Quasi-static Planning Through Contact

Krishna Suresh
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
ksuresh2@andrew.cmu.edu

Robert Mones
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
rmones@andrew.cmu.edu

Giri Anantharaman
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
giria@andrew.cmu.edu

***Abstract*—Manipulation is the ability of an agent to control of its environment through selective contact. Our goal for this class project is to better understand stiff, non-smooth contact dynamics and the resulting exploding / discontinuous gradients, as well as the non-convexity of the planning problem. Planning robot dexterity is challenging due to the non-smoothness introduced by contacts, intricate fine motions, and ever-changing scenarios. Model-based methods have tackled the contact dynamics challenges by smoothing contact dynamics analytically while model-free stochastic RL methods address these challenges by effectively sampling and averaging the contact modes. Recent methods in contact-implicit trajectory optimization solve for control sequences directly without explicitly enumerating changes in dynamics at contact events. Therefore it is possible to solve trajectory optimization problems which can handle making and breaking contact in a quasi-static environment by levraging recent work on log-domain interior-point methods that approximates the classical barrier in a precise sense leading to superior performance. We apply this technique to trajectory optimization problem which can handle making and breaking contact in a quasi-static environment.**

***Index Terms*—Manipulation Planning, Dexterous Manipulation, Motion and Path Planning, Contact Modeling, Convex Optimization, Interior Point Methods, Log Barrier Methods**

I. Introduction

When interacting with the world, humans constantly make and break contact. However, methods for enabling robots to perform the same behaviors struggle due to the challenges posed by the dynamics of contact. Prior work in this area can be broadly categorized into model free approaches and model based approaches.

Recent advances in reinforcement learning (RL) have shown impressive results [5, 10, 4, 3] in manipulation through contact rich dynamics that were difficult to achieve with previous model based methods. However, it is yet unclear what made these methods successful, where model-based methods have struggled.

From a model-based perspective, the most significant obstacle for planning through contact lies in the hybrid nature of contact dynamics (i.e., numerous modes of smooth dynamics separated by guard surfaces). The non smooth nature of the resulting dynamics implies that the Taylor approximation no longer holds locally; thus, the locally linear model constructed with the gradient quickly becomes invalid. This invalidity of the local model presents significant challenges for both iterative gradient-based optimization. Faced with such challenges, many existing works [1, 6, 2, 7, 9, 8] have sought to explicitly consider contact modes by either enumerating or sampling them. With model-based knowledge of the dynamic modes, these planners typically alternate between continuous state planning in the current contact mode and a discrete search for the next mode, resulting in trajectories punctuated by a handful of mode changes. However, methods for enabling robots to perform the same behaviors struggle due to the challenges posed by the dynamics of contact:

- Dynamics non-differentiable at contact events
- Search through contact modes grows combinatorially
- Gradient information about the effects of contact is non-existent before contact occurs

Therefore, recent methods [12] in contact-implicit trajectory optimization solve for control sequences directly without explicitly enumerating changes in dynamics at contact events.

A. Log-barrier interior-point smoothing method

We aim to leverage the log-barrier interior-point smoothing method [11] learned in class to solve trajectory optimization problems which can handle making and breaking contact in a quasi-static environment. Interior-point methods (IPMs) are widely used numerical algorithms for solving convex quadratic programs (QPs) of the form

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}x^T W x + c^T x \\ \text{subject to} \quad & Ax + b \geq 0, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the decision variable, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ define linear inequality constraints, and $W \in \mathbb{R}^{n \times n}$ is a symmetric, positive semidefinite matrix that, together with $c \in \mathbb{R}^n$, defines a convex, quadratic objective. IPMs solve above mentioned optimization problem by tracking the solution $(x, s, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ to the *central-path* conditions

$$\begin{aligned} & A^T\lambda = Wx + c, \;\; s = Ax + b, \\ & ;\lambda \geq 0, s \geq 0, \\ & \quad s_i\lambda_i = \mu \;\; \forall i \in \{1, 2, \ldots, m\} \end{aligned} \tag{2}$$

for a decreasing sequence of $\mu > 0$. When $\mu = 0$, these are precisely the Karush-Kuhn-Tucker (KKT) optimality conditions for the original problem. Hence, by gradually reducing μ to zero, IPMs produce an optimal solution x to the main problem along with an optimal constraint slack s

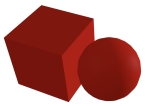
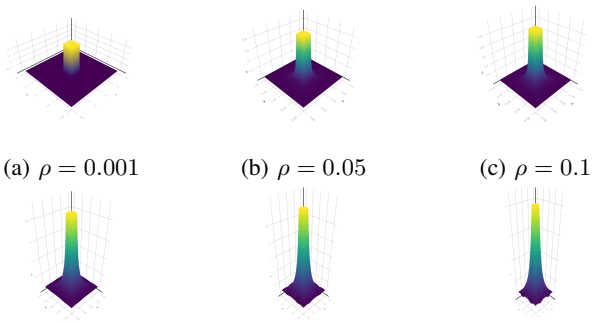


Fig. 1: Block ball system for visualizations of contact force below.



(a) $\rho = 0.001$ (b) $\rho = 0.05$ (c) $\rho = 0.1$

(d) $\rho = 0.5$ (e) $\rho = 1.0$ (f) $\rho = 2.0$

Fig. 2: Effect of ρ on contact dynamics

and corresponding vector λ of Lagrange multipliers. IPMs are really efficient in practice and there are many optimal implementation packages around. Log-domain interior-point methods [11] formulate the problem as follows. The set of nonnegative (s, λ) satisfying $s_i \lambda_i = \mu$ for $i \in \{1, 2, \ldots, m\}$ is easily parameterized in the log-domain: letting $e^v \in \mathbb{R}^m$ denote elementwise exponentiation, this condition holds if and only if $\lambda = \sqrt{\mu} e^v$ and $s = \sqrt{\mu} e^{-v}$ for some $v \in \mathbb{R}^m$. This v-parametrization of s and λ yields the following log-domain reformulation of the central-path conditions:

$$\sqrt{\mu} A^T e^v = Wx + c, \quad \sqrt{\mu} e^{-v} = Ax + b \tag{3}$$

and a template *log-domain interior-point method* for solving the QP:

- Update (v, x) by applying Newton's method to central path for fixed μ.
- Reduce μ and repeat.

II. Method

We use quasi-static dynamics to simplify the problem and use the log-domain trick as described in class to ensure the complementarity problem is solved by construction. This allows us to start with a large ρ to enable force-at-a-distance and non-physicality to allow the solver to find a solution in the right region. Then we can decrease ρ progressively to ensure contact forces occur only at contact. Figure 1 is a visualization of the perceived force at various points in the configuration space for different values of ρ.

For large values of ρ, configurations far away from contact still experience forces, allowing for the optimizer to capture the gross motions of the objects. Then as ρ is reduced the contact forces are only applied at the real contact locations ensuring the realism of the dynamics.

A. Quasi static Contact Dynamics and Optimization

In this section we describe the dynamics of the simple 2D system, that includes a T shaped box and a ball. The ball is actuated, meaning we are able to control it to push and rotate the box. This is reflected in the mass matrix where the ball is assumed to be point mass, with no inertia. In the spring constant matrix, we model the interaction between the ball and T like a spring damper. The state vector has position of the ball and T only. Velocities are assumed to be zero in quasi static formulation. As for Controls they are in $\mathcal{R}^2$ and they only apply to the ball since only ball can be actuated:

$$M = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & m & \\ & & & m \end{bmatrix} \quad K = \begin{bmatrix} k & & & \\ & k & & \\ & & 0 & \\ & & & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

In addition to these we also define, the following:

- Signed distance function: $\phi(q)$
- Delta q: δq
- Contact Jacobian: J and Contact force applied by the box: λ.

$$\mathbf{q} = \begin{bmatrix} q_x^c \\ q_y^c \\ q_x^b \\ q_y^b \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \Delta q_x^c \\ \Delta q_y^c \end{bmatrix} = \begin{bmatrix} q_d - q_x^c \\ q_d - q_y^c \end{bmatrix}$$

Putting it all together we arrive at the equations of motion.

$$M\ddot{q} + P(q) = KBu + J^T \lambda \tag{4}$$

With these basic ingredients, we now formulate the quasi static contact optimization problem as follows:

$$\begin{aligned} &min_{\delta q} \frac{1}{2h} \delta q^T M \delta q + \frac{h}{2} \delta q^T K B \delta q + h(P(q) - KBu)\delta q \\ &\text{subject to } \phi(q_k + \delta q) \geq 0 \end{aligned} \tag{5}$$

B. Planning Through Contact

We linearize the signed distance function constraint, introduce a slack variable, and formulate the log barrier version of the same problem as (in the process we do a change of variables as well):

$$\begin{aligned} &min_{\delta q, s} C(\delta q) - \rho \log(s) \\ &\text{subject to } \phi(q_k) + hJ\delta q - s = 0 \\ &s \geq 0 \end{aligned} \tag{6}$$

The lagrangian for the optimization problem is as follows:

$$
\begin{aligned}
&s = \sqrt{\rho}e^{\sigma} \\
&\lambda = \sqrt{\rho}e^{-\sigma} \\
&\mathcal{L}(\delta q, s, \lambda) = C(\delta q) - \rho \log(s) - \lambda^T(\phi(u) + J\delta q - s) \\
&\nabla_{\delta q}\mathcal{L} : \frac{M\delta q}{h} + hKB\delta q + hP(q) - KBuh - J^T\sqrt{\rho}e^{-\sigma}h \\
&\nabla_s \mathcal{L} : s\lambda = \rho \\
&\nabla_\lambda \mathcal{L} : -\phi(q_k) - J\delta q h + \sqrt{\rho}e^{\sigma}
\end{aligned} \tag{7}
$$

And since $s\lambda$ always must equal ρ, the complementarity condition is satisfied by construction.

C. Planning Through Contact SQP

We can now formulate an SQP to minimize the LQR tracking error for a desired goal configuration $\bar{q}$ while ensuring the contact dynamics by including the KKT conditions above as a equality constraints:

$$
\begin{aligned}
&\min_{q,u,\sigma} \sum_{k=1}^{N} (q_k - \bar{q})^\intercal Q(q_k - \bar{q}) + u_k^\intercal R u_k \\
&\text{subject to} \\
&-v \leq \delta q[1:2] \leq v \\
&-\phi(q_k) - J\delta q h + \sqrt{\rho}e^{\sigma_k} = 0 \\
&\frac{M\delta q}{h} + hKB\delta q + hP(q) - KBuh - J^T\sqrt{\rho}e^{-\sigma_k}h = 0 \\
&\quad \forall k \in 1, 2, \cdots, N-1
\end{aligned} \tag{8}
$$

Where, δq is short hand for $q_{k+1} - q_k$ and J the contact jacobian is a function of q_k. The first constraint imposes a max velocity on the ball's movement.

III. Implementation Details

To test the log-domain formulation of contact dynamics we build on the code provided by Arun Bishop. We implemented a demo of planar push T and 2d push T with rotation. We first test the contact dynamics by simulating a ball moving in a fixed desired trajectory and validating that the contact dynamics are physically valid as we reduce ρ. The implementation of the quasi-static dynamics are dependent on the signed distance function between the objects in the environment. We obtain both the signed-distance function and its gradient from an implementation of DCOL [13]. We also use the gradient of the signed distance function to calculate the contact jacobian for the force normal to the contact point. And for the 2d push T with rotation, we also find the tangent force direction in the contact frame by taking the cross product with the z axis unit vector. Each step of the quasi-static dynamics simulation is computed with newton's method on the KKT conditions (7).

Now that we are able to see that the desired contact physics occurs for a manually specified control input, we can next implement the trajectory tracking SQP where the quasi-static dynamics KKT conditions are specified as equality constraints at each time-step. We then use IPOPT to solve for an optimal control sequence that minimizes the LQR tracking cost.

Fig. 3: Convex Decomposition of Push T

For the push T scenarios since the T is not a convex shape, we manually decompose it into two convex boxes for the top and bottom of the T. This means we need to also double the number of λ's to account for all of the potential contacts. This allows for contact in the inner corner of the T to be represented as two separate contact points.

Through the log-domain trick, we are satisfying complementarity by construction, however, IPOPT may be able to converge to a solution faster if we instead introduce a separate set of σ's for each change of variable and manually impose the relaxed complementarity slackness. Therefore, our quasi-static dynamics KKT conditions look like the following:

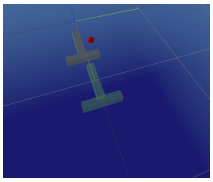
```
[
M*(q_next - q_k)/dt +
    dt*(Pq + K*q_next - K*B*u_k - J'*λ_k)
    # Force balance

signed_dist + J*(q_next - q_k) -
    sqrt(ρ)*exp.(σs_k)
    # Signed distance (positive)

    λ_k - sqrt(ρ)*exp.(σλ_k)
    # Force is positive

    σs_k + σλ_k # Complementarity
]
```

All of our code can be found at https://github.com/RoboticExplorationLab/CILogDomain.jl/tree/ocrl_project.

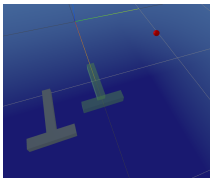
IV. Experiments and Results

A. Rotation-Free Push T

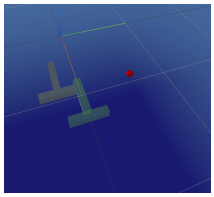
We first experimented with a rotation-free push T task to test our optimization methods and tune parameters. Starting with this task further simplified the dynamics because we were able to model the state of the T with just its position and we did not need to model friction. We experimented with different ρ schedules to find the best performance and found that faster (more rapidly decreasing) ρ schedules led to similar terminal performance while requiring many fewer optimizer iterations. Figures 4 and 5 show the progression of the solutions across many iterations with two different ρ schedules. Each test started with an initial round of 500 iterations followed by rounds of 1000 iterations.



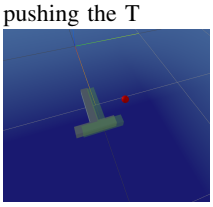
(a) Initial conditions before the ball begins pushing the T



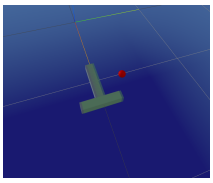
(b) 500 iterations with $\rho = 1$, clear force-at-a-distance



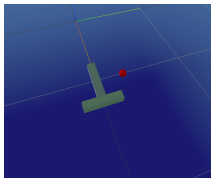
(c) Additional 1000 iterations with $\rho = 10^{-1}$



(d) Additional 1000 iterations with $\rho = 10^{-2}$

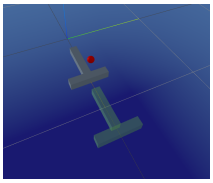


(e) Additional 1000 iterations with $\rho = 10^{-3}$

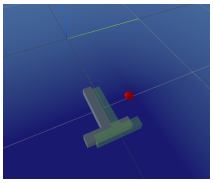


(f) Additional 1000 iterations with $\rho = 10^{-4}$

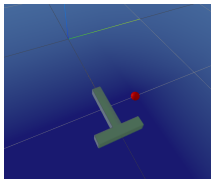
Fig. 4: Rotation-free push T performance over many optimization iterations with a slow ρ schedule.



(a) Initial conditions before the ball begins pushing the T



(b) 500 iterations with $\rho = 10^{-3}$, near goal state

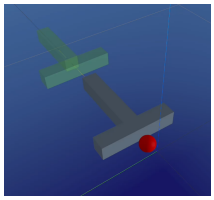


(c) Additional 1000 iterations with $\rho = 10^{-4}$

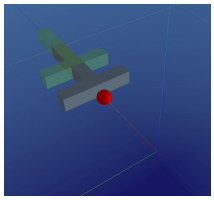
Fig. 5: Rotation-free push T performance over fewer optimization iterations with a faster ρ schedule.

We got the best performance (Figure 5) by starting with $\rho = 10^{-3}$ for the first round and $\rho = 10^{-4}$ for the second. After this, we had a solution that ended very close to the goal state. For slower ρ schedules, the great force-at-a-distance made the ball initially travel away from the T (Figure 4.b). After a few rounds of convergence and ρ decrease, we got a similarly close end configuration to the fast ρ schedule albeit after many more iterations.

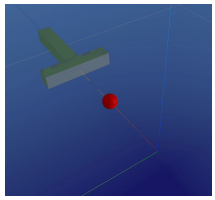
In most of our testing, the solver did not converge but timed out after the iteration limit was reached. This behavior depended on the initial and goal states, with some configurations being able to solve completely in fewer than 300 iterations while many did not converge after several thousand iterations but still produced results that appeared to achieve the goal. We are unsure why the solver exhibits this behavior but believe it may be due to the lack of friction in our rotation-free model and thus the ball's inability to apply any non-normal forces when contacting the T, leading to it being difficult to push the T in some directions while others are more straightforward.



(a) Initial conditions for a push T task with rotation

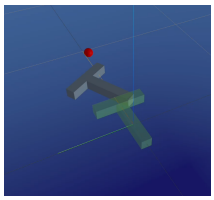


(b) Ball begins pushing the T while maintaining alignment

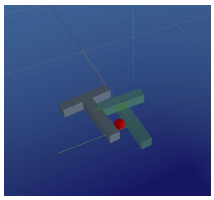


(c) Ball backs away from the T to reach the goal state

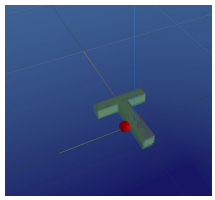
Fig. 6: A push T task with rotation that only requires rotating the T to keep it aligned with the goal state.



(a) Initial conditions for another push T task with rotation



(b) Ball begins pushing the T from the right



(c) Ball finishes pushing the T from the left

Fig. 7: A push T task with rotation which requires rotating the T to align it with the goal state. The solution trajectory first pushed the T from the right and then from the left to achieve the goal state.

B. *Push T with Rotation*

Next, we implemented rotation of the T and a friction model. This allowed the optimizer to solve more complex tasks in addition to simpler ones like shown previously. Figure 6 shows the solution trajectory of the ball pushing the T directly to the goal state. If we allowed the T to rotate without the friction model, even this task would be impractical because the ball needs to use small lateral forces to keep the T aligned while approaching the goal.

Figure 7 shows the solution trajectory of the ball pushing the T to the goal state while rotating it and pushing it from multiple directions. As shown by Figures 7a and 7b, the ball first moved to the right of the T to align it with the goal orientation. As shown by Figures 7b and 7c, the ball then traveled to the other side of the T to push it to the goal position. This behavior demonstrates two important capabilities of the system: 1) the optimizer's ability to solve for trajectories which rotate the T by taking advantage of the lateral friction; and 2) the optimizer's ability to solve for trajectories which involve pushing the block from distinct directions as part of a single trajectory.

Both of these tests with rotation and friction implemented converged in about 300 iterations but we did run similar tests which did not converge after 1000 iterations.

V. Conclusion

We explored a new formulation of contact smoothing through the log domain trick on quasi-static dynamics.

Through this project, we derived the quasi-static dynamics for a ball and block pushing system and extended this to the non-convex push T. We then evaluated the planning performance of trajectory optimization through these contact events with a Sequential Quadratic Program solved with IPOPT. Finally, we tested various sets of starting configurations, initial conditions and parameters to learn about the effectiveness of smoothed contact dynamics. More work is needed to determine exactly what features of this construction prevent the optimizer from converging in a few hundred iterations in some tasks as well as further test its robustness in more complex tasks, especially those taking advantage of contact friction to manipulate objects.

ACKNOWLEDGMENT

We thank folks from CMU RexLab, especially Arun Bishop and Zachary Manchester for their help, support and guidance throughout this project.

REFERENCES

[1] F.H. Clarke. *Optimization and Nonsmooth Analysis*. Wiley New York, 1983.

[2] Alexander Shkolnik, Matthew Walter, and Russ Tedrake. "Reachability-guided sampling for planning under differential constraints". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 2859–2865.

[3] Aravind Rajeswaran et al. *Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations*. 2018. arXiv: 1709.10087 [cs.LG].

[4] Anusha Nagabandi et al. *Deep Dynamics Models for Learning Dexterous Manipulation*. 2019. arXiv: 1909.11652 [cs.RO].

[5] Andrychowicz et al. "Learning dexterous in-hand manipulation". In: *The International Journal of Robotics Research* 39.1 (2020), pp. 3–20.

[6] James Burke et al. "Gradient Sampling Methods for Nonsmooth Optimization". In: Feb. 2020, pp. 201–225. ISBN: 978-3-030-34909-7.

[7] Albert Wu, Sadra Sadraddini, and Russ Tedrake. "R3T: Rapidly-exploring random reachable set tree for optimal kinodynamic planning of nonlinear hybrid systems". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 4245–4251.

[8] Claire Chen et al. "TrajectoTree: Trajectory Optimization Meets Tree Search for Planning Multi-contact Dexterous Manipulation". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 8262–8268.

[9] Shadi Haddad and Abhishek Halder. *Anytime Ellipsoidal Over-approximation of Forward Reach Sets of Uncertain Linear Systems*. 2021. arXiv: 2103.04545 [eess.SY].

[10] Tao Chen, Jie Xu, and Pulkit Agrawal. "A system for general in-hand object re-orientation". In: *Conference on Robot Learning*. PMLR. 2022, pp. 297–307.

[11] Frank Permenter. *Log-domain interior-point methods for convex quadratic programming*. 2022. arXiv: 2212.02294 [math.OC]. URL: https://arxiv.org/abs/2212.02294.

[12] Tao Pang et al. *Global Planning for Contact-Rich Manipulation via Local Smoothing of Quasi-dynamic Contact Models*. 2023. arXiv: 2206.10787 [cs.RO]. URL: https://arxiv.org/abs/2206.10787.

[13] Kevin Tracy, Taylor A. Howell, and Zachary Manchester. *Differentiable Collision Detection for a Set of Convex Primitives*. 2023. arXiv: 2207.00669 [cs.RO]. URL: https://arxiv.org/abs/2207.00669.