Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-21 (2021) 67–72

# Polynomially Bounded Linear Complementarity Problem-based Solvers for Model Predictive Control ⋆

Connor R. Stine *, Ambrose A. Adegbege *

* Department of Electrical and Computer Engineering, The College of New Jersey, 2000 Pennington Road, Ewing, NJ 08628 (e-mail: stinec3@tcnj.edu, adegbega@tcnj.edu).

**Abstract:** We present a computational complexity certificate for a special type of Linear Complementarity problem (LCP) arising in Constrained Linear Quadratic Regulator (LQR) based Model Predictive Control (MPC). By exploiting the special structure of the LCP, we provide sufficient conditions for the LCP to be solvable in a finite number of pivots when processed using Lemke's complementarity pivoting algorithm. The finite time termination is associated with the existence of a covering vector whose length determines the number of pivot operations needed, and guarantees an upper bound on computational complexity. This result is stronger than earlier assumed in the literature and provides a very promising solution technique for use in embedded MPC applications.



*Keywords:* Model Predictive Control, Quadratic Programming, Linear Complementarity Problem, Constrained Control, Constrained LQR

## 1. INTRODUCTION

Model Predictive Control (MPC) is an advanced control strategy that involves online computation of a dynamic program during each plant update (Maciejowski, 2002). MPC consists of an optimizer and a state estimator, which allows the current state of the system to be read into the controller in order to find an optimal control action. The optimizer reads in a predetermined cost function, the known constraints of the problem, and the current state of the system. It then solves the cost function using a quadratic program for the optimal system change. The solution is used to drive the plant output as close to the desired value as possible while accounting for constraints.

Previously, due to huge storage requirement and long computation times, MPC could only be applied in systems such as industrial plants that operate in relatively large time frames of seconds or minutes. Large computation times have resulted mostly from the optimization problem being solved entirely online. In more recent years, fast computation methods have been developed to extend the range of applications of MPC (Wang and Boyd, 2009; Adegbege and Nelson, 2016). One known method is to compute the control output for all possible states in advance to then be later called on as a look up table (Bemporad et al., 2002). This method is appropriate for small scale problems with few constraints and small horizons, however it becomes increasingly inefficient as the problem size increases (Wang and Boyd, 2009).

For most applications of MPC, it is vital to be able to predetermine an upper bound on the computational load of the controller. This guarantee of stable performance improves the safety of the overall system and allows processor specifications to be determined. Recent work in this area has focused on streamlining interior-point algorithms, active set algorithms, and dual-gradient methods for embedded MPC implementation. While the approaches have been largely successful, there is still the need to increase the number of tools available for users and designers of embedded MPC.

In this work, we focus on the Linear Complementarity Problem (LCP) reformulation of the constrained Linear Quadratic Regulator (LQR) problem. We construct a procedure with guaranteed linear termination based on the existence of a special vector employed in Lemke's complementarity pivot scheme. The finite time termination is associated with the existence of a covering vector, whose length determines the number of pivot operations. Similar ideas have recently been pursued in (Okawa and Nonaka, 2021). However, the search algorithm provided places too tight of a restriction on the matrix $M$ from the LCP$(M, q)$ resulting in possible solutions being lost. The strategy employed in this paper corrects this and displays the full range of solutions.

The outline of the paper is as follows. In Section 2, we focus on the MPC problem and its condensed and non-condensed reformulation as quadratic programming problems. In Section 3, we recall some existing results on LCP solution of a class of quadratic problems. In Section 4, we state our main results regarding the existence of a $l$-step vector for the class of LCPs arising from MPC. Finally, we report some experimental results in Section 5.

⋆ This work is supported through the New Jersey Space Grant Consortium (NJSGC) and the TCNJ's Mentored Undergraduate Student Experience (MUSE)


10.1016/j.ifacol.2021.12.012

## 2. MPC PROBLEM FORMULATION

### 2.1 Constrained Linear Quadratic Regulator

We consider the model predictive control formulation in the form of a constrained Linear Quadratic Regulator (LQR) stated as:

$$\min \; x_N^T P x_N + \sum_{k=0}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) \tag{1a}$$

subject to

$$x_0 = \hat{x} \tag{1b}$$

$$x_{k+1} = Ax_k + Bu_k \text{ for } k = 0, 1, 2, \cdots, N-1 \tag{1c}$$

$$\underline{x} \leq x_k \leq \overline{x} \text{ for } k = 0, 1, 2, \cdots, N-1 \tag{1d}$$

$$\underline{u} \leq u_k \leq \overline{u} \text{ for } k = 0, 1, 2, \cdots, N-1 \tag{1e}$$

where $x_k \in \mathbb{R}^n$ is the state vector at sample instant $k$, $u_k \in \mathbb{R}^m$ is the input vector at sample instant $k$, $N$ is the prediction horizon, the weights $Q$ and $R$ are positive definite matrices, and $P$ is a stabilizing solution of the discrete algebraic Ricatti equation

$$P = A^T PA - A^T PB \left( B^T PB - R \right)^{-1} B^T PA + Q. \tag{2}$$

Given an estimate or measurement of the current state $\hat{x}$, the constrained LQR is a problem of solving (1) for control sequence $\{u_0, u_1, \cdots, u_{N-1}\}$ from which $u_0$ is applied to the plant at the current time instant. This whole process is repeated at the next sample instant.

### 2.2 Quadratic Program Reformulation

To solve the constrained LQR problem (1), it is customary to reformulate it as a convex quadratic problem. It is well known that the computational burden of model predictive control algorithms largely depend on the how the problem is reformulated as an optimization problem (Jerez et al., 2012). We consider two different QP formulations: the non-condensed and the condensed.

*Non-Condensed MPC Problem (Wright, 1997)* By defining an augmented variable

$$z = \left[ u_o^T \; x_1^T \; u_1^T \cdots u_{N-1}^T \; x_N^T \right]^T \in \mathbb{R}^{(n+m)N}, \tag{3}$$

and associated bounds $\underline{z} = \left[ \underline{u}^T \; \underline{x}^T \; \underline{u}^T \; \cdots \; \underline{u} \; \underline{x} \right]^T$ and $\overline{z} = \left[ \overline{u}^T \; \overline{x}^T \; \overline{u}^T \; \cdots \; \overline{u} \; \overline{x} \right]^T$, the constrained LQR problem can be restated as the QP problem:

$$z^* = \arg\min_z \; \frac{1}{2} z^T H z \tag{4a}$$

$$\text{subject to } Ez = e, \; Gz \leq g \tag{4b}$$

where the matrices $H \in \mathbb{R}^{(n+m)N \times (n+m)N}$, $E \in \mathbb{R}^{nN \times (n+m)N}$, $G \in \mathbb{R}^{2(n+m)N \times (n+m)N}$ and the vectors $e \in \mathbb{R}^{nN}$ and $g \in \mathbb{R}^{2(n+m)N}$ are described as follows:

$$H = \begin{bmatrix} R & & & & & \\ & Q & & & & \\ & & \ddots & & & \\ & & & Q & & \\ & & & & R & \\ & & & & & P \end{bmatrix}, G = \begin{bmatrix} I \\ -I \end{bmatrix}, g = \begin{bmatrix} \overline{z} \\ -\underline{z} \end{bmatrix} \tag{5}$$

$$E = \begin{bmatrix} -B & I & & & & \\ & -A & -B & I & & \\ & & & \ddots & & \\ & & & -A & -B & I \end{bmatrix}, e = \begin{bmatrix} Ax_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{6}$$

Here the resulting QP is large, but the structure can be exploited for faster computation times (Wright, 1997; Jerez et al., 2012).

*Condensed MPC Problem (Maciejowski, 2002)* For the condensed formulation we eliminate the state constraints from (1) by using the condensing equation (Adegbege and Nelson, 2016):

$$x_k = A^k x_o + \sum_{i=0}^{k-1} A^i B u_{k-1-i}, \; k = 1, 2, \cdots, N. \tag{7}$$

By defining the augmented variable as

$$v = \left[ u_o^T \; u_1^T \; \cdots u_{N-1}^T \right]^T \in \mathbb{R}^{mN}, \tag{8}$$

and associated control bounds as $\underline{v} = \left[ \underline{u}^T \; \underline{u}^T \; \cdots \; \underline{u}^T \right]^T$ and $\overline{v} = \left[ \overline{u}^T \; \overline{u}^T \; \cdots \; \overline{u}^T \right]^T$, and re-writing in vector form, the constrained LQR problem (1) can be restated as the following compact QP:

$$v^* = \arg\min_v \frac{1}{2} v^T \widehat{H} v + v^T F x_0 \tag{9a}$$

$$\text{subject to } \widehat{G} v + \widehat{F} x_0 \leq \widehat{g} \tag{9b}$$

where the matrices $\widehat{H}$, $F$, $\widehat{G}$, $\widehat{F}$ and vector $\widehat{g}$ are obtained from the problem data $A, B$ and the weighting matrices $Q, R$ and $P$ as:

$$\widehat{H} = \Lambda^T \widehat{Q} \Lambda + \widehat{R}, \; F = \Lambda^T \widehat{Q} \Phi,$$

$$\widehat{G} = \begin{bmatrix} \Lambda \\ I \\ -\Lambda \\ -I \end{bmatrix}, \quad \widehat{F} = \begin{bmatrix} \Phi \\ 0 \\ -\Phi \\ 0 \end{bmatrix}, \quad \widehat{g} = \begin{bmatrix} \overline{x} \\ \overline{v} \\ -\underline{x} \\ -\underline{v}. \end{bmatrix} \text{ with}$$

$$\widehat{Q} = \begin{bmatrix} Q & & & & \\ & Q & & & \\ & & \ddots & & \\ & & & Q & \\ & & & & P \end{bmatrix}, \quad \widehat{R} = \begin{bmatrix} R & & & & \\ & R & & & \\ & & \ddots & & \\ & & & R & \\ & & & & R \end{bmatrix},$$

$$\Phi = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} \text{and } \Lambda = \begin{bmatrix} B & & & \\ AB & B & & \\ \vdots & & \ddots & \\ A^{N-1}B & A^{N-2}B & \cdots \; AB & B \end{bmatrix}. \tag{10}$$

Here, the resulting QP is compact and dense. It is possible to introduce sparsity into the condensed formulation to cut down on its memory requirement during computation, but at the cost of having to select an additional design variable (Jerez et al., 2012).

## 3. QUADRATIC PROGRAMMING AS A LINEAR COMPLEMENTARITY PROBLEM

The Linear Complememtarity Problem (LCP) provides a generalized framework for describing the optimality conditions for the QPs in (4) and (9), and hence is a viable means for solving the constrained LQR problem in (1). In what follows, we provide some useful definitions and identify important matrix classes for characterizing LCP problems.

### 3.1 LCP definition

**Definition 3.1.** *Given a vector $q \in \mathbb{R}^l$ and a matrix $M \in \mathbb{R}^{l\times l}$ such that*

$$s \geq 0, q + Ms \geq 0, \tag{11a}$$

$$s^T(q + Ms) = 0. \tag{11b}$$

*The LCP($M, q$) is said to be feasible if there exists a vector $s$ satisfying* (11a). *It is solvable if vector $s$ also satisfies* (11b). *The LCP($M, q$) is trivially solvable if $q \geq 0$ (Cottle et al., 2009).*

We are interested in the class of monotone LCPs such that for every $q$, the LCP($M, q$) is solvable in polynomial time. For certain classes of matrix $M$, the LCP($M, q$) is uniquely solvable and the unique $s$ can be retrieved via a pivoting scheme. Perhaps the most widely known of such methods is the complementarity pivoting scheme due to Lemke (Cottle et al., 2009). In this method a user supplied covering vector $d$ is introduced to obtain an augmented system:

$$s \geq 0,\ q + ds_0 + Ms \geq 0, \tag{12a}$$

$$s_0 \geq 0,\ s^T(q + ds_0 + Ms) = 0. \tag{12b}$$

We denote (12) as LCP ($M, q, d$). A solution to (12) in which $s_0 = 0$ yields a solution to the original LCP($M, q$).

In Algorithm 1, we state a version of the Lemke algorithm known as Scheme I employed in this work.

**Algorithm 1.** Lemke Algorithm (Cottle et al., 2009)

**Step 0** (Initialization.) If $q \geq 0$, stop; $s = 0$ solves LCP($M, q$). Otherwise, consider the problem

$$w = Ms + ds_0 + q \tag{13a}$$

$$w \geq 0, s \geq 0, s_0 \geq 0 \tag{13b}$$

and choose $s_0$ such that

$$s_0 = \max_i\{-q_i/d_i\} \tag{14a}$$

$$r = \arg\max_i\{-q_i/d_i\} \tag{14b}$$

Pivot on the element at position $(w_r, s_0)$ and set the driving variable to $s_r$, the complement of $w_r$.

**Step 1** (Determination of blocking variable.) If the column of the driving variable $s_r$ has at least one positive entry, then use the minimum ratio test

$$t = \arg\min_i\{q_i/m_{ri} : m_{ri} > 0\} \tag{15}$$

to determine $w_t$ that blocks the increase of the driving variable. If the driving variable is unblocked, then stop. The method has failed to find a solution.

**Step 2** (Pivoting.) The driving variable is blocked.

- If $s_0$ blocks $s_r$, then pivot on the element at position $(s_0, s_r)$ and stop. A solution is at hand.
- Otherwise pivot on the element at position $(w_t, s_r)$. Then return to Step 1 using $s_t$, and the complement of the current blocking variable as the new driving variable.

For some matrix classes, the Lemke algorithm stated in Algorithm 1 is known to process the LCP($M, q$) (i.e. provides a solution or determines a solution does not exist) after finitely many pivot steps. We recall some of these matrix classes here.

**Definition 3.2.** *A matrix $M \in \mathbb{R}^{l\times l}$ is called a $P(P_0)$-matrix if all its principal minors are positive (non-negative).*

**Definition 3.3.** *A matrix $M \in \mathbb{R}^{l\times l}$ is called a $Z$-matrix if all its off-diagonal entries are all non-positive. A $Z$-matrix which is also a $P$-matrix is called a $K$-matrix.*

**Definition 3.4.** *A matrix $M \in \mathbb{R}^{l\times l}$ is called a* hidden *$Z$-matrix if there exists $Z$-matrices $X$ and $Y$ and non-negative vectors $r$ and $p$ such that:*

$$MX = Y, \tag{16a}$$

$$r^T X + p^T Y > 0. \tag{16b}$$

*A hidden $Z$-matrix which is also a $P$-matrix is called a* hidden *$K$-matrix.*

**Remark 1.** *Note that if $M$ is a $P$-matrix, Algorithm 1 furnishes the unique solution of LCP($M, q$) for any $q \in \mathbb{R}^l$ in a finite number of pivot steps. However if $M$ is a $Z$-matrix or a $P_0$-matrix and the LCP($M, q$) is feasible, then it is solvable and Algorithm 1 will process the LCP($M, q$) for any $q \in \mathbb{R}^l$ in a finite number of pivot steps (Cottle et al., 2009, Theorem 4.7.5).*

**Remark 2.** *We are interested in a very special type of LCP($M, q$) where $(M, q)$ takes the form $(P\widehat{A}P^T, P\widehat{b})$ for some given $\widehat{b} \in \mathbb{R}^q$ and some matrix $P \in \mathbb{R}^{l\times q}$, and where $\widehat{A} \in \mathbb{R}^{q\times q}$ may be positive definite or positive semi-definite. LCPs of this type arise in convex quadratic programming problems (Adegbege and Heath, 2017; Okawa and Nonaka, 2021) and can be processed using Algorithm 1.*

While the Lemke algorithm is popular for its ability to process LCP($M, q$) with $M$ belonging to a wide variety of matrix classes, it may require an exponential number of pivots (Murty and Yu, 1988). The performance of the algorithm can vary drastically depending on the choice of the covering vector $d$ employed in (12). However, for certain matrix classes and for an appropriately constructed covering vector, the Lemke algorithm can process the problem in at most $l + 1$ pivot steps. This linear termination property is very beneficial particularly for applications where large data has to be processed with limited computational resources such as in embedded MPC (Jerez et al., 2012; Adegbege and Nelson, 2016).

### 3.2 Some Existing Results

We recall some results of sufficient conditions for the linear termination of Algorithm 1. The results rely on the existence of a special vector $d$ that can be employed as a covering vector for the Lemke algorithm. We first state a result that relies on the non-degeneracy of matrix $M$ (i.e. all the principal minors of $M$ are nonzero).

**Definition 3.5** ($l$-step vector)**.** *Chu (2006) A positive vector $d > 0$ is called a $l$-vector for matrix $M$ if for all index set $\alpha \subseteq \{1, \cdots, l\}$ the following condition holds*

$$M_{\alpha\alpha}^{-1} d_\alpha > 0. \tag{17}$$

**Theorem 3.1.** *Pang and Chandrasekaran (1985) Let matrix $M$ be non-degenerate. If there exists a vector $d > 0$ such that condition* (17) *holds, then the Lemke pivoting algorithm in Algorithm 1 with $d$ as a covering vector solves the LCP($M, q$) in at most $l$+1 pivots.*

We note that the problem of finding a suitable $l$-vector for an arbitrary $M$ is non-trivial (Adler et al., 2016). However when $M$ is a $P$-matrix, the existence of a $l$-vector can be characterized in terms of the *hidden* $Z$-property of the transpose of $M$. For this case, condition (17) can be relaxed to $M_{\alpha\alpha}^{-1} d_\alpha \geq 0$.

**Theorem 3.2.** *Pang and Chandrasekaran (1985) Let $M^T$ be an hidden $K$-matrix such that there exist $Z$-matrices $X$ and $Y$ satisfying the conditions in* (16)*. Then condition* (17) *holds for any vector $d > 0$ satisfying $X^T d > 0$. Furthermore, the Lemke pivoting algorithm in Algorithm 1 with $d$ as a covering vector solves the LCP($M, q$) in at most $l$+1 pivots.*

In general $M$ is a not $P$-matrix and therefore is not a hidden $K$-matrix. However, the concept of a $l$-vector can be extended to cover $M$-matrices of the form highlighted in Remark 2. This case corresponds to the degenerate LCP problems where some of the principal minors of matrix $M$ may be zero. To handle this case, we require an extended version of the $l$-step vector. Different versions of the extended vector exists in the literature, but we adopt the one due to Chu (2006).

**Definition 3.6** (Extended $l$-step vector)**.** *Chu (2006) A positive vector $d > 0$ is called an extended $l$-vector for matrix $M$ if the following condition holds*

$$M_{\alpha\alpha}^{-1} d_\alpha \geq 0 \qquad \forall\ M_{\alpha\alpha} \text{ nonsingular}, \tag{18a}$$

$$\text{rank } (M_{\alpha\alpha} \mid d_\alpha) < |\alpha| \qquad \forall\ M_{\alpha\alpha} \text{ singular} \tag{18b}$$

*where $(M_{\alpha\alpha} \mid d_\alpha)$ is an augmented matrix and $|\alpha|$ is the cardinality of the index set $\alpha \subseteq \{1, \cdots, l\}$.*

This definition allows for dealing with LCP problems where the $M$-matrix is singular, but can be permutated or partitioned into blocks with a nonsingular principal sub-matrix (Adegbege and Heath, 2017; Okawa and Nonaka, 2021). The following theorem addresses this situation.

**Theorem 3.3.** *Chu (2006) Let matrix $M \in \mathbb{R}^{l \times l}$ and let $d$ be an extended $l$-step vector satisfying* (18)*. Then, for every $q \in \mathbb{R}^l$, the Lemke pivoting algorithm in Algorithm 1 with $d$ as a covering vector solves the LCP($M, q$) in at most rank* ($M$) *iterations.*

As the LCP $M$ matrices arising from MPC are typically degenerate, we are now in a position to extend Theorem 3.3 to the constrained LQR problem.

## 4. MAIN RESULTS

We reformulate the MPC problem into equivalent LCP problems and we show that the Lemke Algorithm 1 has a linear termination in terms of the number of pivot iterations.

### *4.1 LCP for Condensed MPC*

For the condensed MPC formulation (Jerez et al., 2012), the KKT optimality condition for (9) can be expressed as

$$\begin{aligned} &0 = \widehat{H} v + F x_0 + \widehat{G}^T \widehat{\lambda}, \\ &\widehat{g} - \widehat{G} v - \widehat{F} x_0 \geq 0,\ \widehat{\lambda} \geq 0,\ (\widehat{g} - \widehat{G} v - \widehat{F} x_0)^T \widehat{\lambda} = 0, \end{aligned} \tag{19}$$

where $\widehat{\lambda}$ is the Lagrangian multiplier for the inequality constraint. Observe that (19) describes a set of coupled system of linear equations and a linear complementarity problem. This is commonly termed mixed LCP. Since $\widehat{H}$ is positive definite, (19) can be expressed in terms of the LCP($M, q$) with $s = \widehat{\lambda}$ and

$$M = \widehat{G} \widehat{H}^{-1} \widehat{G}^T \tag{20a}$$

$$q = \widehat{g} + \left[ \widehat{G} \widehat{H}^{-1} F - \widehat{F} \right] x_0. \tag{20b}$$

Suppose $\widehat{\lambda}$ solves the LCP defined by (20), then $v$ satisfying (19) can be retrieved from solving

$$v = -\widehat{H}^{-1} \left[ F x_0 + \widehat{G}^T \widehat{\lambda} \right]. \tag{21a}$$

Since by construction $\widehat{H}$ is positive definite, we show that for $M$ given by (20a) there exist an extended $l$-step vector.

**Lemma 4.1.** *Consider the LCP ($M, q$) defined by matrix $M$ and vector $q$ given by* (20)*. Then there is an extended $l$-step vector for $M$ satisfying* (18)*.*

*Proof.* Using the definitions in (10), we expand out matrix $M = \widehat{G} \widehat{H}^{-1} \widehat{G}^T \in \mathbb{R}^{2(n+m)N \times 2(n+m)N}$ in terms of matrices $\widehat{H}$ and $\Lambda$ as

$$M = \begin{bmatrix} \Lambda \widehat{H}^{-1} \Lambda^T & \Lambda \widehat{H}^{-1} & -\Lambda \widehat{H}^{-1} \Lambda^T & -\Lambda \widehat{H}^{-1} \\ \widehat{H}^{-1} \Lambda^T & \widehat{H}^{-1} & -\Lambda \widehat{H}^{-1} \Lambda^T & -\widehat{H}^{-1} \\ -\Lambda \widehat{H}^{-1} \Lambda^T & -\Lambda \widehat{H}^{-1} & \Lambda \widehat{H}^{-1} \Lambda^T & \Lambda \widehat{H}^{-1} \\ -\widehat{H}^{-1} \Lambda^T & -\widehat{H}^{-1} & -\widehat{H}^{-1} \Lambda^T & \widehat{H}^{-1} \end{bmatrix}. \tag{22}$$

It is obvious that $M$ is positive semi-definite (since $\widehat{H}$ is positive definite by construction and $\Lambda$ is full column ranked (under the assumption that $(A, B)$ is controllable) and hence degenerate. It follows that the principal sub-matrices $\widehat{H}^{-1}$ and $\Lambda \widehat{H}^{-1} \Lambda^T$ of $M$ are respectively positive definite and positive semi-definite. Now consider the index set $\alpha \subseteq (1, 2, \cdots, l)$ where $l = 2(n+m)N$. It follows from (22) that for each index set $\alpha$ with det $(M_{\alpha\alpha}) = 0$, the columns of $M_{*\alpha}$ are linearly dependent. Since $M$ is symmetric, $M_{\alpha *}$ has linearly dependent rows. Suppose $\alpha_k$ is one of such rows that can be expressed as a linear combination of other rows. Define set $\beta = \{\alpha_1, \alpha_2, \cdots, \alpha_r\} \backslash \{\alpha_k\}$. Then $M_{\beta\beta}$ is non-singular, and the pair $(M_{\beta\beta}, d_\beta)$ satisfies the extended $l$-vector property, and hence there is a non-negative solution $x_\beta$ to $M_{\beta\beta} x_\beta = d_\beta$. Set $x_\beta = (x_1, \cdots, x_{k-1}, x_{k+1}, \cdots x_r)^T$. Then $x_\alpha = (x_1, \cdots, x_{k-1}, 0, x_{k+1}, \cdots x_r)^T$ solves the original system $M_{\alpha\alpha} x_\alpha = d_\alpha$.

Now for the index set for which $M_{\alpha\alpha}$ is nonsingular, $M_{\alpha\alpha}$ can be obtained from a corresponding principal sub-matrix of $\widehat{H}^{-1}$ after permutation. In this case, $M_{\alpha\alpha}$ is positive definite, and there exists a positive vector $x_\alpha$ such that $M_{\alpha\alpha} x_\alpha > 0$ (Cottle et al., 2009, Lemma 3.1.3). Hence there is a positive vector $d_\alpha$ satisfying (18). □

We are now in a position to state our result for the condensed LQR problem as a corollary to Theorem 3.3.

**Corollary 4.1.1.** *Let the LCP $(M, q)$ problem be defined by matrix $M$ and vector $q$ given by (20). Then there is an extended $l$-step vector $d$ satisfying (18), and the Lemke pivoting algorithm in Algorithm 1 with $d$ as a covering vector solves the LCP$(M, q)$ in at most rank$(M)$ iterations.*

### 4.2 LCP for Non-condensed MPC

For the non-condensed QP formulation of the constrained LQR problem (1) (Wright, 1997), we write the Karush-Kuhn-Tucker (KKT) condition as:

$$\begin{aligned} 0 = Hz + E^T\xi + G^T\lambda,\ 0 = Ez - e, \\ g - Gz \geq 0,\ \lambda \geq 0,\ (g - Gz)^T\lambda = 0. \end{aligned} \tag{23}$$

Observe that (23) is also a mixed LCP problem. Since $H$ is positive definite and $E$ is full row ranked, the KKT (23) can be expressed in terms of the LCP$(M, q)$ with $s = \lambda$ and

$$M = G\left[H^{-1} - H^{-1}E^T\left(EH^{-1}E^T\right)^{-1}EH^{-1}\right]G^T, \tag{24a}$$

$$q = g - G(EH^{-1})^T\left(EH^{-1}E^T\right)^{-1}e. \tag{24b}$$

Suppose $\lambda$ solves the LCP defined by (24), then $\xi$ and $z$ satisfying (23) can be retrieved respectively from

$$\xi = -\left(EH^{-1}E^T\right)^{-1}\left[EH^{-1}G^T\lambda + e\right], \tag{25a}$$

$$z = -H^{-1}\left[E^T\xi + G^T\lambda\right]. \tag{25b}$$

Note that the matrices in (24) and (25) can be expressed in terms of the inverse of the block KKT matrix:

$$\begin{bmatrix} H & E^T \\ E & 0 \end{bmatrix}^{-1} = \begin{bmatrix} H^{-1} + H^{-1}E^TS^{-1}EH^{-1} & H^{-1}E^TS^{-1} \\ S^{-1}EH^{-1} & S^{-1} \end{bmatrix}$$

where $S = -EH^{-1}E^T$ is the Schur complement of H. This inverse can be computed efficiently by taking advantage of the structure of $H$ and $E$. We only need to store the top blocks to evaluate the LCP$(M, q)$ and to recover the primal solution $z$.

**Lemma 4.2.** *Consider the LCP $(M, q)$ defined by matrix $M$ and vector $q$ given by (24). Then there is an extended $l$-step vector for $M$ satisfying (18).*

*Proof.* By defining $\widehat{M} = H^{-1} + H^{-1}E^TS^{-1}EH^{-1}$, the matrix $M$ becomes:

$$M = \begin{bmatrix} \widehat{M} & -\widehat{M} \\ -\widehat{M} & \widehat{M} \end{bmatrix} \in \mathbb{R}^{2(n+m)N\times 2(n+m)N}. \tag{26}$$

It is easy to check that $\widehat{M}$ is positive semi-definite with exactly $mN$ positive eigenvalues. Note that $H$ is positive definite by construction and $E$ is full row ranked by construction (see (5)), hence $EH^{-1}E^T$ is positive definite. Suppose we define the index set $\alpha \subseteq (1, 2, \cdots, l)$ where $l = 2(n + m)N$. Suppose $\alpha$ includes both $i$ and $i + n + m$ for some $i$. Then $M_{*\alpha}$ has linearly dependent columns and $M_{\alpha\alpha}$ may be obtained from a corresponding principal sub-matrix of $\widehat{M}$ after permutation or scaling by (-1). By symmetry, $M_{\alpha *}$ also has linearly dependent rows. Deleting the rows (say $\alpha_k$) that can be expressed as a linear combination of other rows, we have a reduced system $M_{\beta\beta}x_\beta = d_\beta$ where $\beta = \{\alpha_1, \alpha_2, \cdots, \alpha_r\}\backslash\{\alpha_k\}$. Following a similar procedure in the proof of Lemma 4.1, we have that there is a non-negative vector $x_\alpha$ which solves the system $M_{\alpha\alpha}x_\alpha = d_\alpha$. Hence, there is an extended $l$-vector for $M$ satisfying (18). □

We state a corresponding corollary for the non-condensed constrained LQR problem.

**Corollary 4.2.1.** *Let the LCP $(M, q)$ problem be defined by matrix $M$ and vector $q$ given by (24). Then there is an extended $l$-step vector $d$ satisfying (18), and the Lemke pivoting algorithm in Algorithm 1 with $d$ as a covering vector solves the LCP$(M, q)$ in at most rank$(M)$ iterations.*

**Remark 3.** *Observe that the computational complexity results here only depend on the existence of an $l$-step vector for matrix $M$ while the vector $q$ plays no role. In general, the computational complexity of an algorithm depends not only on the matrix $M$ but also on the vector $q$ furnished by the data. It is therefore possible to take advantage of $q$ to further reduce the computational complexity of the LCP as it is done in (Adegbege and Heath, 2017; Okawa and Nonaka, 2021).*

## 5. EXPERIMENTAL RESULTS

### 5.1 Simulation Setup

We use the well-known double integrator system to test the existence of a $l$-step vector $d$ that satisfies (18) for both the condensed and non-condensed formulations. The system prediction model in (1c) is given as:

$$x_{k+1} = Ax_k + Bu_k$$

where $A = \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} \Delta^2/2 \\ \Delta \end{bmatrix}$.

Following the notation in (Okawa and Nonaka, 2021), we set $Q = I$, $R = 1.5$ and $P$ as a solution to the discrete algebraic Riccati equation (2). The discrete interval $\Delta$ and the prediction horizon $N$ are varied in fixed intervals of $\Delta = 0.01, 0.02, \ldots, 0.30$ and $N = 1, 2, \ldots, 30$.

Using the LCPs derived for both the condensed and the non-condensed constrained LQR MPC formulations, We check if the corresponding matrix $M$ satisfies (18). If so, then a $l$-step vector $d$ exists for the LCP. The test is run for every combination of $\Delta$ and $N$.

### 5.2 Condensed Formulation

For the condensed LCP formulation, the existence of a $l$-step vector using (18) was compared to the search algorithm described in (Okawa and Nonaka, 2021). This search algorithm uses the assumption that the matrix $M$ must be a hidden-$Z$ matrix in order to have a corresponding $l$-step vector. However, this assumption is not necessary in MPC applications and also significantly limits the number of problems for which a $l$-step vector exists. By avoiding this assumption and instead using (18), we provide a much more effective procedure for finding a $l$-step vector $d$ for the condensed MPC problem.

The experimental results are shown in Fig. 1, in which the horizontal axis represents the prediction horizon $N$,

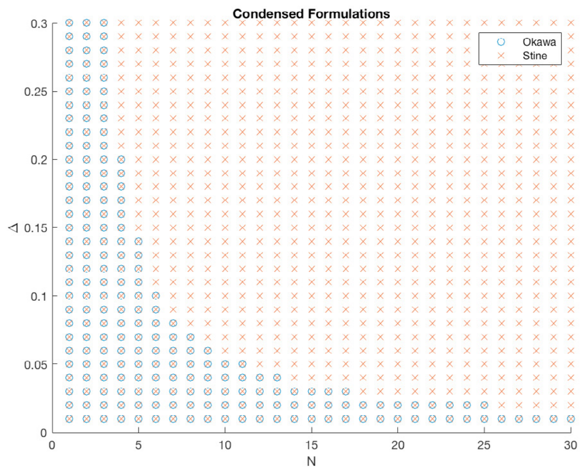


Fig. 1. $l$-step vector existence for $u$ constrained condensed formulation

and the vertical axis represents the discrete time interval $\Delta$. The combinations of $N$ and $\Delta$ for which a $l$-step vector exists using the condition (18) are represented by $\times$'s, and are plotted against those of (Okawa and Nonaka, 2021) which are represented by o's. The result in (Okawa and Nonaka, 2021) indicates that a computational complexity certificate can only be found for a certain region of the problem, as a $l$-step vector only existed for a portion of the combinations of $\Delta$ and $N$. However, we present here a valid search procedure that finds an existing $l$-step vector for every LCP in question. This allows the Lemke algorithm to solve the LCP in at most $l + 1$ steps and for the maximum computational load to be determined offline. This in turn increases the usability of the search procedure, as a concrete upper bound can be found for every combination of $\Delta$ and $N$.

### 5.3 *Non-Condensed Formulation*

Not shown in (Okawa and Nonaka, 2021) is the existence of a $l$-step vector for the non-condensed LCP formulation of the discretized double integrator. Again (18) is used to test for the existence of the $l$-step vector $d$ as for the condensed formulation.

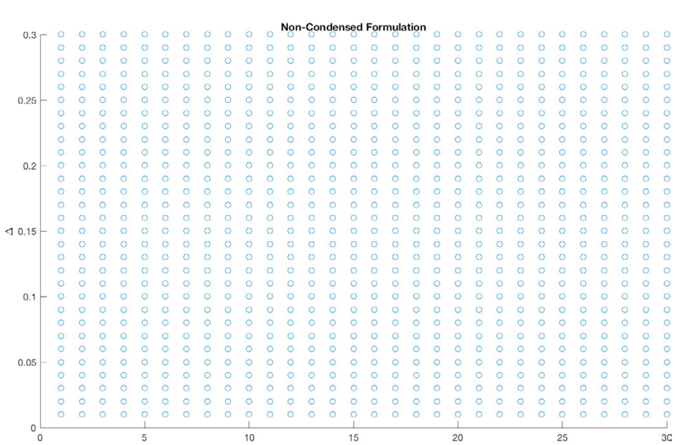


Fig. 2. $l$-step vector existence for non-condensed formulation

The results of the experiment are shown in Fig. 2, where the combinations of $\Delta$ and $N$ that resulted in a $l$-step vector are indicated by o's. We found that there is a $l$-step vector $d$ for all combinations of $\Delta$ and $N$.

## 6. CONCLUSION

We have shown that the LCP arising from MPC always satisfies the condition for an extended $l$-step vector to exist, and therefore can be solved in polynomial time using the Lemke algorithm. Experimental results revealed that such extended $l$ vector can easily be computed without the conservatism in existing results. Since each pivot operation requires at most $\mathcal{O}(n^2)$ arithmetic operations, the associated LCP can be solved in $\mathcal{O}(n^3)$ time.

## REFERENCES


Adegbege, A.A. and Nelson, Z.E. (2016). A gauss–seidel type solver for the fast computation of input-constrained control systems. *Systems & Control Letters*, 97, 132–138.

Adegbege, A.A. and Heath, W.P. (2017). A framework for multivariable algebraic loops in linear anti-windup implementations. *Automatica*, 83, 81–90.

Adler, I., Cottle, R.W., and Pang, J.S. (2016). Some lcps solvable in strongly polynomial time with lemke's algorithm. *Mathematical Programming*, 160(1), 477–493.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Chu, T.H. (2006). A class of polynomially solvable linear complementarity problems. *Mathematical programming*, 107(3), 461–470.

Cottle, R.W., Pang, J.S., and Stone, R.E. (2009). *The linear complementarity problem*. SIAM.

Jerez, J.L., Kerrigan, E.C., and Constantinides, G.A. (2012). A sparse and condensed qp formulation for predictive control of lti systems. *Automatica*, 48(5), 999–1002.

Maciejowski, J.M. (2002). *Predictive control: with constraints*. Pearson education, Harlow, UK.

Murty, K.G. and Yu, F.T. (1988). *Linear complementarity, linear and nonlinear programming*, volume 3. Heldermann, Berlin, Germany.

Okawa, I. and Nonaka, K. (2021). Linear complementarity model predictive control with limited iterations for box-constrained problems. *Automatica*, 125, 109429.

Pang, J.S. and Chandrasekaran, R. (1985). Linear complementarity problems solvable by a polynomially bounded pivoting algorithm. In *Mathematical Programming Essays in Honor of George B. Dantzig Part II*, 13–27. Springer.

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

Wright, S. (1997). Applying new optimization algorithms to model predictive control. In *Chemical Process Control V AIChE Symposium Series*, volume 93, 147–155.