2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Preconditioning Matrix Synthesis for a Projected Gradient Method for Solving Constrained Linear-Quadratic Optimal Control Problems

Y.J.J. Heuts and M.C.F. Donkers

Abstract— **This paper presents a method for synthesizing preconditioning matrices for a heavy-ball accelerated projected primal-dual method. The main focus lies on linear quadratic optimal control problems, as they have a specific structure that can be exploited for fast computational times. This gradient method is rewritten into a Lur'e-type system, such that convergence of the algorithm can be enforced through finding an appropriate Lyapunov function for the Lur'e system. It has been shown that for a small problem, it is possible to synthesize preconditioning matrices and that the method is 10^4 times faster than solving the projection using a dedicated solver.**

I. Introduction

Optimal control problem arise in many fields, where a priori knowledge of the system can be used to predict future states, which can be used to choose the control input such that the system stays within predefined constraints. In order to solve these optimal control problems, online fast optimization algorithms are required, as the effectivity of the solution is increased when a new solution is generated at each sample. It is common to use commercial solvers, such as cplex, [1], or mosek [2], but drawbacks are that these methods cannot be compiled onto an embedded system, like open source solvers, such as osqp [3]. On the other hand, there is rich theory available which enables anyone to code and compile a custom algorithm. One popular approach is to use gradient methods, which are widely used to solve convex optimization problems. Most often, optimal control problems have a quadratic cost function and linear constraints, hence the name linear quadratic (LQ) optimal control. Non-linearities or integer decisions problems can also be solved indirectly by gradient methods by employing sequential quadratic programming, e.g., [4] in the former case, while branch-and-bound methods, e.g., [5] can be applied in the latter case. In both cases, LQ sub-problems have to be solved repeatedly. For many of these problems, it is beneficial to have fast computational times, as they are intended to run in a limited time window. This calls for methods that have a low computational effort and scale well when the number of decision variables increases, i.e., due to larger or more complex models, or longer prediction horizons.

Gradient methods for optimization can often be written as a static linear time-invariant system with a non-linear input function, often referred to as Lur'e-type systems [6]. This implicit connection to system and control theory makes it very attractive to analyse and design algorithms as if they were dynamical systems, e.g., using robust control theory. In recent years, this view on first-order methods has been explored more. Most recent publications focus on analysing achievable worst-case convergence rates, i.e., [7], [8]. On the other hand, algorithms can be synthesized using robust control techniques to find optimal hyperparameters, such as step-sizes for acceleration schemes, such as heavy-ball, Nesterov's acceleration or the triple momentum method. This improves stability or performance characteristics of optimization algorithms, i.e., [9] which in turn improve convergence rates. In general, these methods rely on the fact that the problem is properly conditioned. Preconditioning of the problem can significantly improve stability and performance characteristics of optimization algorithms, as seen in [10], [11]. Many of these previously mentioned articles use heuristics or rules to choose these preconditioning matrices, while the synthesizing of optimal preconditioning matrices has not yet been explored to our knowledge. Therefore, tuning rules have to be devised on a per-method basis, which leads to unnecessary amounts of tuning.

In this paper, we propose an automated method of synthesizing a structured preconditioning matrix for the projected primal-dual method of [12] by means of solving a matrix inequality (MI). Unlike the work in [12], which focuses on LQ problems with diagonal weighting matrices in the cost function, this paper aims at applicability to non-diagonal weighting matrices in the cost function. The preconditioning matrices will be synthesized such that the projection operator simplifies into a clipping action, resulting in a computationally efficient algorithm. A matrix inequality will be formulated by writing the projected primal dual method as a Lur'e-type dynamic system and by making use of Lyapunov functions to enforce stability. This is possible due to the projections being a sector bounded condition. This matrix inequality will be solved by using a linearization scheme, such that each sub-problem becomes a linear matrix inequality (LMI) for which efficient solvers exist. Finally, the paper presents a simulation example that shows the computational efforts of computing the preconditioning matrices, as well as the computational advantages of synthesizing problem-specific preconditioning matrices to solving the projection using a dedicated solver using the previously found preconditioning matrix in [12].

This work has received financial support from the Horizon 2020 programme of the European Union under the grants 'Efficient and environmental friendly LONG distance poweRtrain for heavy dUty trucks aNd coaches' (LONGRUN-874972). The authors are with the Dept. of Electrical Eng. of Eindhoven University of Technology, Netherlands. E-mail: {y.j.j.heuts, m.c.f.donkers}@tue.nl

II. Problem Formulation

In this section, we will introduce the finite-horizon discrete-time linear-quadratic (LQ) optimal control problem (OCP) with state and input constraints that is considered in this paper. We will then rewrite this OCP in a highly structured static optimization problem and present the preconditioned projected primal-dual method proposed in [12] to solve this optimization problem. We will also indicate why the preconditioning matrix proposed in [12] does not work in case the weighting matrices in the OCP are non-diagonal, thereby motivating the need for the method for synthesizing the preconditioning matrices that we will propose in this paper.

A. Constrained Linear Quadratic Optimal Control Problem

Let us consider a discrete-time horizon $\mathcal{K}=\{0,\dots,K-1\}$, where $K \in \mathbb{N}$ is the horizon length and a discrete-time dynamical system with states $x_k \in \mathbb{R}^{n_x}$ and inputs $u_k \in \mathbb{R}^{n_u}$, $k \in \mathcal{K}$. For this system, we define the discrete-time LQ OCP as follows:

$$\begin{aligned}\min_{x_k,u_k} \quad & \sum_{k\in\mathcal{K}} \frac{1}{2}\begin{bmatrix}x_k-r_k\\u_k\end{bmatrix}^\top \begin{bmatrix}Q & S\\ S^\top & R\end{bmatrix}\begin{bmatrix}x_k-r_k\\u_k\end{bmatrix} \\ & \qquad + \tfrac{1}{2}(x_K-r_K)^\top P(x_K-r_K) && \text{(1a)}\\ \text{s.t.} \quad & x_{k+1}=Ax_k+Bu_k && \text{(1b)}\\ & \underline{x}\leqslant x_k \leqslant \bar{x} && \text{(1c)}\\ & \underline{u}\leqslant u_k\leqslant \bar{u}, && \text{(1d)}\end{aligned}$$

for a given initial condition $x_0 = x_{\text{init}}$ and given reference trajectory r_k for all $k \in \mathcal{K}$, and where A and B describe the dynamics of a controllable system. The weighing matrix P is positive definite, as well as $\left[\begin{smallmatrix}Q & S\\ S^\top & R\end{smallmatrix}\right] \succ 0$. Finally, $\overline{x}$ and $\underline{x}$ are the state upper and lower bounds, respectively, and $\overline{u}$ and $\underline{u}$ are the input upper and lower bounds, respectively. We assume that OCP (1) is feasible for the given state and input constraints.

B. A Sparse QP Formulation

As argued in [13] and shown in [12], a sparse formulation of the OCP has computational benefits in terms of computational complexity and memory requirements. We therefore write OCP (1) as a sparse static optimization problem, in which considerable amounts of zero entries are ensured in the matrices. This static optimization problem is given by

$$\begin{aligned}\min_{\omega}\quad & \tfrac{1}{2}\omega^\top \mathbf{G}\omega + \mathbf{F}^\top\omega + \imath_\Omega(\omega) && \text{(2a)}\\ \text{s.t.}\quad & \mathbf{A}\omega - \mathbf{b} = 0, && \text{(2b)}\end{aligned}$$

where $\omega = [x^\top \;\; u^\top]^\top$, in which $x = [x_0^\top \dots x_K^\top]^\top$, and $u = [u_0^\top \dots u_{K-1}^\top]^\top$, $\imath_\Omega(\omega)$ denotes the indicator function satisfying $\imath_\Omega(\omega) = 0$ if $\omega \in \Omega$ and $\imath_\Omega(\omega) = \infty$ if $\omega \notin \Omega$, where Ω is the box-constrained feasible set

$$\Omega := \{\omega \in \mathbb{R}^{(K+1)n_x+Kn_u} \mid \underline{\omega} \leqslant \omega \leqslant \overline{\omega}\}, \qquad (3)$$

in which $\underline{\omega}$ and $\overline{\omega}$ are defined similar to ω. The matrices in the cost function (1a) are fitted into sparse matrices

$$\mathbf{G} = \begin{bmatrix} I\otimes Q & 0 & I\otimes S\\ 0 & P & 0\\ I\otimes S^\top & 0 & I\otimes R\end{bmatrix}, \qquad (4a)$$

$$\mathbf{F} = [-r_0^\top Q,\dots, -r_{K-1}^\top Q, -r_K^\top P, -r_0^\top S,\dots, -r_{K-1}^\top S]^\top, \qquad (4b)$$

where $\otimes$ denotes the Kronecker product and (1b) is captured in the equality constraint (2b), where $\mathbf{A} = [\Gamma_x \;\; \Gamma_u]$, $\mathbf{b} = [x_{\text{init}}^\top \;\; 0 \;\dots\; 0]^\top$ and

$$\Gamma_x = \begin{bmatrix} I & 0 & \dots & 0\\ A & -I & \ddots & \vdots\\ 0 & \ddots & \ddots & 0\\ 0 & 0 & A & -I\end{bmatrix}, \quad \Gamma_u = \begin{bmatrix} 0 & \dots & 0\\ B & 0 & \vdots\\ 0 & \ddots & 0\\ 0 & \dots & B\end{bmatrix}. \qquad (5)$$

The sparsity present in the above matrices allows for the development of computationally efficient algorithms to solve (2) when the underlying problem is OCP (1).

C. A Preconditioned Projected Primal-Dual Method

The optimal solution of the static optimization problem is obtained using the following non-smooth Lagrangian function

$$L(\omega,\lambda) = \tfrac{1}{2}\omega^\top\mathbf{G}\omega + \mathbf{F}^\top\omega + \lambda^\top(\mathbf{A}\omega - \mathbf{b}) + \imath_\Omega(\omega), \qquad (6)$$

which allows the optimal solution to be characterized using the KKT conditions, which are given by

$$0 \in \begin{bmatrix}\mathbf{G} & \mathbf{A}^\top\\ -\mathbf{A} & 0\end{bmatrix}\begin{bmatrix}\omega\\ \lambda\end{bmatrix} + \begin{bmatrix}\mathbf{F}+\mathcal{N}_\Omega(\omega)\\ \mathbf{b}\end{bmatrix}. \qquad (7)$$

In this expression, $\mathcal{N}_\Omega$ is the normal cone operator satisfying $\mathcal{N}_\Omega(\omega) = \{v \in \mathbb{R}^n \mid \sup_{z\in\Omega} v^\top(z-\omega) \leqslant 0\}$ if $\omega \in \Omega$ and $\mathcal{N}_\Omega(\omega) = \varnothing$ if $\omega \notin \Omega$. To find a solution to (7), we will use the method from [12], which is given by

$$\begin{cases}\omega^{i+1} = \operatorname{proj}_\Omega^{\Phi_\omega}\{\omega^i - \Phi_\omega(\mathbf{G}\omega^i + \mathbf{F} + \mathbf{A}^\top\lambda^i)\}\\ \lambda^{i+1} = \lambda^i + \Phi_\lambda(\mathbf{A}\omega^{i+1} - \mathbf{b}) + \beta(\lambda^i - \lambda^{i-1}),\end{cases} \qquad (8)$$

for some positive-definite preconditioning matrix Φ_ω and some non-singular preconditioning matrix Φ_λ. In this expression, $\operatorname{proj}_\Omega^{\Phi_\omega}(v) = \arg\min_{x\in\Omega}(x-v)^\top\Phi_\omega^{-1}(x-v)$ denotes a (weighted) Euclidean projection. In case the preconditioning matrix Φ_ω is diagonal, the Euclidean projection simplifies and (8) becomes

$$\begin{cases}\omega^{i+1} = \max\{\underline{\omega}, \min\{\overline{\omega}, \omega^i - \Phi_\omega(\mathbf{G}\omega^i + \mathbf{F} + \mathbf{A}^\top\lambda^i)\}\}\\ \lambda^{i+1} = \lambda^i + \Phi_\lambda(\mathbf{A}\omega^{i+1} - \mathbf{b}) + \beta(\lambda^i - \lambda^{i-1}),\end{cases} \qquad (9)$$

For the case that $\mathbf{G}$ is diagonal, it has been shown in [12] that this algorithm has competitive computational performance by choosing $\Phi_\omega = \mathbf{G}^{-1}$ and $\Phi_\lambda = (\mathbf{A}^\top\mathbf{G}^{-1}\mathbf{A})^{-1}$. For non-diagonal matrices Φ_ω, (9) is no longer equivalent to (8), while the latter cannot be solved efficiently. The objective of this paper is therefore to synthesize a diagonal matrix Φ_ω and a matrix Φ_λ for the case that $\mathbf{G}$ is non-diagonal so that (9) can be used instead of (8), while still warranting that (9) converges and that the computational performance is competitive.

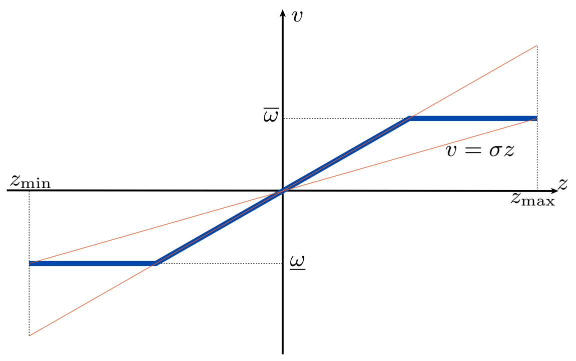

Fig. 1. The sector-bounded non-linearity (10c) indicated with a blue line. For $\sigma > 0$ the output v can be in the blue area.

III. Synthesis

In this section, we will develop a method to synthesize a positive-definite diagonal precondition matrix Φ_ω and a highly-structured preconditioning matrix Φ_λ. We will first formulate the convergence analysis problem for $\beta = 0$ using linear matrix inequalities, and we then propose an iterative method to solve the synthesis problem. We will propose a way to allow for $\beta \neq 0$ in the next section.

The analysis result that we will develop is based in the existence of a Lyapunov function. To do so, we will first rewrite (9) as a Lur'e system, i.e., an interconnection of a linear-time invariant dynamic system with a static non-linearity, leading to

$$\chi^{i+1} = \mathcal{A}\chi^i + \mathcal{B}v^i + \mathcal{E} \tag{10a}$$
$$z^i = \mathcal{C}\chi^i + \mathcal{F} \tag{10b}$$
$$v^i = \max\{\underline{\omega}, \min\{\overline{\omega}, z^i\}\}, \tag{10c}$$

with $\chi = [\omega^\top \quad \lambda^\top]^\top$ and

$$\mathcal{A} = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, \qquad \mathcal{B} = \begin{bmatrix} I \\ \Phi_\lambda \mathbf{A} \end{bmatrix}, \qquad \mathcal{E} = \begin{bmatrix} 0 \\ -\Phi_\lambda \mathbf{b} \end{bmatrix} \tag{11a}$$
$$\mathcal{C} = \begin{bmatrix} (I - \Phi_\omega \mathbf{G}) & -\Phi_\omega \mathbf{A}^\top \end{bmatrix}, \qquad \mathcal{F} = -\Phi_\omega F. \tag{11b}$$

It can be observed in Fig. 1 that the nonlinearity in (10c) satisfies a so-called sector-condition, as was also observed in [12]. In this paper, we modify the analysis procedure of the aforementioned paper to a local analysis. In particular, we assume that the solutions of (9), and equivalently (10) always remain bounded, i.e., they remain $\underline{\omega}/\sigma \leqslant \omega = z < \overline{\omega}/\sigma$, where $\sigma \in (0, 1]$. Under this assumption, we have that (10c) satisfies the following sector condition

$$(v_\ell - \sigma z_\ell)(v_\ell - z_\ell) \leqslant 0, \tag{12}$$

where ℓ denotes the ℓ-th element of v and z. This allows for exponential stability without the need for a LaSalle's argument, yet the results are only valid locally.

A. Convergence Analysis

To show that (9) converges to the optimal solution if (1) is feasible, we will study exponential stability of an equilibrium point $\chi^\star$ of (10). By exponential stability, we mean that its solutions $\|\chi^i - \chi^\star\| \leqslant ce^{-\tau i}\left\|\chi^0 - \chi^\star\right\|$ for all $i \in \mathbb{N}$ and for some $c > 0$ and for some convergence rate $\tau \in [0, 1)$. The theorem below will state the conditions for which this can be guaranteed and uses slack variables as introduced in [14] and the S-procedure as used in [15].

Theorem 1 *Assume* (1) *is feasible, that a positive-definite diagonal matrix* Φ_ω *and nonsingular matrix* Φ_λ *are given and that* $\beta = 0$. *If there exist a positive definite matrix* X, *a diagonal matrix* S, *a matrix* Γ, *and scalars* $\sigma \in (0, 1]$, $\tau \in [0, 1)$, *satisfying*

$$\begin{bmatrix} \tau X & \star & \star \\ \frac{\sigma-1}{2} S\mathcal{C} & S & \star \\ \Gamma(\mathcal{A} + \sigma\mathcal{B}\mathcal{C}) & \Gamma\mathcal{B} & \Gamma + \Gamma^\top - X \end{bmatrix} \succeq 0, \tag{13}$$

then (10) *with* (11) *has an exponentially stable equilibrium point* $\xi^\star$ *with convergence rate less than or equal to* τ, *as long as solutions* $\underline{\omega}/\sigma \leqslant z^i \leqslant \overline{\omega}/\sigma$ *for all* $i \geqslant 0$. *Consequently,* (9) *with* $\beta = 0$ *finds the solution to* (1).

Proof: Since, we assume that problem (1) is feasible, there exists a fixed point $(\omega^\star, \lambda^\star, z^\star, v^\star)$ of (9) satisfying the KKT conditions (7). This fixed point allows rewriting (10) as

$$\chi^{i+1} - \chi^\star = \mathcal{A}(\chi^i - \chi^\star) + \mathcal{B}(v^i - v^\star) \tag{14a}$$
$$z^i - z^\star = \mathcal{C}(\chi^i - \chi^\star) \tag{14b}$$
$$v^i - v^\star = \max\{\underline{\omega}, \min\{\overline{\omega}, z^i\}\} - z^\star, \tag{14c}$$

with $v^\star = \omega^\star = z^\star$ as the problem is feasible, i.e., $\underline{\omega} \leqslant v^\star \leqslant \overline{\omega}$. To prove exponential stability of this system, we define a candidate Lyapunov function

$$V(\chi^i - \chi^\star) = (\chi^i - \chi^\star)^\top X(\chi^i - \chi^\star) \tag{15}$$

for some matrix $X \succeq 0$ that satisfies

$$V(\chi^{i+1} - \chi^\star) - \tau V(\chi^i - \chi^\star)$$
$$\leqslant (\omega^i - \omega^\star - \sigma(z^i - z^\star))S(\omega^i - \omega^\star - z^i + z^\star) \leqslant 0 \tag{16}$$

for some positive-definite diagonal matrix $\mathcal{S}$, as sector condition (12) holds element-wise, and some $\tau \in [0, 1)$.

The proof of this theorem proceeds by showing that (16) is implied by satisfaction of (13). First, let us substitute (14) and (15) into (16), leading to

$$\begin{bmatrix} \chi - \chi^\star \\ v - v^\star \end{bmatrix}^\top \begin{bmatrix} \mathcal{A}^\top X\mathcal{A} - \tau X - \sigma\mathcal{C}^\top S\mathcal{C} & \star \\ \mathcal{B}^\top X\mathcal{A} + \frac{1+\sigma}{2} S\mathcal{C} & \mathcal{B}^\top X\mathcal{B} - S \end{bmatrix} \begin{bmatrix} \chi - \chi^\star \\ v - v^\star \end{bmatrix} \leqslant 0. \tag{17}$$

This is, after applying a Schur complement, equivalent to requiring that

$$\begin{bmatrix} \tau X + \sigma\mathcal{C}^\top S\mathcal{C} & -\frac{1+\sigma}{2}\mathcal{C}^\top S & \mathcal{A}^\top \\ -\frac{1+\sigma}{2} S\mathcal{C} & S & \mathcal{B}^\top \\ \mathcal{A} & \mathcal{B} & X^{-1} \end{bmatrix} \succeq 0 \tag{18}$$

$$\begin{bmatrix} \tau X & \star & \star \\ \frac{\sigma-1}{2}\left[(I-\Phi_\omega \mathbf{G}) \quad -\Phi_\omega \mathbf{A}^\top\right] & \Phi_\omega & \star \\ \hat{\Gamma}\begin{bmatrix} \sigma(I-\Phi_\omega \mathbf{G}) & -\sigma\Phi_\omega\mathbf{A}^\top \\ \sigma\mathbf{A}(I-\Phi_\omega\mathbf{G}) & (1-\sigma)\mathbf{A}\Phi_\omega\mathbf{A}^\top\end{bmatrix} & \hat{\Gamma}\begin{bmatrix}\Phi_\omega \\ \mathbf{A}\Phi_\omega\end{bmatrix} & \hat{\Gamma}\begin{bmatrix} I & 0 \\ 0 & \mathbf{A}\Phi_\omega\mathbf{A}^\top\end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & \mathbf{A}\Phi_\omega\mathbf{A}^\top\end{bmatrix}\hat{\Gamma}^\top - X \end{bmatrix} \succeq 0, \tag{23}$$

Now by permuting the second and the third rows and columns and applying a Schur complement again, we equivalently have that

$$\begin{bmatrix} \tau X + (\sigma - \frac{(1+\sigma)^2}{4})\mathcal{C}^\top S\mathcal{C} & \star \\ \mathcal{A} + \frac{1+\sigma}{2}\mathcal{B}\mathcal{C} & X^{-1} - \mathcal{B}S^{-1}\mathcal{B}^\top \end{bmatrix} \succeq 0 \tag{19}$$

and because $\sigma - \frac{(1+\sigma)^2}{4} = \frac{4\sigma-1-2\sigma-\sigma^2}{4} = \frac{2\sigma-1-\sigma^2}{4} = -\frac{(1-\sigma)^2}{4}$, and likewise $\frac{1+\sigma}{2} = \sigma + \frac{1-\sigma}{2}$, we can write

$$\begin{bmatrix} \tau X - \frac{(1-\sigma)^2}{4}\mathcal{C}^\top S\mathcal{C} & \star \\ \mathcal{A} + (\sigma + \frac{1-\sigma}{2})\mathcal{B}\mathcal{C} & X^{-1} - \mathcal{B}S^{-1}\mathcal{B}^\top \end{bmatrix} \succeq 0. \tag{20}$$

Finally, we apply another Schur complement, permute the second and the third row and columns, and pre- and post-multiply the last rows and columns, respectively, with the slack-variable Γ. This leads to

$$\begin{bmatrix} \tau X & \star & \star \\ \frac{\sigma-1}{2}S\mathcal{C} & S & \star \\ \Gamma(\mathcal{A}+\sigma\mathcal{B}\mathcal{C}) & \Gamma\mathcal{B} & \Gamma X^{-1}\Gamma^\top \end{bmatrix} \succeq 0 \tag{21}$$

which is implied by (13), because $\Gamma X^{-1}\Gamma^\top \succeq \Gamma + \Gamma^\top - X$. Hence, satisfaction of (13) warrants the existence of an exponentially decreasing Lyapunov function exists for system (10). To conclude the proof, observe that (10) is an equivalent representation of (13) for $\beta = 0$. ∎

The above result generalizes the analysis result of [12], where the case in which $\mathbf{G}$ is diagonal was studied and the preconditioning matrices were selected to be $\Phi_\omega = \mathbf{G}^{-1}$, $\Phi_\lambda = \alpha(\mathbf{A}\mathbf{G}^{-1}\mathbf{A}^\top)^{-1}$, for $\alpha \in (0, 2]$.

B. Synthesis Procedure

Using (13) directly for the design of preconditioning matrices Φ_ω and Φ_λ is not possible due to the nonlinear appearance of these matrices in the matrix inequality (13). Furthermore, not all preconditioning matrices would lead to a computationally efficient algorithm. We therefore impose the following addition structure, even though this might introduce conservatism. In particular, we choose

- Φ_ω as a diagonal matrix as this allows us to use (9), instead of (8)
- $S = \Phi_\omega^{-1}$, which has been shown to be a good choice in [12], and reduces the number of nonlinear terms in (13)
- $\Phi_\lambda = (\mathbf{A}\Phi_\omega\mathbf{A}^\top)^{-1}$, as this matrix is an inverse of a block-banded matrix, which can be solved in an extremely efficient manner, as was shown in [12].

Besides making these assumptions, we also substitute the matrices in (11), in which

$$\mathcal{A} + \sigma\mathcal{B}\mathcal{C} = \begin{bmatrix} \sigma(I-\Phi_\omega\mathbf{G}) & -\sigma\Phi_\omega\mathbf{A}^\top \\ \sigma\Phi_\lambda\mathbf{A}(I-\Phi_\omega\mathbf{G}) & I-\sigma\Phi_\lambda\mathbf{A}\Phi_\omega\mathbf{A}^\top \end{bmatrix}, \tag{22}$$

and define $\hat{\Gamma} = \Gamma\begin{bmatrix} I & 0 \\ 0 & \Phi_\lambda^{-1}\end{bmatrix}$, leading to (23), as shown on top of this page.

It can be observed that (23) is still not a linear matrix inequality, due to the product of Φ_ω and $\hat{\Gamma}$. We therefore propose to use an iterative scheme, similar to the one proposed in [16], [17], and is based on the observation that (23) becomes a linear matrix inequality when $\hat{\Gamma}$ is given. We therefore propose to iteratively minimize τ while taking Γ equal to X of the previous iterate. The algorithm is initialized by minimizing τ subject to (23) for Φ_ω equal to the inverse of the diagonal elements of $\mathbf{G}$, thereby approximating the results of [16]. The complete algorithm is given below.

Algorithm 1 Iterative algorithm for synthesizing preconditioning matrices

```
1: Let $\sigma \in (0, 1]$ be given and let $\Phi_\omega = \mathrm{diag}(\mathrm{diag}(\mathbf{G}))^{-1}$
2: $\min \tau$ subject to (23) for the given $\Phi_\omega$.
3: **while** $\tau > 1$ **do**
4:     $\hat{\Gamma} \leftarrow X\begin{bmatrix} I & 0 \\ 0 & \mathbf{A}\Phi_\omega\mathbf{A}^\top\end{bmatrix}$
5:     $\min \tau$ subject to (23) for given $\hat{\Gamma}$.
6: **end while**
7: **return** $\Phi_\omega$ and $\Phi_\lambda = (\mathbf{A}\Phi_\omega\mathbf{A}^\top)^{-1}$
```

IV. Implementation Details

To complete the projected primal-dual method (9) with preconditioning matrices obtained using the procedure proposed in the previous section, we introduce the adaptive restart of the parameter β, the termination criterion used for the algorithm and an overview on the implementation of the algorithm.

A. Adaptive Restart

In order to remove the complexity of tuning both α and β, we choose β such that it sweeps from 0 to 1. One such variable step-size is given in [18], where β_i is chosen to be

$$\beta^i = \frac{i}{i+b}, \tag{24}$$

for $i \in \mathbb{N}$ and $b > 0$ is used to tune the speed at which β reaches the upper limit.

A disadvantage of accelerating gradient schemes is that it can lead to side effects such as high momentum. This causes the algorithm to miss crucial points, as seen in [19]. In order to avoid this phenomenon, the gradient scheme from the same paper is implemented, such that β^i is reset to zero when the Lagrangian of the preconditioned system starts to decrease (as we aim at maximizing this Lagrangian over λ), which equals to

$$\nabla_\lambda L(\lambda^{i-1})^\top \Phi_\lambda(\lambda^i - \lambda^{i-1}) \leqslant 0, \tag{25}$$

where L is the Lagrangian given in (6). For the actual implementation, the scheme is rewritten by introducing $\hat{\lambda}^i = \lambda^i + \beta(\lambda^i - \lambda^{i-1})$, as was proposed in [19] and using the update rule for λ as in (9), allowing us to rewrite (25) as

$$(\lambda^i - \hat{\lambda}^{i-1})^\top(\lambda^i - \lambda^{i-1}) \leqslant 0. \tag{26}$$

The advantage of this notation is that the vectors in this notation are available already, whereas calculation the value of the gradient of the Lagrangian would add extra unnecessary computational complexity.

B. Termination Criteria

The problem is solved as soon as $\omega^\star$ and $\lambda^\star$ satisfy (7). We consider the solution to be of sufficient accuracy as soon as

$$\|\mathbf{A}\omega^{i+1} - \mathbf{b}\|_2 \leqslant \varepsilon_d, \tag{27}$$

and

$$\|\omega^{i+1} - \omega^i\|_2 \leqslant \varepsilon_p, \tag{28}$$

where ε_d and ε_p are the primal and dual tolerances of the algorithm.

C. Algorithm

The complete gradient method that will be used to compute the solution to the optimization problem is shown in Algorithm 2. It should be noted that the algorithm can be coded in Matlab in less than 30 lines of code, making implementation rather simple. The inverse of Φ_λ should not be computed explicitly, as it is much faster to solve the systems of equations due to the sparse structure in the preconditioner.

Algorithm 2 Heavy-Ball Projected Primal-Dual Method with synthesized preconditioner

```
1:  Initialize:
        Select $\alpha$ and $b$ and a desired accuracy $\varepsilon$.
        Determine $\Phi_\omega$ using algorithm 1.
        Compute $\Psi_1 = I - \Phi_\omega\mathbf{G}$, $\Psi_2 = \Phi_\omega\mathbf{A}^\top$,
        $\Psi_3 = \Phi_\omega\mathbf{F}$, $\Phi_\lambda^{-1} = \mathbf{A}\Phi_\omega\mathbf{A}^\top$,
        Set $\lambda = -\Phi_\lambda(\mathbf{b} + \mathbf{A}\mathbf{G}^{-1}\mathbf{F})$, $\lambda^- = \mathbf{0}$, $i = 1$
2:  loop
3:      $\omega \leftarrow \max\{\underline{\omega}, \min\{\overline{\omega}, \Psi_1\omega - \Psi_2\lambda - \Psi_3\}\}$
4:      if $\|A\omega - b\|_2 \leqslant \varepsilon$ then
5:          return $\lambda, \omega$
6:      end if
7:      $\hat{\lambda} \leftarrow \lambda + \frac{i}{i+b}(\lambda - \lambda^-)$
8:      $\lambda^- \leftarrow \lambda$
9:      $\lambda \leftarrow \hat{\lambda} + \Phi_\lambda(A\omega - b)$
10:     if $(\lambda - \hat{\lambda})^\top(\lambda - \lambda^-) \leqslant 0$ then
11:         $i \leftarrow 0$
12:     else
13:         $i \leftarrow i + 1$
14:     end if
15: end loop
```

V. Numerical Example

In this section, we evaluate the computational performance of solving the OCP using (9), and compare the synthesized preconditioning matrices of Algorithm 1 with preconditioning matrices proposed in [12]. These predefined matrices are $\Phi_\omega = \mathbf{G}^{-1}$, which is diagonal, and $\Phi_\lambda = \alpha(\mathbf{A}\Phi_\omega\mathbf{A}^\top)^{-1}$ for some $\alpha \in (0, 2]$.

Example 1 *The inverted pendulum is given in continuous time by $\dot{x} = A_c x + B_c u$ with*

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.048 & 1.58 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.080 & 18.97 & 0 \end{bmatrix}, B_c = \begin{bmatrix} 0 & 0 \\ 0.96 & 1.61 \\ 0 & 0 \\ 1.61 & 0.96 \end{bmatrix}, \tag{29}$$

and is discretized with zero-order hold with a sampling time of 0.1s. We take $Q = P = I$ and $R = 0.1I$ in (1a). *Furthermore, the constraints are $\overline{u}_k = -\underline{u}_k = [4, 4]^\top$, $\overline{x}_k = [10, \hat{x}, 5, 10]^\top$, $\underline{x}_k = [-10, -5, -10, -5]^\top$, $\hat{x}_k = 3\sin(\frac{2\pi k}{9} + 0.5\pi) + 1$ and $r_k = 0$.*

Example 2 *As a second case, a non-diagonal term is added by computing the final cost, P, by solving the algebraic Riccati equation $A_d P A_d^\top - A_d P B_d(B_d^\top P B_d + R)^{-1} B_d^\top P A_d^\top - Q = 0$.*

Example 3 *As a third case within the inverted pendulum example, additional to the extension in Example 2, a more non-diagonal dominant cost function is considered, such that*

$$Q = \begin{bmatrix} 3 & 0 & 0 & -2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -2 & 0 & 0 & 3 \end{bmatrix}. \tag{30}$$

The convergence results from the problem with diagonal cost matrix, i.e., Example 1, can be seen in Fig. 2, where GM indicates gradient method without heavy ball, i.e., $\beta = 0$. GM+HB is the gradient method with heavy ball, i.e., β is given by (24). SYN indicates the synthesized preconditioner using Algorithm 1, while KNOWN indicates the preconditioning from [12]. It can be seen that the synthesis procedure generates a stabilizing preconditioner for the given hyper-parameters, $\tau = 0.47$ and $\sigma = 0.3$. It can be seen in Figure 2 that GM+SYN and GM+KNOWN+HB perform equally well, while being 11.3 times faster than the gradient method without heavy-ball (GM+KNOWN). The addition of heavy-ball to the synthesized preconditioning only improves upon the convergence rate by 3 iterations in this example. Being able to achieve comparable convergence speeds without the need of any heavy-ball, its associated tuning and additional mechanisms, such as resetting as explained in Section IV is a real advantage, as it removes the requirement of hand-tuning the algorithm by the user. Now the user solves the matrix inequality (23) for the given plant model and the tuning will be done automatically.

Besides the diagonal case, a study on non-diagonal cost matrices was also performed, see Example 2 and 3. The results are given in Fig. 3 and 4. Note that the figures do

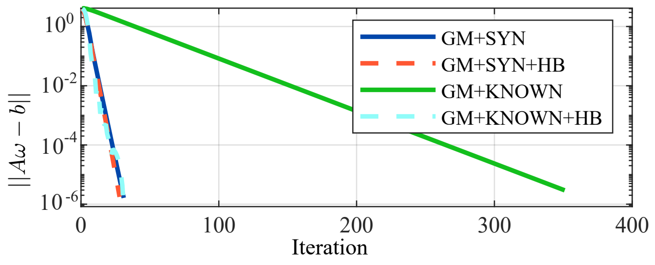

Fig. 2. Convergence rate of example 1 with horizon length $N = 5$. The results were achieved using variable step-size constant $b = 2500$ for the orange line and $b = 25$ for the cyan line.

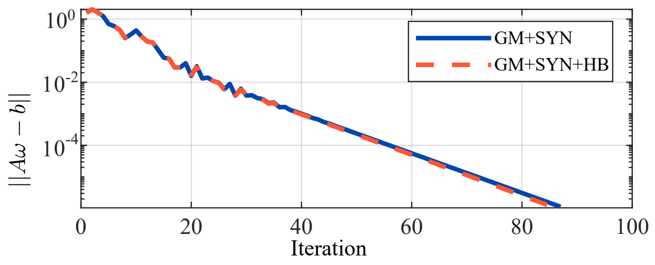

Fig. 3. Convergence rate of Example 2 with horizon length $N = 5$. The results were achieved variable step-size, $b = 10^4$.

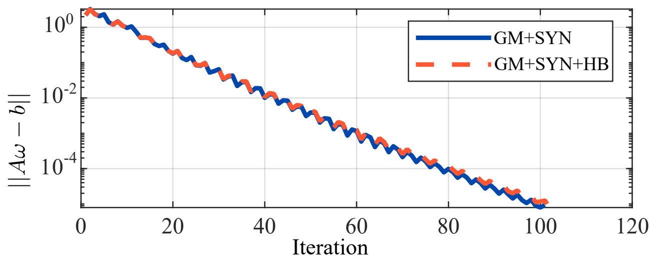

Fig. 4. Convergence rate of Example 3 with horizon length $N = 5$. The results were achieved using variable step-size, $b = 10^4$.

not show monotonic convergence, as our Lyapunov function (15) guarantees monotonic convergence against $\chi - \chi^\star$, while $\|A\omega - b\|$ is shown on the y-axis. The computational times for these two examples are compared to solving the projection, or (8), using mosek, see [2], as a dedicated solver. These computational times are reported in Table I, where it can be seen that using a synthesized method, solving the QP is significantly faster than solving the projection using a dedicated solver.

It should be noted that using Algorithm 1 is not guaranteed to be numerical stable and could lead to long computational times for longer horizons, as the elements in X increase quadratically with the horizon length. For model-predictive-control-type applications, this would not be a huge problem, as the preconditioning could be computed once and stored in memory for use. Yet, for longer-scale problems with long time horizons, it would be interesting to see if the computational times could be reduced, so that the approach is also applicable to larger scale problems.

VI. Conclusion

In this paper, we have proposed a novel way of synthesizing preconditioning matrices for which we have shown that stability is guaranteed. By means of a Lyapunov function, a diagonal preconditioner can be generated, such that the projection in the gradient method simplifies to a clipping operation, which leads to considerable performance benefits. Though the synthesis of the preconditioning matrices may be intractable for larger problems, it has been shown that synthesizing preconditioning matrices using this method leads to very fast convergence rates for shorter time horizons.

References

[1] "V12. 1: User's manual for cplex," *International Business Machines Corporation*, 2009.
[2] *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019.
[3] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, 2020.
[4] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta numerica*, 1995.
[5] S. Boyd and J. Mattingley, "Branch and bound methods," *Notes for EE364b, Stanford University*, 2007.
[6] O. Komarnitskaia, "Stability of nonlinear automatic control systems," *Journal of Applied Mathematics and Mechanics*, 1959.
[7] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, 2016.
[8] M. Li, K. Laib, and I. Lestas, "Convergence rate bounds for the mirror descent method: Iqcs and the bregman divergence," in *IEEE Conf on Decision & Control*, IEEE, 2022.
[9] C. Scherer and C. Ebenbauer, "Convex synthesis of accelerated gradient algorithms," *SIAM Journal on Control and Optimization*, 2021.
[10] P. Giselsson and S. Boyd, "Preconditioning in fast dual gradient methods," in *IEEE Conf on Decision & Control*, 2014.
[11] Y. Liu, Y. Xu, and W. Yin, "Acceleration of primal–dual methods by preconditioning and simple subproblem procedures," *Journal of Scientific Computing*, 2021.
[12] Y. Heuts, G. Padilla, and M. Donkers, "An adaptive restart heavy-ball projected primal-dual method for solving constrained linear quadratic optimal control problems," in *IEEE Conf on Decision & Control*, 2021.
[13] J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides, "A condensed and sparse QP formulation for predictive control," in *Proc IEEE Conf on Decision & Control*, 2011.
[14] J. Daafouz and J. Bernussou, "Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties," *Systems & Control Letters*, 2001.
[15] C. Scherer, *Robust mixed control and LPV control with full block scalings*. Advances in Design & Control, Springer-Verlag, 1999.
[16] M. Donkers, "Decentralised robust controller synthesis for discrete-time polytopic systems with additive uncertainty using an iterative-lmi approach," in *Proc American Control Conf*, 2017.
[17] J. Han and R. Skelton, "An LMI optimization approach for structured linear controllers," in *IEEE Conf on Decision & Control*, 2003.
[18] G. Stathopoulos, M. Korda, and C. N. Jones, "Solving the infinite-horizon constrained LQR problem using splitting techniques," *IFAC Proc.*, 2014.
[19] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Found. of Comput. Math.*, 2013.

TABLE I
Comparison between solving the projection using a dedicated solver and using the clipping operator while reaching $\epsilon_p = 10^{-5}$ and $\epsilon_d = 10^{-5}$.

Example 2	GM+SYN	GM+SYN+HB	GM+HB (proj)
CPU Time	$4.82 \cdot 10^{-4}$ s	$4.80 \cdot 10^{-4}$ s	4.7141 s
Iterations	61	97	382
Example 3			
CPU Time	$2.93 \cdot 10^{-4}$ s	$3.83 \cdot 10^{-4}$ s	6.7632 s
Iterations	101	104	376