



MIT Open Access Articles

Reactive Planar Manipulation with Convex Hybrid MPC

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation	Hogan, Francois Robert, Grau, Eudald Romo and Rodriguez, Alberto. 2018. "Reactive Planar Manipulation with Convex Hybrid MPC." 2018 IEEE International Conference on Robotics and Automation (ICRA).
As Published	10.1109/icra.2018.8461175
Publisher	IEEE
Version	Author's final manuscript
Citable link	https://hdl.handle.net/1721.1/136981

Reactive Planar Manipulation with Convex Hybrid MPC

Francois Robert Hogan, Eudald Romo Grau, and Alberto Rodriguez
Department of Mechanical Engineering — Massachusetts Institute of Technology
<fhogan, eudald, albertor>@mit.edu

Abstract—This paper presents a reactive controller for planar manipulation tasks that leverages machine learning to achieve real-time performance. The approach is based on a Model Predictive Control (MPC) formulation, where the goal is to find an optimal sequence of robot motions to achieve a desired object motion. Due to the multiple contact modes associated with frictional interactions, the resulting optimization program suffers from combinatorial complexity when tasked with determining the optimal sequence of modes.

To overcome this difficulty, we formulate the search for the optimal mode sequences offline, separately from the search for optimal control inputs online. Using tools from machine learning, this leads to a convex hybrid MPC program that can be solved in real-time. We validate our algorithm on a planar manipulation experimental setup where results show that the convex hybrid MPC formulation with learned modes achieves good closed-loop performance on a trajectory tracking problem.

I. INTRODUCTION

While humans naturally make use of sensing and feedback when manipulating objects, robot manipulators traditionally execute actions relying on open-loop control strategies. Given the uncertainty associated with frictional contact interactions [1], [2] and the inherent inaccuracies of contact models [3], [4], the use of feedback can play an important role to address model uncertainty. The long term goal of this work is to endow robots with real-time decision making capabilities to enable reactive manipulation.

This paper focuses on planar manipulation tasks where the physical interactions between manipulator, object, and environment can be modeled from first principles, using rigid body dynamics and Coulomb’s frictional law. A major challenge concerning feedback controller design for systems involving contact interactions is the presence of hybridness and underactuation [5]. Hybridness refers to the fact that frictional interactions between manipulator and object exhibit different contact modes (e.g. contact/separation, sticking/sliding, etc), while underactuation is a result of the limited set of forces and torques that can be transmitted by the robot to the object through frictional interactions. Figure 1 shows an animation of a hybrid manipulation task that exploits multiple contact modalities.

This paper’s main contribution is twofold. First, we present a controller design formulation that can be used to manipulate an object on a flat surface. The approach presented in this paper generalizes to multiple contact interactions

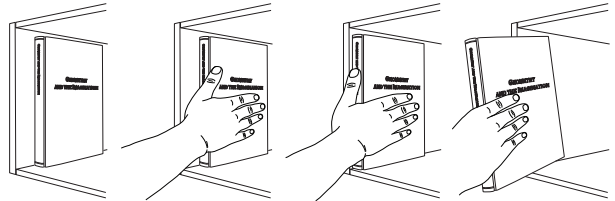


Fig. 1: Animation of a hybrid manipulation task. The hand can interact with the book using one or many contact configurations and by exploiting different contact modalities, namely separation, sticking, and sliding. Figure adapted from [5].

between manipulator and object and for tracking trajectories in the plane. Second, we introduce a method to determine an effective contact mode sequence that leads to a convex hybrid Model Predictive Control (MPC) formulation. Due to the hybridness associated with frictional interactions, hybrid MPC formulations suffer from a combinatorial expansion due to unknown future contact interactions modes. To overcome this difficulty, we formulate the search for optimal modes separately from the search for optimal control inputs, by leveraging machine learning methods to select mode sequences from prior experience. Once the mode sequences are selected, the control problem reduces to solving a convex quadratic program, which can be achieved at very high frequency.

II. RELATED WORK

The mechanics of planar pushing manipulation tasks were first described by [6]. [7] introduced the concept of the *limit surface*, a useful geometric representation which maps the applied frictional forces on an object to its instantaneous velocity. Under the assumption of quasi-static interactions, the limit surface has been successfully used in simulation [6], planning [8], state estimation [9], and feedback control [5], [10], [11] applications. Due to the high computational costs associated with building a true limit surface, [12] proposed an ellipsoidal approximation which yields invertible models from force to motion. Recently, [13] exploited the convex properties of the limit surface to develop an efficient data-driven algorithm for its construction from contact interactions.

There is an ongoing effort to find planning frameworks that can effectively handle the underactuation and hybrid-

This work was supported by NSF award [IIS-1637753] through the National Robotics Initiative.

ness associated with contact models. [14] used sampled-based planning algorithms to plan robot motions for in-hand manipulation tasks. [15] developed a nonlinear trajectory optimization framework that includes frictional forces as decision variables within a nonlinear program and makes use of complementary constraints to encode the different contact interaction modes of the system. Another approach that shows promise is based on Differential Dynamic Programming (DDP), which iteratively approximates locally-quadratic models of the dynamics and cost functions to find a locally optimal path. Most approaches in this line rely on approximating discontinuous dynamics with continuous relaxations. [16] used penalty methods to smooth contact models while [17] modeled the discontinuous dynamics of the system with mixture models.

The application of model-based feedback control to contact rich tasks has been limited to a small number of applications [11], [18], [19]. The control strategies presented in the aforementioned papers are applied to systems with an a priori knowledge of the contact mode sequencing. In [5], a feedback controller design is presented for the pusher-slider system using a Model Predictive Control framework, where a set of contact mode schedules are chosen such that they span a number of dynamic behaviors likely to occur. This method has been shown to work experimentally but requires heuristic methods in order to design candidate contact mode schedules. This paper aims to eliminate the need for contact mode enumeration based on human intuition by developing an algorithm that systematically selects contact modes sequences.

III. NOMENCLATURE

The notation used in the paper is described below:

- H : Convex set representing the limit surface.
- $\mathbf{w} = [f_x \ f_y \ \tau]^T$: Applied wrench on the object resolved in the body frame.
- $\mathbf{t} = [v_x \ v_y \ \omega]^T$: Object twist resolved in the body frame.
- \mathbf{J}_c : Jacobian matrix associated with the contact point c resolved in the body frame.
- $\mathbf{N} = [\mathbf{n}_1^T \ \dots \ \mathbf{n}_C^T]^T$: Matrix of object normal vectors at contact points resolved in body frame.
- $\mathbf{D} = [\mathbf{d}_1^T \ \dots \ \mathbf{d}_C^T]^T$: Matrix of object tangent vectors at contact points resolved in body frame.
- $\mathbf{f}_n = [f_{n,1} \ \dots \ f_{n,C}]^T$: Vector of applied normal forces at contact points resolved in body frame.
- $\mathbf{f}_t = [f_{t,1} \ \dots \ f_{t,C}]^T$: Vector of applied tangential forces at contact points resolved in body frame.
- $\boldsymbol{\phi} = [\phi_1 \ \dots \ \phi_C]^T$: Vector of relative angles of pusher relative to body frame.
- $\mathbf{x} = [x \ y \ \theta \ \boldsymbol{\phi}^T]^T$: System state vector.
- $\mathbf{u}_c = [f_{n,c} \ f_{t,c} \ \phi_c]^T$: Vector of control inputs at contact point c .
- $\mathbf{u}_f = [\mathbf{f}_n^T \ \mathbf{f}_t^T]^T$: Vector of commanded reaction forces.
- $\mathbf{u}_\phi = \boldsymbol{\phi}$: Vector of commanded angular velocities.
- $\mathbf{u} = [\mathbf{u}_f^T \ \mathbf{u}_\phi^T]^T$: Control input.

IV. PLANAR PUSHING MODEL

This section introduces the motion model for planar manipulation tasks that is general to an arbitrary number of contact points and arbitrary object shapes. We adapt the modeling of planar pushing interactions from [20] that we briefly summarize below. All contacts in this work assume Coulomb friction interactions [21], uniform pressure distribution, and quasi-static interactions, where the inertial forces of the object are negligible.

A. Motion Model

The limit surface is a geometric representation that describes the relationship between the applied force on an object and its instantaneous velocity. In this paper, we use the ellipsoidal approximation to the limit surface [12], [20] due its simplicity and invertibility properties. The ellipsoidal limit surface can be expressed in convex quadratic form as $H(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w}$. By the principle of maximal dissipation [7], the object twist is perpendicular to the limit surface for a given wrench

$$\mathbf{t} = \nabla H(\mathbf{w}) = \mathbf{A}\mathbf{w}, \quad (1)$$

where the applied frictional wrench is

$$\mathbf{w} = \sum_{c=1}^C \mathbf{J}_c^T (\mathbf{n}_c f_{n,c} + \mathbf{d}_c f_{t,c}). \quad (2)$$

Consider the planar manipulation task with multiple contact points shown in Fig. 2. The unconstrained motion equations of the system can be expressed as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{R}\mathbf{t} \\ \mathbf{u}_\phi \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

assuming that all points maintain contact with the sliding object.

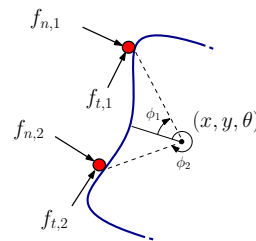
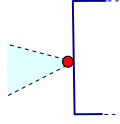


Fig. 2: Free body diagram of a sliding object with $c = 2$ contact points.

B. Frictional Constraints

The motion equations in (3) do not enforce that the reaction forces between manipulator and sliding object are feasible. To ensure that the motion equations are associated with physically reasonable behavior, we must impose constraints on the control input \mathbf{u} , ensuring that the motion model obeys contact interactions laws. Due to the hybrid nature of contact, the physical constraints that dictate the magnitude and direction of the frictional forces vary with

the contact interaction mode. In accordance with Coulomb's frictional law, the following constraints on the inputs must always be satisfied independently of the contact mode:



$$\mathcal{C}_0 : \begin{cases} f_{n,c} \geq 0, \\ |f_{t,c}| \geq \mu_p f_{n,c}, \end{cases} \quad (4)$$

Fig. 3: Friction cone constraint. The applied force must remain within the blue shaded region.

implying the pusher can only exert a compressive force on the object and that the net frictional force applied on the object remains within the bounds of the friction cone in Fig. 3. In addition, we must enforce constraints that depend on the contact interaction mode.

Sticking When the pusher is sticking relative to the object, the tangential velocity is stationary, as in Fig. 4(a)

$$\mathcal{C}_1 : \dot{\phi}_c = 0. \quad (5)$$

Sliding Left When the pusher is sliding left relative to the object, the tangential velocity is strictly positive and the frictional force must remain on the right hand side of the friction cone, as in Fig. 4(b)

$$\mathcal{C}_2 : \begin{cases} \dot{\phi}_c > 0, \\ f_{t,c} = \mu_p f_{n,c}. \end{cases} \quad (6)$$

Sliding Right When the pusher is sliding right relative to the object, the tangential velocity is strictly negative and the frictional force is constrained to remain on the left hand side of the friction cone, as in Fig. 4(c)

$$\mathcal{C}_3 : \begin{cases} \dot{\phi}_c < 0, \\ f_{t,c} = -\mu_p f_{n,c}. \end{cases} \quad (7)$$

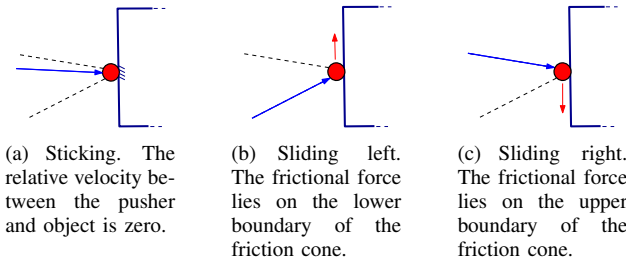


Fig. 4: Mode dependent constraints following Coulomb's frictional interaction law.

V. HYBRID MODEL-PREDICTIVE CONTROL

This section presents a controller design framework for planar manipulation tasks. The proposed controller aims to stabilize the motion of a given object about a nominal trajectory. The approach presented follows a Model Predictive

Control (MPC) formulation where the goal is to determine a sequence of control inputs over a receding horizon to minimize the error between the manipulated object and its desired motion. Due to the hybridness associated with Coulomb's frictional law, the optimization program takes the form of a mixed-integer quadratic program (MIQP).

Optimization Problem MPC (MIQP): Given current error state $\bar{\mathbf{x}}_0$ and nominal trajectory $(\mathbf{x}_i^*, \mathbf{u}_i^*)$, solve

$$\begin{aligned} \min_{\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \mathbf{z}_i} \quad & \bar{\mathbf{x}}_N^\top \mathbf{Q}_N \bar{\mathbf{x}}_N + \sum_{i=0}^{N-1} \left(\bar{\mathbf{x}}_{i+1}^\top \mathbf{Q} \bar{\mathbf{x}}_{i+1} + \bar{\mathbf{u}}_i^\top \mathbf{R} \bar{\mathbf{u}}_i + \mathbf{z}_i^\top \mathbf{W} \mathbf{z}_i \right) \\ \text{subject to} \quad & \bar{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_i + h [\mathbf{A}_i \bar{\mathbf{x}}_i + \mathbf{B}_i \bar{\mathbf{u}}_i], \\ & \bar{\mathbf{u}}_{c,i} \in \mathcal{C}_0, \\ & \bar{\mathbf{u}}_{c,i} \in \mathcal{C}_1 \quad \text{if } c \text{ is sticking (i.e., } z_{1c,i} = 1), \\ & \bar{\mathbf{u}}_{c,i} \in \mathcal{C}_2 \quad \text{if } c \text{ is sliding left (i.e., } z_{2c,i} = 1), \\ & \bar{\mathbf{u}}_{c,i} \in \mathcal{C}_3 \quad \text{if } c \text{ is sliding right (i.e., } z_{3c,i} = 1), \\ & z_{1c,i} + z_{2c,i} + z_{3c,i} = 1, \end{aligned} \quad (8)$$

with $\bar{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{x}_i^*$, $\bar{\mathbf{u}}_i = \mathbf{u}_i - \mathbf{u}_i^*$, and $\mathbf{z}_i = [z_{1c,i}, z_{2c,i}, z_{3c,i}]^\top$. The terms \mathbf{Q} , \mathbf{Q}_N , \mathbf{R} , and \mathbf{W} denote weight matrices associated with the error state, final error state, control input, and contact modes, respectively. We constrain the search to the linearized dynamics of the system and the contact constraints presented in sections IV, where $\mathbf{A}_i = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\mathbf{x}_i^*, \mathbf{u}_i^*}$ and $\mathbf{B}_i = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}|_{\mathbf{x}_i^*, \mathbf{u}_i^*}$.

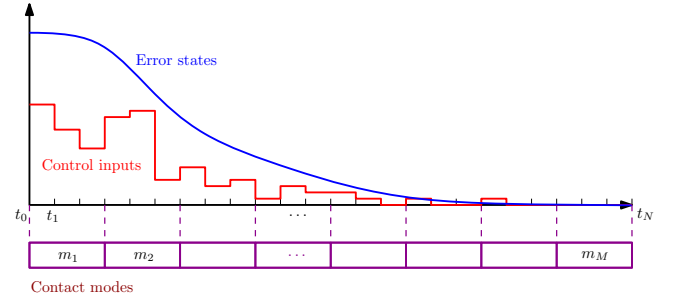


Fig. 5: Hybrid MPC framework. A sequence of control inputs is computed that will drive the predicted states to the reference trajectory while simultaneously finding the schedule of optimal hybrid mode transitions $\mathbf{m} = \{m_1, \dots, m_M\}$. The control input $\bar{\mathbf{u}}_0 + \mathbf{u}_0^*$ is applied to the system.

We introduce integer variables into the optimization program to denote the hybrid mode that is active at time step i . The integer variables $z_{1c,i}$, $z_{2c,i}$, and $z_{3c,i} \in \{0, 1\}$, denote sticking, sliding left, and sliding right for contact point c , respectively, where the integer variable takes the value of 1 if a contact mode is active and 0 otherwise. We enforce that the sum of integers values must be unity at each time step to ensure that only one mode can be active at a time.

To speed up computation, it is often practical to constraint adjacent time steps within a prediction horizon to have the same contact mode. This is shown in Fig. 5, where the agglomerated mode sequence $\mathbf{m} = \{m_1, \dots, m_M\}$ is introduced, with $m_m \in \{1, 2, 3\}$ denoting sticking, sliding left, and sliding

right.

VI. OFFLINE MODE SCHEDULE LEARNING

We can visualize the feedback control architecture proposed in Section V in block diagram form in Fig. 6. Due to the non-convex nature of integer variables in Eq. (8), the solution time is typically slow and not appropriate for high bandwidth feedback control applications. In an effort to increase the control bandwidth, we aim to offload as much of the computational costs offline as possible. To accomplish this, we present a novel formulation that separates the search for the mode schedule selection from the optimal control sequence.

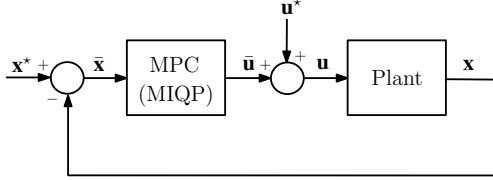


Fig. 6: Block diagram of hybrid controller design described in Eq. (8). The resulting MPC controller design is a non-convex mixed-integer quadratic program.

Consider the controller design architecture proposed in Figure 7. Suppose that given the state error $\bar{\mathbf{x}}$, we had access to an oracle function that returned an effective mode schedule $\mathbf{m} = \{m_1, m_2, \dots, m_N\}$ to be enforced during the prediction horizon. Although we do not have direct access to a real-

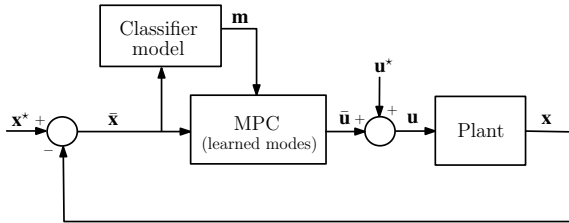


Fig. 7: Block diagram of hybrid controller design with learned mode schedule classifier. The resulting MPC controller design is a convex quadratic program.

time function that determines the optimal mode schedule, we can query the mixed-integer MPC program as much as desired offline to find optimal mode sequences given error state inputs. This formulation lends itself well to a supervised learning setting, where the objective is to train a classifier model that can select an effective mode schedule given the error state. We present the learning framework used to design the classifier model shown in Fig. 8. Using the hybrid MPC (MIQP) formulation presented in Eq. (8), we generate a dataset of E training example $\{\bar{\mathbf{x}}_e, \mathbf{m}_e\}$, where \mathbf{m}_e represents the mode schedule associated with the e^{th} datapoint. The purpose of the machine learning algorithm is to train a candidate classifier model that minimizes the cross-entropy error function of the labelled training set. This new hybrid control architecture leads to a convex optimization program

with a prescribed mode sequence and is referred to as MPC (learned modes).

Optimization Problem MPC (learned modes): Given current error state $\bar{\mathbf{x}}_0$, nominal trajectory $(\mathbf{x}_i^*, \mathbf{u}_i^*)$, and mode schedule \mathbf{m} , solve

$$\begin{aligned} \min_{\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i} \quad & \bar{\mathbf{x}}_N^T \mathbf{Q}_N \bar{\mathbf{x}}_N + \sum_{i=0}^{N-1} \left(\bar{\mathbf{x}}_{i+1}^T \mathbf{Q} \bar{\mathbf{x}}_{i+1} + \bar{\mathbf{u}}_i^T \mathbf{R} \bar{\mathbf{u}}_i \right) \\ \text{subject to} \quad & \bar{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_i + h [\mathbf{A}_i \bar{\mathbf{x}}_i + \mathbf{B}_i \bar{\mathbf{u}}_i], \\ & \bar{\mathbf{u}}_{c,i} \in \mathcal{C}_0, \\ & \bar{\mathbf{u}}_{c,i} \in \mathcal{C}_1 \quad \text{if } c \text{ is sticking,} \\ & \bar{\mathbf{u}}_{c,i} \in \mathcal{C}_2 \quad \text{if } c \text{ is sliding left,} \\ & \bar{\mathbf{u}}_{c,i} \in \mathcal{C}_3 \quad \text{if } c \text{ is sliding right.} \end{aligned}$$

The main attraction of this approach is to convert a non-convex mixed-integer quadratic program into a convex quadratic program that can be solved in real-time.

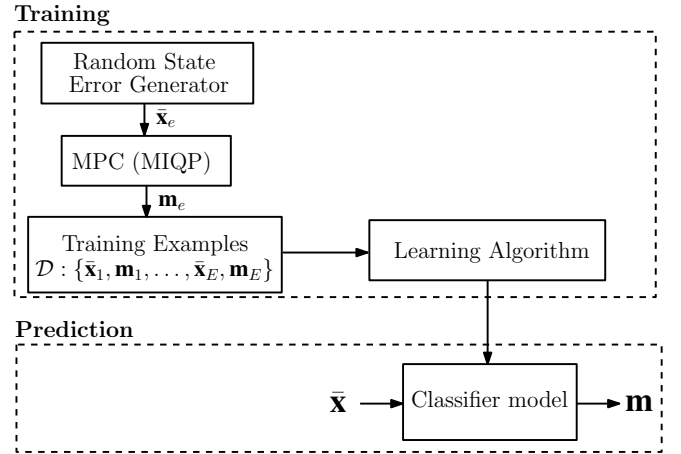


Fig. 8: Supervised learning framework for mode schedule selection. A dataset of E labelled datapoints is generated using the the MPC (MIQP) formulation. From the training examples, a classifier is trained to return the mode schedule based on the state error vector.

VII. RESULTS

In this section, we implement the controller design presented in Section V along with the planar pushing model described in Section IV-A on a planar pushing experimental setup. Videos of the experiments can be found at https://youtu.be/bMMLkyue_ZU. In both of the experiments considered in this section, we parametrize the classifier model introduced in Fig. 7 with a neural network with the properties summarized in Table 1.

A. Case Study A: Single Point Pushing

First, we investigate the performance of the controller design in Fig. 7 on a planar manipulation system where the goal is to track a 2d trajectory in the shape of an 8 shaped trajectory defined by two circles of radii 0.15 meters at a constant velocity of $v = 0.05$ [m/s]. We build the classifier

TABLE I: Experiment parameters.

Property	Symbol	Value
Coefficient of friction (pusher-slider)	μ_p	0.3
Coefficient of friction (slider-table)	μ_g	0.35
Mass of slider (experiment A), kg	m	0.827
Object radius (experiment A), m	r	0.045
Mass of slider (experiment B), kg	m	0.827
Object side length (experiment B), m	a	0.09
Line pusher width (experiment B), m	d	0.03



Fig. 9: Experimental setup for point pusher.

model following the learning framework displayed in Fig. 8, where the training examples are generated using the MPC (MIQP) program presented in Eq. (8). We parametrize the

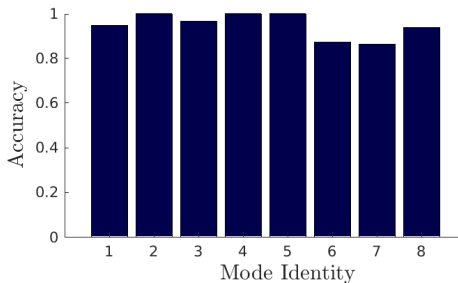


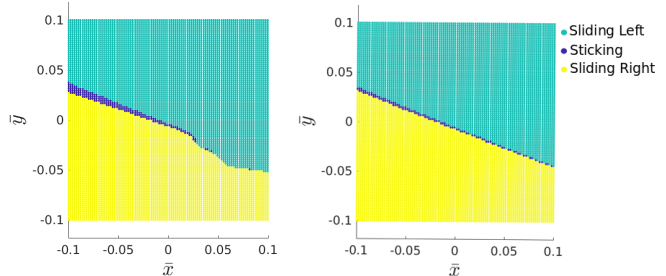
Fig. 10: Accuracy results of the neural network predictions on a validation set of 58815 labelled data points. We evaluate the performance on each mode separately, as defined in Fig. 5.

classifier using a neural network, as detailed in Table 1. The physical properties of the circular object along with the frictional properties of the system are in Table 2. The controller design parameters used in the numerical simulations are $h = 0.3$ seconds, $N = 35$, $\mathbf{Q} = 10 \text{ diag}\{3, 3, 0.1, 0\}$, $\mathbf{Q}_N = 2000 \text{ diag}\{3, 3, 0.1, 0\}$, and $\mathbf{R} = 0.5 \text{ diag}\{1, 1, 0.01\}$. The prediction horizon is split into 8 parts during which the contact modes are held constant. The number of time steps associated with each contact mode section m_m is $\{1, 5, 5, 5, 5, 5, 4\}$ with the associated contact mode weight matrix $\mathbf{W} = 0.1 \text{ diag}\{0, 3, 1, 1, 1, 0, 0\}$.

Figure 10 shows the prediction accuracy of the neural network trained on 117630 labelled data points on a validation set of 58815 labelled data points both generated using sampling the error state from a normal distribution with standard deviation $[0.03 \ 0.03 \ .4 \ 0.025]$. We evaluate the

TABLE II: Neural network parameters.

Property	Value
Number of hidden layers	3
Neurons in hidden layer 1	32
Neurons in hidden layer 2	50
Neurons in hidden layer 3	50
Activation functions	ReLU
Output layer	Softmax
Loss function	Cross entropy



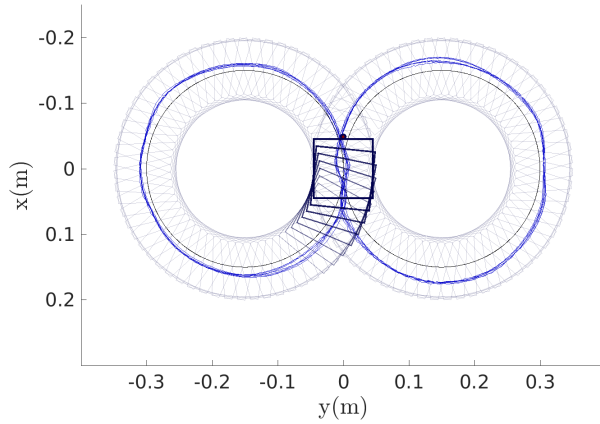
(a) Optimal contact mode (MIQP). (b) Classifier prediction.

Fig. 11: Optimal contact mode for the first time step as a function of initial state errors in x and y while holding $\theta = 5$ [deg] and $\phi = 0$ [deg]. The classifier model captures the important trends of the MPC (MIQP) optimal mode solutions.

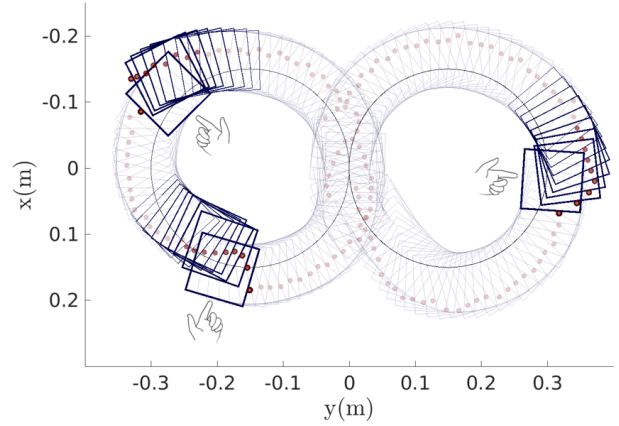
performance on each mode individually, as defined in Fig. 5.

Figure 11 compares the optimal contact mode associated with the first time step of the MPC (MIQP) with the prediction made by the classifier. The contact modes are generated as a function of initial state errors in x and y while holding θ and ϕ constant at 5 degrees. The regions shown in green, yellow, and blue denote the regions where the options actions are sliding left, sliding right, or sticking. From the figure, we notice that the optimal contact modes are separated into distinct region, thus justifying the search for contact mode as a classification problem and facilitating the learning process. We notice that the classifier model succeed in capturing the important trends of the MPC (MIQP) solutions.

Figure 12(a) depicts the robotic point pusher pushing the square object about a 8 track without any external perturbations for 7 consecutive laps. The black line represents the desired trajectory and the blue lines track the center of mass of the object. Although there is a small steady-state error, the controller succeeds in tracking the desired trajectory with accuracy. Figure 12(b) depicts the robotic point pusher pushing the square object about a 8 track with external perturbations for a single lap. The controller quickly succeeds in eliminating the perturbation and returning to the desired trajectory. The novel convex MPC formulation with learned modes achieves good closed-loop performance and permits a much higher bandwidth (250 Hz) than the non-convex MPC (MIQP) formulation (20 Hz).

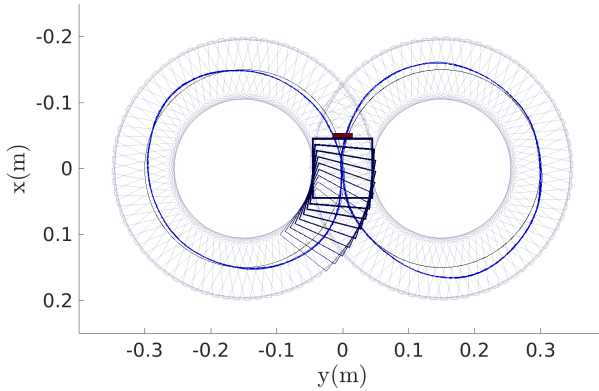


(a) Tracking of the 8 track for 7 consecutive laps. The black line represents the desired trajectory and the blue lines track the center of mass of the object.

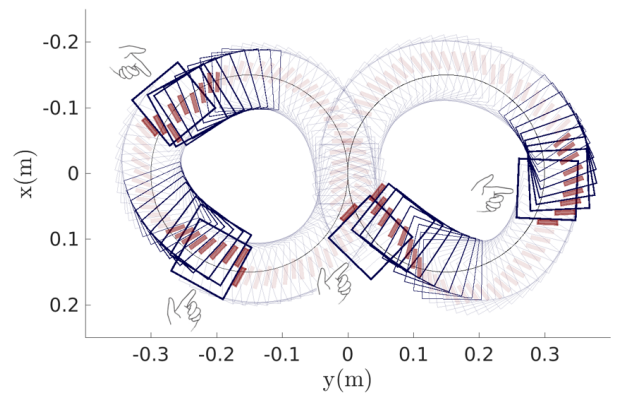


(b) Tracking of the 8 track with external perturbations for a single laps. The black line represents the desired trajectory and the hand represents the locations and directions in which the perturbations were applied.

Fig. 12: Point pusher. Closed-loop implementation of the MPC (learned modes) controller, where the goal is to push a square object about a 8 shaped trajectory.



(a) Tracking of the 8 track for 7 consecutive laps. The black line represents the desired trajectory and the blue lines track the center of mass of the object.



(b) Tracking of the 8 track with external perturbations for a single laps. The black line represents the desired trajectory and the hand represents the locations and directions in which the perturbations were applied.

Fig. 13: Line pusher. Closed-loop implementation of the MPC (learned modes) controller, where the goal is to push a square object about a 8 shaped trajectory.

B. Case Study B: Pushing with Line Contact



Fig. 14: Experimental setup for line pusher.

The experimental setup for the planar manipulation task with a line pusher is shown in Fig. 14. The trajectory tracking

task is shown in Fig. 13, where the goal is to track a 2D trajectory in the shape of an 8 shaped trajectory defined by two circles of radii 0.15 meters at a constant velocity of $v = 0.05$ [m/s]. We model the line pusher as 2 contact points that are constrained to move as a rigid-body, with the position of the center point of pusher denoted as p_c and state vector defined by $\mathbf{x} = [x \ y \ \theta \ p_c]^T$. Following a similar approach to that described in Section VII-A, we train a classifier model using 79410 labelled data points to predict the optimal mode schedule based on the error state of the system. The neural network properties used to parametrize the classifier model and the physical properties are related in Table 1 and 2, respectively. The controller design parameters are $h = 0.3$ seconds, $N = 35$, $\mathbf{Q} = 10 \text{ diag}\{1, 1, 1, 0.1\}$, $\mathbf{Q}_N = 2000 \text{ diag}\{1, 1, 1, 0.1\}$, and $\mathbf{R} = \text{diag}\{1, 1, 1, 1, 0.01\}$. We split the prediction horizon into 8 parts during which the contact modes are held constant. The number of time

steps associated with each contact mode section m_m is $\{1, 5, 5, 5, 5, 5, 5, 4\}$ with the associated weight matrix $\mathbf{W} = 0.1 \text{diag}\{0, 3, 1, 1, 1, 0, 0, 0\}$.

Figure 13(a) depicts the robotic line pusher pushing the square object about a 8 track without any external perturbations for 7 consecutive laps. The black line represents the desired trajectory and the blue lines track the center of mass of the object. The steady-state error is less prominent than in the point pusher case, as the line pusher is a more stable system with additional control authority. Figure 12(b) depicts the robotic point pusher pushing the square object about a 8 track with external perturbations for a single lap. Each time a perturbation is encountered, the pusher reacts to reduce the error by following a fast sliding motion to stabilize the object and then push it back towards the desired trajectory using a sticking phase. The novel MPC (learned modes) controller performs comparably to the MPC (MIQP) formulation and succeeds in tracking the desired trajectory while having significantly faster control bandwidth (200 Hz vs. 15 Hz).

VIII. CONCLUSION

This paper presents a methodology for feedback controller design of hybrid dynamical systems. The control formulation is based on a model predictive control approach, where the hybridness and underactuation associated with contact are explicitly enforced as constraints within a mixed-integer optimization program.

In order to enable real-time implementation, we address the combinatorial complexity resulting from the hybrid expansion of contact modes by separating the search for optimal mode schedules (offline) from the search for optimal control inputs (online). This is made possible by formulating the contact mode selection as a supervised learning problem. This approach enables us to train a classifier model offline by harnessing the solutions returned by the mixed-integer optimization program and building a dataset of optimal mode schedules.

We validate the controller design methodology on a planar manipulation experimental setup, where it is shown that the proposed convex formulation controller achieves comparable performance as its non-convex alternative, while obtaining a 10 fold improvement in the control bandwidth. Most importantly, in contrast to mixed-integer MPC control formulations, the online component of the hybrid MPC formulation with learned modes has the potential to extend to more complex dynamical systems with additional contact interactions, as its associated convex program can be efficiently computed in real-time.

REFERENCES

[1] Yu, K.T., Bauza, M., Fazeli, N., Rodriguez, A.: More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, October 9–14 (2016)

[2] Bauza, M., Rodriguez, A.: A probabilistic data-driven model for planar pushing. IEEE International Conference on Robotics and Automation (ICRA) arXiv:1704.03033 (2017)

[3] Fazeli, N., Donlon, E., Drumwright, E., Rodriguez, A.: Empirical evaluation of common contact models for planar impact. IEEE International Conference on Robotics and Automation (ICRA), Singapore, May 29 – June 3 (2017)

[4] Kolbert, R., Chavan-Dafle, N., Rodriguez, A.: Experimental validation of contact dynamics for in-hand manipulation. International Symposium on Experimental Robotics (ISER). Spring Proceedings in Advanced Robotics. **1** (2016)

[5] Hogan, F.R., Rodriguez, A.: Feedback control of the pusher-slider system: a story of hybrid and underactuated contact dynamics. In Proceedings of the 12th International Workshop on the Algorithmic Foundations of Robotics (WAFR), San Francisco, CA, USA, December 18 – 20. (2016)

[6] Mason, M.T.: Mechanics and planning of manipulator pushing operations. The International Journal of Robotics Research **5**(3) (1996) 53 – 71

[7] Goyal, S., Ruina, A., Papadopoulos, J.: Wear. Planar sliding with dry friction Part I. Limit surface and moment function **143** (1991) 307 – 330

[8] Lynch, K.M., Mason, M.T.: Stable pushing: mechanics, controllability, and planning. The International Journal of Robotics Research **15**(6) (1986) 533 – 556

[9] Yu, K.T., Leonard, J., Rodriguez, A.: Shape and pose recovery from planar pushing. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, September 28 – October 02 (2016)

[10] Lynch, K., Maekawa, H., Tanie, K.: Manipulation and active sensing by pushing using tactile feedback. IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, USA, July 7 – 10, pp. 416–421 (1992)

[11] Woodruff, J.Z., Lynch, K.M.: Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks. IEEE International Conference on Robotics and Automation (ICRA), Singapore, May 29 – June 3 (2017)

[12] Lee, S.H., Cutkosky, M.: Journal of Manufacturing Science and Engineering. Fixture planning with friction **113**(3) (1991) 320 – 327

[13] Zhou, J., Paolini, R., Bagnell, J.A., Mason, M.T.: A convex polynomial force-motion model for planar sliding: Identification and application. IEEE International Conference on Robotics and Automation (ICRA), May 16–21, Stockholm, Sweden, May 16–21 (2016)

[14] Chavan-Dafle, N., Rodriguez, A.: Sampling-based planning for in-hand manipulations with external pushes. International Symposium on Robotics Research (ISRR). Under Review. (2017)

[15] Posa, M., Cantu, C., Tedrake, R.: A direct method for trajectory optimization of rigid bodies through contact. The International Journal of Robotics Research **33**(1) (2014) 69 – 81

[16] Tassa, Y., Todorov, E.: Synthesis and stabilization of complex behaviors through online trajectory optimization. IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, October 7 – 12 (2012)

[17] Pajarinen, J., Kyrki, V., Koval, M., Srinivasa, S., Peters, J., Neumann, G.: Hybrid control trajectory optimization under uncertainty. arXiv:1702.04396 (2017)

[18] Ryu, J., Ruggiero, F., Lynch, K.M.: Control of nonprehensile rolling manipulation: Balancing a disk on a disk. IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, May 14–18, pp. 3232–3237 (2012)

[19] Posa, M., Kuindersma, S., Tedrake, R.: Optimization and stabilization of trajectories for constrained dynamical systems. In Proceedings of the International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, May 16–21 (2016)

[20] Zhou, J., Bagnell, J., Mason, M.: A fast stochastic contact model for planar pushing and grasping: theory and experimental validation. Robotics Science and Systems, Cambridge, MA, USA, July 12–16 (2017)

[21] Mason, M.T.: Mechanics of robotic manipulation. MIT Press, Cambridge, Massachusetts (2001)