Control with Sums-of-Squares Polynomials for Autonomous Hybrid Systems

Sahit Chintalapudi
sahit@mit.edu

***Abstract*—We study control of the cartpole system in an environment where it is surrounded by two compliant (spring-loaded) walls. Using Approximate Dynamic Programming we can model the cost-to-go of the cartpole as a Sums-of-Squares (SOS) polynomial. This approach allows us to produce one such polynomial for each mode of the system and depending on the mode of the system we use the associated polynomial for control. We also experiment with reducing the discontinuities between value functions by solving a second optimization problem, but we find this does not help the resulting policies. The code used to run these experiments can be found at https://github.com/chsahit/springy_cartpole. The accompanying video for this project, including videos of the cartpole motion is available at: https://youtu.be/dKyI0MjVcDs**

I. Introduction

The system we are trying to control is the cartpole surrounded on both sides by compliant walls. In [2], [3] this system is used as a benchmark for algorithms meant to control systems with hybrid dynamics. See Figure 1 for a visualization. The outermost walls are fixed to the world and the inner walls are bound to them with a spring that has known stiffness k and 0 damping. We assume that the cartpole will never make contact with the outerwalls (because of the mass of the cartpole and the sufficiently high stiffness of the spring this assumption holds on simulated runs).

Despite the fact that the force produced by the springs is simply linear in the displacement of the spring (Hooke's law gives us $F_{spring} = -k(x - x_0)$ with x_0 as the neutral position of the spring), these springs make non-trivial changes to the dynamics of the system.

It is also important to note that this system is an *autonomous hybrid system*. The dynamics of the system depend on the mode that the system is currently in. This system has five modes, (1) no contact, (2) cart contact with the left spring, (3) pole contact with the left spring, (4) cart contact with the right spring, (5) and pole contact with the right spring (here ”pole” refers to the point mass at the end of the pole). The guard functions which are used to used to identify changes in mode take the following form.

$$\phi_{12} = \phi_{21} = x_{cart} - x_{01} \quad \phi_{13} = \phi_{31} = x_{cart} + l\sin\theta - x_{01}$$

$$\phi_{14} = \phi_{41} = x_{cart} - x_{02} \quad \phi_{15} = \phi_{51} = x_{cart} + l\sin\theta - x_{02}$$

(Note we are using the same coordinate system as in class, with $\theta = 0$ being the pole laying under the cart). Here x_{01} is the resting position of the left spring and x_{02} is the resting position of the right spring. The reset function here can be left as the identity function and the associated dynamics $f(x, u)$ of the cartpole for each mode i will be denoted as f_i. See Appendix A for the derivation of these functions.

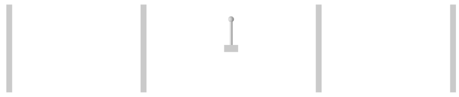

Fig. 1. Underactuated Cartpole surrounded by compliant walls. In experiments, the rigid outer walls are kept sufficiently far away from the cartpole to ensure collisions are only with compliant walls.

By estimating the cost-to-go (expected cost to be incurred) we can build a controller that picks actions that minimize this quantity, achieving the end goal of stabilizing the cartpole from arbitrary initial conditions. In order to address the hybrid nature of the dynamics, we experiment with running a secondary level of optimization problems to inform the cost-to-go functions of one another by minimizing their difference near the edges of the modes.

II. Approximate Dynamic Programming with Sums of Squares Polynomials

This first section reviews the HJB equation, and how this equation can be relaxed into an inequality. The next section discusses how the inequality requiring Positive Definiteness can further be relaxed into a SOS requirement and how we perform this relaxation using the S-procedure. It concludes by presenting the optimization problem that is solved in order to obtain a lower bound on the cost-to-go.

A. Hamilton-Jacobi Bellman Equation

The Hamilton-Jacobi Bellman equation tells us that with the optimal cost-to-go function it holds that

$$0 = \min_u \left[\ell(x, u) + \frac{\partial J^*}{\partial x} f(x, u) \right]$$

Thus, using the optimal control u^* it holds that

$$\frac{\partial J^*}{\partial x} f(x, u^*) = -\ell(x, u^*)$$

Solving this partial differential equation is impractical for many systems. In this work we focus on *lower bounding* the cost-to-go. In other words we relax the above equality to the inequality

$$\frac{\partial J^*}{\partial x} f(x, u^*) \geq -\ell(x, u^*) \leftrightarrow \frac{\partial J^*}{\partial x} f(x, u^*) + \ell(x, u^*) \geq 0 \, \forall x$$

B. Using SOS Polynomials

We can further relax this positive definiteness constraint on the $\frac{\partial J^*}{\partial x} f(x, u^*) + \ell(x, u^*)$ expression by leveraging the structure of polynomials. Specifically, we can use polynomials of a fixed degree to represent J. If we represent the system dynamics as a polynomial and the loss function as a polynomial then the full expression becomes polynomial. Instead of constraining this polynomial to be positive definite, we can use a tighter constraint of making the polynomial a Sums-of-Squares Polynomial.

In order to represent the dynamics as a polynomial, we will leverage the S-Method. The nonlinear terms in the dynamics are $sin\theta$, $\cos\theta$, and $\frac{1}{(m_c + m_p \sin^2\theta)}$. We will represent replace these terms with new variables s, c, z respectively. This adds new terms to the dynamics

$$\dot{s} = c\dot{\theta} \qquad \dot{c} = -s\dot{\theta} \qquad \dot{z} = -2z^2 m_p sc\dot{\theta}$$

When constraining the polynomial J, we add the following ”Lagrange multiplier” like terms to the polynomial

$$L_1(s^2 + c^2 - 1) \qquad L_2(z(m_c + m_p s^2) - 1)$$

Composing these ideas, we can solve for the cost-to-go by solving the program

$$\max_{J, L_1, L_2} \int_X J(x)$$

subject to

$$\dot{J}(x) + \ell(x, u) + L_1(s^2 + c^2 - 1) + L_2(z(m_c + m_p s^2) - 1) \text{ is SOS}$$

$$J([0, 0, 0, -1, 0, \frac{1}{m_c}]) = 0$$

For this work, the cost-to-go takes its arguments in the order $x_{cart}, \dot{x}_{cart}, s, c, \dot{\theta}, z$.

III. Experiments

Experiments are designed an run using the Drake[4] library. These results are generated using the loss function

$$\ell(x, u) = x_{cart}^2 + 10 \cdot (c + 1)^2 + 0.3\dot{\theta}^2$$

We use the model cartpole provided in Drake, where the mass of the cart is 10 kg, the mass of the point mass at the end of the pole is 1 kg, and the pole has a length of 0.5m. Because the loss function does not depend on the control, and the dynamics are affine in the control, the optimal policy is a bang-bang controller.

The controller behaves as follows. If the cartpole is very far from the origin or the pole is significantly offset from the upright position the cartpole picks the control:

$$u = \arg\min_u [\ell(x, u) + \frac{\partial J}{\partial x} f_i(x, u)]$$

$$\text{subject to: } |u| < u_{max}$$

(The appropriate dynamics function f_i is determined based on the state before the control is selected). If the controller is close to the origin and the cartpole is close to being upright, the controller linearizes the system about the upright fixed point and then uses the LQR algorithm to generate an optimal control matrix K. The controller then outputs $u = -Kx$. This output is also clipped such that $|u| < u_{max}$

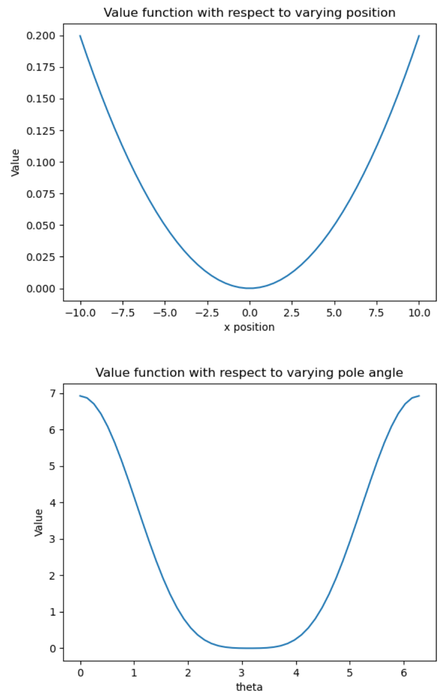

Fig. 2. Plots of the approximate value function as we allow x and theta to vary respectively. Both behave reasonably, with minima at the upright fixed point. Because our polynomial has terms like $\sin\theta, \cos\theta$ in it, we can express behavior that looks richer than a standard quadratic as seen in the second graph.

A. Control of the cartpole in fixed, contact-free dynamics

We begin by studying the policy and cost-to-go function obtained in the no-contact mode. The cost-to-go functions look relatively similar with degrees of 2-4. With degrees of 5-6 the computer begins to run out of memory. The Lagrange-multiplier like terms from the S-Procedure are also set to have the same degree as the polynomial itself. A good first step is to visualize the value function. Since this five dimensional function cannot be visualized directly, we first fix all values except for x_{cart} which we let vary, and then we fix all values except θ which we let vary. Fixed values are held at their value in the upright configuration at the origin. In Figure 2 we see the cost functions obtained for solving the mathematical program. Even modelling them with quartic polynomials, the functions exhibit simple behavior tending away from $x = 0$ (the origin) and $\theta = \pi$ (the upright configuration) respectively.

Our benchmark to compare the controller against is an eneeg-based swingup controller used collocated partial feedback linearization (This implementation follows the ideas in chapter 3 of [1], however the gains of the controller have to be tuned correctly and constants that were set to 1 for

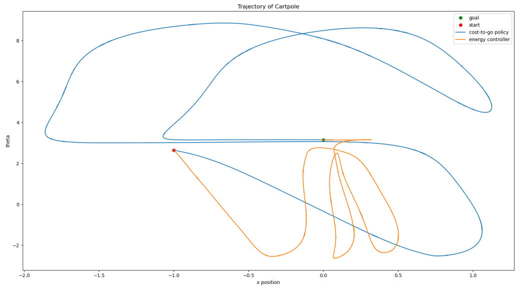

Fig. 3. Plots of the trajectories of the cartpole in a collision free environment. The blue line is the cost-to-go based-policy and the orange line comes from the energy based controller.

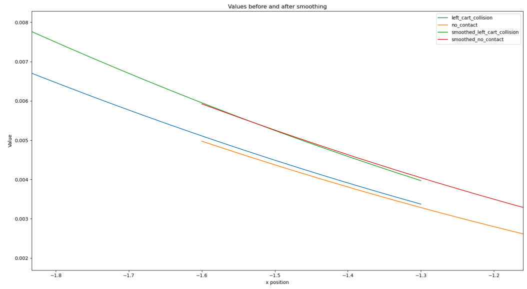

Fig. 4. A zoomed in graph of value functions. The orange line is an initial value function for the contact free mode and the blue line is an initial value function for the contact mode with the left spring. After solving the smoothing optimization problem we get the new red and green value functions.

simplicity were reintroduced). Implementation details of the swingup controller are available in Appendix B.

If we run the policy of minimizing cost-to-go, we see that our controller can sometimes stabilize the cartpole from initially unstable configurations. In Figure 3 we see that our controller is less "efficient" than the energy based controller - i.e it visits a larger area of the state space. Furthermore, our controller is much more brittle to disturbances than the energy based controller. In Figure 3 the cartpole starts 1 meter away from the origin and has an angular displacement of 0.5 radians. If we begin to increase the number the controller fails to stabilize the system altogether. This may be further resolved with more tuning of the constants and terms in the loss function. That being said, the optimized controller takes about 11 seconds to stabilize the system whereas the energy based system took 15 seconds.

B. Control of the cartpole in hybrid dynamics

Each of the cost-to-go functions obtained for the modes have no knowledge that their are other modes in the environment. As a result, their estimates of the cost-to-go in states near the edges of each mode can be inaccurate. For example, if the cartpole in the no-contact mode near the left spring has a negative velocity, it may be more optimal to allow a collision with the spring to be returned to the origin rather than trying to drive towards the origin using the actuator. In this work we apply a post-processing step to the cost-to-go functions that drives the functions together near the boundaries of the modes.

The method is as follows. Given two cost-to-go functions, e.g J_i and J_j we first identify the boundary of the mode $x_{cart} = x_{01}$ and then define $X = [x_{01} - \epsilon, x_{01} + \epsilon]$ We then solve the following optimization problem

$$\min_{\tilde{J}_i, \tilde{J}_j} \left(\int_X \int_{s,c,\dot{\theta},z} \tilde{J}_i - J_j \right)^2 + \left(\int_X \int_{s,c,\dot{\theta},z} \tilde{J}_j - J_i \right)^2$$

subject to

$$(\tilde{a}^i_k - a^i_k)^2 < \delta \quad \forall j$$

$$(\tilde{a}^j_k - a^j_k)^2 < \delta \quad \forall j$$

Where the decision variables a^i_j are coefficients j for the polynomial cost-to-go J^i. Our new cost-to-go functions $\tilde{J}_i$ are kept close to the original lower-bounds on the cost-to-go J_i by the coefficient constraints but also now are encouraged to be closer to each other at the boundary by the cost function being minimized. If J_i, J_j were value functions for the contact free mode and the collision with left spring mode, for example, then the first term encourages the new function $\tilde{J}_i$ to be close to the original J_j near the resting point of the first spring and the second term encourages the new function $\tilde{J}_j$ to be close to the original J_i without significantly changing the coefficients attached to the polynomials.

In Figure 4 we visualize the smoothing effect this procedure has. The original functions J_i, J_j are shown in blue and orange and the new functions J_i, J_j are shown in blue and red. A concern is that the functions are now slightly elevated, and therefore may no longer be a valid lower bound on the cost-to-go. One solution is to iteratively smooth and then use the new smoothed functions as a warm start for the SOS program, this approach is elaborated on in the Future Work section.

In this way, we run 4 smoothing operations, one is run between the no-contact mode and each of the other four modes. If we visualize the trajectories of the cartpole in this hybrid dynamics setting we see that the cartpole is able to stabilize itself despite the presence of springs in the environment. There are caveats to this result. Firstly, the maximum torque limits had to be increased as compared to the torque limits when controlling the cartpole in a spring-free world. This is likely because the springs rob the controller of some authority over the system by imposing external dynamics, and increased torque limits restores some of that authority to the controller. Secondarily, the fusion approach had no practical improvement on the controller. As a matter of fact, both policies follow the same trajectory. This trajectory is visualized in Figure 5 with youtube videos of the cartpole itself in the Youtube link in the abstract.

IV. Conclusion and Future Work

In this work I set out to approximate the cost-to-go function of a cartpole system using a Sums-of-Squares Polynomial.

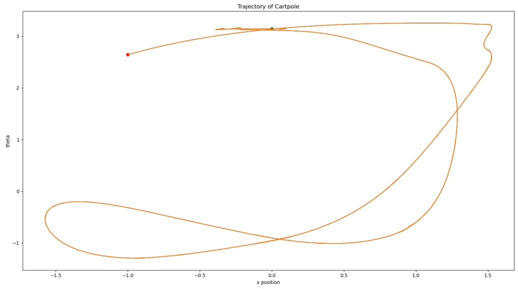

Fig. 5. The trajectory of the cartpole in the hybrid world.

When the cartpole existed in an environment with multiple modes, I aimed to inform each cost-to-go function of the ones in neighboring modes by minimizing their difference subject to not changing the polynomial too much. In the no contact mode, the policy that was induced by the optimization was able to swing the cartpole up from slight offsets, but the policy was very brittle to the loss function and amount of displacement. In the hybrid mode, the system was also to stabilize itself given larger torque limits, but wasn't able to exploit its knowledge of the multi-modal nature of the system.

In the hybrid mode, the original goal was to have the smoothing process be iterative. That is, we solve for lower-bounds in the cost-to-go, then we smooth the resulting functions together, and then we resolve to ensure that the functions are still lower bounds - warm starting from the results of the smoothing process. The cycle would be repeated until convergence or we could not further smooth without violating the lower bound requirement. If this project is continued in the future, this would be a major avenue for improvement.

Another avenue for improvement is being more intelligent about how the parameter ϵ is chosen in the function smoothing step. Currently, we are defining the ball around the mode in terms of the x position of the cart, but this boundary should also include information about the angle of the pole and possibly it's velocities. More aggressive smoothing might be helpful if the cart is moving quickly (this would also require smoothing the cost-to-go functions together in real time).

In this work, we assume the model, dynamics, and state are fully known at all times. In the light of state estimation uncertainty, it might be useful to switch to a dynamic controller, which maintains an internal state to track the mode of the system smoothly. State estimation uncertainty may lead to switches between the modes and abrupt changes in the control, even if the different cost-to-go functions are smoothed to some extent.

ACKNOWLEDGMENT

Thanks to Russ and the TAs for all their support in lecture and on Piazza, I learned so much from this class!

REFERENCES

[1] Russ Tedrake. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). Downloaded on 05/18 from http://underactuated.mit.edu/

[2] Marcucci, T., & Tedrake, R. (2020). Warm start of mixed-integer programs for model predictive control of hybrid systems. IEEE Transactions on Automatic Control.

[3] Aydinoglu, A., Preciado, V. M., & Posa, M. (2020, May). Contact-aware controller design for complementarity systems. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1525-1531). IEEE.

[4] Tedrake, R & The Drake Development Team. (2019). Drake: Model-based design and verification for robotics. https://drake.mit.edu

APPENDIX

A. Derivation of system dynamics

We can use the Lagrangian to produce the equations of motions for the cartpole and then solve the produced system of equations to get our state dynamics. The difference between this derivation and the cartpole dynamics produced in Chapter 3 of [1] is that there is now additional potential energy because of the spring(s) in the system. The quantification of this potential energy changes depending on the mode we are in so we have to repeat this process twice, once if the cart is in contact with the spring and once if the pole at the end of the cart is in contact with the spring. However, the only change in dynamics between if the cartpole system is contacting the left spring or the right spring is the involved neutral position of the spring x_{01} or x_{02}. As a result, we only present the derivation of the dynamics when contacting the left spring with an emphasis on how the springs change the dynamics. x_{01}, x_{02} are not quite the neutral positions of the spring but rather the neutral positions of the spring $+l_{cart}/2$, $-l_{cart}/2$ respectively (because the x position of the cart corresponds to a point in the middle of the cart.

1) Case 1: Cart Contact with Spring: There is now an additional term in the potential energy of the system $Elastic\_Potential\_Energy = \frac{1}{2}k(x - x_{01})^2$

$$T = \frac{1}{2}(m_c + m_p)\dot{x}^2 + m_p\dot{x}\dot{\theta}l\cos\theta + \frac{1}{2}m_pl^2\dot{\theta}^2$$

$$U = -m_pgl\cos\theta + \frac{1}{2}k(x - x_{01})^2$$

Now we can write down the Lagrangian and the Euler-Lagrange Equation for the x-coordinate

$$\mathcal{L} = T - U, \qquad \frac{d}{dt}(\frac{\partial\mathcal{L}}{\partial\dot{x}}) - \frac{\partial\mathcal{L}}{\partial x} = f_x$$

$$(m_c + m_p)\ddot{x} + m_pl(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) - k(x_0 - x) = f_x$$

And also use the Euler-Lagrange equation to get the equation of motion for the theta coordinate. This is the same equation as the standard cartpole because here the spring energy does not depend on θ.

$$m_pl\ddot{x}\cos\theta + m_pl^2\ddot{\theta} + m_pgl\sin\theta = 0$$

We can express these equations of motion using the standard equation. The, inertial, coriolis, and control matrices are all the same. The spring's effects are reflected in a change to the environment compensation term

$$\tau_g(q) = \begin{bmatrix} k(x_0 - x) \\ -m_p g l \sin\theta \end{bmatrix}$$

The manipulator equations can be solved to produce the dynamics

$$\ddot{q} = M^{-1}[\tau_g + Bu - C(q,\dot{q})\dot{q}] \Rightarrow$$

$$\ddot{x} = \frac{k(x_0 - x) + f_x + m_p l \dot{\theta}^2 \sin\theta + m_p g \sin\theta \cos\theta}{m_c + m_p \sin^2\theta}$$

$$\ddot{\theta} = \frac{-\cos\theta(k(x - x_0) + u + m_p l \dot{\theta}^2 \sin\theta) - (m_c + m_p) g \sin\theta}{l(m_c + m_p \sin^2\theta)}$$

2) Case 2: Cart Contact with Pole: Now, we have the new potential energy term

$$U = -m_p g l \cos\theta + \frac{1}{2} k (x + l \sin\theta - x_{01})^2$$

Because the new potential energy term varies with respect to x and θ the Euler-Lagrange equation yields new equations of motion along the x coordinate and θ coordinate

$$(m_c + m_p)\ddot{x} + m_p l(\ddot{\theta}\cos\theta - \dot{\theta}^2 \sin\theta) + k(x + l\sin\theta - x_{01}) = f_x$$

$$m_p l \ddot{x}\cos\theta + m_p l^2 \ddot{\theta} + m_p g l \sin\theta + kl\cos\theta(x + l\sin\theta - x_{01}) = 0$$

The way this influences the manipulator equations is by adding the new terms into the environment compensation term

$$\tau_g(q) = \begin{bmatrix} k(x_{01} - x - l\sin\theta) \\ -m_p g l \sin\theta + kl\cos\theta(x_{01} - x - \sin\theta) \end{bmatrix}$$

Solving the manipulator equations by evaluating the expression $\ddot{q} = M^{-1}[\tau_g + Bu - C(q,\dot{q})\dot{q}]$ gives the dynamics

$$\ddot{x} = \frac{k\sin^2\theta(x_{01} - x - l\sin\theta) + f_x + m_p l \dot{\theta}^2 \sin\theta + m_p g \sin\theta\cos\theta}{m_c + m_p \sin^2\theta}$$

$$\ddot{\theta} = \frac{-\cos\theta(f_x + m_p l \dot{\theta}\sin\theta) - (m_c + m_p) g \sin\theta}{l(m_c + m_p \sin^2\theta)} + \frac{m_c k \cos\theta(x_0 - x - l\sin\theta)}{m_p l(m_c + m_p \sin^2\theta)}$$

maybe consider fixing dynamics (eg 1 - cos2 = sin2, addl term) and also make sure the above eq renders cause it probs wont

B. Swingup Controller Details

We perform partial feedback linearization (PFL) of the equations of motions found in Part A. Since we are linearizing $\dot{x}_{cart}$ this is collocated PFL. The expression is slightly messier than the one found in [1] because we cannot set quantities to 1, but the goal is to modify algorithms to work on the systems we have rather than changing the physical parameters of the system to match the already derived results.

$$\ddot{\theta} = l^{-1}[-\ddot{x}\cos\theta - g\sin\theta]$$

$$(m_c + m_p - m_p\cos^2\theta)\ddot{x} - m_p g \sin\theta\cos\theta - m_p l \dot{\theta}^2 \sin\theta = f_x$$

Thus we can set

$$f_x = (m_c + m_p - m_p\cos^2\theta)\ddot{x}_d - m_p g \sin\theta\cos\theta - m_p l \dot{\theta}^2 \sin\theta$$

The Energy of the pendulum component, with all system constants included is:

$$E(\theta) = \frac{1}{2} m_p \dot{\theta}^2 - m_p g l \cos\theta$$

Ultimately, we end up with the same control law as in class:

$$\ddot{x}_d = k_E \dot{\theta}\cos\theta\tilde{E} - k_p x - k_d \dot{x}$$

For experiments we used:

$$k_E = 1 \qquad k_p = 200 \qquad k_d = 200$$