Contact-Aware Lyapunov Controller Design via Alternating Optimization

Richard Li, Timur Garipov
MIT CSAIL

***Abstract*—In this report, we aim to accomplish three goals. First, we provide a tutorial walking through a paper describing a method for synthesizing Lyapunov stable controllers through contact [1] with detailed and intuitive derivations. Second, we will describe the steps we took to reproduce [1] with a new optimization algorithm in Drake. Finally, we present a derivation of the Lyapunov conditions for a new nonlinear cartpole system.**

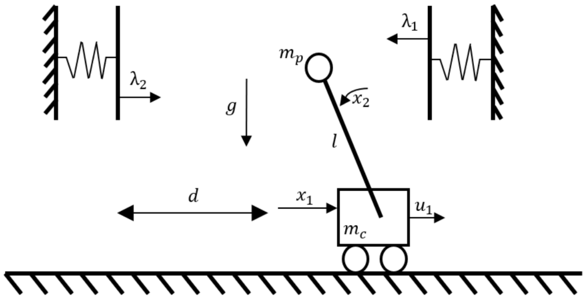


Fig. 1: Diagram of cart-pole with soft walls, showing the contact frames for λ_2 and λ_1 [1].

I. Introduction and motivation

In this paper we consider a spring model of contact dynamics where object penetration is counteracted by a resistive force whose magnitude is determined by Hooke's law. For the contact force between object i and our object of interest O, the spring force is:

$$\lambda_i = \max(0, -k\phi_i(w_O)) \tag{1}$$

where $\phi_i(\cdot)$ is the signed distance in the contact frame of object i and w_O is the witness point on object O. In [1], the main example is a cart-pole with two soft walls, where the objects i are the two walls and the object O is the bob of the pendulum on top of a cart. We are concerned with controlling the cart-pole to regulate the pendulum to the origin while accounting for the contact forces applied by the walls when the bob is in contact.

The main difficulty for Lyapunov controller synthesis in this setting is the piecewise nature of contact, which generates exponential contact modes. Specifically, a contact mode can be defined as a set of binary variables indicating the penetration or non-penetration status between each pair of contacts (based on the signed distance function). Each contact mode is associated with a local "piece" of the piecewise dynamics. For each local dynamics equation $\dot{x_i}$, the enumeration method would require an SOS constraint of an example form:

$$\phi_A(w_O) < 0 \wedge \phi_B(w_O) \geq 0 \wedge \phi_C(w_O) \geq 0 \implies \dot{V}(x;\dot{x_i}) < 0$$

The SOS constraints grow exponentially in the number of contacts, and this quickly becomes computationally intractable.

Instead, [1] replaces the exponential SOS constraints associated with $\dot{V}$ with a single SOS constraint that consists of additive linear complementarity (LCP) constraints in the antecedent of the S-procedure. An example form of this SOS constraint would be:

$$\text{LCP}(x, \lambda_A, \dot{\lambda}_A) \wedge \text{LCP}(x, \lambda_B, \dot{\lambda}_B) \wedge \text{LCP}(x, \lambda_C, \dot{\lambda}_C) \implies \dot{V}(x;\dot{x}) < 0$$

Under this formulation, optimization over the contact modes becomes computationally tractable.

II. Lyapunov and controller reparameterization

For more expressivity, [1] parameterizes the Lyapunov function and controller using the state x *and* the forces λ. We define this reparameterized Lyapunov function:

$$\tilde{V}(x, \lambda) = x^T P x + 2x^T Q\lambda + \lambda^T R\lambda$$

and controller:

$$u(x, \lambda) = Kx + L\lambda$$

The Lyapunov function can also be parameterized only by x:

$$\bar{V}(x) = x^T P x + 2x^T Q\lambda(x) + \lambda(x)^T R\lambda(x)$$

$\tilde{V}(x, \lambda)$ is quadratic in x, λ while $\bar{V}(x)$ is piecewise quadratic in x. To prove asymptotic Lyapunov stability over the state space x, we must prove negative definiteness of the function $\bar{V}(x)$. The directional derivative of $\bar{V}(x)$ is the following:

$$\bar{V}'(x;\dot{x}) = 2x^T P\dot{x} + 2\dot{x}^T Q\lambda(x) + 2x^T Q\dot{\lambda}(x) + 2\lambda(x)^T R\dot{\lambda}(x) \tag{2}$$

For a differentiable Lyapunov function $V(x)$, it is straight forward to compute the time-derivative $\dot{V}(x)$ using the chain rule: $\frac{\partial V}{\partial x}\frac{dx}{dt}$. Unfortunately, we cannot implement the directional derivative in Eq. 2 in our mathematical program due to the piecewise nature of $\lambda(x)$. The solution is to instead compute the directional derivative of the quadratic function

$\tilde{V}(x,\lambda)$ via the chain rule, while treating λ and $\dot{\lambda}$ as indeterminates:

$$\tilde{V}'(x,\lambda;\dot{x},\dot{\lambda}) = \frac{\partial \tilde{V}}{\partial \begin{bmatrix} x \\ \lambda \end{bmatrix}} \begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix}$$

Since λ and λ_i are now indeterminates, we must implicitly enforce 1) constraints on λ relating it back to x and 2) constraints on $\dot{\lambda}$ to correctly compute the directional derivative of λ. The former directly corresponds to the LCP constraints, and the latter is derived from the LCP constraints and specified using additional slack variables (see Sec. III).

III. Incorporating LCP constraints in Lyapunov optimization

We derived the casework for the constraints in Γ_{SOL} and Γ'_{SOL} in [2].

A. LCP constraints on Lyapunov function

$$\mathcal{A} = \left\{ (x,\lambda) \;\middle|\; \begin{array}{l} \lambda \geq 0 \\ Ex + F\lambda + w \geq 0 \\ \lambda_i(E_i^T x + F_{ii}\lambda_i + w_i) = 0 \end{array} \right\}$$

Because λ is now an indeterminate instead of a function of x, it is important to constrain the values of λ to enforce its physical definition and relationship with x. The LCP conditions in set $\mathcal{A}$ constrain λ according to the spring contact model in Sec. I. We can verify the physical correctness of these constraints using casework:

1) No contact: If the signed distance is positive, and since $\lambda \geq 0$, the contact force is 0.

$$\begin{aligned} E_i^T x + w_i > 0 \wedge \lambda \geq 0 &\implies \\ \lambda_i(C + F_{ii}\lambda_i), C > 0 &\implies \\ \lambda_i = 0& \end{aligned}$$

2) In-penetration: If the signed distance is negative, $\lambda_i > 0$, which implies λ_i is defined as the solution to $E_i^T x + F_{ii}\lambda_i + w_i = 0$.

$$\begin{aligned} E_i^x + w_i < 0 \wedge Ex + F\lambda + w \geq 0 &\implies \\ \lambda_i > 0 &\implies \\ E_i^T x + F_{ii}\lambda_i + w_i = 0 &\implies \\ \lambda_i = -\frac{1}{F_{ii}}(E_i^T x - w_i)& \end{aligned}$$

3) Contact boundary: If signed distance is 0, then λ_i is 0.

$$\begin{aligned} F_{ii}\lambda_i^2 = 0 &\implies \\ \lambda_i = 0& \end{aligned}$$

B. LCP constraints on time-derivative of Lyapunov function

$$\mathcal{B} = \left\{ (x,\lambda,\dot{\lambda}) \;\middle|\; \exists\, \xi, \mu : \begin{array}{l} \lambda \geq 0 \\ Ex + F\lambda + w \geq 0 \\ \lambda_i(E_i^T x + F_{ii}\lambda_i + w_i) = 0 \\ \dot{\lambda}_i + \mu_i = 0 \\ \mu_i(E_i^T x + F_{ii}\lambda_i + w_i) = 0 \\ \mu_i \xi_i = 0 \\ \lambda_i \xi_i = 0 \\ E_i^T \dot{x} + F_{ii}\dot{\lambda}_i + \xi_i = 0 \end{array} \right\}$$

We can similarly derive the LCP constraints on $\dot{V}$, by applying casework. Note, for each contact we have two slack variables, μ_i and ξ_i.

1) No contact:

$$\begin{aligned} E_i^T x + w_i > 0 &\implies \lambda_i = 0 \\ &\implies \mu_i = 0 \\ &\implies \dot{\lambda}_i = 0 \end{aligned}$$

2) In-penetration:

$$\begin{aligned} E_i^T x + w_i < 0 &\implies \lambda_i > 0 \\ \lambda_i > 0 \wedge \lambda_i \xi_i = 0 &\implies \xi_i = 0 \\ &\implies E_i^T \dot{x} + F_{ii}\dot{\lambda}_i = 0 \end{aligned}$$

3) Contact boundary:

$$\begin{aligned} E_i^T x + w_i = 0 &\implies \lambda_i = 0 \\ &\implies \dot{\lambda}_i \xi_i = 0 \\ \dot{\lambda}_i &= \begin{cases} -\frac{E_i^T \dot{x}}{F_{ii}} & \xi_i = 0 \\ 0 & \text{else} \end{cases} \end{aligned}$$

Thus, the constraints we use in our Lyapunov optimization problem imply the correct physical dynamics.

Note: We noticed that the contact boundary constraint is actually too restrictive compared to the physical dynamics. This constraint as-is forces our controller to handle both modes of $\dot{\lambda}_i$ at the contact boundary, when only one mode would be visited by the dynamics induced by the controller.

IV. Alternating optimization procedure

We first experimented with the *PENLAB* solver, which is a free version of the *PENBMI* solver used in [1]. While we were able to verify the Lyapunov controller in [1] with *PENLAB*, we were unable to solve for a controller with nonlinear dynamics and contact either from random initializations or LQR initialization. Thus, we were motivated to implement the alterations optimization method, to see if it would scale to the nonlinear case.

The full optimization problem for maximizing the volume of the region of attraction $\{(x,\lambda) \mid \tilde{V}(x,\lambda) \leq \rho\}$ is as follows:

$$
\begin{aligned}
\max\ & \rho \\
\text{s.t. } & \tilde{V}(x,\lambda) - \sum_{i=1}^{2m} \sigma_{g,i}(x,\lambda) g_i(x,\lambda) \\
& \quad - \sum_{i=1}^{m} \sigma_{h,i}(x,\lambda) h_i(x,\lambda) - \gamma_1 \|x\|^2 \text{ is SOS} \\
& \sigma_{g,i}(x,\lambda) \text{ is SOS}, \quad i = 1,\dots,2m \\
& \gamma_2 \|x\|^2 - \tilde{V}(x,\lambda) \text{ is SOS} \\
& -\dot{V}(z) - \sigma_{\text{ROA}}(z)(\rho - \tilde{V}(x,\lambda)) \\
& \quad - \sum_{i=1}^{2m} \sigma_{\hat{g},i}(z) \hat{g}_i(z) \\
& \quad - \sum_{i=1}^{m} \sigma_{\hat{h},i}(z) \hat{h}_i(z) - \sum_{i=1}^{2m} \sigma_{\hat{d},i}(z) \hat{d}_i(z) \\
& \quad - \sum_{i=1}^{3m} \sigma_{\hat{c},i}(z) \hat{c}_i(z) - \gamma_3 \|x\|^2 \text{ is SOS} \\
& \sigma_{\text{ROA}}(z) \text{ is SOS} \\
& \sigma_{\hat{g},i}(z) \text{ is SOS}, \quad i = 1,\dots,2m.
\end{aligned}
\tag{3}
$$

This mathematical program includes

1) Indeterminates: $z = [x, \lambda, \dot{\lambda}, \xi, \mu]$
2) Decision variables:
   - Parameters of the Lyapunov function $\tilde{V}(x,\lambda)$: $P = P^T$, Q, $R = R^T$;
   - Parameters of the controller $u(x,\lambda)$: K, L;
   - Parameter of the Lyapunov ROA (level-set): ρ;
   - Coefficients of the S-procedure multipliers: $\sigma_{g,i}(x,\lambda)$, $\sigma_{h,i}(x,\lambda)$, $\sigma_{\text{ROA}}(z)$, $\sigma_{\hat{g},i}(z)$, $\sigma_{\hat{h},i}(z)$, $\sigma_{\hat{d},i}(z)$, $\sigma_{\hat{c},i}(z)$;

Functions g, h, $\hat{g}$, $\hat{h}$, $\hat{d}$, $\hat{c}$ define the constraints for sets $\mathcal{A}$ and $\mathcal{B}$. In particular g and $\hat{g}$ define linear inequality constraints

$$
g(x,\lambda) = \hat{g}(z) = \begin{bmatrix} \lambda \\ Ex + F\lambda + w \end{bmatrix} \geq 0, \tag{4}
$$

h and $\hat{h}$ define the quadratic equality constraints

$$
h(x,\lambda) = \hat{h}(z) = [\lambda \odot (Ex + F\lambda + w)] = 0, \tag{5}
$$

$\hat{d}$ defines the linear equality constraints involving slack variables

$$
\hat{d}(z) = \begin{bmatrix} \dot{\lambda} + \mu \\ Ex + F\dot{\lambda} + \xi \end{bmatrix} = 0, \tag{6}
$$

and finally $\hat{c}$ defines the quadratic equality constraints involving slack variables

$$
\hat{c}(z) = \begin{bmatrix} \mu \odot (Ex + F\lambda + w) \\ \mu \odot \xi \\ \lambda \odot \xi \end{bmatrix} = 0. \tag{7}
$$

The alternating optimization algorithm we implemented is described in Alg. 1. We split the decision variables into two sets, and alternately optimize over each set, while increasing the region of attraction.

Algorithm 1 Control-design via alternating optimization

```
Variable partition:
$\text{Set}_V = [P, Q, R, \rho]$, $\text{Set}_u = [K, L, \sigma_{\text{ROA}}]$,
$\text{Set}_S = [\sigma_g, \sigma_{\hat{g}}, \sigma_h, \sigma_{\hat{h}}, \sigma_{\hat{d}}, \sigma_{\hat{c}}]$

Initialization: small ROA for LQR
$P \leftarrow S_{\text{LQR}}$, $Q \leftarrow 0$, $R \leftarrow 0$, $K \leftarrow -K_{\text{LQR}}$, $L \leftarrow 0$
Solve (3) for $\text{Set}_S$ with fixed $\text{Set}_V \cup \text{Set}_u$

Alternating optimization
for number of iterations do
    Solve (3) for $\text{Set}_V \cup \text{Set}_S$ with fixed $\text{Set}_u$
    for $\rho_{\text{guess}} \in$ line-search($\rho$) do
        Solve (3) for $\text{Set}_u \cup \text{Set}_S$ with fixed $\text{Set}_V$, $\rho = \rho_{\text{guess}}$
        if feasible solution found then
            break
        end if
    end for
end for
```

Note: For the first alternation set $\text{Set}_V \cup \text{Set}_S$, since both $\tilde{V}$ and ρ are decision variables, maximizing ρ does not correspond to increasing the region of attraction as $\tilde{V}$ may be proportionally scaled. We could fix ρ and minimize the determinant of the matrix of $\tilde{V}$, but minimizing the determinant is not convex. Thus, we are forced to use a heuristic for increasing the RoA - namely, setting bounds on $\tilde{V}$.

V. Simulation results on cart-pole with soft-walls

We tested the controller-design algorithm described in Section IV on a cart-pole with soft walls systems [2].

We implemented the simulation setup in Drake [3]. The contact forces were computed by Drake's contact solver in the point-contact mode. The visualization of the simulation environment is shown in Figure 2a.

Figures 2b and 2c show the results of simulation for LQR controller and our contact-aware controller synthesized via alternations procedure. The Figures show time-evolution of the x-coordinate of the pole $x_{\text{pole}}(t)$. Both trajectories were executed from the same initial state

$$
[x_{\text{cart}}, \theta_{\text{pole}}, \dot{x}_{\text{cart}}, \dot{\theta}_{\text{pole}}] = [0.03, 0, -2.25, 0.9].
$$

LQR has no knowledge of the walls, which leads to diverging trajectory as the dynamics become affected by the wall contact forces. In contrast the contact-aware controller successfully utilizes the force feedback to stabilize the system.

Following [2], we perform quantitative comparison of the controllers. We repeat simulation from $N = 50$ random initial states sampled from the following distribution

$$
\begin{aligned}
10 \cdot x_{\text{cart}}(0) &\sim U(-1,1), \quad \theta_{pole}(0) = 0, \\
\frac{1}{4} \cdot \dot{x}_{\text{cart}}(0) &\sim U(-1,1), \quad \dot{\theta}_{\text{pole}} \sim U(-1,1).
\end{aligned}
$$

We simulated the system for $T = 40$ seconds. We consider the performance of the controller successful if it regulated the system to the origin (up to a certain threshold). The success rates of LQR and our contact-aware controller are

- LQR: 36%;
- Contact-aware: 100%.

VI. Nonlinear cart-pole dynamics and basis

While [1] derived the dynamics for the linearized cart-pole with contacts, we had to manually derive the dynamics and SOS basis for the nonlinear cart-pole with contacts, as well as implement various reparameterizations to make it amenable for SOS optimization. We used the techniques of implicit dynamics parameterization and using a trigonometric basis. While we were able to generate the mathematical formulation, we could not get the optimization to solve for a controller yet.

$$(m_c+m_p)\ddot{x}+m_pl\cos\theta\ddot{\theta}-m_pl\dot{\theta}^2\sin\theta-u+\lambda_1-\lambda_2=0 \quad (8)$$

$$l(m_p\cos\theta\ddot{x}+m_pl\ddot{\theta}-m_pg\sin\theta-\cos\theta\lambda_1+\cos\theta\lambda_2)=0 \quad (9)$$

$$y=\begin{array}{c}y_1\\y_2\\y_3\\y_4\\y_5\\y_6\end{array}\begin{pmatrix}x\\c=\cos(\theta)\\s=\sin(\theta)\\\dot{x}\\\dot{\theta}\\q=(\dot{\theta})^2\end{pmatrix}\qquad \dot{y}=\begin{array}{c}\dot{y}_1\\\dot{y}_2\\\dot{y}_3\\\dot{y}_4\\\dot{y}_5\\\dot{y}_6\end{array}\begin{pmatrix}\dot{x}\\\dot{c}\\\dot{s}\\\ddot{x}\\\ddot{\theta}\\\dot{q}\end{pmatrix} \quad (10)$$

Implicit dynamics: $\dot{y}$ is an indeterminate that implicitly enforces the dynamics via the S-procedure.

$$g(y,\dot{y},\lambda)=$$

$$\begin{bmatrix}\dot{y}_1-y_4\\\dot{y}_2+y_3y_5\\\dot{y}_3-y_2y_5\\(m_c+m_p)\dot{y}_4+m_pl(y_2\dot{y}_5-y_6y_3)-u+\lambda_1-\lambda_2\\m_ply_2\dot{y}_4+m_pl\dot{y}_5-m_pgly_3-ly_2\lambda_1+ly_2\lambda_2\\\dot{y}_6-2y_5\dot{y}_5\end{bmatrix} \quad (11)$$

Dynamical system with LCP:

$$g(y,\dot{y},\lambda)=0 \quad (12)$$

$$0\leq\lambda_1\perp\ell y_3-y_1+\frac{1}{k_1}\lambda_1+d_1\geq 0 \quad (13)$$

$$0\leq\lambda_2\perp-\ell y_3+y_1+\frac{1}{k_2}\lambda_2-d_2\geq 0 \quad (14)$$

$$E=\begin{bmatrix}-1&0&\ell&0&0&0\\1&0&-\ell&0&0&0\end{bmatrix} \quad (15)$$

$$F=\begin{bmatrix}\frac{1}{k_1}&0\\0&\frac{1}{k_2}\end{bmatrix}\qquad w=\begin{bmatrix}d_1\\-d_2\end{bmatrix} \quad (16)$$

$$u(y,\lambda)=Ky+L\lambda \quad (17)$$

$$V(y,\lambda)=y^TPy+2y^TQ\lambda+\lambda^TR\lambda \quad (18)$$

$$\dot{V}(y,\dot{y},\lambda,\dot{\lambda})=2y^TP\dot{y}+2\dot{y}^TQ\lambda+2y^TQ\dot{\lambda}+2\lambda^TR\dot{\lambda} \quad (19)$$

V S-procedure

$$\left.\begin{array}{l}1)\ y_2^2+y_3^2-1=0\\2)\ y_5^2-y_6=0\\3)\ \lambda\geq 0\\4)\ Ey+F\lambda+w\geq 0\\5)\ \lambda_i(E_i^Ty+F_{ii}\lambda_i+w_i)=0\end{array}\right\}\Rightarrow V(y,\lambda)\geq 0 \quad (20)$$

$\dot{V}$ S-procedure

$$\left.\begin{array}{l}1)\ y_2^2+y_3^2-1=0\\2)\ y_5^2-y_6=0\\3)\ \lambda\geq 0\\4)\ Ey+F\lambda+w\geq 0\\5)\ \lambda_i(E_i^Ty+F_{ii}\lambda_i+w_i)=0\\6)\ g(y,\dot{y},\lambda)=0\\7)\ \dot{\lambda}_i+\mu_i=0\\8)\ \mu_i(E_i^Ty+F_{ii}\lambda_i+w_i)=0\\9)\ \mu_i\xi_i=0\\10)\ \lambda_i\xi_i=0\\11)\ E_i^T\dot{y}+F_{ii}\dot{\lambda}_i+\xi_i=0\end{array}\right\}\Rightarrow\dot{V}(y,\dot{y},\lambda,\dot{\lambda})\leq 0 \quad (21)$$

1) is the $\cos^2(\theta)+\sin^2(\theta)=1$ constraint;
2) is the constraint relating $\dot{\theta}$ and $\dot{\theta}^2$
3)-5) are the LCP constraints
6) is the implicit dynamics constraint
7)-11) are the constraints on the $\dot{\lambda}$, time-derivative of the contact-forces.

VII. Project Journey

We began this project with the intention of developing a system for Lyapunov stable grasping and stacking of a cube. While we made significant progress in understanding the theoretical underpinnings of a viable technique for handling contact in Lyapunov controller synthesis, we are still away from our initial goal.

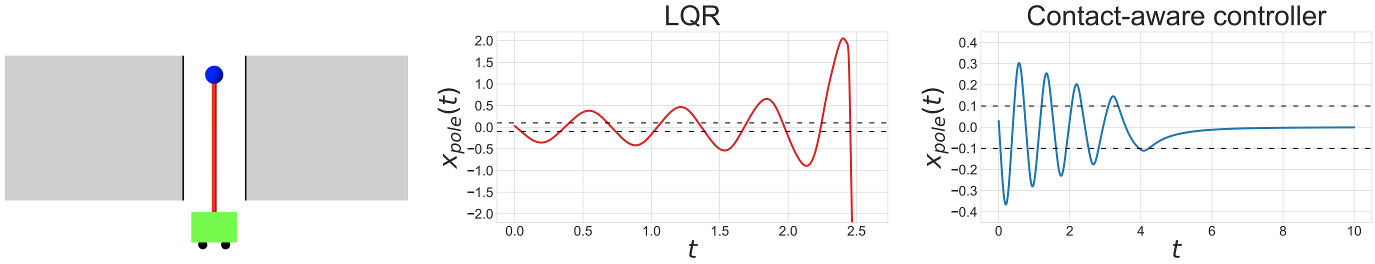
We extensively derived the math and physics in [1], as doing so was essential for reproducing and debugging the code in Drake and for formulating the nonlinear version in Sec. VI. It also laid the groundwork for this tutorial, which is more approachable than the original paper.

We scheduled our coding plan as follows:

1) Verify we can find a controller using our alternating optimization strategy for cart-pole with no walls.
2) Verify cart-pole with walls and contact forces and linear dynamics.
3) Verify cart-pole with walls and contact forces using implicit parameterization of linear dynamics.
4) Verify cart-pole with walls and contact forces using implicit parameterization of nonlinear dynamics.

We were able to accomplish steps 1-3.

Our implementation of the alternating optimization is based on a wrapper of Drake's MathematicalProgram which



(a) Cart-pole with walls simulation setup

(b) Trajectory of the LQR-controlled cart-pole

(c) Trajectory of the cart-pole with contact-aware controller

Fig. 2: Simulation results on the cart-pole with soft-walls system

- sets up the indeterminates, decision variables, the cost, and the constraints in (3) with some decision variable being optionally freezed or initialized from solution obtained on the previous step (depending on the phase of alternations).
- provides a general interface for setting s-procedure multipliers and SOS constraints. We re-implemented part of the Drake's `AddFreePolynomial`, `AddSosPolynomial` functionality to explicitly expose the coefficients decision variables, so that these variables can be freezed or initialized with the previous step values.

VIII. Practical Challenges

A surprising amount of hyperparameter tuning was needed for the SDP solver to converge (with CSDP and even Mosek). We spent time tuning the bounds on V, K and L. Additionally, alternations sometimes would run into numerical issues where the solution to the previous optimization problem would be infeasible for the next alteration iteration. We addressed this by using a strategy where we solve for the maximum ρ in line 4 of Alg. 1, then decrease the optimal value found and solve a second feasibility problem with fixed ρ, which is a heuristic for staying in the interior of the positive semidefinite cone.

IX. Team member contributions

Both team members contributed substantially to project ideation, mathematical derivation, software development, and writing the report and video.

X. Acknowledgements

Thanks to Alp Aydinoglu, Hongkai Dai, and Russ Tedrake for advice and feedback on this project.

References

[1] A. Aydinoglu, V. M. Preciado, and M. Posa, "Contact-aware controller design for complementarity systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1525–1531.

[2] A. Aydinoglu, P. Sieg, V. M. Preciado, and M. Posa, "Stabilization of complementarity systems via contact-aware controllers," *IEEE Transactions on Robotics*, 2021.

[3] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu