2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)
November 4-5, University of Cambridge, United Kingdom

Distributed Contact-Implicit Trajectory Optimization for Collaborative Manipulation

Ola Shorinwa[1] and Mac Schwager[2]

***Abstract*— We present a distributed method for contact-implicit trajectory optimization which enables a group of robots to collaboratively manipulate an object through contact. Each robot computes its torques and contact forces locally to manipulate the object while communicating with its neighbors over a wireless network. While the global collaborative manipulation problem consists of multiple sets of non-smooth contact dynamics constraints for all robots, each robot in our method considers only its own contact dynamics constraints without those of other robots in its trajectory optimization problem, enabling each robot to efficiently compute its forces and torques. Our approach does not require each robot to share its torques, objective functions, and local constraints with other robots. We demonstrate our method in the manipulation of a sliding rod along a surface with friction, as well as in the manipulation of an object falling under gravity by a group of quadrotors.**

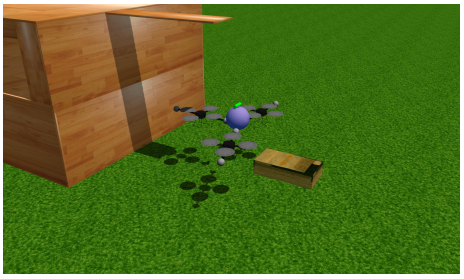

Fig. 1. A group of quadrotors manipulate an object falling under gravity after it has been dropped from a height. Our method enables each robot to compute its forces and torques locally by solving a contact-implicit trajectory optimization problem while communicating with its neighbors.

I. Introduction

In many scenarios, collaborative manipulation requires robots to make and break contact with the object they are manipulating, especially when the robots have to manipulate the object in constrained spaces, e.g., through doorways, windows, and hallways, or when the kinematic limitations of the robots require re-grasping. These situations also arise with multiple robots manipulating furniture in a house, carrying assemblies in a factory, or moving large objects in a warehouse. However, many distributed collaborative manipulation methods assume that the participating robots maintain a fixed grasp for the duration of the manipulation task, an assumption which limits the scope of possible manipulation tasks achievable by these methods. We present a distributed algorithm where the robots compute the trajectories and contact forces required to manipulate an object through contact.

Trajectory optimization involving contact proves particularly challenging due to the non-smooth contact constraints, known as complementarity constraints, arising within these problems. Previous works have attempted to reduce this difficulty by specifying the contact phases and contact points on the robot before solving the optimization problem. Although these approaches generally work well for less complex systems where the possible contact sequences can be specified, these approaches fail to scale to multiple collaborating robots, as the number of contacts grows exponentially with the number of independent contact points (and therefore the number of robots). Other approaches involve solving a contact-implicit trajectory optimization problem with the contact forces included as variables in the problem. These methods represent the contact dynamics using complementarity constraints. While these approaches allow for optimization over the contact sequences, solving the resulting problems often proves challenging due to the non-smooth complementarity constraints imposed at each time interval. Contact-implicit optimization problems for collaborative manipulation face an even greater challenge as each robot requires a separate complementarity constraint for its contact dynamics.

We derive our algorithm from the Alternating Direction Method of Multipliers (ADMM), exploiting a structural property of contact-implicit trajectory optimization to separate the components of the objective function and the problem constraints among the robots. Each robot iteratively (i) optimizes a local objective and (ii) communicates a candidate object trajectory with neighbors until all robots converge to a common optimized trajectory for the object. Each robot solves a contact-implicit trajectory optimization itself, only involving the robot's own contact forces, constraints, and dynamics, without explicitly considering other robots. This reduces the complexity of the underlying global optimization problem, while maintaining the same constraint satisfaction and local optimality guarantees of a centralized approach. The object trajectories computed by each robot converge to the locally optimal solution of the non-convex contact-implicit trajectory optimization problem. Because the number of constraints and optimization variables for the trajectories and torques remains fixed per robot as more robots are added, our approach scales efficiently to large numbers

*This work was supported in part by DARPA YFA award D18AP00064 and NSF NRI awards 1830402 and 1925030.

[1]Ola Shorinwa is with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA shorinwa@stanford.edu

[2]Mac Schwager is with the Department of Aeronautics and Astronautics Engineering, Stanford University, Stanford, CA 94305, USA schwager@stanford.edu

2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS) | 978-1-6654-2926-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/MRS50823.2021.9620665

of collaborating robots. In contrast, centralized through-contact planners typically struggle with a large number of contact points. Moreover, in our method, each robot does not share its own trajectory, torques, objective function, dynamic models, and local constraints with other robots, reducing the communication bandwidth required by our method. Instead, each robot only communicates its computed trajectory for the object itself to its one-hop neighbors in a communication network.

We demonstrate our method in the manipulation of a rod by a group of robots where the robots slide the rod along a surface with friction to a specified desired position and orientation before bringing the rod to rest. In addition, we apply our approach in the manipulation of an object by a group of quadrotors as the object falls under gravity. The quadrotors manipulate the object through contact while supporting it and bring the object to rest at its desired position and orientation.

A. Contributions

Our contributions are as follows:

- We present D-COPT a distributed contact-implicit trajectory optimization method for collaborative manipulation which enables a group of robots to manipulate an object through contact, unlike previous distributed methods which require a fixed grasp of the object.
- In our approach, each robot computes its torques and contact forces locally from an optimization problem considering only its contact dynamics constraints without those of other robots, overcoming the numerical optimization challenges posed by the multiple non-smooth contact dynamics constraints in collaborative manipulation problems.
- Our method achieves a notably higher success rate in solving the global contact-implicit trajectory optimization problem, finding a solution to the problem even when centralized methods fail. In addition, the local optimization problems in our algorithm require shorter computation times compared to these centralized methods.
- Based on our knowledge, our method represents the first distributed approach for solving contact-implicit trajectory optimization problems.

II. Related Works

A. Collaborative Manipulation

Notable challenges arise in centralized approaches for collaborative manipulation from the need to communicate with a central agent for computation of all forces and torques required to manipulate an object [1–3]. Addressing these challenges, distributed approaches allow each robot to obtain its forces and torques without communication with a central agent. Leader-follower methods designate a single robot as a leader which computes the forces and torques for all the robots within the group. In some of these methods, the robots communicate their dynamics models and problem constraints such as the feasible forces and torques which can be applied by the robots to the leader [4]. However, communication of the problem information between the leader and followers makes these methods unsuitable for problems involving varied communication topology and dynamics models. Other leader-follower methods attempt to resolve these challenges by enabling the followers to infer the motion of the leader using force or impedance sensors without communicating with the leader. The followers apply forces and torques to complement the forces applied by the leader using feedback controllers [5], impedance controllers [6], and adaptive control [7]. In these methods, the leader manipulates the object without any consideration of the limitations of the followers, possibly resulting in infeasible trajectories for the followers. In addition, these approaches do not allow for trajectory planning and require the specification of a desired trajectory for the leader to manipulate the object by a human operator [8].

In caging approaches, the robots surround the object and move it to a desired location while keeping the object enclosed for the duration of the task using gradient-based control laws [9, 10]. These approaches introduce significant limitations on the range of possible manipulation tasks as the robots must enclose the object during the task. Other methods take an optimization approach to collaborative manipulation [11]. In the distributed methods, each robot solves an optimization problem for its forces and torques required to manipulate the object locally, enabling the robots to avoid collisions in their environments while considering the dynamics limitations of the robots [12, 13].

All the above methods for collaborative manipulation assume each robot grasps the object before the beginning of the task and maintains the grasp for the duration of the task. This assumption can be debilitating in manipulation within complex environments where the manipulation task can only be achieved through a variety of distinct contact interactions between the robots and the object. In these situations, the robots have to contact the object at distinct times to manipulate the object through narrow paths and around complex features within the environment. Requiring the robot to maintain its grasp for the entirety of the task effectively makes the manipulation task infeasible. We develop a distributed method without any of these limitations, enabling a varied set of contact interactions between the robots and the object. With our approach, each robot makes and breaks contact with the object as required to complete the manipulation task.

B. Trajectory Optimization with Contact

Trajectory optimization methods with contact planning can be classified into multi-phase approaches and contact-implicit approaches, depending on the procedure taken to incorporate the effects of the contact dynamics into the optimization problem. Multi-phase methods require the specification of the contact phases and contact points on the robot and introduce different constraints for each phase, reflecting the robot's dynamics within each phase [14–16]. These methods do not

optimize over the possible contact points on the robot. To allow for optimization over the contact points on the robot and the corresponding surfaces, some methods solve a mixed-integer convex program [17] with additional binary variables indicating the status of a contact point and its assignment to a contact surface. However, these methods require the specification of the possible convex contact surfaces in the environment which might not be readily available. Moreover, mixed-integer optimization presents notable computation challenges. To reduce the computation difficulty, some of these methods introduce the contact dynamics constraints into the objective function, allowing for violations of the constraints. In general, these approaches are only feasible for systems with relatively simple dynamics where the contact phases and contact points can be specified.

In contact-implicit approaches, the contact forces are incorporated as variables into the optimization problem with an associated constraint describing the contact model. These methods introduce the contact dynamics constraints at every time instance within the optimization, and hence do not require specification of the contact phases. Many approaches represent the contact dynamics as a set of linear complementarity constraints [18–21], representing the existence of contact forces only when the two surfaces have a non-zero gap. This approach for modeling the dynamics involving contact has been employed in simulation and trajectory optimization with realistic results. However, the non-smooth contact dynamics makes numerical optimization notably challenging. Some other approaches consider a stochastic representation of the complementarity constraints to account for uncertainties in the physical properties of the colliding surfaces [22].

To reduce the computation difficulty, some methods utilize a smooth contact model with virtual forces at contact points, removing the discontinuities in the contact force, and solve the resulting problem using trust-region based sequential convex programming [23] or an iterative Linear-Quadratic Regulator (iLQR) approach [24, 25]. Other methods take a bilevel optimization approach, where the upper-level problem involves computing the optimal trajectory of the problem which depend on the contact forces computed in the lower-level problem [26, 27]. However, these approaches do not preclude the existence of contact forces at a distance and penetration of the colliding surfaces, producing unrealistic results.

We model the contact dynamics between the robots and the object using linear complementarity constraints. By distributing the complementarity constraints among the robots, each robot solves a contact-implicit trajectory optimization problem considering only its own contact dynamics, overcoming the challenges to numerical optimization posed by the non-smooth contact constraints. With this approach, our method does not require any explicit specification of the discrete contact phases between each robot and the object.

Our paper is organized as follows: in Section III, we formulate collaborative manipulation as a contact-implicit trajectory optimization problem, introducing the contact constraints, dynamics models, and local constraint functions. We derive our distributed method D-COPT which enables each robot to compute its torques and contact forces locally in Section IV and demonstrate our method in the manipulation of a sliding rod on a surface with friction by a group of robots in Section V. We further apply our method in the manipulation of an object by a group of quadrotors as it falls under gravity after it has been dropped from a height, showing our method requires about 15 times shorter computation times to solve each instance of the optimization problem. We conclude in Section VI.

III. Problem Formulation

We consider a manipulation problem where N robots manipulate an object to a desired position and orientation. The robots begin from arbitrary locations at the beginning of the manipulation task and apply normal and friction forces on the object through contact. We assume the inertial properties of the object and robots are known, in addition to the coefficient of friction for tasks involving sliding an object along a surface. Robot i applies a contact force $f_i \in \mathbb{R}^{n_{f,i}}$ consisting of a normal component c_i and j tangential components $\{\alpha_{i,k},\ k = 1, \cdots, j\}$.

A non-zero contact force exists when the distance between a robot and the object equals zero, which we represent as the constraint

$$c_i \cdot \beta_i(x_i, x_{\text{obj}}) = 0 \tag{1}$$

where $c_i \in \mathbb{R}$ represents the magnitude of the normal contact force with $c_i \geq 0$ and $\beta_i(x_i, x_{\text{obj}})$ represents the distance function between robot i and the object with $x_i \in \mathbb{R}^{n_i}$ representing robot i's configuration and $x_{\text{obj}} \in \mathbb{R}^b$ representing the object's configuration. The normal component of the contact force $f_{i,n}$ acts in the direction of the gradient of the distance function with respect to the object's configuration given by

$$f_{i,n} = c_i \cdot \frac{\nabla \beta_i(x_i, x_{\text{obj}})}{\|\nabla \beta_i(x_i, x_{\text{obj}})\|_2} \tag{2}$$

with $\nabla \beta_i(x_i, x_{\text{obj}}) \in \mathbb{R}^b$.

The relationships between the contact force and the distance function between robot i and the object are represented by the complementarity constraints

$$\begin{aligned} &c_i \cdot \beta_i(x_i, x_{\text{obj}}) = 0 \\ &\beta_i(x_i, x_{\text{obj}}) \geq 0, \quad c_i \geq 0 \end{aligned} \tag{3}$$

indicating the existence of non-zero contact forces only when robot i makes contact with the object. We assume the object is manipulated by underactuated robots, where each robot applies joint torques τ to actuate a subset of its joints.

By representing the combined configuration of the robots and the object as $\boldsymbol{x} = [x_{\text{obj}}, x_i, i = 1, \cdots, N]$ with $\boldsymbol{x} \in \mathbb{R}^{n_s}$, the combined torques applied by the robots as $\boldsymbol{\tau} = [\tau_i, i = 1, \cdots, N]$ with $\boldsymbol{\tau} \in \mathbb{R}^{n_\tau}$, and the combined contact forces as $\boldsymbol{f} = [f_i, i = 1, \cdots, N]$ with $\boldsymbol{f} \in \mathbb{R}^{n_f}$, the combined dynamics of the robots and the object is described by

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{g}(\dot{\boldsymbol{x}}, \boldsymbol{x}) = \boldsymbol{V}\boldsymbol{\tau} + \boldsymbol{W}^{\mathsf{T}}\boldsymbol{f} \tag{4}$$

where $\boldsymbol{M} \in \mathbb{R}^{n_s \times n_s}$ represents the combined inertia of the robots and object, $\boldsymbol{g}(\cdot)$ represents the non-linear friction, Coriolis, and gravity terms influencing the dynamics of the robots and object, and $\boldsymbol{W} \in \mathbb{R}^{n_s \times n_f}$ represents the Jacobian for the contact forces in the dynamics. The robots apply the torque $\boldsymbol{\tau}$, and $\boldsymbol{V} \in \mathbb{R}^{n_s \times n_\tau}$ distributes the torques in $\boldsymbol{\tau}$ to the relevant components of the combined dynamics relationship. The dynamics model in (4) applies equally to fully-actuated robots.

A. Friction Constraints

We assume the tangential forces generated by contact between the robot and the object remain within the friction cone. To represent this constraint, we use the Coulomb friction model with

$$\|\alpha_i\|_2 \leq \mu c_i \tag{5}$$

where α_i represents the tangential components of the contact force between robot i and the object, μ represents the coefficient of friction at the surface of contact, and c_i represents the normal component of the contact force.

B. Collision Avoidance

To manipulate the object, we desire contact between the robots and the object while avoiding collisions (unwanted contact) between the robots, differentiating our problem from other collaborative manipulation scenarios in which robots are rigidly attached to the object. We incorporate these constraints into the problem as

$$\psi(x_i, x_j) \geq d_{i,j} \tag{6}$$

using a collision function $\psi(x_i, x_j)$ specifying the distance between robots i and j with the minimum safe distance between the robots represented by $d_{i,j}$.

C. Optimization Problem

With this formulation of the contact dynamics, we compute the torques required by the robots to manipulate the object by solving the problem

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{\tau},\boldsymbol{f}}{\text{minimize}} \;& \sum_{i=1}^{N} \int_0^T \phi_i(x_i, \tau_i, \boldsymbol{f}) \; \mathrm{dt} \\ \text{subject to } & \boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{g}(\dot{\boldsymbol{x}}, \boldsymbol{x}) = \boldsymbol{V}\boldsymbol{\tau} + \boldsymbol{W}^\mathsf{T}\boldsymbol{f} \\ & \boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = 0 \\ & \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{\tau}, \boldsymbol{f}) \leq 0 \end{aligned} \tag{7}$$

where $\boldsymbol{h}(\cdot)$ represents constraints on the initial and desired configuration and velocities of the robots and object and $\boldsymbol{r}(\cdot)$ represents the contact dynamics constraints described by (2), (3), and (5) in addition to the collision avoidance constraint in (6) and constraints on the torques applied by each robot. The objective function $\phi_i(\cdot)$ may depend on time t, although we do not indicate its dependence explicitly in the problem. In many trajectory optimization problems, $\phi_i(\cdot)$ takes a quadratic form, including terms representing the energy expended by the actuators on the robots and the deviation of the configuration of the robots from desired values.

Communication Network

We represent the communication network among the robots as an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, \cdots, N\}$ representing the robots, and an edge (i, j) exists in $\mathcal{E}$ if robots i and j share a communication link. We denote the neighbor set of robot i as $\mathcal{N}_i$, describing the set of robots that can communicate with robot i.

IV. Distributed Trajectory Optimization

To compute the torques required by the robots, we discretize the continuous-time optimization problem in (7) for transcription to a numerical optimization problem using a time interval of Δt seconds and select its value based on the manipulation task. We utilize a receding-horizon approach in solving the numerical optimization problem where we solve the problem over a smaller time span of duration T_r. The robots apply the first set of torques from the resulting solution. Subsequently, we advance the time span by one interval which defines the next optimization problem. We repeat this process until the completion of the manipulation task with each numerical optimization given by

$$\begin{aligned} \mathcal{P}_C: \quad \underset{\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{\tau},\boldsymbol{f}}{\text{minimize}} \;& \sum_{i=1}^{N} \ell_i(x_i, \tau_i, \boldsymbol{f}) \\ \text{subject to } & \boldsymbol{m}(\dot{\boldsymbol{x}}) + \boldsymbol{g}(\dot{\boldsymbol{x}}, \boldsymbol{x}) = \boldsymbol{V}\boldsymbol{\tau} + \boldsymbol{W}^\mathsf{T}\boldsymbol{f} \\ & \boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = 0 \\ & \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{\tau}, \boldsymbol{f}) \leq 0 \end{aligned} \tag{8}$$

with

$$\ell_i(x_i, \tau_i, \boldsymbol{f}) = \sum_{s=0}^{T_r} \phi_i(x_i, \tau_i, \boldsymbol{f}, s) \tag{9}$$

where $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ represent the configuration of the robots and objects and their corresponding velocities concatenated over all time intervals, and $\ell_i(\cdot)$ represents the objective function of robot i (including the sum over all time intervals). We discretize the dynamics model in (4) to obtain the discrete-time dynamics model in (8), where $\boldsymbol{m}(\dot{\boldsymbol{x}})$ represents the inertial term which depends on the mass matrix and the joint accelerations. For notational convenience, we write the resulting dynamics model as a single vector expression applying to all time intervals, rather than breaking out each time step separately. The problem $\mathcal{P}_C$ consists of the objective function and local constraints of all robots, which is unavailable to any single robot, necessitating centralized computation of the required torques at a station with access to all the problem information.

After discretizing the optimization problem in (7), some previous approaches apply the complementarity constraint between the contact forces and the distance function at the beginning of each time interval. This approach allows for the existence of non-zero contact forces even when contact between the robot and the object breaks away within the interval. To avoid this issue, other approaches enforce this constraint at the end of the interval which ensures that contact forces only exist if the robot and object maintain contact at the end of the time interval. However, this approach only

provides a partial remedy as non-zero contact forces can still exist within an interval when the robot does not remain in contact with the object as long as the robot contacts the object at the end of the interval. To preclude this situation, we introduce a constraint on the contact forces and the relative velocity between robot i and the object $\dot{\beta}_i(x_i, x_{\text{obj}})$ given by

$$c_i \cdot (\dot{\beta}_i(t + \Delta t) + \lambda \dot{\beta}_i(t)) = 0 \tag{10}$$

where λ represents the coefficient of restitution. For the manipulation tasks in this work, we focus on inelastic collisions between the robot and the object, described by the coefficient of restitution $\lambda = 0$. This constraint ensures that non-zero contact forces exist between the robot and the object only when the robot remains in contact with the object over the time interval, indicated by zero relative velocity between the robot and the object over this time interval.

We develop a distributed method D-COPT where each robot computes its torques without computing the torques of other robots, while collaboratively manipulating an object. The combined dynamics of the robots and object in (8) introduces coupling between the robots and object. For distributed computation of the torques in (8), we decompose the combined dynamics into the individual dynamics models for each robot and the object, resulting in the set of dynamics models

$$\begin{aligned} m_i(\dot{x}_i) + g_i(\dot{x}_i, x_i) &= V_i \tau_i + W_i^{\mathsf{T}} f_i \\ m_b(\dot{x}_{\text{obj}}) + g_b(\dot{x}_{\text{obj}}, x_{\text{obj}}) &= \boldsymbol{W}^{\mathsf{T}} \boldsymbol{f} \end{aligned} \tag{11}$$

for robot i and the object. Note that the object dynamics includes the combined contact forces of all robots $\boldsymbol{f}$ with the combined Jacobian matrix $\boldsymbol{W}$. We introduce local variables for the contact forces applied by the robots and the object configuration and velocity, resulting in the distributed optimization problem

$$\begin{aligned} \mathcal{P}_D : \quad \underset{\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\tau}, \boldsymbol{f}}{\text{minimize}} \; & \sum_{i=1}^{N} \ell_i(x_i, \tau_i, \boldsymbol{f}_i) \\ \text{subject to } & \boldsymbol{v}_i(\breve{x}_i, \dot{\breve{x}}_i, \tau_i, \boldsymbol{f}_i) = 0 \quad \forall i \\ & h_i(\breve{x}_i, \dot{\breve{x}}_i) = 0 \quad \forall i \\ & r_i(\breve{x}_i, \dot{\breve{x}}_i, \tau_i, f_i) \leq 0 \quad \forall i \\ & \boldsymbol{f}_i = \boldsymbol{f}_j \quad \forall j \in \mathcal{N}_i, \; \forall i \\ & \mathring{x}_i = \mathring{x}_j \quad \forall j \in \mathcal{N}_i, \; \forall i \end{aligned} \tag{12}$$

where $\breve{x}_i = [x_{\text{obj}}, x_i]$ represents the configuration of robot i and the object concatenated over all times steps, $\dot{\breve{x}}_i = [\dot{x}_{\text{obj}}, \dot{x}_i]$ represents the corresponding velocities, $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ include the local variables of the object configuration and velocities, and $\boldsymbol{f}$ includes the local variables of the contact forces applied by the robots. The dynamics constraints in $\boldsymbol{v}_i(\cdot)$ include the dynamics model of robot i and the object in (11) without the dynamics models of the other robots. Consequently, each robot does not require the dynamics models of other robots. In addition, each robot maintains a single set of complementary constraints for its contact dynamics in $r_i(\cdot)$ which also includes constraints on its torques and collision-avoidance constraints. We represent the constraints on the initial and desired configuration and velocities of robot i and the object with $h_i(\cdot)$ and robot i's local variables of the object configuration and velocity as $\mathring{x}_i = [x_{\text{obj},i}, \dot{x}_{\text{obj},i}]$ with $\mathring{x}_i \in \mathbb{R}^{n_o}$.

Proposition 1. *The optimization problem $\mathcal{P}_D$ in* (12) *is equivalent to the optimization problem $\mathcal{P}_C$ in* (8) *with the same optimal solution and optimal objective value.*

Proof. All robots compute the same combined contact force from the equality constraints between $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ $\forall (i,j) \in \mathcal{E}$ in (12), noting that the communication graph $\mathcal{G}$ is connected. Likewise, all robots compute the same trajectory for the object from the equality constraints on $\mathring{x}$. Consequently, we can replace the local combined contact force along with the configuration and velocity of the object computed by each robot by common optimization variables $\tilde{\boldsymbol{f}}$, $\tilde{x}_{\text{obj}}$, and $\dot{\tilde{x}}_{\text{obj}}$ respectively. With these variables, the optimization problem in (12) has the same feasible set with the problem in (8), in addition to having the same objective function. As such, the optimization problems in (12) and (8) have the same optimal solution and optimal objective value. □

Although typical distributed optimization methods can be applied to solve the problem in (12), utilizing these methods will result in a worse computational efficiency compared to the centralized case. Rather, we apply the SOVA optimization method in [28] to derive D-COPT a distributed method for (12). Using SOVA, each robot only computes variables relevant to its local objective and constraint functions rather than computing the entire set of optimization variables, as done in other distributed approaches. Our approach proves particularly efficient in solving the problem in (12) considering the computation challenges introduced by the non-smooth contact dynamics constraints.

We distribute the problem variables among the robots based on the relevance of each variable to each robot. For example, each robot has no use of the torques applied by the other robots, making optimization over these variables unnecessary for each robot. Hence, robot i computes its configuration and the object configuration $\breve{x}_i$, the corresponding velocities $\dot{\breve{x}}_i$, torques τ_i, and its contact forces $\boldsymbol{f}_i$.

Upon distributing the problem variables among the robots, we derive update procedures for the optimization variables of each robot from the augmented Lagrangian of (12). The augmented Lagrangian $\mathcal{L}_a$ is given by

$$\begin{aligned} \mathcal{L}_a(\cdot) = \sum_{i=1}^{N} \Big(& \ell_i(x_i, \tau_i, \boldsymbol{f}_i) \\ & + \sum_{j \in \mathcal{N}_i} \phi_{ij}^{\mathsf{T}}(\boldsymbol{f}_i - a_{ij}) + \nu_{ij}^{\mathsf{T}}(\boldsymbol{f}_j - b_{ij}) \\ & + \sum_{j \in \mathcal{N}_i} \gamma_{ij}^{\mathsf{T}}(\mathring{x}_i - u_{ij}) + \eta_{ij}^{\mathsf{T}}(\mathring{x}_j - w_{ij}) \\ & + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \|\boldsymbol{f}_i - a_{ij}\|_2^2 + \|\boldsymbol{f}_j - b_{ij}\|_2^2 \\ & + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \|\mathring{x}_i - u_{ij}\|_2^2 + \|\mathring{x}_j - w_{ij}\|_2^2 \Big) \end{aligned} \tag{13}$$

with the dual variables $\phi_{ij} \in \mathbb{R}^{n_f}$, $\nu_{ij} \in \mathbb{R}^{n_f}$, $\gamma_{ij} \in \mathbb{R}^{n_o}$, and $\eta_{ij} \in \mathbb{R}^{n_o}$ for the equality constraints between the contact forces and the object configuration and velocity of neighboring robots. Each robot updates its problem variables as the minimizers of the augmented Lagrangian using its dual variables at the previous iteration before subsequently updating its dual variables. Robot i solves the local optimization problem

$$\begin{aligned} \mathcal{P}_L: \quad & \underset{\breve{x}_i, \dot{\breve{x}}_i, \tau_i, \boldsymbol{f}_i}{\text{minimize}} \ \mathcal{J}_i(x_i, \tau_i, \boldsymbol{f}_i) \\ & \text{subject to } \boldsymbol{v}_i(\breve{x}_i, \dot{\breve{x}}_i, \tau_i, \boldsymbol{f}_i) = 0 \\ & \qquad\qquad h_i(\breve{x}_i, \dot{\breve{x}}_i) = 0 \\ & \qquad\qquad r_i(\breve{x}_i, \dot{\breve{x}}_i, \tau_i, f_i) \leq 0 \end{aligned} \tag{14}$$

with

$$\begin{aligned} \mathcal{J}_i(\cdot) = \ell_i(x_i, \tau_i, \boldsymbol{f}_i) + p_i^{k\mathsf{T}} \boldsymbol{f}_i + q_i^{k\mathsf{T}} \mathring{x}_i \\ + \rho \sum_{j \in \mathcal{N}_i} \left\| \boldsymbol{f}_i - \frac{\boldsymbol{f}_i^k + \boldsymbol{f}_j^k}{2} \right\|_2^2 + \left\| \mathring{x}_i - \frac{\mathring{x}_i^k + \mathring{x}_j^k}{2} \right\|_2^2 \end{aligned} \tag{15}$$

to update its problem variables.

Notably, the local contact-implicit trajectory optimization problem ($\mathcal{P}_L$) solved by each robot in (14) consists of a single set of complementarity constraints for the non-smooth contact dynamics of the robot, reducing the numerical challenges to solving the global optimization problem ($\mathcal{P}_C$) in (8). This approach enables the robots to compute a solution to (14) even in cases when solving (7) proves particularly difficult. Moreover, each robot does not require the dynamics models, objective functions, and constraints of other robots in (14). Since the optimization variable computed by each robot in (14) does not change as the number of robots increases, our method scales efficiently to manipulation problems with large groups of collaborating robots.

Remark 1. *Although each robot considers only its local constraints in* (14)*, the resulting trajectories of the robots and object satisfy all the problem constraints in* (12) *since all the robots compute the same object trajectory and all the problem constraints are enforced by at least one robot.*

After computing its problem variables, robot i shares its local object trajectory $\mathring{x}_i^{k+1}$ and contact force $\boldsymbol{f}_i^{k+1}$ with its neighbors and updates its dual variables $p_i \in \mathbb{R}^{n_f}$ and $q_i \in \mathbb{R}^{n_o}$ using

$$\begin{aligned} p_i^{k+1} &= p_i^k + \rho \sum_{j \in \mathcal{N}_i} \left(\boldsymbol{f}_i^{k+1} - \boldsymbol{f}_j^{k+1} \right) \\ q_i^{k+1} &= q_i^k + \rho \sum_{j \in \mathcal{N}_i} \left(\mathring{x}_i^{k+1} - \mathring{x}_j^{k+1} \right) \end{aligned} \tag{16}$$

where the dual variables p_i and q_i represent combined dual variables for ϕ_{ij}, ν_{ij}, γ_{ij}, and η_{ij}. This follows the dual update format introduced in [28].

Theorem 1. *Assuming the objective function of* (12) *has Lipschitz continuous gradients, the trajectories of the robots and object converge to a locally optimal solution of* (12).

Proof. If the objective function of (12) has Lipschitz continuous gradients, the augmented Lagrangian $\mathcal{L}_a(\cdot)$ of (13) decreases monotonically at every iteration until convergence [29]. In addition, the sequence $\{\boldsymbol{x}^k, \dot{\boldsymbol{x}}^k, \boldsymbol{\tau}^k, \boldsymbol{f}^k, p^k, q^k\}$ converges to a stationary point $\{\boldsymbol{x}^\star, \dot{\boldsymbol{x}}^\star, \boldsymbol{\tau}^\star, \boldsymbol{f}^\star, p^\star, q^\star\}$ of $\mathcal{L}_a(\cdot)$, corresponding to a locally optimal solution of (12). □

Algorithm 1 outlines D-COPT our distributed contact-implicit trajectory optimization method for collaborative manipulation.

Algorithm 1: Distributed Contact-Implicit Trajectory Optimization (D-COPT)

do in parallel $i = 1, \cdots, N$
  $(\breve{x}_i, \dot{\breve{x}}_i, \tau_i, \boldsymbol{f}_i) \leftarrow$ *OptimizeTrajectory*$(\breve{x}_i(t), \dot{\breve{x}}_i(t))$
  $\tau_i(t) \leftarrow \tau_i(0)$
  Apply torque $\tau_i(t)$
while *manipulation task is in progress*;

Algorithm 2: OptimizeTrajectory$(\breve{x}_i(t), \dot{\breve{x}}_i(t))$

Initialization:
$k \leftarrow 0$
$(p_i, q_i)^0 \leftarrow 0$
$(\breve{x}_i, \dot{\breve{x}}_i, \tau_i, \boldsymbol{f}_i)^0 \leftarrow$ argminimize$\{$Problem (14)$\}$
do in parallel $i = 1, \cdots, N$
  $(\breve{x}_i, \dot{\breve{x}}_i, \tau_i, \boldsymbol{f}_i)^{k+1} \leftarrow$ Equation (14)
  $(p_i, q_i)^{k+1} \leftarrow$ Equation (16)
  $k \leftarrow k + 1$
while *not converged or stopping criterion is not met*;
return $(\breve{x}_i, \dot{\breve{x}}_i, \tau_i, \boldsymbol{f}_i)$

The robots compute their torques from the *OptimizeTrajectory* procedure to manipulate the object at each time instant, repeating the procedure as the manipulation task occurs over a receding horizon. Each robot does not share its trajectory and torques with other robots, which reduces the communication bandwidth required for implementation of our method. We initialize the succeeding problem using the optimal solution from the previous instance of the optimization problem which improves the convergence of the solution, especially in situations where the optimization problem changes slowly over time.

V. Simulations

We evaluate our distributed contact-implicit trajectory optimization method D-COPT in collaborative manipulation problems, comparing the performance of D-COPT to centralized contact-implicit trajectory optimization methods [19, 20]. We assume each robot can communicate with its neighbors within a communication radius and examine our method in the manipulation of a sliding rod along a surface with friction in addition to the manipulation of an object as it falls under gravity. In both cases, the robots bring the object to

rest at the end of the manipulation task. We solve the resulting contact-implicit optimization problems on a consumer-grade laptop with an Intel i7 processor using IPOPT [30], an interior-point optimization solver, and set the maximum number of iterations at 5000. In each optimization problem, we use the MA-57 linear solver within IPOPT.

A. Manipulating a Sliding Rod

We consider a manipulation task with N robots where the robots collaboratively manipulate a rod through contact by sliding it along a surface with friction. We represent the configuration of the rod as $x_{\text{obj}} \in \mathbb{R}^3$ consisting of the rod's position and orientation, its velocities as $\dot{x}_{\text{obj}} \in \mathbb{R}^3$, and the configuration of robot i as $x_i \in \mathbb{R}^{n_i}$. The robots manipulate the rod from its initial configuration to a desired position and orientation before bringing the rod to rest at the end of the manipulation task. We assume the rod slides on a surface with friction and constrain the tangential components of the contact force using (5). By representing the contact dynamics between each robot and the object using a set of complementarity constraints, the robots compute their torques from the local optimization problem ($\mathcal{P}_L$) in (14). We consider a quadratic objective function given by

$$\ell_i(x_i, \tau_i, \boldsymbol{f}) = \tau_i^{\mathsf{T}} H_i \tau_i \tag{17}$$

where $H_i \in \mathbb{R}^{n_{\tau,i} \times n_{\tau,i}}$ denotes a positive definite matrix. The objective function represents our desire for each robot to manipulate the object using minimal torques.

In Figure 2, a group of 4 robots (in red colors) manipulate a blue rod by sliding it along the surface to a desired position and orientation which is indicated by the gray rod. The robots move the rod through contact before bringing the rod to rest at the end of the manipulation task in Figure 2 (right). The attached video shows the group of robots manipulating the rod along the surface.

We examine the success rate of the interior-point solver in solving the contact-implicit optimization problems in the centralized method and D-COPT for the manipulation task with 4 robots. We discretize the optimization problem over 70 ms intervals, resulting in 1954 optimization variables in the centralized method, with 1186 variables per robot in D-COPT. A problem is solved successfully if the interior-point solver finds a solution within 5000 iterations. We keep the initialization the same across both methods and evaluate the success rate in 100 manipulation tasks. From Table I, the centralized method attains a 28% success rate, highlighting the difficulty in solving the global contact-implicit trajectory optimization problem ($\mathcal{P}_C$). In contrast, D-COPT achieves a higher success rate of 88%, resulting from the decomposition of the global problem into smaller local problems solved by each robot.

We provide the mean computation time along with the standard deviation using the centralized method and D-COPT in Table II with the number of communication rounds in Algorithm 2 set at 12. In D-COPT, each robot required a mean time of 0.111 seconds for each iteration of Algorithm 2, giving a total time of 1.331 seconds to solve the distributed optimization problem. Meanwhile, the centralized method took a mean computation time of 9.910 seconds to solve for the torques of all the robots, showing the difficulty in solving the global problem in (8) with all the non-smooth constraints present.

TABLE I

SUCCESS RATE IN SOLVING THE CONTACT-IMPLICIT OPTIMIZATION PROBLEM IN SECTION V-A

Method	Success Rate (%)
Centralized	28
D-COPT	88

TABLE II

COMPUTATION TIME FOR THE PROBLEM IN SECTION V-A

Method	Time (sec)
Centralized	7.320 ± 0.0385
D-COPT	1.331 ± 0.332

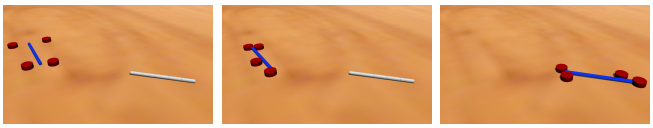

Fig. 2. A group of 4 robots (in red colors) manipulate a blue rod by sliding it along a surface with friction to a desired position and orientation indicated by the gray rod. The robots manipulate the rod through contact (center), before bringing it to rest at its desired position and orientation (right).

In Figure 3, we consider a task with 16 robots sliding the rod along a surface and show the normal and tangential components of the contact force applied by each robot. The magnitudes of the normal and tangential components of the contact force indicate the discrete contact interactions between each robot and the object, with non-zero contact forces existing only when the robot makes contact with the object.

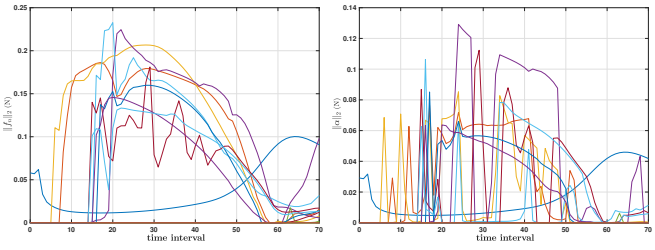

Fig. 3. The normal components (left) and tangential components (right) of the contact forces applied by each robot as 16 robots slide a rod to a desired position and orientation along a surface, showing the discrete contact interactions between each robot and the object as the object slides along its surface.

B. Manipulating an Object Falling Under Gravity

In this manipulation task, N robots manipulate an object falling under gravity, through contact, after it has been

dropped from an elevated platform, bringing the object to rest at a desired position and orientation.

We represent the orientation of the object at time t by a rotation matrix $R(t) \in \mathbb{R}^{3\times3}$ and its angular velocity by $\omega \in \mathbb{R}^3$. By applying an angular velocity ω on the object through contact over a time interval of length Δt, the object rotates through an angle $\phi \in \mathbb{R}$ around the rotation axis $u \in \mathbb{R}^3$. The change in the rotation matrix representing the orientation of the object is described by

$$R_{dt} = \cos(\phi)I + (1 - \cos(\phi))uu^{\mathsf{T}} + \sin(\phi)\hat{\omega} \tag{18}$$

where $I \in \mathbb{R}^{3\times3}$ represents the identity matrix and $\hat{\omega} \in \mathbb{R}^{3\times3}$ represents a skew-symmetric matrix derived from ω with $\phi = \|\omega \cdot \Delta t\|_2$ and $u = \frac{\omega \cdot \Delta t}{\phi}$. The resulting dynamics for the rotation matrix is given by

$$R(t+1) = R_{dt}R(t) \tag{19}$$

which remains a valid rotation matrix.

To compute the torques required to manipulate the object, each robot solves the local optimization problem in (14) where the configuration of the object $x_{\text{obj}}(t) \in \mathbb{R}^{12}$ includes its position and orientation (represented with a rotation matrix) and $\dot{x}_{\text{obj}}(t) \in \mathbb{R}^6$ includes its linear and angular velocities at time t with a single set of complementarity constraints for its contact dynamics. We specify the initial and desired configuration and velocities of robot i and the object in $h_i(\cdot)$.

In Figure 4, a group of 3 quadrotors manipulate an object as it falls under gravity after it has been dropped. The quadrotors contact the object to support and manipulate it to its desired orientation from Figure 4 to Figure 5, noticeable from the green cap on the object's top. The quadrotors bring the object to rest on the wooden box in Figure 5. The attached video shows the quadrotors manipulating the object. We discretize the problem over 70 ms intervals, resulting in 2430 optimization variables in the centralized method, with 2082 variables per robot in D-COPT. We provide the mean computation time and its standard deviation of D-COPT and the centralized method in Table III. In D-COPT, each robot took a mean time of 17.510 seconds to compute its torques, with each of the 12 iterations in Algorithm 2 requiring an average of 1.459 seconds. This optimization problem is significantly larger than the problem in Section V-A, therefore requiring more computation time. In contrast, the centralized method required 49.782 seconds to solve for the torques of all robots, and repeatedly fails to find a solution in many instances of the problem, as noted in Section V-A.

TABLE III
COMPUTATION TIME FOR THE PROBLEM IN SECTION V-B

Method	Time (sec)
Centralized	49.782 ± 0.559
D-COPT	17.510 ± 10.312

In Figure 6, we consider 12 robots manipulating the same object. The normal and tangential components of the contact forces highlight the discrete contact interactions between each robot and the object.

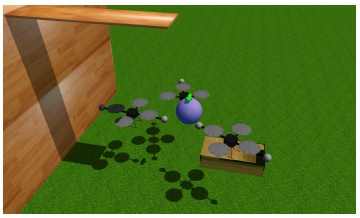

Fig. 4. A group of 3 quadrotors manipulate an object falling under gravity after it has been dropped to a desired position and orientation. The quadrotors approach the object to manipulate it before bringing it to rest on the wooden box.

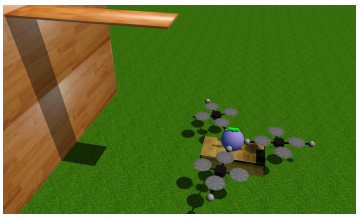

Fig. 5. The quadrotors manipulate the object and bring the object to rest at its desired position and orientation on the wooden box.

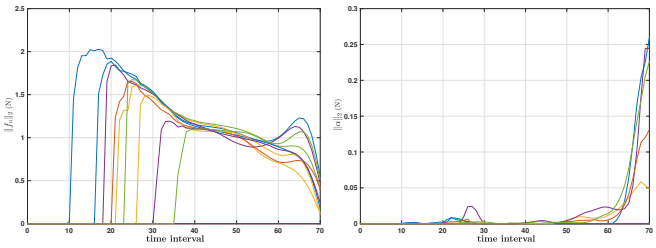

Fig. 6. The normal components (left) and tangential components (right) of the contact forces applied by each quadrotor on the object as 12 quadrotors manipulate an object falling under gravity. Each quadrotor makes discrete contact interactions with the object to manipulate the object.

VI. CONCLUSION

We present D-COPT a distributed contact-implicit trajectory optimization method enabling a group of robots to collaboratively manipulate an object through contact. By separating the objective functions and non-smooth dynamics constraints among the robots, each robot solves a local optimization problem only considering a single set of complementarity constraints for its contact dynamics, overcoming the notable numerical optimization challenges to solving the global contact-implicit trajectory optimization problem. Particularly, our method produces a solution when the global contact-implicit trajectory optimization problem proves difficult to solve. Each robot communicates with its neighbors to compute its torques and contact forces, without computing the torques and contact forces of other robots, allowing our method to scale efficiently to collaborative manipulation problems with large groups of robots. Further, the robots do not share their torques, objective functions, and other local constraints with other robots.

REFERENCES

[1] B. Hichri, L. Adouane, J.-C. Fauroux, Y. Mezouar, and I. Doroftei, "Cooperative mobile robot control architecture for lifting and transportation of any shape payload," in *Distributed Autonomous Robotic Systems*. Springer, 2016, pp. 177–191.

[2] A. Z. Bais, S. Erhart, L. Zaccarian, and S. Hirche, "Dynamic load distribution in cooperative manipulation tasks," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2380–2385.

[3] D. Ortenzi, R. Muthusamy, A. Freddi, A. Monteriù, and V. Kyrki, "Dual-arm cooperative manipulation under joint limit constraints," *Robotics and Autonomous Systems*, vol. 99, pp. 110–120, 2018.

[4] A. Petitti, A. Franchi, D. Di Paola, and A. Rizzo, "Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 441–446.

[5] Z. Wang and M. Schwager, "Force-amplifying n-robot transport system (force-ants) for cooperative planar manipulation without communication," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1564–1586, 2016.

[6] A. Marino and F. Pierri, "A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots," *Robotics and Autonomous Systems*, vol. 103, pp. 122–133, 2018.

[7] P. Culbertson and M. Schwager, "Decentralized adaptive control for collaborative manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 278–285.

[8] I. Mas and C. Kitts, "Object manipulation using cooperative mobile multi-robot systems," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2012.

[9] W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi, "Cooperative manipulation with least number of robots via robust caging," in *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2012, pp. 896–903.

[10] Y. Kobayashi and S. Hosoe, "Cooperative enclosing and grasping of an object by decentralized mobile robots using local observation," *International Journal of Social Robotics*, vol. 4, no. 1, pp. 19–32, 2012.

[11] T. Stouraitis, I. Chatzinikolaidis, M. Gienger, and S. Vijayakumar, "Online hybrid motion planning for dyadic collaborative manipulation via bilevel optimization," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1452–1471, 2020.

[12] C. K. Verginis, A. Nikou, and D. V. Dimarogonas, "Communication-based decentralized cooperative object transportation using nonlinear model predictive control," in *2018 European control conference (ECC)*. IEEE, 2018, pp. 733–738.

[13] O. Shorinwa and M. Schwager, "Scalable collaborative manipulation with distributed trajectory planning," in *2020 Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9108–9115.

[14] A. W. Winkler, C. D. 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, no. 3, pp. 1560–1567, 2018.

[15] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–8, 2012.

[16] J. Carpentier and N. Mansard, "Multicontact locomotion of legged robots," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, 2018.

[17] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2017.

[18] D. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with coulomb friction," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 162–169.

[19] J.-P. Sleiman, J. Carius, R. Grandia, M. Wermelinger, and M. Hutter, "Contact-implicit trajectory optimization for dynamic object manipulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6814–6821.

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

[21] Z. Zhou and Y. Zhao, "Accelerated admm based trajectory optimization for legged locomotion with coupled rigid body dynamics," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 5082–5089.

[22] L. Drnach and Y. Zhao, "Robust trajectory optimization over uncertain terrain with stochastic complementarity," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1168–1175, 2021.

[23] A. O. Onol, P. Long, and T. Padlr, "A comparative analysis of contact models in trajectory optimization for manipulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[24] 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.

[25] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization for legged robots with slipping motions," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3013–3020, 2019.

[26] ——, "Trajectory optimization with implicit hard contacts," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3316–3323, 2018.

[27] I. Chatzinikolaidis, Y. You, and Z. Li, "Contact-implicit trajectory optimization using an analytically solvable contact model for locomotion on variable ground," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6357–6364, 2020.

[28] O. Shorinwa, T. Halsted, and M. Schwager, "Scalable distributed optimization with separable variables in multi-agent networks," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3619–3626.

[29] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.

[30] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.