2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Variant Predictor-Corrector Method for Linear Predictive Control Using Modified Uzawa Algorithm

Tianchen Liu[1], Kushal Chakrabarti[2], Nikhil Chopra[1]

***Abstract*— In this paper, a new variant of the predictor-corrector interior point method (IPM) pipeline is proposed for model predictive control (MPC) problems for linear time-invariant systems, which can be reformulated as quadratic programming (QP) problems. At each iteration in the IPM, finding the search direction via solving a linear system of equations is usually the step with the highest computational cost. A modified Uzawa algorithm is developed to improve the performance in the proposed IPM, which can address the ill-conditioning issue at the late iterations and reduce computational cost. Results of an MPC problem example are presented to show the performance of the proposed pipeline.**

I. Introduction

Model predictive control (MPC) is an optimization-based technique to obtain the optimal control inputs for a system at a certain sampling instant, providing optimal control strategies capable of handling various system requirements [1]. For practical realization of MPC, improving the efficiency of the optimization step is critical. A popular class of algorithms is the interior point methods (IPMs), in which the optimal solution is iteratively approached by moving inside a feasible region. Various IPM frameworks have been developed (see [2], etc. for detailed summaries), which are acknowledged to be more efficient than other algorithms (e.g., active-set) for large-scale problems. IPMs typically converge to the optimum in a small number of iterations. However, at each iteration, a linear system of equations must be solved to find the search direction, referred to as the *Newton step* (Eq. (4)). Hence, the computational cost per iteration can be expensive [3], and it is always of interest to reduce this cost to improve the overall IPM performance.

For solving the Newton step in IPMs, iterative methods are often preferred due to memory [4] and hardware implementation efficiency [5], especially for large-scale problems. However, it is well-known that the matrix in the Newton step becomes extremely ill-conditioned in late iterations [3], resulting in slow convergence for most iterative methods. Hence, preconditioning techniques have been employed, including preconditioners for conjugate gradient (CG) [6], [7], Minimal Residual (MINRES) method [5], [8], Riccati Recursion [9], etc. In this paper, we devise an iterative method based on the Uzawa algorithm for this step. The Uzawa algorithm [10] was designed for solving a linear system of equations with a 2×2 block structure. The algorithm and its variants, including inexact Uzawa [11], [12] and preconditioned Uzawa [12], [13] algorithms, are mainly developed for various saddle point problems.

Among all variants of IPMs, the Mehrotra predictor-corrector (PC) method ([3], [14]), is widely used due to its superior performance. A predictor and a corrector step are computed to find the search direction at each iteration, leading to a faster convergence than other IPM schemes [2]. Using this framework, a new variant PC approach to solve MPC in the quadratic programming (QP) form is proposed in this paper. We derive an *augmented Newton step* (Eq. (8)) for computing the search directions at each iteration, and develop a modified Uzawa algorithm (Algorithm 1) to solve this step. It is shown that the convergence of the approach is not significantly impacted by the condition number of the matrix, which can address the ill-conditioning issue in IPMs. The proposed IPM pipeline is validated by solving a linear MPC problem. To summarize, the main contributions are as follows,

- A new predictor-corrector IPM approach is proposed. A modified Uzawa algorithm is employed to help reduce the computational cost at each iteration.
- Numerical results of an MPC problem are presented to validate the proposed algorithms.

The rest of the paper is organized as follows: Section II describes the background of MPC and IPMs. Section III presents the modified Uzawa algorithm and the overall PC pipeline. A linear MPC example is solved in Section IV to illustrate the proposed algorithm's performance. The conclusion and future work are provided in Section V.

II. Background

A. Model Predictive Control (MPC)

This paper focuses on a linearly constrained MPC problem of a discrete-time linear dynamical system. The

*This work is supported by USDA NIFA Sustainable Agricultural Systems (SAS) Program (Award Number: 20206801231805).
[1] Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA. {tianchen, nchopra}@umd.edu
[2] Division of Data & Decision Sciences, Tata Consultancy Services Research, Mumbai, 400607 India. chakrabarti.k@tcs.com

state equations are formulated as $x_{i+1} = Ax_i + Bu_i$, where $x_i \in \mathbb{R}^{n_x}$, $u_i \in \mathbb{R}^{n_u}$ are the state and input vectors respectively at the i^{th} time instant, and $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ are assumed to be time-invariant. Additionally, linear constraints are enforced for states and inputs:

$$C_1 x_i \leq d_1 \, (i \in \mathbb{Z}_{[1,N]}), \quad C_2 u_i \leq d_2 \, (i \in \mathbb{Z}_{[0,N-1]}),$$

where $C_1 \in \mathbb{R}^{n_{C_1} \times n_x}$ and $C_2 \in \mathbb{R}^{n_{C_2} \times n_u}$ are assumed to be full rank. Redundant rows can be removed if C_1, C_2 are not full rank.

For the MPC problem, optimal control inputs are obtained by solving a nonlinear optimization problem regarding certain objectives within a finite horizon N. The objective is considered to be of a quadratic form,

$$\min_{x_i, u_i} J = \sum_{i=0}^{N-1} \left(\begin{bmatrix} x_i \\ u_i \end{bmatrix}^T \begin{bmatrix} Q_i & S_i^T \\ S_i & R_i \end{bmatrix} \begin{bmatrix} x_i \\ u_i \end{bmatrix} + \begin{bmatrix} q_i \\ r_i \end{bmatrix}^T \begin{bmatrix} x_i \\ u_i \end{bmatrix} \right) + x_N^T Q_N x_N + q_N^T x_N, \quad (1)$$

with $Q_i \in \mathbb{R}^{n_x \times n_x}$, $S_i \in \mathbb{R}^{n_x \times n_u}$, $R_i \in \mathbb{R}^{n_u \times n_u}$, $q_i \in \mathbb{R}^{n_x}$, $r_i \in \mathbb{R}^{n_u}$. For the problem to be convex, it is assumed that $Q_N \succeq \mathbf{0}$, $\begin{bmatrix} Q_i & S_i^T \\ S_i & R_i \end{bmatrix} \succeq \mathbf{0}, \forall i \in \mathbb{Z}_{[0,N-1]}$.

At the time instant t, we define the optimizing variable vector $\xi^{(t)} \in \mathbb{R}^{n_\xi}$, which is the concatenating column vector of variables $x_t, u_t, \ldots, x_{t+N-1}, u_{t+N-1}, x_{t+N}$. For simplicity, we drop the notation t for all variables in $\xi^{(t)}$ as $\xi = [x_0^T, u_0^T, x_1^T, u_1^T, \cdots, u_{N-1}^T, x_N^T]^T$. Hence, the MPC problem follows a QP formulation at instant t,

$$\min_{\xi} J(\xi) = \frac{1}{2}\xi^T \mathbf{Q} \xi + \mathbf{q}^T \xi, \text{ s.t. } \mathbf{A}\xi = \mathbf{b}, \, \mathbf{C}\xi \leq \mathbf{d}. \quad (2)$$

The detailed matrices and vectors can be found in Appendix A. For describing the analysis later, let $n_\xi = N(n_x + n_u) + n_x$, $n_{\text{Eq}} = (N+1)n_x$, and $n_{\text{In}} = N(n_{C_1} + n_{C_2}) + n_{C_1}$ denote the dimension of the optimizing variables, the number of equality and inequality constraints, respectively.

B. IPM & Predictor-Corrector (PC) Method

In this section, a brief introduction of the IPMs and the Mehrotra PC method is presented. The detailed expressions can be found in Appendix A and references such as [2], etc. IPM is a class of algorithms to solve QP (Eq. (2)) efficiently. For convex problems, the solution of the Karush–Kuhn–Tucker (KKT) conditions is the global optimum of Eq. (2), whose formulation is

$$\mathbf{Q}\xi^* + \mathbf{q} + \mathbf{A}^T \eta^* + \mathbf{C}^T \phi^* = \mathbf{0}, \quad (3a)$$

$$\mathbf{A}\xi^* - \mathbf{b} = \mathbf{0}, \quad \mathbf{C}\xi^* - \mathbf{d} + s^* = \mathbf{0}, \quad (3b)$$

$$s_i^* \phi_i^* = 0, \, s_i^*, \phi_i^* \geq 0, \quad \forall i \in \mathbb{Z}_{[1,n_{\text{In}}]}. \quad (3c)$$

Here $(\cdot)^*$ denotes the variables evaluated at the optimum, $s \in \mathbb{R}^{n_{\text{In}}}$ is the slack variable vector, $\eta \in \mathbb{R}^{n_{\text{Eq}}}$ and $\phi \in \mathbb{R}^{n_{\text{In}}}$ are the Lagrange multipliers for the equality and inequality constraints, respectively. Let $z = [\xi^T, \eta^T, s^T, \phi^T]^T$. The concept of the path-following IPMs is to iteratively approach z^* as the solution of Eq. (3) via Newton's method, instead of solving Eq. (3) directly. A centering parameter $\sigma \in [0, 1]$ and a duality measure $\mu = \frac{s^T \phi}{n_{\text{In}}}$ are introduced so that for all i, $s_i \phi_i = \sigma\mu$ is maintained at each iteration. At the k^{th} iteration, let $S_{k-1} := \text{diag}(s_{i,k-1})$ and $\Phi_{k-1} := \text{diag}(\phi_{i,k-1})$, and define $\Theta_k = \text{diag}(\theta_{i,k}) = \Phi_k S_k^{-1}$, i.e., $\theta_{i,k} = \frac{\phi_{i,k}}{s_{i,k}}$. The search direction $\triangle z_k$ is obtained by

$$\begin{bmatrix} \mathbf{Q} & \mathbf{A}^T & \mathbf{0} & \mathbf{C}^T \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Theta_{k-1} & \mathbf{I}_{n_{\text{In}}} \\ \mathbf{C} & \mathbf{0} & \mathbf{I}_{n_{\text{In}}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \triangle\xi \\ \triangle\eta \\ \triangle s \\ \triangle\phi \end{bmatrix}_k = \begin{bmatrix} \mathbf{r}_{11} \\ \mathbf{r}_{12} \\ \mathbf{r}_{21} \\ \mathbf{r}_{22} \end{bmatrix}_k, \quad (4)$$

$$\Leftrightarrow \begin{bmatrix} \mathcal{H}_1 & \mathcal{H}_2^T \\ \mathcal{H}_2 & \mathcal{H}_4 \end{bmatrix} \begin{bmatrix} \triangle z_1 \\ \triangle z_2 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} \quad \Leftrightarrow \quad \mathcal{H}\triangle z = \mathbf{r}, \quad (5)$$

where $\triangle z_1 = [\triangle\xi^T, \triangle\eta^T]^T$, $\triangle z_2 = [\triangle s^T, \triangle\phi^T]^T$. The matrices and vectors are detailed in Appendix A.

Compared with traditional path-following IPMs where $\triangle z$ is solely determined from Eq. (4), the search direction by Mehrotra PC method [14] at each iteration is determined in two steps: i) an affine-scaling 'predictor' direction and ii) a 'corrector' direction. In the predictor step, let the centering parameter $\sigma = 0$, so that $\triangle z_{\text{aff}}$ can be found by solving Eq. (4) as,

$$\begin{bmatrix} \mathbf{Q} & \mathbf{A}^T & \mathbf{0} & \mathbf{C}^T \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Theta & \mathbf{I}_{n_{\text{In}}} \\ \mathbf{C} & \mathbf{0} & \mathbf{I}_{n_{\text{In}}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \triangle\xi_{\text{aff}} \\ \triangle\eta_{\text{aff}} \\ \triangle s_{\text{aff}} \\ \triangle\phi_{\text{aff}} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{11} \\ \mathbf{r}_{12} \\ -\phi \\ \mathbf{r}_{22} \end{bmatrix}. \quad (6)$$

The step lengths $\alpha_{s,\text{aff}}$ and $\alpha_{\phi,\text{aff}}$ along $\triangle z_{\text{aff}}$ are then calculated (see [2]), and we obtain $s_{\text{aff}} = s + \alpha_{s,\text{aff}} \triangle s_{\text{aff}}$ and $\phi_{\text{aff}} = \phi + \alpha_{\phi,\text{aff}} \triangle\phi_{\text{aff}}$. Next, in the corrector step, let $\sigma = (\mu_{\text{aff}}/\mu)^3$, where $\mu = (s^T\phi)/n_{\text{In}}$ and $\mu_{\text{aff}} = ((s_{\text{aff}})^T(\phi_{\text{aff}}))/n_{\text{In}}$. The search direction $\triangle z$ of the corrector step is obtained as,

$$\mathcal{H}\triangle z = \begin{bmatrix} \mathbf{r}_{11} \\ \mathbf{r}_{12} \\ -\phi + \sigma\mu S^{-1}\mathbf{1}_{n_{\text{In}}} - \triangle S_{\text{aff}} \triangle\Phi_{\text{aff}} S^{-1}\mathbf{1}_{n_{\text{In}}} \\ \mathbf{r}_{22} \end{bmatrix}, \quad (7)$$

where $\triangle S_{\text{aff}} = \text{diag}(\triangle s_{\text{aff}})$ and $\triangle\Phi_{\text{aff}} = \text{diag}(\triangle\phi_{\text{aff}})$. The step length α along $\triangle z$ from Eq. (7) can be found (see [2]), and z is updated by $z = z + \alpha\triangle z$.

Note that the Newton step is solved twice (Eq. (6), (7)) at each iteration in PC method. However, only the terms $\mathbf{r}_{21}$, i.e., the third sub-block of the right-hand side vector $\mathbf{r}$, differ between the equations. Moreover, the step lengths are only determined by $\triangle s$ and $\triangle\phi$, i.e., $\triangle z_2$. Hence, the objective is to accelerate the step of solving Eq. (6) and (7), which will be presented in the next section.

III. Proposed Approach

A. Augmented Newton Step

One common approach in IPMs is to eliminate $\triangle s$ or $\triangle\phi$ in Eq. (4) [3], leading to an 'augmented system' to solve $\triangle\xi$ and $\triangle\eta$. However, as discussed above, $\triangle s$

and $\triangle\phi$ are the terms of interest in the PC method, while $\triangle\xi$ and $\triangle\eta$ are not required until the end of the iteration. Hence, we can eliminate $\triangle z_1$ in the Newton step to obtain

$$\begin{aligned}(\mathcal{H}_4 - \mathcal{H}_2\mathcal{H}_1^{-1}\mathcal{H}_2^T)\triangle z_2 &= \mathbf{r}_2 - \mathcal{H}_2\mathcal{H}_1^{-1}\mathbf{r}_1 \\ \Leftrightarrow \qquad \widehat{\mathcal{H}}_2\triangle z_2 &= \widetilde{\mathbf{r}}_2, \end{aligned} \tag{8}$$

where

$$\widehat{\mathcal{H}}_2 = \begin{bmatrix}\Theta & \mathbf{I}_{n_{\text{In}}} \\ \mathbf{I}_{n_{\text{In}}} & -\mathbf{C}M_1\mathbf{C}^T\end{bmatrix} =: \begin{bmatrix}\Theta & \mathbf{I}_{n_{\text{In}}} \\ \mathbf{I}_{n_{\text{In}}} & -T\end{bmatrix}, \tag{9a}$$

$$\widetilde{\mathbf{r}}_2 = \begin{bmatrix}\mathbf{r}_{21} \\ \mathbf{r}_{22} - \mathbf{C}M_1\mathbf{r}_{11} - \mathbf{C}M_2\mathbf{r}_{12}\end{bmatrix} =: \begin{bmatrix}\widetilde{\mathbf{r}}_{21} \\ \widetilde{\mathbf{r}}_{22}\end{bmatrix}. \tag{9b}$$

Here, M_1 and M_2 are the $(1,1)$ and $(1,2)$ sub-blocks of $\mathcal{H}_1^{-1}$ with detailed expressions in Appendix A. Eq. (8) is referred to as the *augmented Newton step*. It is seen that Eq. (6) and (7) follow this augmented Newton step with a slightly different $\widetilde{\mathbf{r}}_2$. After finding $\triangle z_2$ in Eq. (8), $\triangle z_1$ is obtained by

$$\triangle z_1 = \mathcal{H}_1^{-1}(\mathbf{r}_1 - \mathcal{H}_2^T\triangle z_2), \tag{10}$$

and $\triangle z = [\triangle z_1^T\ \triangle z_2^T]^T$ is the search direction solution. Note that many matrices are constant throughout the process, so they only need to be computed once.

Due to the complementary slackness (Eq. (3c)), either one of ϕ_i and s_i will be approximately 0. The i^{th} inequality constraint is inactive if $\phi_i \to 0$, otherwise active if $s_i \to 0$. Hence, $\theta_i = \frac{\phi_i}{s_i}$ will be either very small (inactive) or very large (active), resulting in an ill-conditioned Θ in $\widehat{\mathcal{H}}_2$ as IPM proceeds close to the optimum. This can cause a slow convergence for many iterative methods to solve Eq. (8). Hence, a modified Uzawa algorithm is presented to avoid this issue.

B. A Modified Uzawa Algorithm

To obtain $\triangle z_2$ for Eq. (8), the standard Uzawa algorithm works as follows. Given an initial value of $\triangle z_{22,0}$, the variables are updated at the k^{th} iteration as,

$$\begin{aligned}\triangle z_{21,k+1} &= \Theta^{-1}(\widetilde{\mathbf{r}}_{21} - \triangle z_{22,k}) \\ \triangle z_{22,k+1} &= \triangle z_{22,k} + \alpha_k(\triangle z_{21,k+1} - T\triangle z_{22,k} - \widetilde{\mathbf{r}}_{22})\end{aligned}$$

until convergence, where α_k is the step size fitting the convergence condition. However as discussed previously, at the late IPM iterations, the condition number of Θ is large. In this case, small step size α_k is required to guarantee convergence of the standard Uzawa algorithm, which leads to a slow convergence.

Therefore, to reduce the influence of the condition number of Θ on the convergence, we modify the standard Uzawa algorithm with an exact step size calculation, inspired by [15], to solve Eq. (8). First, we eliminate $\triangle z_{21}$ in Eq. (8) and obtain,

$$(\Theta^{-1} + T)\triangle z_{22} = \Theta^{-1}\widetilde{\mathbf{r}}_{21} - \widetilde{\mathbf{r}}_{22} \Leftrightarrow (\Theta^{-1} + T)\triangle z_{22} = \widehat{\mathbf{r}}. \tag{11}$$

Let $\triangle z_{22}^*$ denote the solution of Eq. (11), then it is clear that $\triangle z_{22}^*$ is also the solution to the following least squares problem $\Gamma(\triangle z_{22})$,

$$\min_{\triangle z_{22}} \Gamma(\triangle z_{22}) := \frac{1}{2}\|(\Theta^{-1} + T)\triangle z_{22} - \widehat{\mathbf{r}})\|. \tag{12}$$

Hence, we can find $\triangle z_{22}^*$ by solving Eq. (12). At the k^{th} iteration, $g_k = \triangle z_{21,k} - T\triangle z_{22,k} - \widetilde{\mathbf{r}}_{22}$ is a descent direction [15]. Thus, along the g_k direction, the exact step size α_k to minimize Eq. (12) is $\alpha_k = (g_k^T(\Theta^{-1} + T)g_k)/(g_k^T(\Theta^{-1} + T)^2 g_k)$. With this information, the modified Uzawa algorithm to solve Eq. (8) is presented in Algorithm 1, where vectors q_k, p_k are introduced to replace the computationally expensive matrix multiplication operations.

Algorithm 1 Modified Uzawa Algorithm

Select an initial $\triangle z_{22,0}$. Set $k = 0$.
Compute $\triangle z_{21,0} = \Theta^{-1}(\widetilde{\mathbf{r}}_{21} - \triangle z_{22,0})$.
while not converged **do**
  Compute $g_k = \triangle z_{21,k} - T\triangle z_{22,k} - \widetilde{\mathbf{r}}_{22}$.
  Compute $q_k = \Theta^{-1}g_k$, $p_k = q_k + Tg_k$, $\alpha_k = \frac{g_k^T p_k}{p_k^T p_k}$.
  $\triangle z_{22,k+1} = \triangle z_{22,k} + \alpha_k g_k$.
  $\triangle z_{21,k+1} = \triangle z_{21,k} - \alpha_k q_k$.
  Stop if $\|[\alpha_k g_k^T,\ \alpha_k q_k^T]^T\|_\infty < \varepsilon_{\text{tol}}$.
end while

Computational complexity: To solve Eq. (8), a direct method to find $\triangle s$ and $\triangle\phi$ can take up to $8n_{\text{In}}^3 + 2n_{\text{In}}^2 - n_{\text{In}}$ floating-point operations. Using Algorithm 1, the total computational cost for obtaining $\triangle s$ and $\triangle\phi$ is $\mathcal{O}((2n_{\text{In}}^2 + 10n_{\text{In}})\bar{N}_{\text{it}})$, where $\bar{N}_{\text{it}}$ is the average number of iterations to converge for Algorithm 1. Hence, for solving Eq. (8), Algorithm 1 and existing iterative methods can be less computationally expensive when $\bar{N}_{\text{it}} < n_{\text{In}}$. However, it is well-known that $\bar{N}_{\text{it}}$ is negatively impacted by the condition number of $\mathcal{H}$ using iterative methods like MINRES. Hence, Algorithm 1 is preferable given $\bar{N}_{\text{it}}$ is not significantly impacted by condition numbers, which is shown next.

Convergence analysis: For the convergence analysis of Algorithm 1, we require a prior result from [15].

Lemma 1. Consider Lemma 3.1 [15] to the augmented Newton step (Eq. (8)) with $\widehat{\mathcal{H}}_2$ matrix structure. Suppose that for all $v \in \mathbb{R}^{n_{\text{In}}}$, a constant $\beta > 0$ exists such that $\langle(\Theta^{-1} + T)v, v\rangle \geq \beta\|v\|^2$ is satisfied, then we have $\langle(\Theta^{-1} + T)v, v\rangle \geq \gamma^{-2}\beta\|v\|^2$ for all $\gamma \geq \frac{\|\Theta\|}{\lambda_{\min}(\Theta)}$. ■

In fact, Θ is a diagonal matrix with all positive entries, so Lemma 1 clearly holds in our problem since $\gamma \geq 1$. Next, convergence analysis of Algorithm 1 is shown.

Theorem 1. Consider Algorithm 1 with any $\triangle z_{22,0} \in \mathbb{R}^{n_{\text{In}}}$. Assume the stabilized condition [15]

$$\langle(\Theta^{-1} + T)v, v\rangle \geq \beta\|v\|^2, \quad \forall v \in \mathbb{R}^{n_{\text{In}}} \tag{13}$$

is satisfied for some constant $\beta > 0$. Then, $\triangle z_{22,k}$ converges to $\triangle z_{22}^*$ with a linear rate as,

$$\frac{\|\triangle z_{22,k+1} - \triangle z_{22}^*\|_M^2}{\|\triangle z_{22,k} - \triangle z_{22}^*\|_M^2} = \frac{\Gamma(\triangle z_{22,k+1})}{\Gamma(\triangle z_{22,k})} \leq 1 - c_0 < 1, \quad (14)$$

where $c_0 = \frac{\beta^2}{\gamma^4 \|(\Theta^{-1}+T)\|^2}$, and $M = (\Theta^{-1} + T)^2$. Hence, $[\triangle z_{21,k}^T, \triangle z_{22,k}^T]^T$ from Algorithm 1 converges to the solution $\triangle z_2$ of the augmented Newton step (8).
Proof: The proof is deferred to Appendix B. ■

Remark 1: At late iterations of the IPMs, some entries of Θ are large while others are nearly 0, leading to the condition number that can cause slow convergence for many iterative methods. Theorem 1 indicates that the convergence rate of the proposed algorithm is close to 1 under this scenario, since c_0 becomes small. Moreover, the algorithm requires a low computational cost at each iteration, leading to an overall improved performance.

C. New Variant PC Method

For the overall PC pipeline, we develop an additional step to simplify the calculations of the search directions using Algorithm 1. In the predictor step, $\triangle s_{\text{aff}}$ and $\triangle \phi_{\text{aff}}$ are solved by $\widetilde{\mathbf{r}}_{21} = -\phi$. In the corrector step, $\widetilde{\mathbf{r}}_{21} = -\phi + \sigma\mu S^{-1}\mathbf{1}_{n_{\text{In}}} - \triangle S_{\text{aff}} \triangle \Phi_{\text{aff}} S^{-1}\mathbf{1}_{n_{\text{In}}}$. Let $\triangle s = \triangle s_{\text{aff}} + \triangle s_e$ and $\triangle \phi = \triangle \phi_{\text{aff}} + \triangle \phi_e$, then $\triangle s_e$ and $\triangle \phi_e$ can be solved by

$$\widehat{\mathcal{H}}_2 \begin{bmatrix} \triangle s_e \\ \triangle \phi_e \end{bmatrix} = \begin{bmatrix} \sigma\mu S^{-1}\mathbf{1}_{n_{\text{In}}} - \triangle S_{\text{aff}} \triangle \Phi_{\text{aff}} S^{-1}\mathbf{1}_{n_{\text{In}}} \\ \mathbf{0} \end{bmatrix}. \quad (15)$$

Algorithm 2 Variant PC Method

Select an initial $z_0 = [\xi_0^T, \eta_0^T, s_0^T, \phi_0^T]^T$ with $s_0, \phi_0 > 0$. Set $k = 0$.
while not converged **do**
  $z_k = [\xi_k^T, \eta_k^T, s_k^T, \phi_k^T]^T$ for the k^{th} iteration.
  $\mu = (s_k^T \phi_k)/n_{\text{In}}$.
  (Prediction Step)
    Obtain $\triangle s_{\text{aff}}, \triangle \phi_{\text{aff}}$ by (8) with $\widetilde{\mathbf{r}}_{21} = \phi$ using **Algorithm 1**.
    Find step lengths $\alpha_{s,\text{aff}}$, $\alpha_{\phi,\text{aff}}$ and scale down as $\alpha_{s,\text{aff}} = \beta_{\text{s}}\alpha_{s,\text{aff}}$, $\alpha_{\phi,\text{aff}} = \beta_{\text{s}}\alpha_{\phi,\text{aff}}$.
    $s_{\text{aff}} = s + \alpha_{s,\text{aff}} \triangle s_{\text{aff}}, \quad \phi_{\text{aff}} = \phi + \alpha_{\phi,\text{aff}} \triangle \phi_{\text{aff}}$.
    $\mu_{\text{aff}} = (s_{\text{aff}}^T \phi_{\text{aff}})/n_{\text{In}}, \quad \sigma = (\mu_{\text{aff}}/\mu)^3$.
  (Correction Step)
    Obtain $\triangle s_e, \triangle \phi_e$ by (15) using **Algorithm 1**.
    $\triangle s = \triangle s_{\text{aff}} + \triangle s_e$, $\triangle \phi = \triangle \phi_{\text{aff}} + \triangle \phi_e$.
    Find step length α and scale down as $\alpha = \beta_{\text{s}}\alpha$.
  Obtain $\triangle z_1$ by Eq. (10).
  Update z by $z_{k+1} = z_k + \alpha \triangle z$.
  Test stopping criteria.
end while

The complete PC IPM pipeline is outlined in Algorithm 2. Note that a small offset exists between $\triangle z_2$ obtained from Algorithm 1 and the actual solution of Eq. (8), which is of the magnitude of ε_{tol}. If the step lengths $\alpha_{s,\text{aff}}$, $\alpha_{\phi,\text{aff}}$ and α are used in the new IPM pipeline, this error may update the variables out of the interior region and thus leads to a sub-optimal solution to the problem. Hence, a parameter $\beta_{\text{s}} \in (0, 1)$ is introduced for scaling down the step lengths to avoid this undesired result. The selection of β_{s} is dependent on ε_{tol} in Algorithm 1 and the matrices of the problem. If β_{s} is a small number, the IPM proceeds more conservatively. Finally, the stopping criteria consist of i) the primal feasibility, ii) the dual feasibility, and iii) the duality gap [16], [17].

Remark 2: Since the majority of computations occur in Algorithm 1, the proposed approach is expected to perform better for the problems where the number of the inequality constraints n_{In} is not significantly larger than the total number of the variables and equality constraints $n_\xi + n_{\text{Eq}}$. Otherwise, it could be computationally cheaper to eliminate $\triangle z_2$ to calculate $\triangle z_1$ as the augmented Newton step.

IV. Numerical Example

In this section, the proposed approach is employed to solve an MPC problem of controlling a servo motor [18]. The continuous-time system model is,

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ -128 & -2.5 & 6.4 & 0 \\ 0 & 0 & 0 & 1 \\ 128 & 0 & -6.4 & -10.2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t),$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1282 & 0 & -64.0 & 0 \end{bmatrix} x(t). \quad (16)$$

The system model is then discretized at $dt = 0.05$ s using the first-order Euler discretization. The objective is to maintain the motor position y_1 to a desired angle at 30°, subject to the constraints that the shaft torque $|y_2| \leq 78.5$ N · m, and the input voltage $|u| \leq 220$V. The weighting matrices are $Q_i = \text{diag}([10^3, 0, 0, 0])$ and $R_i = 10^{-4}$. The initial states are assumed to be $x_0 = \mathbf{0}$.

Four approaches are coded in MATLAB to solve this MPC problem, including the nominal PC method [3], and the pipeline of Algorithm 2 with the augmented Newton step solved by i) MINRES [19], ii) LU decomposition, and iii) the modified Uzawa method. The window size N is selected from 30 to 90 with an increment of 10, and the number of time steps in the overall MPC problem equals $N + 10$. The tolerance values are 10^{-6}. As mentioned earlier, since a minor error in the inner loop results will be created when using MINRES and the modified Uzawa method, the scaling parameter $\beta_{\text{s}} = 0.8$ is selected for both approaches.

Fig. 1 compares the total computational time for the overall MPC problem and the average time at each instant. As N increases, Algorithm 1 performs better than MINRES. The nominal PC method runs faster than

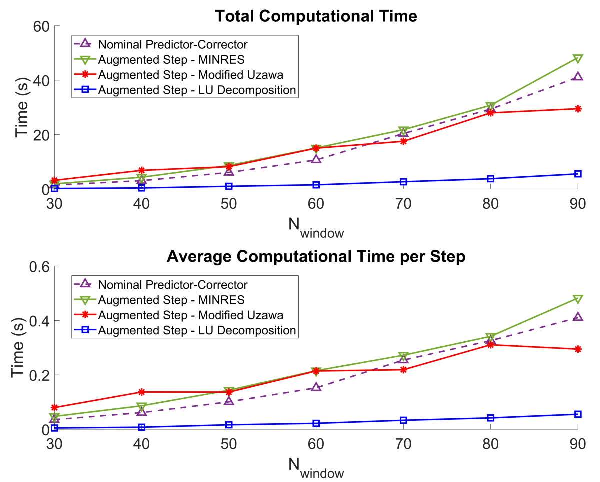

Fig. 1. Computational Time for MPC Example: (top) total time for solving all time instants, (bottom) average time for optimal control problem at each time instant.

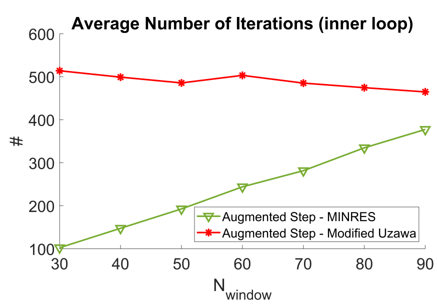

Fig. 2. Average Number of Iterations to Converge for Solving Augmented Newton Step.

Algorithm 2 for relatively small N, but its performance worsens as N becomes large. Fig. 2 presents the average number of iterations for convergence of the augmented Newton step using MINRES and our proposed method. In contrast to MINRES, as the dimension increases, the average number of iterations using Algorithm 1 maintains approximately the same, as expected. Overall, the computational time is affected by i) the number of iterations for convergence, and ii) computational cost at each iteration. While the modified Uzawa method needs more iterations to converge, the computational cost at each iteration is much less than that of MINRES. Hence, our proposed approach requires less computational cost.

Remark 3: It is worth noting that if LU decomposition is employed in Algorithm 2, the computational time is faster than the MATLAB built-in IPM solver (i.e., 'fmincon'), and faster than all the iterative methods presented. However, the proposed iterative method can be preferable due to a low memory requirement and parallel computing capability. For example, Algorithm 1 can be calculated in a distributed manner by splitting the matrix-vector products $T\triangle z_{22}$ and Tv, and the overall computational time can be reduced accordingly. Moreover, as discovered in the inexact IPM framework [20], the exact solution for the Newton step may not be needed at each iteration; i.e., a pre-mature termination of Algorithm 1, which will be studied in future.

In this example, we present that the modified Uzawa method can outperform other iterative methods like MINRES for solving (Eq. (8)), so that performance of the overall IPM can be improved. Other methods for linear MPC have not been implemented currently. Additionally, the algorithm is implemented in MATLAB, thus is not expected to outperform the state-of-the-art QP solvers coded in C++, such as PIQP [17], HPIPM [21], etc. The C++ implementation (with parallel computation by GPU) will be addressed in our future work.

V. Conclusion

This paper proposes a new variant of the predictor-corrector IPM algorithm for solving a linear MPC problem. A new augmented Newton step is formulated to potentially reduce the high computational cost of finding the search direction at each iteration. A modified Uzawa algorithm is then employed to solve this step, whose convergence is not significantly impacted by the ill-conditioning issue at late iterations. Numerical results of a linear MPC problem are presented to validate the proposed algorithm. In future, we will implement the complete IPM pipeline in C++ to compare its performance with state-of-the-art solvers.

Appendix

A. Notation

$\mathbb{R}$ denotes the real number set. $\mathbb{Z}_{[a,b]}$ denotes the integer set from a to b, where integers $a \leq b$. $\mathbb{R}^n$ is the n-dimensional real vector set. $\mathbf{1}_n \in \mathbb{R}^n$ is the vector with all 1 entries. $\mathbb{R}^{n\times m}$ is the $n \times m$-dimensional real matrix set. $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ denotes the $n\times n$ identity matrix. '$\otimes$' refers to the Kronecker product. '$\succ / \succeq \mathbf{0}$' refers to positive definite or semidefinite. $\|v\|$ and $\|v\|_\infty$ denote the L_2 and L_∞ norm of a vector v. $\|M\|$ denotes the induced L_2 norm of a matrix M. $\|v\|_M = \sqrt{v^T M v}$ for a vector v and a symmetric positive definite matrix M. $\lambda_{\min}(M)$ is the smallest eigenvalue of M.

The matrices and vectors in Eq. (2) are as follows,

$$\mathbf{Q} = 2\begin{bmatrix} \begin{matrix} Q_0 & S_0^T \\ S_0 & R_0 \end{matrix} & & & \\ & \ddots & & \\ & & \begin{matrix} Q_{N-1} & S_{N-1}^T \\ S_{N-1} & R_{N-1} \end{matrix} & \\ & & & Q_N \end{bmatrix}, \ \mathbf{q} = \begin{bmatrix} q_0 \\ r_0 \\ \vdots \\ q_{N-1} \\ r_{N-1} \\ q_N \end{bmatrix},$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_N \otimes \begin{bmatrix} -\mathbf{I}_{n_x} & \mathbf{0} \\ A & B \\ \mathbf{0} & \end{bmatrix} & \begin{matrix} \mathbf{0} \\ \mathbf{0} \end{matrix} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{n_x} \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} x_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_N \otimes \begin{bmatrix} C_1 & \mathbf{0} \\ \mathbf{0} & C_2 \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & C_1 \end{bmatrix}, \mathbf{d} = \begin{bmatrix} \mathbf{1}_N \otimes \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \\ d_1 \end{bmatrix},$$

with the dimensions $\mathbf{Q} \in \mathbb{R}^{n_\xi \times n_\xi}$, $\mathbf{q} \in \mathbb{R}^{n_\xi}$, $\mathbf{A} \in \mathbb{R}^{n_{\text{Eq}} \times n_\xi}$, $\mathbf{b} \in \mathbb{R}^{n_{\text{Eq}}}$, $\mathbf{C} \in \mathbb{R}^{n_{\text{In}} \times n_\xi}$, $\mathbf{d} \in \mathbb{R}^{n_{\text{In}}}$.

The right-hand side of Eq. (4) are as follows,

$$\mathbf{r}_{1,k} = \begin{bmatrix} \mathbf{r}_{11,k} \\ \mathbf{r}_{12,k} \end{bmatrix} = \begin{bmatrix} -\mathbf{Q}\xi_{k-1} - \mathbf{q} - \mathbf{A}^T \eta_{k-1} - \mathbf{C}^T \phi_{k-1} \\ -\mathbf{A}\xi_{k-1} + \mathbf{b} \end{bmatrix},$$
$$\mathbf{r}_{2,k} = \begin{bmatrix} \mathbf{r}_{21,k} \\ \mathbf{r}_{22,k} \end{bmatrix} = \begin{bmatrix} -\phi_{k-1} + \sigma_k \mu_k S_{k-1}^{-1} \mathbf{1}_{n_{\text{In}}} \\ -\mathbf{C}\xi_{k-1} + \mathbf{d} - s_{k-1} \end{bmatrix}.$$

The decomposed sub-blocks of $\mathcal{H}$ in Eq. (5) are,

$$\mathcal{H}_1 = \begin{bmatrix} \mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix}, \mathcal{H}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} \end{bmatrix}, \mathcal{H}_4 = \begin{bmatrix} \Theta_{k-1} & \mathbf{I}_{n_{\text{In}}} \\ \mathbf{I}_{n_{\text{In}}} & \mathbf{0} \end{bmatrix}.$$

A closed-form solution of M_1 and M_2 in Eq. (9) can be found [22] as,

$$M_1 = \mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{Q}^{-1}, \tag{17a}$$
$$M_2 = \mathbf{Q}^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^T)^{-1}, \tag{17b}$$

so $T = \mathbf{C}M_1\mathbf{C}^T$ in Eq. (9) is symmetric.

B. Proof of Theorem 1

This proof follows Theorem 3.2 in [15] with $\Theta^{-1} + T$ being a symmetric matrix. Let $\widehat{\Theta} = \Theta^{-1} + T$. Since $\triangle z_{22}^* = \widehat{\Theta}^{-1}\widehat{\mathbf{r}}$ is the solution for Eq. (12), then at the k^{th} iteration,

$$\begin{aligned} & \frac{1}{2}\|\triangle z_{22,k} - \triangle z_{22}^*\|_M^2 \\ = \quad & \frac{1}{2}(\triangle z_{22,k} - \triangle z_{22}^*)^T \widehat{\Theta}^2 (\triangle z_{22,k} - \triangle z_{22}^*) \\ = \quad & \frac{1}{2}(\widehat{\Theta}\triangle z_{22,k} - \widehat{\Theta}\triangle z_{22}^*)^T (\widehat{\Theta}\triangle z_{22,k} - \widehat{\Theta}\triangle z_{22}^*) \\ = \quad & \frac{1}{2}(\widehat{\Theta}\triangle z_{22,k} - \widehat{\mathbf{r}})^T (\widehat{\Theta}\triangle z_{22,k} - \widehat{\mathbf{r}}) = \Gamma(\triangle z_{22,k}). \end{aligned}$$

Given the selected descent direction g_k and step size α_k,

$$\begin{aligned} & \frac{\|\triangle z_{22,k+1} - \triangle z_{22}^*\|_M^2}{\|\triangle z_{22,k} - \triangle z_{22}^*\|_M^2} = \frac{\Gamma(\triangle z_{22,k} + \alpha_k g_k)}{\Gamma(\triangle z_{22,k})} \\ = \quad & \frac{\frac{1}{2} g_k^T \widehat{\Theta}^2 g_k \alpha_k^2 - 2 g_k^T \widehat{\Theta} g_k \alpha_k + g_k^T g_k}{\frac{1}{2}(\widehat{\Theta}\triangle z_{22,k} - \widehat{\mathbf{r}})^T (\widehat{\Theta}\triangle z_{22,k} - \widehat{\mathbf{r}})} \\ = \quad & 1 - \frac{(g_k^T \widehat{\Theta} g_k)^2}{(g_k^T \widehat{\Theta}^2 g_k)(g_k^T g_k)}. \end{aligned} \tag{18}$$

For the convergence of the proposed algorithm, we have $\frac{\|\triangle z_{22,k+1} - \triangle z_{22}^*\|_M^2}{\|\triangle z_{22,k} - \triangle z_{22}^*\|_M^2} < 1$. Hence,

$$\begin{aligned} & \frac{(g_k^T \widehat{\Theta} g_k)^2}{(g_k^T \widehat{\Theta}^2 g_k)(g_k^T g_k)} \leq 1 \Rightarrow \frac{(\widehat{\Theta} g_k, g_k)^2}{\|\widehat{\Theta} g_k\|^2 \|g_k\|^2} \leq 1 \\ \Leftrightarrow \quad & \frac{\gamma^{-4}\beta^2\|g_k\|^2}{\|\widehat{\Theta} g_k\|^2} \leq 1 \Leftrightarrow \frac{\beta^2}{\gamma^4\|\widehat{\Theta}\|^2} \leq 1. \end{aligned} \tag{19}$$

γ is a parameter satisfying $\gamma \geq \frac{\|\Theta\|}{\lambda_{\min}(\Theta)}$, and $\beta \leq \lambda_{\min}(\widehat{\Theta})$. Hence, the proposed algorithm is proved to converge with the notation $c_0 = \frac{\beta^2}{\gamma^4\|\widehat{\Theta}\|^2}$. ■

References

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

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

[3] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer, 1999.

[4] M. D'Apuzzo, V. De Simone, and D. di Serafino, "On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods," *Comput. Optim. Appl.*, vol. 45, no. 2, pp. 283–310, Mar. 2010.

[5] A. Shahzad, E. C. Kerrigan, and G. A. Constantinides, "A fast well-conditioned interior point method for predictive control," in *49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 508–513.

[6] L. Bergamaschi, J. Gondzio, and G. Zilli, "Preconditioning Indefinite Systems in Interior Point Methods for Optimization," *Comput. Optim. Appl.*, vol. 28, no. 2, pp. 149–171, Jul. 2004.

[7] L. Schork and J. Gondzio, "Implementation of an interior point method with basis preconditioning," *Math. Prog. Comp.*, vol. 12, no. 4, pp. 603–635, Dec. 2020.

[8] A. Shahzad, E. C. Kerrigan, and G. A. Constantinides, "Preconditioners for inexact interior point methods for predictive control," in *Proceedings of the 2010 American Control Conference*, Jun. 2010, pp. 5714–5719.

[9] M. R. Wahlgreen and J. B. Jørgensen, "On the implementation of a preconditioned Riccati recursion based primal-dual interior-point algorithm for input constrained optimal control problems," *IFAC-PapersOnLine*, vol. 55, no. 7, pp. 346–351, 2022.

[10] H. Uzawa, "Iterative methods for concave programming," *Studies in Linear and Nonlinear Prog.*, vol. 6, pp. 154–165, 1958.

[11] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, "Analysis of the Inexact Uzawa Algorithm for Saddle Point Problems," *SIAM J. Numer. Anal.*, vol. 34, no. 3, pp. 1072–1092, Jun. 1997.

[12] H. C. Elman and G. H. Golub, "Inexact and Preconditioned Uzawa Algorithms for Saddle Point Problems," *SIAM J. Numer. Anal.*, vol. 31, no. 6, pp. 1645–1661, Dec. 1994.

[13] X. Chen, "On preconditioned Uzawa methods and SOR methods for saddle-point problems," *J. of Comp. and Applied Math.*, vol. 100, no. 2, pp. 207–224, Dec. 1998.

[14] S. Mehrotra, "On the Implementation of a Primal-Dual Interior Point Method," *SIAM J. Optim.*, vol. 2, no. 4, pp. 575–601, 1992.

[15] Z. Xu, T. Jiang, and L. Gao, "A new Uzawa-exact type algorithm for nonsymmetric saddle point problems," *arXiv preprint arXiv:1802.04135*, 2018.

[16] B. O'Donoghue, "Operator splitting for a homogeneous embedding of the linear complementarity problem," *SIAM J. Optim.*, vol. 31, no. 3, pp. 1999–2023, 2021.

[17] R. Schwan, Y. Jiang, D. Kuhn, and C. N. Jones, "PIQP: A proximal interior-point quadratic programming solver," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 1088–1093.

[18] A. Bemporad and E. Mosca, "Fulfilling hard constraints in uncertain linear systems by reference managing," *Automatica*, vol. 34, no. 4, pp. 451–461, 1998.

[19] C. C. Paige and M. A. Saunders, "Solution of sparse indefinite systems of linear equations," *SIAM J. Numer. Anal.*, vol. 12, no. 4, pp. 617–629, 1975.

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

[21] G. Frison and M. Diehl, "Hpipm: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[22] T.-T. Lu and S.-H. Shiou, "Inverses of 2 × 2 block matrices," *Comp. & Math. with Appl.*, vol. 43, no. 1, pp. 119–129, Jan. 2002.