# Real-Time Multi-Contact Model Predictive Control via ADMM

Alp Aydinoglu[1] and Michael Posa[1]

*Abstract*— We propose a hybrid model predictive control algorithm, consensus complementarity control (C3), for systems that make and break contact with their environment. Many state-of-the-art controllers for tasks which require initiating contact with the environment, such as locomotion and manipulation, require *a priori* mode schedules or are so computationally complex that they cannot run at real-time rates. We present a method, based on the alternating direction method of multipliers (ADMM), capable of high-speed reasoning over potential contact events. Via a consensus formulation, our approach enables parallelization of the contact scheduling problem. We validate our results on three numerical examples, including two frictional contact problems, and physical experimentation on an underactuated multi-contact system.

## I. INTRODUCTION

For many important tasks such as manipulation and locomotion, robots need to make and break contact with the environment. Even though such multi-contact systems are common, they are notoriously hard to control. The main challenge is finding policies and/or trajectories that explicitly consider the interaction of the robot with its environment in order to enable stable, robust motion. For a wide range of problems, it is computationally challenging to discover control policies and/or trajectories [1] and the methods are not suitable for running in real-time speeds for complex problems.

Model predictive control (MPC) is one of the most powerful tools for automatic control [2], [3]. Predominant in robotics are MPC-based methods utilizing linearization, leading to quadratic programs which can be solved efficiently [4], [5]. However, for multi-contact systems, the algorithm must also decide when to initiate or break contact. As a result, linearizations are no longer appropriate and a hybrid formulation is required.

When linearizations are not viable, the resulting MPC algorithm includes the hybrid elements that result from making and breaking of contact. Due to the computational complexity of hybrid MPC, these methods can use a predefined sequence of contacts [6], [7], specialize for legged systems [8], [9], [10], [11], quasi-static manipulation [12], rely on an offline phase [13], [14], or approximate the dynamics via smooth contact models and change the dynamics (diagonal approximations of contact matrix) to enable high performance [15], [16], [17]. An unifying theme across much of prior work is that these approaches require significant domain knowledge for simplifying the search for appropriate mode sequences. On the other hand, developing MPC based control techniques that can reason about contact events and that do not require domain knowledge is a much harder task. Marcucci and Tedrake [18] utilize mixed-integer programming to optimize over control actions and mode sequences, but the method requires significant online computation time. Recently, contact-implicit control with a primal-dual interior-point method has been explored with impressive results on simulation but few details are available and the method has not yet been validated on a physical system [19]. Operator splitting has been used for solving hybrid MPC problems, with local guarantees, but this approach is not suitable for multi-contact control due to exponential increase in the number of constraints based on number of contacts [20].

In contrast to prior work, here we focus on a specific structure in multi-contact robotics and exploit this structure via ADMM [21]. It has been explored for general mixed integer programs [22], [23], [24] but has not been explored for multi-contact systems. In this work, we will focus on local linear complementarity system [25] approximations of the nonlinear plants and explore ADMM for this particular class of multi-contact systems.

The primary contribution of this paper is an algorithm, consensus complementarity control (C3), for solving the hybrid MPC problem approximately for multi-contact systems. We exploit the distributed nature of ADMM and demonstrate that the hard part of the problem, reasoning about contact events, can be parallelized. This enables our algorithm to be fast ($\sim$ 16 Hz for a system with ten states and three contacts), robust to disturbances and also minimizes the effect of control horizon on the run-time of the algorithm. We also demonstrate the effectiveness of our method on numerical examples and present results on a hardware setup. Experimental validation of contact-implicit MPC algorithms is extremely rare. To the best of our knowledge, these are the first real-time control results on an underactuated system as complex and dynamic as the hybrid cart-pole.

## II. BACKGROUND

The function $\mathcal{I}_{\mathcal{A}}$ is the $0 - \infty$ indicator function for an arbitrary set $\mathcal{A}$ such that $\mathcal{I}_{\mathcal{A}}(z) = 0$ if $z \in \mathcal{A}$ and $\mathcal{I}_{\mathcal{A}}(z) = \infty$ if $z \notin \mathcal{A}$. $\mathbb{N}_0$ denotes natural numbers with zero.

### A. Linear Complementarity Problem

In this work, we utilize complementarity problems to represent the contact forces. These models are common in modeling multi-contact robotics problems [26] and can also

be efficiently learned from data [27]. The theory of ==linear complementarity problems (LCP)== is well established [28].

*Definition 1:* Given a vector $q \in \mathbb{R}^m$, and a matrix $F \in \mathbb{R}^{m \times m}$, the $\text{LCP}(q, F)$ describes the following mathematical program:

$$\begin{aligned}
\text{find} \quad & \lambda \in \mathbb{R}^m \\
\text{subject to} \quad & y = F\lambda + q, \\
& 0 \leq \lambda \perp y \geq 0.
\end{aligned}$$

Here, the vector $\lambda$ typically represents the contact forces and slack variables ([26]), $y$ represents the gap function and orthogonality constraint embeds the hybrid structure.

### B. Linear Complementarity Systems

Linear complementarity systems (LCS) are useful because they formulate the discrete-time dynamics of a multi-contact system. We define an LCS as a difference equation coupled with a variable that is the solution of an LCP.

*Definition 2:* A linear complementarity system describes the trajectories $(x_k)_{k \in \mathbb{N}_0}$ and $(\lambda_k)_{k \in \mathbb{N}_0}$ for an input sequence $(u_k)_{k \in \mathbb{N}_0}$ starting from $x_0$ such that

$$\begin{aligned}
& x_{k+1} = Ax_k + Bu_k + D\lambda_k + d, \\
& 0 \leq \lambda_k \perp Ex_k + F\lambda_k + Hu_k + c \geq 0,
\end{aligned} \quad (1)$$

where $x_k \in \mathbb{R}^{n_x}$, $\lambda_k \in \mathbb{R}^{n_\lambda}$, $u_k \in \mathbb{R}^{n_u}$, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $D \in \mathbb{R}^{n_x \times n_\lambda}$, $d \in \mathbb{R}^{n_x}$, $E \in \mathbb{R}^{n_\lambda \times n_x}$, $F \in \mathbb{R}^{n_x \times n_\lambda}$, $H \in \mathbb{R}^{n_\lambda \times n_u}$ and $c \in \mathbb{R}^{n_\lambda}$.

For a given $k$, $x_k$ and $u_k$, the corresponding complementarity variable $\lambda_k$ can be found by solving $\text{LCP}(Ex_k + Hu_k + c, F)$ (see Definition 1). Similarly, $x_{k+1}$ can be computed using the first equation in (1) when $x_k, u_k$ and $\lambda_k$ are known. We focus on cases where $(x_k)_{k \in \mathbb{N}_0}$ is unique, and note that this holds for the examples in the paper. While non-uniqueness can arise due to rigidity [29], control in these pathological settings is outside the scope of this paper.

### III. MODEL PREDICTIVE CONTROL OF MULTI-CONTACT SYSTEMS

#### A. Problem Formulation

In this work, we want to solve the following mathematical optimization problem:

$$\begin{aligned}
\min_{x_k, \lambda_k, u_k} \quad & \sum_{k=0}^{N-1}(x_k^T Q_k x_k + u_k^T R_k u_k) + x_N^T Q_N x_N \\
\text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k + D\lambda_k + d, \\
& Ex_k + F\lambda_k + Hu_k + c \geq 0, \\
& \lambda_k \geq 0, \\
& \lambda_k^T(Ex_k + F\lambda_k + Hu_k + c) = 0, \\
& (\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}) \in \mathcal{C}, \\
& \text{for } k = 0, \dots, N-1, \text{given } x_0,
\end{aligned} \quad (2)$$

where $N$ is the planning horizon, $Q_k, Q_N$ are positive semidefinite matrices, $R_k$ are positive definite matrices and

$\mathcal{C}$ is a convex set (e.g. input bounds, safety constraints, or goal conditions) and $\boldsymbol{x}^T = [x_1^T, \dots, x_N^T]$, $\boldsymbol{\lambda}^T = [\lambda_0^T, \lambda_1^T, \dots, \lambda_{N-1}^T]$, $\boldsymbol{u}^T = [u_0^T, u_1^T, \dots, u_{N-1}^T]$.

Given $x_0$, one solves the optimization (2) and applies $u_0$ to the plant and repeats the process in every time step in a receding horizon manner.

#### B. Mixed Integer Formulation

One straightforward, but computationally expensive, approach to solving (2) is via a mixed integer formulation which exchanges the non-convex orthogonality constraints for binary variables:

$$\begin{aligned}
\min_{x_k, \lambda_k, u_k, s_k} \quad & \sum_{k=0}^{N-1}(x_k^T Q_k x_k + u_k^T R_k u_k) + x_N^T Q_N x_N \\
\text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k + D\lambda_k + d, \\
& Ms_k \geq Ex_k + F\lambda_k + Hu_k + c \geq 0, \\
& M(\boldsymbol{1} - s_k) \geq \lambda_k \geq 0, \\
& (\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}) \in \mathcal{C}, s_k \in \{0, 1\}^{n_\lambda}, \\
& \text{for } k = 0, \dots, N-1, \text{given } x_0,
\end{aligned} \quad (3)$$

where $\boldsymbol{1}$ is a vector of ones, $M$ is a scalar used for the big M method and $s_k$ are the binary variables. This approach requires solving $2^{Nn_\lambda}$ quadratic programs as a worst case. This method is popular because, in practice, it is much faster than this worst case analysis. Even still, for the multi-contact problems explored in this work, directly solving (3) via state-of-the-art commercial solvers remains too slow for real-time control. Methods that learn the MIQP problem offline are promising, but this line of work requires large-scale training on every problem instance [30], [14].

#### C. Consensus Complementarity Control (C3)

Utilizing ADMM, we will solve (2) more quickly than with mixed integer formulations. Since the problem we are addressing is non-convex, our method is not guaranteed to find the global solution or converge unlike MIQP-based approaches, but is significantly faster. We rewrite (2), equivalently, in the consensus form [24] where we create copies (named $\delta_k$) of variables $z_k^T = [x_k^T, \lambda_k^T, u_k^T]$ and move the constraints into the objective function using $0 - \infty$ indicator functions:

$$\begin{aligned}
\min_z \quad & c(z) + \mathcal{I}_{\mathcal{D}}(z) + \mathcal{I}_{\mathcal{C}}(z) + \sum_{k=0}^{N-1} \mathcal{I}_{\mathcal{H}_k}(\delta_k) \\
\text{s.t.} \quad & z_k = \delta_k, \ \forall k,
\end{aligned} \quad (4)$$

where $z^T = [z_0^T, z_1^T, \dots, z_N^T]$, $c(z)$ is the cost function in (2), the set $\mathcal{D}$ includes the dynamics constraints:

$$\cap_{k=0}^{N-1}\{z : x_{k+1} = Ax_k + Bu_k + D\lambda_k + d\},$$

the sets $\mathcal{H}_k$ represent the LCP (contact) constraints:

$$\begin{aligned}
\mathcal{H}_k = \{(x_k, \lambda_k, u_k) : & Ex_k + F\lambda_k + Hu_k + c \geq 0, \\
& \lambda_k \geq 0, \lambda_k^T(Ex_k + F\lambda_k + Hu_k + c) = 0\}.
\end{aligned}$$

Note that we leverage the time-dependent structure in the complementarity constraints to separate $\delta_k$'s.

The general augmented Lagrangian ([21], Section 3.4.2.) for the problem in consensus form (4) is:

$$\mathcal{L}(z, \delta, w) = c(z) + \mathcal{I}_\mathcal{D}(z) + \mathcal{I}_\mathcal{C}(z)$$
$$+ \sum_{i=0}^{N-1} \left( \mathcal{I}_{\mathcal{H}_k}(\delta_k) + (r_k^T G_k r_k - w_k^T G_k w_k) \right),$$

where $w^T = [w_0^T, w_1^T, \ldots, w_{N-1}^T]$, $w_k$ are scaled dual variables, $r_k = z_k - \delta_k + w_k$, $G_k$ is a positive definite matrix, and $\delta^T = [\delta_0^T, \delta_1^T, \ldots, \delta_{N-1}^T]$. Observe that the standard augmented Lagrangian [24] is recovered for $G_k = \rho_k I$.

In order to solve (4), we apply the classical ADMM algorithm, consisting of the following operations:

$$z^{i+1} = \operatorname{argmin}_z \mathcal{L}(z, \delta_0^i, \ldots, \delta_{N-1}^i, w_1^i, \ldots w_{N-1}^i), \quad (5)$$
$$\delta_k^{i+1} = \operatorname{argmin}_{\delta_k} \mathcal{L}(z^{i+1}, \ldots, \delta_k, \ldots, \delta_{N-1}^i, w_0^i, \ldots w_{N-1}^i), \quad (6)$$
$$w_k^{i+1} = w_k^i + z_k^{i+1} - \delta_k^{i+1}. \quad (7)$$

Here, (5) requires solving a quadratic program, (6) is a projection onto the LCP constraints and (7) is a dual variable update. Next, we analyze these operations in the given order.

*1) Quadratic Step:* Equation (5) can be represented by the convex quadratic program

$$\min_z \quad c(z) + \sum_{i=0}^{N-1} (z_k - \delta_k^i + w_k^i)^T G_k (z_k - \delta_k^i + w_k^i) \quad (8)$$
$$\text{s.t.} \quad z \in \mathcal{D} \cap \mathcal{C}.$$

The linear dynamics constraints are captured by the set $\mathcal{D}$, and the convex inequality constraints on states, inputs, contact forces are captured by $\mathcal{C}$. The complementarity constraints do not explicitly appear, but their influence is found, iteratively, through the variables $\delta_k^i$.

The QP in (8) can be solved quickly via off-the-shelf solvers and is analogous to solving the MPC problem for a linear system without contact.

*2) Projection Step:* This step requires projecting onto the LCP constraints $\mathcal{H}_k$ and is the most challenging part of the problem. (6) can be represented with the following quadratic program with a non-convex constraint:

$$\min_{\delta_k} \quad (\delta_k - (z_k^{i+1} + w_k^i))^T G_k (\delta_k - (z_k^{i+1} + w_k^i)) \quad (9)$$
$$\text{s.t.} \quad \delta_k \in \mathcal{H}_k,$$

where $\delta_k = [(\delta_k^x)^T, (\delta_k^\lambda)^T, (\delta_k^u)^T]$. In our setting, we will consider three different projections. Two are approximate projections, common for minimization problems over non-convex sets [31].

**Algorithm 1** Consensus Complementarity Control (C3)

**Require:** $\delta_k^0, w_k^0, G_k, D, \theta, \rho_k, N, x_0$
    *Initialization* : $i = 1$
1: **while** $i \leq \theta$ **do**
2:    Compute $z^{i+1}$ via (8)
3:    Compute $\delta_k^{i+1}$ via (10), $\forall k$
4:    $w_k^{i+1} \leftarrow w_k^i + z_k^{i+1} - \delta_k^{i+1}, \forall k$
5:    $G_k \leftarrow \rho_k G_k, \forall k$
6:    $w_k \leftarrow w_k / \rho_k, \forall k$
7: **end while**
8: **return** $u_0$

*a) MIQP Projection:* The projection can be calculated by exactly formulating (9) as a small-scale MIQP

$$\min_{\delta_k} \quad (\delta_k - (z_k^{i+1} + w_k^i))^T U (\delta_k - (z_k^{i+1} + w_k^i))$$
$$\text{s.t.} \quad E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c \geq 0, \quad (10)$$
$$\delta_k^\lambda \geq 0,$$
$$(\delta_k^\lambda)^T (E\delta_k^x + F\delta_k^\lambda + H\delta_k^u + c) = 0,$$

where $U$ is a positive semi-definite matrix. This is a non-convex QP due to the last (orthogonality) constraint. For $U = G_k$, one recovers the problem in (9); however, in our experience, we found significantly improved performance using alternate, but fixed, choices for $U$. Observe that while (10) is non-convex, it is written only in terms of variables corresponding to a single time step $k$. While the original MIQP formulation in (3) has $Nn_\lambda$ binary variables, here we have $N$ independent problems, each with $n_\lambda$ binary variables. This decoupling leads to dramatically improved performance (worst-case $N2^N$ vs $2^{Nn_\lambda}$).

*b) LCP Projection:* In cases where (10) cannot be solved quickly enough, we propose two approximate solutions with faster run-time. Consider the limiting case where $U$ has no penalty on the force elements. Here, (10) can be solved with optimal objective value of 0 by setting $\delta_k^x = z_k^{(i+1),x} + w_k^{(i),x}$ and $\delta_k^u = z_k^{(i+1),u} + w_k^{(i),u}$. Then, $\delta_k^\lambda$ can be found by solving $\text{LCP}(E\delta_k^x + H\delta_k^u + c, F)$. We note that this projection is different than shooting based methods since we are simulating the $z_k^{(i+1),x,u} + w_k^{(i),x,u}$ instead of $z_k^{(i+1),x,u}$.

*c) ADMM Projection:* We note that (10) is a quadratically constrained quadratic program, where prior work has solved problems of this form using ADMM [24]. This leads to nested ADMM algorithms, but this formulation can produce faster solutions than MIQP solvers without guarantees that it produces a feasible or optimal value. Note that this formulation fared poorly when applied directly to the original problem (2), rarely satisfying the complementarity constraints.

Solving the full MIQP (10) typically is the best at exploring a wide range of modes, while the LCP projection is usually fastest (depending on matrix $F$). ADMM projection can be viewed as a middle ground.

After discussing each of the individual steps, we present the

**3416**

PROJECTION RUN-TIME AND AVERAGED COST-TO-GO

| Projection Method | Mean $\pm$ Std $(s)$ | Cost |
|---|---|---|
| LCP | $1.4 \cdot 10^{-5} \pm 1.8 \cdot 10^{-6}$ | 19.16 |
| MIQP | $1 \cdot 10^{-3} \pm 1.2 \cdot 10^{-4}$ | 27.28 |
| ADMM | $5.3 \cdot 10^{-4} \pm 3.3 \cdot 10^{-5}$ | 33.58 |

full C3 algorithm (Algorithm 1). Here, both $\delta_k^0$ and $w_k^0$ are usually initialized as zero vectors. $G_k$ are positive definite matrices, $\theta > 0$ is the number of ADMM steps, $\rho_k > 0$ is the penalty parameter. Given an horizon $N$, and an initial state $x_0$, the algorithm returns $u_0$ which is applied to the system. This procedure is repeated in every time step as in receding horizon controllers. In the interest of real-time rates, we run a fixed number of ADMM steps ($\theta$).

## IV. NUMERICAL EXAMPLES

For these results, OSQP [32] is used to solve quadratic programs and Gurobi [33] is used for mixed integer programs. PATH [34] and Lemke's algorihm have been used to solve LCP's. SI units (meter, kilogram, second) are used. For all the examples, $G_k = \rho_k G_{k-1}$ is used where $G_0 = G$ is positive definite. The experiments are done on a desktop computer with the processor Intel *i7-11800H* and *16GB RAM*. Reported run-times include all steps in the algorithm. Aside from these third-party solvers, code was implemented in Python. We expect that further improvements in runtime would be achievable using C++. The code for all examples is available[1]. Experiments are also shown in the supplementary video.

### A. Cart-pole with Soft Walls

We consider a cart-pole that can interact with soft walls as in [18], [35]. This is a benchmark in contact-based control algorithms. Here, $x^{(1)}$ represents the position of the cart, $x^{(2)}$ represents the position of the pole and $x^{(3)}, x^{(4)}$ represent their velocities respectively. The forces that affect the pole are described by $\lambda^{(1)}$ and $\lambda^{(2)}$ for right and left walls respectively.

The model is linearized around $x^{(2)} = 0$ where $m_c = 0.978$ is the mass of the cart, $m_p = 0.411$ is the combined mass of the pole and the rod, $l_p = 0.6$ is the length of the pole, $l_c = 0.4267$ is the length of the center of mass position, $k_1 = k_2 = 50$ are the stiffness parameter of the walls, $d = 0.35$ is the distance between the origin and the soft walls. Then, we discretize the dynamics using the explicit Euler method with time step $T_s = 0.01$ to obtain the system matrices and use the model in [35]. We note that the MIQP formulation (as in (3)) runs at approximately 10 Hz.

We design a controller where $\theta = 10$, $G = 0.1I$, $\rho_k = 2$, and $N = 10$. There is a clear trade-off between solve time and planning horizon [36]. We test all three projection methods described in Section III and report run-times (single solve of (10)) on Table I averaged for 1000 solves. We also report

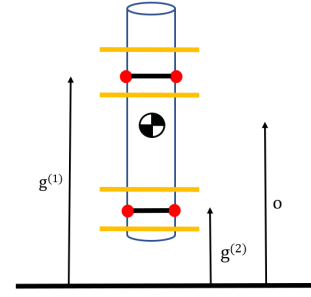[1] https://github.com/AlpAydinoglu/coptimal



Fig. 1. Lifting an object using two grippers indicated by red circles. The soft limits where the grippers should not cross are indicated by yellow lines.

the average of cost-to-go value assuming all of the methods can run at 100 Hz (only the MIQP projection cannot). Even though the MIQP projection is usually better at exploring new contacts, it does not always result in a better cost. We note that the controller can run slightly faster than 240 Hz if LCP-based projection is used. For this example, the QP in (8) has no inequality constraints and can therefore the KKT conditions can be directly solved.

### B. Finger Gaiting

Next, we lift a rigid object upwards using four fingers. The setup for this problem is illustrated in Figure 1. The red circles indicate where the grippers interact the object and we assume that the grippers are always near the surface of the object and the force they apply on the object can be controlled. This force affects the friction between the object and grippers. Since the grippers never leave the surface, we assume that there is no rotation. The goal of this task is to lift the object vertically, while the fingers are constrained to stay close to their original locations (soft constraints shown in yellow). This task, therefore, requires finger gaiting to achieve large vertical motion of the object.

We use the formulation in [26] for modeling the system and denote the positions of the grippers as $g^{(1)}, g^{(2)}$ respectively and position of the object as $o$. We choose $g = 9.81$ as gravitational acceleration and $\mu = 1$ is the coefficient of friction for both grippers.

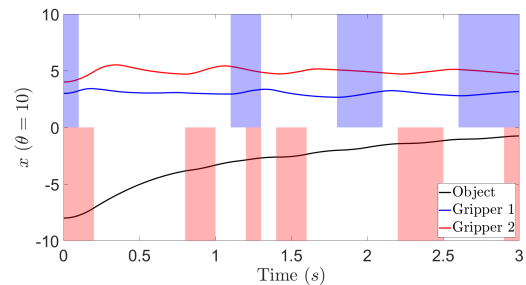We design a two different controllers based on Algorithm



Fig. 2. Finger gaiting with $\theta = 10$. Blue shading implies that gripper 1 is applying normal force to the object whereas red shading implies that gripper 2 is applying normal force.

1 where $G = I$, $N = 10$. Both controllers use the MIQP projection method since other two projection methods failed to find stable motions. For this example, we use $\theta = 10$, $\rho_k = 1.2$. Parallelization leads to $\approx 2.5$x speedup for this particular example and we are able to run at 10 Hz. We also enforce limits:

$$1 \leq g^{(1)} \leq 3, \ \forall k,$$
$$3 \leq g^{(2)} \leq 5, \ \forall k.$$

We performed 100 trials starting from different initial conditions where $o(0) \sim U[-6, -8]$, $g^{(1)}(0) \sim U[2, 3]$, $g^{(2)}(0) \sim U[3, 4]$ are uniformly distributed and both the grippers and the object have zero initial speed. The controller managed to stabilize the system in all cases. As seen in Figure 2, the grippers first throw the object into the air and then catch it followed by some finger gaiting.

### C. Pivoting

In this example, we consider pivoting a rigid object that can make and break contact with the ground inspired by Hogan et. al [37]. Two fingers (indicated via blue) interact with the object as in Figure 3. The goal is to balance the rigid-object at the midpoint.

The positions of the fingers with respect to the object are described via $f_1$, $f_2$ respectively. The normal force that the fingers exert onto the box can be controlled. The center of mass position is denoted by $x$ and $y$ respectively, $\alpha$ denotes the angle with the ground and $w = 1$, $h = 1$ are the dimensions of the object. The coefficient of friction for the fingers are $\mu_1 = \mu_2 = 0.1$, and the coefficient of friction with the ground is $\mu_3 = 1$. We take the gravitational acceleration as $g = 9.81$ and mass of the object as $m = 1$. The fingers start close to the pivot point where $f_1 = -0.3$, $f_2 = -0.7$ and the objects configuration is given by $x = 0$, $y = 1.36$ and $\alpha = 0.2$. The goal is to balance the object at the midpoint ($x = 0$, $y = \sqrt{2}$, $\alpha = \pi/4$) while simultaneously moving the fingers towards the end of the object ($f_1 = f_2 = 0.9$).

We model the system using an implicit time-stepping scheme [26]. The system has 3 contacts, where the finger contacts are represented by 3 complementarity variables each, and the
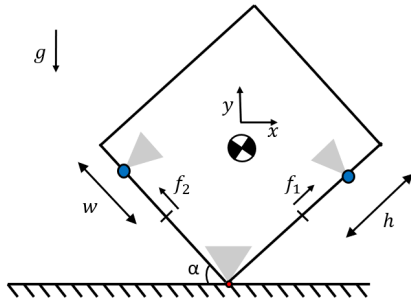


Fig. 3. Pivoting a rigid object with two fingers (blue). The object can make and break contact with the ground and gray areas represent the friction cones.
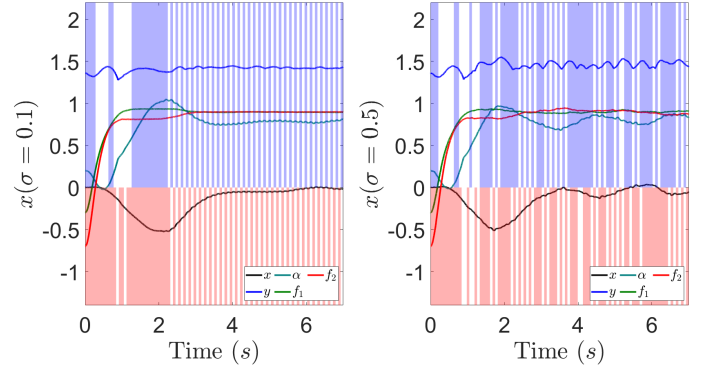


Fig. 4. Pivoting example, Gaussian disturbances with standard deviations $\sigma$. Blue shading implies that gripper 1 is applying normal force to the object whereas red shading implies that gripper 2 is applying normal force.

ground contact is modeled via 4 complementarity variables. The system has ten states ($n_x = 10$), 10 complementarity variables ($n_\lambda = 10$) and 4 inputs ($n_u = 4$). More concretely, it has 36 hybrid modes.

For this example, as the LCS-representation is only an approximation, we compute a new local LCS approximation at every time step $k$. We pick $\theta = 5$, $N = 10$, $\rho_k = 1.1$ and use the local LCS approximation given at time-step $k$ while planning. Parallelization leads to $\approx 4$x speedup for this particular example and controller can run around 16 Hz.

Figure 4 demonstrates the robustness of the controller for different Gaussian disturbances (added to dynamics) with standard deviations (for $\sigma = 0.1, 0.5$). Note that at every time step, all positions and velocities (including angular) are affected by the process noise. The object approximately reaches the desired configuration (midpoint where $\alpha = \frac{\pi}{4}$) for $\sigma = 0, 0.05, 0.1$ and starts failing to get close to the desired configuration for $\sigma = 0.5$. Plots with $\sigma = 0$ and $\sigma = 0.05$ are omitted as those were similar to the one with $\sigma = 0.1$. We emphasize that the process noise causes unplanned mode changes and the controller seamlessly reacts. This example demonstrates that our method works well with successive linearizations as many multi-contact systems can not be captured via a single LCS approximation.

## V. EXPERIMENTAL VALIDATION

We test the multi-contact MPC algorithm on an experimental cart-pole system with soft walls shown in Figure 5 replicating the system described in IV-A. A DC motor with a belt drive generates the linear motion of the cart. Soft walls are made of open-cell polyurethane foam.

We consider the linear model as in IV-A (also used in [35]) where $m_c = 0.978$ is the mass of the cart, $m_p = 0.411$ is the mass of the pole and the rod, $l_p = 0.6$ is the length of the pole, $l_c = 0.4267$ is the length of the center of mass position, $k_1 = k_2 = 100$ are the stiffness parameters of the walls and $d = 0.39$ is the distance between the origin and walls.
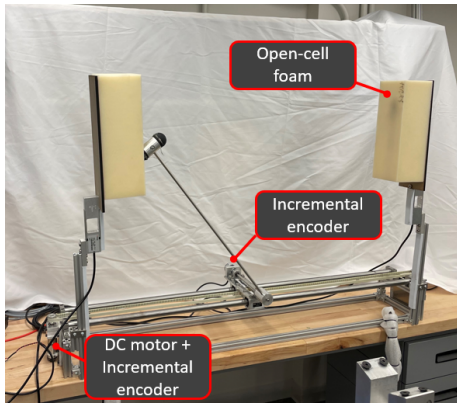
Fig. 5.    Experimental setup for cart-pole with soft walls.



Fig. 7.    Cost-to-go values for the cart-pole experiment.

The parameters of the MPC algorithm are $N = 10$, $\rho = 2.3$, $\theta = 10$, and $G = 0.5I$. We use the LCP-based projection method, and solve the quadratic programs via utilizing the KKT system. We note that the algorithm is capable of running faster than 240 Hz as mentioned previously, but due to high communication delays between our motor controller and computer, we run the system at 100 Hz. In Figure 6, we illustrate, for one particular state, the evolution of the contact force throughout the ADMM process (at ADMM steps 1, 3, and 10). Notice that as the the algorithm progresses, contact forces become more realistic.

We initialize the cart at the origin and introduce random perturbations to cover a wide range of initial conditions that lead to contact events. Specifically, we start the cart-pole at the origin where our controller is active. Then, we apply an input disturbance $u_d \sim U[10, 15]$ for 250 ms to force contact events. We repeated this experiment 10 times and our controller managed to stabilize the system in all trials.

To empirically evaluate the gap between C3, which is sub-optimal, and true solutions to (2), and to assess the impact of modeling errors, we report the cost-to-go values for our method, MIQP solution (as in (3)), and the actual observed states. More precisely, given the current state of the nonlinear plant, we calculate the cost as in (2) using the inputs recovered from both the C3 algorithm and MIQP algorithm.
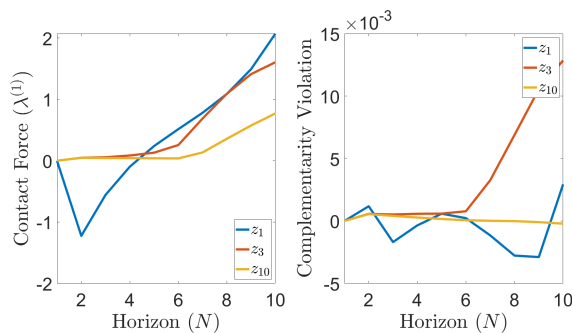


Fig. 6.    Evolution of contact force and complementarity violation during ADMM iteration when the cart is close to a contact surface.
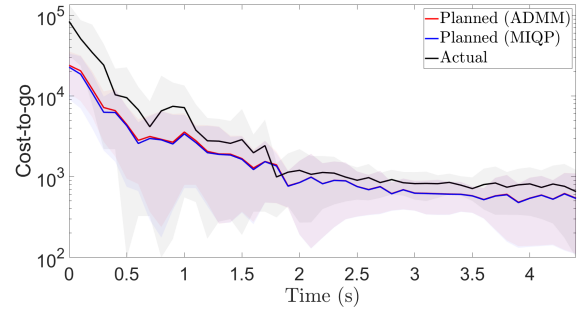
Since C3 algorithm is not guaranteed to produce a $(x, u, \lambda)$ that strictly satisfies complementarity, the predicted cost may not match the simulated cost once $u$ is applied. For the actual plant, we use the data from $N$ steps into the future and calculate the same cost. Notice that even though the cost-to-go of MIQP solution is always lower, as expected, the optimality gap is fairly small. This highlights that C3 finds near-optimal solutions, at least for this particular problem. Also, notice that the actual cost-to-go matches the planned one which further motivates the applicability of LCS representations in model-based control for nonlinear multi-contact systems.

## VI. CONCLUSION

In this work, we present an algorithm, C3, for model predictive control of multi-contact systems. The algorithm relies on solving QP's accompanied by projections and both can be solved efficiently for multi-contact systems. The effectiveness of our approach is verified on three numerical examples and our results are validated on an experimental setup. For fairly complex examples with frictional contact, our method has a fast run-time. We also demonstrate with the experiment that our heuristic can find close-to-optimal strategies.

The framework tackles the hybrid MPC problem by shifting the complexity to the projection sub-problems. While projections are decoupled in time, they are still difficult to solve. Here, we demonstrate three potential approaches to this projection stage, but exploring alternative heuristics is of future interest. In addition, the choice of parameters (such as $G$ and $U$) greatly affects the performance of the algorithm and exploring different approaches for deciding on parameters is of future interest too. Further evaluation, for instance on true dexterous manipulation, and the integration with learned models [27] is in the scope of future work. Additionally, we anticipate that code optimization will lead significant improvements (3x or more) in control rate.

### ACKNOWLEDGEMENTS

REFERENCES

[1] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[2] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[3] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[4] Y. Ding, A. Pandala, and H.-W. Park, "Real-time model predictive control for versatile dynamic motions in quadrupedal robots," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8484–8490, IEEE, 2019.

[5] A. Zhakatayev, B. Rakhim, O. Adiyatov, A. Baimyshev, and H. A. Varol, "Successive linearization based model predictive control of variable stiffness actuated robots," in *2017 IEEE international conference on advanced intelligent mechatronics (AIM)*, pp. 1774–1779, IEEE, 2017.

[6] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2536–2542, IEEE, 2020.

[7] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

[8] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[9] G. Bledt, *Regularized predictive control framework for robust dynamic legged locomotion*. PhD thesis, Massachusetts Institute of Technology, 2020.

[10] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, 2020.

[11] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous robots*, vol. 40, no. 3, pp. 429–455, 2016.

[12] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 755–773, 2020.

[13] T. Marcucci, R. Deits, M. Gabiccini, A. Bicchi, and R. Tedrake, "Approximate hybrid model predictive control for multi-contact push recovery in complex environments," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 31–38, IEEE, 2017.

[14] A. Aydinoglu, M. Fazlyab, M. Morari, and M. Posa, "Stability analysis of complementarity systems with neural network controllers," in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, pp. 1–10, 2021.

[15] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4906–4913, IEEE, 2012.

[16] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the hrp-2 humanoid," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3346–3351, IEEE, 2015.

[17] V. Kumar, Y. Tassa, T. Erez, and E. Todorov, "Real-time behaviour synthesis for dynamic hand-manipulation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6808–6815, IEEE, 2014.

[18] T. Marcucci and R. Tedrake, "Warm start of mixed-integer programs for model predictive control of hybrid systems," *IEEE Transactions on Automatic Control*, 2020.

[19] S. L. Cleac'h, T. Howell, M. Schwager, and Z. Manchester, "Linear contact-implicit model-predictive control," *arXiv preprint arXiv:2107.05616*, 2021.

[20] D. Frick, A. Georghiou, J. L. Jerez, A. Domahidi, and M. Morari, "Low-complexity method for hybrid mpc with local guarantees," *SIAM Journal on Control and Optimization*, vol. 57, no. 4, pp. 2328–2361, 2019.

[21] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[22] B. Stellato, V. V. Naik, A. Bemporad, P. Goulart, and S. Boyd, "Embedded mixed-integer quadratic optimization using the osqp solver," in *2018 European Control Conference (ECC)*, pp. 1536–1541, IEEE, 2018.

[23] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, "A simple effective heuristic for embedded mixed-integer quadratic programming," *International journal of control*, vol. 93, no. 1, pp. 2–12, 2020.

[24] J. Park and S. Boyd, "General heuristics for nonconvex quadratically constrained quadratic programming," *arXiv preprint arXiv:1703.07870*, 2017.

[25] W. Heemels, J. M. Schumacher, and S. Weiland, "Linear complementarity systems," *SIAM journal on applied mathematics*, vol. 60, no. 4, pp. 1234–1269, 2000.

[26] D. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with coulomb friction," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1, pp. 162–169, IEEE, 2000.

[27] S. Pfrommer, M. Halm, and M. Posa, "Contactnets: Learning discontinuous contact dynamics with smooth, implicit representations," *arXiv preprint arXiv:2009.11193*, 2020.

[28] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The linear complementarity problem*. SIAM, 2009.

[29] M. Halm and M. Posa, "Modeling and analysis of non-unique behaviors in multiple frictional impacts," *arXiv preprint arXiv:1902.01462*, 2019.

[30] A. Cauligi, P. Culbertson, B. Stellato, D. Bertsimas, M. Schwager, and M. Pavone, "Learning mixed-integer convex optimization strategies for robot planning and control," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 1698–1705, IEEE, 2020.

[31] S. Diamond, R. Takapoui, and S. Boyd, "A general system for heuristic minimization of convex functions over non-convex sets," *Optimization Methods and Software*, vol. 33, no. 1, pp. 165–193, 2018.

[32] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[33] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2021.

[34] S. P. Dirkse and M. C. Ferris, "The path solver: a nommonotone stabilization scheme for mixed complementarity problems," *Optimization methods and software*, vol. 5, no. 2, pp. 123–156, 1995.

[35] A. Aydinoglu, P. Sieg, V. M. Preciado, and M. Posa, "Stabilization of complementarity systems via contact-aware controllers," *arXiv preprint arXiv:2008.02104*, 2020.

[36] H. Li, R. J. Frei, and P. M. Wensing, "Model hierarchy predictive control of robotic systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3373–3380, 2021.

[37] F. R. Hogan, J. Ballester, S. Dong, and A. Rodriguez, "Tactile dexterity: Manipulation primitives with tactile feedback," in *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 8863–8869, IEEE, 2020.