Real-Time Deformable-Contact-Aware Model Predictive Control for Force-Modulated Manipulation

Lasitha Wijayarathne[1], *Student Member, IEEE,* Ziyi Zhou[2], *Student Member, IEEE,*
Ye Zhao[2*], *Member, IEEE,* and Frank L. Hammond III[1*], *Member, IEEE*

Abstract*—Force modulation of robotic manipulators has been extensively studied for several decades. However, it is not yet commonly used in safety-critical applications due to a lack of accurate interaction contact modeling and weak performance guarantees - a large proportion of them concerning the modulation of interaction forces. This study presents a high-level framework for simultaneous trajectory optimization and force control of the interaction between the manipulator and soft environments, which is prone to external disturbances. Sliding friction and normal contact force are taken into account. The dynamics of the soft contact model and the manipulator dynamics are simultaneously incorporated in a trajectory optimizer to generate desired motion and force profiles. A constrained optimization framework based on Differential Dynamic Programming and Alternative Direction Method of Multipliers has been employed to efficiently generate real-time optimal control inputs and high-dimensional state trajectories in a Model Predictive Control Fashion. Experimental validation of the model performance is conducted on a soft substrate with known material properties using a cartesian space force control mode. Results show a comparison of ground truth and real-time model-based contact force and motion tracking for multiple Cartesian motions in the valid range of the friction model. It is shown that a contact model-based motion planner can compensate for frictional forces and motion disturbances satisfactorily and improve the overall motion and force tracking accuracy. The proposed high-level planning has the potential to be leveraged for medical tasks involving the manipulation of compliant, delicate, and deformable tissues in the task space.

I. Introduction

Robotics applications in the medical domain have gained increasing attention over the past few decades [1], [2]. Within the medical domain, planning and control of the interaction forces between a robot and its environment are essential to various safety-critical tasks. For instance, the interaction force should be modulated accurately in compliant environments, such as surgical settings, micro-assembly, or biological tissue manipulation. Furthermore, force control based on identifiable physical models is essential to identify instability modes (e.g., those caused by the bandwidth and system structure) and maintain reliable force interaction to guarantee safety. Thus, a model-based trajectory planning method with a high-fidelity contact model is essential for successful deployment with satisfactory motion and contact force tracking performance.

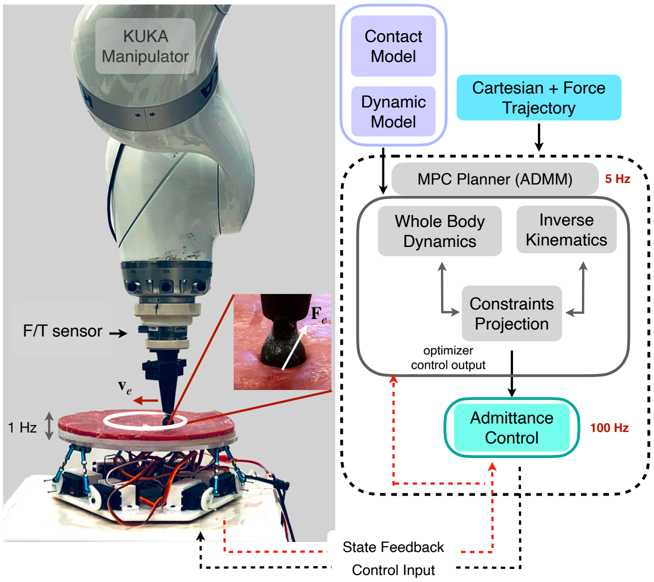


Fig. 1: KUKA manipulator experimentation setup. A manipulator performing a force controlled motion task on a soft surface.

Unlike rigid contact models, soft contact models are subject to challenges posed by non-linear material properties and non-uniformity as well as intensive computation burden due to numerical computation for solutions. Numerous contact models have been presented in the literature to model interactions involving elastic deformation [3], [4]. These models have broad applications and are essential in many engineering areas such as machine design, robotics, multi-body analysis, to name a few. For contact problems that involve elasticity, Hertz adhesive contact theory has been well established [5]. In this study, we focus on robotic tasks interacting with soft tissues, the contact behavior of which is determined by not only external and viscous forces, contact geometry, but also material properties (see Figure 1). Soft contact mechanics are crucial in physical model identification for motion planning applications in surgical robots.

Many robotic tasks require motion planning in the presence of contact in a constrained environment. Simultaneous trajectory generation and force control enable sophisticated manipulation tasks while interacting with complex objects. As a promising approach along this direction, trajectory optimization with contact models has been extensively investigated in

[1]Lasitha Wijayarathne and Frank L. Hammond III are with the Adaptive Robotic Manipulation Laboratory in the Woodruff School of Mechanical Engineering and the Coulter Department of Biomedical Engineering, Georgia Institute of Technology. frank.hammond@me.gatech.edu.

[2]Ziyi Zhou, and Ye Zhao are with The Laboratory of Intelligent Decision and Autonomous Robots at Woodruff School of Mechanical Engineering, Georgia Institute of Technology. ye.zhao@me.gatech.edu. (*co-corresponding authors)

the robotics community [6]–[13]. By incorporating the contact dynamics into the optimization, contact-dynamics-consistent motions can be planned for complex robot behaviors, such as dynamic locomotion or dexterous object manipulation. In this study, we propose a distributed optimization algorithm inspired by the locomotion community [14], [15], which efficiently solves a constrained trajectory optimization with a high-fidelity soft contact model [16]. Our previous work [16] demonstrated offline trajectory generation with contact and experimental validation. The algorithm used provides a general optimization framework to iteratively solve rigid body dynamics, soft contact interactions, articulated robot kinematics, and inequality constraints in a computationally efficient manner. In the current work, we extend it with modifications to the algorithm structure and demonstrate real-time execution in model predictive control (MPC) fashion.

Manipulator contact models are naturally framed and executed in the task space. In safety-critical tasks such as soft material manipulation and medical applications, force-torque control plays a significant role. These include interaction with humans in proximity or with direct physical contact. In approaching the contact interaction problem, data-driven techniques have been explored to learn the interaction between robotic manipulators and the environment [17], [18] or object [19], [20]. Unlike rigid contacts, the soft environment is prone to uncertainties spatially as well as with time. Thus, it is challenging to learn the contact model and robot dynamics simultaneously through data. In this study, we present a model for contact interaction and embed it into the high-level trajectory planning via enforced constraints.

Contact-rich environments tend to be prone to disturbances which are difficult to model or predict. In such cases, to maintain the contact and stability, an additional control layer will need to be deployed on top of the high level planner. In this work, we add a low-level admittance force controller [16] to handle uncertainites arising from the model as well as from disturbances in the environment. Finally, we implement the proposed algorithm in a MPC fashion to demonstrate the capability of real-time tracking.

The main contributions of this work are listed below:

- Presentation of a dynamic interaction model based on soft contact mechanics for a predefined geometry with Hertz visco-static theory.
- Incorporation of the interaction model into a constrained trajectory optimization to generate the desired cartesian path and force profile in an efficient, distributed fashion.
- Experimental validation of the derived contact dynamical model and real-time implementation of the proposed trajectory optimization algorithm with model predictive control (MPC).

A conference version of the work presented in this paper was published in [21]. The work presented here extends the previous work in three respects. We introduce a trajectory optimization from a distributed contact-aware distributed framework that splits the main problem into subproblems and solves them independently to find a consensus solution. In brief, the newly introduced inverse kinematics (IK) block is in charge of joint trajectory generation and the differential dynamic programming (DDP) block is for contact dynamics. Yet, the other block is for constraint handling. We also implement a low-level controller to aid the high-level controller in motion and force trajectory tracking. Furthermore, our experimental demonstrations are real-time executed in an MPC fashion. Finally, we experimentally demonstrate the efficacy of this framework on motion/force tracking tasks on a static and dynamic disturbance-induced platform.

II. Related Work

Contact Models. Elastic contact mechanics [5] have been extensively studied in various research fields where contact modeling is imperative for safety and performance requirements. Existing works in [4], [22]–[24] have used soft contact models for both modelling and control. These works include quasi-static assumptions and studies of [25], [26] explore cases where high-velocity impacts on soft material are considered. In the impact cases, visco-elastic models have been widely investigated. For instance, studies in [4], [25] compared various visco-elastic models with experimental validations. A majority of these works show that the Hertzian-based Hunt-Crossey model is the one most suitable for visco-elastic cases. Furthermore, fundamentals of frictional sliding motion are established in the works of [27], [28], where the main focus is on rigid body contacts but generalizable to soft contacts. More recent works in [29], [30] propose contact-area-based models.

Trajectory Optimization. Trajectory optimization (TO) is a powerful tool to generate reliable and intelligent robot motions. Various numerical methods have been proposed to solve a TO [31]–[33]. Among them, Differential Dynamic Programming (DDP) and iterative Linear Quadratic Regulator (iLQR) have aroused much attention in solving TO in the context of unconstrained problems, where only dynamics constraint is enforced in the forward-pass. The Ricatti-like backward pass in DDP or iLQR effectively reduces the complexity of solving an approximated LQR problem over the entire horizon, and the optimization is solved in an iterative fashion. In [34], DDP is used in a balancing task of a humanoid robot with high degrees of freedom (DoFs). A more recent work [35], demonstrates a Model Predictive Control (MPC) implementation based on DDP. However, standard DDP algorithms are not capable of addressing constraints. In [36], [37], DDP-type variants are proposed to cope with state and control constraints. Instead, our approach employs an augmented Lagrangian method named as Alternating Direction Methods of Multipliers (ADMM) [15], [38]–[41] to address various constraints. The ADMM framework is capable of tackling more constraints by introducing additional optimization blocks, making the algorithm suitable for parallel computing and utilizing the most efficient solver for each sub-problem.

Trajectory Optimization with Soft Contact. Contact-aware TO are often built upon the conservative assumption of rigid point contact dynamics [6]–[10] – an assumption present in the majority of robot control and planning literature. This assumption largely overlooks underlying patch contact dynamics such as surface deformation and elasticity [42]. Although there are some exceptions such as [11]–[13]

that directly integrate a soft contact model into the system dynamics, and implicitly optimize both contact force and other control inputs, the contact models are still relatively simple. In [43], a soft contact model was taken into account in the optimization formulation for whole-body locomotion control. However, most of the works above assumed spring-damper type soft contact models, which still largely mismatched the contact surface deformation or elasticity in reality. Therefore, advanced planning algorithms that accurately model complex contact dynamics are imperative to enable maneuvering over complex terrain or grasping irregular objects. To date, TO incorporating a high-fidelity deformable contact model remains under-explored in the field.

Model Predictive Control (MPC). The generation of trajectories for a given model and fixed horizon is computationally prohibitive in nature, making it difficult to deploy in real-time contact-rich applications, where model uncertainties and environmental disturbances are ubiquitous. Model Predictive Control (MPC) is a powerful strategy widely used to generate motion plans in real time and be adaptive to state changes due to environmental disturbances [44], [45]. Recent advances in fast automatic differentiation (AutoDiff) [46] and AutoDiff compatible rigid body models has enabled real time optimal control. The study in [47] showed a hardware implementation of MPC with a DDP optimizer framework at an update rate of 1000 Hz on a 7-DOF robot for a vision based point-to-point trajectory planning.

Admittance Control. To cope with un-modeled modalities of the contact, we use a low-level force controller based on admittance control [16] which has a fast control update rate compared to the high-level planner. It can compensate for the uncertainties that arise spatially over the surface (e.g., stiffness, slipperiness, and damping). Admittance control has been long studied and proven to work efficiently in compliant environments [48]. Furthermore, low-level controller mitigates instabilities [49] arises from the contact caused by the control update rate, stiffness mismatch, and high gains.

III. Deformable Contact Modeling

A. Contact modeling via Hertz's theory

In this section, we model the interaction dynamics between an application tool mounted on a manipulator and a soft tissue in terms of contact geometry and mechanics. In the example shown in this study, the contact part of manipulation is assumed to be a spherical indentation (for simplicity, but not limited to). Further, we assume that the application tool used is rigid and has a high stiffness compared to the contact surface. Along with these assumptions, we derive a dynamic model based on the contact friction theory and pressure distribution based contact modelling. According to Hertz's theory, the largest static indentation is achieved at the central point of the circle (see Figure 2) and can be expressed as:

$$d = \left[\frac{9F^2}{16E^2R}\right]^{\frac{1}{3}} \tag{1}$$

where E is the reduced Young's modulus of tool and surface, R is the radius of the tool end, F is the force imparted on the surface by manipulator end-effector. Combined Young's modulus of the tool and the soft contact surface material can be lumped to one term as:

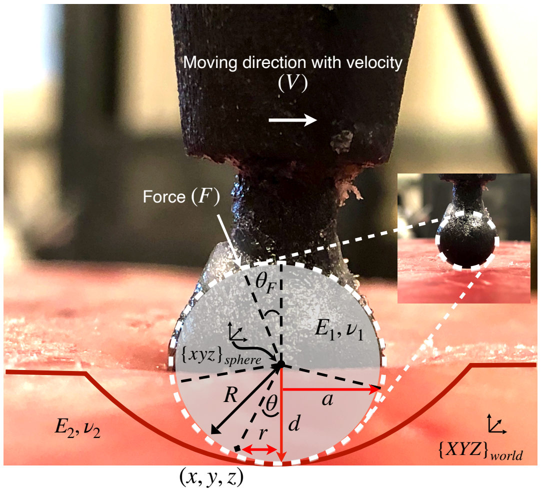


Fig. 2: A graphical illustration of the soft contact model between surface and the end-effector tool

$$\frac{1}{E} = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}$$

where E_1, E_2 and ν_1, ν_2 are Young's moduli and Poisson ratios of the end-effector and contact surface material, respectively. In our scenario, we assume the contact part as a rigid object and thus the Young's modulus of the spherical cap E_2 is approximated as infinity. Accordingly, we have $E = E_1/(1-\nu_1^2)$. The deformation and stress distributions on the surface are approximated by the universal Hooke's law and Hertz's theory. Details of normal, radical, and hoop (i.e., moving direction) stress distributions within the contact area in the cylindrical coordinate system are provided in the Appendix.

Accordingly, the deformation distribution is derived from the stress distribution equations as follows:

$$u_z = \begin{cases} \frac{3\pi}{8a}\left[\frac{1-\nu^2}{E}\right]p_m(2a^2-r^2), & (r \le a) \\ \frac{3}{4a}\left[\frac{1-\nu^2}{E}\right]p_m\Big[(2a^2-r^2)\sin^{-1}\left(\frac{a}{r}\right) \\ \quad +a(r^2-a^2)^{\frac{1}{2}}\Big], & (r \ge a) \end{cases}$$

where $p_m = F/(\pi a^2)$ is the average stress applied in contact part by manipulation and $a = \sqrt{Rd}$ is the radius of contact area (see Figure 2). The dynamic contact model for a contact spherical cap (i.e., spherical geometry) is applied with a force vector F at an angle θ_F to the perpendicular and moves in a circular path of radius R with a uniform velocity $\mathbf{v}_e$ in frame $\{\text{sphere}\}$. It represents the scenario of manipulating an application tool to work with soft tissues. For simplicity, our model focuses on sliding friction and ignores other frictional sources such as adhesion and rolling induced by deformation. Due to the symmetry of our contact scenario, σ_θ represents the principal stress within the contact circle. Thus, we can represent the stress tensor of any contact point (r, θ, z) in cylindrical coordinates relative to frame {sphere} via the Cauchy stress theory [5].

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_r & 0 & \sigma_{rz} \\ 0 & \sigma_\theta & 0 \\ \sigma_{zr} & 0 & \sigma_z \end{bmatrix} \tag{2}$$

Since the task is defined in the Cartesian frame, we convert parameters to Cartesian coordinates from cylindrical coordinates. The stress tensor in Cartesian coordinate is $\boldsymbol{\sigma}_c = T^T\boldsymbol{\sigma}T$, where the transformation matrix T is defined in the Appendix X-A.

At an arbitrary point on contact surface $(x, y, z)_{\{\text{sphere}\}}$, the normal vector from this point to centroid of spherical cap is $\mathbf{n} = [s\theta \quad 0 \quad c\theta]^T$.[1] Then, the normal stress of the contact surface is $\sigma_n = \mathbf{n}^T\boldsymbol{\sigma}_c\mathbf{n}$ with

$$\sigma_n = \sigma_r c^2\theta s^2\theta + \sigma_\theta s^4\theta + \sigma_z c^2\theta + 2\sigma_z s\theta c^2\theta$$

Given this stress expression, the overall friction force of the contact surface is represented as

$$df = \mu\sigma_n dS = \mu\sigma_n \times 2\pi r\frac{dr}{c\theta} \tag{3}$$

$$f = \int df c\theta = 2\pi\mu\int_0^a \sigma_n r dr \tag{4}$$

where, df, dr, dS are the differential elements of the friction, r and contact area. In the surface normal direction, it is assumed that the surface is in contact with the end point of the tool. As a result, Eq. (1) always holds. The derivative form of Eq. (1) is

$$\dot{z} = -\dot{d} = -\left[\frac{1}{6E^2RF_z}\right]^{\frac{1}{3}}\dot{F}_z \tag{5}$$

where z represents the position along the surface normal direction of the contact point and force along the normal direction is defined as $F_z = F\cos\theta_F$. In the moving direction, Eq. (4) and μF_z give the frictional force caused by the normal force F_z, which is:

$$\mathbf{F}_f = f = \mu F_z\left[1 + (2\nu - 1)\frac{3a^2}{10R^2}\right]\mathbf{n}_v + k_d\mathbf{v}_e \tag{6}$$

where k_d is a damping coefficient in the moving direction. By substituting $a = \sqrt{Rd}$ and Eq. (6), we have the derivative form of Eq. (6). $\mathbf{n}_v$ is the unit vector of the velocity and $\mathbf{v}_e$ is the end-effector velocity at the contact point.

$$\dot{\mathbf{F}}_f = \dot{f} = \mu\dot{F}_z + \frac{3\mu(2\nu - 1)}{10R}\left(\dot{F}_z d - F_z\dot{z}\right)\mathbf{n}_v + k_d\dot{\mathbf{v}}_e \tag{7}$$

The overall model with the frictional and normal force components of the contact model can be written in a compact form as:

$$\begin{aligned}\dot{\mathbf{F}}_e = &\Big(\left(6E^2RF_z\right)^{\frac{1}{3}}\dot{d}\Big)\mathbf{n}_z \\ &+ \Big(\mu\dot{F}_z + \frac{3\mu(2\nu - 1)}{10R}\left(\dot{F}_z d + F_z\dot{d}\right)\Big)\mathbf{n}_v + k_d\dot{\mathbf{v}}_e\end{aligned} \tag{8}$$

$\mathbf{n}_v$ is unit vector and d is the deformation at central point of contact circle and is calculated from Eq. (1), and $\dot{d} = \dot{x}_z$. F_z is the vertical force (the surface normal direction) applied on the surface by the manipulator and $\mathbf{v}_e = ||\mathbf{J}\dot{\mathbf{x}}_M||$ is the moving velocity of the tool contact point.

B. Contact Constraint Modeling

The motion studied in this paper is primarily in the sliding mode, which an equality constraint can describe. Since the sliding is embedded in the contact model, additional constraints for sliding are not required. However, a constraint is added to make sure the robot is only sliding in the desired path where the path is curved. Figure 4 shows the two components of the force acting on the tool, namely, frictional and centripetal forces along the path of curvature R_c as shown in Figure 3. The contact model provides the sliding friction, and the centripetal force constraint is added as a constraint .

$$\frac{\mathbf{J}^{-T}\mathbf{M}(\mathbf{q})\mathbf{J}^{-1}\|\mathbf{J}\dot{\mathbf{x}}_M\|^2}{R_c} \le \mu\mathbf{N}^T\mathbf{F}_e\mathbf{N} \tag{9}$$

$\mathbf{F}_e \in \mathbb{R}^3$ is the force vector at the end effector and $\mathbf{J}$ is the contact jacobian. $\mathbf{J}^{-T}\mathbf{M}(\mathbf{q})\mathbf{J}^{-1}$ is the effective mass at the contact point of the robot with a mass matrix of $\mathbf{M}(\mathbf{q})$ and $\mathbf{J}\dot{\mathbf{x}}_M$ is the moving velocity of the contact point. $\mathbf{N}$ is the surface average normal vector (as precept by the force-torque sensor). In this study, we use $\mathbf{N} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$, which is only in the z direction[2]. The constraint represented by Eq. 9 keeps the robot in contact when the cartesian tracking path has a curvature, and the robot is operated in a lower impedance mode. It is an effect of the resulting centripetal force on the effective mass at the contact. For instance, if the velocity at a curve is high, it would slip in the orthogonal direction of the moving direction unless high positional gains are used to compensate for it.

IV. Problem formulation

The optimization problem is to solve a control trajectory that would result in a desired cartesian trajectory along a desired force profile. The state for our trajectory optimization is represented as: $\mathbf{x}_M = \begin{bmatrix}q_1 \; q_2 \; q_3 \; q_4 \; q_5 \; q_6 \; q_7\end{bmatrix}, \mathbf{x} = [\mathbf{x}_M \; \dot{\mathbf{x}}_M \; \mathbf{F}_e]^T, \mathbf{u} = \boldsymbol{\tau}_u$. The overall problem is formulated in Formulation 1 as shown below.

where $\delta\mathbf{x}[i] = (\mathbf{x}[i] - \mathbf{x}_d[i])$ and $\delta\mathbf{F}[i] = (F_z[i] - F_d[i])$ are the position error and normal force error with respect to reference $\mathbf{x}_d[i]$ and $F_d[i]$. $\mathbf{Q} \in \mathbb{R}^{n\times n}$ and $\mathbf{R} \in \mathbb{R}^{m\times m}$ are the state and control weighting matrices, $\mathcal{FK} \in \mathbb{R}^{4\times 4}$ is the forward kinematics function of the manipulator and $W_p \in \mathbb{R}^{4\times 4}$ is the state weighting matrix for the forward function. For simplicity, we use $\boldsymbol{\phi} = (\mathbf{x}[0, \ldots, N], \mathbf{u}[0, \ldots, N-1])$ to represent the sequence of state-control pairs.

The manipulator model dynamics are expressed below.

$$\ddot{\mathbf{x}}_{\mathbf{M}} = \mathbf{M}(\mathbf{q})^{-1}(\boldsymbol{\tau}_u - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \mathbf{J}^T\mathbf{F}_e) \tag{11}$$

where $\mathbf{q}$ is the joint state vector, $\mathbf{M}(\mathbf{q})$ is the joint space mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis term, $\mathbf{G}(\mathbf{q})$ is the gravity term,

[1] we denote $\cos\theta = c\theta$ and $\sin\theta = s\theta$.

[2] $\mathbf{N}$ varies with the surface deformation which is stochastic and can estimated through an external force-torque sensor attached to the end-effector

Formulation 1 Simultaneous trajectory and force optimization

$$\text{(Tracking Task)} \quad \min_{\boldsymbol{\phi}} \sum_{i=0}^{N} \overbrace{\delta\mathbf{F}[i]^T\mathbf{Q}_F\,\delta\mathbf{F}[i]}^{\text{force tracking}} + \mathbf{u}[i]^T\mathbf{R}\mathbf{u}[i] + \overbrace{W_p\big\|\mathcal{FK}(\mathbf{x}_M^c[i]) - \mathbf{x}_e^d[i]\big\|_2^2}^{\text{pose tracking}}$$

$$\text{(Decision Variables)} \quad \boldsymbol{\phi}[i] = [[\overbrace{\mathbf{x}_M[i]\ \dot{\mathbf{x}}_M[i]\ \mathbf{F}_e[i]}^{\mathbf{x}[i]}]^T, \mathbf{u}[i]^T]^T \quad \forall i = 1, \dots, N-1 \tag{10a}$$

$$\text{(Dynamics)} \quad \text{s.t.} \quad \mathbf{x}[i+1] = \mathcal{F}(\mathbf{x}[i], \mathbf{u}[i]) \tag{10b}$$

$$\text{(Initial Condition)} \quad \mathbf{x}[0] = \mathbf{x}_0 \tag{10c}$$

$$\text{(Joint Limits)} \quad \underline{\mathbf{x}}_M \le \mathbf{x}_M \le \bar{\mathbf{x}}_M \tag{10d}$$

$$\text{(Torque Limits)} \quad \underline{\mathbf{u}} \le \mathbf{u} \le \bar{\mathbf{u}} \tag{10e}$$

$$\text{(Contact Constraint)} \quad \frac{\mathbf{J}^{-T}\mathbf{M}(\mathbf{q})\mathbf{J}^{-1}\|\mathbf{J}\dot{\mathbf{x}}_M\|^2}{R_c} \le \mu\mathbf{N}^T\mathbf{F}_e\mathbf{N} \tag{10f}$$

$\boldsymbol{\tau}_u$ is the torque applied at joints, $\mathbf{J}$ is the contact Jacobian and $\mathbf{F}_e$ is the external Cartesian wrench at the end-effector.

V. Constrained Trajectory Optimization with Contact Dynamics

Given the manipulator and the contact dynamic models, Differential Dynamic Programming (DDP) is used to generate desired joint and Cartesian motion as well as force profiles obeying dynamic constraints. DDP is well received for effectively solving unconstrained trajectory optimization [36]. It represents an indirect method which only optimizes control inputs, and the dynamics constraint is implicitly satisfied during the forward trajectory rollout. Given an initial guess of control inputs, an updated state trajectory is generated by forward propagating the differential equation of system dynamics. Then a quadratic approximation is constructed for the cost function and dynamics around the current trajectory, so that a Riccati recursion can be used to derive the optimal feedback control law. By iteratively updating the state and control trajectories, the optimization will converge to an optimal solution.

One limitation of DDP stems from its difficulty in addressing constraints other than the dynamics constraint enforced during the forward pass rollout. Since our contact model enforces state, control, frictional constraints, it is desired to incorporate these contact constraints along with the state and control constraints. Our previous work in [50] proposed an iterative and distributed method based on Alternating Direction Method of Multipliers (ADMM) to incorporate the contact dynamics and constraints. In this work, we introduce an inverse kinematics sub-problem and extend the entire ADMM framework to be a consensus variant to further improve the computational efficiency. Note that a sequential variant can also be established. Details are demonstrated in Sec.X and we benchmark multiple variants in Sec.VII.

Formulation 2 Distributed constrained optimization (consensus)

$$\text{(Tracking Task)} \quad \min_{\boldsymbol{\phi}, \hat{\boldsymbol{\phi}}, \bar{\boldsymbol{\phi}}} \sum_{i=0}^{N} \delta\mathbf{F}[i]^T\mathbf{Q}_F\,\delta\mathbf{F}[i] + \mathbf{u}[i]^T\mathbf{R}\mathbf{u}[i] + W_p\big\|\mathcal{FK}(\mathbf{x}_M^c[i]) - \mathbf{x}_e^d[i]\big\|_2^2 + I_{\mathcal{J},\mathcal{U},\mathcal{F}}(\bar{\mathbf{x}}_M[i], \bar{\mathbf{u}}[i], \bar{\boldsymbol{\lambda}}[i])$$

$$\begin{aligned} &\text{(Variables-DDP)} && \boldsymbol{\phi}[i] = [[\overbrace{\mathbf{x}_M[i]\ \dot{\mathbf{x}}_M[i]\ \mathbf{F}_e[i]}^{\mathbf{x}[i]}]^T, \mathbf{u}[i]^T]^T \\ &\text{(Variables-IK)} && \hat{\boldsymbol{\phi}}[i] = \hat{\mathbf{x}}_M[i]^T \\ &\text{(Variables-Proj)} && \bar{\boldsymbol{\phi}}[i] = [\bar{\mathbf{x}}_M[i]^T, \bar{\mathbf{u}}[i]^T, \bar{\boldsymbol{\lambda}}[i]^T]^T \\ &&& \forall i = 0, 1, \dots, N-1 \end{aligned} \tag{12a}$$

$$\text{(Dynamics)} \quad \text{s.t.} \quad \mathbf{x}[i+1] = \mathcal{F}(\mathbf{x}[i], \mathbf{u}[i]) \tag{12b}$$

$$\text{(Initial Condition)} \quad \mathbf{x}[0] = \mathbf{x}_0 \tag{12c}$$

$$\text{(Consistency Constraints)} \quad \begin{cases} \hat{\mathbf{x}}_M = \bar{\mathbf{x}}_M \\ \mathbf{x}_M = \bar{\mathbf{x}}_M \\ \mathbf{u} = \bar{\mathbf{u}} \\ \boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}} \end{cases} \tag{12d}$$

The ADMM algorithm decomposes a large-scale, holistic optimization problem into sub-problems and solves each sub-problem iteratively. In each iteration, the primal and dual variables are updated sequentially. Under mild conditions, both primal and dual variables converge to the optimal solutions. More details about ADMM algorithm are referred to [51]. To apply this algorithm for our soft-contact manipulation problem, we define various sets of copies and the corresponding consistency constraints. Therefore, the original Formulation 1 is transcribed into a distributed version as shown in Formulation 5. For simplicity, we define $\boldsymbol{\lambda} = (\dot{\mathbf{x}}_M^T, \mathbf{F}_e{}^T)^T$ and $\bar{\boldsymbol{\phi}} = (\bar{\mathbf{x}}_M[0, \dots, N], \bar{\mathbf{u}}[0, \dots, N-1], \bar{\boldsymbol{\lambda}}[0, \dots, N])$[3] to express the concatenated copies of states and controls that are required to be projected. Meanwhile, the variable set $\hat{\boldsymbol{\phi}} = (\hat{\mathbf{x}}_M[0, \dots, N])$ denotes a copy of joint position $\mathbf{x}_M$ and handles the end-effector tracking, i.e., the inverse kinematics (IK). The closed and convex sets $\mathcal{J}$, $\mathcal{U}$ and $\mathcal{F}$ stand for joint limit (10d), control limit (10e) and contact constraint (10f), respectively.

By utilizing indicator functions, the above constraints are encoded inside the new cost function in Formulation 5. In general, a compact indicator function regarding sets A, B is defined as

$$I_{A,B}(\mathbf{x}, \mathbf{y}) = \begin{cases} 0, & \mathbf{x}, \mathbf{y} \in A, B \\ +\infty, & \text{otherwise} \end{cases} \tag{13}$$

Given the transcribed optimization problem, an augmented Lagrangian (AL) can be derived (See X for a complete expression). Then based on ADMM, the original problem is divided into three sub-problems which are also known as sub-blocks.

[3] The decision variables $\mathbf{x}_M$ and $\boldsymbol{\lambda}$ are subsets of the full state $\mathbf{x}$

Each sub-block only requires part of the aforementioned AL as the local cost function:

Contact dynamics sub-block:

$$\begin{aligned}\mathcal{L}_{\text{cd}} =& \sum_{i=0}^{N} \left(\delta\mathbf{F}[i]^T\mathbf{Q}_F\, \delta\mathbf{F}[i] + \mathbf{u}[i]^T\mathbf{R}\mathbf{u}[i]\right) \\ &+ \frac{\rho_j}{2}\|\mathbf{x}_M - \bar{\mathbf{x}}_M + \mathbf{v}_j^k\|_2^2 + \frac{\rho_u}{2}\|\mathbf{u} - \bar{\mathbf{u}} + \mathbf{v}_u^k\|_2^2 \\ &+ \frac{\rho_f}{2}\|\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}} + \mathbf{v}_f^k\|_2^2\end{aligned} \tag{14a}$$

Inverse kinematics sub-block:

$$\begin{aligned}\mathcal{L}_{\text{ik}} =& \sum_{i=0}^{N} W_p\left\|\mathcal{FK}(\hat{\mathbf{x}}_M[i]) - \mathbf{x}_e^d[i]\right\|_2 \\ &+ \frac{\rho_j}{2}\|\hat{\mathbf{x}}_M - \bar{\mathbf{x}}_M + \mathbf{v}_{\text{ik}}^k\|_2^2\end{aligned} \tag{14b}$$

Projection sub-block:

$$\begin{aligned}\mathcal{L}_{\text{proj}} =& \sum_{i=0}^{N} I_{\mathcal{J},\mathcal{U},\mathcal{F}}(\bar{\mathbf{x}}_M[i], \bar{\mathbf{u}}[i], \bar{\boldsymbol{\lambda}}[i]) \\ &+ \frac{\rho_j}{2}\|\hat{\mathbf{x}}_M - \bar{\mathbf{x}}_M + \mathbf{v}_{\text{ik}}^k\|_2^2 + \frac{\rho_j}{2}\|\mathbf{x}_M - \bar{\mathbf{x}}_M + \mathbf{v}_j^k\|_2^2 \\ &+ \frac{\rho_u}{2}\|\mathbf{u} - \bar{\mathbf{u}} + \mathbf{v}_u^k\|_2^2 + \frac{\rho_f}{2}\|\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}} + \mathbf{v}_f^k\|_2^2\end{aligned} \tag{14c}$$

where ρ_j, ρ_u, ρ_f are step-size parameters corresponding to each constraint. Note that since the first two consistency constraints possess the same projection goal $\bar{\mathbf{x}}_M$, parameter ρ_j is adopted in both cases. Then for each ADMM iteration k, the updating sequence in a scaled form is

$$\boldsymbol{\phi}^{k+1} = \arg\min_{\boldsymbol{\phi}} \mathcal{L}_{\text{cd}} \quad \text{s.t. Eq. } (21b) \tag{15a}$$

$$\hat{\boldsymbol{\phi}}^{k+1} = \arg\min_{\boldsymbol{\phi}_{\text{ik}}} \mathcal{L}_{\text{ik}} \tag{15b}$$

$$\bar{\boldsymbol{\phi}}^{k+1} = \arg\min_{\bar{\boldsymbol{\phi}}} \mathcal{L}_{\text{proj}} \tag{15c}$$

$$\mathbf{v}_{\text{ik}}^{k+1} = \mathbf{v}_{\text{ik}}^k + \hat{\mathbf{x}}_M^{k+1} - \bar{\mathbf{x}}_M^{k+1} \tag{15d}$$

$$\mathbf{v}_j^{k+1} = \mathbf{v}_j^k + \mathbf{x}_M^{k+1} - \bar{\mathbf{x}}_M^{k+1} \tag{15e}$$

$$\mathbf{v}_u^{k+1} = \mathbf{v}_u^k + \mathbf{u}^{k+1} - \bar{\mathbf{u}}^{k+1} \tag{15f}$$

$$\mathbf{v}_f^{k+1} = \mathbf{v}_f^k + \boldsymbol{\lambda}^{k+1} - \bar{\boldsymbol{\lambda}}^{k+1} \tag{15g}$$

where $\boldsymbol{\phi}$, $\hat{\boldsymbol{\phi}}$ and $\bar{\boldsymbol{\phi}}$ are primal variables. $\mathbf{v}_{\text{ik}}$, $\mathbf{v}_j$, $\mathbf{v}_u$ and $\mathbf{v}_f$ are dual variables related to each consistency constraints defined in (21d).

To efficiently solve the constrained optimization problem in (15a), DDP is deployed and the state trajectory is always dynamically feasible by performing the forward pass. For Eq. (15c), this minimization problem reduces to a projection operator on convex sets $\mathcal{J}$, $\mathcal{U}$, and $\mathcal{F}$

$$\begin{aligned}\bar{\boldsymbol{\phi}}^{k+1} =& \arg\min_{\bar{\boldsymbol{\phi}}\in C} \frac{\rho_j}{2}\|\hat{\mathbf{x}}_M^{k+1} - \bar{\mathbf{x}}_M + \mathbf{v}_{\text{ik}}^k\|_2^2 \\ &+ \frac{\rho_j}{2}\|\mathbf{x}_M^{k+1} - \bar{\mathbf{x}}_M + \mathbf{v}_j^k\|_2^2 \\ &+ \frac{\rho_u}{2}\|\mathbf{u}^{k+1} - \bar{\mathbf{u}} + \mathbf{v}_u^k\|_2^2 + \frac{\rho_f}{2}\|\boldsymbol{\lambda}^{k+1} - \bar{\boldsymbol{\lambda}} + \mathbf{v}_f^k\|_2^2 \\ C =& \{(\bar{\mathbf{x}}_M, \bar{\mathbf{u}}, \bar{\boldsymbol{\lambda}}) | \bar{\mathbf{x}}_M \in \mathcal{J}, \bar{\mathbf{u}} \in \mathcal{U}, \bar{\boldsymbol{\lambda}} \in \mathcal{F}\}\end{aligned}$$

Algorithm 3 ADMM trajectory optimization

```
$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi}^0, \hat{\boldsymbol{\phi}} \leftarrow \hat{\boldsymbol{\phi}}^0, \bar{\boldsymbol{\phi}} \leftarrow \bar{\boldsymbol{\phi}}^0$
$\mathbf{v}_j \leftarrow \mathbf{v}_j^0, \mathbf{v}_{\text{ik}} \leftarrow \mathbf{v}_{\text{ik}}^0, \mathbf{v}_u \leftarrow \mathbf{v}_u^0, \mathbf{v}_f \leftarrow \mathbf{v}_f^0$
repeat
    $\boldsymbol{\phi} \leftarrow$ DDP $(\boldsymbol{\phi}, \bar{\mathbf{x}}_M - \mathbf{v}_j, \bar{\mathbf{u}} - \mathbf{v}_u, \bar{\boldsymbol{\lambda}} - \mathbf{v}_f)$ {Eq. 14a}
    $\hat{\boldsymbol{\phi}} \leftarrow$ IK $(\bar{\mathbf{x}}_M - \mathbf{v}_{\text{ik}})$ {Eq. 14b}
    $\bar{\boldsymbol{\phi}} \leftarrow$ Projection $(\frac{1}{2}(\hat{\mathbf{x}}_M + \mathbf{v}_{\text{ik}} + \mathbf{x}_M + \mathbf{v}_j), \mathbf{u} + \mathbf{v}_u, \boldsymbol{\lambda} + \mathbf{v}_f)$ {Eq. 14c}
    $\mathbf{v}_{\text{ik}} \leftarrow \mathbf{v}_{\text{ik}} + \hat{\mathbf{x}}_M - \bar{\mathbf{x}}_M$
    $\mathbf{v}_j \leftarrow \mathbf{v}_j + \mathbf{x}_M - \bar{\mathbf{x}}_M$
    $\mathbf{v}_u \leftarrow \mathbf{v}_u + \mathbf{u} - \bar{\mathbf{u}}$
    $\mathbf{v}_f \leftarrow \mathbf{v}_f + \boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}}$
until stopping criterion is satisfied
return $\boldsymbol{\phi}$
```

Then a saturation function can be used to efficiently project the infeasible values onto the boundaries induced by different constraints:

$$\begin{aligned}\bar{\boldsymbol{\phi}}^{k+1} = \Pi_{\mathcal{J},\mathcal{U},\mathcal{F}}\Big[&\frac{1}{2}(\hat{\mathbf{x}}_M^{k+1} + \mathbf{v}_{\text{ik}}^k + \mathbf{x}_M^{k+1} + \mathbf{v}_j^k), \\ &\mathbf{u}^{k+1} + \mathbf{v}_u^k, \boldsymbol{\lambda}^{k+1} + \mathbf{v}_f^k\Big]\end{aligned} \tag{16}$$

The whole process of our ADMM algorithm is shown in Algorithm 1. The selection of $\bar{\boldsymbol{\phi}}$ and dual variables $\mathbf{v}$ are arbitrary, and we initialized to be zero. The initial trajectory of $\boldsymbol{\phi}$ is generated by running forward dynamics with an initial guess of controls. In each ADMM iteration, the controls from last ADMM iteration will be sent to the current DDP solver as a warm-start, which makes the DDP solver converge faster within around ten iterations in each ADMM iteration after the initial one. Then the trajectories are solved iteratively until a stopping criterion with regard to primal residuals (see [51], Sec. 3.3) is satisfied(residuals of magnitude 10^{-2}).

VI. Model Predictive Control

We implement the ADMM planner in a Model Predictive Control (MPC) fashion in the real-time deployment. At each MPC cycle, the optimization in Eq. (10) is solved with a horizon $H = N_{\text{steps}}\delta t$ and its solution is used as the warm-start for the next MPC cycle, i.e. $(\mathbf{x}_0, \mathbf{u}_0) \leftarrow (\mathbf{x}[i]^*, \mathbf{u}[i]^*)$ where $\mathbf{x}_0$ is replaced by some initial state $\mathbf{x}[i]^*$.

$$\begin{aligned}\mathbf{u}[i] &= \mathbf{u}[i]^* + PD(\mathbf{x}[i]^*, \hat{\mathbf{x}}) \\ &= \overbrace{\mathbf{u}[i]^*}^{ff} + \overbrace{\mathbf{K}(\hat{\mathbf{x}}_M - \mathbf{x}_M[i]^*)}^{fd} + \overbrace{\delta\mathbf{u}_{FC}(\mathbf{F}_e[i]^*, \hat{\mathbf{F}}_e)}^{fc}\end{aligned} \tag{17}$$

where, $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_M \ \dot{\hat{\mathbf{x}}}_M \ \hat{\mathbf{F}}_e]^T$, is the filtered (low-pass) current state and $\mathbf{K} \in \mathbb{R}^{n\times n}$ is a gain matrix. $\mathbf{u}[i]$ is composed of three terms; the feed-forward(ff), feed-back(fd), and the admittance force controller(fc).

Remark 1: The low-level controller in the KUKA Sunrise software [52] uses user-defined joint space impedance control internally. We set a safe robot impedance to execute our commands as well as to allow the manipulator to behave safely in the compliant environment.

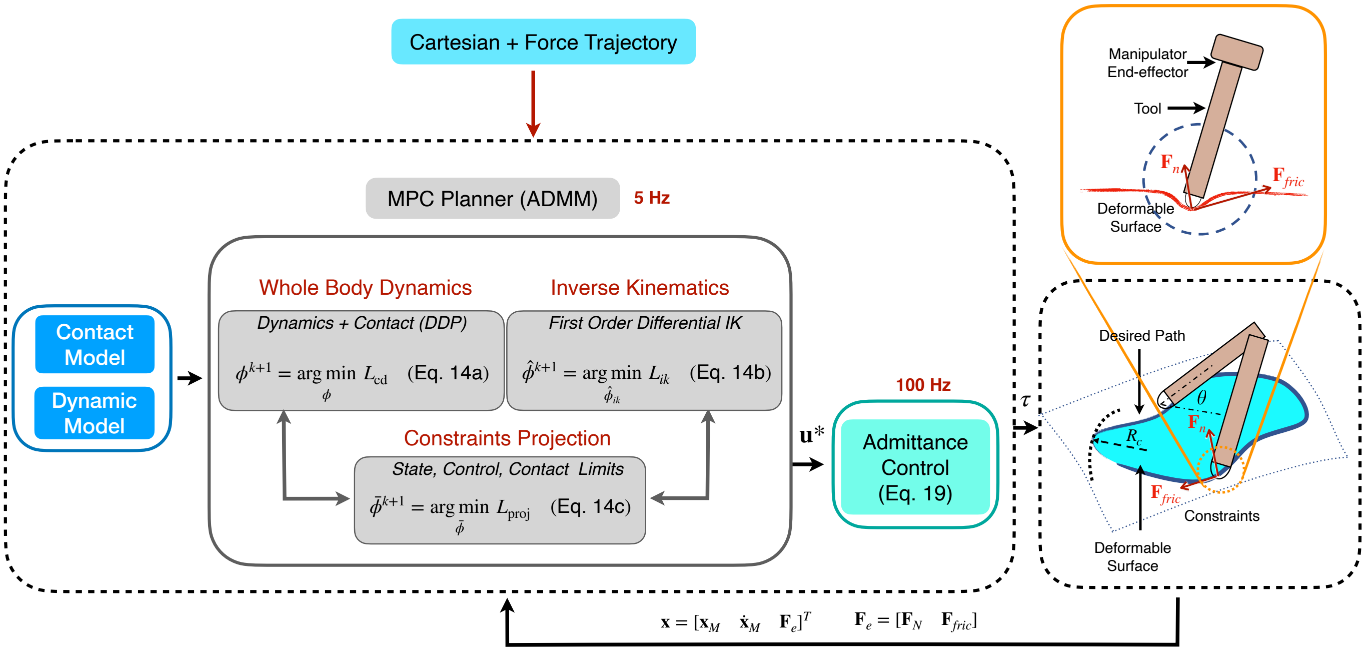

Cartesian + Force Trajectory
MPC Planner (ADMM)
5 Hz
Contact Model
Dynamic Model
Whole Body Dynamics
Dynamics + Contact (DDP)
$\phi^{k+1} = \arg\min_{\phi} L_{cd}$ (Eq. 14a)
Inverse Kinematics
First Order Differential IK
$\hat{\phi}^{k+1} = \arg\min_{\hat{\phi}_{ik}} L_{ik}$ (Eq. 14b)
Constraints Projection
State, Control, Contact Limits
$\bar{\phi}^{k+1} = \arg\min_{\bar{\phi}} L_{proj}$ (Eq. 14c)
$\mathbf{u}^*$
100 Hz
Admittance Control (Eq. 19)
τ
Manipulator End-effector
Tool
$\mathbf{F}_n$
$\mathbf{F}_{fric}$
Deformable Surface
Desired Path
θ
R_c
$\mathbf{F}_n$
$\mathbf{F}_{fric}$
Constraints
Deformable Surface
$\mathbf{x} = [\mathbf{x}_M \quad \dot{\mathbf{x}}_M \quad \mathbf{F}_e]^T \qquad \mathbf{F}_e = [\mathbf{F}_N \quad \mathbf{F}_{fric}]$

Fig. 3: Planning Framework. The planner is composed of an ADMM based solver executed in MPC fashion at 5 Hz. ADMM solver contains three sub-blocks for optimization. Manipulator and contact dynamic models are used in the whole body dynamics block. Inverse kinematics block uses first-order differential kinematics to generate Cartesian trajectory. States are projected to satisfy the state and contact constraints in the projection block. The low-level force controller which is the admittance controller is executed at 100 Hz. The desired Cartesian and force trajectory is tracked while obeying the constraints provided.

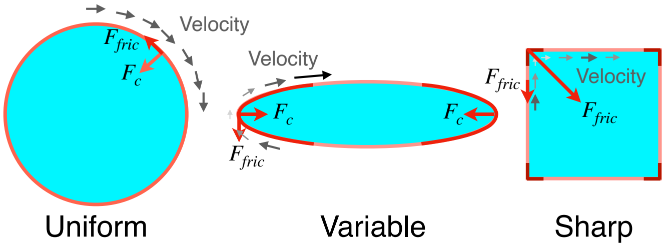


Fig. 4: Different Cartesian Path Geometries. At sharp corners of a Cartesian path, the robot needs to vary the velocity to compensate for centripetal and frictional force components caused through the contact

The high-level MPC loop is implemented at 5 Hz with a horizon of 1 s and a time step of $dt = 0.02$ s. In each MPC iteration, we solve for a trajectory by the ADMM planner as described in Section V and we use C++11 in all of our code implementations[4]. To maintain the contact force accuracy and to avoid instabilities resulting from low frequency control update rate [16], [53], we use a low-level force controller which runs at 100 Hz where the update rate is appropriate for the compliant (low-stiffness) environment.

In one of the ADMM planner blocks, we use DDP to solve for the robot and contact dynamics, which consumes most of the computational power. To make the real-time implementation feasible, we use automatic differentiation for derivatives provided by CppADCodeGen [54], and RobCoGen [46] is used to derive the analytical rigid body dynamic model. On a Linux machine (Intel i7) with 3.4 GHz clock speed, average computation time for a horizon of $1s$ took 150 ms.

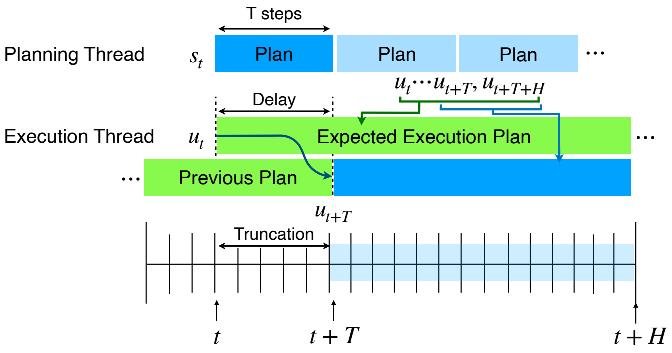


Fig. 5: Model Predictive Control (MPC) Implementation. H, T and $\mathbf{s}_t$ are the number of timesteps for the horizon, timesteps for the compute delay and current state from the system. Computation takes T steps. During the time, execution thread keeps executing the previous plan. Once planning thread is finished, the current execution plan is updated; the time-steps corresponding to delay is truncated.

Without AutoDiff and analytical models, it takes 1300 ms. With AutoDiff, it is possible to run MPC at a approximate rate of 5 Hz.

Each iteration of the DDP takes on average 10 ms while we limit the number of DDP iterations to 10 per ADMM cycle to ensure the solver returns an optimal trajectory on time. We run a maximum of 5 ADMM iterations in each trajectory computation cycle. While the constraint residuals are not guaranteed to reach the same threshold every trajectory iteration, our experimental results demonstrated that 5 ADMM iterations were sufficient to reduce residuals to the order of 10^{-2}. In an ideal MPC setup, the first control input from the output trajectory is applied to the robot and, the current state is

[4] The code implementation can be found at https://github.com/lasithagt/admm

used as the initial state for the MPC computation at the next iteration. However, there could be a significant computation burden for the trajectory planner in a practical setting. To take into account the computation and communication delay in the hardware system, we use an asynchronous MPC similar to [55] as shown in Figure 5. The current state ($\mathbf{x}[t]$) is applied to the planner. It takes T time-steps to compute the trajectory. During that period, the previous trajectory is executed. After T time steps, trajectory $\mathbf{U}_{t,t+H} = \{\mathbf{u}_t, \ldots, \mathbf{u}_{t+H}\}$ is returned by the planner. In the actual execution of the trajectory, we truncate the control input sequence up to T time-steps as shown in Figure 5. The planner and the execution threads are run in parallel in different threads using the threading library in C++11.

Although MPC is beneficial in handling model uncertainties and environmental disturbances, this is not sufficient for tasks that require force modulation. This is due to the instabilities that could arise from contact and determined by the control update rate and surface material properties [16], [53]. We use a force controller which updates control input at a rate of 100 Hz to avoid contact-induced instabilities. Admittance control is proven [48] to be better suited for compliant environments where impedance control is best suited for stiff environments.

Force control is important as the surface parameters are not uniform and prone to un-modelled dynamics (e.g., damping, restitution, slipping). To compensate for it, we use admittance control as the force controller in the low-level control. To mitigate the instabilities that would arise from position-based admittance control [56], we use torque as the control input as opposed to position control. The stiffness of the contact surface material was estimated as described in [16].

$$\boldsymbol{\delta}\mathbf{u}_{FC}[t] = \mathbf{C}(x, y, z)\, \mathbf{J}^T(\mathbf{F}_e^*[t] - \hat{\mathbf{F}}_e[t]) \tag{18}$$

where $\mathbf{J}(\mathbf{q})$ is the kinematic jacobian matrix, $\boldsymbol{\delta}\mathbf{u}_{FC}[t]$ is joint-space torque and $\mathbf{C}(x, y, z)$ is the compliance matrix which can vary spatially (in the space (x, y, z)[5].

VII. Experiments

To validate the applicability of theoretical attributes of our work in a practical setting, we demonstrate it via physical experiments on a custom-designed platform, shown in Figure 6. Moreover, we compare our framework with other state-of-the-art methods which are used widely in robotics motion planning in simulation (in MATLAB©). Then, we show the contact parameter identification methods and compare results with the disturbance-induced tracking task.

A. Trajectory Optimization Algorithm Comparisons

The primary motivation for a distributed motion planning scheme such as ADMM is to use different optimization methods that specialize for each sub-problem and incorporate constraints into it. For example, it is efficient to use first-order differential methods to generate IK solutions and solve for the dynamic model separately to find a consensus between them. These were benchmarked in MATLAB© with a 3.4 GHz i7-core processor. In addition, we benchmark our method against other standard methods, namely:

1) **Our Method:** ADMM with 3-block architecture (Concensus). *The ADMM architecture with three blocks: nonlinear dynamics, IK and projection blocks are solved and concensus is found in the ADMM update. More details can be found in Appendix X-B*
2) Sequential Quadratic Programming (SQP). *Direct collocation with trapizoidal transcription were used with the dynamical system with contact dynamics;*
3) Iterative Linear Quadratic Regulator (iLQR - vanilla DDP). *iLQR [36] was used with a combined cost of the desired cartesian trajectory ($SE(3)$), force trajectory, state, control, and contact constraints;*
4) ADMM with 2-block architecture (Sequential). *ADMM scheme with 2-blocks as implemented in [50] was used. In the nonlinear dynamics block, desired state and control cost is used in the DDP solver. Projection block projects to state, control and contact constraints;*
5) ADMM with 3-block architecture (Sequential). *The ADMM architecture with three blocks: nonlinear dynamics, IK and projection blocks are solved sequentially in the ADMM update. More details can be found in Appendix X-B.*

Figure 7 shows the comparison of cost reduction and the contact residuals reduction for the methods described listed above. For the ADMM variants, $x - axis$ represents the number of dynamic solver (DDP) iterations instead of ADMM iterations for a fair comparison with other methods (e.g., SQP and iLQR).

B. Cost Reduction

One advantage of using ADMM with a 3-block architecture is to track a reference Cartesian trajectory ($SE(3)$) without adding an extra cost term in the dynamics block, which could impede the fast convergence of DDP. The cartesian trajectory can be solved efficiently with differential IK and redundancy resolution for redundant systems. Moreover, the solution can warm start the dynamics block. It is evident in the convergence of ADMM (Figure 7) with 3-block architecture solved in the consensus manner. However, sequential variants tend to converge slower with the penalty parameters (ρ) used. While the convergence is fast, more iterations are needed for the constraint residuals to drop to a satisfactory threshold of 10^{-2}.

C. Constraint Satisfaction

Constraint residuals for each iteration in Figure 7 (right) are recorded, and the cost coefficients are tuned for each method. It is observed that residuals in the consensus ADMM with 3-block architecture drop relatively faster than other methods. Moreover, the SQP method initially started with a lower residual value and increased (and decreased again). This observation can be attributed to the pre-processing phase of the SQP solver to find an initial solution that is constraint satisfied regardless of the objective cost. Furthermore, the iLQR keeps

[5] $\mathbf{C}$ is set to be a constant in this study and identified experimentally

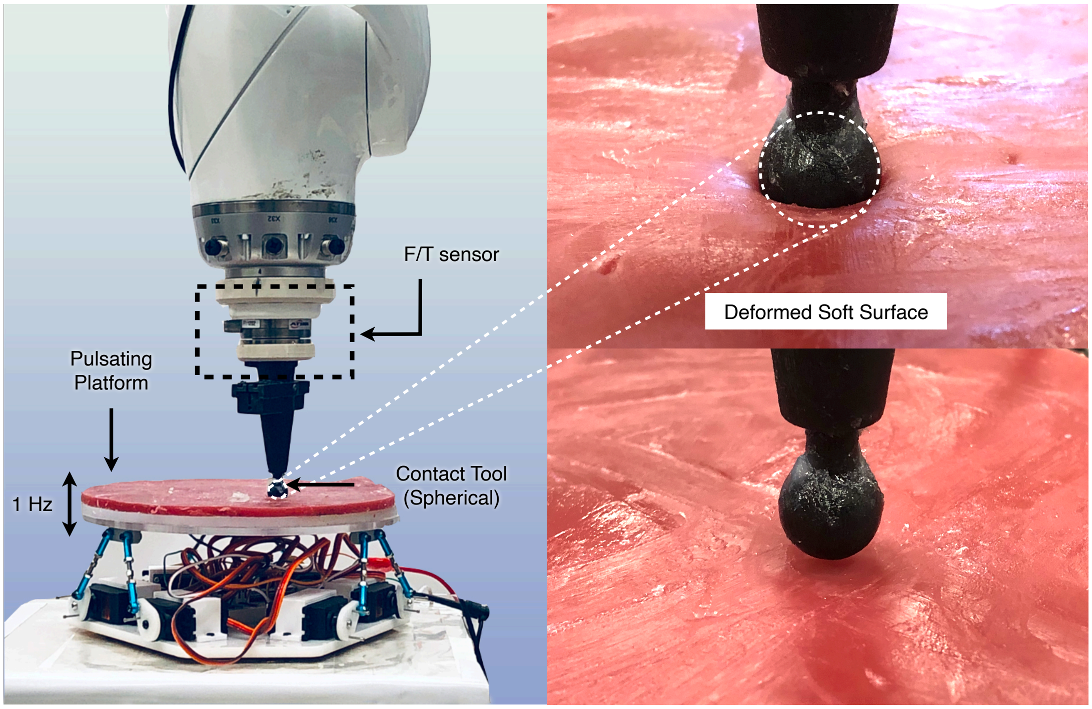


Fig. 6: Experimental Setup. KUKA manipulator is equipped with spherical tool and a ATI force/torque sensor distally to the contact point. The pulsation platform moves up and down at a fixed rate of $1Hz$. Surface is covered with a thick soft substrate. The tool of the end-effector is comprised of a spherical indenter which is in contact with the deformable surface.

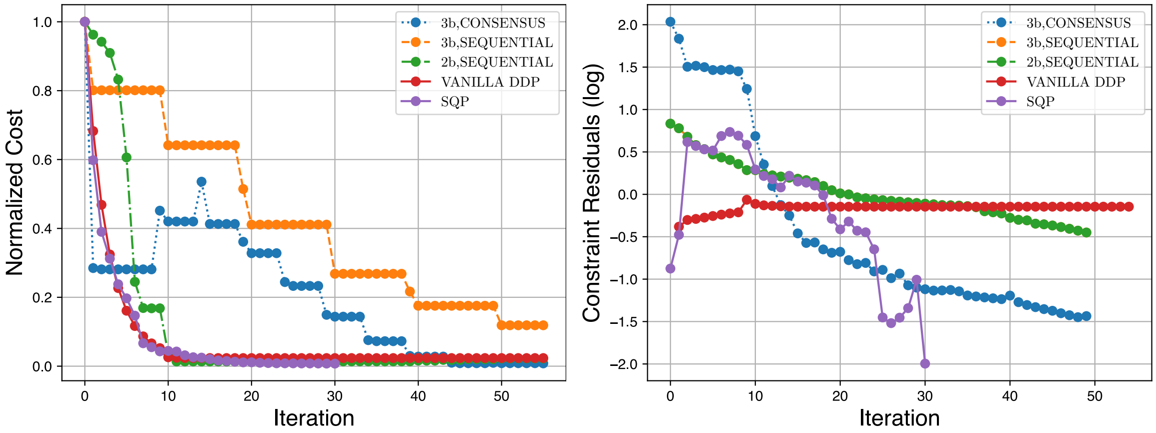


Fig. 7: Comparison of optimization methods. 3-block consensus, 3-block sequential, 2-block sequential, vanilla-DDP, SQP. *left:* Cost vs Iterations. The cost is normalized such that the initial value is scaled to be unity. *right:* Constraint Residuals vs Iterations

the residual constant without significant improvements. It is due to the cost and constraints combined cost term, prioritizing cost over constraints.

D. Experimental Results

In our previous work [50], we demonstrated the validity of the contact model experimentally when the environmental platform is stationary (as shown in Figure 8). In this work, we extend it to a non-stationary environment where periodic disturbances are applied ($1Hz$ pulsations) as shown in Figure 6. Moreover, we demonstrate the feasibility of the real-time deployment of the proposed framework. The experimental results can be summarized to the categories below:

1) Open-loop trajectories in the presence and absence of environmental disturbances.
2) Open-loop trajectories with low-level force controller activated.
3) Model predictive controller (MPC) with low-level force controller in the presence and absence of disturbances.

E. System Identification of Material Properties

To experimentally validate the proposed soft contact model, parameters related to contact body material need to be identified, e.g., frictional coefficient and Young modulus. The friction coefficient is estimated by performing pre-determined

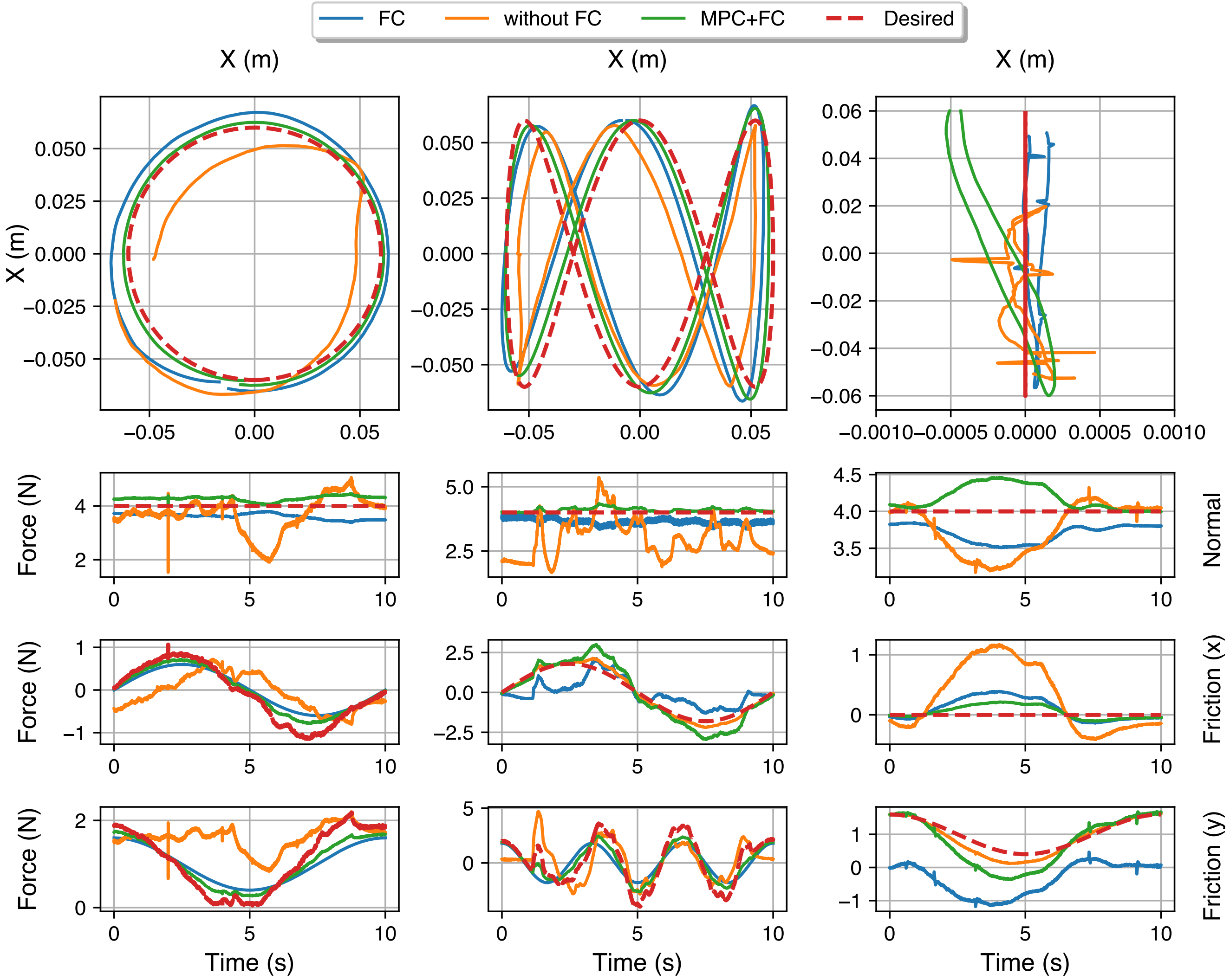


Fig. 8: *Top row:* Paths tracked by the end-effector vs commanded. *Bottom Row:* Comparison of ground truth of normal and frictional forces vs predicted by the model.

motions along the surface of the material surface while recording the force/torque data through an ATI mini45 sensor which is attached to the end-effector in Figure 6.

The Young modulus is estimated through performing cyclic linear probing on the surface of the material with the same end-effector point geometry of a sphere (was tested on a material testing platform INSTRON©). It is performed through a non-linear least square estimator by using Eqs. (1) and (6) was used to estimate the frictional coefficients by non-linear least squares estimation as below. Formulation 4 summarizes parameter estimation problem.

Formulation 4 presents the identification of parameters. Frictional force magnitude in the moving direction $\mathbf{F}_{\text{fric}}$, velocity magnitude $\mathbf{v}_e$, and normal contact force $\mathbf{F}_z$ are calculated from the collected data. A three dimensional robust least square approximation is fit with a logistic distance function in MATLAB©. This fitting is used to mitigate the sensitivity to the model deviation as the deformation increases (see Figure 10). Identified Young Modulus and friction coefficient were incorporated into the overall optimization in Eq. (10). Desired states to track are the desired end-effector position (x_e, y_e, z_e) and the desired normal contact force F_z.

Formulation 4 Contact Parameter Identification

$$\min_{E} \sum_{k=0}^{k=N} \|\bar{d}_k - \frac{9F_k^2}{16E^2R}^{\frac{1}{3}} \|_2 \quad \text{Derived from Eq. (1)}$$

$$\min_{\mu, k_d} \sum_{k=0}^{k=N} \|\bar{d}_k - \mu F_z \left[1 + (2\nu - 1)\frac{3a^2}{10R^2}\right] \mathbf{n}_v$$

$$- k_d \mathbf{v}_e\|_2 \quad \text{Derived from Eq. (6)}$$

The purpose of identifying material properties is two-fold. First, to validate that the used models are well suited and to use in the trajectory optimization framework to generate optimal open-loop trajectories. Friction data fitting results are presented in Figure 10. Data were fit with a resulting R-

	Motion Path 1		Motion Path 2		Motion Path 3	
	path (m)	force (N)	path (m)	force (N)	path (m)	force (N)
without FC	0.284	1.042	0.392	1.328	0.302	0.891
with FC	0.0183	0.384	0.0258	0.481	0.0172	0.0273
MPC + FC	**0.0164**	**0.283**	**0.0191**	**0.319**	**0.0233**	**0.0084**

TABLE I: RSME for motion and force and motion, FC-Admittance Force Control, MPC-Model Predictive Control, Motion Path 1(Circle), Motion Path 2(Eight), Motion Path (Line)

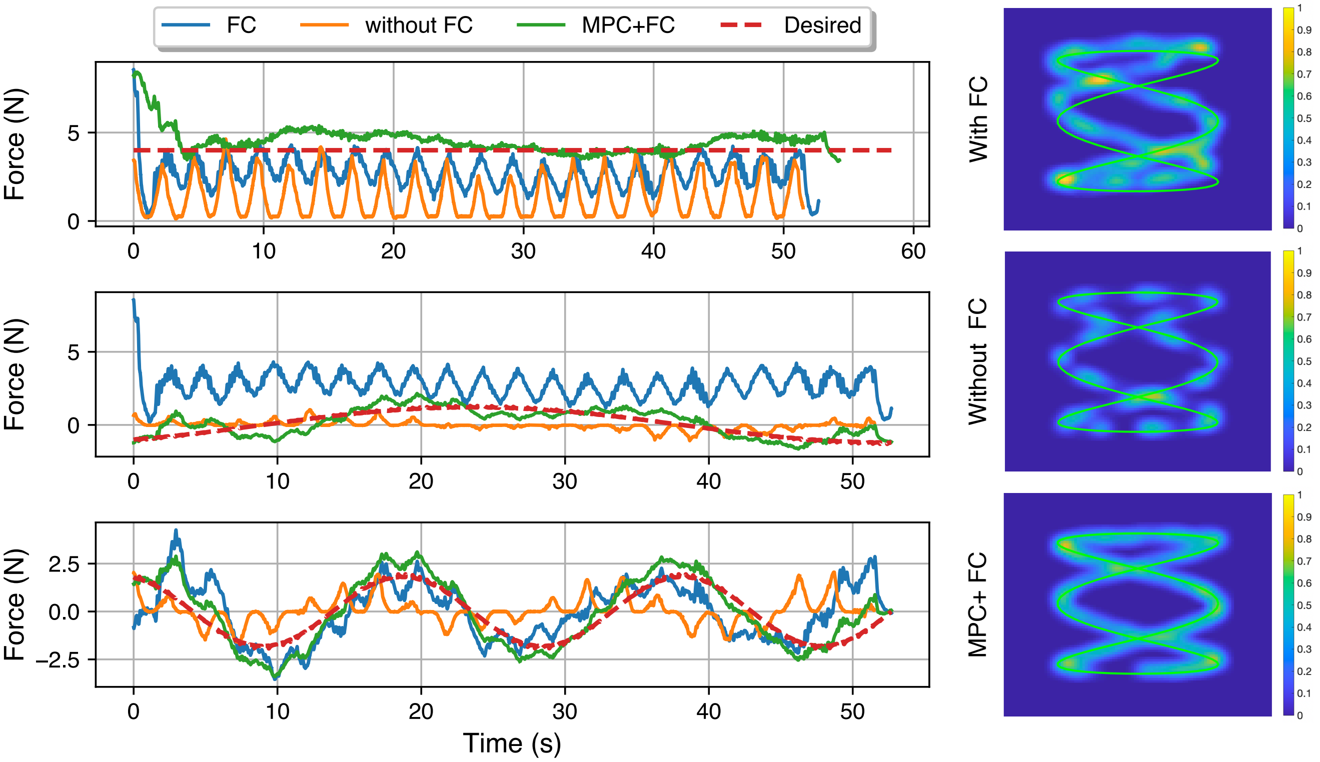


Fig. 9: *left:* Normal and frictional forces with surface pulsations. Comparison of the activated force controller (*blue*), inactivated force control ($orange$), and MPC and activated force control ($green$). $right$ Normalized force and motion superimposed on a heat map. Note that, motion pulsations of the experimental platform (Figure 6) cause the ripples observed in force profiles, for which the controller is attempting to compensate.

squared value of 0.9103. Frictional coefficient ($\mu = 0.4512$) and damping coefficient ($k_d = 13.1315$) were identified.

It is observed that with the increase of normal force on the surface, the effects of deformation dominates the frictional forces. This phenomenon is due to the increased rolling friction and material-specific artifacts, e.g., non-uniformity in frictional coefficient and stress distribution. Moreover, the presence of fluids or any micro-particular particles will increase the non-uniformity.

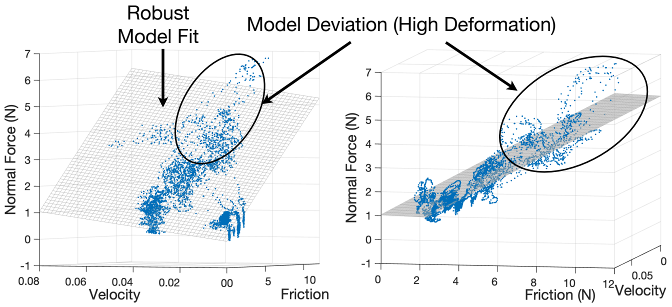


Fig. 10: Friction model validation and identification. $\mu = 0.4512, K_d = 13.1315$, R-squared $= 0.9103$.

In the implementation, optimal state trajectories and inputs are generated through the optimization formulated in Eq. (10). Constraints were satisfied within 10^{-2} residual value violations in both primal and dual stopping criteria. Figure 10 shows that the contact model used is valid for a range of normal forces. Therefore, the desired contact force is maintained within the valid bound of the friction model.

F. Open-loop Trajectory Generation without Environmental Disturbances

Open-loop trajectories generated from our trajectory optimization method are used as reference trajectories for the experiments. Any mismatch in contact forces (e.g., those due to friction and deformation) would directly affect the motion and vice versa. It is observed that the control input solved via TO was able to track the motion ($\mathbf{x}_e^d$) and force profile ($\mathbf{F}_d$) significantly better than a position-controlled robot with force modulation as shown in Figure 8. This is due to its open-loop compensation of frictional and centripetal forces encountered during the contact interaction. However, the tracking performance is not superior but reasonably adequate due to the model imperfections caused by the soft material's unmodelled dynamics, stiction, and non-uniformity.

As described in Section VII-F, the same open-loop full trajectories were executed with force control as described in Section VI. Figure 8 shows experimental results, which suggest low-level force control alone (the blue trajectory) can improve the reference force tracking accuracy significantly. However, the reference motion tracking accuracy degrades due to the frictional forces encountered on the surface observed in the experimental results. The reference force and motion tracking performance improve with active force control and MPC (the green trajectory). Furthermore, Table I shows quantified results on multiple motion trajectories with different geometries with different curvatures. In Cartesian geometries that contain sharp curves (e.g., rectangular geometry), the centripetal force component is an addition to the frictional force in contact force compensation, the velocity at the corners needs to vary to maintain the path and the contact as illustrated in Figure 4. Failing to compensate for it could result in sliding and deviating from the desired motion (the *without FC* case in Figure 8). Only friction force needs to be compensated in a straight line, while both centripetal and frictional components are present in a curved geometry.

G. Model Predictive Control with Low-level Force Controller under Periodic Environmental Disturbances

Previously, it was shown that force control and MPC could improve both force and motion tracking accuracy. However, the environment is subject to motion disturbances in a realistic setting. To cope with such disturbances and compensate for frictional forces simultaneously, we experimentally show the efficacy of the proposed TO method. Figure 9 illustrates the force and motion tracking accuracy when under force control, without force control, and with force control, and MPC. Similar to the static case, low-level force control improves force tracking, but the "reactive" nature can be observed in the force tracking (in Figure 9 right column, second row subplot). As a result, the MPC with contact model information improves both force and motion tracking accuracy significantly. Moreover, the system remains stable with externally induced motion disturbances and model uncertainties. Such a success can be attributed to the lower-level admittance controller and the model-based TO run in a MPC fashion. Experiments were run on multiple Cartesian trajectories to validate on varying geometries, and quantified results are reported in Table II.

H. Discussion and Limitations

The experiments and results demonstrated the importance of incorporating contact in the TO framework for better force and motion tracking accuracy. In safety-critical applications, stiff, position-controlled robots are not desired as they could raise safety concerns. Contact-model-based force-controlled control architectures could provide safe and improved performance. Our static environment results exhibit better performance compared to a motion-induced dynamic environment. Fitting the environment motion disturbance (e.g., breathing, heartbeat) to a parametric model can further improve the tracking performance. In our current work, the robot's impedance is set to a predefined mode that can be limiting. For example, a high impedance robot is more suited for tasks requiring more motion accuracy over force and vice versa. Adapting the robot's impedance depending on tasks and applications can further improve safety, force, and motion tracking accuracy.

While our work presents a method for force and motion TO, we acknowledge multiple limitations pertaining to the application and computational techniques. For example, the contact model we use is only valid locally. For large deformable bodies, the contact model will require a more computationally expensive method (e.g., Finite Element Methods). Furthermore, the range of force magnitudes was maintained throughout the experiments to be within a local range (in the linear range of Figure 10). To compensate for the friction in high deformation and lubrication cases, additional factors of the deformation friction will have to be learned or modeled.

Although this work achieved MPC online planning, there are more promising TO parallelization mechanisms [57]–[59] that can further improve the computational performance. For instance, (i) in the DDP backward pass, a computation speed-up is achievable through a parallelization method of Riccati recursion [44]; (ii) In the DDP forward pass, computation speed-up can be targeted by first proposing an approximate physics model, where a computationally cheap estimate of a coarse model can be evaluated [60]. Then the generated coarse trajectory can be used as an initial seed of the TO with a fine-grained model. This coarse-fine problem can be solved in an iterative procedure. (iii) Finally, these mechanisms above will suit a paralleled ADMM implementation on GPU processors. Although this work does not focus on TO parallelization mechanisms, these potential directions are worth to be reported and can be insightful to the deformable tissue manipulation and medical robotics community.

VIII. Conclusion and Future Work

In automation tasks requiring physically soft tissue contact, it is paramount to design soft contact interaction models where controllers can be designed to guarantee safety performance. Contact modeling is crucial in correctly identifying the contact material and performing mundane tasks such as incisions along given paths and motion disturbance compensation. This study presented a coherent framework for simultaneous motion and force modulation on compliant surfaces. Moreover, we presented a distributed (ADMM), real-time framework executed in a MPC fashion capable of handling state, control, and contact constraints. Further, we incorporated a soft contact dynamical model into the trajectory optimization (TO). Results proved that motion and force tracking accuracy is significantly improved in both static and dynamic environments. Potential applications of this work include contact manipulation in soft tissues or safety-critical environments.

	Motion Path 1		Motion Path 2		Motion Path 3	
	path (m)	force (N)	path (m)	force (N)	path (m)	force (N)
Pulsations, without FC	**0.0142**	2.723	**0.0264**	2.470	**0.0075**	1.377
Pulsations, with FC	0.0839	1.095	0.0757	1.289	0.0187	**0.8450**
Pulsations, MPC + FC	0.0258	**0.573**	0.0384	0.852	0.0153	**0.3230**

TABLE II: RSME for motion and force and motion, FC-Admittance Force Control, MPC-Model Predictive Control

Trajectories solved from the TO were experimentally validated on a soft surface (EcoFlex©) with the aid of a robot manipulator with an attached spherical shaped tooltip. Surface material properties were estimated and further used in generating optimal trajectories. Experiments were performed on a static and a motion-induced dynamic environment. Results of MPC, with and without force control, were presented. Ground truth forces were obtained using a force-torque sensor (ATI mini45) and compared against the obtained results. MPC with force control was able to track both motion and force both in a static and a dynamic environment with significant improvements. Results and discussion conclude model-based contact modeling and hierarchical TO (e.g., low-level and high level) provide a better alternative for safe simultaneous force and motion generation.

The future extension of this work is to improve the generability (i.e., "richness") of the contact model to adapt to a wide range of material properties. Furthermore, real-time estimation of the contact model properties can improve the adaptability of the planning framework. Moreover, we intend to extend the work to plan trajectories in three-dimensional surfaces to demonstrate practical applications such as planning robotic incisions on a human body.

IX. ACKNOWLEDGEMENTS

The authors would like to thank Qie Sima for his valuable insights and contribution to developing and testing the contact model.

X. APPENDIX

A. Deformable Patch Contact Model

In this Appendix, we provide the details on several stress distributions. First, the normal Stress Distribution σ_z:

$$\frac{\sigma_z}{p_m} = -\frac{3}{2}\left(1 - \frac{r^2}{a^2}\right)^{\frac{1}{2}} \quad (r \leq a) \tag{20}$$

Radical Stress Distribution σ_r:

$$\frac{\sigma_r}{p_m} = \frac{2\nu - 1}{2}\frac{a^2}{r^2}\left[1 - \left(1 - \frac{r^2}{a^2}\right)\right] - 3\nu\left(1 - \frac{r^2}{a^2}\right)^{\frac{1}{2}} \quad (r \leq a)$$

Hoop Stress Distribution σ_θ:

$$\frac{\sigma_\theta}{p_m} = \frac{1 - 2\nu}{2}\frac{a^2}{r^2}\left[1 - \left(1 - \frac{r^2}{a^2}\right)\right] - \frac{3}{2}\left(1 - \frac{r^2}{a^2}\right)^{\frac{1}{2}} \quad (r \leq a)$$

where $p_m = F/(\pi a^2)$ is the average stress applied in contact part by manipulation and $a = \sqrt{Rd}$ is the radius of contact area (refer to figure 2). The transformation matrix T is

$$T = \begin{bmatrix} c\theta & s\theta & 0 \\ -s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

B. ADMM Formulations

1) Augmented Lagrangian: The augmented Lagrangian for our three-block consensus ADMM is written as follows:

$$\begin{aligned}\min_{\boldsymbol{\phi}, \boldsymbol{\phi}_{\text{ik}}, \bar{\boldsymbol{\phi}}} \quad & \sum_{i=0}^{N} \left(\delta\mathbf{F}[i]^T \mathbf{Q}_F \, \delta\mathbf{F}[i] + \mathbf{u}[i]^T \mathbf{R}\mathbf{u}[i]\right) \\ & + \; W_p \left\| \mathcal{FK}(\hat{\mathbf{x}}_M[i]) - \mathbf{x}_e^d[i] \right\|_2 \\ & + I_{\mathcal{J},\mathcal{U},\mathcal{F}}(\bar{\mathbf{x}}_M[i], \bar{\mathbf{u}}[i], \bar{\boldsymbol{\lambda}}[i]) \\ & + \frac{\rho_j}{2}\|\hat{\mathbf{x}}_M - \bar{\mathbf{x}}_M + \mathbf{v}_{\text{ik}}^k\|_2^2 \\ & + \frac{\rho_j}{2}\|\mathbf{x}_M - \bar{\mathbf{x}}_M + \mathbf{v}_j^k\|_2^2 \\ & + \frac{\rho_u}{2}\|\mathbf{u} - \bar{\mathbf{u}} + \mathbf{v}_u^k\|_2^2 \\ & + \frac{\rho_f}{2}\|\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}} + \mathbf{v}_f^k\|_2^2 \end{aligned}$$

2) Three-block Sequential ADMM: Instead of establishing a consistency constraint between the decision variables from the IK sub-block the projection sub-block, the sequential ADMM enforces an equality between DDP sub-block and IK sub-block as shown in Formulation 5.

Different from the consensus variant, the original optimization problem is separated into:

Contact dynamics sub-block:

$$\begin{aligned}\mathcal{L}_{\text{cd}} = & \sum_{i=0}^{N}(\delta\mathbf{F}[i]^T \mathbf{Q}_F \, \delta\mathbf{F}[i] + \mathbf{u}[i]^T \mathbf{R}\mathbf{u}[i]) \\ & + \frac{\rho_j}{2}\|\mathbf{x}_M - \hat{\mathbf{x}}_M + \mathbf{v}_{\text{ik}}^k\|_2^2 + \frac{\rho_j}{2}\|\mathbf{x}_M - \bar{\mathbf{x}}_M + \mathbf{v}_j^k\|_2^2 \\ & + \frac{\rho_u}{2}\|\mathbf{u} - \bar{\mathbf{u}} + \mathbf{v}_u^k\|_2^2 + \frac{\rho_f}{2}\|\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}} + \mathbf{v}_f^k\|_2^2 \end{aligned}$$

Inverse kinematics sub-block:

$$\begin{aligned}\mathcal{L}_{\text{ik}} = & \sum_{i=0}^{N} W_p \left\| \mathcal{FK}(\hat{\mathbf{x}}_M[i]) - \mathbf{x}_e^d[i] \right\|_2 \\ & + \frac{\rho_j}{2}\|\mathbf{x}_M - \hat{\mathbf{x}}_M + \mathbf{v}_{\text{ik}}^k\|_2^2 \end{aligned}$$

Formulation 5 Distributed Constrained Optimization (Sequential)

$$\begin{aligned}
&\text{(Tracking Task)} && \min_{\phi,\hat{\phi},\bar{\phi}} \sum_{i=0}^{N} \delta\mathbf{F}[i]^T\mathbf{Q}_F\,\delta\mathbf{F}[i] + \mathbf{u}[i]^T\mathbf{R}\mathbf{u}[i] \\
& && \quad + W_p\big\|\mathcal{FK}(\mathbf{x}_M^c[i]) - \mathbf{x}_e^d[i]\big\|_2^2 \\
& && \quad + I_{\mathcal{J},\mathcal{U},\mathcal{F}}(\bar{\mathbf{x}}_M[i], \bar{\mathbf{u}}[i], \bar{\boldsymbol{\lambda}}[i]) \\
&\text{(Variables-DDP)} && \boldsymbol{\phi}[i] = [[\overbrace{\mathbf{x}_M[i]\ \dot{\mathbf{x}}_M[i]\ \mathbf{F}_e[i]}^{\mathbf{x}[i]}]^T, \mathbf{u}[i]^T]^T \\
&\text{(Variables-IK)} && \hat{\boldsymbol{\phi}}[i] = \hat{\mathbf{x}}_M[i]^T \\
&\text{(Variables-Proj)} && \bar{\boldsymbol{\phi}}[i] = [\bar{\mathbf{x}}_M[i]^T, \bar{\mathbf{u}}[i]^T, \bar{\boldsymbol{\lambda}}[i]^T]^T \\
& && \forall i = 0, 1, \ldots, N-1 \qquad \text{(21a)} \\
&\text{(Dynamics)} && \text{s.t.}\quad \mathbf{x}[i+1] = \mathcal{F}(\mathbf{x}[i], \mathbf{u}[i]) \qquad \text{(21b)} \\
&\text{(Initial Condition)} && \mathbf{x}[0] = \mathbf{x}_0 \qquad \text{(21c)} \\
&\text{(Consistency Constraints)} && \begin{cases} \mathbf{x}_M = \hat{\mathbf{x}}_M \\ \mathbf{x}_M = \bar{\mathbf{x}}_M \\ \mathbf{u} = \bar{\mathbf{u}} \\ \boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}} \end{cases} \qquad \text{(21d)}
\end{aligned}$$

Projection sub-block:

$$\begin{aligned}
\mathcal{L}_{\text{proj}} = &\sum_{i=0}^{N} I_{\mathcal{J},\mathcal{U},\mathcal{F}}(\bar{\mathbf{x}}_M[i], \bar{\mathbf{u}}[i], \bar{\boldsymbol{\lambda}}[i]) \\
&+ \frac{\rho_j}{2}\|\mathbf{x}_M - \bar{\mathbf{x}}_M + \mathbf{v}_j^k\|_2^2 + \frac{\rho_u}{2}\|\mathbf{u} - \bar{\mathbf{u}} + \mathbf{v}_u^k\|_2^2 \\
&+ \frac{\rho_f}{2}\|\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}} + \mathbf{v}_f^k\|_2^2
\end{aligned}$$

References

[1] M. Solis, "New Frontiers in Robotic Surgery: The latest high-tech surgical tools allow for superhuman sensing and more," *IEEE pulse*, vol. 7, no. 6, pp. 51–55, 2016.

[2] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. W. Kim, "Supervised autonomous robotic soft tissue surgery," *Science Translational Medicine*, vol. 8, no. 337, p. 337ra64, 2016.

[3] R. Elandt, E. Drumwright, M. Sherman, and A. Ruina, "A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects," *arXiv preprint arXiv:1904.11433*, 2019.

[4] G. Gilardi and I. Sharf, "Literature survey of contact dynamics modelling," *Mechanism and Machine Theory*, vol. 37, pp. 1213–1239, 10 2002.

[5] K. L. Johnson, *Contact Mechanics*. Cambridge: Cambridge University Press, 1985.

[6] 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.

[7] 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.

[8] Z. Manchester and S. Kuindersma, "Variational contact-implicit trajectory optimization," in *Robotics Research*, pp. 985–1000, Springer, 2020.

[9] A. Patel, S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler, "Contact-implicit trajectory optimization using orthogonal collocation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2242–2249, 2019.

[10] J.-P. Sleiman, J. Carius, R. Grandia, M. Wermelinger, and M. Hutter, "Contact-implicit trajectory optimization for dynamic object manipulation," *Macau, China*, 2019.

[11] 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, 2016.

[12] 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.

[13] 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," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[14] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2703–2710, IEEE, 2016.

[15] Z. Zhou and Y. Zhao, "Accelerated ADMM based Trajectory Optimization for Legged Locomotion with Coupled Rigid Body Dynamics," in *American Control Conference*, 2020.

[16] L. Wijayarathne and F. L. Hammond, "Identification of Compliant Contact Parameters and Admittance Force Modulation on a Non-stationary Compliant Surface," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6826–6832, 2020.

[17] K. Pereida and A. P. Schoellig, "Data-Efficient Multirobot , Multitask Transfer," vol. 3, no. 2, pp. 1260–1267, 2018.

[18] M. Parigi Polverini, S. Formentin, L. Merzagora, and P. Rocco, "Mixed Data-Driven and Model-Based Robot Implicit Force Control: A Hierarchical Approach," *IEEE Transactions on Control Systems Technology*, pp. 1–14, 2019.

[19] C. Wang, S. Wang, B. Romero, F. Veiga, and E. Adelson, "Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5633–5640, IEEE, 2020.

[20] Y. Han, R. Batra, N. Boyd, T. Zhao, Y. She, S. Hutchinson, and Y. Zhao, "Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer," *arXiv preprint arXiv:2112.06374*, 2021.

[21] L. Wijayarathne, Q. Sima, Z. Zhou, Y. Zhao, F. L. H. Iii, and I. Member, "Simultaneous Trajectory Optimization and Force Control with Soft Contact Mechanics,"

[22] R. Schindeler and K. Hashtrudi-Zaad, "Online Identification of Environment Hunt-Crossley Models Using Polynomial Linearization," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 447–458, 2018.

[23] T. Lefebvre, S. Dutré, J. de Schutter, H. Bruyninckx, J. Katupitiya, J. de Geeter, and S. Demey, "Estimating First-Order Geometric Parameters and Monitoring Contact Transitions during Force-Controlled Compliant Motion," *The International Journal of Robotics Research*, vol. 18, no. 12, pp. 1161–1184, 2003.

[24] D. W. Marhefka and D. E. Orin, "A compliant contact model with nonlinear damping for simulation of robotic systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 29, no. 6, pp. 566–572, 1999.

[25] A. Pappalardo, A. Albakri, C. Liu, L. Bascetta, E. De Momi, and P. Poignet, "Hunt-Crossley model based force control for minimally invasive robotic surgery," *Biomedical Signal Processing and Control*, vol. 29, pp. 31–43, 2016.

[26] J. Sun, N. Lam, L. Zhang, D. Ruan, and E. Gad, "International Journal of Impact Engineering Short Communication A note on Hunt and Crossley model with generalized visco-elastic damping," *International Journal of Impact Engineering*, vol. 121, no. May, pp. 151–156, 2018.

[27] R. D. Howe and M. R. Cutkosky, "Practical force-motion models for sliding manipulation," *International Journal of Robotics Research*, vol. 15, no. 6, pp. 557–572, 1996.

[28] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, 1986.

[29] X. Mu, Y. Xue, and Y. B. Jia, "Robotic cutting: Mechanics and control of knife motion," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 3066–3072, 2019.

[30] N. Fazeli, R. Kolbert, R. Tedrake, and A. Rodriguez, "Parameter and contact force estimation of planar rigid-bodies undergoing frictional contact," *International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1437–1454, 2017.

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

[32] D. H. Jacobson and D. Q. Mayne, "Differential dynamic programming," 1970.

[33] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*, vol. 19. Siam, 2010.

[34] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and Stabilization of Complex Behaviors through Online Trajectory Optimization,"

[35] J. Koenemann, A. D. Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, N. Mansard, A. D. Prete, Y. Tassa, O. Stasse, and N. Mansard, "Whole-body Model-Predictive Control applied to the HRP-2 Humanoid," p. 1137021, 2015.

[36] Y. Tassa, N. Mansard, and E. Todorov, "Control-Limited Differential Dynamic Programming,"

[37] Z. Xie, C. K. Liu, and K. Hauser, "Differential dynamic programming with nonlinear constraints," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 695–702, IEEE, 5 2017.

[38] B. O'Donoghue, G. Stathopoulos, and S. Boyd, "A splitting method for optimal control," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2432–2442, 2013.

[39] Z. Zhao, Z. Zhou, M. Park, and Y. Zhao, "Sydebo: Symbolic-decision-embedded bilevel optimization for long-horizon manipulation in dynamic environments," *IEEE Access*, vol. 9, pp. 128817–128826, 2021.

[40] B. O'donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 1042–1068, 2016.

[41] V. Sindhwani, R. Roelofs, and M. Kalakrishnan, "Sequential operator splitting for constrained nonlinear optimal control," in *2017 American Control Conference (ACC)*, pp. 4864–4871, IEEE, 2017.

[42] G. Gilardi and I. Sharf, "Literature survey of contact dynamics modelling," *Mechanism and machine theory*, vol. 37, no. 10, pp. 1213–1239, 2002.

[43] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini, "Stance: Locomotion adaptation over soft terrain," *IEEE Transactions on Robotics*, 2020.

[44] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 577–584, IEEE, 2017.

[45] R. Cortesao and P. Poignet, "Motion compensation for robotic-assisted surgery with force feedback," in *2009 IEEE International Conference on Robotics and Automation*, pp. 3464–3469, IEEE, 5 2009.

[46] M. Frigerio, J. Buchli, D. G. Caldwell, and C. Semini, "RobCoGen : a code generator for efficient kinematics and dynamics of articulated robots , based on Domain Specific Languages," *Journal of Software Engineering for Robotics*, vol. 7, no. July, pp. 36–54, 2016.

[47] S. Kleff, A. Meduri, R. Budhiraja, N. Mansard, S. Kleff, A. Meduri, R. Budhiraja, N. Mansard, and L. Righetti, "High-Frequency Nonlinear Model Predictive Control of a Manipulator To cite this version : HAL Id : hal-02993058 High-Frequency Nonlinear Model Predictive Control of a Manipulator," 2020.

[48] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified Impedance and Admittance Control," in *2010 IEEE International Conference on Robotics and Automation*, pp. 554–561, IEEE, 5 2010.

[49] W. S. Newman, "Stability and Performance Limits of Interaction Controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 114, pp. 563–570, 12 1992.

[50] L. Wijayarathne, Q. Sima, Z. Zhou, Y. Zhao, and F. L. Hammond, "Simultaneous trajectory optimization and force control with soft contact mechanics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3164–3171, IEEE, 2020.

[51] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[52] "Kuka sunrise.connectivity fri 1.7 v3 manual."

[53] H. Kazerooni, "On the Robot Compliant Motion Control," vol. 111, no. September, 1989.

[54] M. Giftthaler, M. Neunert, M. Stäuble, M. Frigerio, C. Semini, and J. Buchli, "Automatic differentiation of rigid body dynamics for optimal control and estimation," *Advanced Robotics*, vol. 31, pp. 1225–1237, 11 2017.

[55] Y. Yang, K. Caluwaerts, A. Iscen, T. Zhang, J. Tan, and V. Sindhwani, "Data Efficient Reinforcement Learning for Legged Robots," no. CoRL, pp. 1–10, 2019.

[56] W. S. Newman, "Stability and Performance Limits of Interaction Controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 114, pp. 563–570, 12 1992.

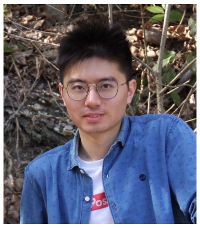
[57] C. Park, J. Pan, and D. Manocha, "High-dof robots in dynamic environments using incremental trajectory optimization," *International Journal of Humanoid Robotics*, vol. 11, no. 02, p. 1441001, 2014.

[58] B. Ichter, B. Landry, E. Schmerling, and M. Pavone, "Perception-aware motion planning via multiobjective search on gpus," in *Robotics Research*, pp. 895–912, Springer, 2020.

[59] B. Plancher and S. Kuindersma, "A performance analysis of parallel differential dynamic programming on a gpu.," in *WAFR*, pp. 656–672, 2018.

[60] W. Agboh, D. Ruprecht, and M. Dogar, "Combining coarse and fine physics for manipulation using parallel-in-time integration," in *ISRR 2019 Springer Tracts in Advanced Robotics*, Springer, 2019.

Lasitha Wijayarathne Lasitha Wijayarathne received B.S. and M.S. degrees in mechanical engineering from Georgia Institute of Technology, Atlanta, GA, USA. in 2015 and 2019. He is currently a Ph.D. candidate in robotics at Georgia Institute of Technology. His research interests centers around robot design optimization, manipulation, contact-rich trajectory planning on robotic systems.



Ziyi Zhou Ziyi Zhou received the B.S. degree in automation from Northeastern University, Shenyang, China in 2018 and the M.S. degree in electrical and computer engineering from Georgia Institute of Technology in 2021. He is currently pursuing a Ph.D. degree in electrical and computer engineering at Georgia Institute of Technology with a focus on robotics. His research interests centers around contact-rich trajectory optimization and task planning applied on legged robots.

Ye Zhao Ye Zhao (Member, IEEE) received the Ph.D. degree in mechanical engineering from The University of Texas at Austin, Austin, TX, USA, in 2016. He was a Post-Doctoral Fellow with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. He is currently an Assistant Professor with the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include robust task and motion planning, contact-rich trajectory optimization, legged locomotion and manipulation.

Dr. Zhao serves as an Associate Editor of IEEE Robotics and Automation Letters (RA-L) and IEEE Control Systems Letters (L-CSS). He was a Co-Chair of the IEEE Robotics and Automation Society (RAS) Student Activities Committee and an ICT Chair of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Frank L. Hammond III Frank L. Hammond III (Member, IEEE) received a B.S. in electrical engineering from Drexel University in 2002, M.S degrees in electrical and mechanical engineering from the University of Pennsylvania in 2006, and a Ph.D. in mechanical engineering from Carnegie Mellon University in 2010. He worked as a postdoctoral research follow at Harvard University from 2010-2014 and a postdoctoral researcher at MIT from 2014-2015. He is currently an Assistant Professor of mechanical and biomedical engineering at the Georgia Institute of Technology. Dr. Hammond is a member American Society of Mechanical Engineering (ASME) and the Biomedical Engineering Society (EMBS). His research interests include the development of soft robotic actuators and sensors, fluidic circuits, wearable rehabiliation devices, teloperated maniupulators, and human augmentation.