions in the parameter space exploration. The formulation of soft constraints employing slack variables is used in the vehicle stability controller of Chapter 5 and the traction controller of Section 6.2 to formulate the tracking objective and to allow a certain control error dead-band avoiding an activation logic while preserving feasibility.

With an appropriate choice of the approximation tolerances, reasonable sub-optimality levels of the explicit controller and acceptable performance degradations due to the approximate solution can be achieved. There is a trade-off between controller performance and complexity that can be influenced by the tolerances. The choice of the sub-optimality bounds needs to be done in an iterative approach and reviewed based on simulations of the specific control problems. A direct comparison is presented in Section 6.2 for the TC study and in Chapter 7 for the AJC study showing good agreement of the implicit and the explicit controller performance quantified by objective performance indicators.

The generation of memory-optimised binary search trees, embedded in the post-processing scheme, can significantly reduce the memory requirements of the explicit controllers. This is observable from all three control problems where solely the binary
search tree generation allows for a reduction in the file size of up to 28% for the VSC of Chapter 5, up to 43% for the TC of Section 6.2, and up to 57% for the AJC of Chapter 7. In combination with other post-processing methods, the memory requirements can be even further reduced. The generation of balanced binary search trees may require extensive computational effort offline and still be in the need of significant online memory. This can be seen from the 5-parameter controllers in Chapter 5 which are characterised by a high number of regions per partition. The main motivation for the binary search of reducing the online complexity in terms of function evaluations is, however, not investigated in detail in this thesis. Nevertheless, the evaluation times are empirically assessed for the controllers of Section 6.1 and 6.3.

The use of disjoint optimal and suboptimal merging procedures in the post-processing of the explicit approximate receding horizon feedback law can considerably reduce the number of polyhedral critical regions. Prerequisite for region merging (and the clipping-based method) is the generation of the receding horizon law by manipulation of the explicit approximate solution extracting the information associated with the first sample only. For the presented case studies, the merging can bring a reduction in regions of up to 19% for the vehicle stability controller of Chapter 5, up to 27% for the traction controller of Section 6.2, and up to 54% for the anti-jerk controller of Chapter 7. From Chapter 5 it can be seen that the effectiveness of region merging procedures is generally lowered with a high number of control inputs. In fact, it becomes less likely that neighbouring critical regions, with the same feedback law for each of the inputs, form a convex union.

Clipping-based complexity reduction methods in the post-processing algorithm can bring further reductions in the number of polyhedral critical regions of the receding horizon feedback law. In combination with the merging procedures, the clipping-based method reduces the number of regions by up to 20% for the VSC of Chapter 5, by up to 28% for the TC of Section 6.2, and by up to 69% of the AJC of Chapter 7. Equivalently to the above-mentioned merging, the effectiveness of the clipping-based complexity reduction generally does not scale well with the number of manipulated variables. With an increasing number of control inputs, it becomes less likely that they are jointly saturated at minimum or maximum. The claim from the literature that the clipping-based method scales significantly better with growing complexity than region merging cannot be verified by the case study of Chapter 5. In that study, only one post-processing algorithm fails based on merging as opposed to four algorithms that fail based on clipping combined with merging.

The proposed post-processing algorithm shows good performance and can significantly reduce the complexity of explicit controllers in terms of polyhedral critical regions and memory requirements. The combination of memory-optimised binary search tree generation, disjoint optimal and suboptimal merging procedures, and clipping-based
complexity reduction brings the following results: A reduction in memory requirements of up to 40% for the vehicle stability controller of Chapter 5, up to 52% for the traction controller of Section 6.2, and up to 64% for the anti-jerk controller of Chapter 7 along with a reduction in polyhedral critical regions of up to 20% for the VSC, up to 28% for the TC, and up to 69% for the AJC.

The memory requirements of the explicit controller can vary greatly based on the specific application and may be considerably high. For the vehicle stability controller of Chapter 5, there is a medium to very high demand on the memory of the embedded hardware depending on the formulation in terms of parameters. A feasible real-time implementation can, therefore, only be considered on hardware with suitable memory specifications, e.g., rapid control prototyping units, such as the utilised dSPACE MicroAutoBox II. The complexity of the traction controller in Section 6.2 is moderate with a medium to high memory demand, depending on the number of driven corners and, thus, separate TC modules. Finally for the anti-jerk controller of Chapter 7, the memory demand ranges from low to very low making it, therefore, very attractive for implementation on automotive micro-controller units even with more stringent memory specifications. The conclusions were drawn based on the MATLAB file sizes. However, no detailed analysis was carried out on the correlation between the file size and the RAM (random-access memory) requirements of the compiled real-time executable C-code.

The empirically assessed execution times of the explicit solutions, deployed on an automotive rapid control prototyping unit, are in the sub-millisecond range. Indeed, the computing times are in the ten-microsecond range. For example, the anti-lock braking system of Section 6.3 requires computing times < 95 µs on the 900 MHz dSPACE MicroAutoBox II. The online execution of the traction controller in Section 6.1 is characterised by execution times in the range of 5 µs to 25 µs on the same automotive platform. The proof-of-concept implementation and deployment on a RCP unit of the two wheel slip controllers demonstrate the real-time capability of the proposed explicit NMPC approach. However, no detailed analysis on the online complexity of the explicit controller in terms of CPU time needed to evaluate the expressions was carried out.

A straightforward delay compensation strategy can be easily employed with an explicit NMPC approach. To minimise performance degradations due to transportation delays in the control system, the predicted states at the end of the delay period using the internal prediction model are controlled, instead of the currently measured or estimated states. For the TC in Section 6.1 and the ABS in Section 6.3 such a compensation strategy was designed and found necessary to ensure the correct performance of the controller.

The explicit approach to NMPC facilitates verification of the solution. Based on the explicit approximate solution, its characteristics can be thoroughly investigated and the feasibility reviewed in detail beforehand. For all case studies, i.e. the vehicle
stability controller in Chapter 5, the wheel slip controllers in Chapter 6, and the anti-jerk controller in Chapter 7, visual inspections and numerical checks based on Monte Carlo and brute force methods were performed.

Pros and cons  In conclusion, the findings can be broken down into the following advantages and disadvantages of the explicit NMPC approach. The benefits of explicit non-linear model predictive control include:

- The ability to formally incorporate non-linear system dynamics in the controller design.
- The reduction of online computing times facilitating real-time execution compared to an implicit approach.
- The general benefit of MPC allowing for systematic constraint satisfaction.
- The possibility of a priori verification and validation for safety-critical applications.

However, some drawbacks of the explicit approach to non-linear MPC are:

- The increased memory requirements compared to an implicit approach.
- The suitability and the success of the application depend on the specific system, on design choices, and the number of states and parameters.
- The general drawback of MPC requiring expert knowledge for control system design and parametrisation compared to classical control methods.

Thus, the scope of explicit non-linear model predictive control is on dynamic systems with:

- A low number of system states.
- A minimal number of additional parameters.
- A limited number of control inputs.
- Strong non-linearities.
- Fast dynamics.

Relevant applications of explicit NMPC are:

- Safety-critical.
- Subjected to strict performance requirements.
- Characterised by low computational power but sufficient memory capacity.
8.2 Outlook and future work

This thesis contains comprehensive work on automotive applications of explicit non-linear model predictive control. However, there are still several interesting aspects that can be explored in more detail.

For all presented case studies, estimation techniques were necessary since some of the required physical quantities can not be directly or cost-effectively measured. Further work on the controller and estimator interaction addressing practical concerns and analysing the performance and robustness is necessary.

Although extensively addressed in this thesis, more work can be done on the topic of numerical optimal control in view of explicit NMPC. This could include detailed analyses on the influence of further discretisation and numerical integration methods, and the utilisation of collocation methods.

Moreover, an in-depth study on the run times of the explicit controllers on specific computing platforms using profiling tools together with a general analysis on the online complexity would bring complementary insights.

The development of further applications of explicit NMPC could be in the scope of future research. Relevant topics are autonomous driving where the benefits of eNMPC for safety-critical systems can be fully exploited. To address the capability of coping with fast system dynamics, the explicit NMPC can be employed for power electronics control problems such as electric vehicle traction inverter control or battery emulation.

Finally, an assessment of the industrialisation potential would be very relevant future work. As addressed in the conclusion on the explicit solution, an anti-jerk control system based on explicit NMPC is very attractive for implementation on embedded hardware and is, therefore, a viable candidate for such study.
References


References


References


References


References


