2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

A regularized Newton solver for linear model predictive control

Alexander Malyshev[1], Rien Quirynen[2], Andrew Knyazev[2] and Stefano Di Cairano[2]

Abstract— **We investigate direct numerical solvers in linear model predictive control, where the prediction model is given by linear systems subject to linear inequality constraints on the state and the input, and the performance index is convex and quadratic. The inequality constraints are treated by the primal-dual interior-point method. We propose a novel direct solver based on the augmented Lagrangian regularization of a reduced Hessian. The new solver has the same arithmetic complexity as the factorized Riccati recursion. The direct solver can be implemented in terms of BLAS3 matrix operations.**

I. Introduction

This work is concerned with fast numerical methods for solving optimal control problems applicable in the model predictive control (MPC) approach, which is described, for example, in [1]. We consider the so-called Newton-type methods. A rather comprehensive list of numerical techniques used in Newton-type methods for MPC is given, e.g., in [2]. The work in [2], however, focuses more on nonlinear MPC and moving horizon estimation.

The core computational problem in MPC is the quadratic program (QP), where the cost function is determined by the convex quadratic MPC performance index, the linear equality constraints are determined by the linear model dynamics, and the inequality constraints by the linear constraints on the model states and inputs. The inequality constraints in the QP are usually treated by the active-set (AS) [3], [4] or by the interior-point (IP) methods [5]–[12]. The main computationally intensive effort in both approaches is the execution of Newton's method in the control prediction problem over a finite horizon, which consists in solving a series of structured systems of linear equations. The matrices of such systems are very sparse and can be banded under suitable orderings of their rows and columns.

In this paper, we restrict ourselves to direct methods for solving the structured systems of linear equations arising in IP methods. The IP convergence is claimed in [13]–[15] to be almost independent of the problem size and conditioning. If we denote by n_x, n_u, n_c, and N the dimensions of the state vector and the control input vector, the number of the inequality constraints, and the horizon length, respectively, then arithmetic complexity of structured direct methods usually amounts to $O(N(n_x^3 + n_u^3 + n_c^3))$ flops.

*The work of the first author was supported by MERL

[1]Alexander Malyshev is with University of Bergen, Department of Mathematics, PB 7803, 5020 Bergen, Norway,
alexander.malyshev@uib.no

[2]Rien Quirynen, Andrew Knyazev and Stefano Di Cairano are with Mitsubishi Electric Research Laboratories (MERL) 201 Broadway, 8th floor, Cambridge, MA 02139, USA,
quirynen@merl.com, knyazev@merl.com,
dicairano@merl.com

Active-set solvers may have lower arithmetic cost per iteration by exploiting the low-rank updates of matrix factorizations when changing the current guess for the active set [16]–[18]. Such updates can be implemented in terms of only BLAS2 operations. Direct methods, in contrast, can be implemented in terms of BLAS3 operations, which are highly efficient on contemporary computer architectures [14], [15]. A modern implementation of the BLAS kernels [19] for embedded optimization can be found in [20].

We consider four direct solvers, which are used in the IP method for solving optimal control problems. Historically, the first one is the classical Riccati recursion (Section IV), which has been tailored to the MPC method in [7]. Another popular direct solver is proposed and thoroughly tuned to the warm-started MPC implementation in [10] (Section V). The latter method is referred to as the Schur complement based solution in [21], while the resulting block-tridiagonal system is originally known as the normal equations in [6].

The authors of [21] have recently revised the classical Riccati recursion and the Schur complement based solver from [10]. As a result, they have discovered a more efficient version of the Riccati recursion called the *factorized Riccati recursion*. The arithmetic cost of the new version is about 1.7 times lower than the cost of the classical Riccati recursion. An additional contribution of [21] consists in the accurate presentation of the considered direct methods in terms of the BLAS operations and precise evaluation of all corresponding arithmetic costs.

Our contribution is a novel structured direct solver for the IP iterations applied to the implementation of MPC. The idea of this solver is inspired by our interest in preconditioned iterative methods for solving saddle-point linear systems as in [22]. One of the preconditioners used in [22], [23] is related to the augmented Lagrangian method [4], which is quite popular in numerical optimization of constrained problems. As we have been writing the present paper, a variant of the augmented Lagrangian regularization is proposed in [24]. Section VI describes the new direct solver in detail and provides its arithmetic complexity. It turns out that the arithmetic costs of the factorized Riccati recursion and of the solver based on the augmented Lagrangian regularization are exactly equal. The other two methods, the classical Riccati recursion and the Schur complement based solution, have a higher arithmetic complexity.

Section VII reports on several numerical experiments with the novel direct solver. In particular, we illustrate the behavior of the residuals and number of iterations under various choices of the regularization parameter and indicate the interval for the parameter, where the method works.

II. Interior point method for linear MPC

We consider the MPC problem over the finite horizon $t = \tau_0 < \tau_1 < \cdots < \tau_N = t+T$ of length N with the dynamics described by the following discrete-time linear system

$$x_{i+1} = A_i x_i + B_i u_i + a_i, \quad i = 0, 1, \ldots, N-1, \tag{1}$$

where $A_i \in \mathbb{R}^{n_x \times n_x}$, $B_i \in \mathbb{R}^{n_x \times n_u}$, $x_i \in \mathbb{R}^{n_x}$ is the state vector, $u_i \in \mathbb{R}^{n_u}$ is the control input vector. A measurement or estimate of the state at the current time instant is taken as the initial state x_0 and denoted by $\hat{x}_0 = x_0 \in \mathbb{R}^{n_x}$. The objective of the prediction is to find the sequence of optimal control inputs $u_0, \ldots, u_{N-1}$ subject to the equality constraints (1) and the inequality constraints

$$G_i^x x_i + G_i^u u_i \le g_i, \quad i = 0, \ldots, N-1, \tag{2}$$

$$G_N^x x_N \le g_N, \tag{3}$$

that minimizes the quadratic performance index

$$\sum_{i=0}^{N-1} \frac{1}{2}\left(x_i^T Q_i x_i + u_i^T R_i u_i + 2u_i^T S_i x_i\right) + q_i^T x_i + r_i^T u_i + \frac{1}{2} x_N^T Q_N x_N + q_N^T x_N, \tag{4}$$

where $G_i^x \in \mathbb{R}^{n_{c_i} \times n_x}$, $G_i^u \in \mathbb{R}^{n_{c_i} \times n_u}$, $Q_i \in \mathbb{R}^{n_x \times n_x}$, $R_i \in \mathbb{R}^{n_u \times n_u}$, and $S_i \in \mathbb{R}^{n_u \times n_x}$.

By packing the decision variables into the vector

$$d = \begin{bmatrix} x_0^T & u_0^T & x_1^T & u_1^T & \cdots & u_{N-1}^T & x_N^T \end{bmatrix}^T, \tag{5}$$

the optimal control problem is recast in the form of the quadratic program

$$\min_d \frac{1}{2} d^T H d + h^T d \quad \text{subject to } Fd = f,\ Gd \le g, \tag{6}$$

where $d, h \in \mathbb{R}^{n_h}$, $H \in \mathbb{R}^{n_h \times n_h}$, $F \in \mathbb{R}^{n_f \times n_h}$, $G \in \mathbb{R}^{n_g \times n_h}$, $n_h = (n_x + n_u)N + n_x$, $n_f = n_x(N+1)$, and $n_g = \sum_{i=0}^{N} n_{c_i}$. The sparse matrices H, F, and G and the vectors h, f, and g are as follows:

$$H = \begin{bmatrix} Q_0 & S_0^T & \cdots & 0 & 0 & 0 \\ S_0 & R_0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & Q_{N-1} & S_{N-1}^T & 0 \\ 0 & 0 & \cdots & S_{N-1} & R_{N-1} & 0 \\ 0 & 0 & \cdots & 0 & 0 & Q_N \end{bmatrix},$$

$$F = \begin{bmatrix} -I & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ A_0 & B_0 & -I & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & A_{N-1} & B_{N-1} & -I \end{bmatrix},$$

$$G = \begin{bmatrix} G_0^x & G_0^u & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & G_{N-1}^x & G_{N-1}^u & 0 \\ 0 & 0 & \cdots & 0 & 0 & G_N^x \end{bmatrix},$$

$$h = \begin{bmatrix} q_0 \\ r_0 \\ \vdots \\ r_{N-1} \\ q_N \end{bmatrix}, \quad f = -\begin{bmatrix} \hat{x}_0 \\ a_0 \\ \vdots \\ a_{N-1} \end{bmatrix}, \quad g = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_N \end{bmatrix}.$$

We derive algebraic optimality conditions by means of the Lagrangian function with a logarithmic barrier

$$\mathcal{L} = \frac{1}{2} d^T H d + h^T d + y^T (Fd - f) + z^T (Gd - g + s) - \mu \mathbf{1}^T \log s,$$

where $\mathbf{1} = [1 \ldots 1]^T \in \mathbb{R}^{n_g}$, $y \in \mathbb{R}^{n_f}$ and $z \in \mathbb{R}^{n_g}$ are the Lagrange multipliers. The positive slack variables $s \in \mathbb{R}^{n_g}$ and the small parameter $\mu > 0$ are required by the infeasible-start interior-point method [6], [25]. The first order optimality (KKT) conditions are as follows

$$Hd + F^T y + G^T z + h = 0, \tag{7}$$

$$Fd - f = 0, \quad Gd - g + s = 0, \tag{8}$$

$$Z\mathbf{1} - \mu S^{-1}\mathbf{1} = 0, \tag{9}$$

where the matrices $Z = \operatorname{diag}(z)$ and $S = \operatorname{diag}(s)$ are diagonal. Equation (9) is nonlinear and commonly substituted with the equation $ZS\mathbf{1} - \mu\mathbf{1} = 0$, which is better scaled for the residuals in Newton's method.

A search direction at iteration k of Newton's method is determined by the system of linear equations

$$\underbrace{\begin{bmatrix} H & F^T & G^T & 0 \\ F & 0 & 0 & 0 \\ G & 0 & 0 & I \\ 0 & 0 & S^k & Z^k \end{bmatrix}}_{\mathcal{A}_0^k} \begin{bmatrix} \Delta d^k \\ \Delta y^k \\ \Delta z^k \\ \Delta s^k \end{bmatrix} = -\underbrace{\begin{bmatrix} r_H^k \\ r_F^k \\ r_G^k \\ r_S^k \end{bmatrix}}_{b_0^k} \tag{10}$$

with the residuals

$$r_H^k = Hd^k + F^T y^k + G^T z^k + h, \tag{11}$$

$$r_F^k = Fd^k - f, \tag{12}$$

$$r_G^k = Gd^k - g + s^k, \tag{13}$$

$$r_S^k = Z^k S^k - \sigma\mu^k \mathbf{1}, \tag{14}$$

where the scalar $\sigma \in (0, 1)$ is called a centering parameter and the current value for $\mu^k = (z^k)^T s^k / n_g$ is related to the duality gap [6], [11]. Several algorithms use Mehrotra's variant [5], [7] of the interior point iteration, whose practical efficiency on linear and quadratic programming problems is well documented. When some inequality constraints are active, the matrix $\mathcal{A}_0^k$ becomes increasingly ill-conditioned. Nevertheless, the interior point method successfully computes the solution, when (10) is solved by direct methods. Such an effect is discussed and justified in [26].

III. Reduction by block Gaussian elimination

The system (10) can be solved by modern algorithms for solving systems of linear equations with sparse matrices, which take advantage of the Gauss elimination method with special pivoting, i.e., with permutations of unknown variables

and equations. Note that these permutations are the same at all iterations of the IP method and can be computed only once before starting the IP iterations, for example, as in [9], [13], [27]. Alternatively, it may be efficient to eliminate the zero blocks and the diagonal blocks I, Z^k and S^k in $\mathcal{A}_0^k$ explicitly by block Gaussian eliminations and then to solve the resulting systems by specialized methods. We choose the latter alternative and present several special direct methods for the reduced linear systems below.

The first reduction is the elimination of Δs^k from (10), which yields the following linear system of smaller size

$$\underbrace{\begin{bmatrix} H & F^T & G^T \\ F & 0 & 0 \\ G & 0 & -W^k \end{bmatrix}}_{\mathcal{A}_1^k} \begin{bmatrix} \Delta d^k \\ \Delta y^k \\ \Delta z^k \end{bmatrix} = \underbrace{-\begin{bmatrix} r_H^k \\ r_F^k \\ r_W^k \end{bmatrix}}_{b_1^k}, \tag{15}$$

$$\Delta s^k = -(Z^k)^{-1}\left(r_S^k + S^k \Delta z^k\right), \tag{16}$$

with the diagonal matrix $W^k = (Z^k)^{-1}S^k$ and the residual $r_W^k = r_G^k - (Z^k)^{-1}r_S^k$.

The second reduction is the elimination of Δz^k from system (15), which results in the 2×2 block system of linear equations

$$\underbrace{\begin{bmatrix} H + G^T(W^k)^{-1}G & F^T \\ F & 0 \end{bmatrix}}_{\mathcal{A}_2^k} \begin{bmatrix} \Delta d^k \\ \Delta y^k \end{bmatrix} = \underbrace{-\begin{bmatrix} r_E^k \\ r_F^k \end{bmatrix}}_{b_2^k}, \tag{17}$$

$$\Delta z^k = (W^k)^{-1}\left(G\Delta d^k + r_W^k\right), \tag{18}$$

where $r_E^k = r_H^k + G^T(W^k)^{-1}r_W^k$.

It is important to observe here that system (17) can be obtained as KKT conditions for a QP without inequality constraints but with the modified cost matrices

$$\tilde{Q}_i = Q_i^k = Q_i + (G_i^x)^T(W^k)^{-1}G_i^x, \tag{19}$$

$$\tilde{R}_i = R_i^k = R_i + (G_i^u)^T(W^k)^{-1}G_i^u, \tag{20}$$

$$\tilde{S}_i = S_i^k = S_i + (G_i^u)^T(W^k)^{-1}G_i^x. \tag{21}$$

Thus, at any iteration k, the system of linear equations (17) can be rewritten in the form

$$\underbrace{\begin{bmatrix} \tilde{H} & F^T \\ F & 0 \end{bmatrix}}_{\mathcal{A}_3^k} \begin{bmatrix} \Delta d^k \\ \Delta y^k \end{bmatrix} = \underbrace{-\begin{bmatrix} r_E^k \\ r_F^k \end{bmatrix}}_{b_3^k}, \tag{22}$$

where $\tilde{H}$ possesses the block structure of H with the blocks Q_i, R_i, and S_i substituted by $\tilde{Q}_i$, $\tilde{R}_i$, and $\tilde{S}_i$ from (19)–(21).

Arithmetic complexity of (19)–(21) depends on the sparsity of the matrices G_i^x and G_i^u. For example, the box constraints $\underline{d}_j \le d_j \le \bar{d}_j$ on the decision variables result in the diagonal matrices $(G_i^x)^T(W^k)^{-1}G_i^x$, $(G_i^u)^T(W^k)^{-1}G_i^u$, and $(G_i^u)^T(W^k)^{-1}G_i^x$.

IV. Solution by the Riccati recursion

Let Π be the permutation matrix such that

$$\Delta v^k = \Pi^T\left[(\Delta d^k)^T \quad (\Delta y^k)^T\right]^T = \Delta[y_0^T, x_0^T, u_0^T, y_1^T, \ldots, y_{N-1}^T, x_{N-1}^T, u_{N-1}^T, y_N^T, x_N^T]^T.$$

Linear system (22) is equivalent to the system of linear equations $(\Pi\mathcal{A}_3^k\Pi)\Delta v^k = \Pi b_3^k$, where the symmetric matrix $\mathfrak{B} = \Pi\mathcal{A}_3^k\Pi$ is banded and has the block structure

$$\mathfrak{B} = \begin{bmatrix} 0 & -I & & & & & & & & \\ -I & \tilde{Q}_0 & \tilde{S}_0^T & A_0^T & & & & & & \\ & \tilde{S}_0 & \tilde{R}_0 & B_0^T & & & & & & \\ & A_0 & B_0 & 0 & -I & & & & & \\ & & & & \ddots & & & & & \\ & & & & -I & \tilde{Q}_{N-1} & \tilde{S}_{N-1}^T & A_{N-1}^T & & \\ & & & & & \tilde{S}_{N-1} & \tilde{R}_{N-1} & B_{N-1}^T & & \\ & & & & & A_{N-1} & B_{N-1} & 0 & -I & \\ & & & & & & & -I & \tilde{Q}_N \end{bmatrix}.$$

The Riccati recursion is a special upward Gaussian elimination method without pivoting applied to the system of linear equations $\mathfrak{B}\Delta v^k = \Pi b_3^k$. The Gaussian elimination in the Riccati recursion is aimed at obtaining equations of the form $-y_i + P_i x_i = \beta_i$, $i = N, \ldots, 0$, and implemented by the aid of the Schur complement at the block $\tilde{Q}_i$ for the block matrices

$$\begin{bmatrix} \tilde{Q}_i & \tilde{S}_i^T & A_i^T & 0 \\ \tilde{S}_i & \tilde{R}_i & B_i^T & 0 \\ A_i & B_i & 0 & -I \\ 0 & 0 & -I & P_{i+1} \end{bmatrix}.$$

The Schur complement matrices P_i are symmetric owing to the symmetry of $\mathfrak{B}$ and $P_N = \tilde{Q}_N$. The Riccati recursion does not fail if all $\tilde{R}_i + B_i^T P_{i+1} B_i$ are nonsingular. Since

$$Z_i = \begin{bmatrix} \tilde{R}_i & B_i^T & 0 \\ B_i & 0 & -I \\ 0 & -I & P_{i+1} \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -P_{i+1} & -I \\ 0 & -I & 0 \end{bmatrix} + \begin{bmatrix} I \\ P_{i+1}B_i \\ B_i \end{bmatrix} (\tilde{R}_i + B_i^T P_{i+1} B_i)^{-1} \begin{bmatrix} I & B_i^T P_{i+1} & B_i^T \end{bmatrix},$$

the matrices P_i, $i = N-1, \ldots, 0$, satisfy the matrix equations

$$P_i = \tilde{Q}_i - \begin{bmatrix} \tilde{S}_i^T & A_i^T & 0 \end{bmatrix} Z_i \begin{bmatrix} \tilde{S}_i \\ A_i \\ 0 \end{bmatrix} = \tilde{Q}_i + A_i^T P_{i+1} A_i - M_i^T (\tilde{R}_i + B_i^T P_{i+1} B_i)^{-1} M_i, \tag{23}$$

where

$$M_i = \tilde{S}_i + B_i^T P_{i+1} A_i. \tag{24}$$

Thus, the matrix P_i is recursively computed as the right-hand side of (23), which is called the *classical Riccati recursion* in [21]. We denote by $\mathrm{chol}(A)$ the upper triangular Cholesky factor of a symmetric positive semidefinite matrix A. Let us compute P_i by the sequence of matrix operations $t_A = P_{i+1}A_i$, $t_B = P_{i+1}B_i$, $M_i = \tilde{S}_i + B_i^T t_A$, $t_R = \mathrm{chol}(\tilde{R}_i + B_i^T t_B)$, $t_M = t_R^{-1}M_i$, $P_i = \tilde{Q}_i + A_i^T t_A - t_M^T t_M$. These are BLAS3 operations except for chol(). Their arithmetic complexity respectively equal $2n_x^3$, $2n_x^2 n_u$, $2n_x^2 n_u$, $2n_x n_u^2 + n_u^3/3$, $n_x n_u^2$, $2n_x^3 + 2n_x^2 n_u$, cf. [21]. Therefore,

the total arithmetic cost of computing P_i for $i = N, \ldots, 0$ in the classical Riccati recursion is

$$N\left(4n_x^3 + 6n_x^2 n_u + 3n_x n_u^2 + n_u^3/3\right). \tag{25}$$

The authors of [21] propose to compute P_i via its Cholesky factor $U_i = \text{chol}(P_i)$ and call it the *factorized Riccati recursion*. Namely, the upper triangular U_i is computed by the BLAS3 matrix operations and chol() as $t_A = U_{i+1}A_i$, $t_B = U_{i+1}B_i$, $M_i = \tilde{S}_i + t_B^T t_A$, $t_R = \text{chol}(\tilde{R}_i + t_B^T t_B)$, $t_M = t_R^{-1} M_i$, $U_i = \text{chol}(\tilde{Q}_i + t_A^T t_A - t_M^T t_M)$. The arithmetic costs are, respectively, n_x^3, $n_x^2 n_u$, $2n_x^2 n_u$, $n_x n_u^2 + n_u^3/3$, $n_x n_u^2$, $n_x^3 + n_x^2 n_u + n_x^3/3$. The total arithmetic cost of the factorized Riccati recursion equals

$$N\left(\frac{7}{3}n_x^3 + 4n_x^2 n_u + 2n_x n_u^2 + \frac{1}{3}n_u^3\right), \tag{26}$$

where the terms, quadratic or linear in n_x and n_u, as well as those independent of N are dropped from the cost. Formulas (25) and (26) have been derived, e.g., in [21], where the arithmetic cost of obtaining the solution vector Δv^k is also derived for both versions of the Riccati recursion. The computation of Δv^k consists of BLAS2 operations but has a lower order arithmetic complexity than that of computing the matrices P_i, and hence it is negligible in the total arithmetic cost.

V. Schur complement based solution

This section is devoted to the system of linear equations (22). We recall that the block $\tilde{H}$ is block diagonal and has N blocks of order $n_x + n_u$ and one block of order n_x on the main diagonal. If $\tilde{H}$ is positive definite, then system (22) reduces to the system of linear equations with respect to Δy^k

$$\begin{aligned}\left(F\tilde{H}^{-1}F^T\right)\Delta y^k &= r_F^k - F\tilde{H}^{-1}r_E^k, \\ \Delta d^k &= -\tilde{H}^{-1}\left(F^T\Delta y^k + r_E^k\right).\end{aligned} \tag{27}$$

System (27) is sometimes called the normal equations, e.g., in [6]. The positive definite matrix $F\tilde{H}^{-1}F^T$ is block tridiagonal and, therefore, (27) can be solved by means of the Cholesky factorization. This approach was proposed for MPC and implemented in [10]. The authors of [21] refer to this approach as the *Schur complement based solution*. The arithmetic complexity of the method is

$$N\left(\frac{19}{3}n_x^3 + 4n_x^2 n_u + 2n_x n_u^2 + \frac{1}{3}n_u^3\right). \tag{28}$$

Formula (28) takes into account both the computation of the matrix $F\tilde{H}^{-1}F^T$ and its Cholesky factorization. The computational cost in (28) can be further reduced in the common case, where $\tilde{H}$ is a diagonal matrix, as discussed in [14], to $N\left(\frac{10}{3}n_x^3 + n_x^2 n_u\right)$.

If the matrix $\tilde{H}$ is singular, then the normal equations (27) can not be formed. The following simple regularization, where the matrix $\tilde{H} + \tau I$ with a small positive parameter τ substitutes for $\tilde{H}$ in (27) repairs the method. Next, we propose a somewhat similar regularization for system (22).

VI. Solution based on the augmented Lagrangian regularization

We are interested in numerically solving the system of linear equations (22). The solution is then used for Newton's step in the interior point iteration. It is well known, see, e.g., [4], that the KKT matrix of Newton's method can be approximated. This observation gives rise to the so-called inexact Newton methods [4].

We follow this idea of approximating the KKT matrix in (22) and solve the system of linear equations

$$\underbrace{\begin{bmatrix}\tilde{H} & F^T \\ F & -\tau I\end{bmatrix}}_{\mathcal{A}_4^k}\begin{bmatrix}\Delta d^k \\ \Delta y^k\end{bmatrix} = \underbrace{-\begin{bmatrix}r_E^k \\ r_F^k\end{bmatrix}}_{b_4^k}, \tag{29}$$

resulting in an inexact Newton iteration, instead of the exact Newton step in (22). The positive regularization parameter τ must be sufficiently small. The regularized matrices of the form $\mathcal{A}_4^k$ are often used in iterative methods for numerical solution of saddle-point problems, especially for purposes of preconditioning [22], [23]. This regularization is tightly connected with the augmented Lagrangian method in numerical optimization, see, for example, [4].

The linear system in (29) is equivalently transformed into the block triangular system

$$\begin{bmatrix}\tilde{H} + \frac{1}{\tau}F^T F & 0 \\ F & -\tau I\end{bmatrix}\begin{bmatrix}\Delta d^k \\ \Delta y^k\end{bmatrix} = -\begin{bmatrix}r_E^k + \frac{1}{\tau}F^T r_F^k \\ r_F^k\end{bmatrix}. \tag{30}$$

The solution to (30) is given by the formulas

$$\begin{aligned}\Delta d^k &= -(\tilde{H} + \frac{1}{\tau}F^T F)^{-1}(r_E^k + \frac{1}{\tau}F^T r_F^k), \\ \Delta y^k &= \frac{1}{\tau}(r_F^k + F\Delta d^k).\end{aligned} \tag{31}$$

The main computational burden is solving the system with the symmetric positive definite matrix $\tilde{H} + \frac{1}{\tau}F^T F$ in (31). We recall that this matrix depends on k and equals

$$H + \frac{1}{\tau}F^T F + G^T (W^k)^{-1} G. \tag{32}$$

Note that the constant part $H + \frac{1}{\tau}F^T F$ of the matrix (32) can be computed only once before the IP iterations. The most time consuming part of the product $F^T F$ consists of N matrix products $\begin{bmatrix}A_i & B_i\end{bmatrix}^T\begin{bmatrix}A_i & B_i\end{bmatrix}$, $i = 0, \ldots, N-1$. The total arithmetic cost of all the products is $N(n_x + n_u)^2 n_x$.

Let us introduce the matrices

$$\begin{aligned}\tilde{\tilde{Q}}_i &= Q_i^k + \frac{1}{\tau}(A_i^T A_i + I), \\ \tilde{\tilde{R}}_i &= R_i^k + \frac{1}{\tau}B_i^T B_i, \quad \tilde{\tilde{S}}_i = S_i^k + \frac{1}{\tau}B_i^T A_i,\end{aligned}$$

where the matrices Q_i^k, R_i^k, and S_i^k are defined in (19)–(21).

The matrix (32) has the following block-tridiagonal structure

$$\begin{bmatrix} \tilde{\tilde{Q}}_0 & \tilde{\tilde{S}}_0^T & -\frac{1}{\tau}A_0^T & & & & \\ \tilde{\tilde{S}}_0 & \tilde{\tilde{R}}_0 & -\frac{1}{\tau}B_0^T & & & & \\ -\frac{1}{\tau}A_0 & -\frac{1}{\tau}B_0 & \tilde{\tilde{Q}}_1 & \tilde{\tilde{S}}_1^T & -\frac{1}{\tau}A_1^T & & \\ & & \tilde{\tilde{S}}_1 & \tilde{\tilde{R}}_1 & -\frac{1}{\tau}B_1^T & & \\ & & -\frac{1}{\tau}A_1 & -\frac{1}{\tau}B_1 & \tilde{\tilde{Q}}_2 & & \\ & & & & & \ddots & \\ & & & & & & \tilde{\tilde{Q}}_N \end{bmatrix}.$$

The Cholesky factorization of this matrix is computed by means of the Cholesky factorization for the submatrices

$$\begin{bmatrix} \tilde{\tilde{Q}}_i & \tilde{\tilde{S}}_i^T & -\frac{1}{\tau}A_i^T \\ \tilde{\tilde{S}}_i & \tilde{\tilde{R}}_i & -\frac{1}{\tau}B_i^T \\ -\frac{1}{\tau}A_i & -\frac{1}{\tau}B_i & \tilde{\tilde{Q}}_{i+1} \end{bmatrix}.$$

The factorization of the last corner block serves as the factorization of the first corner block in the next submatrix.

It is easy to calculate that the total arithmetic complexity of the Cholesky factorization for (32) equals

$$N\left(\frac{7}{3}n_x^3 + 4n_x^2 n_u + 2n_x n_u^2 + \frac{1}{3}n_u^3\right). \tag{33}$$

This cost coincides with that in (26), which means that the factorized Riccati recursion and the augmented Lagrangian regularization (ALR) methods are best of the considered four methods in terms of the arithmetic cost. The factorized Riccati recursion is a parameter-free method while the ALR method depends on a regularization parameter $\tau > 0$.

VII. Case study: chain of oscillating masses

The test problem consists of the chain of $n/2$ equal masses of value 1 connected by springs and to walls at the ends. The stiffness of all springs equals 1, and there is no damping. The continuous-time equations of the spring-mass system are given by the differential equations

$$\begin{aligned} \ddot{q}_1 &= -2q_1 + q_2 + f_1, \\ \ddot{q}_2 &= q_1 - 2q_2 + q_3 + f_2, \\ &\vdots \\ \ddot{q}_m &= q_{m-1} - 2q_m + q_{m+1} + f_m, \\ \ddot{q}_{m+1} &= q_m - 2q_{m+1} + q_{m+2}, \\ &\vdots \\ \ddot{q}_{n/2} &= q_{n/2-1} - 2q_{n/2}, \end{aligned}$$

where q_i is the coordinate of the ith mass with respect to its equilibrium position and f_i represents the control force acting on the ith mass. There are m actuators connected to the first m masses. The state-space form of the linear system is determined by the vectors of the state, control, and output, respectively,

$$x = \begin{bmatrix} q_1^T & q_2^T & \dots & q_{n/2}^T & \dot{q}_1^T & \dot{q}_2^T & \dots & \dot{q}_{n/2}^T \end{bmatrix}^T,$$
$$u = \begin{bmatrix} f_1^T & f_2^T & \dots & f_m^T \end{bmatrix}^T,\ y = \begin{bmatrix} q_1^T & q_2^T & \dots & q_{n/2}^T \end{bmatrix}^T.$$

The inputs and outputs are subject to the following inequality constraints

$$\begin{aligned} -0.5 \le u(i) \le 0.5, &\qquad i = 0, \dots, N-1, \\ -3.5 \le y(i) \le 3.5, &\qquad i = 1, \dots, N. \end{aligned}$$

The continuous-time state-space system is discretized using a sample time of $\Delta\tau = 0.5$ while keeping the inputs constant between sample instants. The discrete-time dynamics are $x_{i+1} = A_i x_i + B_i u_i$, where

$$A_i = \exp(\Delta\tau A_c), \quad B_i = A_c^{-1}(A_i - I_n)\begin{bmatrix} 0_{n/2} \\ I_m \\ 0_{n/2-m} \end{bmatrix}$$

and

$$A_c = \begin{bmatrix} 0_{n/2} & I_{n/2} \\ T_{n/2} & 0_{n/2} \end{bmatrix}, \quad T = \begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 \end{bmatrix}.$$

We use the following regulator tuning matrices $R_i = 10^{-6}I$, $S_i = 0$, and $Q_i = C^T C = \begin{bmatrix} I_{n/2} & 0 \end{bmatrix}^T \begin{bmatrix} I_{n/2} & 0 \end{bmatrix}$ and the initial values

$$\bar{x} = 3.5\begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \end{bmatrix}^T.$$

Numerical experiments have been carried out in MATLAB with the test problem, where the dimension of the state is $n = 12$, the number of control inputs is $m = 3$, the horizon length is $N = 30$. The number of inequality constraints at each time instance equals $n + 2m = 18$. Our MATLAB code implements the interior point method using the direct solver for the system of linear equations (29) described in Section VI. Other constants used in the IP iterations and in the stopping criteria, see [25], include $\sigma = 0.1$, $\gamma = 10^{-3}$, $\beta = 2$, $\epsilon = 10^{-6}$. We report only the results of solving MPC at the initial time from a cold start, determined by the initial values of x, u, y, z, s equal to $\mathbf{1}$.

For comparison, we have run the IP iterations, where the linear system (22) is solved exactly by the backslash operator in MATLAB, which implements the Gauss method with partial pivoting. Our program using the backslash operator for solving (22) returns a solution of QP after 17 IP iterations and with the 2-norm of the IP residual $\|[r_H^T, r_F^T, r_G^T, r_S^T]^T\|_2 = 1.76 \cdot 10^{-5}$.

The solver for (29) needs the small regularization parameter $\tau > 0$. Smaller values of τ give better approximation to the KKT matrix but a too small value of τ may lead to extremely large condition numbers of the matrices (32), for which the algorithm breaks down.

We have experimented with 100 values of τ uniformly distributed between $3 \cdot 10^{-3}$ and $3 \cdot 10^{-14}$ in the logarithmic scale. Note that our code does not work outside these bounds. The stopping criterion for the IP method is taken from [25]. Figure 1 displays the number of interior-point iterations with respect to τ, and Figure 2 displays the 2-norm of the interior-point residual with respect to τ. The numerical results for τ between 10^{-11} and 10^{-4} coincide with those

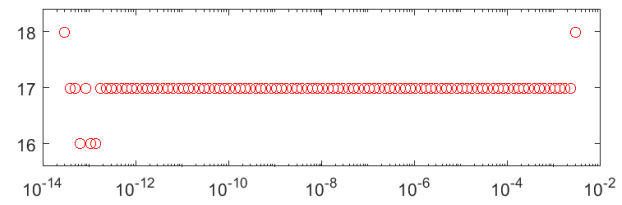

Fig. 1. The number of interior-point iterations with respect to τ.

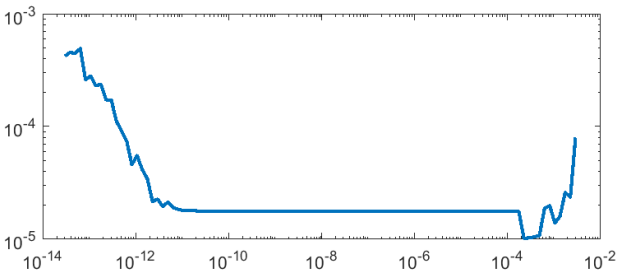

Fig. 2. The 2-norm of the interior-point residual with respect to τ.

for the backslash operator, i.e., the number of interior-point iterations equals 17 and the 2-norm of the interior-point residual approximately equals $1.76 \cdot 10^{-5}$.

The experiment with the 100 values of the regularization parameter τ has been repeated for the dimension $n = 240$ of the state, other parameters of the test problem were the same as above. The number of interior-point iterations again equals 17 for all τ at the interval $[10^{-12}, 10^{-4}]$. The corresponding 2-norm of the interior-point residuals for $\tau \in [10^{-12}, 10^{-4}]$ is about $3.2 \cdot 10^{-5}$.

VIII. Conclusion

We have analyzed four direct solvers for computation of the Newton step in the interior-point method applied to the linear MPC problem. The solvers include the classical and factorized Riccati recursions, the Schur complement based solution of the normal equations, and a novel solver based on the technique of the augmented Lagrangian regularization.

The complexity analysis reveals that the factorized Riccati recursion and the augmented Lagrangian regularization solver have the same arithmetic cost $N\left(\frac{7}{3}n_x^3 + 4n_x^2 n_u + 2n_x n_u^2 + \frac{1}{3}n_u^3\right)$ and are more efficient than the other two solvers. The most computationally expensive parts of all considered direct solvers have cubic complexity with respect to n_x and n_u, however, they can be implemented in terms of highly efficient BLAS3 operations and compete with methods having quadratic complexity.

Our future work concerns developing efficient preconditioning with quadratic complexity for the proposed regularized Newton's method, which would allow outperforming the solvers based on the factorized Riccati recursion.

References

[1] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control. Theory and Design*. Madison, WI: Nob Hill Publishing, 2013.

[2] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear MPC and moving horizon estimation," in *Nonlinear Model Predictive Control*, ser. LNCIS 384, L. M. et al., Ed. Heidelberg, Germany: Springer, 2009, pp. 391–417.

[3] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. Wiley, 1987.

[4] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[5] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM J. Optim.*, vol. 2, no. 4, pp. 575–601, 1992.

[6] S. J. Wright, *Primal-Dual Interior Point Methods*. Philadelphia: SIAM, 1997.

[7] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," *J. Optimiz. Theory Appl.*, vol. 99, pp. 723–757, 1998.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[9] A. Wähter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program., Ser. A*, vol. 106, pp. 25–57, 2006.

[10] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[11] J. Gondzio, "Interior point methods 25 years later," *European J. Operational Research*, vol. 218, pp. 587–601, 2012.

[12] ——, "Convergence analysis of an inexact feasible interior point method for convex quadratic programming," *SIAM J. Optim.*, vol. 23, no. 3, pp. 1510–1527, 2013.

[13] R. Bartlett, A. Wächter, and L. Biegler, "Active set vs. interior point strategies for model predictive control," in *Proc. American Control Conf.*, Chicago, IL, 2000, pp. 4229–4233.

[14] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, "Efficient interior point methods for multistage problems arising in receding horizon control," in *Proc. 51st IEEE Conf. Decision and Control*, Maui, Hawaii, USA, 2012, pp. 668–674.

[15] G. Frison, H. B. Sorensen, B. Dammann, and J. B. Jørgensen, "High-performance small-scale solvers for linear model predictive control," in *Proc. European Control Conf. (ECC)*, France, 2014, pp. 128–133.

[16] P. Gill, G. Golub, W. Murray, and M. A. Saunders, "Methods for modifying matrix factorizations," *Math. Comp.*, vol. 28, no. 126, pp. 505–535, 1974.

[17] C. Kirches, H. G. Bock, J. P. Schlöder, and S. Sager, "A factorization with update procedures for a KKT matrix arising in direct optimal control," *Math. Prog. Comp.*, vol. 3, no. 4, pp. 319–348, 2011.

[18] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Math. Prog. Comp.*, vol. 6, pp. 327–363, 2014.

[19] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*. Philadelphia: SIAM, 1999.

[20] G. Frison, D. Kouzoupis, A. Zanelli, and M. Diehl, "BLASFEO: basic linear algebra subroutines for embedded optimization," arXiv:1704.02457, pp. 1–29, 2017.

[21] G. Frison and J. B. Jørgensen, "Efficient implementation of the Riccati recursion for solving linear-quadratic control problems," in *Proc. IEEE Multi-conference on Systems and Control*, India, 2013, pp. 1117–1122.

[22] M. Benzi and A. J. Wathen, *Some preconditioning techniques for saddle point problems*, ser. Mathematics in Industry, vol 13. Berlin, Heidelberg, Germany: Springer, 2008, pp. 195–211.

[23] T. Rees and C. Greif, "A preconditioner for linear systems arising from interior point optimization methods," *SIAM J. Scientific Computing*, vol. 29, no. 5, pp. 1992–2007, 2007.

[24] N.-Y. Chiang, R. Huang, and V. M. Zavala, "An augmented Lagrangian filter method for real-time embedded optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6110–6121, 2017.

[25] 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 J. Optim.*, vol. 22, no. 4, pp. 1369–1393, 2012.

[26] M. H. Wright, "Ill-conditioning and computational error in interior point methods for nonlinear programming," *SIAM J. Optim.*, vol. 9, no. 1, pp. 84–111, 1998.

[27] V. M. Zavala and L. T. Biegler, "Nonlinear programming strategies for state estimation and model predictive control," in *Nonlinear Model Predictive Control*, ser. LNCIS 384, L. M. et al., Ed. Heidelberg, Germany: Springer, 2009, pp. 419–432.