

Online Multi-Contact Feedback Model Predictive Control for Interactive Robotic Tasks

Seo Wook Han¹, Maged Iskandar², Jino Lee^{2,3,†} and Min Jun Kim^{1,†}

Abstract—In this paper, we propose a model predictive control (MPC) that accomplishes interactive robotic tasks, in which multiple contacts may occur at unknown locations. To address such scenarios, we made an explicit contact feedback loop in the MPC framework. An algorithm called Multi-Contact Particle Filter with Exploration Particle (MCP-EP) is employed to establish real-time feedback of multi-contact information. Then the interaction locations and forces are accommodated in the MPC framework via a spring contact model. Moreover, we achieved real-time control for a 7 degrees of freedom robot without any simplifying assumptions by employing a Differential-Dynamic-Programming algorithm. We achieved 6.8kHz, 1.9kHz, and 1.8kHz update rates of the MPC for 0, 1, and 2 contacts, respectively. This allows the robot to handle unexpected contacts in real time. Real-world experiments show the effectiveness of the proposed method in various scenarios.

I. INTRODUCTION

Physical interaction capabilities are crucial in interactive robotic tasks. Especially for collaborative robots, which are designed to share a workspace with humans, there is an increased need for effective handling of physical interactions. For example, table polishing requires the robot to accomplish hybrid motion-force control. Moreover, human-robot collaboration scenarios may involve multiple unknown contacts, which need to be safely managed in such a way that the interaction force is maintained below a dangerous level.

There have been a number of attempts to enable robots to perform such interactive tasks. For example, a hybrid control method that simultaneously realizes both motion and force control has been proposed in [1]. Additionally, to ensure safe interaction, compliant behavior of robots through impedance or admittance control has been implemented in [2]–[8].

With recent advancements in optimization techniques, there have been attempts to address interactive robotic tasks using model predictive control (MPC). By utilizing the MPC, such interaction tasks can be accomplished through cost function designs [9], while imposing constraints on the state and input variables [10]–[12].

This work was supported by ITECH R&D program of MOTIE/KEIT under Project No. 20014398 and No. 20014485, and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (Project No. 2021R1C1C1005232 and No. 2021R1A4A3032834).

¹The authors are with Intelligent Robotic Systems Laboratory, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea. E-mail: tjdnr7117, minjun.kim@kaist.ac.kr

²The authors are with Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. E-mail: first.name.last.name@dlr.de

³The author is also with the Department of Mechanical Engineering, KAIST, Daejeon, Republic of Korea.

[†]Co-corresponding authors.

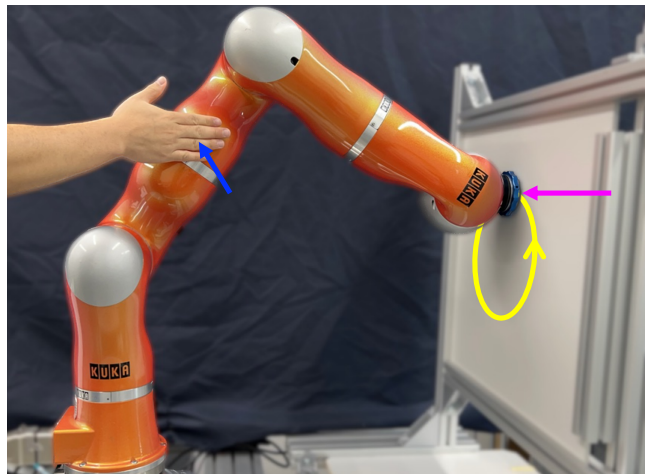


Fig. 1. The simultaneous accomplishment of hybrid motion-force control at the end-effector (yellow and magenta colored arrows) and safe handling of unexpected contact caused by human interaction (blue colored arrow).

To use the MPC framework for interaction tasks, a contact model should be defined to predict the contact-involved robot motion in the prediction horizon. While approaches based on kinematic constraints show promising results [13], [14], other types of contact models are also being explored. Particularly, a spring contact model, in which force feedback can be incorporated, is often employed [15]–[23]; e.g., interaction tasks are realized via admittance or impedance behavior of the robot with constraints on the contact force and robot state [15]–[17], and studies in [19]–[23] have reported successful results for hybrid motion-force control.

Nevertheless, the aforementioned methods are valid based on a common assumption that the contact occurs at a single and known location, typically at the end-effector where a force/torque (F/T) sensor is installed. This assumption seems reasonable in a well-structured workspace because all contacts can be planned in advance. However, in an unstructured workspace, such assumptions clearly have limitations. For example, in scenarios that involve physical interaction with humans or with unknown obstacles, unexpected contact may occur at multiple locations on the robot.

To tackle the problems of unknown contacts, algorithms that simultaneously estimate the contact location and force have been proposed [24]–[26]. Some studies have presented control methods that can safely handle a single contact using those algorithms [27], [28]. For example, authors in [27] use a kinematic constraint-based contact model, to generate a path that maintains the contact force below the pre-specified value, and in [28], a collision avoidance path is generated

along the force direction. Moreover, the safe handling of multiple contacts has been demonstrated using tactile sensors [29], [30]. For example, authors in [31], [32] utilize tactile sensor feedback to minimize resulting multiple contact forces in contact-rich environments. However, it is not straightforward how to handle unknown multiple contacts, occurring outside the tactile sensor, although such capability is crucial in accomplishing complex interaction tasks.

Accordingly, this paper proposes a *contact-feedback MPC framework* that addresses complex interaction tasks in which *multiple contacts* may occur *at arbitrary locations* (see, e.g., Fig. 1). To the best of our knowledge, this is the first attempt that embeds multiple unknown contacts in the MPC framework. The multi-contact feedback is provided through our previous work called Multi-Contact Particle Filter with Exploration Particles (MCP-EP) [26]. This method is capable of simultaneously estimating multiple contact locations and forces, regardless of the contact links. The outline of this algorithm is described in Section II-B.

Compared to the prior work, the proposed multi-contact feedback MPC framework offers several advantages: (i) By making an explicit contact feedback loop using MCP-EP, force control can be accomplished even in scenarios with multiple contacts regardless of contact locations¹. (ii) By employing the Differential Dynamic Programming (DDP) algorithm, the proposed MPC operates in real-time for a 7 degrees of freedom (DOF) manipulator without any simplifying assumptions on the robot model, which are common in literature [16]–[20], [27], [28], [31], [32]. For example, some of the time derivatives are assumed to be constant over the prediction horizon in [19], [20], and [27] employs a quasi-static robot assumption. Additionally, a linearized robot model is employed in [17], [32], and the robot joints are assumed to be perfectly velocity controlled in [16], [28], or position controlled in [31]. In fact, we were able to achieve 6.8kHz, 1.9kHz, and 1.8kHz when there are 0, 1, and 2 contacts, respectively. This feature enables the handling of multiple contacts in real-time, as illustrated in Fig. 1.

This paper is organized as follows. Section II presents the robot system and contact model. The proposed contact-feedback MPC framework is presented in Section III. In Section IV, the proposed MPC framework is validated through real-world experiments with a DLR-KUKA LWR IV+.

II. SYSTEM AND CONTACT MODELING

To predict the robot’s motion over the prediction horizon of the MPC, the system model is given as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)), \quad \mathbf{x}(0) = \tilde{\mathbf{x}}, \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state of the system, \mathbf{f} is the contact-involved robot dynamics, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the control input, and $\tilde{\bullet}$ represents the initial value of \bullet , while $\boldsymbol{\lambda}(t)$ represents k contact forces, i.e., $\boldsymbol{\lambda}(t) = \{\boldsymbol{\lambda}_1(t), \dots, \boldsymbol{\lambda}_k(t)\} \in \mathbb{R}^{3 \times k}$, of

which the i^{th} element can be expressed as follows:

$$\boldsymbol{\lambda}_i(t) = \mathbf{g}(\mathbf{x}(t); \boldsymbol{\theta}_i), \quad \boldsymbol{\theta}_i = \mathbf{h}(\tilde{\mathbf{r}}_{c,i}, \tilde{\boldsymbol{\lambda}}_i), \quad \forall i = \{1, 2, \dots, k\}, \quad (2)$$

where the subscript i denotes the index of k contacts, \mathbf{g} is the spring contact model, $\boldsymbol{\theta}_i$ is the parameter required for the spring contact model, $\mathbf{r}_{c,i}(t) \in \mathbb{R}^3$ is the position vector of the i^{th} contact point, and \mathbf{h} represents the procedure of computing $\boldsymbol{\theta}_i$ based on the i^{th} contact feedback ($\tilde{\mathbf{r}}_{c,i}$ and $\tilde{\boldsymbol{\lambda}}_i$). For clarity, $\mathbf{r}_{c,i}(t)$ and $\boldsymbol{\lambda}_i(t)$ are represented with respect to the world frame, if not specified otherwise.

We point out that the contact model (2) does not depend on the control input. This type of contact model allows the MPC framework to incorporate contact force feedback by avoiding an algebraic loop between $\boldsymbol{\lambda}_i(t)$ and $\mathbf{u}(t)$ [13]. Details of the contact model are described in Section II-C. For the sake of brevity, the time argument of quantities is omitted in the following.

A. Contact-involved robot dynamics

Consider the following articulated multi-body dynamics:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_c + \boldsymbol{\tau}_{ext}, \quad (3)$$

where $\mathbf{q} \in \mathbb{R}^n$ is the joint variable, $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis/centrifugal matrix, $\mathbf{g}(\mathbf{q})$ is the gravity vector, $\boldsymbol{\tau}_c$ is the commanded joint torque, and $\boldsymbol{\tau}_{ext}$ is the external joint torque caused by the contact forces.

Since we consider scenarios with k contacts, the external joint torque $\boldsymbol{\tau}_{ext}$ is given as

$$\boldsymbol{\tau}_{ext} = \sum_{i=1}^k \mathbf{J}_i^T(\mathbf{q}, \mathbf{r}_{c,i}) \boldsymbol{\lambda}_i, \quad (4)$$

where $\mathbf{J}_i(\mathbf{q}, \mathbf{r}_{c,i}) \in \mathbb{R}^{3 \times n}$ is the associated positional Jacobian matrix at the i^{th} contact point. Using (3) and (4), we can express the contact-involved robot dynamics of (1) by setting $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{u} = \boldsymbol{\tau}_c$.

B. Multi-contact feedback

The initial states of the multiple contacts ($\tilde{\mathbf{r}}_{c,i}$ and $\tilde{\boldsymbol{\lambda}}_i$) are required for the contact model in (2). Accurate real-time estimation of these variables is essential to handle unexpected contacts. In our previous work [26], we proposed a contact estimation algorithm, MCP-EP. This algorithm can estimate the multiple contacts in a robot manipulator using proprioceptive sensors, e.g., an F/T sensor mounted at the base and joint torque sensors (JTSs). MCP-EP is suitable for providing contact feedback for the following reasons: (i) The run-time of MCP-EP is 2200Hz for single-contact and 600Hz for dual-contact; (ii) The average root mean squared error (RMSE) of the contact location is 0.16cm, and that of the contact force is 0.01N, for single-contact. For dual-contact, these values are 1.08cm and 2.00N, respectively. These numbers are based on a quantitative evaluation of the algorithm in a simulation. Given these capabilities, in the remainder of this paper, we consider $\tilde{\mathbf{r}}_{c,i}$ and $\tilde{\boldsymbol{\lambda}}_i$ to be available.

It is worth noting that the proposed MPC does not strictly rely on MCP-EP, while one can employ any method as long

¹To avoid misunderstanding, it should be mentioned that the MCP-EP algorithm can estimate up to one contact per link.

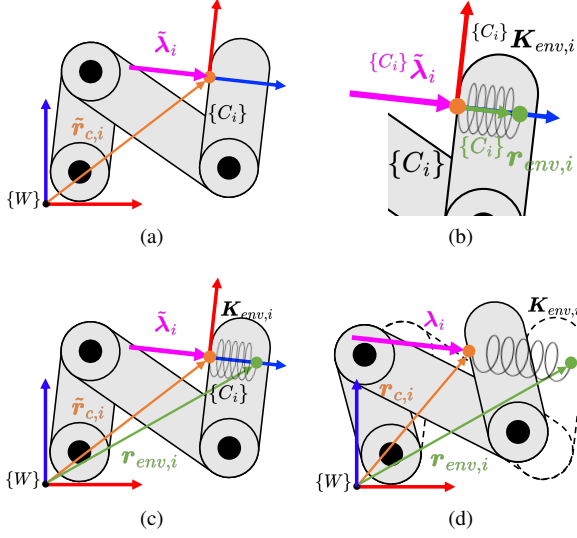


Fig. 2. Red and blue colored arrows indicate the x and z axes, respectively, of the frame. (a) The i^{th} contact frame $\{C_i\}$ is obtained using the contact feedback $\tilde{r}_{c,i}$ and $\tilde{\lambda}_i$. (b) $\{C_i\}r_{env,i}$ is computed using (7) within the contact frame. (c) $r_{env,i}$ and $K_{env,i}$ are calculated by transforming coordinates from $\{C_i\}$ to $\{W\}$. (d) $r_{c,i}$ is fixed on the robot surface (see the orange dot representing $r_{c,i}$). The predictive contact force is derived from $\lambda_i = g(\mathbf{x}; \theta_i)$.

as multi-contact information is provided in real time. The remaining step is to define the contact model and θ_i in (2) using the contact feedback.

C. Spring contact model

We use the following simple spring model, in which the i^{th} contact force is determined by the stiffness and deformation of the environment:

$$\lambda_i = K_{env,i} \Delta r_i, \quad (5)$$

where $K_{env,i} \in \mathbb{R}^{3 \times 3}$ and $\Delta r_i = r_{env,i} - r_{c,i} \in \mathbb{R}^3$ are the i^{th} environment stiffness matrix and deformation, respectively, and $r_{env,i}$ is the i^{th} environment location. The variables associated with the environment are defined as $\theta_i = \{K_{env,i}, r_{env,i}\}$.

The procedure of computing θ_i is illustrated in Fig. 2. As shown in Fig. 2(a), the i^{th} contact frame $\{C_i\}$ is obtained using $\tilde{r}_{c,i}$ and $\tilde{\lambda}_i$. The origin of the $\{C_i\}$ coincides with $\tilde{r}_{c,i}$, and by convention, its z -axis aligns with $\tilde{\lambda}_i$. In the following, variables expressed in $\{C_i\}$ are denoted with a left superscript. We define the i^{th} environment stiffness matrix with respect to the contact frame by

$$\{C_i\}K_{env,i} = \text{diag}(0, 0, k_{env,i}), \quad (6)$$

where $\text{diag}(\cdot)$ represents a diagonal matrix with the given values on its diagonal, and $k_{env,i} \in \mathbb{R}^+$ is the i^{th} environment stiffness. Since only z -directional stiffness exists in (6), any movement of the contact point within the xy -plane of the contact frame does not change the force. This implies that an unconstrained motion can be achieved in the tangential direction of the contact force. Subsequently, as shown in Fig. 2(b), $\{C_i\}r_{env,i}$ is obtained using

$$\{C_i\}r_{env,i} = \{C_i\}K_{env,i}^{-1} \{C_i\}\tilde{\lambda}_i. \quad (7)$$

Lastly, θ_i is derived from the coordinate transformation from the i^{th} contact frame to the world frame (see Fig. 2(c)). Given that R_i represents the rotation matrix for such transformation, $\theta_i = \{K_{env,i}, r_{env,i}\}$ can be obtained as follows:

$$K_{env,i} = R_i \{C_i\}K_{env,i} R_i^T, \quad (8)$$

$$r_{env,i} = \tilde{r}_{c,i} + R_i \{C_i\}r_{env,i}. \quad (9)$$

With this setup, when $r_{c,i} = \tilde{r}_{c,i}$, the contact force computed with (5) and θ_i is the same as $\tilde{\lambda}_i$.

The remaining step is to determine the update law for the contact point ($r_{c,i}$). As depicted in Fig. 2(d), $r_{c,i}$ is fixed on the robot surface, i.e., $r_{c,i}(0) = \tilde{r}_{c,i}$, and it is updated with the robot's forward kinematics.

In summary, when the contact feedback is received, θ_i is computed with (6)-(9), $r_{c,i}$ is updated with robot's forward kinematics, and then the contact force is computed using (5).

III. CONTACT FEEDBACK MODEL PREDICTIVE CONTROL

Based on the robot system and contact model discussed in Section II, this section presents the proposed contact-feedback MPC.

A. Optimal control problem

The proposed contact-feedback MPC is based on iteratively solving the following discretized optimal control problem (OCP) with the time step Δt :

$$[\mathbf{X}^*, \mathbf{U}^*] = \underset{\mathbf{X}, \mathbf{U}}{\text{argmin}} \sum_{t=0}^{T-1} L(\mathbf{x}^{[t]}, \mathbf{u}^{[t]}) + L_f(\mathbf{x}^{[T]}) \quad (10a)$$

$$\text{s.t. } \mathbf{x}^{[t+1]} = \mathbf{F}(\mathbf{x}^{[t]}, \mathbf{u}^{[t]}, \boldsymbol{\lambda}), \quad \mathbf{x}^{[0]} = \tilde{\mathbf{x}}, \quad (10b)$$

$$\lambda_i^{[t]} = g(\mathbf{x}^{[t]}; \theta_i), \quad (10c)$$

$$\theta_i = h(\tilde{r}_{c,i}, \tilde{\lambda}_i), \quad \forall i = \{1, 2, \dots, k\}, \quad (10d)$$

$$\mathbf{x}^{[t]} \in \mathcal{X}, \quad \forall t = \{0, 1, \dots, T\}, \quad (10e)$$

$$\mathbf{u}^{[t]} \in \mathcal{U}, \quad \forall t = \{0, 1, \dots, T-1\}, \quad (10f)$$

where \mathbf{X} and \mathbf{U} denote the state and control input trajectories, e.g., $\mathbf{X} = \{\mathbf{x}^{[0]}, \mathbf{x}^{[1]}, \dots, \mathbf{x}^{[T]}\}$ and $\mathbf{U} = \{\mathbf{u}^{[0]}, \mathbf{u}^{[1]}, \dots, \mathbf{u}^{[T-1]}\}$, the superscript $*$ indicates that the variable has been optimized, L and L_f are the discretized running and terminal cost functions, respectively, and T is the prediction horizon length. Here, (10b) is Euler discretization of (1), (10c) and (10d) are the spring contact model described in Section II-C, and the state and control input constraints are considered in (10e) and (10f), respectively. Particularly, in our case, the joint angle, velocity, and control input limits are considered.

Equation (10) is solved iteratively, and the first element of the optimal control input trajectory, i.e., $\mathbf{u}^{[0]*}$, is used as a command for a torque-controlled robot. A diagram illustrating the proposed contact-feedback MPC framework is shown in Fig. 3, and more details are described in Section IV.

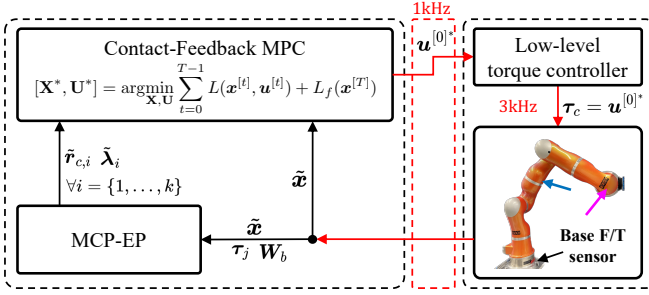


Fig. 3. τ_j and W_b are the JTSs and base F/T sensor measurements, respectively, and red arrows represent real-time communication. Utilizing the robot states (\tilde{x}) and contact information ($\tilde{r}_{c,i}$ and $\tilde{\lambda}_i$), the contact-feedback MPC (10) calculates \mathbf{X}^* and \mathbf{U}^* . Then, $\mathbf{u}^{[0]*}$ is commanded to the torque-controlled robot.

B. Numerical solution of OCP

As highlighted in Section I, we use the DDP algorithm for (10) to achieve a real-time MPC. Specifically, we implemented the method of Box-FDDP [11] to incorporate the spring contact model described in Section II-C. This algorithm allows us to impose box-constraints on the control input, e.g., $\mathbf{u}_{min} \leq \mathbf{u}^{[t]} \leq \mathbf{u}_{max}$. As a result, (10f) can be treated as hard constraints, whereas (10e) cannot. To address this limitation, we adopted a penalization approach for (10e). For example, when elements in $\mathbf{x}^{[t]}$ exceed the state constraints (\mathbf{x}_{min} , \mathbf{x}_{max}), a steeply increasing quadratic cost is added. This prevents exceeding the joint and velocity limits during operation.

Additionally, since the DDP algorithm is based on the gradient method, partial derivatives of the cost functions and the contact-involved robot dynamics (\mathbf{f}) are required. For faster computation, we derive these analytically rather than using finite-difference or auto-diff methods. Since $\mathbf{r}_{c,i}$ is fixed on the robot surface, the partial derivatives of the contact model (5) with respect to the state and control input are

$$\frac{\partial \lambda_i}{\partial \mathbf{q}} = -\mathbf{K}_{env,i} \mathbf{J}_i(\mathbf{q}, \mathbf{r}_{c,i}), \quad \frac{\partial \lambda_i}{\partial \dot{\mathbf{q}}} = \frac{\partial \lambda_i}{\partial \mathbf{u}} = \mathbf{0}. \quad (11)$$

Consequently, the partial derivatives of \mathbf{f} can be analytically derived by combining (11) with the partial derivatives of the robot forward dynamics that exclude the contact model [33].

C. Cost function design

In this subsection, we present our cost functions used in (10) with which the proposed MPC framework is able to accomplish interaction scenarios. For brevity, $c_\bullet \in \mathbb{R}^+$ represents the gain of each cost in the following. Consider the running and terminal cost functions defined by

$$l(\mathbf{x}, \mathbf{u}) = l_m(\mathbf{x}) + l_c(\lambda_i) + c_u \|\mathbf{u} - \mathbf{u}_0\|^2, \quad (12)$$

$$l_f(\mathbf{x}) = l_m(\mathbf{x}) + l_c(\lambda_i), \quad (13)$$

where l and l_f are continuous time version of L and L_f in (10), and $\|\mathbf{u} - \mathbf{u}_0\|^2$ is the penalization term for the control input. Here, l_c and l_m represent the costs for contact force and motion control, respectively.

1) *Motion control*: We use the following cost function to penalize the joint velocity and regulate the end-effector of the robot to the desired position and orientation:

$$l_m(\mathbf{x}) = c_v \|\dot{\mathbf{q}}\|^2 + c_p \|\mathbf{p}_{ee}(\mathbf{q}) - \mathbf{p}_{des}\|^2 + c_r \|\mathbf{R}_{ee}(\mathbf{q}) \ominus \mathbf{R}_{des}\|^2, \quad (14)$$

where $\mathbf{p}_{ee}(\mathbf{q})$ and $\mathbf{R}_{ee}(\mathbf{q})$ represent the position and orientation of the end-effector, respectively. Similarly, \mathbf{p}_{des} and \mathbf{R}_{des} represent the desired position and orientation, and \ominus denotes the difference in $\text{SO}(3)$. In particular we used $\mathbf{R}_1 \ominus \mathbf{R}_2 = \log(\mathbf{R}_1^T \mathbf{R}_2)$.

2) *Contact force control*: The cost for contact is designed to allow for defining different tasks on different links (e.g., contact force regulation on link 7 while force barrier on the other links). To this end, for notational convenience, the i^{th} contact link is denoted as $\gamma_i \in \{1, 2, \dots, n\}$.

To regulate the i^{th} contact force to the desired value ($\lambda_{i,des} \in \mathbb{R}^3$), the following cost function is used:

$$l_{c,reg}(\lambda_i, \gamma_{set}) = \begin{cases} c_\lambda \|\mathbf{A}_d(\lambda_i - \lambda_{i,des})\|^2 & \text{if } \gamma_i \in \gamma_{set} \\ 0 & \text{otherwise} \end{cases}, \quad (15)$$

where γ_{set} is the set of link indices used to indicate that this cost is activated, and $\mathbf{A}_d \in \mathbb{R}^{3 \times 3}$ determines the direction of the force control. For example, only the x -directional force is controlled with $\mathbf{A}_d = \text{diag}(1, 0, 0)$.

For safe handling of unexpected contacts, we limit the allowed contact force. To this end, the following force barrier with quadratic function is considered:

$$l_{c,bar}(\lambda_i, \gamma_{set}) = \begin{cases} c_\lambda (\|\lambda_i\| - \lambda_{i,max})^2 & \text{if } \|\lambda_i\| > b\lambda_{i,max} \text{ and } \gamma_i \in \gamma_{set} \\ 0 & \text{otherwise} \end{cases}, \quad (16)$$

where $b \in (0, 1)$ is the scale factor to smooth the constraint. As a result, the total cost for the contact $l_c(\mathbf{x})$ is set as a combination of (15) and (16) depending on the applications.

IV. EXPERIMENTS

The proposed multi contact-feedback MPC is validated in experiments on the DLR-KUKA LWR IV+ torque-controlled robot. The low-level torque controller runs at a rate of 3kHz. We use all seven joints of the robot, i.e., $n_x = 14$ and $n_u = 7$. To employ the MCP-EP algorithm that estimates contact locations and forces, an ATI Omega F/T sensor is mounted at the base of the robot. We implemented the proposed contact-feedback MPC in C++, and it operates at approximately 6.8kHz, 1.9kHz, and 1.8kHz when there are 0, 1, and 2 contacts, respectively. Since the contact-feedback MPC, MCP-EP, and low-level torque controller operate at different rates, each program is run with the latest available data. Additionally, the update rate of $\mathbf{u}^{[0]*}$ to the low-level torque controller and that of the sensor measurements (\tilde{x} , JTSs, base F/T sensor) are consistently set as 1kHz using the real-time setting (see Fig. 3).

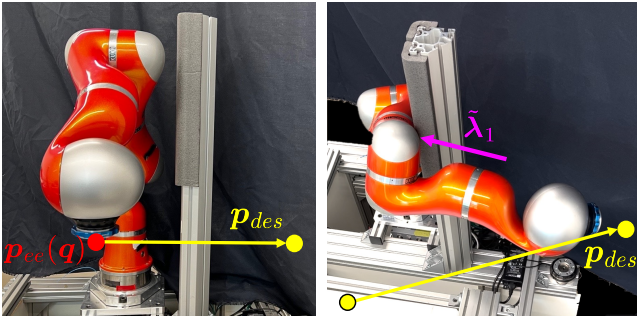


Fig. 4. Scenario #1: The current end-effector position is represented as a red dot, and the desired end-effector position trajectory is given by linear interpolation, as indicated by a yellow arrow. While tracking the trajectory, the robot's body makes contact with the obstacle.

A. OCP parameter selection

1) *Prediction horizon length*: In the following experiments, we use a relatively short planning horizon 125ms with $T = 5$ and $\Delta t = 25$ ms. It is well known that a longer planning horizon would improve control performance unless it significantly reduces the control frequency. Despite the fact that, in scenarios involving unexpected contacts, a long planning horizon would not be advantageous. This is because, as soon as contact occurs, the OCP (10) should be modified to include l_c in the cost function (10a) and to include the spring contact model in (10b). This change requires solving an entirely new OCP, distinct from the prior iteration. This can be problematic for the DDP algorithm, which relies on using the previous step's output as an initial value to enable warm-start. Specifically, with a long planning horizon, the state trajectory might deviate from the initial state. This can result in extremely large contact forces in the prediction horizon, leading to numerical issues.

2) *Environment stiffness*: When $k_{env,i}$ has a small value, the robot needs to move significantly within the prediction horizon to control the force. This can lead to large motion for the actual robot, potentially causing vibrations in practice. In contrast, when $k_{env,i}$ is too large, each MPC step causes small motion for the actual robot, which may result in too slow convergence of the force. Finding an appropriate balance between these two is crucial. In our implementation, we used $k_{env,i} = 3500$ N/m for all experiments.

Notice again that we have no pre-knowledge about the contacts; instead, the proposed MPC handles the unexpected contacts in real-time. In the following, two experimental scenarios are presented. All the experimental videos, including two additional experiments, can be found in the attached file (also see [34]).

B. Comparative study: with and without contact feedback

In scenario #1, we demonstrate the effectiveness of the contact feedback. To this end, for comparison purposes, we implemented the MPC without contact feedback, in which the OCP (10) does not include λ and l_c . For both cases (i.e., with and without contact feedback), the cost for motion control consists of the end-effector position regulation cost and the joint velocity penalization term; i.e., $c_p > 0$, $c_v > 0$ and $c_r = 0$ in (14).

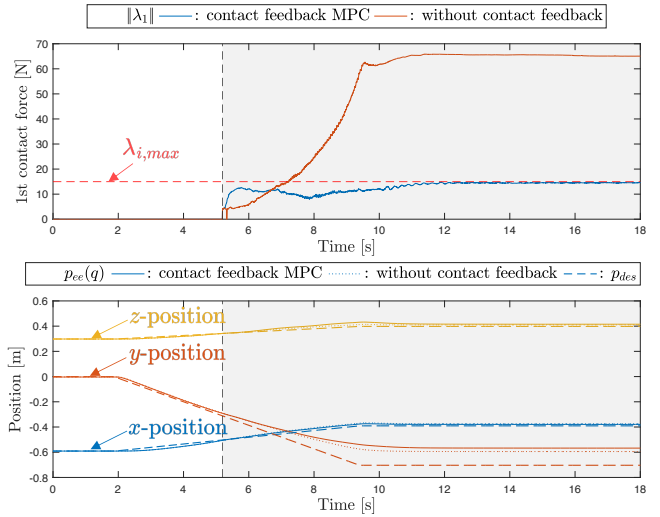


Fig. 5. Scenario #1: The region is shaded when the contact exists. **Top**: The magnitude of the contact force ($\|\tilde{\lambda}_1\|$). **Bottom**: The actual and desired end-effector trajectories (p_{ee} and p_{des}).

The experimental setup is illustrated in Fig. 4. The robot tracks a desired end-effector position trajectory (p_{des}) in the environment where an unknown obstacle is located close to the robot. p_{des} is set in such a way that the robot cannot reach due to the obstacle. During the trajectory tracking, a contact occurs on the 3rd link at 5.2s.

Fig. 5 shows the contact force ($\|\tilde{\lambda}_1\|$), the actual end-effector position ($p_{ee}(q)$), and the desired one (p_{des}). As can be seen in the figure, when there is no contact feedback, the magnitude of the contact force increases as the end-effector position error increases. The maximum contact force was 65.87N.

To maintain the interaction force small, the proposed method was implemented using (16):

$$l_c(x) = \sum_{i=1}^k l_{c,bar}(\lambda_i, \gamma_{set}), \quad (17)$$

where $\gamma_{set} = \{1, 2, \dots, 7\}$, and the upper limit of the contact force ($\lambda_{i,max}$) is set to 15N. With the contact feedback, the contact force was maintained below this limit. Over the entire time span, the maximum contact force was 14.74N.

C. Hybrid motion-force control under multi-contact

In scenario #2, the proposed MPC demonstrates hybrid motion-force control, while handling an unknown interaction. The experimental scenario is sequentially described in Fig. 6. The robot makes contact with the environment by moving the end-effector along the x -direction (see Fig. 6(a),(b)). As shown in Fig. 6(c), hybrid motion-force control is performed at the end-effector: the contact force in the x -direction is regulated to 20N while tracking a circular trajectory in yz -plane. In addition to the hybrid motion-force control at the end-effector, a human operator applies force on the robot, as shown in Fig. 6(d).

In this scenario, the proposed method is expected to handle unknown contact safely, while maintaining 20N at the end-effector. To this end, the cost for the contact is defined as

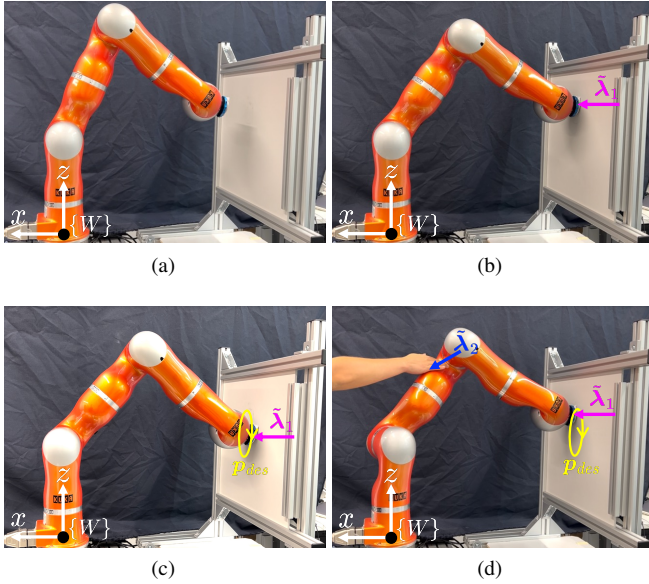


Fig. 6. Scenario #2: (a) The initial configuration of the robot, not in contact. (b) The robot moves in the x -direction and makes contact with the environment. (c) Hybrid motion-force control is performed at the end-effector. (d) While performing hybrid motion-force control, an additional contact is applied by a human.

follows:

$$l_c(\mathbf{x}) = \sum_{i=1}^k l_{c,bar}(\lambda_i, \gamma_{set,1}) + l_{c,reg}(\lambda_i, \gamma_{set,2}), \quad (18)$$

where $\gamma_{set,1} = \{1, 2, \dots, 6\}$, $\lambda_{i,max} = 15\text{N}$, $\gamma_{set,2} = \{7\}$, $\mathbf{A}_d = \text{diag}(1, 0, 0)$, and the first element of $\lambda_{i,des}$ is 20N .

To track the desired trajectory, a high gain was set for c_p , while a slightly lower value was allocated to the orientation part c_r . The joint velocity penalization term was also used; i.e., $c_v > 0$. Note that with the contact model presented in Section II-C, robot motion can be achieved in the unconstrained tangential direction of the contact force, allowing for hybrid motion-force control.

Similar to the previous, Fig. 7 shows the first contact force ($\tilde{\lambda}_1$), the second contact force ($\|\tilde{\lambda}_2\|$), the end-effector's actual position ($p_{ee}(\mathbf{q})$), and the desired position (p_{des}). At 3s, the end-effector makes contact with the environment. From 5.9s to 18.4s, only the hybrid motion-force control at the end-effector is performed without human interaction. In this case, the RMSE of the end-effector position was 0.86cm , and that of the contact force was 0.58N .

On the other hand, while tracking the second circular trajectory (from 19.4s to 31.9s), additional contact force ($\tilde{\lambda}_2$) is applied to the 3rd link by a human. As shown in Fig. 7, the robot immediately reacts to maintain the contact force below 15N . At the same time, the motion-force control at the end-effector performs reasonably. Specifically, at the end-effector, the RMSE for the position was 1.14cm and for the contact force was 0.89N .

Finally, from 33.5s to the end, the force regulation control is performed with a constant desired pose. The RMSE values for the end-effector position and contact force were 0.20cm and 0.10N , respectively.

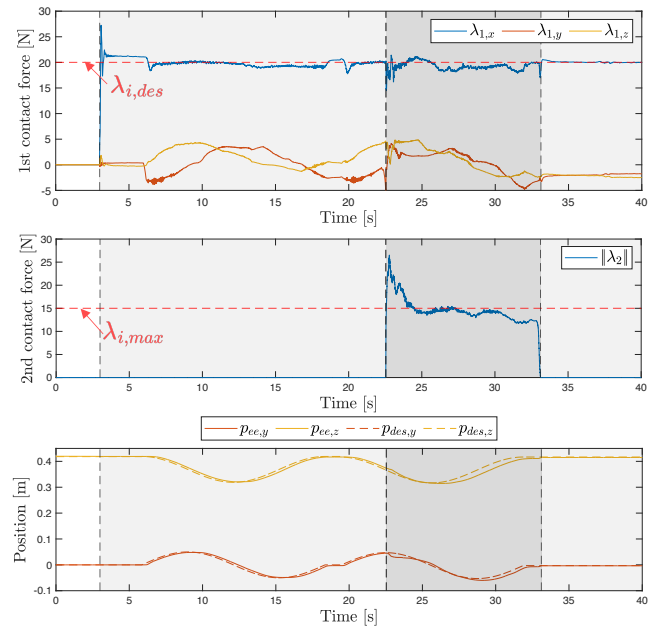


Fig. 7. Scenario #2: When there are two contacts, the region is shaded more intensely. **Top**: The contact force at the end-effector ($\tilde{\lambda}_1$). **Middle**: The magnitude of the contact force at the robot's body ($\|\tilde{\lambda}_2\|$). **Bottom**: The actual and desired end-effector trajectories (p_{ee} and p_{des}).

V. CONCLUSION AND FUTURE WORK

In this work, we propose a contact-feedback MPC that accomplishes interactive robotic tasks in which multiple contacts may occur at unknown locations. To accomplish such scenarios, we adopt a spring contact model to predict the contact-involved robot motion over the prediction horizon of the MPC. Here, the multi-contact information, that initializes the spring contact model, is provided by our previous work called MCP-EP. To the best of our knowledge, this paper presented the MPC that explicitly handles unknown multiple contacts for the first time. Furthermore, since the proposed method achieves fast enough runtime for a 7-DOF robot (update rate was 1.9kHz for the single-contact, and 1.8kHz for the dual-contact), the robot is able to handle unexpected contacts swiftly. The effectiveness of the proposed method is validated through the real-world experiments using a 7-DOF DLR-KUKA LWR IV+.

Although our approach has shown promising results, there remains room for improvement. When the contact feedback is received, the contacts remain throughout the entire prediction horizon of the OCP. However, since this contact model is derived from the contact feedback, its validity is limited only locally. For example, when employing a long horizon length, the issue with extremely high contact force can arise as discussed in Section IV-A.1. Moreover, the spring contact model used in this paper does not account for instances where the contact might vanish. Considering these issues could allow a longer prediction horizon in the OCP, potentially improving overall controller performance. Furthermore, to truly address the unexpected contacts, online determination of $k_{env,i}$ is essential, and therefore, this would be our future work as well.

REFERENCES

- [1] M. Iskandar, C. Ott, A. Albu-Schäffer, B. Siciliano, and A. Dietrich, "Hybrid force-impedance control for fast end-effector motions," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3931–3938, 2023.
- [2] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified impedance and admittance control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 554–561.
- [3] M. J. Kim, F. Beck, C. Ott, and A. Albu-Schäffer, "Model-free friction observers for flexible joint robots with torque measurements," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1508–1515, 2019.
- [4] J. Jeong, H. Mishra, C. Ott, and M. J. Kim, "A memory-based SO(3) parameterization: Theory and application to 6D impedance control with radially unbounded potential function," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8338–8344.
- [5] M. J. Kim, A. Werner, F. Loeffl, and C. Ott, "Passive impedance control of robots with viscoelastic joints via inner-loop torque control," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 584–598, 2022.
- [6] M. J. Kim, W. Lee, C. Ott, and W. K. Chung, "A passivity-based admittance control design using feedback interconnections," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 801–807.
- [7] A. Q. Keemink, H. van der Kooij, and A. H. Stienen, "Admittance control for physical human-robot interaction," *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1421–1444, 2018.
- [8] M. J. Kim, W. Lee, J. Y. Choi, G. Chung, K.-L. Han, I. S. Choi, C. Ott, and W. K. Chung, "A passivity-based nonlinear admittance control with application to powered upper-limb control under unknown environmental interactions," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 4, pp. 1473–1484, 2019.
- [9] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4906–4913.
- [10] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1168–1175.
- [11] C. Mastalli, W. Merkt, J. Marti-Saumell, H. Ferrolho, J. Solà, N. Mansard, and S. Vijayakumar, "A feasibility-driven approach to control-limited DDP," *Autonomous Robots*, vol. 46, no. 8, pp. 985–1005, 2022.
- [12] Y. Aoyama, G. Boutselis, A. Patel, and E. A. Theodorou, "Constrained differential dynamic programming revisited," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9738–9744.
- [13] S. Kleff, E. Dantec, G. Saurel, N. Mansard, and L. Righetti, "Introducing force feedback in model predictive control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 379–13 385.
- [14] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocodyl: An efficient and versatile framework for multi-contact optimal control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2536–2542.
- [15] K. J. Kazim, J. Bethge, J. Matschek, and R. Findeisen, "Combined predictive path following and admittance control," in *Annual American Control Conference (ACC)*, 2018, pp. 3153–3158.
- [16] A. Wahrburg and K. Listmann, "MPC-based admittance control for robotic manipulators," in *IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 7548–7554.
- [17] M. Bednarczyk, H. Omran, and B. Bayle, "Model predictive impedance control," in *IEEE international conference on robotics and automation (ICRA)*, 2020, pp. 4702–4708.
- [18] T. Gold, A. Völz, and K. Graichen, "Model predictive interaction control for industrial robots," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9891–9898, 2020.
- [19] —, "Model predictive position and force trajectory tracking control for robot-environment interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7397–7402.
- [20] —, "Model predictive interaction control for robotic manipulation tasks," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 76–89, 2022.
- [21] J. Matschek, J. Bethge, P. Zometa, and R. Findeisen, "Force feedback and path following using predictive control: Concept and application to a lightweight robot," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9827–9832, 2017.
- [22] J. Matschek, R. Jordanowa, and R. Findeisen, "Direct force feedback using Gaussian process based model predictive control," in *IEEE Conference on Control Technology and Applications (CCTA)*, 2020, pp. 8–13.
- [23] J. Matschek, J. Bethge, and R. Findeisen, "Safe machine-learning-supported model predictive force and motion control in robotics," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 6, pp. 2380–2392, 2023.
- [24] M. Iskandar, O. Eiberger, A. Albu-Schäffer, A. De Luca, and A. Dietrich, "Collision detection, identification, and localization on the dlr sara robot with sensing redundancy," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3111–3117.
- [25] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5062–5069.
- [26] S. W. Han and M. J. Kim, "Proprioceptive sensor-based simultaneous multi-contact point localization and force identification for robotic arms," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 099–12 105.
- [27] T. Pang and R. Tedrake, "Easing reliance on collision-free planning with contact-aware control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8375–8381.
- [28] A. Zube, J. Hofmann, and C. Frese, "Model predictive contact control for human-robot interaction," in *Proceedings of ISR: 47th International Symposium on Robotics*. VDE, 2016, pp. 1–7.
- [29] J. M. Gandarias, A. J. García-Cerezo, and J. M. Gómez-de Gabriel, "CNN-based methods for object recognition with high-resolution tactile sensors," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6872–6882, 2019.
- [30] J. Liang, J. Wu, H. Huang, W. Xu, B. Li, and F. Xi, "Soft sensitive skin for safety control of a nursing robot using proximity and tactile sensors," *IEEE Sensors Journal*, vol. 20, no. 7, pp. 3822–3830, 2020.
- [31] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *The International Journal of Robotics Research*, vol. 32, no. 4, pp. 458–482, 2013.
- [32] M. D. Killpack, A. Kapusta, and C. C. Kemp, "Model predictive control for fast reaching in clutter," *Autonomous Robots*, vol. 40, pp. 537–560, 2016.
- [33] J. Carpentier and N. Mansard, "Analytical derivatives of rigid body dynamics algorithms," in *Robotics: Science and systems (RSS)*, 2018.
- [34] KAIST IRSL, "Multi-Contact Feedback MPC for interactive robotic tasks (ICRA2024)," (Feb. 27, 2024). Accessed: Feb. 27, 2024. [Online Video]. Available: <https://youtu.be/pZbkHaxT1ao>