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*—**

I. Introduction

Planning and control through contact is an important task in Robotics. Robots interacting with the environment need to make and break contact in order to complete tasks but this has proved challenging due to the non-smooth nature of contact dynamics.

Another way in which control through contact is difficult is that these systems are highly underactuated. The state of the unactuated objects cannot be controlled directly.

Two main approaches, smoothing [1], [2], and hybrid systems [3] have been proposed to deal with the non-smoothness of contact dynamics. Smoothing methods approximate the non-smooth contact dynamics with a smooth function. Hybrid systems explicitly enumeate the contact modes.

Current methods for control through contact include hybrid planners and controllers with guards and resets where different dynamics are considered based on enumerated contact modes. An alternative approach has been proposed in [2] where contact dynamics are smoothed to create more informative gradients in global planning of contact-rich manipulation tasks. In this paper we apply these same smoothing techniques to explore the controllers that this enables.

convex synthesis of contraction metric and controller, which we have powerful tools for like SOS programming.

The same controller exponentially stabilizes to arbitrary, time-varying feasible trajectories, while Lyapnov-based approaches typically need to be designed for a specific equilibrium.

provide guarantees on stability and convergence rates

This work builds on [4] which demonstrates the effectiveness of CCMs on cannonical underactuated systems such as Cart-Pole.

as well as [5] that extends work in [6] to discrete time.

II. Preliminaries and Problem Formulation

A. Quasistatic Assumptions

We will assume that our system is quasistatic, meaning at each time step velocities and accelerations of the system are 0. 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 [2]. Contact dynamics are formulated as an unconstrained convex program where the contact and friction contraints 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.

C. Planar Pushing System

The state of

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

control input u are absolute position commands for the sphere

$$u = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{2}$$

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

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

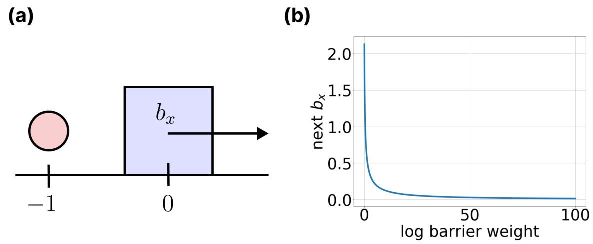

Fig. 1: Force at a distance effect of smoothed contact dynamics. (**a**) A system consisting of an actuated point finger at $x = -1$ and unactuated box at $x = 0$, (**b**) The next b_x after rolling out one step of the analytically smoothed dynamics with different log barrier weights.

where f and g are smooth functions due to the smoothing of the contact dynamics described in the previous section.

The differential dynamics are defined as

$$\delta_{x_k} = A(x_k)\delta_{x_k} + B(x_k)\delta_{u_k} \tag{4}$$

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 state feedback control law

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

where K is the state dependent feedback gain matrix.

Generalized infinitesimal squared distance in the positive definite metric M is denoted V_k

$$V_k = \delta_{x_k}^\top M_k \delta_{x_k} \tag{6}$$

And by substituting the differential dynamics and control law, we can see that the generalized infinitesimal squared distance at the next time step is

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

The contraction condition can then be expressed as

$$V_{k+1} - V_k \leq -\beta V_k < 0 \tag{8}$$

which simplifies to

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

[5] showed that equation 9 can be transformed via Schur's complement (among other transformations) into

$$\begin{bmatrix} W_{k+1} & A_k + B_k L_k \\ (A_k + B_k L_k)^\top & (1-\beta)W_k \end{bmatrix} > 0 \tag{10}$$

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

III. Methods

A. Contraction Metric and Controller Synthesis

1) Sum of Squares (SOS) Programming: In order to synthesize the contraction metric and controller, we use the SOS programming framework described in [5] with some slight modifications.

$$\begin{aligned} &\min_{l_c, w_c, r} r \\ &s.t. \forall k, w^\top \Omega w - r w^\top w \in \Sigma(x_k, u_k, w) \\ &\quad r \geq 0.1 \end{aligned} \tag{11}$$

where $\Sigma(x_k, u_k, w)$ is the set of SOS polynomials that satisfy the contraction condition in equation 10. l_c are the polynomial coefficients of L and w_c are the polynomial coefficients of W.

$$\begin{aligned} \Omega &= \begin{bmatrix} W_{k+1} & A_k + B_k L_k \\ (A_k + B_k L_k)^\top & (1-\beta)W_k \end{bmatrix} \\ W_k &= \begin{bmatrix} W_{11_k} & W_{12_k} & W_{13_k} & W_{14_k} & W_{15_k} \\ W_{12_k} & W_{22_k} & W_{23_k} & W_{24_k} & W_{25_k} \\ W_{13_k} & W_{23_k} & W_{33_k} & W_{34_k} & W_{35_k} \\ W_{14_k} & W_{24_k} & W_{34_k} & W_{44_k} & W_{45_k} \\ W_{15_k} & W_{25_k} & W_{35_k} & W_{45_k} & W_{55_k} \end{bmatrix} \\ L_k &= \begin{bmatrix} L_{11_k} & L_{12_k} & L_{13_k} & L_{14_k} & L_{15_k} \\ L_{21_k} & L_{22_k} & L_{23_k} & L_{24_k} & L_{25_k} \end{bmatrix} \end{aligned} \tag{12}$$

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

$$v(x_k) = [x_{k_4}^4, x_{k_3}x_{k_4}^3, x_{k_3}^2 x_{k_4}^2, \cdots, x_{k_1}, x_{k_0}, 1] \tag{13}$$

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

We note the difference in the way we use the slack variable r compared to [5]. In [5], 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 whihc 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. 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 [7], 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 sets an upper bound of the size of the optmization program, which for a monomial basis of degree 4, was around 2000 samples. 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. 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.

B. Online Geodesic and Controller Computation

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

For a smooth curve $c(s), s \in [0,1]$ that connects two points in state space x_0 and x_1, [6] defines the Riemannian length and energy of the curve as

$$\begin{aligned} L(c) &= \int_0^1 \sqrt{\frac{\partial c(s)}{\partial s}^\top M(c(s)) \frac{\partial c(s)}{\partial s} ds} \\ E(c) &= \int_0^1 L(c)^2 ds \end{aligned} \tag{14}$$

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

$$\begin{aligned} \gamma(x_0, x_1) &= \underset{c}{\operatorname{argmin}}\, L(c) \\ &= \arg\min_c E(c) \end{aligned} \tag{15}$$

By the contraction condition we enforced, we see that the Riemanian energy of geodesic decreases exponentially as the system evolves and thus can be thought of as an incremental Lyapunov function [6]. 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

$$\begin{aligned} \bar{\gamma}(x_k^*, x_k) = \underset{\substack{x[\cdot], \Delta x_s[\cdot], \\ \Delta s[\cdot], m[\cdot], y[\cdot]}}{\operatorname{argmin}} & \sum_{i=0}^{N-1} y[i] + \Delta s[i]^2 \\ s.t. \forall i, y \geq & \Delta s[i] \Delta x_s[i]^\top M(m[i]) \Delta x_s[i] \\ & x[0] = x_k^*, \quad x[N] = x_k \\ & \forall i, x[i+1] = x[i] + \Delta x_s[i] \Delta s[i] \\ & \forall i, s[i] > 0, \quad \sum_{i=0}^{N-1} s[i] = 1 \\ & \forall i, M(m[i]) W(x[i]) = I \end{aligned} \tag{16}$$

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 Riemanian energy and N is the number of segments the 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 $M(m[i])W(x[i]) = \mathbf{I}$ enforces $M(m[i]) = W(x[i])^{-1}$. Adding $\Delta s[i]^2$ to the objective serves to spread out the discretized points evenly along the geodesic.

As this is a non-convex program, we use Drake, this time using SNOPT under the hood [7].

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

$$\begin{aligned} u_k &= u_k^* + \sum_{i=0}^{N-1} \Delta s[i] K(x[i]) \Delta x_s[i] \\ &= u_k^* + \sum_{i=0}^{N-1} \Delta s[i] L(x[i]) W(x[i])^{-1} \Delta x_s[i] \end{aligned} \tag{17}$$

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. Results and Discussion

A. Pinacle of success

B. Effect of different parameters

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

C. Degree of metric on

Higher degree means its more able to find a contraction metric for a system with more ”warped” dynamics, i.e. if there is less smoothing.

But higher degree also means less samples can be used, and increased computation time.

D. Effect of number of samples on controller performance

References

[1] 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).

[2] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake. (Feb. 27, 2023). 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], [Online]. Available: http://arxiv.org/abs/2206.10787 (visited on 03/15/2023), preprint.

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

[4] 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).

[5] 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).

[6] 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.

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