

Technical University of Denmark



An online re-linearization scheme suited for Model Predictive and Linear Quadratic Control

Henriksen, Lars Christian; Poulsen, Niels Kjølstad

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Henriksen, L. C., & Poulsen, N. K. (2010). An online re-linearization scheme suited for Model Predictive and Linear Quadratic Control. (March 4, 2011, Rev. 2. ed.) Kgs. Lyngby: Technical University of Denmark, DTU Informatics, Building 321. (IMM-Technical Report-2010-13).

DTU Library

Technical Information Center of Denmark

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

An online re-linearization scheme suited for Model Predictive and Linear Quadratic Control

L. C. Henriksen¹ and N. K. Poulsen²

¹ Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark, larh@risoe.dtu.dk

² Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark, nkp@imm.dtu.dk

ABSTRACT

This technical note documents the equations for primal-dual interior-point quadratic programming problem solver used for MPC. The algorithm exploits the special structure of the MPC problem and is able to reduce the computational burden such that the computational burden scales with prediction horizon length in a linear way rather than cubic, which would be the case if the structure was not exploited. It is also shown how models used for design of model-based controllers, e.g. linear quadratic and model predictive, can be linearized both at equilibrium and non-equilibrium points, making the presented extension of the controller formulation equivalent to that of the extended Kalman filter compared to an ordinary Kalman filter.

KEYWORDS

linear quadratic control, model predictive control, primal-dual interior-point quadratic programming problem

1 Introduction

State-space control methods based on linear models such as linear quadratic control (LQ), model predictive control (MPC) etc. are typically based on linearizations around equilibrium points, where the control objectives seek to minimize the distance from the current point of operation to the equilibrium point. Gain scheduling frameworks for different operating conditions can be used to handle nonlinearities. The classical gain scheduling framework, consisting of a Jacobian linearization family as described by Rugh and Shamma [1], is comprised of a family of linearizations all done at equilibrium points depending on the chosen scheduling variable. The mentioned control methods all have the same shortcoming, linearizations around equilibrium points are not guaranteed to resemble the actual dynamics at the current operating point. More advanced gain scheduling methods such as linear parameter varying (LPV) methods enable linearizations within a finite parameter space and are not limited to a set of equilibrium points. With an increasing number of scheduling variables and with an increasing discretization of the scheduling variables, the dimensions of the parameter space quickly grows and renders the practical implementation close to impossible, especially if constraints on states, outputs and inputs are to be handled by the controller. The easy fix to avoid the explosion of dimensions in the parameter space is to re-linearize the model online at each sample time and design a controller based on the linearized model, taking non-equilibrium residuals of the linearization into account in the controller design. No theoretical guarantees concerning stability etc. can however be given to this ad-hoc procedure compared to the theoretically better founded LPV framework.

If one or more outputs are to be steered towards their desired reference values several methods exists: The first is to augment the system with an integrator for each reference tracked output and seek to minimize the integrated the error between the output and the reference value. For varying reference values or constraints on e.g. control signals or control signal rates, integrator wind-up can occur, giving rise to unwanted overshoot performance of the closed-loop. A second option is to use target calculation to determine the new steady state target values for states and control signals and use an origin-shifting controller framework, e.g. as proposed by Pannocchia and Rawlings [2].

In this note, it is shown how to gather reference tracking and relinearization at non-stationary points, into a unified formulation, which serves as an extension to the standard Linear Quadratic (LQ) formulation. Model Predictive Control (MPC) with quadratic cost functions is similar to the standard LQ control and the unified control formulation can be applied to MPC as well. Rao et al. [3] has shown how to ease the computational burden of MPC by exploiting the structure given by an inequality constrained optimal control problem. The method suggested by Rao et al. [3] is extended to accommodate the unified formulation presented in this paper. Similar algorithms exploiting the structure of the interior-point optimization problem have subsequently been implemented by others, e.g. Edlund et al. [4] and Haverbeke et al. [5] for MPC and Moving Horizon Estimation purposes, respectively. The extended MPC formulation is also able to act as the quadratic programming problem (QP) solving algorithm used by a sequential quadratic programming solver, when applied to Nonlinear Model Predictive Control (NMPC) as done by Tenny et al. [6].

This documents outline is: In the first section unconstrained MPC and reference-tracking non-stationary LQ control is presented. In the second section it is shown how to exploit the structure of an inequality constrained MPC adapted to the unified control framework when solving the QP. To relate the second section to a quadratic programming problem solver, theory for a general purpose primal-dual interior-point quadratic programming is revisited in the third section. In the last section it is shown how to obtain time discrete dynamic models at non-stationary points.

2 Unconstrained linear model predictive control

The finite horizon optimal control problem is given as

$$\min \phi_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} \phi_k(\mathbf{x}_k, \mathbf{u}_k) \quad (1a)$$

where the stagewise cost function

$$\phi_k(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2} \|\mathbf{r} - \mathbf{g}_r(\mathbf{x}_k, \mathbf{u}_k)\|_{\mathbf{W}_r}^2 + \frac{1}{2} \|\mathbf{g}_z(\mathbf{x}_k, \mathbf{u}_k)\|_{\mathbf{W}_z}^2 \quad (1b)$$

consist of two terms: The first term seeks to drive the plant reference outputs $\mathbf{g}_r(\cdot)$ towards the desired reference \mathbf{r} . The second term seeks to minimize dynamic variations given by $\mathbf{g}_z(\cdot)$ such as e.g. velocities, accelerations. The optimization problem is subject to an initial constraint

$$\mathbf{x}_0 = \bar{\mathbf{x}} \quad (1c)$$

and to the nonlinear state progress equation constraint

$$\mathbf{x}_{k+1} = \underline{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k), \text{ for } k = 0, \dots, N-1 \quad (1d)$$

The nonlinear equations can be assumed linear throughout the entire prediction horizon to ease the problem solving. The state progress equation as well as the equations for the reference outputs $\mathbf{g}_r(\cdot)$ and the dynamic outputs $\mathbf{g}_z(\cdot)$ are linearized around $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$

$$\underline{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) \approx \underline{\mathbf{A}}\mathbf{x}_k + \underline{\mathbf{B}}\mathbf{u}_k + \underline{\boldsymbol{\delta}} \quad (2a)$$

$$\mathbf{g}_r(\mathbf{x}_k, \mathbf{u}_k) \approx \mathbf{C}_r\mathbf{x}_k + \mathbf{D}_r\mathbf{u}_k + \boldsymbol{\gamma}_r \quad (2b)$$

$$\mathbf{g}_z(\mathbf{x}_k, \mathbf{u}_k) \approx \mathbf{C}_z\mathbf{x}_k + \mathbf{D}_z\mathbf{u}_k + \boldsymbol{\gamma}_z \quad (2c)$$

The linearized stagewise cost function can be put on a more general form

$$\phi_k(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}(\mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k + 2\mathbf{x}_k^T \mathbf{M} \mathbf{u}_k + 2\mathbf{q}^T \mathbf{x}_k + 2\mathbf{r}^T \mathbf{u}_k) \quad (3)$$

where

$$\mathbf{Q} = \mathbf{C}_r^T \mathbf{W}_r \mathbf{C}_r + \mathbf{C}_z^T \mathbf{W}_z \mathbf{C}_z \quad (4a)$$

$$\mathbf{R} = \mathbf{D}_r^T \mathbf{W}_r \mathbf{D}_r + \mathbf{D}_z^T \mathbf{W}_z \mathbf{D}_z \quad (4b)$$

$$\mathbf{M} = \mathbf{C}_r^T \mathbf{W}_r \mathbf{D}_r + \mathbf{C}_z^T \mathbf{W}_z \mathbf{D}_z \quad (4c)$$

$$\mathbf{q}^T = [\gamma_r - r]^T \mathbf{W}_r \mathbf{C}_r + \gamma_z^T \mathbf{W}_z \mathbf{C}_z \quad (4d)$$

$$\mathbf{r}^T = [\gamma_r - r]^T \mathbf{W}_r \mathbf{D}_r + \gamma_z^T \mathbf{W}_z \mathbf{D}_z \quad (4e)$$

and the final state cost

$$\phi_N(\mathbf{x}_N) = \frac{1}{2}(\mathbf{x}_N^T \mathbf{\Pi}_N \mathbf{x}_N + 2\boldsymbol{\pi}_N^T \mathbf{x}_N) \quad (5)$$

The Lagrangian to optimal control problem is

$$\mathcal{L} = \phi_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} \phi_k(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\nu}_0^T (\bar{\mathbf{x}} - \mathbf{x}_0) + \sum_{k=0}^{N-1} \boldsymbol{\nu}_{k+1}^T (\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \boldsymbol{\delta} - \mathbf{x}_{k+1}) \quad (6)$$

The optimal solution is given by $\nabla \mathcal{L} = 0$, which can be found by a recursion of the stagewise Lagrangian gradients

$$\nabla_{\mathbf{x}_k} \mathcal{L} = \mathbf{Q}\mathbf{x}_k + \mathbf{M}\mathbf{u}_k + \mathbf{q} + \mathbf{A}^T \boldsymbol{\nu}_{k+1} - \boldsymbol{\nu}_k \quad (7)$$

$$\nabla_{\mathbf{u}_k} \mathcal{L} = \mathbf{M}^T \mathbf{x}_k + \mathbf{R}\mathbf{u}_k + \mathbf{r} + \mathbf{B}^T \boldsymbol{\nu}_{k+1} \quad (8)$$

starting at the end of the prediction horizon

$$\nabla_{\mathbf{x}_N} \mathcal{L} = \mathbf{\Pi}_N + \boldsymbol{\pi}_N - \boldsymbol{\nu}_N \quad (9)$$

inserting the final stage Lagrangian gradient (9) and the state progress equation (2a) into the $N-1$ stage Lagrangian gradient yields

$$\nabla_{\mathbf{x}_{N-1}} \mathcal{L} = [\mathbf{Q}_{N-1} + \mathbf{A}^T \mathbf{\Pi}_N \mathbf{A}] \mathbf{x}_{N-1} + \tilde{\mathbf{M}}_{N-1} \mathbf{u}_{N-1} + \tilde{\mathbf{q}}_{N-1} - \boldsymbol{\nu}_{N-1} \quad (10)$$

$$\nabla_{\mathbf{u}_{N-1}} \mathcal{L} = \tilde{\mathbf{M}}_{N-1}^T \mathbf{x}_{N-1} + \tilde{\mathbf{R}}_{N-1} \mathbf{u}_{N-1} + \tilde{\mathbf{r}}_{N-1} \quad (11)$$

where

$$\tilde{\mathbf{R}}_{N-1} = \mathbf{R} + \mathbf{B}^T \mathbf{\Pi}_N \mathbf{B} \text{ and } \tilde{\mathbf{M}}_{N-1} = \mathbf{M} + \mathbf{A}^T \mathbf{\Pi}_N \mathbf{B} \quad (12)$$

and

$$\tilde{\mathbf{q}}_{N-1} = \mathbf{q} + \mathbf{A}^T \mathbf{\Pi}_N \boldsymbol{\delta} + \mathbf{A}^T \boldsymbol{\pi}_N \text{ and } \tilde{\mathbf{r}}_{N-1} = \mathbf{r} + \mathbf{B}^T \mathbf{\Pi}_N \boldsymbol{\delta} + \mathbf{B}^T \boldsymbol{\pi}_N \quad (13)$$

are introduced to simplify notation. A control law can be derived from (11)

$$\mathbf{u}_{N-1} = -\mathbf{K}_{N-1} \mathbf{x}_{N-1} - \boldsymbol{\kappa}_{N-1} \quad (14)$$

where

$$\mathbf{K}_{N-1} = \tilde{\mathbf{R}}_{N-1}^{-1} \tilde{\mathbf{M}}_{N-1}^T \text{ and } \boldsymbol{\kappa}_{N-1} = \tilde{\mathbf{R}}_{N-1}^{-1} \tilde{\mathbf{r}}_{N-1} \quad (15)$$

which can be inserted into (10) giving

$$\nabla_{\mathbf{x}_{N-1}} \mathcal{L} = \mathbf{\Pi}_{N-1} + \boldsymbol{\pi}_{N-1} - \boldsymbol{\nu}_{N-1} \quad (16)$$

where

$$\mathbf{\Pi}_{N-1} = \mathbf{Q} + \mathbf{A}^T \mathbf{\Pi}_N \mathbf{A} - \tilde{\mathbf{M}}_{N-1} \mathbf{K}_{N-1} \text{ and } \boldsymbol{\pi}_{N-1} = \tilde{\mathbf{q}}_{N-1} - \tilde{\mathbf{M}}_{N-1} \boldsymbol{\kappa}_{N-1} \quad (17)$$

The recursion can then be continued for the stages $N - 2$, $N - 3$ etc. until the beginning of the prediction horizon. The variables can be reconstructed, if needed, with a forward recursion initiated with

$$\bar{\mathbf{x}} = \mathbf{x}_0 \quad (18)$$

$$\boldsymbol{\nu}_0 = \mathbf{\Pi}_0 \mathbf{x}_0 + \boldsymbol{\pi}_0 \quad (19)$$

and then going from $k = 0, \dots, N - 1$ where

$$\mathbf{u}_k = -\mathbf{K}_k \mathbf{x}_k - \boldsymbol{\kappa}_k \quad (20)$$

$$\mathbf{x}_{k+1} = \mathbf{A} \mathbf{x}_k + \mathbf{B} \mathbf{u}_k + \boldsymbol{\delta} \quad (21)$$

$$\boldsymbol{\nu}_{k+1} = \mathbf{\Pi}_{k+1} \mathbf{x}_{k+1} + \boldsymbol{\pi}_{k+1} \quad (22)$$

For a LTI system and $N \rightarrow \infty$ the solution becomes an algebraic set of equations. The quadratic cost for the terminal cost is found by the discrete-time algebraic Riccati equation (DARE)

$$\mathbf{\Pi} = \mathbf{Q} + \mathbf{A}^T \mathbf{\Pi} \mathbf{A} - \tilde{\mathbf{M}} \mathbf{K} \quad (23)$$

where to simplify notation, the matrices

$$\tilde{\mathbf{M}} = \mathbf{M} + \mathbf{A}^T \mathbf{\Pi} \mathbf{B} \quad \text{and} \quad \tilde{\mathbf{R}} = \mathbf{R} + \mathbf{B}^T \mathbf{\Pi} \mathbf{B}$$

are introduced, leading to the feedback gain $\mathbf{K} = \tilde{\mathbf{R}}^{-1} \tilde{\mathbf{M}}^T$. The linear terminal cost term $\boldsymbol{\pi}$ is determined from the same recursion as the DARE

$$\boldsymbol{\pi} = \mathbf{q} - \tilde{\mathbf{M}} \boldsymbol{\kappa} + \mathbf{A}^T \mathbf{\Pi} \boldsymbol{\delta} + \mathbf{A}^T \boldsymbol{\pi} \quad (24)$$

where the control action contribution $\boldsymbol{\kappa}$

$$\boldsymbol{\kappa} = \tilde{\mathbf{R}}^{-1} [\mathbf{r} + \mathbf{B}^T \mathbf{\Pi} \boldsymbol{\delta} + \mathbf{B}^T \boldsymbol{\pi}] \quad (25)$$

is part of the optimal control law

$$\mathbf{u}_k = -\mathbf{K} \mathbf{x}_k - \boldsymbol{\kappa} \quad (26)$$

The DARE should be solved using a specialized DARE solver [7] to achieve fast and robust results and the linear cost term can be found from (24)

$$\boldsymbol{\pi} = [\mathbf{I} - \mathbf{A}^T + \mathbf{K}^T \mathbf{B}^T]^{-1} [\mathbf{q} - \mathbf{K}^T [\mathbf{r} + \mathbf{B}^T \mathbf{\Pi} \boldsymbol{\delta}] + \mathbf{A}^T \mathbf{\Pi} \boldsymbol{\delta}] \quad (27)$$

3 Inequality constrained MPC

In this section, the temporal subscript k has been omitted for the Jacobians \mathbf{A} , \mathbf{B} etc. even though that they can vary within the prediction horizon, if used within a NMPC framework. This is done to ease notation and because they remain constant within the iterations of the QP solver.

The model predictive controller entails the computation of the control signal within a prediction horizon in the range $k = (0, \dots, N - 1)$. The MPC is formulated as a dual mode horizon where the first part, i.e. $k = (0, \dots, N - 1)$, is considered constrained. In the second horizon, i.e. $k = (N + \dots, \infty)$, it is assumed that the plant has reached a state where the unconstrained solution is feasible [8]. The dual mode optimization problem is

$$\min \sum_{k=0}^{N-1} \phi_k(\mathbf{x}_k, \mathbf{u}_k) + \sum_{k=1}^{N-1} \|\boldsymbol{\sigma}_k\|_{\mathbf{W}_\sigma}^2 + \sum_{k=N}^{\infty} \phi_k(\mathbf{x}_k, \mathbf{u}_k) \quad (28a)$$

where the stagewise cost function $\phi_k(\cdot)$ is given by (1b). The term $\|\boldsymbol{\sigma}_k\|_{\mathbf{W}_\sigma}^2$ is only included in the first part of the prediction horizon and seeks to minimize the violation of the soft constraints $\boldsymbol{\sigma}$. The optimization problem is subject to an initial constraint as in (1c) and to the state progress

equation (2a) in the interval ($k = 0, \dots, \infty$). Whereas the soft and hard inequality constraints are only active in the first part of the prediction horizon

$$\mathbf{g}_s(\mathbf{x}_k) - \boldsymbol{\sigma}_k \approx \mathbf{C}_s \mathbf{x}_k + \boldsymbol{\gamma}_s - \boldsymbol{\sigma}_k \leq \mathbf{s}, \quad k = (1, \dots, N-1) \quad (28b)$$

$$\boldsymbol{\sigma}_k \geq 0, \quad k = (1, \dots, N-1) \quad (28c)$$

$$\mathbf{g}_h(\mathbf{x}_k, \mathbf{u}_k) \approx \mathbf{C}_h \mathbf{x}_k + \mathbf{D}_h \mathbf{u}_k + \boldsymbol{\gamma}_h \leq \mathbf{h}, \quad k = (0, \dots, N-1) \quad (28d)$$

The constants contributions and the inequality limits can be combined in a redefined inequality limit to ease notation

$$\bar{\mathbf{s}} \equiv \mathbf{s} - \boldsymbol{\gamma}_s \text{ and } \bar{\mathbf{h}} \equiv \mathbf{h} - \boldsymbol{\gamma}_h$$

The second part of the optimization problem can be reduced to a terminal cost, consisting of a quadratic $\boldsymbol{\Pi}_N$ and a linear term $\boldsymbol{\pi}_N$

$$\phi_N(\mathbf{x}_N) = \sum_{k=N}^{\infty} \phi_k(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_N^T \boldsymbol{\Pi}_N \mathbf{x}_N + \boldsymbol{\pi}_N^T \mathbf{x}_N \quad (29)$$

given by the unconstrained DARE (23) and (27). The Lagrangian for the inequality constrained problem can be written as

$$\begin{aligned} \mathcal{L} = & \phi_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} \phi_k(\mathbf{x}_k, \mathbf{u}_k) + \sum_{k=1}^{N-1} \|\boldsymbol{\sigma}_k\|_{\mathbf{W}_\sigma}^2 \\ & + \boldsymbol{\nu}_0^T (\bar{\mathbf{x}} - \mathbf{x}_0) + \sum_{k=0}^{N-1} (\boldsymbol{\nu}_{k+1}^T (\underline{\mathbf{A}} \mathbf{x}_k + \underline{\mathbf{B}} \mathbf{u}_k + \underline{\boldsymbol{\delta}} - \mathbf{x}_{k+1})) \\ & + \sum_{k=0}^{N-1} \left((\boldsymbol{\lambda}_k^h)^T (\mathbf{C}_h \mathbf{x}_k + \mathbf{D}_h \mathbf{u}_k - \bar{\mathbf{h}}) \right) + \sum_{k=1}^{N-1} \left((\boldsymbol{\lambda}_k^s)^T (\mathbf{C}_s \mathbf{x}_k - \boldsymbol{\sigma}_k - \bar{\mathbf{s}}) \right) + \sum_{k=1}^{N-1} \left((\boldsymbol{\lambda}_k^\sigma)^T (-\boldsymbol{\sigma}_k) \right) \end{aligned} \quad (30)$$

where $\boldsymbol{\nu}_0$ is Lagrange multiplier for (1c); $\boldsymbol{\nu}_k$ for $k = 1, \dots, N$ is the Lagrange multiplier for (2a); $\boldsymbol{\lambda}_k^h, \boldsymbol{\lambda}_k^s$ and $\boldsymbol{\lambda}_k^\sigma$ are the Lagrange multipliers for (28d), (28b) and (28c) respectively.

The Karush-Kuhn-Tucker (KKT) conditions for optimality are

$$\nabla \mathcal{L} = 0 \quad (31)$$

$$\bar{\mathbf{x}} - \mathbf{x}_0 = 0 \quad (32)$$

$$\underline{\mathbf{A}} \mathbf{x}_k + \underline{\mathbf{B}} \mathbf{u}_k + \underline{\boldsymbol{\delta}} - \mathbf{x}_{k+1} = 0 \quad (33)$$

$$\mathbf{C}_h \mathbf{x}_k + \mathbf{D}_h \mathbf{u}_k - \bar{\mathbf{h}} \leq 0 \quad (34)$$

$$\mathbf{C}_s \mathbf{x}_k - \boldsymbol{\sigma}_k - \bar{\mathbf{s}} \leq 0 \quad (35)$$

$$-\boldsymbol{\sigma}_k \leq 0 \quad (36)$$

$$\text{diag}(\boldsymbol{\lambda}_k^h) \text{diag}(\mathbf{C}_h \mathbf{x}_k + \mathbf{D}_h \mathbf{u}_k - \bar{\mathbf{h}}) \mathbf{e} = 0 \quad (37)$$

$$\text{diag}(\boldsymbol{\lambda}_k^s) \text{diag}(\mathbf{C}_s \mathbf{x}_k - \boldsymbol{\sigma}_k - \bar{\mathbf{s}}) \mathbf{e} = 0 \quad (38)$$

$$\text{diag}(\boldsymbol{\lambda}_k^\sigma) \text{diag}(-\boldsymbol{\sigma}_k) \mathbf{e} = 0 \quad (39)$$

$$(\boldsymbol{\lambda}_k^h, \boldsymbol{\lambda}_k^s, \boldsymbol{\lambda}_k^\sigma) \geq 0 \quad (40)$$

where (37), (38) and (39) are the complementary slackness conditions. By the introduction of the slack variables $\mathbf{t}_k^h, \mathbf{t}_k^s$ and \mathbf{t}_k^σ for $\boldsymbol{\lambda}_k^h, \boldsymbol{\lambda}_k^s$ and $\boldsymbol{\lambda}_k^\sigma$ respectively, the KKT condition can be rewritten

to

$$\mathcal{F} = \begin{bmatrix} \nabla \mathcal{L} \\ \bar{x} - x_0 \\ \underline{\mathbf{A}}x_k + \underline{\mathbf{B}}u_k + \underline{\delta} - x_{k+1} \\ \underline{\mathbf{C}}_h x_k + \underline{\mathbf{D}}_h u_k - \bar{h} + t_k^h \\ \underline{\mathbf{C}}_s x_k - \sigma_k - \bar{s} + t_k^s \\ -\sigma_k + t_k^\sigma \\ \mathbf{T}_k^h \Lambda_k^h e \\ \mathbf{T}_k^s \Lambda_k^s e \\ \mathbf{T}_k^\sigma \Lambda_k^\sigma e \end{bmatrix} = 0 \quad (41)$$

$$(\lambda_k^h, t_k^h, \lambda_k^s, t_k^s, \lambda_k^\sigma, t_k^\sigma) \geq 0 \quad (42)$$

where there appropriate ranges of k are omitted for ease of notation. $\mathbf{T}_k^h \equiv \text{diag}(t_k^h)$, $\Lambda_k^h \equiv \text{diag}(\lambda_k^h)$, $\mathbf{T}_k^s \equiv \text{diag}(t_k^s)$, $\Lambda_k^s \equiv \text{diag}(\lambda_k^s)$, $\mathbf{T}_k^\sigma \equiv \text{diag}(t_k^\sigma)$, $\Lambda_k^\sigma \equiv \text{diag}(\lambda_k^\sigma)$ and $e = [1 \ 1 \ \dots \ 1]^T$. The Jacobian of the Lagrangian $\nabla \mathcal{L}$ consist of

$$\nabla_{x_0} \mathcal{L} = \mathbf{Q}x_0 + \mathbf{M}u_0 + q_0 - \nu_0 + \underline{\mathbf{A}}^T \nu_1 + \underline{\mathbf{C}}_h^T \lambda_0^h \quad (43)$$

$$\nabla_{x_k} \mathcal{L} = \mathbf{Q}x_k + \mathbf{M}u_k + q_k - \nu_k + \underline{\mathbf{A}}^T \nu_{k+1} + \underline{\mathbf{C}}_h^T \lambda_k^h + \underline{\mathbf{C}}_s^T \lambda_k^s \quad (44)$$

$$\nabla_{x_N} \mathcal{L} = \mathbf{\Pi}_N x_N + \pi_N - \nu_N \quad (45)$$

$$\nabla_{u_k} \mathcal{L} = \mathbf{R}u_k + \mathbf{M}^T x_k + r_k + \underline{\mathbf{B}}^T \nu_{k+1} + \underline{\mathbf{D}}_h^T \lambda_k^h \quad (46)$$

$$\nabla_{\epsilon_k} \mathcal{L} = \mathbf{Z}\sigma_k + z - \lambda_k^s - \lambda_k^\sigma \quad (47)$$

The Newton like step

$$\nabla \mathcal{F} \Delta w = -\mathcal{F} = r \quad (48)$$

will be used to iterate towards a solution. The full KKT matrix $\nabla \mathcal{F}$ multiplied with the variable step $\Delta w = [\Delta x_0 \ \Delta u_0 \ \dots]$ and the residual vector $r = [r_0^x \ r_0^u \ \dots]$ are

$$\nabla \mathcal{F} \Delta w = \begin{bmatrix} \mathbf{Q}\Delta x_0 + \mathbf{M}\Delta u_0 - \Delta \nu_0 + \underline{\mathbf{A}}^T \Delta \nu_1 + \underline{\mathbf{C}}_h^T \Delta \lambda_0^h \\ \mathbf{Q}\Delta x_k + \mathbf{M}\Delta u_k - \Delta \nu_k + \underline{\mathbf{A}}^T \Delta \nu_{k+1} + \underline{\mathbf{C}}_h^T \Delta \lambda_k^h + \underline{\mathbf{C}}_s^T \Delta \lambda_k^s + q_k \\ \mathbf{\Pi}_N \Delta x_N - \Delta \nu_N \\ \mathbf{R}\Delta u_k + \mathbf{M}^T \Delta x_k + \underline{\mathbf{B}}^T \Delta \nu_{k+1} + \underline{\mathbf{D}}_h^T \Delta \lambda_k^h \\ \mathbf{Z}\Delta \sigma_k - \Delta \lambda_k^s - \Delta \lambda_k^\sigma \\ -\Delta x_0 \\ \underline{\mathbf{A}}\Delta x_k + \underline{\mathbf{B}}\Delta u_k - \Delta x_{k+1} \\ \underline{\mathbf{D}}_h \Delta u_k + \underline{\mathbf{C}}_h \Delta x_k + \Delta t_k^h \\ \underline{\mathbf{C}}_s \Delta x_k - \Delta \sigma_k + \Delta t_k^s \\ -\Delta \sigma_k + \Delta t_k^\sigma \\ \mathbf{T}_k^h \Delta \lambda_k^h + \Lambda_k^h \Delta t_k^h \\ \mathbf{T}_k^s \Delta \lambda_k^s + \Lambda_k^s \Delta t_k^s \\ \mathbf{T}_k^\sigma \Delta \lambda_k^\sigma + \Lambda_k^\sigma \Delta t_k^\sigma \end{bmatrix} \quad (49)$$

$$\begin{bmatrix} r_0^x \\ r_k^x \\ r_N^x \\ r_k^u \\ r_k^\sigma \\ r_0^\nu \\ r_{k+1}^\nu \\ r_k^{\lambda^h} \\ r_k^{\lambda^s} \\ r_k^{\lambda^\sigma} \\ r_k^{t^h} \\ r_k^{t^s} \\ r_k^{t^\sigma} \end{bmatrix} = -\mathcal{F} = \begin{bmatrix} -\mathbf{Q}x_0 - \mathbf{M}u_0 - q_0 + \nu_0 - \underline{\mathbf{A}}^T \nu_1 - \mathbf{C}_h^T \lambda_0^h \\ -\mathbf{Q}x_k - \mathbf{M}u_k - q_k + \nu_k - \underline{\mathbf{A}}^T \nu_{k+1} - \mathbf{C}_h^T \lambda_k^h - \mathbf{C}_s^T \lambda_k^s \\ -\Pi_N x_N - \pi_N + \nu_N \\ -\mathbf{R}u_k - \mathbf{M}^T x_k - r_k - \underline{\mathbf{B}}^T \nu_{k+1} - \mathbf{D}_h^T \lambda_k^h \\ -\mathbf{Z}\sigma_k - z + \lambda_k^s + \lambda_k^\sigma \\ -\bar{x} + x_0 \\ -\underline{\mathbf{A}}x_k - \underline{\mathbf{B}}u_k - \underline{\delta} + x_{k+1} \\ -\mathbf{C}_h x_k - \mathbf{D}_h u_k + \bar{h} - t_k^h \\ -\mathbf{C}_s x_k + \sigma_k + \bar{s} - t_k^s \\ \sigma_k - t_k^\sigma \\ -\mathbf{T}_k^h \Lambda_k^h e - \Omega_h \\ -\mathbf{T}_k^s \Lambda_k^s e - \Omega_s \\ -\mathbf{T}_k^\sigma \Lambda_k^\sigma e - \Omega_\sigma \end{bmatrix} \quad (50)$$

where $\Omega_h = \text{diag}(\Delta t_k^h) \text{diag}(\Delta \lambda_k^h) - \sigma \mu e$, $\Omega_s = \text{diag}(\Delta t_k^s) \text{diag}(\Delta \lambda_k^s) - \sigma \mu e$ and $\Omega_\sigma = \text{diag}(\Delta t_k^\sigma) \text{diag}(\Delta \lambda_k^\sigma) - \sigma \mu e$ are the centering terms for the corrector step in the predictor-corrector algorithm.

The slack variables Δt_k^h , Δt_k^s and Δt_k^σ are eliminated using $(\Delta t_k^h = (\Lambda_k^h)^{-1}(r_k^{t^h} - \mathbf{T}_k^h \Delta \lambda_k^h)$, $(\Delta t_k^s = (\Lambda_k^s)^{-1}(r_k^{t^s} - \mathbf{T}_k^s \Delta \lambda_k^s)$ and $(\Delta t_k^\sigma = (\Lambda_k^\sigma)^{-1}(r_k^{t^\sigma} - \mathbf{T}_k^\sigma \Delta \lambda_k^\sigma)$ respectively, giving

$$\nabla \mathcal{F} \Delta w = \begin{bmatrix} \mathbf{Q} \Delta x_0 + \mathbf{M} \Delta u_0 - \Delta \nu_0 + \underline{\mathbf{A}}^T \Delta \nu_1 + \mathbf{C}_h^T \Delta \lambda_0^h \\ \mathbf{Q} \Delta x_k + \mathbf{M} \Delta u_k - \Delta \nu_k + \underline{\mathbf{A}}^T \Delta \nu_{k+1} + \mathbf{C}_h^T \Delta \lambda_k^h + \mathbf{C}_s^T \Delta \lambda_k^s \\ \Pi_N \Delta x_N - \Delta \nu_N \\ \mathbf{R} \Delta u_k + \mathbf{M}^T \Delta x_k + \underline{\mathbf{B}}^T \Delta \nu_{k+1} + \mathbf{D}_h^T \Delta \lambda_k^h \\ \mathbf{Z} \Delta \sigma_k - \Delta \lambda_k^s - \Delta \lambda_k^\sigma \\ -\Delta x_0 \\ \underline{\mathbf{A}} \Delta x_k + \underline{\mathbf{B}} \Delta u_k - \Delta x_{k+1} \\ \mathbf{D}_h \Delta u_k + \mathbf{C}_h \Delta x_k - \Sigma_k^h \Delta \lambda_k^h \\ \mathbf{C}_s \Delta x_k - \Delta \sigma_k - \Sigma_k^s \Delta \lambda_k^s \\ -\Delta \sigma_k - \Sigma_k^\sigma \Delta \lambda_k^\sigma \end{bmatrix} \quad (51)$$

where $\Sigma_k^h = (\Lambda_k^h)^{-1} \mathbf{T}_k^h$, $\Sigma_k^s = (\Lambda_k^s)^{-1} \mathbf{T}_k^s$ and $\Sigma_k^\sigma = (\Lambda_k^\sigma)^{-1} \mathbf{T}_k^\sigma$

$$\begin{bmatrix} r_0^x \\ r_k^x \\ r_N^x \\ r_k^u \\ r_k^\sigma \\ r_0^\nu \\ r_{k+1}^\nu \\ \bar{r}_k^{\lambda^h} \\ \bar{r}_k^{\lambda^s} \\ \bar{r}_k^{\lambda^\sigma} \end{bmatrix} = \begin{bmatrix} r_0^x \\ r_k^x \\ r_N^x \\ r_k^u \\ r_k^\sigma \\ r_0^\nu \\ r_{k+1}^\nu \\ r_k^{\lambda^h} - (\Lambda_k^h)^{-1} r_k^{t^h} \\ r_k^{\lambda^s} - (\Lambda_k^s)^{-1} r_k^{t^s} \\ r_k^{\lambda^\sigma} - (\Lambda_k^\sigma)^{-1} r_k^{t^\sigma} \end{bmatrix} \quad (52)$$

Eliminating $\Delta \lambda_k^h$, $\Delta \lambda_k^s$ and $\Delta \lambda_k^\sigma$ (using $\Delta \lambda_k^h = (\Sigma_k^h)^{-1}(\mathbf{D}_h \Delta u_k + \mathbf{C}_h \Delta x_k - \bar{r}_k^{\lambda^h})$, $\Delta \lambda_k^s = (\Sigma_k^s)^{-1}(\mathbf{C}_s \Delta x_k - \Delta \sigma_k - \bar{r}_k^{\lambda^s})$ and $\Delta \lambda_k^\sigma = (\Sigma_k^\sigma)^{-1}(-\Delta \sigma_k - \bar{r}_k^{\lambda^\sigma})$), and afterwards eliminating $\Delta \sigma_k$ (using $\Delta \sigma_k = \bar{\mathbf{Z}}^{-1}(\mathbf{C}_s^T (\Sigma_k^s)^{-1} \Delta x_k + \bar{r}_k^\sigma)$ where $\bar{r}_k^\sigma = r_k^\sigma - (\Sigma_k^s)^{-1} \bar{r}_k^{\lambda^s} - (\Sigma_k^\sigma)^{-1} \bar{r}_k^{\lambda^\sigma}$ yields

$$\nabla \mathcal{F} \Delta w = \begin{bmatrix} \bar{\mathbf{Q}}_0 \Delta x_0 + \bar{\mathbf{M}}_0 \Delta u_0 - \Delta \nu_0 + \underline{\mathbf{A}}^T \Delta \nu_1 \\ \bar{\mathbf{Q}}_k \Delta x_k + \bar{\mathbf{M}}_k \Delta u_k - \Delta \nu_k + \underline{\mathbf{A}}^T \Delta \nu_{k+1} \\ \Pi_N \Delta x_N - \Delta \nu_N \\ \bar{\mathbf{R}}_k \Delta u_k + \mathbf{M}_k^T \Delta x_k + \underline{\mathbf{B}}^T \Delta \nu_{k+1} \\ -\Delta x_0 \\ \underline{\mathbf{A}} \Delta x_k + \underline{\mathbf{B}} \Delta u_k - \Delta x_{k+1} \end{bmatrix} \quad (53)$$

and the residuals

$$\begin{bmatrix} \tilde{\mathbf{r}}_0^x \\ \tilde{\mathbf{r}}_k^x \\ \tilde{\mathbf{r}}_N^x \\ \tilde{\mathbf{r}}_k^u \\ \mathbf{r}_0^\nu \\ \mathbf{r}_{k+1}^\nu \end{bmatrix} = \begin{bmatrix} \mathbf{r}_0^x + \mathbf{C}_h^T (\boldsymbol{\Sigma}_0^h)^{-1} \tilde{\mathbf{r}}_0^{\lambda^h} \\ \mathbf{r}_k^x + \mathbf{C}_h^T (\boldsymbol{\Sigma}_k^h)^{-1} \tilde{\mathbf{r}}_k^{\lambda^h} + \mathbf{C}_s^T (\boldsymbol{\Sigma}_k^s)^{-1} \tilde{\mathbf{r}}_k^{\lambda^s} + \mathbf{C}_s^T (\boldsymbol{\Sigma}_k^s)^{-1} \bar{\mathbf{Z}}_k^{-1} \tilde{\mathbf{r}}_k^\sigma \\ \mathbf{r}_N^x \\ \mathbf{r}_k^u + \mathbf{D}_h^T (\boldsymbol{\Sigma}_k^h)^{-1} \tilde{\mathbf{r}}_k^{\lambda^h} \\ \mathbf{r}_0^\nu \\ \mathbf{r}_{k+1}^\nu \end{bmatrix} \quad (54)$$

where

$$\bar{\mathbf{R}}_k = \mathbf{R} + \mathbf{D}_h^T (\boldsymbol{\Sigma}_k^h)^{-1} \mathbf{D}_h \quad (55)$$

$$\bar{\mathbf{M}}_k = \mathbf{M} + \mathbf{C}_h^T (\boldsymbol{\Sigma}_k^h)^{-1} \mathbf{D}_h \quad (56)$$

$$\bar{\mathbf{Z}}_k = \mathbf{Z} + (\boldsymbol{\Sigma}_k^s)^{-1} + (\boldsymbol{\Sigma}_k^\sigma)^{-1} \quad (57)$$

$$\bar{\mathbf{Q}}_0 = \mathbf{Q} + \mathbf{C}_h^T (\boldsymbol{\Sigma}_0^h)^{-1} \mathbf{C}_h \quad (58)$$

where

$$\bar{\mathbf{Q}}_k = \mathbf{Q} - \mathbf{C}_h^T (\boldsymbol{\Sigma}_k^h)^{-1} \mathbf{C}_h + \mathbf{C}_s^T [(\boldsymbol{\Sigma}_k^s)^{-1} - (\boldsymbol{\Sigma}_k^s)^{-1} \bar{\mathbf{Z}}_k^{-1} (\boldsymbol{\Sigma}_k^s)^{-1}] \mathbf{C}_s \quad (59)$$

using

$$\Delta \mathbf{u}_{N-1} = -\mathbf{K}_{N-1} \Delta \mathbf{x}_{N-1} + \boldsymbol{\kappa}_{N-1} \quad (60)$$

where

$$\tilde{\mathbf{R}}_{N-1} = [\bar{\mathbf{R}}_{N-1} + \mathbf{B}^T \boldsymbol{\Pi}_N \mathbf{B}] \quad (61)$$

$$\tilde{\mathbf{M}}_{N-1} = [\bar{\mathbf{M}}_{N-1} + \mathbf{A}^T \boldsymbol{\Pi}_N \mathbf{B}] \quad (62)$$

$$\mathbf{K}_{N-1} = \tilde{\mathbf{R}}_{N-1}^{-1} \tilde{\mathbf{M}}_{N-1}^T \quad (63)$$

$$\boldsymbol{\Pi}_{N-1} = \bar{\mathbf{Q}}_{N-1} + \mathbf{A}^T \boldsymbol{\Pi}_N \mathbf{A} - \tilde{\mathbf{M}}_{N-1} \mathbf{K}_{N-1} \quad (64)$$

$$\boldsymbol{\kappa}_{N-1} = \tilde{\mathbf{R}}_{N-1}^{-1} [\tilde{\mathbf{r}}_{N-1}^u + \mathbf{B}^T \tilde{\mathbf{r}}_N^x + \mathbf{B}^T \boldsymbol{\Pi}_N \mathbf{r}_N^\nu] \quad (65)$$

$$\boldsymbol{\pi}_{N-1} = \tilde{\mathbf{r}}_{N-1}^x + \mathbf{A}^T \boldsymbol{\Pi}_N \mathbf{r}_N^\nu + \mathbf{A} \tilde{\mathbf{r}}_N^x - \tilde{\mathbf{M}}_{N-1} \boldsymbol{\kappa}_{N-1} \quad (66)$$

The recursion can be continued for $k = N - 1, \dots, 0$

$$\tilde{\mathbf{R}}_k = [\bar{\mathbf{R}}_k + \mathbf{B}^T \boldsymbol{\Pi}_{k+1} \mathbf{B}] \quad (67)$$

$$\tilde{\mathbf{M}}_k = [\bar{\mathbf{M}}_k + \mathbf{A}^T \boldsymbol{\Pi}_{k+1} \mathbf{B}] \quad (68)$$

$$\mathbf{K}_k = \tilde{\mathbf{R}}_k^{-1} \tilde{\mathbf{M}}_k^T \quad (69)$$

$$\boldsymbol{\Pi}_k = \bar{\mathbf{Q}}_k + \mathbf{A}^T \boldsymbol{\Pi}_{k+1} \mathbf{A} - \tilde{\mathbf{M}}_k \mathbf{K}_k \quad (70)$$

$$\boldsymbol{\kappa}_k = \tilde{\mathbf{R}}_k^{-1} [\tilde{\mathbf{r}}_k^u + \mathbf{B}^T \tilde{\mathbf{r}}_{k+1}^x + \mathbf{B}^T \boldsymbol{\Pi}_{k+1} \mathbf{r}_{k+1}^\nu] \quad (71)$$

$$\boldsymbol{\pi}_k = \tilde{\mathbf{r}}_k^x + \mathbf{A}^T \boldsymbol{\Pi}_{k+1} \mathbf{r}_{k+1}^\nu + \mathbf{A} \tilde{\mathbf{r}}_{k+1}^x - \tilde{\mathbf{M}}_k \boldsymbol{\kappa}_k \quad (72)$$

A forward recursion can be used to construct variables

$$\Delta \mathbf{x}_0 = -\mathbf{r}_0^\nu \quad (73)$$

$$\Delta \boldsymbol{\nu}_0 = \boldsymbol{\Pi}_0 \Delta \mathbf{x}_0 - \boldsymbol{\pi}_0 \quad (74)$$

For $k = 0, \dots, N - 1$

$$\Delta \mathbf{u}_k = -\mathbf{K}_k \Delta \mathbf{x}_k + \boldsymbol{\kappa}_k \quad (75)$$

$$\Delta \mathbf{x}_{k+1} = \mathbf{A} \Delta \mathbf{x}_k + \mathbf{B} \Delta \mathbf{u}_k - \mathbf{r}_{k+1}^\nu \quad (76)$$

$$\Delta \mathbf{t}_k^h = -\mathbf{D}_h \Delta \mathbf{u}_k - \mathbf{C}_h \Delta \mathbf{x}_k + \mathbf{r}_k^{\lambda^h} \quad (77)$$

$$\Delta \boldsymbol{\lambda}_k^h = (\mathbf{T}_k^h)^{-1} (\mathbf{r}_k^{\lambda^h} - \boldsymbol{\Lambda}_k^h \Delta \mathbf{t}_k^h) \quad (78)$$

For $k = 1, \dots, N - 1$

$$\Delta \boldsymbol{\sigma}_k = \bar{\mathbf{Z}}^{-1} (\mathbf{C}_s^T (\boldsymbol{\Sigma}_k^s)^{-1} \Delta \mathbf{x}_k + \bar{\mathbf{r}}_k^\sigma) \quad (79)$$

$$\Delta \boldsymbol{\nu}_k = \boldsymbol{\Pi}_k \Delta \mathbf{x}_k - \boldsymbol{\pi}_k \quad (80)$$

$$\Delta \mathbf{t}_k^s = -\mathbf{C}_s \Delta \mathbf{x}_k + \Delta \boldsymbol{\sigma}_k + \mathbf{r}_k^{\lambda^s} \quad (81)$$

$$\Delta \mathbf{t}_k^\sigma = \Delta \boldsymbol{\sigma}_k + \mathbf{r}_k^{\lambda^\sigma} \quad (82)$$

$$\Delta \boldsymbol{\lambda}_k^s = (\mathbf{T}_k^s)^{-1} (\mathbf{r}_k^{t^s} - \boldsymbol{\Lambda}_k^s \Delta \mathbf{t}_k^s) \quad (83)$$

$$\Delta \boldsymbol{\lambda}_k^\sigma = (\mathbf{T}_k^\sigma)^{-1} (\mathbf{r}_k^{t^\sigma} - \boldsymbol{\Lambda}_k^\sigma \Delta \mathbf{t}_k^\sigma) \quad (84)$$

4 Linear constrained quadratic optimization

This section describes the underlying quadratic programming problem (QP) solver based on an interior-point primal-dual formulation. The algorithm (Alg. 1) is taken from Rao et al. [3], similar algorithms can be found in Wright [9] and Nocedal and Wright [10]. The mentioned algorithms are all based on the predictor-corrector method developed by Mehrotra [11]. The standard QP is given as

$$\min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \quad (85a)$$

subject to

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (85b)$$

$$\mathbf{C} \mathbf{x} \leq \mathbf{d} \quad (85c)$$

where the Hessian \mathbf{Q} is a symmetric positive semidefinite matrix. The Lagrangian to (85) is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{C} \mathbf{x} - \mathbf{d}) + \boldsymbol{\nu}^T (\mathbf{A} \mathbf{x} - \mathbf{b}) \quad (86)$$

Leading to the Karush-Kuhn-Tucker (KKT) conditions

$$\mathbf{Q} \mathbf{x} + \mathbf{c} + \mathbf{A}^T \boldsymbol{\nu} + \mathbf{C}^T \boldsymbol{\lambda} = 0 \quad (87a)$$

$$\mathbf{A} \mathbf{x} - \mathbf{b} = 0 \quad (87b)$$

$$\mathbf{C} \mathbf{x} - \mathbf{d} \leq 0 \quad (87c)$$

$$\boldsymbol{\lambda} \geq 0 \quad (87d)$$

$$\boldsymbol{\lambda}^T (\mathbf{C} \mathbf{x} - \mathbf{d}) \mathbf{e} = 0 \quad (87e)$$

where (87a) is the stationarity condition, (87b) and (87c) are the primal feasibility conditions, (87d) is the dual feasibility condition and (87e) is the complimentary slackness condition for the primal-dual problem. Introducing the slack variable \mathbf{t} to (85c) simplifies the KKT conditions and gives the following system to solved

$$\mathcal{F}(\mathbf{w}) = \begin{bmatrix} \mathbf{Q} \mathbf{x} + \mathbf{c} + \mathbf{A}^T \boldsymbol{\nu} + \mathbf{C}^T \boldsymbol{\lambda} \\ \mathbf{A} \mathbf{x} - \mathbf{b} \\ \mathbf{C} \mathbf{x} - \mathbf{d} - \mathbf{t} \\ \boldsymbol{\Lambda}^T \mathbf{T} \mathbf{e} \end{bmatrix} = 0, \text{ s.t. } (\boldsymbol{\lambda}, \mathbf{t}) \geq 0 \quad (88)$$

where $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$ and $\mathbf{T} = \text{diag}(\mathbf{t})$, $\mathbf{w} = (\mathbf{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}, \mathbf{t})$. Newton like steps $\nabla \mathcal{F} \Delta \mathbf{w} = -\mathcal{F} = \mathbf{r}$ can be taken to iterate towards the solution. Multiple methods exist but one the most used is the predictor-corrector method by Mehrotra [11]. The method starts with a predictor step

$$\begin{bmatrix} \mathbf{Q} & \mathbf{A}^T & \mathbf{C}^T & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{T} & \boldsymbol{\Lambda} \end{bmatrix} \begin{pmatrix} \Delta \mathbf{x}^{aff} \\ \Delta \boldsymbol{\nu}^{aff} \\ \Delta \boldsymbol{\lambda}^{aff} \\ \Delta \mathbf{t}^{aff} \end{pmatrix} = - \begin{pmatrix} \mathbf{r}^x \\ \mathbf{r}^\nu \\ \mathbf{r}^\lambda \\ \mathbf{r}^t \end{pmatrix} \quad (89)$$

where

$$\mathbf{r}^x = \mathbf{Q}\mathbf{x} + \mathbf{c} - \mathbf{A}^T\boldsymbol{\nu} - \mathbf{C}^T\boldsymbol{\lambda} \quad (90)$$

$$\mathbf{r}^\nu = \mathbf{A}\mathbf{x} - \mathbf{b} \quad (91)$$

$$\mathbf{r}^\lambda = \mathbf{C}\mathbf{x} - \mathbf{t} - \mathbf{d} \quad (92)$$

$$\mathbf{r}^t = \boldsymbol{\Lambda}\mathbf{T}\mathbf{e} \quad (93)$$

The largest possible step size without violating the inequality constraints is calculated

$$\alpha^{aff} = \arg \max\{\alpha \in [0, 1] | (\boldsymbol{\lambda}, \mathbf{s}) + \alpha(\boldsymbol{\Delta}\boldsymbol{\lambda}^{aff}, \boldsymbol{\Delta}\mathbf{s}^{aff}) \geq 0\} \quad (94)$$

The current complementary measure

$$\mu = \boldsymbol{\lambda}^T \mathbf{s} / \dim(\boldsymbol{\lambda}) \quad (95)$$

is a measure of the current feasibility and the affine complementary measure

$$\mu^{aff} = (\boldsymbol{\lambda} + \alpha^{aff} \boldsymbol{\Delta}\boldsymbol{\lambda})^T (\mathbf{s} + \alpha^{aff} \boldsymbol{\Delta}\mathbf{s}) / \dim(\boldsymbol{\lambda}) \quad (96)$$

is a measure of the feasibility if the full predictor step was taken. The predictor step is followed by a corrector step which takes the centering parameter into account

$$\sigma = \left(\frac{\mu^{aff}}{\mu} \right)^3 \quad (97)$$

to keep the iterate at the central path between the primal and the dual problem and thus away from infeasibility. This is done by modifying the residual \mathbf{r}^t with the corrector term $\Omega = \text{diag}(\boldsymbol{\Delta}\mathbf{t}^{aff})\text{diag}(\boldsymbol{\Delta}\boldsymbol{\lambda}^{aff})\mathbf{e} - \sigma\mu\mathbf{e}$

$$\begin{bmatrix} \mathbf{Q} & \mathbf{A}^T & \mathbf{C}^T & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{T} & \boldsymbol{\Lambda} \end{bmatrix} \begin{pmatrix} \boldsymbol{\Delta}\mathbf{x} \\ \boldsymbol{\Delta}\boldsymbol{\nu} \\ \boldsymbol{\Delta}\boldsymbol{\lambda} \\ \boldsymbol{\Delta}\mathbf{t} \end{pmatrix} = - \begin{pmatrix} \mathbf{r}^x \\ \mathbf{r}^\nu \\ \mathbf{r}^\lambda \\ \mathbf{r}^t + \Omega \end{pmatrix} \quad (98)$$

The corrected step size can then be calculated

$$\alpha = \arg \max\{\alpha \in [0, 1] | (\boldsymbol{\lambda}, \mathbf{s}) + \alpha(\boldsymbol{\Delta}\boldsymbol{\lambda}, \boldsymbol{\Delta}\mathbf{s}) \geq 0\} \quad (99)$$

and the variables can be updated

$$\mathbf{w}^+ = \mathbf{w} + \alpha\beta\boldsymbol{\Delta}\mathbf{w}, \quad 0 < \beta < 1 \quad (100)$$

where β is a damping factor typically close to 1, this damping is imposed on the step to improve the numerical stability of the algorithm. Convergence is assumed if

$$\mu \leq \text{tol}_\mu \text{ and } \|\mathbf{r}\|_\infty \leq \text{tol}_r \|(\mathbf{Q}, \mathbf{A}, \mathbf{C}, \mathbf{c}, \mathbf{b}, \mathbf{d})\|_\infty \quad (101)$$

Care should also be taken when determining an initial guess, see e.g. Mehrotra [11] to determine good starting points for $\boldsymbol{\lambda}$ and \mathbf{t} such that they are sufficiently far away from infeasibility but at the same time not too far away from each other as convergence speed would be impaired otherwise.

5 Linearization of models

A dynamic system can be described by the differential state equation and an output equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (102a)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \quad (102b)$$

Algorithm 1: Interior-point primal-dual quadratic programming solver

Initial guess;
if *Constraints are not violated* **then**
 └ Return with result;
for *ITER from 1 to IMAX* **do**
 Solve $\nabla\mathcal{F}\Delta\mathbf{w}^{aff} = -\mathbf{r}$;
 Calc. $\alpha^{aff} = \max\{\alpha \in (0, 1] | (\mathbf{t}, \boldsymbol{\lambda}) + \alpha(\Delta\mathbf{t}^{aff}, \Delta\boldsymbol{\lambda}^{aff}) \geq 0\}$;
 Calc. $\mu = \mathbf{t}^T \boldsymbol{\lambda} / m$;
 Calc. $\mu^{aff} = (\mathbf{t} + \alpha^{aff} \Delta\mathbf{t}^{aff})^T (\boldsymbol{\lambda} + \alpha^{aff} \Delta\boldsymbol{\lambda}^{aff}) / m$;
 Calc. $\sigma = (\mu^{aff} / \mu)^3$;
 Calc. residuals with corrector terms: $\mathbf{r}^{corr} = (\mathbf{r}^x, \mathbf{r}^\nu, \mathbf{r}^\lambda, \mathbf{r}^t + \Omega)$;
 Solve $\nabla\mathcal{F}\Delta\mathbf{w} = -\mathbf{r}^{corr}$;
 Calc. $\alpha = \max\{\alpha \in (0, 1] | (\mathbf{t}, \boldsymbol{\lambda}) + \alpha(\Delta\mathbf{t}, \Delta\boldsymbol{\lambda}) \geq 0\}$;
 Update variables: $\mathbf{w}^+ = \mathbf{w} + \alpha\beta\Delta\mathbf{w}$;
 Calc. residuals used by next predictor step: \mathbf{r} ;
 if *Convergence* **then**
 └ Terminate algorithm;

For the application of model predictive control several methods of time-discretization exists, for nonlinear model predictive control where the model is re-linearized at each temporal point in the prediction horizon, the time-discretization method influences the convergence properties of the NLP solver. Forward Euler time integration is probably the most simple choice. Runge-Kutta time integration schemes are common choices for time discretization of the models. The computation of the sensitivity/Jacobian matrices ($\underline{\mathbf{A}}, \underline{\mathbf{B}}$) of the state progress equation can be computationally expensive. Methods such as the one suggested by Kristensen et al. [12], seeks to minimize the computational burden by reusing already calculated information. Yet another time discretization method, which can be used for NMPC is collocation of time-discrete points [13], which shall not be discussed further in this work.

If the models can be assumed to be linear within a time step of the prediction horizon, the computational burden of time-integration might be reduced significantly. The differential state equation and the measured outputs are linearized around $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ using first order Taylor series approximation

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \approx \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \boldsymbol{\delta} \quad (103a)$$

$$\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) \approx \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \boldsymbol{\gamma} \quad (103b)$$

where the constant contributions $\boldsymbol{\delta}$ and $\boldsymbol{\gamma}$ are

$$\boldsymbol{\delta} = \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) - \mathbf{A}\bar{\mathbf{x}} - \mathbf{B}\bar{\mathbf{u}} \quad (104)$$

$$\boldsymbol{\gamma} = \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) - \mathbf{C}\bar{\mathbf{x}} - \mathbf{D}\bar{\mathbf{u}} \quad (105)$$

The time-discrete state progress equation

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \underbrace{\int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) dt}_{\underline{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k)} \quad (106)$$

can, under the assumption of being linear and having a constant input \mathbf{u} within a time step, be time-discretized by the zero-order-hold method [14] and written as

$$\underline{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) \approx \mathbf{x}_k + \int_{t_k}^{t_{k+1}} \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) dt + \int_{t_k}^{t_{k+1}} \boldsymbol{\delta} dt = \underline{\mathbf{A}}\mathbf{x}_k + \underline{\mathbf{B}}\mathbf{u}_k + \underline{\boldsymbol{\delta}} \quad (107)$$

where

$$\underline{\boldsymbol{\delta}} = \Delta t \boldsymbol{\delta} \quad (108)$$

and where

$$\begin{bmatrix} \underline{\mathbf{A}} & \underline{\mathbf{B}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \text{expm} \left(\Delta t \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right) \quad (109)$$

where **expm** is the matrix exponential function, usually approximated by a Padé approximation with scaling and squaring [15]. Linear forward Euler time integration can be used instead of the zero-order-hold method. The time step Δt can be divided into n even smaller equidistant time steps and a forward recursion can calculate the state values at the next time step

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{\Delta t}{n} \dot{\mathbf{x}}_i = \mathbf{x}_i + \frac{\Delta t}{n} (\mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i + \boldsymbol{\delta}) \quad (110)$$

where i is the local time index between time t_k and t_{k+1} . The time discrete state and input matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ are initiated as

$$\underline{\mathbf{A}} = \mathbf{I} \text{ and } \underline{\mathbf{B}} = \mathbf{0} \quad (111)$$

and updated as

$$\underline{\mathbf{A}} = (\mathbf{I} + \frac{\Delta t}{n} \mathbf{A}) \underline{\mathbf{A}} \text{ and } \underline{\mathbf{B}} = (\mathbf{I} + \frac{\Delta t}{n} \mathbf{A}) \underline{\mathbf{B}} + \frac{\Delta t}{n} \mathbf{B} \quad (112)$$

during the n number of time steps, if the time discrete sensitivity matrices are needed.

Notice that the linearized functions (103) are functions of the original variable and not perturbations around the linearization points. If used for NMPC, the iterations within the sequential programming solver uses the perturbations instead of the actual value and a reformulation is required.

Bibliography

- [1] W.J. Rugh and J.S. Shamma. Research on gain scheduling. *Automatica (UK)*, 36(10):1401–1425, 2000. ISSN 00051098.
- [2] Gabriele Pannocchia and James B. Rawlings. Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2):426–437, 2003. ISSN 00011541.
- [3] C.V. Rao, S.J. Wright, and J.B. Rawlings. Application of interior-point methods to model predictive control. *J. Optim. Theory Appl. (USA)*, 99(3):723–757, 1998. ISSN 00223239.
- [4] K. Edlund, L.E. Sokoler, and J.B. Jrgensen. A primal-dual interior-point linear programming algorithm for mpc. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 351–356, dec. 2009. doi: 10.1109/CDC.2009.5400440.
- [5] N. Haverbeke, M. Diehl, and B. De Moor. A structure exploiting interior-point method for moving horizon estimation. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 1273 – 1278, Dec. 2009.
- [6] Matthew J. Tenny, Stephen J. Wright, and James B. Rawlings. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Computational Optimization and Applications*, 28(1):87–121, 2004. ISSN 15732894.
- [7] III Arnold, W.F. and A.J. Laub. Generalized eigenproblem algorithms and software for algebraic riccati equations. *Proc. IEEE (USA)*, 72(12):1746–1754, 1984. ISSN 00189219.
- [8] P.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *Automatic Control, IEEE Transactions on*, 43(8):1163 –1169, 1998. ISSN 00189286.
- [9] Stephen J. Wright. *Primal-Dual Interior-Point Methods*. Society For Industrial & Applied Mathematics,U.S., 1997.

- [10] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
- [11] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992. doi: 10.1137/0802028. URL <http://link.aip.org/link/?SJE/2/575/1>.
- [12] Morten Rode Kristensen, John Bagterp Jørgensen, Per Grove Thomsen, Morten Rode Kristensen, and Sten Bay Jørgensen. An esdirk method with sensitivity analysis capabilities. *Comput. Chem. Eng.*, 28(12):2695–2707, 2004. ISSN 00981354.
- [13] Lorenz T. Biegler. An overview of simultaneous strategies for dynamic optimization. *Chem. Eng. Process.: Process Intensif.*, 46(11):1043–1053, 2007. ISSN 02552701.
- [14] Gene F. Franklin, J. David Powell, and Michael Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, 2nd edition, 1990.
- [15] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM journal on matrix analysis and applications*, 26(4):1179–1193, 2005. ISSN 10957162.