Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Efficient Solution of Equality Constrained Quadratic Programming Arising in Model Predictive Control

Giuliano Costantini * **Daniel Görges** *

* *Technische Universität Kaiserslatuern, Erwin-Schrödinger-Straße 12 67663 Kaiserslautern (e-mail: {costantini,görges}@eit-uni-kl.de).*

Abstract: Fast Model Predictive Control (Fast MPC) is a set of techniques which aim at reducing the complexity of solving receding horizon control optimization problems. One method consists in exploiting the structure induced by the system dynamics. This drastically reduces the complexity of the problem from cubic to linear dependence on the horizon length. Such approach is possible for the multi-stage MPC formulation which is characterized by having an almost separable structure, i.e. coupling occurs only between consecutive stages. This paper proposes a novel technique that efficiently solves general linear-quadratic optimal control problems and hence does not require a multi-stage formulation. The method also provides other advantages including: higher solution accuracy, warm starting, and approximate solutions by early termination. This is achieved by embedding the Riccati recursion in a Projected Conjugate Gradient (PCG) method.

Keywords: Fast Model Predictive Control, Equality Constrained Quadratic Programming, Optimal Control, Projected Conjugate Gradient Method, Riccati Recursion.

1. INTRODUCTION

Many nonlinear optimization algorithms reduce the original problem to a sequence of equality constrained quadratic programs. The solution of these problems can thus be regarded as the fundamental step of these algorithms. Since often this is also the most computationally demanding step, it is paramount to provide an efficient solution to it. Over the years both active set and interior point methods emerged as the standard choice to solve optimization problems arising in MPC (Borrelli, 2003). In these methods the resulting base step is the minimization of a quadratic cost function subject to the dynamics of the system. An efficient solution can be found when the cost function is separable across the stages, and hence the only coupling occurs in the system dynamics. To ensure this structure, the MPC problem should be formulated as multi-stage finite horizon optimal control problem (Domahidi et al., 2012). Problems with rate constraints or rate penalty weights can be readily recast into this form by conveniently introducing integrator dynamics. The resulting base problem is known as the extended linear-quadratic control problem (Jørgensen et al., 2012) and several efficient solutions have been proposed in the literature (Frison and Jørgensen, 2013b). These can be classified into three groups: i) state variable elimination, ii) Sparse Schur complement (Wang and Boyd, 2009; Domahidi et al., 2012), and iii) Riccati recursion (Rao et al., 1998; Frison and Jørgensen, 2013a). In the first category the problem is reformulated to include only the system input as optimization variables. This results in a dense problem with fewer variables but cubic complexity on the horizon length. The latter two categories use both state and input as decision variables, however, they differ in how the optimality conditions are handled. In ii) primal and dual optimality conditions are separately grouped and the Schur complement method is then applied to obtain a block-tridiagonal system. In iii) the variables are reordered to get a particular banded structure which can be solved using the Riccati recursion. In both cases the complexity is reduced to be linear in the horizon length. There exists also a mixed approach (Axehill and Morari, 2012) which exploits the structure in the dense formulation and results in a quadratic complexity.

However, these efficient methods can only be employed under a multi-stage separable structure assumption. There are many applications whose requirements impair this structure and induce coupling among different stages other than the system dynamics. Requirements such as rate constraints, minimum rise time, or trajectory monotonicity are all practical examples where the multi-stage formulation cannot be enforced. A survey of these constraints can be found in (Camacho and Bordons Alba, 2007, chap. 7). In some cases, as already mentioned, it is possible to augment the state of the system by adding additional integrator dynamics in order to transform consecutive stage constraints into simple stage constraints (Rawlings and Mayne, 2009, sec. 1.2.5). The drawback of this approach is a larger number of decision variables which increases linearly with the horizon length. Furthermore, constraints that couple all stages together cannot be practically handled with this approach. A simple example is to relax a stage constraint on some input or state variable and considering instead a constraint on its average value over the horizon length.

This paper proposes a new efficient approach for dense convex quadratic cost functions subject to the system dynamics. The proposed method is based on the Projected Gradient Method (Coleman, 1994; Gould et al., 2001). It will be shown that, under a suitable structured preconditioning, the projection step can be efficiently solved with the Riccati recursion or any other of the above mentioned techniques. For multi-stage formulations the proposed method can also be used to improve the accuracy of the solution or to provide an approximate solution by early termination. Finally this approach can take advantage from the fact that a sequence of similar problems needs to be solved and warm starting can be used.

The reminder of this paper is organized as follows. Section 2 gives an introduction on the problem formulation, including the multi-stage MPC and its efficient solution via the Riccati recursion. The multi-stage assumption is then removed in Section 3 where the proposed method for the solution of a general linear-quadratic control problem is described. Finally, simulation results are given in Section 4 and conclusions are drawn in Section 5.

2. PRELIMINARIES

2.1 Problem Formulation

Consider the following control optimization problem

$$\min \quad \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) \tag{1a}$$

$$\text{s.t.} \quad x_{k+1} = A_k x_k + B_k u_k, \qquad x_0 = \bar{x} \tag{1b}$$

$$\gamma_k(x_k, u_k) \leq 0, \quad \gamma_N(x_N) \leq 0 \tag{1c}$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the optimization variables at stage $k = 0 \dots N$ over the prediction horizon N. As usual the variables represent the state and input of the discrete time-varying linear system in (1b). The current state of the system $\bar{x}$ is assumed to be known either through measurement or observation.

The cost function and the inequality constraint at stage $k = 0 \dots N-1$ are

$$l_k(x_k, u_k) = \frac{1}{2} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \begin{bmatrix} Q_k & S_k^\top \\ S_k & R_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \begin{bmatrix} q_k \\ r_k \end{bmatrix}^\top \begin{bmatrix} x_k \\ u_k \end{bmatrix} \tag{2a}$$

$$\gamma_k(x_k, u_k) = \left\{ \begin{matrix} x_k \in \mathbb{R}^n \\ u_k \in \mathbb{R}^m \end{matrix} : \begin{bmatrix} G_k^x \\ G_k^u \end{bmatrix}^\top \begin{bmatrix} x_k \\ u_k \end{bmatrix} \leq g_k \right\}, \tag{2b}$$

and at the terminal stage

$$l_N(x_N) = \frac{1}{2} x_N^\top Q_N x_N + q_N^\top x_N \tag{3a}$$

$$\gamma_N(x_N) = \{ x_N \in \mathbb{R}^n : G_N^x x_N \leq g_N^x \}. \tag{3b}$$

It is assumed that the functions l_k for $k = 0 \dots N$ are strictly convex. This assumption is actually stronger than what required for existence of the solution but it allows faster computations.

The optimization problem here described is a typical *multi-stage linear MPC* instance and it can be solved with several optimization algorithms. As mentioned in the introduction, the solution process typically reduces to a sequence of highly structured quadratic programs with equality constraints. In conformity with Frison and Jørgensen (2013b) this special problem is called *Extended Linear-Quadratic Control Problem*. Such step will be also referred to as inner optimization to distinguish it from the original problem which is referred to as outer optimization.

The problem is stated in the following form [1]

$$\min \quad \sum_{k=0}^{N-1} \tilde{l}_k(x_k, u_k) + \tilde{l}_N(x_N) \tag{4a}$$

$$\text{s.t.} \quad x_{k+1} = A_k x_k + B_k u_k + \bar{x}_{k+1}, \quad x_0 = \bar{x}_0 \tag{4b}$$

where the stage and terminal cost are quadratic forms with similar structure as the original ones. The new data in the cost function is obtained according to the algorithm being used. For instance the affine step in interior point methods produces the following new stage data at each iteration

$$\begin{bmatrix} \tilde{Q}_k & \tilde{S}_k^\top \\ \tilde{S}_k & \tilde{R}_k \end{bmatrix} = \begin{bmatrix} Q_k & S_k^\top \\ S_k & R_k \end{bmatrix} + \begin{bmatrix} G_k^x \\ G_k^u \end{bmatrix} \Delta_k \begin{bmatrix} G_k^x \\ G_k^u \end{bmatrix}^\top \tag{5}$$

$$\begin{bmatrix} \tilde{q}_k \\ \tilde{r}_k \end{bmatrix} = \begin{bmatrix} \bar{q}_k \\ \bar{r}_k \end{bmatrix} - \begin{bmatrix} G_k^x \\ G_k^u \end{bmatrix} \Delta_k \, \bar{g}_k \quad . \tag{6}$$

The vectors $\bar{x}_k, \bar{q}_k, \bar{r}_k, \bar{g}_k$ are the current iteration residuals and the diagonal matrix Δ_k is constructed from the Lagrange multipliers and the slack variables associated with the corresponding inequality. A similar definition holds for the new terminal cost $\tilde{l}_N$.

2.2 Optimality Conditions and Riccati Recursion

The efficient solution of the inner optimization problem involves exploiting the multi-stage structure in both the cost function and the dynamic constraint. As mentioned this could be done in several ways, however, numerical experience indicates the Riccati recursion to be the best choice for the present framework. Since it serves as basis for the proposed algorithm, in what follows this approach is briefly reviewed. For a detailed discussion see Rao et al. (1998) and the references therein.

The Riccati recursion can be obtained either by using dynamic programming or by imposing the optimality conditions for the whole problem. Here the latter approach is presented. The dynamic constraint in problem (4) is relaxed by introducing for each stage a Lagrange multiplier λ_k and forming then the Lagrangian function

$$\mathcal{L} = \sum_{k=0}^{N-1} \tilde{l}_k(x_k, u_k) + \lambda_{k+1}^\top (A_k x_k + B_k u_k + \bar{x}_{k+1} - x_{k+1}) + \\ + \lambda_0^\top (\bar{x}_0 - x_0) + \tilde{l}_N(x_N).$$

The solution of problem (4) is retrieved by imposing the first-order optimality condition $\nabla \mathcal{L} = 0$ for each optimization variable. This leads to

$$\begin{aligned} -x_0 \quad &= -\bar{x}_0 \\ &\vdots \\ -\lambda_k + \tilde{Q}_k x_k + \tilde{S}_k^\top u_k + A_k^\top \lambda_{k+1} &= -\tilde{q}_k \qquad (7a) \\ \tilde{S}_k x_k + \tilde{R}_k u_k + B_k^\top \lambda_{k+1} &= -\tilde{r}_k \qquad (7b) \\ A_k x_k + B_k u_k - x_{k+1} &= -\bar{x}_{k+1} \qquad (7c) \\ &\vdots \\ -\lambda_N + \tilde{Q}_N x_N &= -\tilde{q}_N. \end{aligned}$$

[1] Many algorithms actually seek optimal directions δx_k and δu_k in the inner optimization. To avoid cluttering the notation the same optimization variables x_k and u_k as in the original problem are used.

Note that the resulting linear system of equations has a special block structure

$$\begin{bmatrix} & & & & & \\ \ddots & -I & & & & \\ -I & \tilde{Q}_k & \tilde{S}_k^\top & A_k^\top & & \\ & \tilde{S}_k & \tilde{R}_k^\top & B_k^\top & & \\ & A_k & B_k & 0 & -I & \\ & & & -I & \ddots & \end{bmatrix} \begin{bmatrix} \vdots \\ \lambda_k \\ x_k \\ u_k \\ \lambda_{k+1} \\ x_{k+1} \\ \vdots \end{bmatrix} = - \begin{bmatrix} \vdots \\ \bar{x}_k \\ \tilde{q}_k \\ \tilde{r}_k \\ \bar{x}_{k+1} \\ \tilde{q}_{k+1} \\ \vdots \end{bmatrix}. \quad (8)$$

It can be shown that it holds in general for stage k

$$\lambda_k = P_k x_k + p_k \quad (9a)$$

$$u_k = F_k x_k + f_k \quad (9b)$$

$$x_{k+1} = A x_k + B u_k + \bar{x}_{k+1}. \quad (9c)$$

Equation (9c) is simply the dynamics of the system. Note that, at the last stage, equation (9a) is readily verified by imposing $P_N = Q_N$ and $p_N = q_n$. Assume now that at stage $k{+}1$ it holds $\lambda_{k+1} = P_{k+1}x_{k+1}{+}p_{k+1}$. By eliminating the state x_{k+1} using (7c)

$$\begin{aligned} \lambda_{k+1} &= P_{k+1}x_{k+1} + p_{k+1} \\ &= P_{k+1}A_k x_k + P_{k+1}B_k u_k + P_{k+1}\bar{x}_{k+1} + p_{k+1} \end{aligned}$$

and substituting the result in (7a) and (7b), one obtains

$$-\lambda_k + \check{Q}_k x_k + \check{S}_k^\top u_k = -\check{q}_k \quad (10a)$$

$$\check{S}_k x_k + \check{R}_k u_k = -\check{r}_k \quad (10b)$$

with

$$\begin{aligned} \check{Q}_k &= \tilde{Q}_k + A_k^\top P_{k+1} A_k & \check{p}_{k+1} &= p_{k+1} + P_{k+1}\bar{x}_{k+1} \\ \check{S}_k &= \tilde{S}_k + B_k^\top P_{k+1} A_k & \check{q}_k &= \tilde{q}_k + A_k^\top \check{p}_{k+1} \\ \check{R}_k &= \tilde{R}_k + B_k^\top P_{k+1} B_k & \check{r}_k &= \tilde{r}_k + B_k^\top \check{p}_{k+1}. \end{aligned}$$

The affine state feedback law (9b) is then obtained from (10b) by left-multiplying by R_k^{-1} and setting

$$F_k = -\check{R}_k^{-1}\check{S}_k \qquad f_k = -\check{R}_k^{-1}\check{r}_k.$$

On the other hand, the multiplier λ_k in (9a) is recovered from (10a) by using the new expression for u_k and setting

$$P_k = \check{Q}_k + \check{S}_k^\top F_k \quad p_k = \check{q}_k + \check{S}_k^\top f_k.$$

The problem solution is finally obtained by starting from $x_0 = \bar{x}_0$ and alternatively computing u_k and x_{k+1} using (9b) and (9c). In the process, λ_k can also be computed with (9a) if required. In some optimization algorithms, for instance in Predictor-Corrector Interior Point methods, it might be necessary to find several solutions to (9) where only the right hand side differs. Therefore it might be reasonable to reorganize the recursion scheme in two steps. During the first, matrices P_k and F_k are computed by backward recursion. The solution is retrieved in the second step by performing a backward recursion to find p_k and f_k followed by a forward recursion to find x_k, u_k and possibly λ_k. These are summarized in the following implementation scheme which exploits the symmetry of the problem.

1: Factorization Step

init: $P_N = Q_n$,

/* backward */
for $k = N-1 \ \dots \ 0$
- $V = \text{chol}(P_{k+1})$
- $V_x = V A_k$
- $V_u = V B_k$
- $Q = \tilde{Q}_k + V_x^\top V_x$
- $S = \tilde{S}_k + V_x^\top V_u$
- $R = \tilde{R}_k + V_u^\top V_u$
- $U_k = \text{chol}(R)$
- $\check{F} = \text{solve}(U_k^\top, S_k)$
- $P_k = \check{Q}_k - \check{F}^\top \check{F}$
- $F_k = \text{solve}(U_k, -\check{F})$

end

2: Solve Step

init: $p_N = q_n$

/* backward */
for $k = N-1 \ \dots \ 0$
- $v = P_{k+1}\bar{x}_{k+1} + p_{k+1}$
- $q = \tilde{q}_k + A_k^\top v$
- $r = \tilde{r}_k + B_k^\top v$
- $p_k = q_k + F_k^\top r$
- $f = \text{solve}(U_k^\top, \check{r}_k)$
- $f_k = \text{solve}(U_k, -f)$

end

/* forward */
init: $x_0 = \bar{x}_0$
for $k = 1 \ \dots \ N-1$
- $\lambda_k = P_k x_k + p_k$
- $u_k = F_k x_k + f_k$
- $x_{k+1} = A_k x_k + B_k u_k + \bar{x}_{k+1}$

end
$\lambda_N = P_N x_N + p_N$

Note that only variables with iteration index k as subscript are actually stored. For a positive definite matrix $X \succ 0$, the notation $Z = chol(X)$ stands for an upper Cholesky factorization step which yields $X = Z^\top Z$ with Z upper triangular. For matrices X and Y, the notation $Z = \text{solve}(X, Y)$ stands for a linear solve step which yields Z satisfying the equation $XZ = Y$. The latter notation also holds when the right hand side matrix Y is replaced by the vector y. For a more efficient implementation the reader is refered to Frison and Jørgensen (2013a).

The solution obtained through these steps is highly efficient and depends only linearly on the horizon N. However this algorithm is based on the fundamental assumption that there is only a single dependence between consecutive stages (due to the dynamics). If this assumption is removed then the special block structure in (8) is lost.

3. EFFICIENT PROJECTED CONJUGATE GRADIENT METHOD

In this section, an efficient algorithm scheme is proposed which does not require the multi-stage structure in (4). Here the *General Linear-Quadratic Control Problem* is proposed

$$\begin{aligned} \min \quad & \frac{1}{2} z^\top H z + c^\top z \\ \text{s.t.} \quad & Az = b \end{aligned} \quad (11)$$

with possibly dense Hessian $H \succ 0$. The decision variable is composed of each stage variable

$$z^\top = [x_0^\top, u_0^\top, \ \dots \ , x_{N-1}^\top, u_{N-1}^\top, \ x_N^\top] \in \mathbb{R}^l$$

and the dynamics of the system is recast in matrix form

$$A = \begin{bmatrix} -I & 0 & & & & & & \\ A_0 & B_0 & -I & 0 & & & & \\ 0 & 0 & A_1 & B_1 & -I & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{N-1} & B_{N-1} \ -I \end{bmatrix}, \ b = - \begin{bmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_N \end{bmatrix}.$$

It is remarked that no assumption is made on the cost function structure.

To solve problem (11), a tailored projected conjugate gradient method (PCG) is employed

Projected Conjugate Gradient Method [2]

input $z^{(0)}$ s.t. $Az^{(0)} = b$

init $\rho^{(0)} = -(Hz^{(0)} + c)$

for $\nu = 0 \ \dots \ \nu_{\max}$

$\gamma^{(\nu)} = \Pi_A[\rho^{(\nu)}]$ /* projection onto null space of A */

$\tau^{(\nu)} = \langle \rho^{(\nu)},\ \gamma^{(\nu)} \rangle$

if $\tau \leq \varepsilon^2$ **stop**

$$\phi^{(\nu)} = \begin{cases} \gamma^{(\nu)} & \text{if } \nu = 0 \\ \gamma^{(\nu)} + \dfrac{\tau^{(\nu)}}{\tau^{(\nu-1)}}\, \rho^{(\nu)} & \text{if } \nu > 0 \end{cases}$$

$\alpha^{(\nu)} = \frac{\tau_\nu}{\langle \phi^{(\nu)},\ H\phi^{(\nu)} \rangle}$

$z^{(\nu+1)} = z^{(\nu)} + \alpha^{(\nu)} \phi^{(\nu)}$

$\rho^{(\nu+1)} = \rho^{(\nu)} - \alpha^{(\nu)} H\phi^{(\nu)}$

end

A good introduction to the PCG method for equality constrained quadratic programming is given in Gould et al. (2001). The conjugate gradient method, being an indirect method, requires several iterations before an accurate solution can be found. This is in contrast to the direct method used in the previous section. The idea of using indirect methods to solve MPC problems is far from being new, see for instance (Shahzad et al., 2010a,b). More recently, a structure-aware indirect method for MPC has been proposed in Malyshev et al. (2018) which exploits the condensing method in Frison and Jørgensen (2013b). In these references, different Krylov subspaces methods are used as indirect solver and several preconditioners are also suggested. The method proposed in this paper differs from these approaches in that the projection operation $\gamma = \Pi_A[\rho]$ is used to induce a stage separable strucure also in the presence of a dense Hessian H. As a result, the projection operation can be solved efficiently under structured preconditioners.

There are several advantages in using indirect methods: i) higher solution accuracy can be reached, ii) it is possible to provide an initial guess, iii) it can be stopped early to get an approximate solution. Since this method is used to solve a sequence of similar problems it makes sense to use the previous optimal solution to warm start the next computation. Approximate solution can be used within an Inexact Interior Point method as proposed in Shahzad et al. (2010b). The main disadvantage of indirect methods is that they may require longer to converge, especially when the problem is ill-conditioned. This point can be mitigated by using a suitable preconditioner. Several choices are proposed and discussed later on.

In the Projected Gradient Method the search direction ρ, which points opposite to the gradient of the cost function $Hz + c$, is projected onto the null space of A by means of the projector $\Pi_A[\cdot]$. This ensures that if the initial guess satisfies the equality constraint then all successive iterates lie within the affine space $\{z \in \mathbb{R}^l : Az = b\}$. Note that by construction the conjugate direction ϕ, which is actually used for updating z, also satisfies $A\phi = 0$. In addition the projector needs to satisfy

$$\langle \rho,\ \Pi_A[\rho] \rangle > 0 \qquad \forall \rho \in \mathbb{R}^l \setminus \{0\} \tag{12}$$

in order to ensure that $\gamma = \Pi_A[\rho]$ is a descent direction. Condition (12) basically requires that the negative gradient and the search direction form an acute angle and therefore there exists a small step in this direction which decreases the cost function.

To be able to fully exploit the structure in the equality constraint, the following projector is considered

$$\gamma = \Pi_A[\rho]: \quad \underset{\gamma}{\operatorname{argmin}}\ \frac{1}{2}\, \gamma^\top M \gamma - \rho^\top \gamma \quad \text{s.t.} \quad A\gamma = 0. \tag{13}$$

The matrix $M \succ 0$ works here as a preconditioner and it is chosen to have a block-diagonal structure with blocks

$$M_k = \begin{bmatrix} M_k^{xx} & M_k^{ux\top} \\ M_k^{ux} & M_k^{uu} \end{bmatrix} \succ 0. \tag{14}$$

This leads to

$$\underset{\gamma}{\operatorname{argmin}}\ \frac{1}{2} \sum_{k=0}^{N} \begin{bmatrix} \gamma_k^x \\ \gamma_k^u \end{bmatrix}^\top \begin{bmatrix} M_k^{xx} & M_k^{ux\top} \\ M_k^{ux} & M_k^{uu} \end{bmatrix} \begin{bmatrix} \gamma_k^x \\ \gamma_k^u \end{bmatrix} - \begin{bmatrix} \rho_k^x \\ \rho_k^u \end{bmatrix}^\top \begin{bmatrix} \gamma_k^x \\ \gamma_k^u \end{bmatrix}$$
$$\text{s.t.} \quad \gamma_{k+1}^x = A_k \gamma_k^x + B_k \gamma_k^u \qquad \gamma_0^x = 0 \tag{15}$$

with the partitions

$$\rho^\top = \left[\rho_0^{x\top}, \rho_0^{u\top}, \rho_1^{x\top}, \rho_1^{u\top}, \cdots, \rho_{N-1}^{x\top}, \rho_{N-1}^{u\top}, \rho_N^{x\top}\right]$$
$$\gamma^\top = \left[\gamma_0^{x\top}, \gamma_0^{u\top}, \gamma_1^{x\top}, \gamma_1^{u\top}, \cdots, \gamma_{N-1}^{x\top}, \gamma_{N-1}^{u\top}, \gamma_N^{x\top}\right].$$

The chosen projection problem has essentially the same structure as the problem in the previous section and therefore the same efficient Riccati recursion can be employed. Note that (12) is satisfied for any choice of $M \succ 0$. To see this, assume $\gamma^\star$ is the optimal solution to the projection problem and $\lambda^\star$ its associated Lagrange multiplier. The solution pair satisfies the first order optimality condition

$$\begin{bmatrix} M & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \gamma^\star \\ \lambda^\star \end{bmatrix} = \begin{bmatrix} \rho \\ 0 \end{bmatrix}.$$

Therefore $\rho = M\gamma^\star + A^\top \lambda^\star$ and

$$\begin{aligned} \langle \gamma^\star,\ \rho \rangle &= \langle \gamma^\star,\ M\gamma^\star \rangle + \langle \gamma^\star,\ A^\top \lambda^\star \rangle \\ &= \langle \gamma^\star,\ M\gamma^\star \rangle + \langle A\gamma^\star,\ \lambda^\star \rangle \\ &= \langle \gamma^\star,\ M\gamma^\star \rangle > 0 \end{aligned}$$

where feasibility of the solution $A\gamma^\star = 0$ was used. Note also that a less strict condition for (12) only requires M to be positive definite on the null space of A. However, as mentioned, condition (14) brings some computational advantage. An important property of solving the projection problem using the Riccati recursion is that $A\gamma^\star = 0$ is delivered with a very high accuracy thanks to the forward solution step. Therefore the proposed algorithm is not affected by the problem arising in Gould et al. (2001) where several projection steps per iteration are required to reduce the residual to an acceptable level.

Regarding the implementation scheme, since the preconditioner M_k does not change throughout the iterations of PCG, it suffices to factor the matrices only once at the first iteration. Afterwards, only solution steps with different right hand sides are computed. It is also remarked that the equality constraint equation for the preconditioner has

[2] To avoid cluttering the notation with double superscript, the scalar product is here denoted by $\langle v,\ w \rangle$ instead of $v^\top w$

no right hand side. This leads to further simplify some operations during the solve step of the Riccati recursion.

Now let $(z^{\circledast}, \lambda^{\circledast})$ be the optimal solution pair to problem (11). It can be shown that the Lagrange multiplier $\lambda^{\star}$ for the projection problem converges to $\lambda^{\circledast}$ independently from which M is chosen. This follows directly from the projection optimality condition

$$M\gamma^{\star} + A^{\top}\lambda^{\star} = \rho$$

and the fact that $\gamma^{\star} \to 0$ and $\rho \to -(Hz^{\circledast} + c)$. Therefore computation of $\lambda^{\star}$ in the forward Riccati recursion is not required at each iteration step of PCG but only when the method converges. This reduces further the number of required operations.

In summary the proposed method combines the efficiency of the solution obtained by direct methods with the advantages that indirect methods bring. Furthermore the structure of the system dynamics can be exploited also in the case of a general positive definite Hessian H. However the effectiveness of the method still depends on the problem conditioning. For instance this is particularly critical during the last iterations of Interior Point Methods, where some of the elements of Δ_k goes to zero while other grows to infinity. In this case preconditioning can drastically decrease the number of iterations required for convergence.

3.1 Preconditioning

In this section three standard preconditioning techniques are initially presented. First consider the simple choice $M = I$. This has the advantage that the factorization step can be calculated offline since it only involves matrices A_k and B_k. Overall this choice is rather effective when the problem is well-conditioned. However an accurate solution may still require many iterations. The second choice is to select M as the diagonal of H. In this case the factorization should be run each time H changes, for instance at the beginning of each interior point iteration. This choice leads to a better behaviour of the proposed method but requires some additional computations. Finally the best performance is obtained by selecting M to be the block-diagonal of H ignoring any stage coupling. When the latter is block-diagonal then PCG can be terminated at the first iteration since this is exactly the solution obtained using the standard Riccati recursion. If required, a better solution can be obtained by additional iterations. This is particularly useful for ill-conditioned problems. If H is a dense matrix then the approximation provided by the block-diagonal still leads to an effective solution with PCG. Note that in this case the standard Riccati recursion cannot be used. Before concluding this section, another special class of preconditioners is introduced. The following block-diagonal structure is considered

$$M_k = \begin{bmatrix} 0 & 0 \\ 0 & M_k^{uu} \end{bmatrix}$$

with $M_k^{uu} \succ 0$. The advantage in using this type of preconditioners is that no factorization step is required at all. This can be seen from the optimality conditons (7) adapted to this special preconditioner problem.

Backward	Forward
$\lambda_N = \rho_N^x$	$\gamma_0^x = 0$
$\lambda_k = A_k^{\top}\lambda_{k+1} + \rho_k^x$	$\gamma_k^u = (M_k^{uu})^{-1}(B_k^{\top}\lambda_k + \rho_k^u)$
	$\gamma_{k+1}^x = A_k\gamma_k^x + B_k\gamma_k^u$

It is evident that this algorithm results in a considerably reduced amount of matrix operations. Regarding the choice of M_k^{uu}, both I and $\tilde{R}_k$ have been tested. Although computationally appealing, in practice these preconditioners appear to be advantageous only for single input systems. A deeper investigation will be carried out in future work.

4. SIMULATION

The proposed algorithm is run on the following test system with $n = m$

$$\begin{aligned} x_{1,k+1} &= -2x_{1,k} + u_{1,k} \\ x_{2,k+1} &= -2x_{2,k} + x_{1,k} + u_{2,k} \\ &\vdots \qquad\qquad \vdots \\ x_{i,k+1} &= -2x_{i,k} + x_{i-1,k} + u_{i,k} \\ &\vdots \qquad\qquad \vdots \\ x_{n,k+1} &= -2x_{n,k} + x_{n-1,k} + u_{n,k} \end{aligned}$$

where $x_{i,k} \in \mathbb{R}$ and $u_{i,k} \in \mathbb{R}$ are the i-th state and input at time step k. The initial condition is $x_{i,0} = 1$ for all states and the cost function for the MPC formulation is parametrized by

$$S_k = I_n \quad Q_k = 5I_n \quad R_k = 10I_n \quad Q_N = 100I_n,$$

where I_n is the n-dimensional identity matrix. Box constraints are considered on the state and input variables as stage inequalities

$$-5 \le x_{i,k} \le 5 \qquad -10 \le u_{i,k} \le 10.$$

So far the problem presents the multi-stage formulation. The structure is however compromised by introducing for instance a dynamic inequality constraint. For the test configuration the following constraint is introduced

$$-2 \le \frac{1}{N+1}\sum_{k=0}^{N} x_{i,k} \le 2 \qquad i = 1 \dots n \tag{16}$$

which limits the average value over the horizon of each state variable x_i. Note that this constraint couples all stages together.

In what follows, z is the problem optimization variable constructed as in Section 3. The full problem Hessian $\frac{1}{2}z^{\top}H\,z$ is constructed from stage matrices Q_k, S_k and R_k and all inequality constraints are stacked in $G^{\top}z \le g$, including the complicating constraint (16). To keep the analysis simple and possibly consider many different cases, the proposed PCG method is embedded in a random instance of the Interior Point method. Accordingly, the Hessian is modified by $\bar{H} = H + G\Delta G^{\top}$ [cf. (5) (6)] where the diagonal matrix Δ is constructed by randomly selecting its elements. In order to favour possibly ill-conditioned problems, the values are chosen with a very large variance. As stopping criterion for the PCG method the residual $\sqrt{\rho^{\top}\gamma}$ is checked. Due to rounding errors at later iterations, it is possible that $\rho^{\top}\gamma < 0$. If that happens, the method is also stopped. A summary of the

	n	N	res	iter	[ms]
dense schur	10	50	3.44×10^{-14}	1	52.92
PCG $M = I$	10	50	1.11×10^{-13}	20	146.71
PCG $M = \bar{H}_1$	10	50	2.75×10^{-14}	16	118.48
PCG $M = \bar{H}_2$	10	50	4.48×10^{-14}	18	129.73
dense schur	10	100	3.43×10^{-14}	1	285.12
PCG $M = I$	10	100	2.09×10^{-13}	34	658.62
PCG $M = \bar{H}_1$	10	100	1.19×10^{-13}	19	364.28
PCG $M = \bar{H}_2$	10	100	4.48×10^{-14}	14	276.20
dense schur	10	150	2.69×10^{-14}	1	821.58
PCG $M = I$	10	150	4.25×10^{-12}	23	681.04
PCG $M = \bar{H}_1$	10	150	8.05×10^{-14}	20	593.31
PCG $M = \bar{H}_2$	10	150	3.66×10^{-14}	13	381.52
dense schur	30	50	5.03×10^{-14}	1	885.25
PCG $M = I$	30	50	1.04×10^{-13}	24	384.18
PCG $M = \bar{H}_1$	30	50	6.02×10^{-14}	19	314.44
PCG $M = \bar{H}_2$	30	50	5.35×10^{-14}	14	254.32
dense schur	50	50	6.16×10^{-14}	1	3097.16
PCG $M = I$	50	50	1.53×10^{-13}	25	708.18
PCG $M = \bar{H}_1$	50	50	9.07×10^{-14}	15	427.53
PCG $M = \bar{H}_2$	50	50	7.30×10^{-14}	14	395.03

Table 1. Simulation results

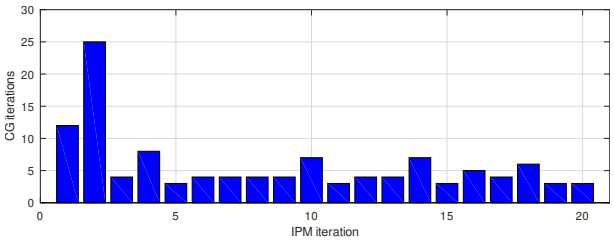

Fig. 1. Number of PCG iterations across outer IPM iterations until convergence

obtained results is reported in Table I. Here $\bar{H}_1$ and $\bar{H}_2$ are the diagonal and block-diagonal approximation of $\bar{H}$ respectively. It is noted that the PCG algorithm is implemented in MATLAB and therefore computation times are subject to function call overhead, dimension checking and unoptimized loops. In contrast, the dense Schur complement solution uses the internal optimized factorization and solve functions. So it is expected that computation times for PCG will be much lower with a low-level implementation. Anyway it is clear from Table 1 that using the structure in the preconditioners leads to a great advantage over the dense solution. Finally in Figure 1 it is shown for a simple case $N = 10$ and $n = m = 10$ the entire run of a real (not randomly generated) interior point method until the solution is found. The preconditioner used is $M = I$ which shows that it is quite effective in practice and does not require online factorization.

5. CONCLUSION

In the present paper a general linear-quadratic control problem is studied and an efficient solution has been proposed. These problems arise in many applications due to coupling between the stages induced by dynamic constraints and weights. For this class of problems the efficient Riccati recursion cannot be directly used since it needs a multi-stage structure in the cost function. In this paper it has been shown how to extend the Riccati recursion for these general problems by taking advantage of a structured projection step in the projected conjugate gradient method. The method has shown promising results and a deeper computational study with a low-level algorithm implementation will be conducted as future work.

REFERENCES

Axehill, D. and Morari, M. (2012). An alternative use of the Riccati recursion for efficient optimization. *Systems & Control Letters*, 61(1), 37–40.

Borrelli, F. (2003). *Constrained optimal control of linear and hybrid systems*, volume 290. Springer.

Camacho, E.F. and Bordons Alba, C. (2007). *Model Predictive Control*. Springer, first edition.

Coleman, T.F. (1994). Linearly constrained optimization and projected preconditioned conjugate gradients. In *Proceedings of the 5th SIAM Conference on Applied Linear Algebra*, 118–122.

Domahidi, A., Zgraggen, A.U., Zeilinger, M.N., Morari, M., and Jones, C.N. (2012). Efficient interior point methods for multistage problems arising in receding horizon control. In *Proceedings of 51st IEEE Conference on Decision and Control (CDC)*, 668–674. IEEE.

Frison, G. and Jørgensen, J.B. (2013a). Efficient implementation of the Riccati recursion for solving linear-quadratic control problems. In *Proceedings of 2013 IEEE International Conference on Control Applications (CCA)*, 1117–1122. IEEE.

Frison, G. and Jørgensen, J.B. (2013b). A fast condensing method for solution of linear-quadratic control problems. In *Proceedings of 52nd IEEE Conference on Decision and Control*, 7715–7720. IEEE.

Gould, N.I., Hribar, M.E., and Nocedal, J. (2001). On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM Journal on Scientific Computing*, 23(4), 1376–1395.

Jørgensen, J.B., Frison, G., Gade-Nielsen, N.F., and Damman, B. (2012). Numerical methods for solution of the extended linear quadratic control problem. *IFAC Proceedings Volumes*, 45(17), 187–193.

Malyshev, A., Quirynen, R., and Knyazev, A. (2018). Preconditioned Krylov iterations and condensing in interior point MPC method. *IFAC-PapersOnLine*, 51(20), 388–393.

Rao, C.V., Wright, S.J., and Rawlings, J.B. (1998). Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications*, 99(3), 723–757.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Pub., second edition.

Shahzad, A., Kerrigan, E.C., and Constantinides, G.A. (2010a). A fast well-conditioned interior point method for predictive control. In *Proceedings of 49th IEEE Conference on Decision and Control (CDC)*, 508–513. IEEE.

Shahzad, A., Kerrigan, E.C., and Constantinides, G.A. (2010b). Preconditioners for inexact interior point methods for predictive control. In *Proceedings of the 2010 American Control Conference*, 5714–5719. IEEE.

Wang, Y. and Boyd, S. (2009). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278.