A Comparative Analysis of Contact Models in Trajectory Optimization for Manipulation*

Aykut Özgün Önol, Philip Long, and Taşkın Padır[1]

Abstract— In this paper, we analyze the effects of contact models on contact-implicit trajectory optimization for manipulation. We consider three different approaches: (1) a contact model that is based on complementarity constraints, (2) a smooth contact model, and our proposed method (3) a variable smooth contact model. We compare these models in simulation in terms of physical accuracy, quality of motions, and computation time. In each case, the optimization process is initialized by setting all torque variables to zero, namely, without a meaningful initial guess. For simulations, we consider a pushing task with varying complexity for a 7 degrees-of-freedom robot arm. Our results demonstrate that the optimization based on the proposed variable smooth contact model provides a good trade-off between the physical fidelity and quality of motions at the cost of increased computation time.

I. INTRODUCTION

Discovering contact-rich motions for manipulation and locomotion tasks without specifying a contact schedule is a captivating idea. Thus, contact-implicit trajectory optimization attracted many researchers from fields such as robotics, computer graphics, and bio-mechanics. The main idea of this approach is to define an optimization problem with costs and constraints describing a task of interest so that a contact schedule and corresponding forces are found as a result of trajectory optimization.

In this method, selection of the contact model is crucial since it provides optimization with a way to infer contact dynamics. Complementarity constraints that describe rigid-body collisions have been extensively used for this purpose. A time-stepping scheme that uses complementarity constraints to model rigid-body dynamics with inelastic collisions and Coulomb friction was first proposed by Stewart & Trinkle [1]. Anitescu & Potra [2] modified this method to handle any configuration and any number of contacts. Based on this time-stepping scheme, Posa et al. [3] proposed a direct method for contact-implicit trajectory optimization to prevent the "combinatorial explosion" of hybrid models. Indeed, the idea of transcribing a non-smooth trajectory optimization problem with impacts and discontinuities into a nonlinear optimization problem with complementarity constraints was first introduced by Yunt & Glocker [4]. However, in [4], the optimization of control inputs is decoupled from the optimization of states and contact forces; whereas, in [3],

they are all optimized simultaneously. Nonetheless, in both methods, contact dynamics is modeled as a complementarity problem. [5] also solves a complementarity problem to find a contact schedule.

On the other hand, smooth contact models facilitate convergence of gradient-based solvers. For this reason, [6–9] use smoother fragments of complementarity constraints; whereas [10–12] directly define contact forces as smooth functions of distance. In [13] and [14], contacts are modeled within the cost function by using auxiliary variables. In addition, there are other contact modeling approaches that fall outside of this classification such as the ones in [15] and [16].

Although contact-implicit trajectory optimization is, by definition, independent of task and can be generalized to both locomotion and manipulation tasks, a significant part of the related literature focuses on locomotion tasks (e.g., [5], [7–9], [12], [16]). Nonetheless, for instance, [3], [13], [14], [17] investigate manipulation tasks but their analysis is either limited to planar case or based on animated characters (i.e., physical realism is not as critical as in robotics).

In manipulation tasks, distances between the robot and objects of interest can be quite large as compared to those in locomotion tasks (i.e., between feet and ground). Hence, smooth contact models are likely to be advantageous for manipulation since they would ease discovering contacts that are initially distant as well as improve convergence. However, existing smooth contact models sacrifice physical accuracy to achieve that. Thus, in this study, we propose a variable smooth contact model that allows the optimizer to adjust the smoothness of the contact model. Moreover, we analyze impacts of different contact models on the performance of contact-implicit trajectory optimization for manipulation. We compare a complementarity constraints-based contact model (CCCM), a smooth contact model (SCM), and the proposed variable smooth contact model (VSCM) for a non-prehensile manipulation task of pushing a cubic object on a tabletop by a Sawyer robot which has a 7 degrees-of-freedom (DOF) arm. We test all methods in simulation for different initial distances between the end effector and the object. The results suggest that the VSCM provides a good trade-off between the physical accuracy and quality of resulting motions at the cost of increased computation time.

The rest of the paper is organized as follows. In Section II, we describe the dynamic model of the system, the contact models, and the corresponding optimization problems. The results and their analyses are presented in Section III. Finally, concluding remarks and the future research directions are given in Section IV.

[1]RIVeR Lab, Northeastern University, Boston, MA, USA {onol.a, p.long, t.padir}@northeastern.edu

II. METHODOLOGY

A. Dynamic Model

The dynamics of an n DOF robot that is in contact with environment is given by:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \tau + \mathbf{J}_{ext}(\mathbf{q})^T \lambda_{ext}, \qquad (1)$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ are the joint positions, velocities, and accelerations, respectively; $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertial matrix; $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ represents the Coriolis, centrifugal, and gravitational terms; $\tau \in \mathbb{R}^n$ is the generalized joint forces; $\mathbf{J}_{ext}(\mathbf{q}) \in \mathbb{R}^{6n_{ext} \times n}$ is the Jacobian matrix mapping the joint velocities to the Cartesian velocities at the external contact points; and $\lambda_{ext} \in \mathbb{R}^{6n_{ext}}$ is the generalized contact forces at the contact points for n_{ext} external contacts. In this study, we use MuJoCo physics engine [18] to model the dynamics.

The vector of generalized joint forces can be decomposed as follows:

$$\tau = \mathbf{u} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{J}_c(\mathbf{q})^T \lambda_c(\gamma), \qquad (2)$$

where $\hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$ is an estimation of $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ and $\lambda_c \in \mathbb{R}^{6n_c}$ and $\mathbf{J}_c \in \mathbb{R}^{6n_c \times n}$ are respectively the generalized contact forces due to contact model and the corresponding Jacobian matrix for n_c contact candidates. $\mathbf{u}$ is the control input that is handled by the optimization. λ_c is calculated from the magnitude of normal contact force $\gamma \in \mathbb{R}^n_c$ that is determined by either optimization or contact mode. λ_c acts on the environment in addition to the forces due to the contact mechanics in MuJoCo (i.e., λ_{ext}); thus, it can be deemed a virtual force. As a result of this decomposition, the robot can compensate Coriolis, centrifugal, and gravitational effects as well as virtual forces when the control input is zero.

B. Contact Models

In this study, the following three contact models are investigated. Although the contact models only generate normal contact forces as only pushing manipulation is considered here, MuJoCo takes into account frictional forces in simulations.

1) Complementarity Constraints-based Contact Model: A rigid-body contact model can be formed by the following complementarity constraints:

$$\phi(\mathbf{q}) \geq \mathbf{0}, \qquad (3)$$

$$\gamma \geq \mathbf{0}, \qquad (4)$$

$$\gamma^T \phi(\mathbf{q}) = 0, \qquad (5)$$

where $\phi(\mathbf{q}) : \mathbb{R}^n \to \mathbb{R}^{n_c}$ is the signed distance function mapping the joint positions to the closest distance between bodies, i.e., (3) prevents interpenetration between bodies. $\gamma \in \mathbb{R}^{n_c}$ is the Lagrange multiplier that physically corresponds to the magnitude of the normal contact force, thus (4) ensures that bodies can only push each other. Equation (5), which is evaluated element-wise due to computational reasons ([3], [22]), guarantees that either distance or contact force must be zero; and these three constraints constitute a complementarity condition that can be denoted by $\mathbf{0} \leq \gamma \perp \phi(\mathbf{q}) \geq \mathbf{0}$.

Recently, Manchester & Kuindersma [9] proposed a variational contact-implicit trajectory optimization that is similar to [3]. However, in [9], equality constraints in complementarity conditions are relaxed through slack variables to improve the convergence, as in [23], by allowing positive contact forces to act from distance while also penalizing this term and thus minimizing deviation from strict rigid-body contact conditions. In this work, we use a similar approach and relax (5) through a slack variable $s \geq 0$ that determines the deviation from the strict rigid-body contact:

$$\gamma^T \phi(\mathbf{q}) \leq s. \qquad (6)$$

Consequently, (3), (4), and (6) form the complementarity constraints-based contact model that we use in this study.

2) Smooth Contact Model: The relaxation in the complementarity constraints provides a smoother contact model; however, using such a model is restrictive since it requires a constrained optimization algorithm. On the other hand, using directly a smooth contact model, such as those in [10–12], would eliminate this limitation. In this study, we use the following exponential formulation for the magnitude of the normal contact force, γ, that is based on the contact model proposed in [12] and analogous to a spring model:

$$\gamma(\mathbf{q}) = k \, e^{(-\alpha \phi(\mathbf{q}))}, \qquad (7)$$

where k is the spring stiffness and α determines the curvature of the contact force with respect to the distance, $\phi(\mathbf{q})$. We neglect the damping component in the original spring-damper model for the sake of simplicity. Figure 1 shows the variation of the magnitude of the normal contact force, γ, vs. the distance, $\phi(\mathbf{q})$, with negative distance being penetration into the contact surface.

3) Variable Smooth Contact Model: In the smooth contact model, a fixed value for each parameter is selected such that the convergence of gradient-based optimization is satisfactory. In other words, discovery of contacts is facilitated by allowing, albeit very small, contact forces to act from distance at the cost of loss in physical fidelity. As a consequence, there is a trade-off between the physical accuracy and the convergence of optimization. Thus, the parameters need to be tuned very sensitively based on the task. In order to mitigate this problem, we propose a variable soft contact model in which the spring stiffness k is a decision variable of optimization, and the curvature parameter is a function of k, i.e., $\alpha = c/k$ where c is a constant.

In Fig.1, the variation of γ vs. $\phi(\mathbf{q})$ curves with respect to k and c is shown. For the first three curves (indicated by red, yellow, and green), $c = 10^3$; and for the remaining curves, $c = 5 \times 10^3$. In this study, we used the greater value for c because the distance - contact force relation converges to a rigid-body contact model as c becomes larger, while it is more similar to a soft contact model for the smaller c. It is also noted that the contact force due to this contact model completely vanishes when $k = 0$.

Using small values for k is possible when tasks such as locomotion are considered because usually either there are existing contacts or the distances between the contact
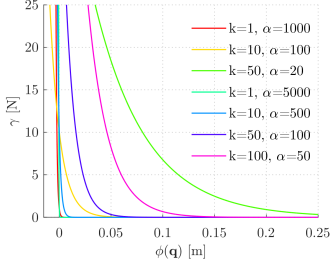
Fig. 1. Variation of the magnitude of the normal contact force with respect to the distance for various stiffness and curvature values.

candidates on the robot and the environment are small. Nevertheless, if one considers a manipulation task that requires, e.g., pushing an object that is far away from the robot, it is necessary to use a much larger k in order for the optimization to be able to discover such motions. However, a large k becomes problematic as the distance goes to zero since making an actual contact with dynamic objects becomes almost impossible. Thus, the VSCM allows the optimization to handle k such that the virtual force γ is minimized.

C. Contact-Implicit Trajectory Optimization

In the contact-implicit trajectory optimization, the main goal is to find contact forces along with control inputs given a high-level definition of the task in terms of an optimization problem. The optimal control problem of trajectory optimization (i.e., an infinite-dimensional problem) can be converted into a finite-dimensional problem through transcription methods which can be divided into three categories as single shooting, multiple shooting, and direct transcription [24]. In this study, we transcribe the problem into a single-shooting optimization problem by using evenly separated collocation points for the control inputs.

We consider the task of non-prehensile manipulation of pushing a cubic object on a tabletop by a 7 DOF robotic arm. The corresponding cost function can be written in terms of final costs and integrated costs. In this case, the final costs are based on the distance of the object's center of mass (CoM) from a desired position, $\mathbf{p}_o^e$, and the deviation of the object's orientation from desired Euler angles, θ_o^e; while the integrated costs are based on the end-effector velocity, $\dot{\mathbf{x}}(\mathbf{q}, \dot{\mathbf{q}})$, and the magnitude of the normal contact force, γ. Moreover, we normalize these cost terms such that $\mathbf{p}_o^e$ is divided by the norm of the initial position error, which yields $\mathbf{p}_{o,normalized}^e$, and the integrated costs are divided by N so that the optimization process is less sensitive to the task and the duration. As a result, the final and integrated components of the task-related cost (c^f and c^i, respectively) are calculated in terms of the weights $w_{1,...,4}$, the control sampling period t_c, the time step l, and the number of time steps N by:

$$c^f = w_1 ||\mathbf{p}_{o,normalized}^e||_2^2 + w_2||\theta_o^e||_2^2, \tag{8}$$

$$c^i = \frac{t_c}{N} \sum_{l=1}^{N} (w_3||\dot{\mathbf{x}}_l(\mathbf{q}_l, \dot{\mathbf{q}}_l)||_2^2 + w_4||\gamma_l||_2^2). \tag{9}$$

The following optimization problems are solved by a sequential quadratic programming (SQP) algorithm by running forward simulations in MuJoCo to evaluate cost and constraint functions.

1) Optimization Problem for CCCM: The optimization problem that includes the relaxed complementarity constraints is defined as:

$$\underset{\mathbf{u}_{1,...,N}, \gamma_{1,...,N}, s_{1,...,N}}{\text{minimize}} \quad c^f + c^i + w_5 \sum_{l=1}^{N} s_l^2 \tag{10}$$

subject to

$$\phi_l(\mathbf{q}), \gamma_l, s_l \geq \mathbf{0} \text{ for } l = 1, ..., N \tag{11}$$

$$\phi_l^T(\mathbf{q})\gamma_l \leq s_l \text{ for } l = 1, ..., N \tag{12}$$

For this problem, the weights are selected as $w_1 = 10^4$, $w_2 = 10^4$, $w_3 = 2 \times 10^{-1}$, $w_4 = 2 \times 10^{-1}$, and $w_6 = 10^3$.

2) Optimization Problem for Smooth Contact Models: The optimization problem takes the following unconstrained form for smooth contact models:

$$\underset{\mathbf{u}_{1,...,N}, (k_{1,...,N})}{\text{minimize}} \quad c^f + c^i. \tag{13}$$

The only difference is that the stiffness for each control time step k_l is a decision variable for the variable SCM. The weights for both problems are same and selected as $w_1 = 10^3$, $w_2 = 10^3$, $w_3 = 2 \times 10^{-1}$, and $w_4 = 2 \times 10^{-2}$.

III. SIMULATION EXPERIMENTS

A. Experimental Setup

The algorithms are tested in simulation for a simple scenario of pushing a cubic object on a tabletop with a Sawyer robot that has a 7 DOF arm. For simulations, we use MuJoCo physics engine since it is shown to be favorable for robotic systems with contacts [19]. Optimization problems are solved through an SQP-based solver called SNOPT [20]. Additionally, we use IFOPT [21] as an interface between MuJoCo and SNOPT.

In order to keep the complexity of the task low for the sake of comparison, we consider a simple pushing task in which the robot needs to push a cubic box 0.25 m along the *x*-axis of the world frame (see Fig. 2) without rotating it. Therefore, we take into account only two contact candidates: one on the object's closest surface to the robot, and one at the center of the end-effector plate of the robot. The simulation environment and the contact frames associated with the contact candidates on the robot and the object are depicted in Fig. 2. Nonetheless, it is noteworthy that the robot can make contact at arbitrary positions on its surface.

For all simulations, the duration is 1 s, and the control sampling period $t_c = 50$ ms, i.e., $N = 20$. The initial configuration of the arm is calculated by the following function of β that also determines the distance between the contact candidates $\mathbf{q}_0 = [-\pi/12, -\pi/\beta, 0, 3\pi/\beta, -\pi/6, -2\pi/\beta, 0]^T$.

We run simulations for three different values of β that are 5, 6, and 7, and the corresponding initial distances between the contact candidates are 0.30 m, 0.17 m, and 0.11 m, respectively. Figure 3 demonstrates the three initial
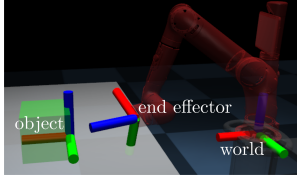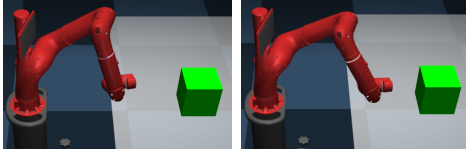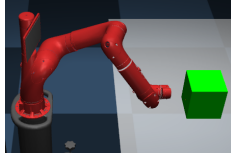
Fig. 2. Simulation environment along with the world frame and the frames associated with the contact candidates on the robot and the object.

(a) $\beta = 5$, $\phi_0 = 0.30$ m   (b) $\beta = 6$, $\phi_0 = 0.17$ m

(c) $\beta = 7$, $\phi_0 = 0.11$ m

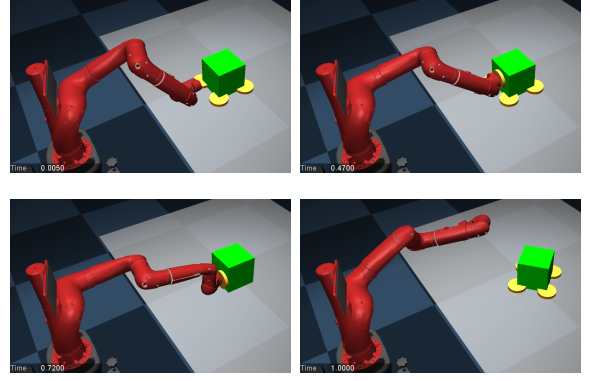Fig. 3. The initial configurations of the robot for different values of β.

Fig. 4. Snapshots of a resulting motion obtained for the CCCM for t = 5, 470, 720, and 1000 ms. Yellow cylinders represent active contacts.
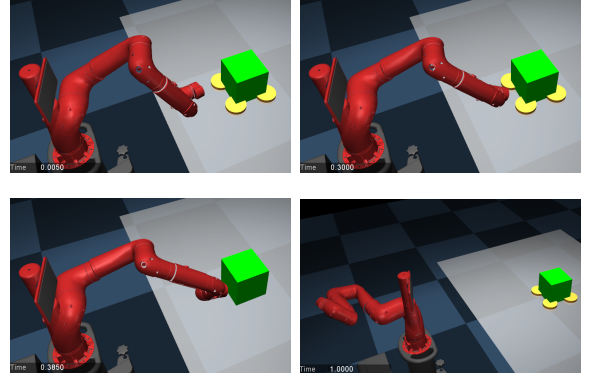

Fig. 5. Snapshots of a resulting motion obtained for the SCM for t = 5, 300, 385, and 1000 ms. Yellow cylinders represent active contacts.

configurations of the arm. It is seen that not only the initial position but also, albeit slightly, the orientation of the end effector vary depending on β. The purpose of different initial configurations is to change the complexity of the task in a controlled manner by assuming that difficulty in discovering contacts augments as the distance increases.

The stiffness parameter of the SCM is selected as 100 after a careful tuning considering the overall performance, and the corresponding α is calculated by $c = 5 \times 10^3/k$, as in the VSCM. We use the same value to initialize and upper bound k in the VSCM-based optimization (VSCMO). This value seems reasonable when one looks at the distance range here and the γ vs. $\phi(\mathbf{q})$ curves in Fig. 1. In all cases, the initial guess for u is zero, namely the robot is stationary throughout the simulation.

B. Visual Analysis

In order to observe the discrepancies between motions that are optimized by assuming different contact models, we visualize the resulting motions in the following.[1] First, a motion found by the CCCM-based optimization (CCCMO) is shown in Fig. 4. In this motion, the first contact between the robot and the object is made at t=470 ms. Then, the contact starts to slide in the $-y$ direction by causing an undesired rotation of the object in addition to the desired translation. After the contact is broken, the arm rises due to the impact.

Second, Fig. 5 illustrates a resulting motion of the SCM-based optimization (SCMO). The robot approaches the object, and the object is still stationary at t=300 ms. However,

[1] Please see the accompanying video for all simulation results: https://youtu.be/u06ZTmut0N0.

at t=385 ms, the object starts to move due to the force generated by the SCM, even though there is no actual contact between the robot and the object. At the end, the object is successfully manipulated with no visible rotation; yet the arm is repelled significantly due to the large γ values during pushing. The problem of pushing without an actual contact might be alleviated by a very sensitive tuning of k; however, an actual contact still would be very hard to achieve in conjunction with satisfactory motion and convergence. Moreover, repeated sensitive tuning is likely to be necessary even for slight changes in the task, such as the initial configuration of the robot.

Last, a motion obtained from the VSCMO is demonstrated in Fig. 6. Here, the object does not move until there is an active contact at t=715 ms. Then, the object is pushed successfully to the desired pose, while the arm is not repelled significantly and stops by making contact with the table. In this case, the arm actually contacts the object. This is possible owing to the adjustment of stiffness parameter k by the optimization.

In addition to these, it is clearly seen that the contact at t=715 ms occurs at neither of the contact candidates on the robot nor the object. This is possible due to the fact that the robot can make and break contacts at arbitrary parts of its surface by the virtue of MuJoCo physics engine.
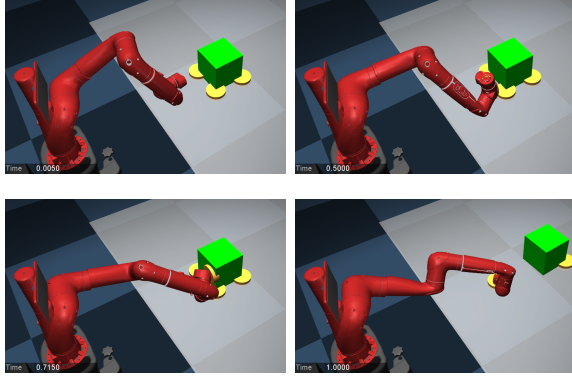
Fig. 6. Snapshots of a resulting motion obtained for the VSCM for t = 5, 500, 715, and 1000 ms. Yellow cylinders represent active contacts.

C. Numerical Analysis

After the visual analysis, we investigate the numerical results obtained from the simulations for all the contact models and the initial configurations in the following. First, we focus on the physical fidelity of the resulting motions. In order to measure the physical inaccuracy throughout a simulation, we use a metric that is calculated by integrating the magnitude of the normal contact force acting from distance, i.e., $t_c \sum_{k=1}^{N} \gamma_k$. Table I shows the values of this metric for all cases. It is seen that the CCCMO provides the best physical fidelity by far for the first two cases. Whereas, the VSCMO has the minimum physical inaccuracy in the last case, namely there is almost no virtual force. Moreover, it provides a much better physical accuracy over the SCMO in all cases.

Tables II and III demonstrate the final position and orientation errors for all cases, respectively. It is seen that in all cases, the VSCMO provides the best performance in terms of the manipulation task, when it is assumed that the differences between the orientation errors for the SCM and VSCM in the first two cases are negligible. Furthermore, it is the only method that can perform the task satisfactorily (i.e., with a position error smaller than 10 cm) for all initial configurations. On the other hand, the SCMO is also able to perform the task successfully except for the last case; while the CCCMO can find an *almost* successful motion only for the smallest initial distance, and even in that case, the orientation error is quite large.

The computation times for all cases are shown in Table IV. We admit that our implementation is significantly slower than the state-of-the-art counterparts such as [11] and, in particular, [12]. However, an optimized implementation of

the presented methods is out of scope of this work. Here, we are rather interested in comparing them in terms of the convergence speed. Based on these results, it is not possible to say that there is a clear relationship between the initial distance (or the complexity of the task) and the computation time; i.e., there is a positive correlation for the CCCM, a negative correlation for the SCM, and no correlation for the VSCM. However, it is fair to say that, excluding the second case, the SCMO converges significantly faster than the VSCMO and has the fastest convergence, as anticipated. On the other hand, the CCCMO has the slowest convergence speed except for the first case in which its computation time is slightly larger than SCM.

D. Variation of Stiffness for VSCMO

Finally, the k trajectories obtained from the VSCMO are depicted in Fig. 7 for all initial configurations.[2] It is seen that the optimization handles the stiffness k as if it is a binary variable; i.e., k is either around 100, its initial value and the upper bound, or equal to zero. The collocation points when it is equal to zero correspond to the instants of the simulation when the arm is approaching to the object. This is caused by the fact that the virtual force generated by the contact model, γ, that is proportional to k and inversely proportional to the distance, ϕ, is penalized. Consequently, in the VSCMO, contacts can be easily discovered from distance as in the SCMO as well as the robot can actually make contact with the object to physically move it, as distinct from the case for the SCM with fixed parameters (see Figs. 5 and 6). Moreover, this can be achieved for a wide range of tasks without a sensitive tuning of the contact model parameters.

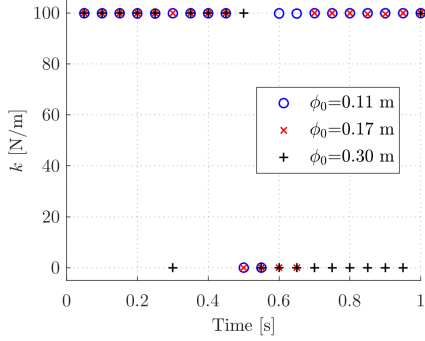[2]Please see the accompanying video for the virtual and actual contact force trajectories for all cases.

Fig. 7. k trajectories found by the VSCM-based optimization for the three values of the initial distance between the contact candidates.

IV. CONCLUSION

In this paper, we analyze the impact of contact models on the performance of contact-implicit trajectory optimization for manipulation. The trajectory planning method in question is a single shooting optimization. Three different contact modeling approaches are investigated: (1) a complementarity constraints-based contact model, (2) a smooth contact model, and (3) our proposed variable smooth contact model. We perform simulations of a 7 DOF Sawyer robot arm carrying out a pushing manipulation task of varying complexity.

An analysis of the resulting trajectories yields the following observations. First, all of the methods are capable of discovering a pushing motion given zero initial values for the torque variables. Second, the proposed approach, VSCMO, is the most reliable as it finds satisfactory motions for all initial configurations. Third, the VSCMO provides the best performance when both the quality and the physical accuracy of the motions found are considered. Notwithstanding, its convergence is slower than the SCMO which outperforms the CCCMO in all cases. However, the SCMO is very sensitive to the parameters of the SCM, and therefore, it is hard to tune for a range of tasks. On the other hand, the VSCM allows the optimization to vary the smoothness of the contact model as necessary. Consequently, the VSCMO is more robust to changes in the task and, thus, more suitable for contact-implicit trajectory optimization for manipulation.

Nevertheless, the task we investigate here in this paper is limited to one contact candidate on both the robot and the object. We aim to test the proposed method on more complex tasks and behaviours. Furthermore, in future work, we aim to implement our proposed method experimentally.

REFERENCES

[1] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *Int. Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[2] M. Anitescu and F. A. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dynamics*, vol. 14, no. 3, pp. 231–247, 1997.

[3] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The Int. Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[4] K. Yunt and C. Glocker, "Trajectory optimization of mechanical hybrid systems using SUMT," in *9th IEEE Int. Workshop on Advanced Motion Control, 2006*. IEEE, 2006, pp. 665–671.

[5] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Hierarchical planning of dynamic movements without scheduled contact sequences," in *2016 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4636–4641.

[6] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 4906–4913.

[7] I. Mordatch, K. Lowrey, and E. Todorov, "Ensemble-CIO: Full-body dynamic motion planning that transfers to physical humanoids," in *2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5307–5314.

[8] M. Neunert, F. Farshidian, and J. Buchli, "Efficient whole-body trajectory optimization using contact constraint relaxation," in *2016 IEEE-RAS 16th Int. Conf. on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 43–48.

[9] Z. Manchester and S. Kuindersma, "Variational contact-implicit trajectory optimization," in *Proceedings of the Int. Symposium on Robotics Research (ISRR), Puerto Varas, Chile*, 2017.

[10] T. Marcucci, M. Gabiccini, and A. Artoni, "A two-stage trajectory optimization strategy for articulated bodies with unscheduled contact sequences," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 104–111, 2017.

[11] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1502–1509, 2017.

[12] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *arXiv preprint arXiv:1712.02889*, 2017.

[13] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 43, 2012.

[14] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*. Eurographics Association, 2012, pp. 137–144.

[15] S. Jain, Y. Ye, and C. K. Liu, "Optimization-based interactive motion synthesis," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 1, p. 10, 2009.

[16] D. Zimmermann, S. Coros, Y. Ye, R. W. Sumner, and M. Gross, "Hierarchical planning and control for complex motor tasks," in *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2015, pp. 73–81.

[17] M. Gabiccini, A. Artoni, G. Pannocchia, and J. Gillis, "A computational framework for environment-aware robotic manipulation planning," in *Robotics Research*. Springer, 2018, pp. 363–385.

[18] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 5026–5033.

[19] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX," in *2015 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4397–4404.

[20] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Rev.*, vol. 47, pp. 99–131, 2005.

[21] A. W. Winkler, D. C. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Systems through Phase-based End-Effector Parameterization." *IEEE Robotics and Automation Letters*, vol. 3, pp. 1560–1567, 2018.

[22] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes, "Local convergence of SQP methods for mathematical programs with equilibrium constraints," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 259–286, 2006.

[23] R. Fletcher and S. Leyffer, "Solving mathematical programs with complementarity constraints as nonlinear programs," *Optimization Methods and Software*, vol. 19, no. 1, pp. 15–40, 2004.

[24] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.