Discrete-time Control Contraction Metrics (DCCM) for Quasistatic Planar Pushing using Smoothed Dynamics

MIT 2.152[J] Nonlinear Control Project Report

Shao Yuan Chew Chia *Harvard University*
Cambridge, MA
shaoyuan_chewchia@college.harvard.edu

Abstract—The paper explores the feasibility and performance of Discrete-time Control Contraction Metrics (DCCM) on a planar pushing system with smoothed dynamics, to address the challenges of non-smoothness and underactuation of control through contact. While they were unable to achieve many of the theoretical advantages of the control contraction metric framework, they managed to demonstrate the feasibility of synthesizing and running a DCCM controller on a planar pushing system with highly smoothed dynamics. The limitations and gaps in applying their implementation in the real world are also discussed. The paper serves as a good starting point for future work to realize the full potential of DCCMs applied to systems with contact.

I. INTRODUCTION

Planning and control through contact is an important capability in robotics. In order for robots to manipulate objects in their environment, they need to be able to reason about how to make and break contact. However, this has proved challenging for model-based methods due to two factors:

1) **Non-smooth or hybrid nature of contact dynamics**. Contact dynamics involve different modes of contact (e.g. sticking, sliding, no-contact), each with smooth but different dynamics and constraints. When switching between modes, the dynamics are discontinuous and change abruptly. This poses problems for gradient-based methods such as optimal control, as their locally linear models constructed using gradients quickly become inaccurate [1].
2) **Underactuatedness of the system**. Systems that involve contact are underactuated as the state of the unactuated objects cannot be controlled directly. Instead, control inputs to the robot are mediated through friction cones at contact points between the robot and the object, thus severely limiting the control authority of the inputs.

Current methods for model based planning and control through contact typically fall into one of three categories: methods that apply smoothing to the dynamics [1], [2], methods that explicitly enumerate the contact modes [3], and methods that implicitly reason about the contact modes [4]–[6]. Two methods that have been applied to the control of nonlinear underactuated systems are Linear Quadratic Regulator (LQR)-Trees and Control Contraction Metrics (CCM). LQR-Trees

use locally optimal linear control policies to stabilize planned trajectories. Each trajectory has a basin of attraction and these regions are pieced together to cover a larger region of state space [7]. CCMs, first established in [8], were shown to be effective at controlling underactuated systems in [9].

CCMs theoretically provide 4 key advantages over other control methods as described in [8]–[11]:

1) **Certificates of stability and convergence rates**. In general, nonlinear MPC does not provide guarantees on stability or performance. While Control Lyapunov Functions (CLFs) and LQR based methods can produce certificates, the guarantees provided by CCMs are not tied to specific equilbria or trajectories.
2) **Trajectory independent controllers**. CCMs generate controllers that stabilize every feasible trajectory in a region, while CLFs and LQR based methods require designing controllers for each trajectory and stitching them together to cover larger regions.
3) **Convex synthesis of the controller**. The synthesis of CCMs can be formulated as a convex optimization problem, which is easier to solve numerically and can leverage powerful tools like SOS programming.
4) **Faster online computation of the control law**. The online computation of the geodesic in CCMs is typically faster and simpler than solving the nonlinear optimal control problem as is done in MPC.

In this paper we use the analytic smoothing of contact dynamics described in [1] to address the challenge of non-smoothness, and explore the effectiveness of building a Discrete-time Control Contraction Metric (DCCM) as demonstrated by [11] on top of that smoothed system to handle the underactuatedness and nonlinearity of the system. Our hope was that we would be able to synthesize a DCCM for a system with low amounts of smoothing and that this controller would also stabilize a "real-world" system with exact dynamics, however we did not manage to achieve that goal. We only managed to synthesize a controller that worked around a particular desired trajectory under significant smoothing, but are hopeful that our findings will serve as a good jumping off points for future work to realize the full potential of DCCMs applied to systems with contact.

II. PRELIMINARIES AND PROBLEM FORMULATION

A. Quasistatic Assumption

We will assume that our system is quasistatic, meaning the velocities and accelerations of the system are 0 at the end of each time step [1]. This corresponds to having a high amount of damping and is a reasonable assumption in the 2D planar pushing setup where we restrict pushing velocities to be low and there is a large amount of friction between the object and table surface. As a result, the state of our system only consists of positions.

B. Analytically Smoothed Contact Dynamics

In this work we use the analytically smoothed contact dynamics and corresponding simulator developed by the authors of [1]. Contact dynamics are formulated as an unconstrained convex program where the contact and friction constraints are moved into the objective function using a log barrier function. The effect of this is that there is a log barrier penalty for violating the contact constraints. Constraints can exert force even if they are not active and this translates to producing a force at a distance.

We plot the force at a distance effect of the smoothed contact dynamics in figure (1). We see that for a high weight, which corresponds to a small force at a distance, the next b_x is close to 0, but as the log barrier weight decreases, the box is pushed further to the right for the same position of the spherical robot.
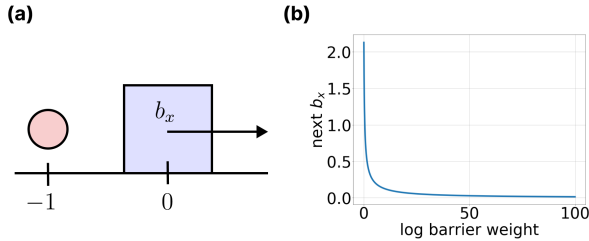

Fig. 1: Force at a distance effect of smoothed contact dynamics. (**a**) A system consisting of an actuated spherical robot at $x = -1$ and unactuated box at $x = 0$. (**b**) The next b_x after rolling out one step of the analytically smoothed dynamics with different log barrier weights.

C. Planar Pushing System and its Contraction Condition

The state of system is represented as:

$$x = \begin{bmatrix} b_x & b_y & b_\theta & s_x & s_y \end{bmatrix}^\top \qquad (1)$$

Control inputs u are absolute position commands for the sphere

$$u = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \qquad (2)$$

The system evolves nonlinearly and is control affine. The discrete-time dynamics are defined as

$$x_{k+1} = f(x_k) + g(x_k)u_k \qquad (3)$$

where f and g are smooth functions due to the smoothing of the contact dynamics described in section (II-B).
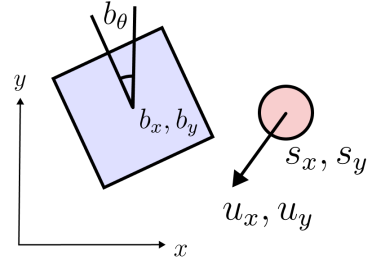

Fig. 2: The components of state x and position command u

The differential dynamics are defined as

$$\delta_{x_{k+1}} = A(x_k)\delta_{x_k} + B(x_k)\delta_{u_k} \qquad (4)$$

where $A(x_k) = \frac{\partial(f(x_k)+g(x_k)u_k)}{\partial x_k} \in \mathbb{R}^{5\times5}$ and $B(x_k) = \frac{\partial(f(x_k)+g(x_k)u_k)}{\partial u_k} \in \mathbb{R}^{5\times2}$ are the Jacobians of the dynamics.

We can define the differential state feedback control law

$$\delta_{u_k} = K(x_k)\delta_{x_k} \qquad (5)$$

where K is the state dependent feedback gain matrix.

We denote the generalized infinitesimal squared distance in the positive definite metric M as

$$V_k = \delta_{x_k}^\top M_k \delta_{x_k} \qquad (6)$$

By substituting the differential dynamics (4) and control law (5) into (6), we can see that the generalized infinitesimal squared distance at the next time step is

$$\begin{aligned} V_{k+1} &= \delta_{x_{k+1}}^\top M_{k+1}\delta_{x_{k+1}} \\ &= \delta_{x_k}^\top (A_k + B_k K_k)^\top M_{k+1}(A_k + B_k K_k)\delta_{x_k} \end{aligned} \qquad (7)$$

The contraction condition can then be expressed as

$$V_{k+1} - V_k \le -\beta V_k < 0 \quad \forall x, \delta_x \in \mathbb{R}^5, \beta \in (0,1] \qquad (8)$$

where the contraction rate is $(1 - \beta)$.

Equation (8) is equivalent to

$$(A_k + B_k K_k)^\top M_{k+1}(A_k + B_k K_k) - (1-\beta)M_k < 0 \qquad (9)$$

If we are able to find a metric M and control gain K that satisfy the contraction condition (9), we can find a stabilizing controller that is able to track arbitrary feasible trajectories. To run the controller online, at each time step we will need to find the minimum length (w.r.t the metric M) curve between our desired state x_k^* and our actual state x_k.

For a smooth curve $c(s), s \in [0,1]$ that connects two points in state space x_0 and x_1, [8] defines the Riemannian length and energy of the curve as

$$L(c) = \int_0^1 \sqrt{\frac{\partial c(s)}{\partial s}^\top M(c(s))\frac{\partial c(s)}{\partial s}}\,ds$$

$$E(c) = \int_0^1 L(c)^2 ds \qquad (10)$$

The geodesic $\gamma(x_0, x_1)$ is the curve that minimizes the Riemannian length and energy between x_0 and x_1

$$\begin{aligned} \gamma(x_0, x_1) &= \underset{c}{\arg\min}\, L(c) \\ &= \underset{c}{\arg\min}\, E(c) \end{aligned} \qquad (11)$$

By the contraction condition we enforced, we see that the Riemannian energy of geodesic decreases exponentially as the system evolves and thus can be thought of as an incremental Lyapunov function [8].

The controller that tracks the desired trajectory (x^*, u^*) will then be

$$u_k = u_k^* + \int_0^1 K(\gamma(s)) \frac{\partial \gamma(s)}{\partial s} \qquad (12)$$

III. METHODS

A. Contraction Metric and Controller Synthesis

1) Sum of Squares (SOS) Programming: [11] showed that equation (9) can be transformed via Schur's complement as well as left and right multiplying by an invertible matrix into

$$\begin{bmatrix} W_{k+1} & A_k + B_k L_k \\ (A_k + B_k L_k)^\top & (1-\beta)W_k \end{bmatrix} > 0 \qquad (13)$$

where $W := M^{-1}$ and $L := KW$.

In order to synthesize the contraction metric and controller, we use the SOS programming formulation described in [11] with some slight modifications

$$\min_{l_c, w_c, r} \quad r$$
$$s.t. \quad \forall k, w^\top \Omega w - r w^\top w \in \Sigma(x_k, u_k, w) \qquad (14)$$
$$r \geq 0.1$$

where $\Sigma(x_k, u_k, w)$ is the set of SOS polynomials that satisfy the contraction condition in equation (13). l_c are the polynomial coefficients of L and w_c are the polynomial coefficients of W.

$$\Omega = \begin{bmatrix} W_{k+1} & A_k + B_k L_k \\ (A_k + B_k L_k)^\top & (1-\beta)W_k \end{bmatrix}$$
$$W_k = \begin{bmatrix} W_{11_k} & \cdots & W_{15_k} \\ \vdots & \ddots & \vdots \\ W_{15_k} & \cdots & W_{55_k} \end{bmatrix} \qquad (15)$$
$$L_k = \begin{bmatrix} L_{11_k} & \cdots & L_{15_k} \\ L_{21_k} & \cdots & L_{25_k} \end{bmatrix}$$

where each $W_{..k} = w_{..c} v(x_k)$ is a polynomial constructed from the row vector of coefficients of $w_{..c}$ and the monomial basis vector $v(x_k)$. For example, if the degree of the polynomial is chosen to be 4,

$$v(x_k) = [x_{k_4}^4, x_{k_3} x_{k_4}^3, x_{k_3}^2 x_{k_4}^2, \cdots, x_{k_1}, x_{k_0}, 1] \qquad (16)$$

where $v(x_k)$ has 126 elements. $L_{..k} = l_{..c} v(x_k)$ is similarly defined.

In general, the dimensions of the matrices are as follows: $\Omega : 2 \cdot dim(x) \times 2 \cdot dim(x), W : dim(x) \times dim(x), L : dim(u) \times dim(x)$

We note the difference in the way we use the slack variable r compared to [11]. In [11], the constraints on the optimization program are $w^\top \Omega w - rI \in \Sigma(x_k, u_k, w), r \geq 0$. In practice, we found that in some cases, especially when generating a higher degree metric, that the solver would return a trivial solution where r, w_c and l_c are extremely small numbers (on the order of $1e-17$). By setting a higher bound on r, we force the solver to find a solution where the contraction condition is satisfied with a greater buffer and the returned coefficients of w_c and l_c are larger.

Another difference is that we do not have closed form equations for the A and B matrices which would allow us to enforce the contraction conditions over all states. Instead, we sample a set of (state, control action) pairs and enforce the contraction condition over these samples. We use the contact dynamics solver described in [1] to calculate the next state x_{k+1}, and the Jacobians A_k and B_k, for each sample of the current state x_k and control action u_k, and substitute these values into the constraints in optimization program (14). Since M is smooth, we can expect the contraction condition to be satisfied over at least a small local region around each sample. However, if the samples are too sparse, the contraction condition may not be satisfied over all the states around the desired trajectory.

We solve this SOS program using Drake's Mathematical Program [12], which uses MOSEK under the hood to solve this Semidefinite Program (SDP).

2) Sampling Strategy: To get a contraction metric valid over the entire state space we would have to densely sample the entire state space. However, the available RAM on the machine used set an upper bound of the size of the optimization program we were able to run. For a monomial basis of degree 4, the maximum number of samples we could add to our constraints was around 2000. Thus, to get a contraction metric and controller that had good performance at least in the vicinity of the desired trajectory, we only sampled states and control actions from a small region around the desired trajectory as shown in figure (3). This is a clear limitation of the current approach and future work would involve finding a way to enforce the contraction condition over a larger portion of state space.
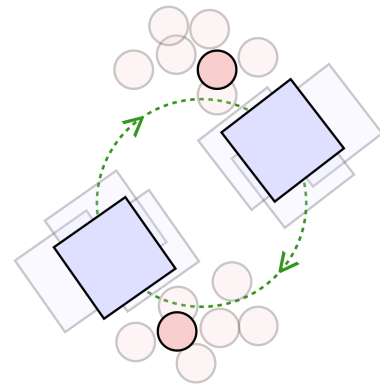

Fig. 3: Sampled box and sphere positions around the circular target trajectory

B. Online Geodesic and Controller Computation

With the contraction metric synthesized, we now need to compute the control action that enforces the contraction condition.

In order to numerically approximate the geodesic $\gamma(x_k^*, x_k)$, we discretize the curve into N segments and solve the following optimization program

$$\bar{\gamma}(x_k^*, x_k) = \operatorname*{argmin}_{\substack{x[\cdot],\Delta x_s[\cdot],\\ \Delta s[\cdot],m[\cdot],y[\cdot]}} \sum_{i=0}^{N-1} y[i] + \Delta s[i]^2$$

$$s.t. \quad \forall i, y \geq \Delta s[i]\Delta x_s[i]^\top M(m[i])\Delta x_s[i]$$
$$x[0] = x_k^*, \quad x[N] = x_k$$
$$\forall i, x[i+1] = x[i] + \Delta x_s[i]\Delta s[i]$$
$$\forall i, \Delta s[i] > 0, \quad \sum_{i=0}^{N-1} \Delta s[i] = 1$$
$$\forall i, M(m[i])W(x[i]) = I$$
$$(17)$$

where $x[i]$ is the state at the start of the ith segment of the geodesic, $\Delta s[i]$ is a small positive scalar, $\Delta x_s[i]$ is the discretized displacement vector, $y[i]$ is a slack variable that represents the Riemannian energy and N is the number of segments the geodesic γ is discretized into. $m[i]$ is a slack variable introduced such that the 5×5 symbolic matrix $W(x[i])$ does not need to be explicitly inverted which was found to be a severe computational bottleneck. The constraint $M(m[i])W(x[i]) = I$ is a surrogate for enforcing $M(m[i]) = W(x[i])^{-1}$. Adding the regularization term $\Delta s[i]^2$ to the objective serves to spread out the discretized points evenly along the geodesic.

To solve this non-convex optimization problem we use Drake's Mathematical Program, this time using SNOPT under the hood [12].

With the geodesic $\bar{\gamma}(x_k^*, x_k)$ computed, we can compute the control action u_k that enforces the contraction condition (9)

$$u_k = u_k^* + \sum_{i=0}^{N-1} \Delta s[i]K(x[i])\Delta x_s[i]$$
$$= u_k^* + \sum_{i=0}^{N-1} \Delta s[i]L(x[i])W^{-1}(x[i])\Delta x_s[i]$$
$$(18)$$

An important point to note is that the integration is done from x_k^* to x_k and not the other way around. While it might seem intuitive that we want to calculate the δ_u that brings the system from x_k to x_k^*, this is actually not the right way to think about it. First, the δ_u in equation 5 does not lead to a change in state δ_x on the right side of the same equation. Instead, 5 tells us for a change in state δ_x from a nominal trajectory, what is the corresponding δ_u that enforces the contraction condition. In this case, the nominal trajectory is x^* and u^*, thus to calculate the δ_u that enforces the contraction condition, we need to integrate from x_k^* to x_k.

IV. EXPERIMENTAL SETUP

A. Creating Feasible Desired Trajectories

To create the desired trajectory, we first parameterized an arbitrary circular trajectory with the robot behind the object at each time step and the object slowly rotating as it travels around the circle. We then rolled out the dynamics from the initial condition, using the position of the robot at the next time step as the open loop position command for the robot

and recorded the actual positions of the object under this open loop control. We created two feasible circular trajectories of slightly different radii in this way and spliced them together in the middle $(t = 5)$ in order to introduce a step change in the desired trajectory. Finally, we introduce an additional initial disturbance by adding an offset to the initial state of the system from the first state of the desired trajectory. We do this process twice to create desired trajectories for both the log barrier weight 10 and 100 levels of smoothing.

V. RESULTS AND DISCUSSION

Our original goal was to synthesize a DCCM with a small amount of smoothing that would transfer well to stabilizing the real system to arbitrary feasible desired trajectories under non-smooth, exact contact dynamics. Unfortunately, we were not able to achieve this goal. We found that when using a small amount of smoothing, a higher degree monomial basis was required to enforce the contraction condition across all samples. Furthermore, since the dynamics were less smooth, a greater density of samples was required in order for the contraction condition to hold around the desired trajectory. This can be seen from the result in table (I) where even though we were able to synthesize DCCMs for 500 samples for both log barrier weights, the DCCM for log barrier weight 10 was able to stabilize the system while the DCCM for log barrier weight 100 was not.

Both of these factors (requiring a higher degree, and more samples) led to the optimization (14) becoming intractable due to the computer running out of memory. Ultimately, we were only able to find a stabilizing controller for a log barrier weight of 10, which corresponds to the robot exerting 0.1N of force on an object that is 1m away.

Log Barrier Weight	Deg.	# Samples	100	500	1000	2000	3000
10	4	Synthesizes	✓	✓	✓	✓	!
		Stabilizes	✗	✓	✓	✓	-
	6	Synthesizes	✓	✓	!	-	-
		Stabilizes	✗	✗	-	-	-
100	4	Synthesizes	✓	✓	✗	-	-
		Stabilizes	✗	✗	-	-	-
	6	Synthesizes	✓	✓	!	-	-
		Stabilizes	✗	✗	-	-	-

TABLE I: Feasibility (whether a DCCM can be synthesized) and performance (whether the found DCCM stabilizes to the desired trajectory) across different log barrier weights (10 is high smoothing, 100 is low smoothing), degree of monomial basis used, and number of sampled points at which the contraction condition is enforced. Symbols: ✓(succeeds), ✗(fails), ! (program crashes), - (did not/could not run test).

A. Controller Performance

Figure (4) demonstrates the performance of the controller synthesized and run with parameters listed in table (II). We

see that the controller is able to stabilize the system from the initial offset at $t = 0$ and step change in desired trajectory at $t = 5$, with the geodesic energy $E(\bar{\gamma})$ decreasing exponentially after each instantaneous reference change.

Log Barrier Weight	Deg.	# Samples	# Geodesic Segments	β
10	4	2000	1	0.1

TABLE II: Parameters of the controller shown in figure (4), where # Geodesic Segments is the parameter N in optimization problem 17, and $\beta = (1-\text{convergence rate w.r.t. DCCM})$ as shown in 15.
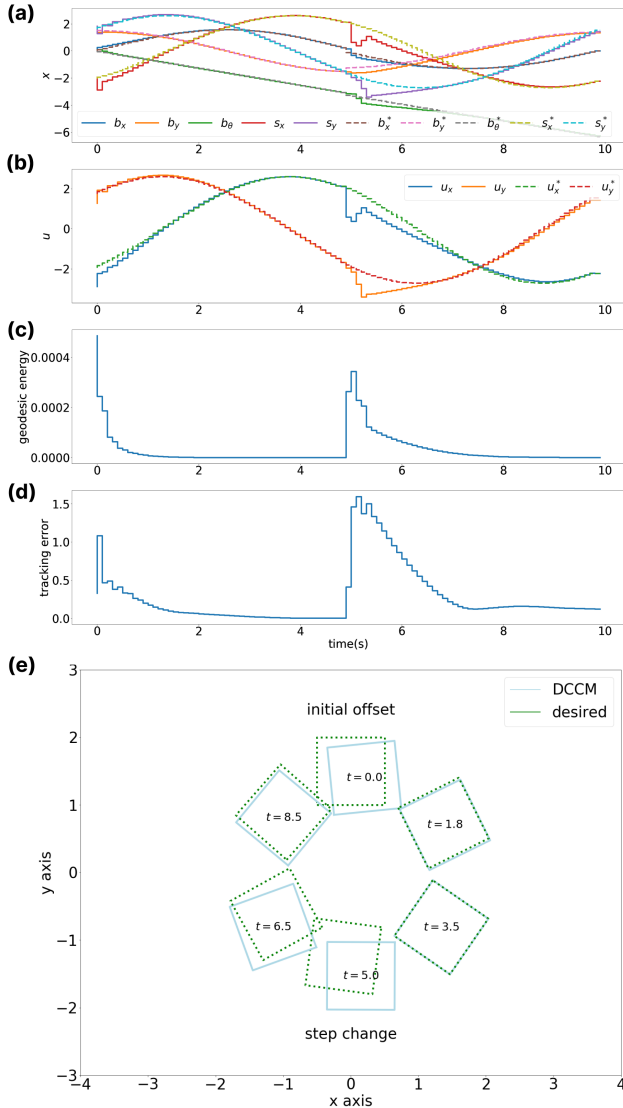

Fig. 4: Performance of DCCM with parameters (II); **(a)** desired state x^* and actual state x; **(b)** nominal input u^* and DCCM input u; **(c)** geodesic energy $E(\bar{\gamma})$, (10), (17); **(d)** l_2-norm tracking error; **(c)** desired and actual positions of the box at various time steps.

B. Effect of number of samples on controller performance

Figure (5) shows the performance of 3 controllers, synthesized with 500, 1000, and 2000 samples respectively. Note that the controller synthesized with 100 samples was not included in the figure as the controller was not able to stabilize the system. We can see that as the number of samples increases, the controller is better able to regulate the geodesic energy. For the controller with 500 samples, geodesic energy does not exponentially decrease, in fact, it increases at times. We reason that this is due to the contraction condition not being enforced over the states of the trajectory due to the sparse sampling.

It is also interesting to note that even though the 2000 sample controller consistently has the lowest geodesic energy, and fastest decrease in geodesic energy after instantaneous reference change, it also has the largest l2-norm tracking error. In order to stabilize the system and make the geodesic energy decrease exponentially, the sphere robot has to deviate a greater amount from the reference trajectory in order to push the box back towards the reference trajectory, leading to the spike in error.

Log Barrier Weight	Deg.	# Samples	# Geodesic Segments	β
10	4	500, 1000, 2000	1	0.1

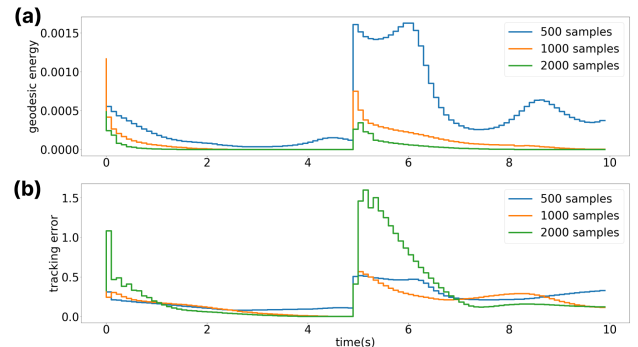TABLE III: Parameters of the controllers shown in figure 5.

Fig. 5: Comparison of performance of controllers synthesized with varying numbers of samples. Parameters listed in table (III); **(a)** geodesic energy $E(\bar{\gamma})$, (10), (17); **(b)** l_2-norm tracking error.

C. Discretization of Geodesic

In testing, we found that trying to find the geodesic for the planar pushing system with $N > 1$ (17) consistently failed. This was slightly unexpected as even though there are more decision variables for larger N, and while acknowledging that this is a non-convex problem, it was interesting that SNOPT was not able to find even the straight line solution. We were able to get larger Ns in our implementation of the simpler 2-dimensional CSTR dynamical system from [11], but did notice that it significantly increased the online computation time.

For future work, we hope to explore warm-starting the larger N geodesic optimization with the solution of the $N = 1$ geodesic, and then warm starting subsequent geodesic

optimizations with the previous time-step's solution. Given smoothness of the system and the metric, the previous solution should serve as a good initial guess.

D. Effect of Convergence Rates on DCCM Synthesis

Another peculiar observation of our implementation was that synthesis of the DCCM consistently failed when using $\beta > 0.3$. We think that it would be strictly easier to satisfy the slower convergence rate enforced by a larger β, however that is not what we found. Furthermore, while we were able to synthesize a DCCM with 500 samples for both $\beta = 0.1$ and $\beta = 0.3$, the $\beta = 0.1$ controller was able to stabilize the system while the $\beta = 0.3$ controller was not. This may speak to the brittleness of the controller when using insufficient samples, or it may be indicative of some underlying issue that is also contributing to the synthesis of controllers with larger β failing. More investigation into this issue is required to fully understand it.

E. Computation Time

The DCCM synthesis with a degree 4 monomial basis and 500 samples took approximately 18 minutes, while using a degree 6 basis with the same number of samples took approximately 3 hours 20 minutes. The synthesis of the DCCM with degree 4 basis and 2000 samples shown in figure 4 took approximately 1 hour.

Online computation of the 1-segment geodesic for a DCCM synthesized with degree 4 basis took 1.54 seconds, while the DCCM synthesized with degree 6 basis took 3.81 seconds.

All experiments were run on a Linux machine with 31.3 GB of RAM, with an Intel Core i7-6700 CPU @ 3.40GHz x 8 processor.

VI. CONCLUSION AND FUTURE WORK

The feasibility of synthesizing and running a DCCM controller on a planar pushing system with smoothed dynamics was demonstrated in this paper. Limitations of this preliminary implementation were also discussed. Unfortunately, in this work we were largely unable to achieve the benefits discussed in section (I):

1) **Certificates of stability and convergence rates**. Two major gaps remain to be overcome in order to certify the stability and performance of a DCCM controller on a real system. The first is that theoretical arguments need to be made about density of samples on which the contraction condition is enforced and a connection drawn to the smoothness of the system in order to bound the error between the closed loop dynamics of the sample-based DCCM, and the smoothed dynamics. Second, the error between the smooth dynamics and exact dynamics also needs to be bounded. Only after mathematically accounting for these two sources of "sim to real" gap will we be able to certify the controller on the real system.

2) **Trajectory independent controllers**. We were not able to realize this benefit in our implementation due to not having closed form equations of the A and B matrices of our system and computational limitations on the number of samples we could add to our synthesis SOS program. This resulted in low coverage of the state space and meant that there was only a narrow basin of attraction around our desired trajectory.

3) **Convex synthesis of the controller**. While SOS programming was able to synthesize DCCMs that worked, we ran into challenges with the high degree of monomial basis required. Future work should address this issue, possibly by applying more computational power to the system, or exploring other synthesis methods such as learning the contraction metric [13], [14]

4) **Faster online computation of the control law**. In this paper we used a basic implementation of the geodesic calculation. We avoided working with the symbolic inversion of the W matrix in (17) but that made the burden of finding the inverse shift to the optimizer. In future work, more advanced methods should be explored, such as the warm starting schemes described in section (V-C) or the pseudospectral approach described in [15].

Another interesting direction we are excited to explore is reformulating the problem such that we only care about stabilizing the state of the object to the desired trajectory and do not require the trajectories of the robot to be contracting. Stabilization of submanifolds is briefly discussed in [8], but more work need to be done to translate it to the discrete-time case. This formulation may be more true to our goals of manipulating the object and have the benefit of reducing the dimensionality of the problem which might make both offline and online calculations easier.

While more work needs to be done to realize the full benefits of applying the control contraction metric framework to the problem of control through contact, we are hopeful that this paper serves as a good first step on that path.

REFERENCES

[1] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake. "Global Planning for Contact-Rich Manipulation via Local Smoothing of Quasi-dynamic Contact Models." Comment: The first two authors contributed equally to this work. arXiv: 2206.10787 [cs]. (Feb. 27, 2023), [Online]. Available: http://arxiv.org/abs/2206.10787 (visited on 03/15/2023), preprint.

[2] D. E. Stewart and M. Anitescu, "Optimal control of systems with discontinuous differential equations," *Numerische Mathematik*, vol. 114, no. 4, pp. 653–695, Feb. 1, 2010, ISSN: 0945-3245. DOI: 10.1007/s00211-009-0262-2. [Online]. Available: https://doi.org/10.1007/s00211-009-0262-2 (visited on 05/12/2023).

[3] F. R. Hogan and A. Rodriguez. "Feedback Control of the Pusher-Slider System: A Story of Hybrid and Underactuated Contact Dynamics." arXiv: 1611.08268 [cs]. (Nov. 24, 2016), [Online]. Available: http://arxiv.org/abs/1611.08268 (visited on 05/12/2023), preprint.

[4] J.-P. Sleiman, J. Carius, R. Grandia, M. Wermelinger, and M. Hutter, "Contact-Implicit Trajectory Optimization for Dynamic Object Manipulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Comment: 8 Pages, 10 Figures. Submitted to the International Conference on Intelligent Robots and Systems (IROS), 2019, Nov. 2019, pp. 6814–6821. DOI: 10.1109/IROS40897.2019.8968194. arXiv: 2103.01104 [cs]. [Online]. Available: http://arxiv.org/abs/2103.01104 (visited on 03/17/2023).

[5] K. Nakatsuru, W. Wan, and K. Harada. "Implicit Contact-Rich Manipulation Planning for a Manipulator with Insufficient Payload." arXiv: 2302.13212 [cs]. (Feb. 25, 2023), [Online]. Available: http://arxiv.org/abs/2302.13212 (visited on 03/17/2023), preprint.

[6] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, Jan. 1, 2014, ISSN: 0278-3649. DOI: 10.1177/0278364913506757. [Online]. Available: https://doi.org/10.1177/0278364913506757 (visited on 03/17/2023).

[7] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, Jul. 1, 2010, ISSN: 0278-3649. DOI: 10.1177/0278364910369189. [Online]. Available: https://doi.org/10.1177/0278364910369189 (visited on 05/13/2023).

[8] I. R. Manchester and J.-J. E. Slotine, "Control Contraction Metrics: Convex and Intrinsic Criteria for Nonlinear Feedback Design," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017, ISSN: 1558-2523. DOI: 10.1109/TAC.2017.2668380.

[9] I. R. Manchester, J. Z. Tang, and J.-J. E. Slotine, "Unifying Robot Trajectory Tracking with Control Contraction Metrics," in *Robotics Research: Volume 2*, ser. Springer Proceedings in Advanced Robotics, A. Bicchi and W. Burgard, Eds., Cham: Springer International Publishing, 2018, pp. 403–418, ISBN: 978-3-319-60916-4. DOI: 10.1007/978-3-319-60916-4_23. [Online]. Available: https://doi.org/10.1007/978-3-319-60916-4_23 (visited on 05/05/2023).

[10] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust Online Motion Planning via Contraction Theory and Convex Optimization," May 29, 2017. DOI: 10.1109/ICRA.2017.7989693.

[11] L. Wei, R. Mccloy, and J. Bao, "Control Contraction Metric Synthesis for Discrete-time Nonlinear Systems," *IFAC-PapersOnLine*, 16th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2021, vol. 54, no. 3, pp. 661–666, Jan. 1, 2021, ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2021.08.317. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896321010910 (visited on 04/16/2023).

[12] R. T. a. the Drake Development Team. "Drake: Model-Based Design and Verification for Robotics." (2019), [Online]. Available: https://drake.mit.edu/ (visited on 05/10/2023).

[13] S. Singh, V. Sindhwani, J.-J. E. Slotine, and M. Pavone. "Learning Stabilizable Dynamical Systems via Control Contraction Metrics," arXiv.org. (Jul. 31, 2018), [Online]. Available: https://arxiv.org/abs/1808.00113v2 (visited on 05/04/2023).

[14] G. Chou, N. Ozay, and D. Berenson, "Model Error Propagation via Learned Contraction Metrics for Safe Feedback Motion Planning of Unknown Systems," Dec. 14, 2021, pp. 3576–3583. DOI: 10.1109/CDC45484.2021.9683354.

[15] K. Leung and I. R. Manchester. "Nonlinear Stabilization via Control Contraction Metrics: A Pseudospectral Approach for Computing Geodesics." arXiv: 1607.04340 [cs]. (Nov. 4, 2017), [Online]. Available: http://arxiv.org/abs/1607.04340 (visited on 05/06/2023), preprint.