C3-SDP: Multi-Contact Consensus Complementarity Control via ADMM and Moment Relaxation

Xihang Yu*

College of Literature, Science, and the Arts, University of Michigan.

Abstract

This research conducts a thorough comparison between Mixed Integer Quadratic Programming (MIQP) and two versions of Semidefinite Programming (SDP) formulations, specifically tailored for hybrid Model Predictive Control (MPC) in multi-contact scenarios. Drawing inspiration from linear complementarity (LCP) framework [4] and building upon Aydinoglu's work [1], we address the challenges posed by a nonconvex projection subproblem (LCP) in the Alternating Direction Method of Multipliers (ADMM), a critical obstacle in real-time, high-dimensional control. The study delves into the complexities of this projection subproblem and its MIQP formulation (MIQP), exploring the trade-off between speed and robustness in ADMM and LCP projections, and the robustness but slower performance of the MIQP projection. Our novel approach integrates SDP with moment relaxation to potentially overcome the computational limitations of these existing methods. Our results reveal that while SDP does not outpace MIQP in terms of speed, it successfully completes pivoting tasks where LCP and ADMM projections falter. We delve into the comparison between two SDP formulations, noting the faster yet suboptimal performance of the LCP-based SDP and the slower, more accurate results of the MIQP-based SDP. This work not provides some insights into the SDP and MIQP formulations, their implementation, and their implications in hybrid MPC problems.

1 Introduction

In this research, we focus on developing a SDP formulation tailored for the hybrid MPC problem, with a special emphasis on multi-contact scenarios framed within a linear complementarity formulation [4]. Building upon Aydinoglu's pioneering work [1], which solves an MPC problem (MPC) for multi-contact issues using ADMM [2] to achieve real-time performance, we identify and address a significant challenge.

$$
\begin{aligned}
f^\star = \min_{x_k,\lambda_k,u_k} & \sum_{k=0}^{N-1} (x_k^T Q_k x_k + u_k^T R_k u_k) + x_N^T Q_N x_N \qquad \text{(MPC)}\\
\text{subject to} \quad & x_{k+1} = Ax_k + Bu_k + D\lambda_k + d,\\
& Ex_k + F\lambda_k + Hu_k + c \geq 0,\\
& \lambda_k \geq 0,\\
& \lambda_k^T (Ex_k + F\lambda_k + Hu_k + c) = 0,\\
& (x, \lambda, u) \in \mathcal{C}, \text{ for } k = 0, ..., N-1, \text{ given } x_0
\end{aligned}
$$

*Course project in EECS563 Hybrid Control Systems at University of Michigan. Email: `xihangyu@umich.edu`

However, a nonconvex projection subproblem (LCP) in ADMM remains the hardest part and a bottle neck of real-time implementation for higher dimension control. [1]

$$\begin{aligned} \rho^\star_{\mathsf{LCP}} = & \min_{\delta_k} (\delta_k - z)^T U (\delta_k - z) \\ \text{subject to} \quad & E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c \geq 0, \\ & \delta_k^\lambda \geq 0, {\delta_k^\lambda}^T (E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c) = 0. \end{aligned} \tag{LCP}$$

where $z \in \mathbb{R}^{n_\lambda + n_x + n_u}, U \in \mathbb{S}^{n_\lambda + n_x + n_u}, E \in \mathbb{R}^{n_\lambda \times n_x}, F \in \mathbb{R}^{n_\lambda \times n_\lambda}, H \in \mathbb{R}^{n_\lambda \times n_u}, c \in \mathbb{R}^{n_\lambda}$ are constants. $\delta_k := (\delta_k^x, \delta_k^\lambda, \delta_k^u) \in \mathbb{R}^{n_\lambda + n_x + n_u}$ where δ_k^x is the state, δ_k^λ is the contact force and δ_k^u is the control input. Optimization LCP has a MIQP formulation using big $\mathbf{M}$ method:

$$\begin{aligned} \rho^\star_{\mathsf{MIQP}} = & \min_{\delta_k, s_k} (\delta_k - z)^T U (\delta_k - z) \\ \text{subject to} \quad & M s_k \geq E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c \geq 0, \\ & M(1 - s_k) \geq \delta_k^\lambda \geq 0, \\ & s_k \in \{0, 1\}^{n_\lambda}. \end{aligned} \tag{MIQP}$$

where M is a matrix of the $M = mI$ where m is a pre-specified large number. [1] proposes three methods to solve a projection subproblem in ADMM - MIQP projection, ADMM projection, LCP projection. Despite faster performance in ADMM and LCP projections, these methods suffer from robustness [1]. MIQP prjection, on the other hand, is robust to all experiments as shown in [1] thanks to optimal guarantee. However, MIQP projection is slow due to the exponential worst run-time. In the pursuit of enhancing the efficiency of MIQP while keeping optimal guarantee with rank certificate, this thesis proposes an innovative approach leveraging semidefinite relaxation. The study delves deep into the integration of SDP and moment relaxation to formulate complex mathematical problems, aiming at transcending the computational barriers of existing methodologies.

Our findings indicate that while SDP does not offer a speed advantage in optimization solutions compared to MIQP, both SDP and MIQP uniquely succeed (while LCP and ADMM projections fail) in completing the pivoting task. The relatively slower performance of SDP can be attributed to several factors: (1) the interior point method used in the MOSEK solver is a general-purpose tool, not specifically optimized for this problem. (2) the problem's dimension, approximately 30, is a small scale for MIQP's enumerative approach. Interestingly, our analysis revealed some noteworthy observations. The SDP formulation, when based on the LCP), operates faster but at the expense of increased suboptimality. In contrast, the SDP formulation derived from an MIQP framework, which incorporates integer decision variables expressed as polynomial constraints, demonstrates slower execution yet achieves significantly tighter suboptimalities, by an order of magnitude of 5. We will delve into the details of these two formulations in Section 2 and subsequently discuss the experimental results in Section 3.

2 Semidefinite Relaxation

We observe that the objective function and constraints of Optimization (LCP) are polynomials. Hence, we can directly apply moment relaxation on the polynomial optimization (POP). However, we can not apply SDP relaxation to (MIQP) directly since the decision variable s_k is integer.

[1]In ADMM, the projection step is to project the solution back to the nonconvex constraint after solving a relaxed augmented Lagrangian problem.

However, the integer constraints can be written in equivalent form:

$$\begin{aligned} \rho^\star_{\mathsf{MIQP}} = & \min_{\delta_k, s_k} (\delta_k - z)^T U(\delta_k - z) \\ \text{subject to} \quad & M s_k \geq E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c \geq 0, \\ & M(1 - s_k) \geq \delta_k^\lambda \geq 0, \\ & s_k(s_k - 1) = 0. \end{aligned} \tag{1}$$

Now the objective function and constraints are polynomials and we can apply SDP relaxation. Based on the two formulated (LCP) and (1), We can now design the following two semidefinite relaxations. We call them LCP-based SDP and MIQP-based SDP respectively.

Proposition 1 (LCP-based SDP)**.** *The following SDP*

$$\begin{aligned} f^\star = & \quad \min_{X \in \mathbb{S}^{1+n_\lambda+n_x+n_u}} \operatorname{tr}(CX) \\ \textit{subject to} & \quad l_i \leq \textstyle\sum_i \operatorname{tr}(A_i, X) \leq u_i, i = 1, ..., t \\ X = & \begin{bmatrix} 1 & \delta_k^{xT} & \delta_k^{\lambda T} & \delta_k^{uT} \\ \delta_k^x & \delta_k^x \delta_k^{xT} & \delta_k^x \delta_k^{\lambda T} & \delta_k^x \delta_k^{uT} \\ \delta_k^\lambda & \delta_k^\lambda \delta_k^{xT} & \delta_k^\lambda \delta_k^{\lambda T} & \delta_k^\lambda \delta_k^{uT} \\ \delta_k^u & \delta_k^u \delta_k^{xT} & \delta_k^u \delta_k^{\lambda T} & \delta_k^u \delta_k^{uT} \end{bmatrix} \succeq 0 \end{aligned} \tag{2}$$

is a convex relaxation to (LCP) *and* $f^\star \leq \rho^\star_{\mathsf{LCP}}$. *Let* $X^\star$ *be a global minimizer of* (2)*. If* $\operatorname{rank}(X^\star) = 1$*, then* $X^\star$ *can be factorized as* $X^\star = (x_k^\star)^\mathsf{T} x_k^\star$*, where* $x_k^\star \in \mathbb{R}^{1+n_\lambda+n_x+n_u}$ *is a global optimizer to* (LCP)*.*

Proof. It is easy to verify that the objective function can be written as:

$$(\delta_k - z)^T U(\delta_k - z) = \operatorname{tr}(CX) \tag{3}$$

where

$$C = \begin{bmatrix} z^T U z & (-Uz)^T \\ -Uz & U \end{bmatrix} \tag{4}$$

For the constraint $E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c \geq 0$, it can be written in n_λ constraints:

$$-c_i \leq \operatorname{tr}(A_{1,i} X) \tag{5}$$

where c_i is i^{th} element of c and $A_{1,i}$ is as follows:

$$A_{1,i} = \begin{bmatrix} 0 & \frac{1}{2}E_i & \frac{1}{2}F_i & \frac{1}{2}H_i \\ (\frac{1}{2}E_i)^T & \mathbf{0}_{n_x} & \mathbf{0}_{n_x \times n_\lambda} & \mathbf{0}_{n_x \times n_u} \\ (\frac{1}{2}F_i)^T & \mathbf{0}_{n_\lambda \times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda \times n_u} \\ (\frac{1}{2}H_i)^T & \mathbf{0}_{n_u \times n_x} & \mathbf{0}_{n_u \times n_\lambda} & \mathbf{0}_{n_u} \end{bmatrix} \tag{6}$$

with E_i, F_i, H_i are i^{th} row of E, F, H. For the constraint $\delta_k^\lambda \geq 0$, it can be written in n_λ constraints:

$$0 \leq \operatorname{tr}(A_{2,i} X) \tag{7}$$

where $A_{2,i}$ is as follows:

$$A_{2,i} = \begin{bmatrix} 0 & \mathbf{0}_{1 \times n_x} & \frac{1}{2}e^T_{n_\lambda,i} & \mathbf{0}_{1 \times n_u} \\ \mathbf{0}_{n_x \times 1} & \mathbf{0}_{n_x} & \mathbf{0}_{n_x \times n_\lambda} & \mathbf{0}_{n_x \times n_u} \\ \frac{1}{2}e_{n_\lambda,i} & \mathbf{0}_{n_\lambda \times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda \times n_u} \\ \mathbf{0}_{n_u \times 1} & \mathbf{0}_{n_u \times n_x} & \mathbf{0}_{n_u \times n_\lambda} & \mathbf{0}_{n_u} \end{bmatrix} \tag{8}$$

with $e_{n_\lambda,i} \in \mathbb{R}^{n_\lambda}$ be unit vector where i^{th} entry is 1 and other entries are all zeros. For the constraint $\delta_k^{\lambda T}(E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c) = 0$, it can be written as:

$$\operatorname{tr}(A_{3,i}X) = 0 \tag{9}$$

where $A_{3,i}$ is as follows:

$$A_{3,i} = \begin{bmatrix} 0 & \mathbf{0}_{1\times n_x} & (\frac{1}{2}c)^T & \mathbf{0}_{1\times n_u} \\ \mathbf{0}_{n_x\times 1} & \mathbf{0}_{n_x} & (\frac{1}{2}E)^T & \mathbf{0}_{n_x\times n_u} \\ \frac{1}{2}c & \frac{1}{2}E & \frac{1}{2}(F+F^T) & \frac{1}{2}H \\ \mathbf{0}_{n_u\times 1} & \mathbf{0}_{n_u\times n_x} & (\frac{1}{2}H)^T & \mathbf{0}_{n_u} \end{bmatrix} \tag{10}$$

□

Similarly, we can prove the following proposition and obtain the second form of SDP:

Proposition 2 (MIQP-based SDP)**.** *The following SDP*

$$f^\star = \quad \min_{Y\in\mathbb{S}^{1+2n_\lambda+n_x+n_u}} \operatorname{tr}(BY) \tag{11}$$

$$\textit{subject to} \quad l_i \leq \textstyle\sum_i \operatorname{tr}(D_i, Y) \leq u_i, i = 1, ..., q \tag{12}$$

$$Y = \begin{bmatrix} 1 & s_k^T & \delta_k^{xT} & \delta_k^{\lambda T} & \delta_k^{uT} \\ s_k & s_k s_k^T & s_k\delta_k^{xT} & s_k\delta_k^{\lambda T} & s_k\delta_k^{uT} \\ \delta_k^x & \delta_k^x s_k^T & \delta_k^x\delta_k^{xT} & \delta_k^x\delta_k^{\lambda T} & \delta_k^x\delta_k^{uT} \\ \delta_k^\lambda & \delta_k^\lambda s_k^T & \delta_k^\lambda\delta_k^{xT} & \delta_k^\lambda\delta_k^{\lambda T} & \delta_k^\lambda\delta_k^{uT} \\ \delta_k^u & \delta_k^\lambda s_k^T & \delta_k^u\delta_k^{xT} & \delta_k^u\delta_k^{\lambda T} & \delta_k^u\delta_k^{uT} \end{bmatrix} \succeq 0$$

is a convex relaxation to (MIQP) *and* $f^\star \leq \rho^\star_{\mathsf{MIQP}}$. *Let* $Y^\star$ *be a global minimizer of* (11). *If* $\operatorname{rank}(Y^\star) = 1$, *then* $Y^\star$ *can be factorized as* $Y^\star = (y_k^\star)^\mathsf{T} y_k^\star$, *where* $y_k^\star \in \mathbb{R}^{1+2n_\lambda+n_x+n_u}$ *is a global optimizer to* (MIQP).

Proof. It is easy to verify that the objective function can be written as:

$$(\delta_k - z)^T U(\delta_k - z) = \operatorname{tr}(BY) \tag{13}$$

where

$$B = \begin{bmatrix} z^T U z & \mathbf{0}_{1\times n_\lambda} & (-Uz)^T \\ \mathbf{0}_{n_\lambda\times 1} & \mathbf{0}_{n_\lambda\times n_\lambda} & \mathbf{0}_{n_\lambda\times(n_\lambda+n_x+n_u)} \\ -Uz & \mathbf{0}_{(n_\lambda+n_x+n_u)\times n_\lambda} & U \end{bmatrix} \tag{14}$$

For the constraint $E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c - Ms_k \leq 0$, it can be written in n_λ constraints:

$$\operatorname{tr}(D_{1,i}X) \leq -c_i \tag{15}$$

where c_i is i^{th} element of c and $D_{1,i}$ is as follows:

$$D_{1,i} = \begin{bmatrix} 0 & -\frac{1}{2}M_i & \frac{1}{2}E_i & \frac{1}{2}F_i & \frac{1}{2}H_i \\ (-\frac{1}{2}M_i)^T & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_u} \\ (\frac{1}{2}E_i)^T & \mathbf{0}_{n_x\times n_\lambda} & \mathbf{0}_{n_x} & \mathbf{0}_{n_x\times n_\lambda} & \mathbf{0}_{n_x\times n_u} \\ (\frac{1}{2}F_i)^T & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_u} \\ (\frac{1}{2}H_i)^T & \mathbf{0}_{n_u\times n_\lambda} & \mathbf{0}_{n_u\times n_x} & \mathbf{0}_{n_u\times n_\lambda} & \mathbf{0}_{n_u} \end{bmatrix} \tag{16}$$

with M_i, E_i, F_i, H_i are i^{th} row of M, E, F, H. For the constraint $E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c \geq 0$, it can be written in n_λ constraints:

$$-c_i \leq \mathrm{tr}\left(D_{2,i}X\right) \tag{17}$$

where c_i is i^{th} element of c and $D_{2,i}$ is as follows:

$$D_{2,i} = \begin{bmatrix} 0 & \mathbf{0}_{1\times n_\lambda} & \frac{1}{2}E_i & \frac{1}{2}F_i & \frac{1}{2}H_i \\ \mathbf{0}_{n_\lambda\times 1} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_u} \\ (\frac{1}{2}E_i)^T & \mathbf{0}_{n_x\times n_\lambda} & \mathbf{0}_{n_x} & \mathbf{0}_{n_x\times n_\lambda} & \mathbf{0}_{n_x\times n_u} \\ (\frac{1}{2}F_i)^T & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_u} \\ (\frac{1}{2}H_i)^T & \mathbf{0}_{n_u\times n_\lambda} & \mathbf{0}_{n_u\times n_x} & \mathbf{0}_{n_u\times n_\lambda} & \mathbf{0}_{n_u} \end{bmatrix} \tag{18}$$

with E_i, F_i, H_i are i^{th} row of E, F, H. For the constraint $\delta_k^\lambda + Ms_k \leq M$, it can be written in n_λ constraints:

$$\mathrm{tr}\left(D_{3,i}X\right) \leq M \tag{19}$$

where $D_{3,i}$ is as follows:

$$D_{3,i} = \begin{bmatrix} 0 & \frac{1}{2}M_i & \mathbf{0}_{1\times n_x} & \frac{1}{2}e_{n_\lambda,i}^T & \mathbf{0}_{1\times n_u} \\ (\frac{1}{2}M_i)^T & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_u} \\ \mathbf{0}_{n_x\times 1} & \mathbf{0}_{n_x\times n_\lambda} & \mathbf{0}_{n_x} & \mathbf{0}_{n_x\times n_\lambda} & \mathbf{0}_{n_x\times n_u} \\ \frac{1}{2}e_{n_\lambda,i} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_u} \\ \mathbf{0}_{n_u\times 1} & \mathbf{0}_{n_u\times n_\lambda} & \mathbf{0}_{n_u\times n_x} & \mathbf{0}_{n_u\times n_\lambda} & \mathbf{0}_{n_u} \end{bmatrix} \tag{20}$$

For the constraint $\delta_k^\lambda \geq 0$, it can be written in n_λ constraints:

$$0 \leq \mathrm{tr}\left(D_{4,i}X\right) \tag{21}$$

where $D_{4,i}$ is as follows:

$$D_{4,i} = \begin{bmatrix} 0 & \mathbf{0}_{1\times n_\lambda} & \mathbf{0}_{1\times n_x} & \frac{1}{2}e_{n_\lambda,i}^T & \mathbf{0}_{1\times n_u} \\ \mathbf{0}_{n_\lambda\times 1} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_u} \\ \mathbf{0}_{n_x\times 1} & \mathbf{0}_{n_x\times n_\lambda} & \mathbf{0}_{n_x} & \mathbf{0}_{n_x\times n_\lambda} & \mathbf{0}_{n_x\times n_u} \\ \frac{1}{2}e_{n_\lambda,i} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_x} & \mathbf{0}_{n_\lambda} & \mathbf{0}_{n_\lambda\times n_u} \\ \mathbf{0}_{n_u\times 1} & \mathbf{0}_{n_u\times n_\lambda} & \mathbf{0}_{n_u\times n_x} & \mathbf{0}_{n_u\times n_\lambda} & \mathbf{0}_{n_u} \end{bmatrix} \tag{22}$$

For the constraint $s_k(s_k - 1) = 0$, it can be written as:

$$\mathrm{tr}\left(D_{5,i}X\right) = 0 \tag{23}$$

where $D_{5,i}$ is as follows:

$$D_{5,i} = \begin{bmatrix} 0 & \frac{1}{2}e_{n_\lambda,i}^T & \mathbf{0}_{1\times(n_x+n_\lambda+n_u)} \\ \frac{1}{2}e_{n_\lambda,i} & e_{n_\lambda,i}e_{n_\lambda,i}^T\mathbf{I}_{n_\lambda} & \mathbf{0}_{n_\lambda\times(n_x+n_\lambda+n_u)} \\ \mathbf{0}_{(n_x+n_\lambda+n_u)\times 1} & \mathbf{0}_{(n_x+n_\lambda+n_u)\times n_\lambda} & \mathbf{0}_{n_x+n_\lambda+n_u} \end{bmatrix} \tag{24}$$

where $I_{n_\lambda} \in \mathbb{R}^{n_\lambda}$ is identity matrix.

$\square$

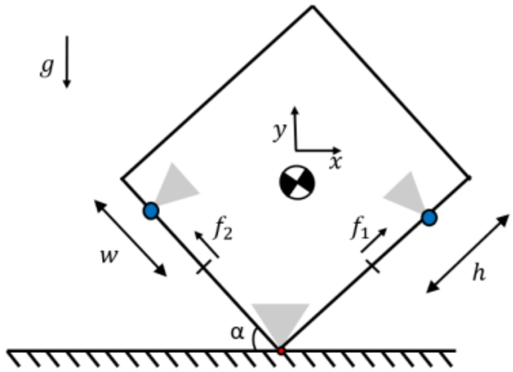

Figure 1: Pivoting a rigid object with two fingers (blue). The object can make and break contact with the ground and gray areas represent the friction cones.

3 Experiments

Suboptimality. In practice, checking the rank condition of the optimal solution of (2) and (11) can be sensitive to numerical thresholds. Therefore, we always project $X^\star$ and $Y^\star$ to a solution $\hat{\delta}$ by Singular Vale Decomposition (SVD) that is also feasible for problem (LCP) and (MIQP). We evaluate the objective of (LCP) and (MIQP) at $\hat{\delta}$, denoted as $\hat{\rho}$ and satisfies

$$f^\star \leq \rho^\star_{\mathsf{MIQP}} = \rho^\star \leq \hat{\rho}. \tag{25}$$

We then compute the *relative suboptimality*

$$\eta = \frac{\hat{\rho} - f^\star}{1 + |f^\star| + |\hat{\rho}|}. \tag{26}$$

Clearly, $\eta = 0$ certifies global optimality of the solution $\hat{\delta}$ and tightness of the SDP relaxation.

Setup. In our current study, we replicate the experimental setup previously described and tested in [1]. To briefly introduce this task, the dynamics of pivoting a rigid object is explored, inspired by the work of Hogan et al. [3], with an emphasis on balancing it at its midpoint. As shown in Figure 1, the interaction involves two fingers (denoted in blue), with their positions relative to the object represented as f_1 and f_2. The object is characterized by a controlled normal force exerted by these fingers, a center of mass at positions x and y, an angle α with the ground, and dimensions $w = 1$, $h = 1$. The friction coefficients are set as $\mu_1 = \mu_2 = 0.1$ for the fingers and $\mu_3 = 1$ for ground contact. The object, with a mass $m = 1$ and subject to a gravitational acceleration $g = 9.81$, is modeled through an implicit time-stepping scheme [5]. The system incorporates 3 contact points and is described by 10 states ($n_x = 10$), 10 complementarity variables ($n_\lambda = 10$), and 4 inputs ($n_u = 4$). For practical implementation, [1] employs a local LCS approximation, recalculated at each time step k. The system's objective is to balance the object at the midpoint ($x = 0$, $y = \sqrt{2}$, $\alpha = \frac{\pi}{4}$) while adjusting the finger positions ($f_1 = f_2 = 0.9$). The controller effectively manages unplanned mode changes caused by process noise, illustrating the method's efficacy with successive linearizations in multi-contact systems that defy single LCS approximations. To compare the time and suboptimality of two SDP solvers and the MIQP solution to projection step, we vary the Gaussian disturbances in the dynamics, with different standard deviations ($\sigma = 0.1, 0.5$).

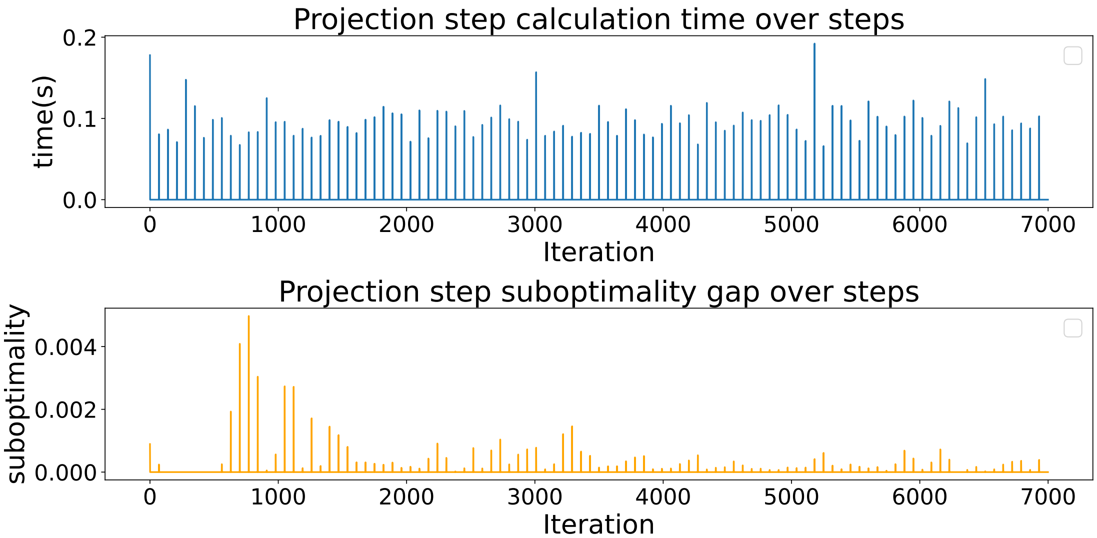

Figure 2: Computational time and suboptimality for LCP-based SDP.

Table 1: Time and suboptimality ($\sigma = 0.5$)

	LCP-based SDP	MIQP-based SDP	MIQP
time(s)	$9.1449e-02 \pm 4.4005e-02$	$1.5137e-01 \pm 4.4231e-02$	$2.1604e-02 \pm 2.8890e-03$
duality gap	$8.9880e-06 \pm 1.1486e-04$	$1.2949e-11 \pm 1.2094e-10$	/

Results. In our comparative analysis, we focused on two projection methods: SDP in both versions and MIQP, which were successful in accomplishing pivoting tasks. In contrast, the LCP and the ADMM methods were not effective in this regard. We plotted the time (noting that each iteration included 10 steps in the Hybrid MPC horizon) and average suboptimality (with $\sigma = 0.5$) for each iteration in our sequence of 7000 iterations, as shown in Figure 2 and Figure 3. Our observations revealed that the iteration time remained stable across the horizon. However, we noted variability in suboptimality, particularly for the LCP-based SDP, where suboptimalities around the 1000th iteration were notably higher than in other iterations.

Further, we conducted Monte Carlo experiments with 20 runs for both $\sigma = 0.5$ and $\sigma = 0.1$, the results of which are detailed in Table 1 and Table 2. The MIQP method demonstrated quick problem-solving capabilities and outperformed SDP solvers, possibly due to the enumeration of a small number of integer variables ($n_\lambda = 10$). Interestingly, the LCP-based SDP exhibited faster computation than the MIQP-based SDP. This might be because the MIQP-based SDP needs to handle a positive semidefinite cone decision variable of size $1 + 2n_\lambda + n_x + n_u$, while the LCP-based SDP manages a smaller size of $1 + n_\lambda + n_x + n_u$. In terms of constraints, the MIQP formulation involves $4n_\lambda$ inequality constraints and n_λ equality constraint, whereas the LCP-based formulation includes $2n_\lambda$ inequality constraints and one equality constraint. Despite this, in terms of accuracy, the SDP method with the MIQP formulation showed significantly better results, by an order of magnitude of 5.

4 Conclusions

This project opens up numerous avenues for further exploration. One of the intriguing questions that remain unanswered is why SDP underperforms compared to MIQP. Additionally, it's unclear

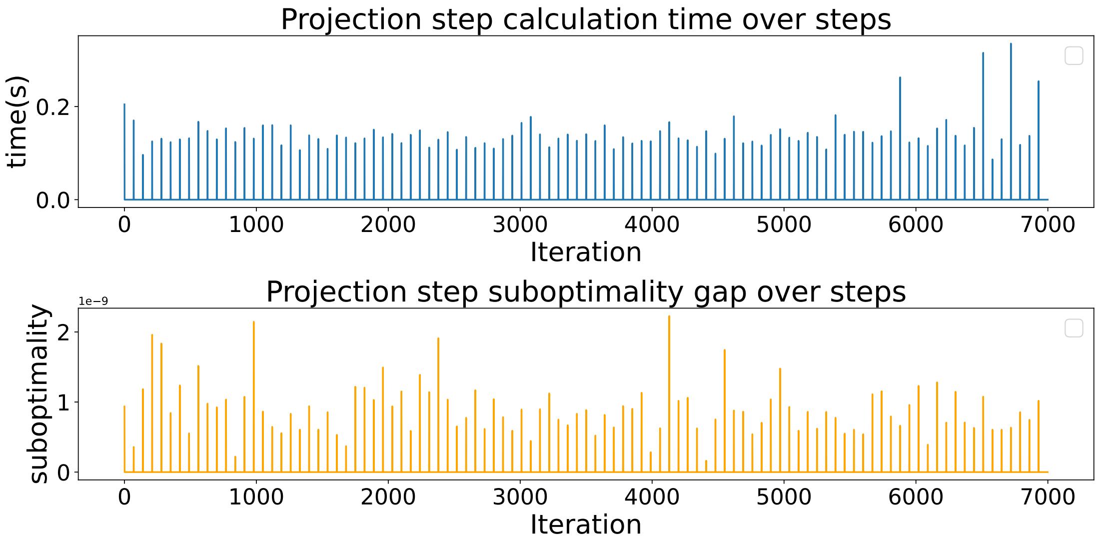

Figure 3: Computational time and suboptimality for MIQP-based SDP.

Table 2: Time and suboptimality ($\sigma = 0.1$)

	LCP-based SDP	MIQP-based SDP	MIQP
time(s)	$9.2539e-02 \pm 2.8506e-02$	$1.5265e-01 \pm 4.4073e-02$	$2.1832e-02 \pm 2.3884e-03$
duality gap	$5.5045e-06 \pm 9.0045e-05$	$1.1824e-11 \pm 1.0462e-10$	/

why the MIQP-based SDP demonstrates higher accuracy than the LCP-based SDP. We are also keen to explore the geometric properties of the SDP formulation in the context of LCP problems. Specifically, we want to understand whether SDP is a geometrically suitable formulation for LCP problems and what might be lost in the Semidefinite Relaxation process.

Another promising direction for extending this work involves hybrid MPC. Our preliminary findings suggest that a wider range of hybrid MPC problems could potentially be formulated using SDP by redefining the decision variables as polynomial constraints. This area, in particular, warrants deeper investigation to fully understand its potential and limitations.

References

[1] Alp Aydinoglu, Adam Wei, and Michael Posa. Consensus complementarity control for multi-contact mpc. *arXiv preprint arXiv:2304.11259*, 2023. 1, 2, 6

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011. 1

[3] Francois R Hogan, Jose Ballester, Siyuan Dong, and Alberto Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 8863–8869. IEEE, 2020. 6

[4] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014. 1

[5] David E Stewart and Jeffrey C Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15):2673–2691, 1996. 6