Components of Contact: An Analysis of Contact-Implicit Control Strategies on Manipulation Primitives

Grey Sarmiento, Michael Posa

***Abstract*— In uncertain, complex, and dynamic robotic tasks, contact with objects and the environments must often be planned or adapted during execution to achieve reasonable performance. Due to discontinuous and non-smooth contact dynamics, model-based control has historically struggled to find real-time solutions. However, recent years have seen the introduction of contact-implicit controllers which introduce relaxations into the dynamics and optimal control problems, forgoing optimality in favor of realizing rapid solutions for receding horizon control. Although these different methods have led to impressive, but isolated, demonstrations and results, the implications of these relaxations and resulting tradeoffs between them remain unstudied. Furthermore, evaluations lack a unified framework for more thorough comparison, discussion, and analysis. We present a method for comparing and analyzing contact-rich control through the lens of a set of "atomic tasks" which isolate the following components: make-break transitions, frictional stick-slip transitions, and contact-point repositioning. We perform a systematic comparison of three distinct contact-implicit controllers. By analyzing performance on these atomic tasks, in the context of their strategies for approximating the underlying optimal control problem, we shed light on the strengths and weaknesses of these distinct strategies. Results are analyzed in a unified simulation testbed.**

I. Introduction

Model predictive control (MPC) remains a cornerstone in robotic control for achieving efficient, optimal motion that is robust to disturbances. This influence is particularly clear in flat-ground legged locomotion, where MPC algorithms formed the foundation of decades of robotics research [15]. However, progress in model-based control for dexterous manipulation and whole-body locomotion has been comparatively slow. The core challenge arises from the discontinuous (equivalently, non-smooth or hybrid) nature of contact dynamics, where a control algorithm must decide what to touch, when, and where; this combinatoric problem quickly becomes computationally intractable across complex scenes and planning horizons. And yet, the last few years have seen remarkable progress in a suite of strategies known as contact-implicit MPC (e.g. [1, 3, 6–9, 11, 14, 16] and others), where contact forces and events are co-optimized in the control problem, rather than being specified beforehand in an explicit contact sequence. For predictive or model-based reasoning to enable online robot performance in novel tasks, contact-implicit approaches are almost a necessity.

The demands on accuracy for success in contact-rich tasks are especially brittle to misestimation and error, as contact transitions (in or out of contact, frictional sticking or sliding) occur at discrete boundaries. From the perspective of differentiable control, being a meter away from an object and almost grazing its surface both generate zero contact force and gradients provide no indication of an appropriate corrective action. This non-smooth nature of contact must be navigated for many critical tasks in robotics like dexterous and nonprehensile manipulation or whole-body locomotion where arms and legs must be combined to quickly and safely maneuver.

Contact-implicit MPC offers one method for reasonably-optimal control on a nonconvex and discontinuous dynamics landscape. In the past few years, developments in contact-implicit MPC have demonstrated impressive results in dexterous manipulation [1, 7, 9–11, 16] and legged locomotion [6, 8], offering performance competitive with the best learned methods (without requiring pre-training). However, to achieve real-time rates, recent progress has come through various approximations to the underlying optimal-control problem. These approximations lead to suboptimality, which can manifest in poor performance on some tasks or sensitivity to parameter tuning in a manner that remains poorly understood. As each paper demonstrates performance on its own set of task with relatively little overlap, the degree to which one idea might transfer to another task is also unclear. Given this, there is a need for a method of comparison and analysis that will reveal strengths and weaknesses and identify where research efforts should be focused for tangible breakthroughs and progress.

A. Contributions

We present a unified framework for comparing and analyzing contact-implicit MPC controllers through a set of manipulation primitives representing the core decisions involved in contact-rich control. Specifically, we propose a set of "atomic tasks" which through their simplicity are relatively easy to implement and, more critically, to analyze.

Through this framework, we argue that each approach to CI-MPC is ultimately governed by a set of hyperparameters which dictate the controller's capacity to make contact mode transitions. This capacity comes at the cost of some aspect of the task performance, and we work to illuminate this tradeoff for each controller, with the understanding that these tradeoffs will ultimately manifest when the controller is applied to more complex composite tasks.

We demonstrate this analysis framework on three contact-implicit model predictive controllers which span the three categories above, with specific controllers chosen due to availability of functional open-source implementations. (1) iLQR with a soft MuJoCO [13] model, (2) the C3 controller which leverages ADMM and (3) pure predictive sampling from MJPC [4]. All three algorithms are implemented on a common simulation testbed in Drake [12].

In summary, our contributions are as follows:

- A framework for analyzing control of contact-rich tasks through a set of tasks that isolate each contact mode transition.
- Preliminary results for two of these tasks that present relationship between core hyperparameters of each controller and task performance as measured by cost.

Given that our analysis and study is still in progress, we discuss the following as next work in this line of research:

- Additional controllers for analysis.
- Further quantities for measurement and comparison.
- Further tuning of our simulation experiments to improve performance and present results more clearly.

II. Background

In this section we establish notation for the control problem and contact constraints as well as providing an overview of each controller studied.

A. Problem Formulation and Notation

The controllers we focus on in this study are real-time, contact-implicit model predictive controllers for rigid-body robots and mostly rigid environments. We assume access to the robot's state as well as a full model of the robot, objects, and the environment.

We define the robot's state (generalized positions and velocities) $x \in \mathbb{R}^n$, input torques $u \in \mathbb{R}^m$, subject to contact forces $\lambda \in \mathbb{R}^n$, with cost function $c(x_k, u_k, \lambda_k)$ and dynamics function $f(x_k, u_k)$.

B. Complementarity Contact Formulation

When encoding contact as a constraint, the complementarity formulation is common:

$$0 \leq \lambda \perp \varphi(x, u, \lambda) \geq 0$$

where $\varphi(x, u, \lambda)$ represents a distance function encoding both no-force-at-a-distance and friction cone constraints.

C. Methods Analyzed

This study analyzes three different controllers which each take a different approach to contact-implicit control: MuJoCo's Predictive Sampling and iLQG controllers presented in [4] and Consensus Complementarity Control presented in [1]. The structure of each is briefly summarized here.

1) MuJoCo MPC Predictive Sampling (MJPC): The premise of MuJoCo MPC's Predictive Sampling [4] is to sample variations on a reference input (action) trajectory, roll them out using MuJoCo's simulation platform, evaluate the cost of each trajectory, and choose the trajectory with the lowest cost as the new nominal trajectory for the next timestep.

2) MuJoCo MPC iLQR: MuJoCo MPC's implementation of iLQG is stated to not take advantage of the Gaussian component of the algorithm, and is instead an actual implementation of iLQR (the iterative linear quadratic regulator). iLQR performs a time-varying version of the linear quadratic regulator at every timestep, taking advantage of both the first and second order terms of the cost in order to improve the nominal control input. In order to allow these derivatives to reflect change in cost due to contact, contact must be softened. MJPC does this through MuJoCo's compliant contact model, which also provides contact gradients.

3) Consensus Complementarity Control (C3): While MJPC can be thought of as handling contact modes completely discontinuously, and MJPC-iLQR can be thought of as using soft contact to create one continuous dynamics mode, C3 can be considered as allowing for disagreement between contact dynamics and the rest of the problem dynamics. The main premise of C3 is that if solving for contact force can be decoupled from resolving the dynamics of the problem, MPC can overcome discontinuous contact dynamics and still consider all possible contact pairs at a real-time control rate.

The C3 algorithm consists of three steps:

1) Quadratic program (QP) to minimize the task objective, unconstrained by contact
2) Projection of the solution onto the LCP constraints
3) Dual variable update based on the ADMM algorithm

Across each timestep of the control horizon, a parallel ADMM process is run with dynamics continuity between parallel processes being enforced in the QP step. This decoupling of solutions across time allows for C3 to run at realtime rates despite the rate-limiting projection step, which scales with number of contact pairs.

III. Evaluation

A. Atomic Task Framework

One possible approach to comparing these algorithms is to implement each controller on a range of complex tasks, and compare performance between them to draw broader conclusions. In our attempts at this approach, we found that controlling contact-rich tasks could fail for many possible reasons depending on the specifics of the scene and the algorithm: it was not always clear if failures were caused by inherent weaknesses in the controller or by poor tuning, especially as tasks became more complex.

In an effort to gain more control over the experimental setup, we instead approach this study with the hypothesis that the task of effective contact can be decomposed into a series of transitions between contact modes. A robot must decide at each timestep, for each possible contact, whether it is in contact vs. out of contact and whether it is in the friction cone (sticking) vs. on the boundary of the friction cone (sliding). We define *atomic tasks* which each encapsulate one such decision and study relationship between algorithm hyperparameters and the capacity of the controller transition contact modes when necessary. In the following sections we introduce each task.

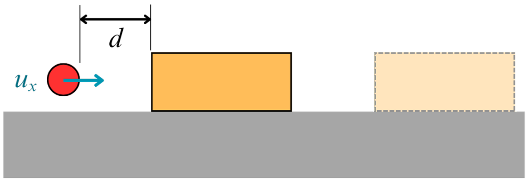


Fig. 1: Make/break task.

1) Make/Break Contact: This task studies the transition from out-of-contact to in-contact. A 1D spherical end-effector is allowed to travel on a virtual prismatic joint in the X direction. The robot must approach and contact a block on the ground, which must be pushed at a constant velocity in the +X direction. The task parameter for this transition is the starting distance between the end-effector and the block surface.

2) Stick/Slide: This task studies the transitions between sticking frictional contact and sliding frictional contact. A 2D spherical end-effector can travel in X and Z and must make contact with a block on its top face to slide it across the ground in the +X direction. To succeed, the end-effector should be in sticking contact on the top face and sliding contact between the block and the ground. This requirement leads to a band of successful inputs whose width is determined by the friction coefficients between the block and the end-effector and between the block and the ground. The difference in friction coefficients, with the block-ground coefficient remaining fixed, is the task parameter for this transition.

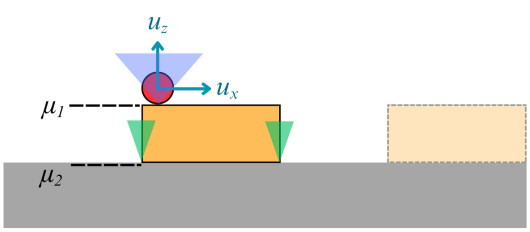


Fig. 2: Stick/slip task.

3) Reposition Contact: This task studies the transition from in-contact to out-of-contact for the purpose of repositioning a contact point on the same surface. This type of solution can be thought of as "gaiting" and is required for in-hand reorientation, some formulations of locomotion, and other tasks where incremental progress is required because the range of motion needed for success is limited by the robot's hardware.

A 2D spherical end-effector can once again travel in X and Z and must make contact with a block on its top face to slide it across the ground in the +X direction, similar to the stick/slide task. However, the robot's range of motion is constrained by two walls which allow the block but not the robot to pass underneath. The task parameter here is the distance between the two constraint walls.

4) Core Hyperparameters: We argue that the capacity of each controller to make contact transitions is dependent on a specific hyperparameter based on the nature of the algorithm. These hyperparameters are summarized in Table I.

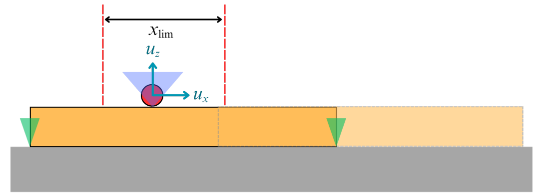


Fig. 3: Repositioning contact task.

Controller	Core Hyperparameter
Sampling: MJPC-PS	sample variance
Diff. contact: MJPC-iLQR	MJC contact softness
Disparity: C3	consensus parameters G & U

TABLE I: Core hyperparameter varied for each controller algorithm.

- For MJPC Predictive Sampling, contact transitions are governed by the numerics of sampling, and are influenced by sample variance compared to the relative width of the band of successful inputs. This was done by modifying the `sampling_exploration` hyperparameter, which controls the variance of samples.
- For MJPC iLQR, contact transitions are controlled by the relaxation parameter used for calculating contact gradients. This was done by modifying MuJoCo's `solimp` vector for contact solution impedance.
- For C3, lower relative weights on the projection step help lead to more out-of-mode contact force solutions during the QP step. This was done by varying the ratio of `g` and `u` consensus terms on λ.

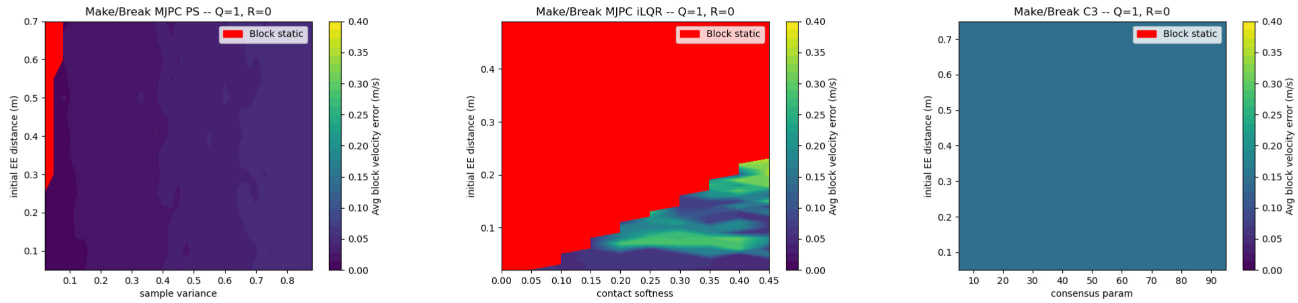
B. Testbed Setup

Each controller was evaluated with as much of the original code implementation as possible. To achieve this, a modular controller and simulation system was used, where each controller sends commands and receives states from a common simulator (Drake) by message-passing over LCM [5].

IV. Experiments

Each controller was tuned to a standard of reasonable performance for each task before experiments were run. Due to the individual tuning of each controller, we show here the equivalent per-time "cost" that would be accrued by a unit weight on state error, rather than the raw cost terms minimized internally by each controller. We found this to be the clearest way to understand the performance trends across the experiments.

The preliminary results for the make/break 1D taks are summarized in Figure 4, and in Figure 5 for the stick/slip 2D task. In each, a measure of task failure is masked in red, with average state tracking error of the block plotted across each controller hyperparameter and task parameter. Current findings are summarized in the captions of each, although more thorough analysis is necessary and ongoing.

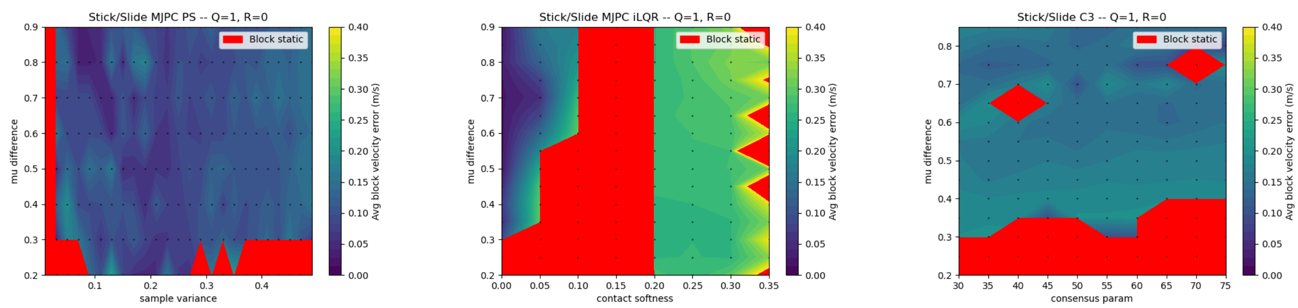


(a) MJPC Predictive Sampling. When provided a wide enough sample variance, controller is highly performant with low block tracking error variation. Performance variance within the successful parameter region is negligible.

(b) MJPC iLQR. There is an approximately linear relationship between contact softness and maximum distance of contact approach, represented here by the boundary between the static-block region and variable cost in the lower right. High velocity tracking error occurs at higher contact softness, due to artifacts of force-at-distance.

(c) C3. C3 is able to push the block forward across all trials, with a constant offset error in velocity of about 0.14 m/s regardless of approach distance or consensus. Further analysis and trials are necessary to understand the boundaries of contact-mode change as they relate to consensus.

Fig. 4: Average block velocity tracking error results across end-effector start position for the make / break task. Full failure is masked in red and represents failure of the end-effector in initiating approach to the block.



(a) MJPC Predictive Sampling. Controller is highly performant above μ difference of 0.3 and variance of 0.01.

(b) MJPC iLQR. Controller tracks block velocity at low contact softness. At higher softness, input causes both block and EE to slide, which is the incorrect combination of modes for full success.

(c) C3. C3 is able to push the block forward above a μ difference of 0.3-0.4, with average velocity error around 0.1 m/s. Further analysis is necessary to understand relationship to consensus.

Fig. 5: Average block velocity tracking error results across end-effector start position for the stick/slip task. Full failure masked in red represents the end-effector sliding off the block, causing < 0.01 m/s of motion at any instance across the trial.

V. Ongoing and Future Work

Work is in progress on experiments and analysis for the remaining tasks. We intend to perform a more complete analysis and clearer overview of each controller's performance. We also plan to implement the following controllers and tasks:

A. Controller: IDTO

We plan to implement a version of IDTO [7] based on the published code. This would be another differentiable-contact controller like MJPC-iLQR, where the core hyperparameter is the force-at-a-distance parameter for contacts.

B. Controller: ComFree-Sim

If time, we would also like to implement the MPPI controller presented in Borse et al's ComFree-Sim [2] as another sampling-based controller.

C. Task: Contact count scale

We plan to investigate the effect of scaling contact count with the problem size and control-loop time of each controller. In this task, a series of blocks are arranged in a row, and the robot is tasked with pushing all blocks forward to a goal region. The task parameter is number of blocks (and therefore number of contacts), with blocks able to contact each other on the table as well.

VI. Conclusion

In this work, we present a framework for comparison and analysis of contact-implicit MPC controllers by looking at atomic tasks which isolate different contact-mode transitions.

References

[1] Alp Aydinoglu et al. "Consensus Complementarity Control for Multicontact MPC". In: *IEEE Transactions on Robotics* 40 (2024), pp. 3879–3896. DOI: 10.1109/TRO.2024.3435423.

[2] Chetan Borse et al. *ComFree-Sim: A GPU-Parallelized Analytical Contact Physics Engine for Scalable Contact-Rich Robotics Simulation and Control*. 2026. arXiv: 2603.12185 [cs.RO]. URL: https://arxiv.org/abs/2603.12185.

[3] Iordanis Chatzinikolaidis and Zhibin Li. "Trajectory Optimization of Contact-Rich Motions Using Implicit Differential Dynamic Programming". In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 2626–2633. ISSN: 2377-3774. DOI: 10.1109/lra.2021.3061341. URL: http://dx.doi.org/10.1109/LRA.2021.3061341.

[4] Taylor Howell et al. *Predictive Sampling: Real-time Behaviour Synthesis with MuJoCo*. 2022. arXiv: 2212.00541 [cs.RO]. URL: https://arxiv.org/abs/2212.00541.

[5] Albert S. Huang, Edwin Olson, and David C. Moore. "LCM: Lightweight Communications and Marshalling". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 4057–4062. DOI: 10.1109/IROS.2010.5649358.

[6] Gijeong Kim et al. "Contact-implicit Model Predictive Control: Controlling diverse quadruped motions without pre-planned contact modes or trajectories". In: *The International Journal of Robotics Research* 44.3 (Oct. 2024), pp. 486–510. ISSN: 1741-3176. DOI: 10.1177/02783649241273645. URL: http://dx.doi.org/10.1177/02783649241273645.

[7] Vince Kurtz et al. "Inverse dynamics trajectory optimization for contact-implicit model predictive control". In: *The International Journal of Robotics Research* (2023), p. 02783649251344635.

[8] Simon Le Cleac'h et al. "Fast Contact-Implicit Model Predictive Control". In: *IEEE Transactions on Robotics* 40 (2024), pp. 1617–1629. DOI: 10.1109/TRO.2024.3351554.

[9] Albert H Li et al. "Drop: Dexterous reorientation via online planning". In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2025, pp. 14299–14306.

[10] Miquel Oller, Dmitry Berenson, and Nima Fazeli. "Tactile-Driven Non-Prehensile Object Manipulation via Extrinsic Contact Mode Control." In: *Robotics: Science and Systems*. 2024.

[11] HJ Terry Suh et al. "Dexterous contact-rich manipulation via the contact trust region". In: *The International Journal of Robotics Research* (2025), p. 02783649251398875.

[12] Russ Tedrake and the Drake Development Team. *Drake: Model-based design and verification for robotics*. 2019. URL: https://drake.mit.edu.

[13] Emanuel Todorov, Tom Erez, and Yuval Tassa. "Mujoco: A physics engine for model-based control". In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2012, pp. 5026–5033.

[14] Sharanya Venkatesh et al. "Approximating global contact-implicit mpc via sampling and local complementarity". In: *IEEE Robotics and Automation Letters* (2025).

[15] Patrick M Wensing et al. "Optimization-based control for dynamic legged robots". In: *IEEE Transactions on Robotics* 40 (2023), pp. 43–63.

[16] William Yang and Michael Posa. *Dynamic On-Palm Manipulation via Controlled Sliding*. 2024. arXiv: 2405.08731 [cs.RO]. URL: https://arxiv.org/abs/2405.08731.