A Hamiltonian Decomposition for Fast Interior-Point Solvers in Model Predictive Control

Poupard, Eduardo and Heath, William P. and Güttel, Stefan

2020

MIMS EPrint: **2020.6**

ISSN 1749-9097

A Hamiltonian Decomposition for Fast Interior-Point Solvers in Model Predictive Control ⋆

Eduardo Poupard [a], William P. Heath [b], Stefan Güttel [c]

[a] *Corporate Research and Innovation, Process Industries Division, Festo AG & Co.KG, 73770 Denkendorf, Germany*

[b] *Department of Electrical and Electronic Engineering, The University of Manchester, Sackville St. Bldg., Manchester, UK, M13 9PL*

[c] *Department of Mathematics, The University of Manchester, Alan Turing Bldg., Manchester, UK, M13 9PL*

Abstract

Optimal decision-making tools are essential in industry to achieve high performance. One of these tools is Model Predictive Control (MPC), which is an advanced control technique that generates an action that affects the controlled variables, while satisfying the process' operational constraints. At the core of the MPC algorithm lies an optimization problem that is solved by a numerical method at every sample time. New demand for more self-contained modular processes has seen MPC embedded in small-scale platforms. This has prompted a need for custom-made numerical methods that help to efficiently run the computationally demanding optimization algorithms. In this paper, we propose two approaches that factorize the Newton system of the interior-point method (IPM) based on the two-point boundary-value (TPBV) problem structure, rarely explored in MPC. Exploiting the Hamiltonian form of the augmented system, we derive an incomplete LU factorization. A direct method is available to compute the solution of the system using a forward substitution of a series of matrices. An iterative method is also available. We propose a preconditioned Krylov method that converges within a small number of iterations only depending on the number of states.

Key words: Predictive control, Model based control, Optimal control, Iterative methods, Real-time systems.

1 Introduction

The history of MPC dates back to the 1970*s*, especially in the petrochemical industry where MPC was applied to multivariable systems that required careful handling of physical and operational restrictions. Some of the first research published on MPC were Richalet et al. (1977) and Cutler and Ramaker (1979), where MPC was called Model Algorithmic Control (MAC) and Dynamic Matrix Control (DMC), respectively. During the last two decades, some of the major control software vendors have adopted MPC and it can now be found in a wide variety of industries, especially in the process industry (Qin and Badgwell, 2003). There is considerable interest in applying MPC in systems with fast sampling times (Wills et al., 2008) and in small-scale embedded platforms (Jerez et al., 2014).

From a mathematical viewpoint, an MPC controller computes the solution of an optimization problem at every sample time. This can be expressed as a nonlinear system of equations that has to be solved by a numerical method. One of these techniques is the interior point method or IPM (Wright, 1997), which is based on Newton's method. The IPM solves a fixed-size system of equations at every sample time, distinguishing it from the active set method whose structure size can change. It also ensures convergence in polynomial time (Wright, 1997). The replacement of inequality constraints with a barrier function within MPC algorithms has been studied by Wills and Heath (2004). This led to the development of fast optimal solvers, such as the one presented in Wang and Boyd (2010) that applies a Cholesky factorization to the main matrix (as it has a block tridiagonal structure) of the normal equation. Likewise, a fast suboptimal solver based on a barrier function can be found in Feller and Ebenbauer (2018). Iterative methods (Gondzio, 2012) have also been used in interior-point solvers for the MPC problem such as in Shahzad et al. (2012) where the ill-conditioned problem is addressed by splitting the inequality constraints into active and inactive sets.

A different line of research initiated by Kalman (1960) showed that the linear quadratic regulator (LQR) system, in

⋆ This paper was not presented at any IFAC meeting. Corresponding author Eduardo Poupard. Tel. +49(711)347-50335.

Email addresses: eduardo.poupard@festo.com (Eduardo Poupard), william.heath@manchester.ac.uk (William P. Heath), stefan.guettel@manchester.ac.uk (Stefan Güttel).

the Hamiltonian form (two-point boundary-value (TPBV) problem), can be solved by applying the Riccati equation backwards in time. The relation between the MPC structure and the linear-quadratic problem was first highlighted in a number of works by Sznaier and Damborg (1987), Chmielewski and Manousiouthakis (1996) and Scokaert and Rawlings (1998). Rao et al. (1998) show that the Riccati equation can also be applied to factorize the linear system of equations that appear within the algorithm to solve the constrained MPC. Riccati recursion methods have also been used to construct efficient interior-point solvers such as by Frison et al. (2014) and by Nielsen and Axehill (2018) for active-set solvers. The linear-quadratic problem is also the natural structure that appears within the Sequential Quadratic Programming (SQP) algorithm, which is an optimization method widely used for nonlinear MPC (Diehl et al., 2009 and Diehl et al., 2002). An application using SQP based on the Riccati recursion can be found in Katliar et al. (2017).

This paper is part of a broader body of work that analyzes the Hamiltonian form that arises naturally in optimal control formulations and can be exploited to find fast solvers for constrained MPC (Poupard and Heath, 2018 and Poupard et al., 2019). Here we design two preconditioners (or factors) for an incomplete LU factorization of the Newton system in Hamiltonian form. Their implementation allows the solution of the system to be obtained by two methods, a direct substitution of symplectic submatrices and an iterative Krylov method whose convergence is bounded and quantifiable.

In Section 2 the fundamentals of MPC are presented. In Section 3 we illustrate how the standard Riccati recursion that is used in fast MPC solvers can be derived via sparse factorization of the Newton system in Hamiltonian form. In Section 4 we propose an alternative incomplete LU factorization of the Hamiltonian form using two preconditioners that leads to a companion-like matrix; this in turn leads to two methods that are presented in Sections 5 and 6 respectively. In Section 5 we propose a direct method that carries out a forward substitution of symplectic submatrices. In Section 6 we propose an iterative Krylov method based on GMRES and provide bounds for its convergence rate. In Section 7 we present results from numerical experiments which validate the feasibility and convergence of the iterative method in the presence of rounding errors. Finally in Section 8 we illustrate how the method can easily be modified to deal with singular systems.

2 MPC formulation

The fundamental structure of MPC is the two-point boundary-value problem (TPBV) of optimal control theory (Bryson and Ho, 1975, Sect. 2.2), in which the initial condition of the states x_0 is given and the final condition of the Lagrange multiplier of the equality constraints is defined as

$$\lambda_N = \left(\frac{\partial \phi}{\partial x_N}\right), \tag{1}$$

where ϕ is a terminal cost that evaluates the final state x_N at the sample time $k = N$. For convex quadratic optimization problems, where a linear dynamic system and linear constraints are considered, the final condition (1) reduces to $\lambda_N = P x_N$, where P is a weighting matrix for the final state vector.

2.1 Problem formulation

We define the free final state linear MPC problem as

$$\begin{aligned}\min_{x_k \in \mathscr{X}, u_k \in \mathscr{U}} \quad & \frac{1}{2} x_N^\intercal P x_N + \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^\intercal Q x_k + u_k^\intercal R u_k\right)\\ s.t. \quad & x_{k+1} = A x_k + B u_k, \quad k = 0,\dots,N-1,\\ & x_k \in \mathscr{X}, \quad u_k \in \mathscr{U}, \quad k = 0,\dots,N-1,\\ & x_N \in \mathscr{X}_f, \quad x_0 = x(0)\end{aligned} \tag{2}$$

where the linear equality constraints represent the system dynamics, $x_k \in \mathscr{X}$ are the state variables, $u_k \in \mathscr{U}$ are the input variables and N is the prediction horizon. The constraint sets $\mathscr{X} \subseteq \mathbb{R}^{n_x}$ and $\mathscr{U} \subseteq \mathbb{R}^{n_u}$ are polyhedra, whereas $x_N \in \mathscr{X}_f \subseteq \mathbb{R}^{n_x}$ is a terminal polyhedral region. For simplicity, we will consider a time-invariant system. The weighting matrices Q, $P \geq 0$ and $R > 0$ are symmetric. Problem (2) can be expressed as a convex quadratic problem (Boyd and Vandenberghe, 2004, Sect. 4.4)

$$\begin{aligned}\min_{\theta} \quad & f(\theta) = \frac{1}{2}\theta^\intercal \mathscr{H} \theta + g^\intercal \theta\\ s.t. \quad & \mathscr{F}\theta = b\\ & \mathscr{C}\theta \leq d,\end{aligned} \tag{3}$$

where $\theta \in \mathbb{R}^n$ is the decision vector, the Hessian matrix $\mathscr{H} \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $\mathscr{F} \in \mathbb{R}^{m\times n}$ and $\mathscr{C} \in \mathbb{R}^{p\times n}$. The overall aim is to find a decision vector θ which minimizes the quadratic performance index $f(\theta)$, subject to both constraints. A nonlinear system of equations used to solve (3) is

$$\begin{aligned}\mathscr{H}\theta + \mathscr{F}^\intercal \lambda + \mathscr{C}^\intercal \mu + g &= 0,\\ \mathscr{F}\theta - b &= 0,\\ \mathscr{C}\theta - d + t &= 0,\\ \mathscr{M}\mathscr{T} e &= \sigma\eta e,\\ (\mu, t) &\geq 0.\end{aligned} \tag{4}$$

When $\sigma\eta e = 0$, system (4) gives the necessary Karush-Kuhn-Tucker (KKT) conditions for optimality, where λ and μ are the so-called Lagrange multipliers of the equality and inequality constraints respectively and t is a slack variable. Then $\mathscr{M} = \mathrm{diag}(\mu_1, \mu_2, ..., \mu_p)$, $\mathscr{T} = \mathrm{diag}(t_1, t_2, ..., t_p)$ and $e = (1, 1, ..., 1)^\intercal$. Additionally, $\sigma \in [0, 1]$ is the centering parameter and the duality gap is $\eta = \mu^\intercal t / p$. The term $\sigma\eta e$ plays a stabilizing role to allow the algorithm to converge steadily towards the solution of (3) (Wright, 1997, p. 36-40). We will use an infeasible interior-point algorithm (IPM) to solve system (4). Examples of this type of algorithm are: (i)

Mehrotra's Predictor-Corrector algorithm (Mehrotra, 1992) and (ii) the Exact/Inexact Infeasible IPM algorithm presented in Shahzad et al. (2012). For iterative methods, the latter is preferred since the factorization of the search direction system is computed only once. Both algorithms have relatively low computational cost, mainly vector-vector operations. The most computationally demanding part is the augmented Newton system, which requires the solution of a linear system of equations $J(\theta,\lambda,\mu,t)\Delta p = -F(\theta,\lambda,\mu,t)$, defined as

$$\begin{bmatrix} \bar{\mathscr{H}} & \mathscr{F}^\mathsf{T} \\ \mathscr{F} & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta\lambda \end{bmatrix} = -\begin{bmatrix} \bar{r}_\theta \\ r_\lambda \end{bmatrix}, \tag{5}$$

where $\Delta\theta = [\Delta x^\mathsf{T}, \Delta u^\mathsf{T}]^\mathsf{T}$ and $\bar{\mathscr{H}} = \mathscr{H} + \mathscr{C}^\mathsf{T}\left(\mathscr{M}^{-1}\mathscr{T}\right)^{-1}\mathscr{C}$, and the residuals are $r_\theta = \left[r_x^\mathsf{T}, r_u^\mathsf{T}\right]^\mathsf{T} = \mathscr{H}\theta + \mathscr{F}^\mathsf{T}\lambda + \mathscr{C}^\mathsf{T}\mu + g$, $\bar{r}_\theta = r_\theta + \mathscr{C}^\mathsf{T}\left(\mathscr{M}^{-1}\mathscr{T}\right)^{-1}\left(r_\mu - \mathscr{M}^{-1}r_t\right)$, $r_\lambda = \mathscr{F}\theta - b$, $r_\mu = \mathscr{C}\theta - d + t$ and $r_t = \mathscr{M}\mathscr{T}e - \sigma\eta e$. The final elements of Δp can be computed as

$$\begin{aligned} \Delta\mu &= \left(\mathscr{M}^{-1}\mathscr{T}\right)^{-1}\left(\mathscr{C}\Delta\theta + r_\mu - \mathscr{M}^{-1}r_t\right), \\ \Delta t &= -\mathscr{M}^{-1}\left(\mathscr{T}\Delta\mu + r_t\right), \end{aligned}$$

where

$$\mathscr{C} = \begin{bmatrix} \omega & \cdots & 0 & 0 & \phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \omega & 0 & 0 & \cdots & \phi \\ 0 & \cdots & 0 & \omega & 0 & \cdots & 0 \end{bmatrix},$$

and

$$\omega = \begin{bmatrix} -I_{n_x} \\ I_{n_x} \end{bmatrix}, \quad \phi = \begin{bmatrix} -I_{n_u} \\ I_{n_u} \end{bmatrix}. \tag{6}$$

Here $I_{n_x} \in \mathbb{R}^{n_x \times n_x}$ and $I_{n_u} \in \mathbb{R}^{n_u \times n_u}$ are identity matrices. For the MPC problem (2), the augmented system (5) is in the form of the two-point boundary-value problem as we will demonstrate in the next section.

3 Hamiltonian system decomposition

The system (5) can be further reduced by eliminating the elements Δu, i.e.

$$\begin{bmatrix} \bar{\mathscr{H}}_Q & \mathscr{F}_A^\mathsf{T} \\ \mathscr{F}_A & \mathscr{G} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = -\begin{bmatrix} \bar{r}_x \\ \bar{r}_\lambda \end{bmatrix}, \tag{7}$$

where $\bar{\mathscr{H}}_Q$ and $\mathscr{G}$ are block diagonal matrices defined as

$$\begin{aligned} \bar{\mathscr{H}}_Q &= \operatorname{diag}\left(Q_k, \ldots, Q_{N-1}, P_N\right), \\ \mathscr{G} &= \operatorname{diag}\left(0, -G_k, \ldots, -G_{N-1}\right), \end{aligned}$$

where $G_i = BR_i^{-1}B^\mathsf{T}$ and

$$\mathscr{F}_A = \begin{bmatrix} -I & & & \\ A & \ddots & & \\ & \ddots & -I & \\ & & A & -I \end{bmatrix}.$$

The elements Δu_i are

$$\Delta u_i = -R_i^{-1}\left(B^\mathsf{T}\Delta\lambda_{i+1} + \bar{r}_{u_i}\right),$$

and the residual is $\bar{r}_{\lambda_{i+1}} = -r_{\lambda_{i+1}} + BR_i^{-1}\bar{r}_{u_i}$. System (7) can be rearranged as

$$D_{mpc} z_{mpc} = b_{mpc} \tag{8}$$

where

$$D_{mpc} = \begin{bmatrix} -I & & & & & & & & \\ -A & 0 & I & G_k & & & & & \\ Q_k & -I & 0 & A^\mathsf{T} & \ddots & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & -A & 0 & I & G_{N-1} & \\ & & & & Q_{N-1} & -I & 0 & A^\mathsf{T} & \\ & & & & & & P_N & -I & \end{bmatrix},$$

$$\begin{aligned} z_{mpc} &= \left[\Delta x_k^\mathsf{T} \; \Delta\lambda_k^\mathsf{T} \; \Delta x_{k+1}^\mathsf{T} \; \Delta\lambda_{k+1}^\mathsf{T} \; \ldots \; \ldots \; \Delta x_N^\mathsf{T} \; \Delta\lambda_N^\mathsf{T}\right]^\mathsf{T}, \\ b_{mpc} &= -\left[r_{\lambda_k}^\mathsf{T} \; \bar{r}_{\lambda_{k+1}}^\mathsf{T} \; \bar{r}_{x_k}^\mathsf{T} \; \bar{r}_{\lambda_{k+2}}^\mathsf{T} \; \ldots \; \ldots \; \bar{r}_{x_{N-1}}^\mathsf{T} \; \bar{r}_{x_N}^\mathsf{T}\right]^\mathsf{T}. \end{aligned}$$

The matrix D_{mpc} has dimension $2(N+1)n_x \times 2(N+1)n_x$ and the vectors z_{mpc} and b_{mpc} are of dimension $2(N+1)n_x$. System (8) is similar in form to the Linear Quadratic Regulator (LQR) system or the LQ tracking problem (Poupard and Heath, 2018). The sole difference between these systems is the right-hand side vector b_{mpc}, which in this case represents the residuals of the Newton system of the interior-point method. It is easy to recognize the subsystems from iterations k to $N-1$ in the matrix D_{mpc}, i.e.

$$\begin{bmatrix} -A & 0 \\ Q_k & -I \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta\lambda_k \end{bmatrix} + \begin{bmatrix} I & G_k \\ 0 & A^\mathsf{T} \end{bmatrix} \begin{bmatrix} \Delta x_{k+1} \\ \Delta\lambda_{k+1} \end{bmatrix} = -\begin{bmatrix} \bar{r}_{\lambda_{k+1}} \\ \bar{r}_{x_k} \end{bmatrix}. \tag{9}$$

This is the Hamiltonian system of the two-point boundary-value problem with the boundary conditions $\Delta x_0 = r_{\lambda_0}$ and

$$\Delta\lambda_N = P_N \Delta x_N + \bar{r}_{x_N}. \tag{10}$$

It is standard to factorize system (5) using the Riccati recursion method as it was shown in Rao et al. (1998). In Poupard et al. (2019), the Riccati equation (and the vector sequence computation) can also be derived from system (8). One of the advantages of this approach is that the complexity of the algorithm is reduced to $\mathcal{O}(Nn_x^3)$, as opposed to the dense approach that is $\mathcal{O}\left(N^3 n_x^3\right)$. Therefore, this method is attractive for state-of-the-art optimization solvers such as the one presented in Frison et al. (2014). In next section, we propose an alternative method that decomposes the main matrix into an incomplete LU factorization, which has recently proposed for MPC (Poupard, 2018).

4 Incomplete LU factorization

In this section, we will present a method that takes advantage of the special structure of the Hamiltonian system (9). This approach consists of the formulation of two preconditioners (left and right), which resemble the two-point boundary-value problem structure. Multiplying them onto system (8) gives rise to a companion-like matrix, whose last column is computed as a forward recursive sequence. This sequence is in fact the factorization that allows to solve the linear system directly. Furthermore, the application of the two preconditioners allows to solve the system by an iterative method like GMRES. An application of the GMRES method to non-linear MPC can be found in Ohtsuka (2004). To derive the preconditioners, system (8) is first slightly rearranged as

$$D_{hqr} z_{hqr} = b_{hqr} \tag{11}$$

where

$$D_{hqr} = \begin{bmatrix} I & G_k & & & & & & & 0 \\ 0 & A^\mathsf{T} & & & & & & & -I \\ -A & 0 & I & G_{k+1} & & & & & 0 \\ Q_{k+1} & -I & 0 & A^\mathsf{T} & \ddots & & & & 0 \\ & & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ & & & \ddots & -A & 0 & I & G_{N-1} & 0 \\ & & & & Q_{N-1} & -I & 0 & A^\mathsf{T} & 0 \\ & & & & & & P_N & -I & 0 \end{bmatrix},$$

$$z_{hqr} = \left[\Delta x_{k+1}^\mathsf{T} \ \Delta\lambda_{k+1}^\mathsf{T} \ \Delta x_{k+2}^\mathsf{T} \ \Delta\lambda_{k+2}^\mathsf{T} \ \dots \ \Delta x_N^\mathsf{T} \ \Delta\lambda_N^\mathsf{T} \ \Delta\lambda_k^\mathsf{T}\right]^\mathsf{T},$$
$$b_{hqr} = -\left[\bar{r}_{\lambda_{k+1}}^\mathsf{T} \ \bar{r}_{x_k}^\mathsf{T} \ \bar{r}_{\lambda_{k+2}}^\mathsf{T} \ \bar{r}_{x_{k+1}}^\mathsf{T} \ \dots \ \bar{r}_{\lambda_{N-1}}^\mathsf{T} \ \bar{r}_{x_{N-1}}^\mathsf{T} \ \bar{r}_{x_N}^\mathsf{T}\right]^\mathsf{T}.$$

The matrix D_{hqr} has dimension $(2N+1)n_x \times (2N+1)n_x$, and the vectors z_{hqr} and b_{hqr} are of dimension $(2N+1)n_x$. System (11) is equivalent to (8), differing only in size for two reasons: (i) Δx_k is eliminated since it is known that $\Delta x_k = r_{\lambda_k} = 0$; (ii) the column corresponding to $\Delta\lambda_k$ is swapped to the last position.

4.1 Tailor-made preconditioners

Let $H_{hqr} = H_Q H_R$ be nonsingular and apply it as a preconditioner for the system (11), namely

$$\tilde{D}_{hqr} \tilde{z}_{hqr} = \tilde{b}_{hqr}, \tag{12}$$

where

$$\tilde{D}_{hqr} = H_Q^{-1} D_{hqr} H_R^{-1}, \quad \tilde{b}_{hqr} = H_Q^{-1} b_{hqr}.$$

The matrices H_Q and H_R are called the left and right preconditioners, respectively. The solution of the original system (11) is obtained by solving $H_R z_{hqr} = \tilde{z}_{hqr}$. We design the preconditioners so that when (12) is solved iteratively with a Krylov method, the convergence to the true solution is fast. In order to achieve this, the preconditioner H_{hqr} should meet the following criteria (Golub and Loan, 2013)

(1) The structure of matrix H_{hqr} should resemble the original matrix D_{hqr} in some sense, e.g., $H_{hqr} \approx D_{hqr}$. Hence, $H_Q^{-1} D_{hqr} H_R^{-1}$ should be somehow close to the identity matrix I, $\tilde{D}_{hqr} \approx I$.
(2) Likewise, the matrices H_Q and H_R should be easy to factor (e.g., block banded), and their application dominated by matrix-vector operations.

We will demonstrate that our preconditioners fulfil the above criteria with some additional advantages:

a) The matrix $\tilde{D}_{hqr}$ is a companion-like matrix, which emulates the identity matrix in all its columns but the last one.
b) The GMRES method converges to the exact solution in a small number of $n_x + 1$ steps (it essentially behaves like a direct solver).
c) The preconditioners H_Q and H_R are triangular matrices, which makes their factorization trivial.
d) A matrix-free algorithm will be proposed that avoids building the matrices explicitly.

We now state our proposed preconditioners. Let the right preconditioner H_R be the upper block-triangular matrix

$$H_R = \begin{bmatrix} I & G_k & & & & & & & \\ & I & \ddots & & & & & & \\ & & \ddots & \ddots & & & & & \\ & & & \ddots & \ddots & & & & \\ & & & & I & G_{N-2} & & & \\ & & & & & I & & & \\ & & & & & & I & G_{N-1} & \\ & & & & & & & I & \\ & & & & & & & & I \end{bmatrix}, \tag{13}$$

and the left preconditioner H_Q be the lower block-triangular matrix

$$H_Q = \begin{bmatrix} I & & & & & & & & \\ 0 & A^{\mathsf{T}} & & & & & & & \\ -A & AG_k & I & & & & & & \\ Q_{k+1} & F_{k+1} & 0 & A^{\mathsf{T}} & & & & & \\ & \ddots & \ddots & \ddots & \ddots & & & & \\ & & \ddots & \ddots & \ddots & \ddots & & & \\ & & & \ddots & -A & AG_{N-2} & I & & \\ & & & & Q_{N-1} & F_{N-1} & 0 & A^{\mathsf{T}} & \\ & & & & & & P_N & F_N & I \end{bmatrix}, \quad (14)$$

where $F_{k+1} = -(Q_{k+1}G_k + I)$ for $k = 0,\ldots,N-1$ and $F_N = -(P_N G_{N-1} + I)$ for $k = N$. For H_Q to be nonsingular, we clearly require that the matrix A is nonsingular. To show that the matrix $H_{hqr} = H_Q H_R$ has the same structure as D_{hqr}, we solve the subsystems $H_{Q_{(1:2n_x,1:2n_x)}} H_{R_{(1:2n_x,1:2n_x)}} = D_{hqr_{(1:2n_x,1:2n_x)}}$ and $H_{Q_{(3n_x:4n_x,1:2n_x)}} H_{R_{(1:2n_x,1:2n_x)}} = D_{hqr_{(3n_x:4n_x,1:2n_x)}}$, i.e.

$$H_{Q_{(1:2n_x,1:2n_x)}} = \begin{bmatrix} I & G_k \\ 0 & A^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} I & -G_k \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & A^{\mathsf{T}} \end{bmatrix},$$

and

$$H_{Q_{(3n_x:4n_x,1:2n_x)}} = \begin{bmatrix} -A & 0 \\ Q_{k+1} & -I \end{bmatrix} \begin{bmatrix} I & -G_k \\ 0 & I \end{bmatrix} = \begin{bmatrix} -A & AG_k \\ Q_{k+1} & F_{k+1} \end{bmatrix}.$$

We can continue solving the subsequent systems in the same fashion down to $K = N-1$. For $k = N$, the following subsystem is solved

$$\begin{aligned} H_{Q_{(2Nn_x+1:(2N+1)n_x,2(N-1)n_x:2Nn_x)}} &= \begin{bmatrix} P_N & -I \end{bmatrix} \begin{bmatrix} I & -G_{N-1} \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} P_N & F_N \end{bmatrix}. \end{aligned}$$

This can be summarized in the following lemma.

Lemma 4.1 *The matrix $H_{hqr} = H_Q H_R$ with H_R and H_Q defined in (13) and (14) respectively, is of the form*

$$H_{hqr} = \begin{bmatrix} I & G_k & & & & & & & 0 \\ 0 & A^{\mathsf{T}} & & & & & & & 0 \\ -A & 0 & I & G_{k+1} & & & & & 0 \\ Q_{k+1} & -I & 0 & A^{\mathsf{T}} & \ddots & & & & 0 \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & & \ddots & -A & 0 & I & G_{N-1} & 0 \\ & & & & Q_{N-1} & -I & 0 & A^{\mathsf{T}} & 0 \\ & & & & & & P_N & -I & I \end{bmatrix}. \quad (15)$$

Remark 4.2 *Note that H_{hqr} coincides with D_{hqr} everywhere but in the last n_x-block column. The matrix D_{hqr} has a block $-I$ placed in the second n_x-block row, whereas the matrix H_{hqr} has a block I in the last n_x-block row. Hence, H_Q and H_R are in fact the L and U factors of an incomplete LU factorization of matrix D_{hqr} whose residual matrix is $R = H_Q H_R - D_{hqr} \neq 0$, as opposed to a standard LU factorization in which $R = 0$ (Saad, 2003, sec. 10.3)*

4.2 Companion-like matrix

Due to the upper block-triangular structure of H_R in (13), it is easy to see that the product of matrices D_{hqr} and H_R^{-1} is

$$D_{hqr} H_R^{-1} = \begin{bmatrix} I & & & & & & & & 0 \\ 0 & A^{\mathsf{T}} & & & & & & & -I \\ -A & AG_k & I & & & & & & 0 \\ Q_{k+1} & F_{k+1} & 0 & A^{\mathsf{T}} & & & & & 0 \\ & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\ & & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ & & & \ddots & -A & AG_{N-2} & I & & 0 \\ & & & & Q_{N-1} & F_{N-1} & 0 & A^{\mathsf{T}} & 0 \\ & & & & & & P_N & F_N & 0 \end{bmatrix}, \quad (16)$$

which is the same as the preconditioner H_Q (14) in all its columns except for the last one. The resulting matrix after the multiplication of the preconditioners H_Q (pre-multiplied) and H_R (post-multiplied) in matrix D_{hqr} is depicted in the following result.

Theorem 4.3 *The matrix $\tilde{D}_{hqr} = H_Q^{-1} D_{hqr} H_R^{-1}$ is given as*

$$\tilde{D}_{hqr} = \begin{bmatrix} I & & & & & & & & W_k \\ & I & & & & & & & Y_k \\ & & I & & & & & & W_{k+1} \\ & & & I & & & & & Y_{k+1} \\ & & & & \ddots & & & & \vdots \\ & & & & & \ddots & & & \vdots \\ & & & & & & I & & W_{N-1} \\ & & & & & & & I & Y_{N-1} \\ & & & & & & & & Z_N \end{bmatrix}, \qquad (17)$$

a companion-like matrix [1], *with the entries in its last $n_x-block$ column satisfying the recursion from $k = 1, \ldots, N-1$.*

$$\begin{bmatrix} W_{k+1} \\ Y_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} -A & AG_k \\ A^{-\intercal}Q_{k+1} & A^{-\intercal}F_{k+1} \end{bmatrix}}_{\mathscr{K}_{H_k}} \begin{bmatrix} W_k \\ Y_k \end{bmatrix}, \qquad (18)$$

and for $k = N$

$$Z_N = P_N W_{N-1} + F_N Y_{N-1}, \qquad (19)$$

with initial conditions $W_k = 0$ and $Y_k = -A^{-\intercal}$.

Proof. Since matrices (14) and (16) are equivalent except for the last $n_x-block$ column, it is simple to demonstrate the structure of (17), from the first $n_x-block$ column to the second last $n_x-block$ column. In this sense, the product $H_Q^{-1} D_{hqr} H_R^{-1}$ corresponding to such column produces a diagonal of identity matrices I of size $n_x \times n_x$. Regarding the last $n_x-block$ column of the companion-like matrix, it can be observed that the only elements of H_Q^{-1} that will be affected by the product are the ones located at the second $n_x-block$ column, since the last $n_x-block$ column of $D_{hqr} H_R^{-1}$ is almost empty except for the identity matrix in its second $n_x-block$ row. Namely

$$\underbrace{\begin{bmatrix} \mathscr{C}_{(0,2)} \\ \mathscr{C}_{(1,2)} \\ \mathscr{C}_{(2,2)} \\ \mathscr{C}_{(3,2)} \\ \vdots \\ \vdots \\ \mathscr{C}_{(N-2,2)} \\ \mathscr{C}_{(N-1,2)} \\ \mathscr{C}_{(N,2)} \end{bmatrix}}_{H_Q^{-1}(:,2n_x)} \times \underbrace{\begin{bmatrix} -I \end{bmatrix}}_{D_{hqr}H_R^{-1}(2n_x, Nn_x)} = \underbrace{\begin{bmatrix} -\mathscr{C}_{(0,N)} \\ -\mathscr{C}_{(1,N)} \\ -\mathscr{C}_{(2,N)} \\ -\mathscr{C}_{(3,N)} \\ \vdots \\ \vdots \\ -\mathscr{C}_{(N-2,N)} \\ -\mathscr{C}_{(N-1,N)} \\ -\mathscr{C}_{(N,N)} \end{bmatrix}}_{\tilde{D}_{hqr}(:,Nn_x)},$$

where i and j, from $\mathscr{C}_{(i,j)}$, are the $i^{th} n_x-block$ row and the $j^{th} n_x-block$ column respectively. The next step is to derive the coefficients of $H_Q^{-1}(:,2n_x)$, which can be done by solving the system $H_Q X = \mathscr{B}$ with respect to solely the columns $(:,2n_x)$. If $X = [X_0, \ldots, X_N]^\intercal$ and $\mathscr{B} = [\mathscr{B}_0, \ldots, \mathscr{B}_N]^\intercal$, it derives from (14) that the coefficients of $\mathscr{B}_1$ are described by the following recursion

$$\begin{aligned} X_0 &= \underbrace{[0]}_{\mathscr{C}_{(0,2)}} \mathscr{B}_1, \\ X_1 &= \underbrace{\left[A^{-\intercal}\right]}_{\mathscr{C}_{(1,2)}} \mathscr{B}_1, \\ X_2 &= \underbrace{\left[(A)\mathscr{C}_{(0,2)} - (AG_0)\mathscr{C}_{(1,2)}\right]}_{\mathscr{C}_{(2,2)}} \mathscr{B}_1, \\ X_3 &= \underbrace{\left[-\left(A^{-\intercal}Q_1\right)\mathscr{C}_{(0,2)} - \left(A^{-\intercal}F_1\right)\mathscr{C}_{(1,2)}\right]}_{\mathscr{C}_{(3,2)}} \mathscr{B}_1, \\ &\vdots \qquad\qquad \vdots \\ X_N &= \underbrace{\left[-(P_N)\mathscr{C}_{(N-2,2)} - (F_N)\mathscr{C}_{(N-1,2)}\right]}_{\mathscr{C}_{(N,2)}} \mathscr{B}_1, \end{aligned}$$

which resembles precisely the recursive sequence of (18) and the structure of the companion-like matrix (17) is now verified. □

5 Hamiltonian recursion algorithm

The form of the companion-like matrix (17) allows the direct factorization of system (11). This direct method, which we call the Hamiltonian recursion method, is stated in Algorithm 1.

[1] Matrices of the form (17) do not seem to have a special name. Companion matrices have identities on the first subdiagonal and nonzeros in the last column, while here we are dealing with an upper-triangular matrix having identities on the diagonal and nonzeros in the last column. This is also a special case of an arrowhead matrix. We use the term *companion-like* mainly for brevity.

Algorithm 1 Factorization of system (11) via the companion-like matrix $\tilde{D}_{hqr}$

function `ham-dire-mpc`(Q_i, R_i, P_N, A, B, b_{hqr})
  1) Computation of $\tilde{b}_{hqr} = H_Q^{-1} b_{hqr}$**:**
    $\tilde{b}^w_{hqr_k} = -\bar{r}_{\lambda_{k+1}}$, $\tilde{b}^y_{hqr_k} = -A^{-\mathsf{T}}\bar{r}_{x_k}$
    for $k = 0$ to $N-1$ **do**
      $$\begin{bmatrix} \tilde{b}^w_{hqr_{k+1}} \\ \tilde{b}^w_{hqr_{k+1}} \end{bmatrix} = -\mathscr{K}_{H_k} \begin{bmatrix} \tilde{b}^w_{hqr_k} \\ \tilde{b}^y_{hqr_k} \end{bmatrix} - \begin{bmatrix} \bar{r}_{\lambda_{k+2}} \\ \bar{r}_{x_{k+1}} \end{bmatrix}$$
    end for
    $\tilde{b}_{hqr_N} = -P_N \tilde{b}^w_{hqr_{N-1}} - F_N \tilde{b}^y_{hqr_{N-1}} - \bar{r}_{x_N}$
  2) Recursive computation:
    $W_k = 0$, $Y_k = A^{-\mathsf{T}}$
    for $k = 0$ to $N-1$ **do**
      $$\begin{bmatrix} W_{k+1} \\ Y_{k+1} \end{bmatrix} = \mathscr{K}_{H_k} \begin{bmatrix} W_k \\ Y_k \end{bmatrix}$$
    end for
    $Z_N = P_N W_{N-1} + F_N Y_{N-1}$
    $\tilde{z}_{hqr_N} = Z_N^{-1} \tilde{b}_{hqr_N}$
  3) Solving system $\tilde{D}_{hqr}\tilde{z}_{hqr} = \tilde{b}_{hqr}$**:**
    for $k = 0$ to $N-1$ **do**
      $$\begin{bmatrix} \tilde{z}^w_{hqr_k} \\ \tilde{z}^y_{hqr_k} \end{bmatrix} = \begin{bmatrix} \tilde{b}^w_{hqr_k} \\ \tilde{b}^y_{hqr_k} \end{bmatrix} - \begin{bmatrix} W_k \\ Y_k \end{bmatrix} \tilde{z}_{hqr_N}$$
    end for
  4) Compute $z_{hqr} = H_R^{-1}\tilde{z}_{hqr}$**:**
    **Initial conditions:* $\Delta\lambda_0 = \tilde{z}_{hqr_N}$
    for $k = N$ to 1 **do**
      $\Delta x_k = \tilde{z}^y_{hqr_{k-1}}$
      $\Delta\lambda_k = -G_{k-1}\Delta x_k + \tilde{z}^w_{hqr_{k-1}}$
    end for
    return $z_{hqr} = [\Delta x, \Delta\lambda]^{\mathsf{T}}$ ▷ `From system (11)`
end function

To complete the step direction vector Δp of the Newton system (5), Algorithm 2 computes the remaining part, $\tilde{\Delta} p = [\Delta u, \Delta\mu, \Delta t]^{\mathsf{T}}$.

Algorithm 2 Remaining step direction computation $\tilde{\Delta} p$

$\Delta\mu_{x_0} = \left(\mathscr{M}_{x_0}^{-1}\mathscr{T}_{x_0}\right)^{-1}\left(\omega\Delta x_0 + r^x_{\mu_0} - \mathscr{M}_{x_0}^{-1} r^x_{t_0}\right)$
$\Delta t_{x_0} = -\mathscr{M}_{x_0}^{-1}\left(\mathscr{T}_{x_0}\Delta\mu_{x_0} + r^x_{t_0}\right)$
for $k = 0$ to $N-1$ **do** ▷ `Forward substitution`
  $\Delta u_k = -R_k^{-1}\left(B^{\mathsf{T}}\Delta\lambda_{k+1} + \bar{r}_{u_k}\right)$
  $f_\mu = \left(\omega\Delta x_{k+1} + r^x_{\mu_{k+1}} - \mathscr{M}_{x_{k+1}}^{-1} r^x_{t_{k+1}}\right)$
  $\Delta\mu_{x_{k+1}} = \left(\mathscr{M}_{x_{k+1}}^{-1}\mathscr{T}_{x_{k+1}}\right)^{-1} f_\mu$
  $\Delta\mu_{u_k} = \left(\mathscr{M}_{u_k}^{-1}\mathscr{T}_{u_k}\right)^{-1}\left(\phi\Delta u_k + r^u_{\mu_k} - \mathscr{M}_{u_k}^{-1} r^u_{t_k}\right)$
  $\Delta t_{x_{k+1}} = -\mathscr{M}_{x_{k+1}}^{-1}\left(\mathscr{T}_{x_{k+1}}\Delta\mu_{x_{k+1}} + r^x_{t_{k+1}}\right)$
  $\Delta t_{u_k} = -\mathscr{M}_{u_k}^{-1}\left(\mathscr{T}_{u_k}\Delta\mu_{u_k} + r^u_{t_k}\right)$
end for
$\tilde{\Delta} p = [\Delta u, \Delta\mu, \Delta t]^{\mathsf{T}}$

5.1 Recursive sequence structure

The matrix in system (18) has a symplectic structure, as the Hamiltonian system (9). The invertibility of this matrix is guaranteed (throughout the entire recursion $k = 0, \ldots, N-1$) as the following lemma shows.

Lemma 5.1 *Let A be nonsingular. Then the system matrix* (18)

$$\mathscr{K}_{H_k} = \underbrace{\begin{bmatrix} A & \\ & A^{-\mathsf{T}} \end{bmatrix}}_{\mathscr{K}_A} \underbrace{\begin{bmatrix} -I & G_k \\ Q_{k+1} & -(Q_{k+1}G_k + I) \end{bmatrix}}_{\mathscr{K}_S},$$

is nonsingular.

Proof. $\mathscr{K}_A$ is invertible provided that A is nonsingular, i.e.

$$\mathscr{K}_A^{-1} = \begin{bmatrix} A^{-1} & \\ & A^{\mathsf{T}} \end{bmatrix}.$$

Regarding the case of $\mathscr{K}_S$, a more elaborated analysis is required. As demonstrated in (Rao et al., 1998, sec. 3.2), the matrices Q_k are symmetric positive semidefinite for $k = 1, \ldots, N-1$, and the matrices R_k are symmetric positive definite, and so is $G_k = BR_k{}^{-1}B^{\mathsf{T}}$. Likewise, the matrix $\mathscr{K}_S$ is symplectic (Fassbender, 2000) since it satisfies

$$\mathscr{K}_S^{\mathsf{T}} \mathscr{J} \mathscr{K}_S = \mathscr{J},$$

where

$$\mathscr{J} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}.$$

The inverse of the symplectic matrices is easy to find

$$\mathscr{K}_S^{\mathsf{T}} \mathscr{J} = \mathscr{J} \mathscr{K}_S^{-1},$$
$$\mathscr{J}^{-1} \mathscr{K}_S^{\mathsf{T}} \mathscr{J} = \mathscr{K}_S^{-1},$$

and since $\mathscr{J}^{-1} = -\mathscr{J}$ we have $\mathscr{K}_S^{-1} = -\mathscr{J}\mathscr{K}_S^{\mathsf{T}}\mathscr{J}$. After multiplications

$$\mathscr{K}_S^{-1} = \begin{bmatrix} -(Q_{k+1}G_k + I) & -G_k \\ -Q_{k+1} & -I \end{bmatrix}.$$

Therefore, the matrix $\mathscr{K}_{H_k}^{-1}$ exists at every iteration k. ☐

Remark 5.2 *Algorithm 1 is valid if and only if matrix Z_N of (19) is nonsingular. Given the matrix D_{hqr} of the Newton system (11) is nonsingular, and the invertibility of preconditioners H_Q and H_R is guaranteed, the product of these matrices, i.e., $\tilde{D}_{hqr}$ (companion-like matrix (17)) is nonsingular*

(Meyer, 2004, p.121). Since $\tilde{D}_{hqr}$ is a block upper-triangular matrix, its determinant is the product of the determinants of the diagonal blocks (Horn and Johnson, 2012, p.25). All this implies that these determinants are nonzero, including the determinant of matrix Z_N.

6 Hamiltonian/GMRES method

The factorization of the Newton system in the previous section corresponds to using a direct solver with the interior-point method. Another attractive possibility is to consider an iterative method for such a task. We will show that the special structure of the companion-like matrix ensures a fast convergence of the GMRES method within a bounded number of iterations. We call this the Hamiltonian/GMRES method. GMRES stands for generalized minimal residual and it was proposed in Saad and Schultz (1986). This Krylov method is used for solving general (non-symmetric) systems of linear equations such as in our case, as opposed to the MINRES method which is used for symmetric matrices, for instance the augmented system (5).

6.1 Convergence proof

The algorithmic complexity of direct solvers for linear systems is bounded. Namely, the factorization of a given system is carried out by a predetermined recursive sequence (such as LU, QR or Cholesky). Its computation directly gives rise to the solution (only a `for-loop` needed). On the contrary, for iterative methods (Golub and Loan, 2013) the solution of the system is obtained after a sequence of approximate solutions converges (a `while-loop` is needed).
From a practical viewpoint, ensuring fast convergence is important for efficient optimization solvers in MPC. Recently, there have been several works that have addressed this topic for different types of iterative methods. To name a few of them:

(1) In Richter et al. (2009), the fast-gradient method is used. This work also provided an a-priori upper bound for the required number of iterations. A drawback of this method is that it is only applicable for input-constrained MPC.
(2) In Shahzad et al. (2010), the MINRES method is used. By setting up a special structure of the system that splits into two parts the active and inactive constraints, the condition number of the matrix in (5) is improved. In doing so, they numerically demonstrate that the rate of convergence is faster.
(3) In Malyshev et al. (2018b), the conjugate gradient method (CG) is used, where the matrix factors of the augmented system (or the normal equations) are used as preconditioners for the CG method. The authors introduce a regularization parameter (which is a common practice for iterative methods to improve the condition number of the main matrix). The method appears to be sensitive to the choice of the regularization parameter.
(4) In Malyshev et al. (2018a), different Krylov methods are used, GMRES, QMR and BiCG. The authors compare the convergence on systems with and without preconditioners, using a method that intertwines the approach in Shahzad et al. (2010) with a condensing procedure in the main matrices.

None of the previous works related to interior-point solvers give a theoretical proof for a bound on the number of iterations in their corresponding methods. We will demonstrate that that GMRES applied to the companion-like matrix converges within a small number of iterations.
To characterize the convergence of the GMRES method applied to solve the linear system in (11) with our proposed preconditioner H_{hqr} (15), we examine the degree of the minimal polynomial of the preconditioned system matrix $\tilde{D}_{hqr}$, defined in (17). Recall that the minimal polynomial $p_{\tilde{D}}$ of $\tilde{D}_{hqr}$ is defined as the monic (that is, with leading coefficient 1) polynomial of smallest degree such that $p_{\tilde{D}}\left(\tilde{D}_{hqr}\right) = 0$. Saad and Schultz (1986) and Campbell et al. (1996) prove that, in exact arithmetic, GMRES requires at most as many iterations as the degree of $p_{\tilde{D}}$.
It is easy to see that the degree of $p_{\tilde{D}}$ is bounded by $n_x + 1$, where n_x is the number of states x and is also the dimension of matrix Z_N defined in (19). Let p_{Z_N} denote the minimal polynomial of Z_N, and consider the polynomial $p(z) = (z-1)\, p_{Z_N}(z)$. Then

$$\begin{aligned} p\left(\tilde{D}_{hqr}\right) &= \left(\tilde{D}_{hqr} - I\right) p_{Z_N}\left(\tilde{D}_{hqr}\right) \\ &= 0 \end{aligned}$$

where

$$\left(\tilde{D}_{hqr} - I\right) = \begin{bmatrix} 0 & & & & & W_k \\ & 0 & & & & Y_k \\ & & \ddots & & & \vdots \\ & & & 0 & & W_{N-1} \\ & & & & 0 & Y_{N-1} \\ & & & & & (Z_N - I) \end{bmatrix},$$

$$p_{Z_N}\left(\tilde{D}_{hqr}\right) = \begin{bmatrix} p_{Z_N}(I) & & & & & * \\ & p_{Z_N}(I) & & & & * \\ & & \ddots & & & \vdots \\ & & & p_{Z_N}(I) & & * \\ & & & & p_{Z_N}(I) & * \\ & & & & & 0 \end{bmatrix}.$$

Hence, p is a monic polynomial of at most $n_x + 1$ degree that satisfies $p\left(\tilde{D}_{hqr}\right) = 0$. The minimal polynomial of $\tilde{D}_{hqr}$ cannot be of higher degree, thus we conclude the following result.

Theorem 6.1 *Assume that GMRES is applied to solve (11) with a nonsingular system matrix having blocks of size $n_x \times n_x$, and that the left and right preconditioners H_Q (14) and H_R (13) are used. Then, in exact arithmetic, GMRES will compute the exact solution in at most $n_x + 1$ iterations.*

The numerical experiments in Section 7 will confirm that this termination property of GMRES is also observed in the presence of rounding errors. Algorithm 3 replaces the steps 2 and 3 of Algorithm 1, which now requires only matrix-vector operations.

Algorithm 3 Hamiltonian/GMRES solver for solution of system (11) using $\tilde{D}_{hqr}$

2) GMRES implementation:

function `ham(@companion-iteration,` $\tilde{b}_{hqr}$`)`
  GMRES code*: **in $\longrightarrow r_{gmres}$, **out** $\longrightarrow q_{gmres}$
  function `companion-iteration(`q_{gmres}`)`
    $w_i = 0,\ y_i = A^{-\mathsf{T}} q_{gmres_N}$
    $r_i^w = q_i^w - w_i,\ r_i^y = q_i^y - y_i$
    for $k = i$ to $N-1$ **do**
      $$\begin{bmatrix} w_{k+1} \\ y_{k+1} \end{bmatrix} = -\mathscr{K}_{H_k} \begin{bmatrix} w_k \\ y_k \end{bmatrix}$$
      $r_{k+1}^w = q_{k+1}^w - w_{k+1}$
      $r_{k+1}^y = q_{k+1}^y - y_{k+1}$
    end for
    $z_N = -P_5 w_{N-1} - F_N y_{N-1}$
    $r_{gmres_N} = q_{gmres_N} - z_N$
    return r_{gmres}
  end function
  return $\tilde{z}_{hqr}$ ▷ `Solution of system (12)`
end function

7 Numerical experiments

7.1 Feasibility

To test the Hamiltonian/GMRES solver, we use a well-known testbench for MPC: the simulation of oscillating masses that has been used in Wang and Boyd (2010) and Malyshev et al. (2018c). This problem consists of a predefined number of masses connected by springs to each other in series, and to walls on either side. Actuators between the masses exert tensions. The masses have value of 1 and all spring constants are set at 1 with no damping. The state vector $x_k \in \mathbb{R}^{n_x}$ encompasses the displacements and velocities of the masses $x = \left[q_1, q_2, \ldots, q_{n_x/2}, \dot{q}_1, \dot{q}_2, \ldots, \dot{q}_{n_x/2}\right]^{\mathsf{T}}$, and the actuators as the control input $u_k \in \mathbb{R}^{n_u}$, $u_k = [f_1, f_2, \ldots, f_{n_u}]^{\mathsf{T}}$, where $n_u = \frac{n_x}{2} - 1$. For the feasibility and convergence test, Algorithm 1 (including Algorithm 3) is coded with a sample time of $t = 0.5s$; the number of masses is set at $m = 20$ (hence $n_x = 2m = 40$), inputs $n_u = 19$ and horizon $N = 5$. The control inputs can exert a maximum force of ± 0.5 and the displacement of the states cannot exceed ± 3.8. The weighting matrices $Q = I$, $R = I$ are set; and terminal cost P is the solution to the infinite horizon LQR. The result is shown in Figure 1.

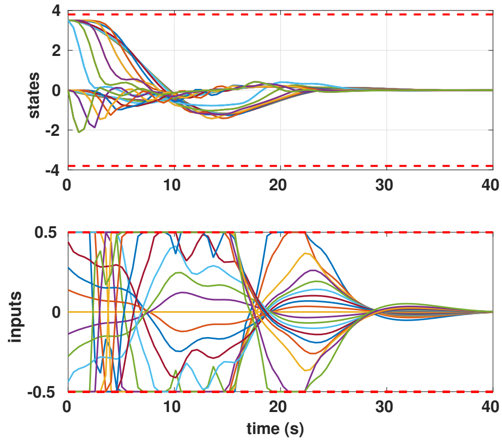


Fig. 1. Evolution of the states (with initial conditions $x_k = 3.5$ for the displacements, and $x_k = 0$ for the velocities) and the control input profile of the oscillating-masses example with $n_x = 40$ states.

7.2 Convergence under rounding errors

Theorem 6.1 establishes that for this specific example the maximum number of iterations should not surpass the limit $n_x + 1 = 41$. Figure 2 shows that this is in fact the case since one can observe that the number of GMRES iterations (with a stopping criteria of 1×10^{-6}) fluctuates between 20 to 41. The profile of the relative residual is shown in Figure 3. Figure 4 shows the profile of the GMRES iterations for a simulation with the same setting as Figure 2, except with $n_x = 160$. Here, the number of iterations fluctuate between 100 to 150.

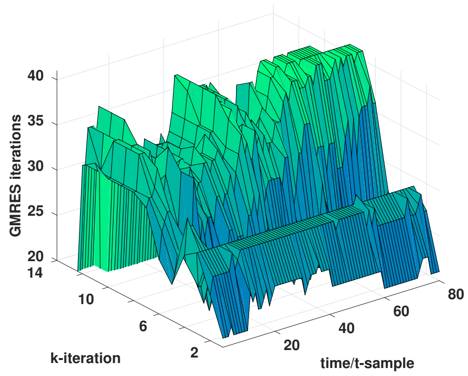


Fig. 2. GMRES convergence for the system (12) with $n_x = 40$ states throughout all the interior-point iterations over the complete runtime.

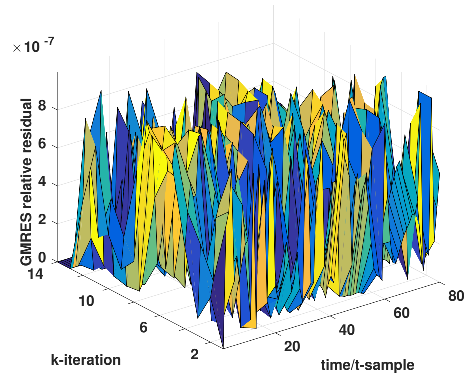


Fig. 3. Relative residual norms for system (12) with $n_x = 40$ states throughout all the interior-point iterations over the complete runtime.

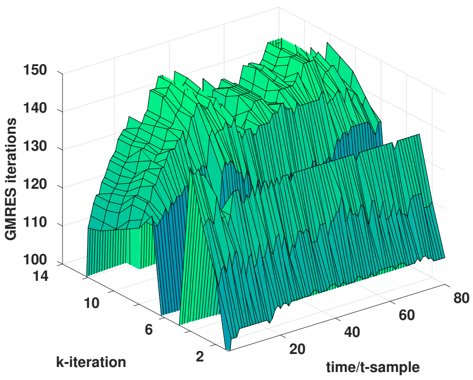
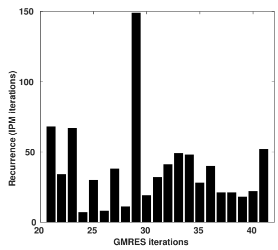


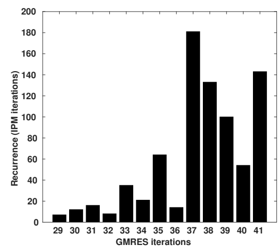
Fig. 4. GMRES convergence for the system (12) with $n_x = 160$ states ($m = 80$ masses) throughout all the interior-point iterations over the complete runtime.

7.3 *Convergence subject to different horizons*

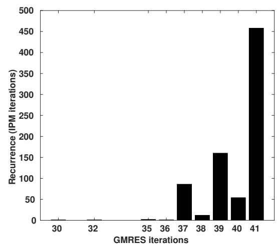
The convergence of the Hamiltonian/GMRES method is visualized in Figure 2 and 4. However, another interesting experiment is to vary the value of the horizon N in order to see how the number of GMRES iterations behaves with respect to the total of IPM iterations. Figure 5 shows four simulations of the oscillating-masses example with the same setup of parameters (such as $n_x = 40$) as previous experiments, except for $N = 5, 10, 20, 40$. Although all four tests reach the maximum number of GMRES iterations at 41 (confirming Theorem 6.1), their distributions are rather different. For instance, regarding test $N = 5$ (Figure 5a), the iterations are roughly equally distributed from iteration 21 to 41; but as the horizon increases, for instance in tests $N = 20$ and $N = 40$ (Figures 5c and 5d respectively), their distributions are assembled around the maximum number of iterations. Therefore, from this specific example we can conclude that the longer the horizon N, the closer the total number of iterations (in the presence of rounding errors) will be to its theoretical bound of $n_x + 1$ in each run of the interior-point algorithm.



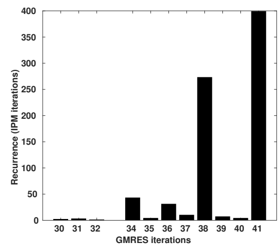
(a) $N = 5$.



(b) $N = 10$.



(c) $N = 20$.

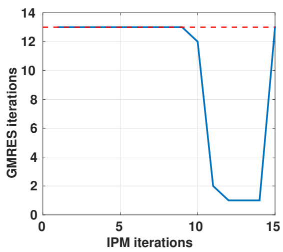


(d) $N = 40$.

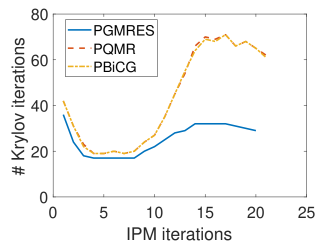
Fig. 5. Number of GMRES-iteration recurrences for different horizons throughout all the interior-point iterations over the complete runtime.

7.4 *Comparison*

It is necessary to carry out a comparison test in order to give a reasonable perspective about the feasibility and potential that the Hamiltonian/GMRES solver might have. Malyshev et al. (2018a) report several experiments with heuristically designed preconditioners applied to different configurations of the Newton method, using different Krylov methods such as GMRES, BiCG and QMR. Likewise, they also apply the approach presented in Shahzad et al. (2012), which splits the inequality constraints into two sets (active and inactive).



(a) Iterations (blue) and the theoretical bound $n_x + 1$ (red).



(b) The GMRES iterations are represented by the blue line.

Fig. 6. Comparison between the Hamiltonian/GMRES solver (Figure 6a), and the GMRES preconditioned method proposed in Malyshev et al. (2018a), in Figure 6b.

By doing so the condition number of the relevant matrix (ei-

ther with the augmented system or the normal equations) is improved and the convergence of such iterative method is sped up. The testbench used by Malyshev et al. (2018a) is the same as the one we used in our previous experiments. To match up this setup, the parameters of the oscillating-masses example that need to be changed are the states $n_x = 12$ (number of masses $m = 6$), the control inputs $n_u = 3$ and the horizon at $N = 30$. The simulation will be run for only one sample time. Although we know in advance that the maximum convergence that this simulation will have is $n_x + 1 = 13$ (due to Theorem 6.1), we replicate the same experiment so as to show a fair comparison of both proposals.
It is interesting to notice from Figure 6 the following:

(1) In Figure 6a, the Hamiltonian/GMRES solver has a maximum number of GMRES iterations of 13, going down sometimes to 2 and 1. Whereas with the GMRES method from Malyshev et al. (2018a) in Figure 6b, the minimum number of iterations is considerably above ours, around 19. It also varies greatly, reaching a maximum limit of around 38.
(2) With the Hamiltonian/GMRES solver, the total number of iterations is 15, whereas with the other GMRES method it is 20.

8 The case of A being singular

In real-time applications, the systems are normally subject to time delays. In this case, the matrix A of the state space representation becomes singular. When this happens the inverse of the left preconditioner H_Q (14) no longer exists and hence the companion-like matrix (17) cannot be formed. Therefore, we will propose a change on the left preconditioner H_Q in order to deal with matrix A being singular.

8.1 Formulation

Proposition 8.1 *Let D_{hqr} and H_R be the matrices defined in (11) and (13) respectively. The left preconditioner H_Q (a lower block-triangular matrix) is*

$$H_{Q_\alpha} = \begin{bmatrix} I &&&&&&&&& \\ 0 & A_\alpha^\intercal &&&&&&&& \\ -A & AG_k & I &&&&&&& \\ Q_{k+1} & F_{k+1} & 0 & A_\alpha^\intercal &&&&&& \\ & \ddots & \ddots & \ddots & \ddots &&&&& \\ && \ddots & \ddots & \ddots & \ddots &&&& \\ &&& \ddots & -A & AG_{N-2} & I &&& \\ &&&& Q_{N-1} & F_{N-1} & 0 & A_\alpha^\intercal && \\ &&&&&& P_N & F_N & I & \end{bmatrix}, \quad (20)$$

so that not only does its inverse exist but also $\tilde{D}_{hqr_\alpha} = H_{Q_\alpha}^{-1} D_{hqr} H_R^{-1}$. The Matrix $A_\alpha^\intercal = (A^\intercal + \alpha I)$ is nonsingular provided that $\alpha > 0$.

The direct method presented in Algorithm 1 can no longer be used since it requires the explicit form of the companion-like matrix $\tilde{D}_{hqr}$ (17), which is not available when the α parameter is introduced into the diagonal of preconditioner H_{Q_α}. It is simple to observe that the best selection of the α parameter (ideally) should be such that matrix $\tilde{D}_{hqr_\alpha}$ resembles as close as possible to matrix $\tilde{D}_{hqr}$, but without compromising the increase on its condition number κ.
We will present next some numerical experiments that propose some heuristic rules on how to set up (tuning) this α parameter. Likewise, we explore the spectra properties of the system matrix.

8.2 Numerical experiment

We use the following discrete-time state space model of a citation aircraft with delay that appears in Maciejowski (2002)

$$x_{k+1} = Ax_k + Bu_k, \quad (21)$$

where

$$A = \begin{bmatrix} 0.85679 & 0 & 0.08312 & 0 & -0.03109 \\ -0.02438 & 1 & 0.09057 & 0 & -0.02788 \\ -0.46049 & 0 & 0.80976 & 0 & -0.27938 \\ -12.03761 & 12.82000 & 0.03680 & 1 & 0.05578 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.07163 & -0.05156 & -1.25304 & 0.11096 & 1 \end{bmatrix}^\intercal.$$

The matrix A is evidently singular. We will first show the variation of the condition number κ of matrix $\tilde{D}_{hqr_\alpha}$ when α parameter varies. It is necessary that $\alpha > 0$, as $\kappa = \infty$ otherwise. Matrices H_{Q_α} (20), H_R (13) and D_{hqr} (11) are built with parameters: horizon $N = 2$, $Q = I$, $R = I$ and P being the solution of the Riccati equation on the infinite-horizon LQR. Figure 7 shows a simulation where the α parameter varies from 0.01 to 1. It is obvious that when $\alpha \to 0$, the condition number $\kappa \to \infty$. Conversely, when α increases, κ decreases. Namely, the main objective is to emulate as much as possible $\tilde{D}_{hqr_\alpha} \approx \tilde{D}_{hqr}$, which means $\alpha \to 0$, so that we can be close to the convergence established in Theorem 6.1. However, a big condition number should be avoided as it is a main factor for cancellations within the algorithm.

8.3 Eigenvalue trajectories

The eigenvalues of the companion-like matrix $\tilde{D}_{hqr}$ (17) are divided in two sets. First, $2Nn_x$ eigenvalues are exactly 1 and second, the eigenvalues of matrix Z_N (19), which

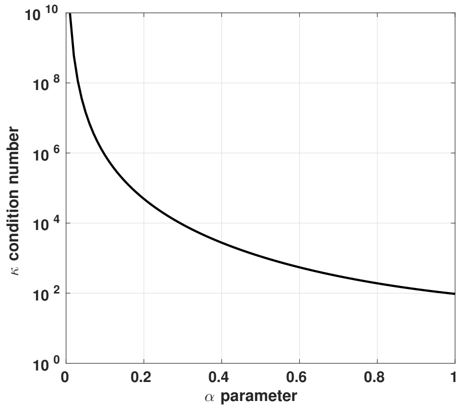
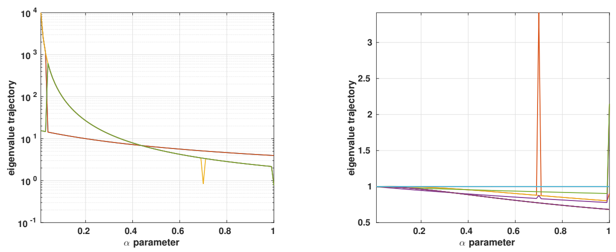


Fig. 7. Condition number κ of matrix $\tilde{D}_{hqr_\alpha} = H_{Q_\alpha}^{-1} D_{hqr} H_R^{-1}$ subject to variations of the α parameter. Experiment set with horizon $N = 2$.

appear to be arbitrary. Hence, the value of the α parameter has to be such that the $2Nn_x$ eigenvalues of matrix $\tilde{D}_{hqr_\alpha}$ are clustered as close as possible to 1. Figure 8a shows the



(a) Eigenvalues of Z_N-*like* matrix. The number of these eigenvalues is $n_x = 5$.

(b) $2Nn_x$ eigenvalues. Since $N = 2$ and $n_x = 5$, the number of these eigenvalues is 20.

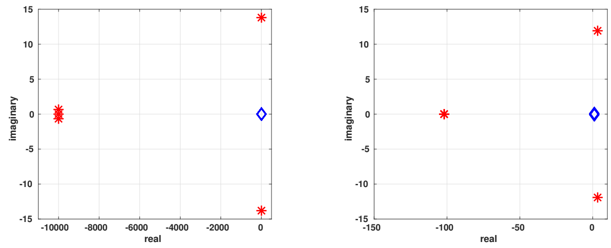
Fig. 8. Eigenvalue trajectories of matrix $\tilde{D}_{hqr_\alpha} = H_{Q_\alpha}^{-1} D_{hqr} H_R^{-1}$. Each trajectory ($y$ axis) is the absolute value $|\mu|$ of an eigenvalue $\mu = a + bi$. The x axis represents the variation of the α parameter.

n_x eigenvalues corresponding to the Z_N-*like* matrix. These eigenvalues are arbitrary and depend on the parameters of the problem shown in Algorithm 1. The important result is the one depicted in Figure 8b regarding the $2Nn_x$ eigenvalues of matrix $\tilde{D}_{hqr_\alpha}$. For small values of $\alpha \to 0$, these eigenvalues are clearly clustered around 1 and as α increases this cluster starts dispersing.

Figure 9 shows the locus of the eigenvalues lying in the imaginary plane for four different values of $\alpha = 0.01, 0.10, 0.50, 1.00$. It shows that for small values of α, i.e. 0.01 and 0.10, the $2Nn_x$ eigenvalues (in blue) are clustered very close to one, as opposed to larger values of α, i.e. 0.50 and 1.00, where these eigenvalues are now dispersed.

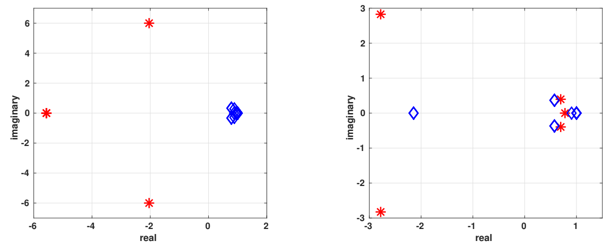
8.4 Convergence

We now carry out an experiment that includes the solution of the preconditioned system (12), with $\tilde{D}_{hqr_\alpha} = H_{Q_\alpha}^{-1} D_{hqr} H_R^{-1}$



(a) $\alpha = 0.01$.

(b) $\alpha = 0.10$.



(c) $\alpha = 0.50$.

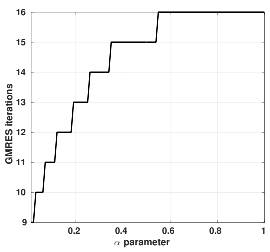
(d) $\alpha = 1.00$.

Fig. 9. The locus of $2Nn_x$ eigenvalues (blue) and n_x eigenvalues (red). Imaginary plane for the eigenvalues $\mu = a + bi$ of matrix $\tilde{D}_{hqr_\alpha} = H_{Q_\alpha}^{-1} D_{hqr} H_R^{-1}$. Experiment set with horizon $N = 2$.

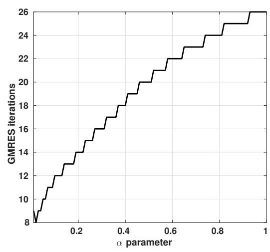
for the discrete-time system (21). The initial conditions x_0 are set to zero, and $b_{hqr} = [Ax_0, -Qx_0, 0, \ldots, 0] \in \mathbb{R}^{(2N+1)n_x}$. Figure 10 shows the result of this experiment. The GMRES convergence becomes slower as the horizon N and α parameter rise. Nonetheless, for small values of α convergence is comparable to that of the companion-like matrix (17). For a fifth-order system this would be convergence within at most $n_x + 1 = 6$ iterations. Figure 11 shows the consequences of using small values of α. Namely, the condition number κ of $\tilde{D}_{hqr_\alpha}$ increases significantly, which is likely to cause numerical cancellations on the algorithm.

9 Conclusion

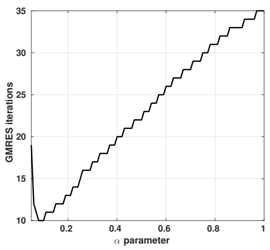
In this paper, new numerical algorithms for solving the Newton system of the interior-point method have been proposed. When the system is transformed into the Hamiltonian form, we define left and right preconditioners that happen to be the L and U factors respectively, of an incomplete LU factorization of the main matrix D_{hqr}. When the preconditioners are applied to this matrix, the resulting matrix has a companion-like form. The solution of this system can be carried out by two methods. (i) In section 5, we showed a direct algorithm (Hamiltonian recursion) that factorizes the companion-like matrix by computing a forward substitution of a sequence of symplectic submatrices, where we proved that the invertibility of these submatrices is guaranteed. (ii) In section 6, we proposed an iterative algorithm (Hamiltonian/GMRES) based on a Krylov method and showed that this combination will compute the exact solution of the Newton system in at most $n_x + 1$ iterations, where n_x is the number of states. With numerical experiments, we also demonstrated that this



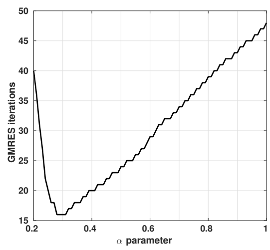
(a) $N = 2$.



(b) $N = 5$.

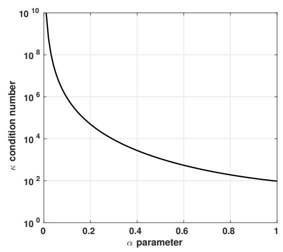


(c) $N = 10$.

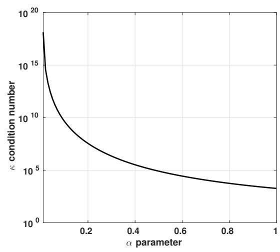


(d) $N = 20$.

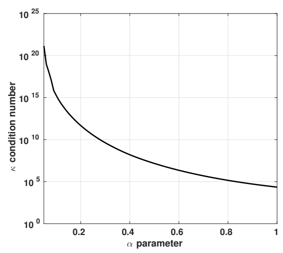
Fig. 10. *GMRES* iterations (y axis) for different values of the horizon N. The x axis represents the α parameter.



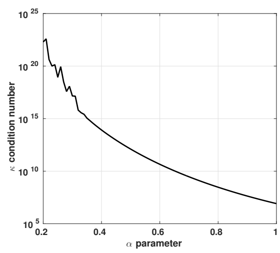
(a) $N = 2$.



(b) $N = 5$.



(c) $N = 10$.



(d) $N = 20$.

Fig. 11. GMRES experiment with different values of the horizon N. The x axis represents the α parameter and the y axis is the condition number κ of matrix $\tilde{D}_{hqr_\alpha} = H_{Q_\alpha}^{-1} D_{hqr} H_R^{-1}$.

termination property also holds in the presence of rounding errors.
The companion-like structure is lost when the matrix A is singular. For the Hamiltonian/GMRES method, we addressed this problem by introducing a parameter α that with proper tuning can result in GMRES convergence close to the bound $n_x + 1$ iterations. A tradeoff is that for small values of α the condition number κ increases. Our methods presented here are not immune to problems with ill-conditioning, similarly to most other interior-point algorithms that are currently available. Such problems remain the subject of future work.

Acknowledgements

The research leading to these results has received funding from two entities, Festo AG & Co.KG and the Mexican Council of Science and Technology (CONACYT). We would also like to thank Prof. David J. Bell for the helpful discussions during the course of this work and Alexander Malyshev for kindly providing Figure 6b.

References

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

A. E. Bryson and Y. C. Ho. *Applied Optimal Control; Optimization, Estimation and Control*. Halsted Press Book, 1975.

S.L. Campbell, I.C.F. Ipsen, C.T.Kelley, and C.D. Meyer. GMRES and the minimal polynomial. *BIT Numerical Mathematics*, 36:664–675, 1996.

D. Chmielewski and V. Manousiouthakis. On Constrained Infinite-time Linear Quadratic Optimal Control. *Systems and Control Letters*, 29:121–129, 1996.

C. R. Cutler and B. L. Ramaker. Dynamic matrix control -a computer control algorithm. *AICHE national meeting*, 1979.

M. Diehl, H.G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and nonlinear Model Predictive Control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577–585, 2002.

M. Diehl, H. J. Ferreau, and N. Haverbeke. *Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation*, volume 384 of *Lecture Notes in Control and Information Sciences*. Springer, 2009.

H. Fassbender. *Symplectic Methods for the Symplectic Eigenproblem*. Kluwer Academic Plenum Publishers, 2000.

C. Feller and C. Ebenbauer. Sparsity-Exploiting Anytime Algorithms for Model Predictive Control: A Relaxed Barrier Approach. *IEEE Transactions on Control Systems Technology*, pages 1–11, 2018.

G. Frison, D. K. M. Kufoalor, L. Imsland, and J. B. Jørgensen. Efficient Implementation of Solvers for Linear Model Predictive Control on Embedded Devices. *Proceedings of IEEE Conference on Control Applications*, pages 1954–1959, 2014.

G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins, fourth edition, 2013.

J. Gondzio. Matrix-free interior point method. *Computational Optimization and Applications*, 52:457–480, 2012.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, New York, NY, USA, 2nd edition, 2012.

J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari. Embedded Online Optimization for Model Predictive Control at Megahertz Rates. *IEEE Transactions on Automatic Control*, pages 3238–3251, 2014.

R.E. Kalman. Contributions to the theory of optimal control. *Boletin de la Sociedad Mexicana de Matematicas*, 5:102–119, 1960.

M. Katliar, J. Fischer, G. Frison, M. Diehl, H. Teufel, and H. H. Bülthff. Nonlinear Model Predictive Control of a Cable-robot-based motion simulator. *Proceedings of the 20th IFAC World Congress*, 50:9833–9839, July 2017.

J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.

A. Malyshev, R. Quirynen, and A. Knyazev. Preconditioned Krylov iterations and condensing in interior point MPC method. *Proceedings of the 6th IFAC Conference on Nonlinear Model Predictive Control*, pages 478–483, 2018a.

A. Malyshev, R. Quirynen, and A. Knyazev. Preconditioning of conjugate gradient iterations in interior point MPC method. *Proceedings of the 6th IFAC Conference on Nonlinear Model Predictive Control*, pages 484–489, 2018b.

A. Malyshev, R. Quirynen, A. Knyazev, and S. Di Cairano. A regularized Newton solver for linear Model Predictive Control. *Proceedings of the European Control Conference (ECC)*, 2018c.

S. Mehrotra. On the implementation of a Primal-Dual Interior-Point method. *Society for Industrial and Applied Mathematics*, 2:4:575–601, 1992.

C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2004.

I. Nielsen and D. Axehill. Low-Rank Modifications of Riccati Factorizations for Model Predictive Control. *IEEE Transactions on Automatic Control*, 63(3):872–879, 2018.

T. Ohtsuka. A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4):563 – 574, 2004.

E. Poupard. *Hamiltonian Decomposition for Online Implementation of Model Predictive Control*. PhD thesis, The University of Manchester, School of Electrical and Electronic Engineering, 2018.

E. Poupard and W. P. Heath. Hamiltonian Decomposition for Model Predictive Control. *Proceedings of the 6th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control*, 24:193–198, May 2018.

E. Poupard, W. P. Heath, and S. Güttel. A Hamiltonian Decomposition-based Splitting Method for Interior Point Solvers in Model Predictive Control. *Proceedings of the 58th IEEE Conference Decision and Control*, pages 4337–4342, 2019.

S. J. Qin and T. A. Badgwell. A survey of industrial Model Predictive Control technology. *Control Engineering Practice*, pages 733–764, 2003.

C.V. Rao, S.J. Wright, and J.B. Rawlings. Application of Interior-point Methods to Model Predictive Control. *Optimization Theory and Applications*, 99:723–757, 1998.

J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model algorithmic control of industrial processes. *IFAC Proceedings Volumes*, 10(16):103–120, 1977.

S. Richter, C. N. Jones, and M. Morari. Real-time Input-Constrained MPC Using Fast Gradient Methods. *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 7387–7393, 2009.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, pages 856–869, 1986.

P. O. M. Scokaert and James B. Rawlings. Constrained Linear Quadratic Regulation. *IEEE Transactions on Automatic Control*, 43:1163–1169, 1998.

A. Shahzad, E. C. Kerrigan, and G. A. Constantinides. A Fast Well-conditioned Interior Point Method for Predictive Control. *Proceedings of the IEEE Conference Decision and Control*, pages 508–513, 2010.

A. Shahzad, E. C. Kerrigan, and G. A. Constantinides. A Stable and Efficient Method for Solving a Convex Quadratic Program with Application to Optimal Control. *SIAM Journal on Optimization*, 22:1369–1393, 2012.

M. Sznaier and M. J. Damborg. Suboptimal Control of Linear Systems with State and Control Inequality Constraints. *Proceedings of the 25th Conference on Decision and Control*, pages 761–762, 1987.

Y. Wang and S. Boyd. Fast Model Predictive Control Using Online Optimization. *IEEE Transactions on Control Systems Technology*, 18:267–278, 2010.

A. G. Wills and W. P. Heath. Barrier function based Model Predictive Control. *Automatica*, 40:1415–1422, 2004.

A. G. Wills, D. Bates, A. J. Fleming, B. Ninness, and S. O. R. Moheimani. Model Predictive Control applied to Constraint Handling in Active Noise and Vibration Control. *IEEE Transactions on Control Systems Technology*, 16: 3–12, 2008.

S. J. Wright. *Primal-Dual Interior Point Methods*. Society for Industrial and Applied Mathematics, 1997.