

Impact-Aware Bimanual Catching of Large-Momentum Objects

Lei Yan, Theodoros Stouraitis, Joao Moura, Michael Gienger, Sethu Vijayakumar

2024

Preprint:

This is an accepted article published in IEEE Transaction on Robotics. The final authenticated version is available online at:

<https://doi.org/10.1109/TRO.2024.3381551> Copyright 2024 IEEE

Impact-Aware Bimanual Catching of Large-Momentum Objects

Lei Yan^{*,1}, Theodoros Stouraitis^{*,2}, João Moura³, Wenfu Xu¹, Michael Gienger², and Sethu Vijayakumar³

Abstract—This paper investigates one of the most challenging tasks in dynamic manipulation—catching large-momentum moving objects. Beyond the realm of quasi-static manipulation, dealing with highly dynamic objects can significantly improve the robot’s capability of interacting with its surrounding environment. Yet, the inevitable motion mismatch between the fast moving object and the approaching robot will result in large impulsive forces, which lead to the unstable contacts and irreversible damage to both the object and the robot. To address the above problems, we propose an online optimization framework to: 1) estimate and predict the linear and angular motion of the object; 2) search and select the optimal contact locations across every surface of the object to mitigate impact through sequential quadratic programming (SQP); 3) simultaneously optimize the end-effector motion, stiffness, and contact force for both robots using multi-mode trajectory optimization (MMTO); and 4) realise the impact-aware catching motion on the compliant robotic system based on indirect force controller. We validate the impulse distribution, contact selection, and impact-aware MMTO algorithms in simulation and demonstrate the benefits of the proposed framework in real-world experiments including catching large-momentum moving objects with well-defined motion, constrained motion and free-flying motion.

Index Terms—Impact, large-momentum object, dynamic manipulation, bimanual catching, trajectory optimization.

I. INTRODUCTION

IMPACTS are inevitable when performing efficient contact-rich motions such as running, kicking a ball, catching and tossing objects. They are also inherent to highly dynamic dexterous manipulation [1] that goes beyond the realm of quasi-static problems [2]. Enabling robots with extra manipulation capabilities from prehensile throwing [3] and tossing [4], [5] to non-prehensile pushing [6] and batting [7] can boost task efficiency. Yet, when considering dynamic robot manipulation capabilities, such as prehensile intercepting [8] and soft

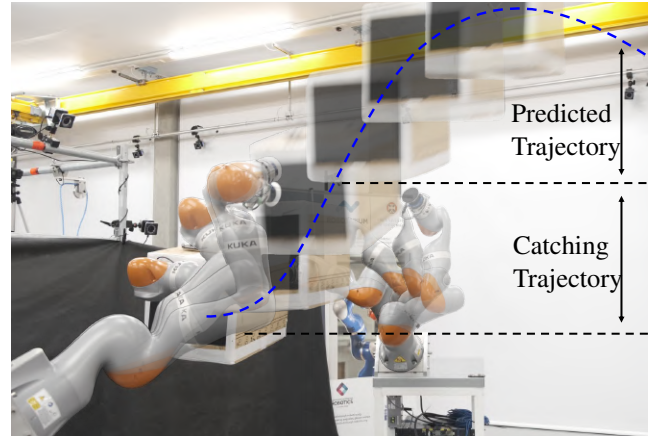


Fig. 1. Two KUKA-iiwa robots catching a flying large-momentum box that weighs 4.2 kg and travels with speed larger than 3.5 m/s.

catching [9], for heavy and large objects (see Fig. 1), non-prehensile manipulation and impacts handling are essential—as it is not possible to cage those large objects, due to their size, momentum, and the fact that contacts made at speed can break or slide at any point and time.

A. The challenging problems

This work focuses on impact-aware dynamic manipulation, and particularly the challenging problem of coping with impacts to catch large, heavy, fast and tumbling objects in motion with a dual-arm robotic system. This goes beyond the state of art on dynamic object manipulation, where either light flying objects were caught with a single arm (negligible impulses) [7], [9], [10] or heavy quasi-static objects were caught with a dual-arm robot (negligible impulses) [11], [12], [13]. For our problem, the objects travel with arbitrary orientation along projectile trajectories with a duration of less than one second and contact making is prone to impacts. For these tasks, simply intercepting the object [8], [14] is impractical, as the object cannot be caged and it can bounce off. Thus, for the robots to make stable contact with the object they need to follow its motion for at least a short duration of time. Furthermore, this following motion needs to coincide with the short time window where the trajectory of the large object and the dual-arm workspace intersect, as shown in Fig. 2.

Catching large fast-moving objects is extremely challenging, its success depends on many factors and requires the solution of the following three main challenging problems.

(C1) Estimating the object motion and predicting its future trajectory for ≈ 1 sec. This translates into: (i) using a small fragment of position and orientation data of the object to

^{*}Denotes equal contribution for both authors.

This work is supported by the National Natural Science Foundation of China (Grant No. 62203140), Shenzhen Excellent Scientific and Technological Innovation Talent Training Project (Grant No. RCBS20221008093122054 and RCJC20200714114436040). This work is also supported by EPSRC U.K. RAI Hub in Future AI and Robotics for Space (FAIR-SPACE: EP/R026092/1). (Corresponding author: Wenfu Xu.)

¹Lei Yan, Wenfu Xu are with the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, China, and also with Guangdong Provincial Key Laboratory of Intelligent Morphing Mechanisms Adaptive Robots, Key University Laboratory of Mechanism & Machine Theory and Intelligent Unmanned Systems of Guangdong. (e-mail: lei.yan@hit.edu.cn, wfxu@hit.edu.cn).

²Theodoros Stouraitis, Michael Gienger are with with Honda Research Institute Europe (HRI-EU), Germany. (e-mail: theostou@honda-ri.de, michael.gienger@honda-ri.de).

³João Moura, Sethu Vijayakumar are with the School of Informatics, University of Edinburgh, Edinburgh, U.K. (e-mail: jpousad@ed.ac.uk, sethu.vijayakumar@ed.ac.uk).

estimate its momentary linear and angular velocities, and (ii) using these estimates to predict the trajectory of a moving object with nonlinear rotation. Both the estimation and the prediction computations need to happen in a fraction ($<1/10$ sec) of the object's total flying motion.

To address (i) we utilize prior work [7], [15] and to address (ii) we solve an Initial Value Problem [16], which predicts the motion of the object by integrating its forward dynamics model in time [17], while considering environment constraints.

C2) Planing impact-aware dual-arm motion to catch the moving object. This involves the following considerations. (I) Establishing geometrically feasible and suitable contacts between the dual-arm robot and the moving, spinning object. This is far from trivial, as it depends on both geometry and motion. Hence, our first research question is (Q1) *how to optimally select contacts on the moving object?*

(II) Obtaining low impulsive forces while making contact at speed [18] to avoid breaking the recently established contact [19]. We study the influence of contact selection on the magnitude of the impulsive forces, which leads to the following key questions: (Q2) *how to minimize the impulsive force through contact selection?* and (Q3) *how to analyze the distribution of impulse along different directions?*

(III) Coordinating motion and stiffness variation of the two arms with the object motion while respecting the limited workspace. Intuitively to minimize impulsive forces, we aim to minimize the relative contact velocity and be "soft". Hence, we can frame the scientific question as: (Q4) *what is an appropriate model that captures the relation between stiffness, motion and impulsive force?*, along with the more practical question: (Q5) *how to jointly optimize the motion, stiffness and force of a robot to mitigate impact, while taking into account environment and system constraints, such as workspace limits?*

C3) Tracking motions and making contacts through impact in spite of estimation and execution errors. Making contact with a large-momentum object at non-zero speed has a direct influence on the continuity of contact and the robot dynamics [20]. This requires a controller able to track motion and force profiles through impact and to accommodate substantial disturbances (e.g. impulses exceeding the robot payload) resulting from inevitable estimation errors and motion mismatches. Hence, the design of robot controller is subject of research, which we summarize with the last question; (Q6) *what is an apt control scheme for tracking motions and making contact with fast-moving objects?*

Failing to answer any of the above six questions will result in a large impulsive force when making contact. As a result, the object will be likely dropped, damaged or bounce off after contact. Therefore, the aim of this paper is to decrease impulsive forces when making contact to enable impact-aware dynamic manipulation tasks with large-momentum objects.

B. Focus of our work and problem description

In this article, we investigate these six questions and propose a system with multiple modules. These are: (i) the *estimation* module where measurements of the object pose are collected and its velocity (linear and angular) is estimated using an EKF, (ii) the *prediction* module where the future states of the object

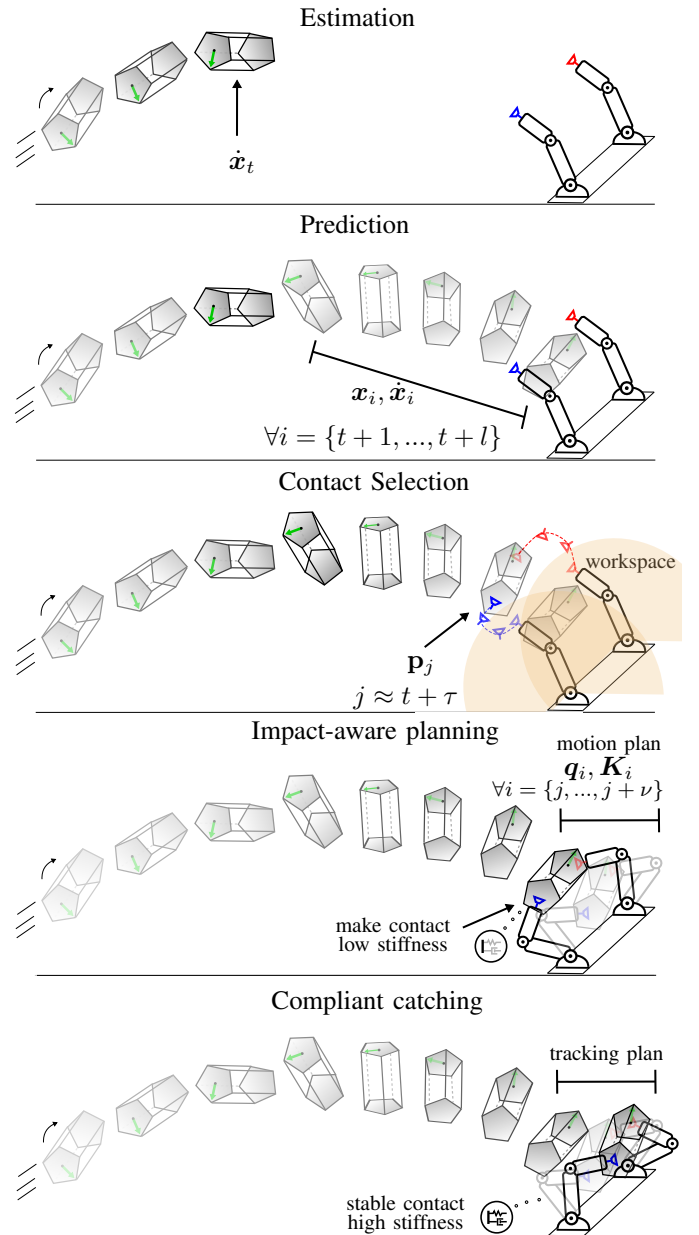


Fig. 2. Step-by-step pictorial description of catching a tumbling-flying object.

are predicted by solving an Initial Value Problem (IVP) with a constrained trajectory optimization (TO), (iii) *impacts models*, a key ingredient of our framework, that are used in the two following modules, (iv) the *contact selection* module where the optimal impact-wise contact locations on the object are found with our CD-SQP method, (v) the *planning* module where our Multi-Mode Trajectory Optimization (MMTO) is used to generate the optimal dual-arm motions, stiffness and force profiles on-the-fly, (vi) the *indirect force-control* module where the robots' set-points are adapted to track the desired force, and (vii) the *IK* module (inverse kinematics) that computes the corresponding robots' configurations. Fig. 3 illustrates the flow of information between these modules in our system.

Our goal is to develop an optimisation-based approach that minimizes the impulsive force for bi-manual catching of large momentum objects. We denote with $x_t \in \mathbb{R}^o$ the pose of an

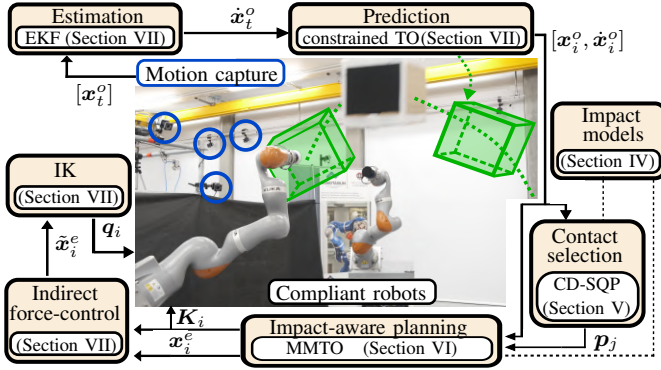


Fig. 3. Overview of the proposed system. The experimental setup is in the center. The flow of information starts from the motion capture and results into the input of the robots. Solid arrows denote information transmission between modules and dotted lines denote functional dependency between components.

object and with \dot{x}_t its velocity at time t . Assuming that the future trajectory of the object intersects the workspace of the robots, we denote with $q_t \in \mathbb{R}^m$ the trajectory of the robot arms in the configuration space, with m Degrees of Freedom (DoF), at times $t = \{t+1, \dots, t+\tau+\nu\}$, where $t+\tau$ is the time it takes for the object to intersect the robot's workspace and ν is the planning horizon. Further, $K_t \in \mathbb{R}^o$ denotes the stiffness of the robot end-effectors, with a task space of dimensionality o . Given the known mass M , inertia I and shape S of the object, the catching problem at time t can be described by:

$$q_{t|t}, K_{t|t} = 0(x_t, \dot{x}_t; M, I, S). \quad (1)$$

The notation $t|t$ in the subscript indicates that the values of q and K are planned for the times t given the current information available at time t .

Formally, an optimisation problem for catching large momentum objects (see (1)) can be written as:

$$\min_{q_t, K_t, x_t, \dot{x}_t} J(q_t, K_t, x_t, \dot{x}_t) \quad (2a)$$

$$\text{s.t. } \dot{x}_i = e(x_{0,\dots,t}; I), \quad \forall i = [0, \dots, t], \quad (2b)$$

$$[x_i, \dot{x}_i] = p(x_t, \dot{x}_t; M, I), \quad \forall i = [t+1, \dots, t+l], \quad (2c)$$

with $j = t + \tau$, where $l > \tau$,

$$p_j = c(x_j, \dot{x}_j; S), \quad (2d)$$

$$q_i, K_i = m(x_j, \dot{x}_j, p_j; M, I), \forall i = [j, \dots, j + \nu], \quad (2e)$$

where in (2a) $J(\cdot)$ is the cost function, in (2b) $e(\cdot)$ denotes the *estimation* process (see Fig. 2(a)) and in (2c) $p(\cdot)$ denotes the *prediction* process, with l being the prediction horizon (see Fig. 2(b)). Assuming that the object's predicted trajectory intersects the robot's workspace, the index $t+\tau$ indicates that moment in time the object enters the workspace. Based on the predicted state of the object at $t+\tau$, in (2d) $c(\cdot)$ the *contact selection* process (see Fig. 2(c)) and in (2e) $m(\cdot)$ the *impact-aware motion planning* process (see Fig. 2(d)) are completed.

Problem Statement: *Design an optimisation-based system able to smoothly catch large-momentum objects given only position measurements of the object till time t . This requires estimating and predicting the object's state, selecting contacts on its surface, planning the arms' motion, force and stiffness to cope with impacts, and realizing the catching motion.*

C. Contributions

As shown in Fig. 3, we propose an optimization-based approach to minimize the impulsive force for bimanual catching of large-momentum objects. Our framework enables the bimanual robots to generate on-the-fly impact-aware contacts, motion, force and stiffness plans that are conditioned on the position, speed, and shape of the object, as well as the environment that constraints the motion of the object. Our scientific focus is on the impact models, contact selection and impact-aware planning modules, while for the estimation, prediction, IK, and indirect force-control modules we used existing approaches. Yet, to achieve the goals stated above, we implemented and integrated all these individual components. The main contributions of this paper, under the assumptions that we have a good priori knowledge of the mass and inertia of the objects, are the following:

- 1) The impulse along tangential and normal directions of the contact are analysed using a 3D compliant impact model, as a function of contact location, direction and duration. Based on that, the contact selection principles are proposed.
- 2) According to the contact selection principles, an online impact-aware contact selection method is developed based on Sequential Quadratic Programming (SQP). This allows online-searching of multiple contacts on arbitrary shaped objects, which minimize the impulse between the robots and the moving object.
- 3) Our contact force transmission model and corresponding parametric programming technique [21] are extended to encode both hybrid dynamics and hybrid control for multi-contact and 3D scenarios. Together with the proposed contact searching method, this enables simultaneous and online optimization of contact locations, motions, force and stiffness profiles in 3D for dual-arm robotic systems.
- 4) The proposed methods and system are experimentally validated on hardware using two KUKA LBR iiwa robots, for the highly dynamic manipulation tasks of capturing swinging, spinning, tumbling, and flying large-momentum objects.

We structured the paper as follows. Section II contains the related work. In Section III, we provide the preliminaries on impact modelling and a set of hypotheses for impact-aware contact selection. In Section IV, we describe the proposed impact-aware contact searching method and Section V presents the multi-mode trajectory optimization used for impact-aware planning. Section VI and Section VII present several simulation and hardware experimental results of bi-manual catching of large swinging, spinning, and flying object, along with the implementation details on estimation and prediction. Sections VIII and IX discuss and conclude our work, respectively.

II. RELATED WORK

A. Object Motion Estimation and Prediction

In order to catch a fast-moving object, such as swinging or flying object, we strive for methods that are able to estimate

both the linear and angular velocities of the object in a fraction of a second. Hence, here we review methods towards this goal.

1) *Model-based methods*: Towards capturing a small spherical object [14] where the orientation of the object can be ignored, both Kalman Filter (KF) without air drag and with air drag [22] have been used to estimate the state of the object. Given these estimates, an Initial Value Problem (IVP) [16] can be solved to predict the future trajectory of the object by numerically integrating its dynamics equation. *These works only consider free-flying objects that are not constrained by the environment and only the linear part of their motion.*

A hybrid estimator was proposed [7] for in-flight objects in the case of 2D motion, where an Extended Kalman Filter (EKF) was used to estimate the linear position and velocity, and a least-square fitting method to estimate a single axis angular motion. Yet, to estimate and predict an object's motion in 3D, its inertia characteristics are also required. Masutani *et al.* [23] first studied the problem of inertial ratio parameters identification in 3D, given the angular velocity. Similar works motivated by space applications proposed EKF-based methods [24] to estimate the state and the inertial ratio parameters of objects. Yet, identifying the inertial parameters of an unperturbed free-flying object is a lagging process, due to the observability of the problem. *These approaches require tens or hundreds of seconds to converge, hence they are unsuitable for catching fast-moving objects.*

2) *Model-free methods*: In addition to the model-based approaches, recent works [8], [9], [25] proposed model-free estimation methods using learned models. These approaches are particularly useful in cases where the dynamics model of the object are not available, *e.g.* half-filled water bottle [8]. *However, their prediction accuracy is highly dependent on the collected data [8], while predictions of unseen objects relies on datasets [25] that are only feasible for small and light objects. For heavy and large objects, the data collection process can be impractical and strenuous.*

In this work, first we use a EKF [26] to estimate the current linear and angular velocities of the object given its inertia parameters. Secondly, we adopt a nonlinear program [17] to predict the future trajectory of the object which can be generalized to a variety of environments that may constrain the motion of the object, *e.g.* an object sliding on a table or being suspended from the ceiling.

B. Impact Mechanics

When catching large momentum objects, contacts may stick, slide, or bounce—due to impact. Hence, to analyze impacts under various conditions we study works on the mechanics of impact.

A key concept for analysing impacts is the *angle of incidence*. This is defined as the angle between the pre-impact relative velocity vector and the normal vector of the contact plane, and is used to differentiate oblique impacts from direct impacts [27], *i.e.* whether there is a tangential impact component. Brach [28] studied the 2D collisions and showed that the coefficient of restitution and the ratio of tangential impulse to normal impulse are related to energy loss and the existence of sliding. Keller [29] and Stewart *et al.* [30]

analysed impact of two rigid bodies with friction and showed that frictional impulse can be calculated based on the sliding velocity. More recently, Jia [31] studied an energy-based 3D impact model with both tangential compliance and friction, which considers both sliding and sticking contact modes. A comprehensive survey of impact models can be found in [32].

In this paper, we aim to inform the contact selection with impact-specific optimization criteria, such as *angle of incidence*. Thus, we use the 3D impact model with tangential compliance [31] to investigate a variety of impact scenarios and identify impact-specific metrics for contact selection. Also, we use a simpler model [33] of impact that have been used to relate impact properties such as duration and restitution coefficient to the mass, damping and stiffness parameters of the mechanical system. This model may be less accurate, but they are particularly useful when used for online planning and control. An extensive study on hybrid models and their use on robotics can be found in [34].

C. Contact Selection in Manipulation

Manipulation can be split into two paradigms [35], [36]. The prehensile, *i.e.* grasping, where the goal is to constrain the object within a “hand” such that the manipulation problem can be treated as a motion planning problem [37]. The non-prehensile—with which dexterity is exhibited—where the relative pose between the object and the end-effector or “finger” can change at anytime.

In both paradigms contacts are crucial. In the former, contact forces on the object are considered as passive or active according to the grasp properties such as form-closure or force-closure [38], [39]. These grasp conditions are quasi-static and applied in problems where the object is graspable. *As a result this paradigm is only relevant if the object is moving slowly or if the object can get attached to the robot “hand” immediately [8], [9].*

For non-prehensile manipulation, objects' state is altered by applying unilateral contact forces at the selected contact locations. This allows manipulation of large and heavy objects with as few as a single contact at a time and in dynamic setups (subject to impact handling). The main charm of this paradigm [40], [41] is that it can be used to solve any type of manipulation problem [42]. Hence, it has become increasingly used in tasks such as planar pushing [6], [43], pivoting objects [44], large object manipulation under gravity [12], [45], reactive use of tools [46], and small object stopping on an inclined air-hockey table [47].

In non-prehensile manipulation, to manipulate an object to a desired position, contact locations need to be particularly selected. The selection process is achieved either via sampling [48] or optimization [49], [50], [51] of the contact locations. In the former, the contact locations augment the configuration space (object pose), which are sampled such that the object can be pushed (quasi-statically) to the goal location. In dynamic setups, the object state is high-dimensional as it includes position, velocity, and acceleration, *etc.* Hence, *finding optimal solution via sampling in such configuration spaces is time-consuming.* In the latter, contact searching is achieved by following contact-informative gradients. In terms

of feasibility, the point-of-attack [49] was introduced to find the wrench exchange between two rigid bodies which is consistent with both contact geometry and dynamics equation. A sequence of QPs were bridged via a projection mechanism [51] such that a sequence of local optima leads to global optima. These approaches require a smooth surface representation, *e.g.* convex sphere-swept shapes [42], splines [50] or learned signed-distance-fields for nonconvex objects [52].

Here, we represent objects with 3D meshes and we propose a novel impact-informed optimization method based on SQP. This allow us to find the (impact-wise) optimal contact locations for catching fast-moving objects of arbitrary shape (even non-convex) with minimal impulsive forces.

D. Hybrid Motion Planning

In robotics, there is a variety of Hybrid Trajectory Optimization methods [43], [42] that are able to synthesize multi-contact manipulation plans. Typically, planning-focused methods study problems where impacts are not likely to occur. However, there has been increasing interest [53], [54] in transferring multi-contact manipulation behaviours to the hardware while simultaneously increasing the operation speed of robots. *These setups generate increasingly strong impacts*, therefore a variety of techniques were adopted to cope with them, such as inertia matrix shaping [55], stiffness regulation [21], motion regulation and impact-invariant projection [20].

In our previous work [12], we studied catching large objects by separating the motion planning into three phases; (i) pre-contact, (ii) contact, (iii) post-contact. This provides sub-optimal solutions and requires re-designing these distinct but consecutive phases in every scenario, *e.g.* when changing the workspace, the object shape or velocity. Further, to eliminate impacts, we induced a common heuristic [56] that constrains the relative velocity at contact to zero.

In this paper, to cope with impacts in very dynamic tasks and to relax the above assumption, we extend our multi-mode trajectory optimization framework (MMTO) [21] from 1D to 3D, from one to two arms and to scenarios with gravity. This enables us to generate optimal solutions online that involve synchronous motion of the two arms with respect to the object motion in 3D as well as the internal coordination of the force and stiffness profiles of the bi-manual robot.

E. Hybrid Motion Control

Control-focused methods have also been proposed towards regulating the transitions between free motion and motion in contact, *i.e.* impacts. *Although, dedicated controllers for impacts are essential, given the inherent limitations of the hardware [57], the impacts that a stand-alone controller is capable of dealing with are limited [21].*

To cope with this, the reference spreading [58] concept was introduced. According to this concept more than one reference are provided in the neighborhood of impact, such that the appropriate jump can be triggered at the moment of impact. This has been used in hybrid controllers [59], and QP controllers [60]. On another line of work, to avoid explicitly dealing with the combinatorial aspects of hybrid motions, provably stable control policies were synthesized

into optimization algorithms based on complementary contact constraints [61].

A variety of compliance controllers have also been used to deal with the imprecise timing of the transition in the reference motions. Depending on the conditions, whether the robot is in free-motion or in contact, a hybrid force controller that switches between impedance and admittance control was proposed [62]. Similarly, in [63] the gains of an adaptive impedance controller were optimized for each conditions.

In this paper, we adopt an indirect force control mechanism based on an impedance controller [64], which allows us to track the desired force when in contact and concurrently modulate the stiffness of the robot to deal with the imprecision during and after the contact transition.

III. PRELIMINARIES

Impact events occur when two rigid bodies (*e.g.* an object and a robot end-effector, see Fig. 4(1)) collide at non-zero relative velocity. The impulse resulting from these events is analyzed here based on a 3D compliance impact model. The analysis led to a set of hypotheses that are used for the contact selection (see Section IV), while a simpler compliance impact model is used to determine the optimal contact force profile in trajectory optimization (see Section V-C).

A. Impacts

The relationship between the impulse and the velocity jump of the object before and after impact can be described as:

$$M\Delta\mathbf{v} = M(\mathbf{v}^+ - \mathbf{v}^-) = \int_0^{\Delta t} \mathbf{F} dt = \mathbf{\Lambda}, \quad (3)$$

$$\mathbf{I}\Delta\boldsymbol{\omega} = \mathbf{I}(\boldsymbol{\omega}^+ - \boldsymbol{\omega}^-) = \int_0^{\Delta t} \mathbf{r} \times \mathbf{F} dt = \mathbf{r} \times \mathbf{\Lambda}, \quad (4)$$

where $M, \mathbf{I} \in \mathbb{R}^{3 \times 3}$ are the mass and inertia matrix of the object, respectively; $\mathbf{v} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ are the linear and angular velocity of the object with respect to the robot; $\mathbf{r} \in \mathbb{R}^3$ is the position vector of the contact location with respect to the center of mass of the object. We assume that the contact between the robot and the object is modelled as point-contact. The superscript “-” and “+” respectively represent the states before and after impact. The impulse $\mathbf{\Lambda} \in \mathbb{R}^3$ can be calculated by the integral of contact force $\mathbf{F} \in \mathbb{R}^3$. Assuming small enough impact duration Δt , the contact force \mathbf{F} can be approximated to be constant. Hence, we can rewrite (3) and (4) as:

$$\begin{bmatrix} M\mathbf{E}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}^+ - \mathbf{v}^- \\ \boldsymbol{\omega}^+ - \boldsymbol{\omega}^- \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{E}_3 & \mathbf{O}_3 \\ \mathbf{r}^\times & \mathbf{O}_3 \end{bmatrix}}_{(i)} \underbrace{\begin{bmatrix} \mathbf{u}_F \\ \mathbf{O}_{3 \times 1} \end{bmatrix}}_{(ii)} \underbrace{\|\mathbf{F}\|}_{(iii)} \underbrace{\Delta t}_{(iv)}, \quad (5)$$

where \mathbf{E}_n and \mathbf{O}_n are respectively $n \times n$ identity and zero matrices; \mathbf{r}^\times is the skew symmetric matrix of \mathbf{r} ; \mathbf{u}_F is the unit vector of \mathbf{F} and $\|\mathbf{F}\|$ is 2-norm of \mathbf{F} . It can be seen from (5) that the impact is related to four elements; (i) the contact location, (ii) the contact force direction, (iii) the contact force magnitude, and (iv) the impact event duration.

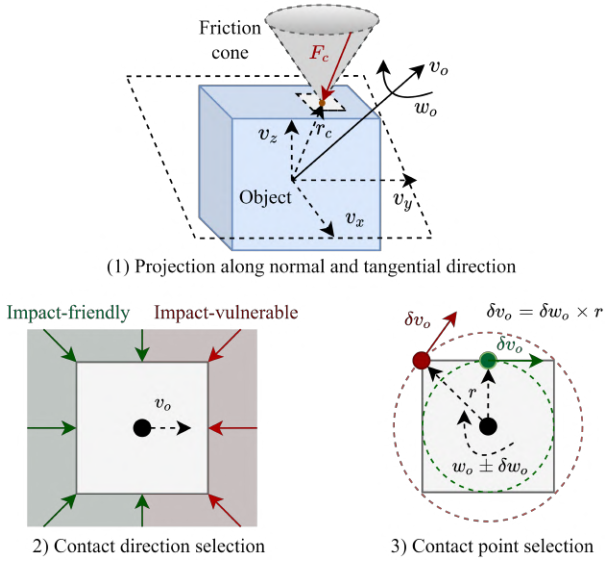


Fig. 4. Projection of contact velocity and force along normal and tangential directions (1) and 2D illustration for contact selection (2-3).

In this paper, we present a model-based optimization approach to obtain the optimal combination of these four quantities, *i.e.* our system determines on-the-fly the impact-resistant contact location r and contact direction u_F , as well as the magnitude $\|F\|$ and duration Δt of contact force F . We will present principles for determining the impact-resistant contact location r and direction u_F based on impulse analysis in this section, while the optimization of the magnitude $\|F\|$ and time duration Δt of contact force will be explained in Section V.

B. Impulse Analysis using Compliance Model of Impact

To analyse the impulse using (5), we need to know Δv and $\Delta \omega$ in advance (see in (3) and (4)), which implies that the velocities (v^+ and ω^+) after the impact need to be known. Consequently, (3), (4) and (5) are only suitable for calculating the impulse after impact, which is not useful for simulating forward the impulse, nor analysing the effects of its four elements (see (5)).

Our goal here is to investigate the effect of the contact location, direction and duration, highlighted in (5), on the resulting impulse. To model impact and analyse the corresponding impulses, mass-spring-damper systems have been adopted [31], [33]. Accordingly, we model the dynamic interaction between the object and the robot as well as the relationship between the Cartesian position and the contact force of the robot as:

$$M\Delta\ddot{x} + B\Delta\dot{x} + K\Delta x = F. \quad (6)$$

where M is set to the desired mass (as the effective Cartesian inertia of the robot is configuration-dependent and variant) while B, K are the desired damping and stiffness of the robot. x, \dot{x} and $\ddot{x} \in \mathbb{R}^3$ are respectively the position, velocity and acceleration of the robot end-effector; and F is the contact force. The Laplace transform of (6) can be written as

$$\Delta x = \frac{1}{Ms^2 + Bs + K} F = \frac{\frac{K}{M}}{s^2 + \frac{B}{M}s + \frac{K}{M}} \frac{F}{K}, \quad (7)$$

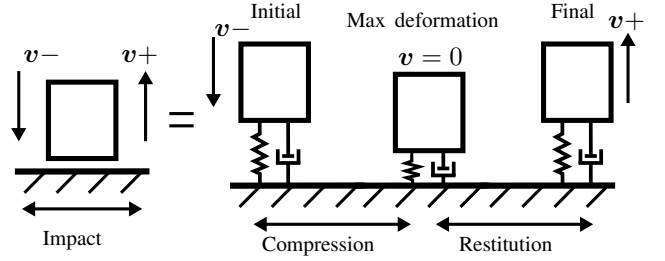


Fig. 5. Correspondence between Newton's restitution model for impacts and the mass-spring-damper system.

where the damping ratio is set to $\zeta = 1$, *i.e.* $B = 2\sqrt{MK}$, to guarantee the desired critically damped behavior of the dynamic contact interaction.

The energy dissipation of such a system is caused by the damper and can be calculated as follows:

$$E_d = \int_0^{\Delta t} B\Delta\dot{x}dx = \int_0^{\Delta t} B\Delta\dot{x}^2 dt. \quad (8)$$

Following [33], the energy loss during impact can be modelled with a mass-spring-damper system, where the dissipated energy during compression and restitution stages (see Fig. 5) is related to both stiffness and damping parameters.

Here, we adopt a 3D compliance impact model with three spatial springs [31] which has been used to simulate forward the compression and restitution impulses and compute the bodies' velocities after impact. Utilizing the energies stored in the springs, the impulse and the velocity changes during impact can be updated as follows:

$$\Lambda_{i+1} = \Lambda_i + \left(\frac{\partial \Lambda_x}{\partial \Lambda_z} \hat{x} + \frac{\partial \Lambda_y}{\partial \Lambda_z} \hat{y} + \hat{z} \right) \Delta \Lambda_z \quad (9)$$

$$\Delta v = \frac{\Lambda}{M}, \quad \Delta \omega = I^{-1} r \times \Lambda, \quad (10)$$

where Λ_x, Λ_y and Λ_z are respectively the three components of the impulse Λ ; $\Delta \Lambda_z$ is the incremental impulse along the normal direction at each iteration.

This model allows us to simulate the evolution of both the normal and tangential components of the impulse, while considering friction and the fact that the instantaneous contact between the two bodies can be sticking or sliding. The 3D impact model is briefly introduced in Appendix-A. Based on the forward simulation of impulse, we can decide the best contact direction and location by analysing the normal and tangential impulse components.

C. Contact Selection based on Impulse Analysis

In the 3D compliance impact model, the contact force is decomposed into the normal contact force F_z that is unlimited¹ and the tangential contact force $F_{tan} = F_x + F_y$ that is constrained by the friction cone. Further, the included angle between the object velocity and the contact force affects the distribution of impulse along normal and tangential directions. Based on these two observations we make the following

¹ Normal contact force F_z can be infinite unless the mechanical damage is reached.

hypotheses. These, along with the two observations above, are both validated with a variety of simulations in Section VI.

Hypothesis 1. *To minimize the magnitude of the impulse, the ideal contact direction (see Fig. 4) is defined to be perpendicular to the linear velocity of the object at the contact point, i.e. $\mathbf{u}_F \perp \mathbf{v}$. This implies that the linear velocity at the contact point lies in the tangential plane of the contact surface. Hence, the normal impulse is minimal and the tangential impulse will be limited by the friction cone of the contact surface.*

By selecting the contact surface according to *Hypothesis 1*, the unconstrained contact force lies in the orthogonal space of the object motion, and hence it will not be directly used for halting the object motion. On the other hand, the constrained contact force along tangential direction can be used to absorb all the momentum of the object. The algorithmic details on how to select optimal contact surfaces according to *Hypothesis 1* are given in Section IV.

Remark 1. *As shown in Fig. 4 (2), the contact surface that allows to apply normal forces in the same direction with object's velocity vector would also minimize the magnitude of the impulse. However, we are only interested in contacts that are able to reduce the velocity of the moving object after the impact event. Thus, such contact surfaces are deemed not useful and are not considered in this paper.*

Additionally, similar to the grasp quality measurement in [36], we introduce the following assumption for selecting optimal contact locations which minimize the effect of the motion uncertainty on the impact.

Hypothesis 2. *Given the determined optimal contact surface, the optimal contact location should be selected such that the effect of motion uncertainty on the impact is minimized, i.e. minimizing the contact velocity $\delta \mathbf{v} = \delta \boldsymbol{\omega} \times \mathbf{r}$ resulting from the angular motion uncertainty $\delta \boldsymbol{\omega}$. Without loss of generality, in the case of isotropic uncertainty ($\delta \omega_x = \delta \omega_y = \delta \omega_z$), this condition is equal to minimizing $\|\mathbf{r}\|$ (see (5)).*

The algorithmic details on how to select optimal contact locations according to *Hypothesis 2* are given in Section IV. The numerical simulation results of impulse with different contact locations and directions are given in Section VI.

Remark 2. *In this paper, the optimal contact locations are selected to be impact-resistant. However, for tasks that require impact, such as batting [7], we can also select contact locations that maximise impact.*

IV. IMPACT-AWARE CONTACT SELECTION

Based on the predicted object trajectory and the contact selection principles described in Section III, here we describe in detail the computational procedure to select optimal contacts (including the contact direction and location) for catching the moving object, i.e. *contact selection* process (see Section I-B).

A. Problem Formulation of Contact Selection

Assuming that the shape of the object (3D model) is known in advance, we can generate the corresponding 3D mesh of

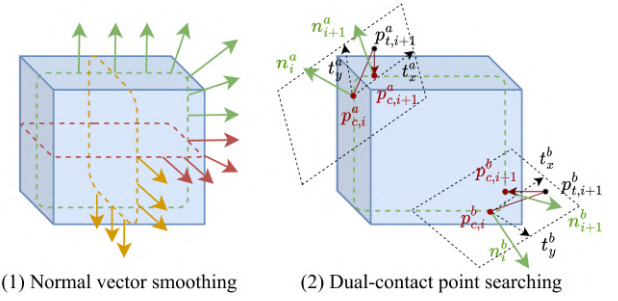


Fig. 6. Normal vector smoothing and dual-contact point searching.

the object using *PyMesh* [65] or *Open3D* [66]. The mesh file includes all the vertices and faces of the object that form the contact surfaces and allows us to obtain the normal vector for each candidate contact location. Given the object mesh, we can search for the suitable contact points $\{\mathbf{p}^k\}$ along the object surfaces by solving the following nonlinear programming problem (NLP):

$$\begin{aligned} \min_{\mathbf{p}^k} e(\mathbf{p}^k, \mathbf{n}^k, \mathbf{v}^o) & , \quad \forall k \in \{0, \dots, \mathcal{K}\}, \\ \text{s.t. } \mathbf{p}_k \in \mathcal{S} & \end{aligned} \quad (11)$$

where \mathcal{K} is the number of robots; $\mathbf{p}^k \in \mathbb{R}^3$; \mathcal{S} represents the face set of the object mesh (surface); $e(\cdot)$ is the cost function that depends on the contact location \mathbf{p}^k of the end-effector of robot k , and the normal vector $\mathbf{n}^k \in \mathbb{R}^3$ at the closest contact location corresponding to \mathbf{p}^k .

Cost function: Our objective is to find the \mathcal{K} contact locations on the surface of the object that minimize impulse and simultaneously provide contact forces such that the motion of the object can be halted. Hence, the proposed cost function is $e(\cdot) = \mathbf{e}^\top \mathbf{e}$ with

$$\mathbf{e}^k = \begin{bmatrix} w_1(\mathbf{n}^k \cdot \mathbf{v}^o) \\ w_2(\mathbf{p}^k - \mathbf{p}^o) \\ w_3 \sum_j (\mathbf{n}^k \cdot \mathbf{n}^j - \cos(\frac{2\pi}{\mathcal{K}})) \end{bmatrix}^\top \quad (12)$$

for $\forall k \in \{0, \dots, \mathcal{K}\}$ and $j \in \{0, \dots, \mathcal{K} \mid j \text{ is adjacent to } k\}$. $\mathbf{p}^o, \mathbf{v}^o$ are the position and velocity vectors of the center of mass (CoM) of the object. The first and second terms in (12) are respectively designed according to *Hypothesis 1* and *Hypothesis 2* in Section III, while the third term is designed for improving the catching quality which is typically used in grasping [36].

B. Gradient-based Contact Searching

The contact point search problem (11) (with cost given in (12)) is a nonlinear constrained optimization problem, which can be solved efficiently via Sequential Quadratic Programming (SQP) following Coordinate Descent (CD) [67]. The outline of the method is provided in Algorithm 1. Given the pose \mathbf{x}^o and velocity \mathbf{v}^o of the object and the end-effector pose $\{\mathbf{x}^{e,k}\}$ of \mathcal{K} robots, the gradient-based contact searching algorithm can find the optimal contact locations $\{\mathbf{p}^k\}$.

Coordinate Descent SQP: Optimizing multiple interacting contact points concurrently makes the problem non-convex due to cross-coupling factors, such as the coordination (third) term in (12). To accommodate for that, we optimize convex

Algorithm 1 CD-SQP for Multi-Contact Selection

```

1: Initialize:  $\mathbf{x}^o, \mathbf{v}^o, \{\mathbf{x}^{e,k}\}; i = 1$ 
2: for ( $k = 0; k < \mathcal{K}; ++k$ ) do
3:    $\mathbf{p}_0^k \leftarrow \mathbf{x}^{e,k}; \mathbf{p}_1^k \leftarrow \text{Proj}(\mathbf{x}^{e,k})$ 
4:   while  $\bigwedge_{k=\{0,\dots,\mathcal{K}\}} \{\|\mathbf{p}_i^k - \mathbf{p}_{i-1}^k\| \geq \tau\}$  do
5:      $i = i + 1$ 
6:     for ( $k = 0; k < \mathcal{K}; ++k$ ) do
7:        $j = \{0, \dots, \mathcal{K} \mid j \neq k\}$ 
8:        $\hat{\mathbf{p}}_i^k \leftarrow \text{SCS-QP}(\mathbf{p}_{i-1}^k, \mathbf{n}_{i-1}^k, \mathbf{x}^o, \mathbf{p}_{i-1}^j, \mathbf{n}_{i-1}^j)$ 
9:        $\mathbf{p}_i^k \leftarrow \text{Proj}(\hat{\mathbf{p}}_i^k)$ 
10:      if ( $i > \text{max\_iter}$ ) then
11:        break
12:   return  $\{\mathbf{p}_i^k\}, \forall k \in \{0, \dots, \mathcal{K}\}$ 

```

subproblems in a loop (see lines 5-8 in *Algorithm 1*). Within each iteration we optimize the location of a single contact point \mathbf{p}^k and we fix all other contact points \mathbf{p}^j – referred to as *Single Contact Searching QP*. The scheduling can be seen as a cyclic CD where every time we change the contact point \mathbf{p}^k that we optimize for, we alter the direction of descent.

Single Contact Searching QP: Each subproblem is responsible for optimizing the location of a contact point, while keeping it on the surface ($\mathbf{p}^k \in \mathcal{S}$). This can be expressed via non-linear equality constraints, e.g. $\mathcal{S}(\mathbf{p}^k) = 0$. The challenge here is that for closed surfaces the constraint $\mathcal{S}(\cdot)$ is non-linear and should to be differentiable (for a gradient-based solver). Although, there are potential approaches using non-linear solvers [50], [52], we adopt ideas from [51] to perform a series of linear steps along consecutive tangential planes of the surface based on smoothed normal vectors, i.e. we iteratively linearize $\mathcal{S}(\cdot)$. This allows our method to work with arbitrary number of contact points and meshes (surfaces). As shown in *Algorithm 1*, each linear step is realized by solving a *Single Contact Searching QP* subproblem (line 8) and it's followed by a projection (line 9) back to the surface of the object. The series of linear steps are performed until convergence for all the contact points (see lines 4-11).

Contact Location Iteration: Given a contact location $\mathbf{p}_{i-1}^k \in \mathcal{S}$ at iteration $i-1$, the i th QP subproblem aims to find the optimal displacement $\Delta\mathbf{p}^k$ of the contact point \mathbf{p}^k within a tangential plane (see Fig. 6). The i th tangential plane is defined based on the contact location \mathbf{p}_{i-1}^k and the corresponding normal vector \mathbf{n}_{i-1}^k . The i th contact point within the tangential plane is $\hat{\mathbf{p}}_i^k = \mathbf{p}_{i-1}^k + \Delta\mathbf{p}_{i-1}^k$ and the optimal contact point \mathbf{p}_i^k on the object's surface is obtained by projecting $\hat{\mathbf{p}}_i^k$ on the surface of the object.

The actual QP subproblem formulation is:

$$\begin{aligned} \min_{\Delta\mathbf{p}^k} \quad & \frac{1}{2}(\mathbf{J}^k \Delta\mathbf{p}^k)^\top (\mathbf{J}^k \Delta\mathbf{p}^k) + \mathbf{e}^k \mathbf{J}^k \Delta\mathbf{p}^k \\ \text{s.t.} \quad & \Delta\mathbf{p}_{\min} \leq \Delta\mathbf{p}^k \leq \Delta\mathbf{p}_{\max}, \end{aligned} \quad (13)$$

where the Jacobian of contact location optimization (13) is:

$$\mathbf{J}^k = \frac{\partial \mathbf{e}^k}{\partial \Delta\mathbf{p}^k} = \begin{pmatrix} \omega_1 \mathbf{v}^{o,\top} \frac{\partial \mathbf{n}^k}{\partial \Delta\mathbf{p}^k} \\ \omega_2 \frac{\partial \mathbf{p}^k}{\partial \Delta\mathbf{p}^k} \\ \omega_3 \mathbf{n}^j \frac{\partial \mathbf{n}^k}{\partial \Delta\mathbf{p}^k} \end{pmatrix}, \quad (14)$$

Algorithm 2 QP for Single Contact Searching (*SCS – QP*)

```

1: Initialize:  $\mathbf{p}_{i-1}^k, \mathbf{n}_{i-1}^k, \mathbf{x}^o, \mathbf{p}_{i-1}^k, \mathbf{n}_{i-1}^k$ 
2:  $\mathbf{J}_{\text{pos}}, \mathbf{J}_{\text{ori}}, \mathbf{J}_{\text{coo}} \leftarrow \partial \mathbf{e}_i^k / \partial \Delta\mathbf{p}^k$ 
3: Jacobian matrix:  $\mathbf{J}^k \leftarrow [\mathbf{J}_{\text{pos}}, \mathbf{J}_{\text{ori}}, \mathbf{J}_{\text{coo}}]^\top$ 
4:  $\mathbf{Q} \leftarrow \mathbf{J}^{k,\top} \mathbf{J}^k; \mathbf{q} \leftarrow \mathbf{J}^{k,\top} \mathbf{e}_i^k; \mathbf{lb}, \mathbf{ub} \leftarrow \Delta\mathbf{p}_{\min}, \Delta\mathbf{p}_{\max}$ 
5:  $\Delta\mathbf{p}^k \leftarrow \text{QP\_solver}(\mathbf{Q}, \mathbf{q}, \mathbf{lb}, \mathbf{ub})$ 
6:  $\hat{\mathbf{p}}_i^k \leftarrow \mathbf{p}_{i-1}^k + \Delta\mathbf{p}^k$ 
7: return  $\hat{\mathbf{p}}_i^k$ 

```

$$\frac{\partial \mathbf{p}^k}{\partial \Delta\mathbf{p}^k} = (\mathbf{t}_x^k, \mathbf{t}_y^k), \quad (15)$$

$$\frac{\partial \mathbf{n}^k}{\partial \Delta\mathbf{p}^k} = \left(\frac{\mathbf{n}^k|_{+x} - \mathbf{n}^k|_{-x}}{2\delta}, \frac{\mathbf{n}^k|_{+y} - \mathbf{n}^k|_{-y}}{2\delta} \right), \quad (16)$$

where $\mathbf{n}^k|_{\pm x}$ ($\mathbf{n}^k|_{\pm y}$) is the normal vector of the contact surface at contact location \mathbf{p}^k when Δp_x (Δp_y) is $\pm\delta$.

The outline of the single contact searching method is provided in *Algorithm 2*. Also, note that as shown in Fig. 6 (1), the normal vectors on the surfaces of the object are smoothed to ensure that the contact location \mathbf{p}^k can be optimized across all the noncontinuous object surfaces.

By solving *contact selection problem* using the CD-SQP, the proposed method is able to compute the optimal contact locations \mathbf{p}^k and normals \mathbf{n}^k on the moving object, which are used in the *impact-aware planning* process (see Section V).

V. MULTI-MODE TRAJECTORY OPTIMIZATION

In this section, we describe the proposed motion planning method which is used to optimize the capturing motion, i.e. the *impact-aware planning* process. A dual-arm catching motion comprises of a variety of modes, such as free motion, motion in contact, free single-arm motion, constrained dual-arm motion, and compliant or stiff behaviour in interaction. In general, systems that transit through different modes are typically referred as hybrid. These are typically not causal with respect to the state and control, hence the notion of mode is introduced. Specifically, for transitions from *free-motion* to motion *in-contact* and vice-versa we denote the dynamics mode [68] with $\mathcal{M}^d \in \mathbb{Z}$. For transitions from one controller to the other, such as from stiff to soft behaviour and vice-versa we denote the control mode with $\mathcal{M}^c \in \mathbb{Z}$. For transitions from single-arm free motion to dual-arm coordinated motion and vice-versa we denote the planning mode with $\mathcal{M}^p \in \mathbb{Z}$.

Our goal is to optimize a single trajectory that includes transitions from one mode to the other. To this end, we define a single combination of contact mode, planning mode and corresponding controller as a mode tuple of the system [21] $\mathcal{M} = (\mathcal{M}^d, \mathcal{M}^c, \mathcal{M}^p)$. Given a sequence of these modes $\mathcal{M} : \{\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_N\}$ of the trajectory, we express the Multi-Mode Trajectory Optimization (MMTO) as follows:

$$\min_{\mathbf{x}, \mathbf{a}, \mathbf{u}} \quad \mathbf{c}(\mathbf{x}, \mathbf{a}, \mathbf{u}, \mathcal{M}) \quad (17a)$$

$$\text{s.t.} \quad \mathbf{x}_{n+1} = \mathbf{f}(\mathbf{x}_n, \mathbf{a}_n, \mathbf{u}_n, \mathcal{M}_n), \quad (17b)$$

$$\mathbf{a}_{n+1} = \mathbf{h}(\mathbf{a}_n, \mathcal{M}_n), \quad (17c)$$

$$\mathbf{g}(\mathbf{x}_n, \mathbf{a}_n, \mathbf{u}_n, \mathcal{M}_n) \leq 0, \quad (17d)$$

where $\forall n \in \{0, \dots, N\}$; \mathbf{x}_n is the state of the system, \mathbf{a}_n is the control attributes of the robot (e.g. robot stiffness) and \mathbf{u}_n is the control input applied on the environment (e.g. force applied on the object). (17a) - (17d) are piece-wise functions from which the corresponding pieces are selected based on \mathcal{M}_n . (17a) defines the objective function, (17b) and (17c) are difference functions (discretize differential functions) and $\mathbf{g}(\cdot)$ in (17d) describes both the equality and the inequality constraints. Note that the formulation above defines an Optimal Control problem where both dynamics (17b) and control (17c) are hybrid. Unless otherwise specified, all the variables are represented in the world coordinate system.

The discretized optimization problem in (17) involves $N-1$ collocation knots² using direct transcription [17]. The transcription of MMTO is based on the phase-based parameterization used in our previous work [21], [50] and is similar in spirit to [42]. For each n_{th} knot, the decision variables are (i) the pose of the object \mathbf{x}_n^o , (ii) the velocity of the object $\dot{\mathbf{x}}_n^o$, (iii) action timings $\Delta \mathbf{T}_n$, (iv) the end-effector's position \mathbf{x}_n^e and velocity $\dot{\mathbf{x}}_n^e$, (v) the contact force \mathbf{F}_n and a parameter α_n (described in detail in the following section). We group these quantities into three vectors

$$\mathbf{x}_n = [\mathbf{x}_n^o \quad \dot{\mathbf{x}}_n^o \quad \mathbf{x}_n^e \quad \dot{\mathbf{x}}_n^e]^\top, \quad (18)$$

$$\mathbf{a}_n = [\alpha_n \quad \Delta \mathbf{T}_n]^\top, \quad (19)$$

$$\mathbf{u}_n = [\mathbf{F}_n \quad \dot{\mathbf{F}}_n]^\top, \quad (20)$$

that all together describe a multi-mode motion, which is tailored via the mode sequence \mathcal{M} . This results in a TO problem with different constraints for each one of the modes. Next, we describe the constraints that form (17b) - (17d) according to the different modes.

A. Hybrid Motion Equations for the Object and Robots

Both the object's and robots' motions depend on the current and the subsequent dynamics modes. Next, we present the models used to describe the behaviour of the object and the robots with respect to the dynamics modes, i.e. free-motion mode ($\mathcal{M}^d = 0$) and contact mode ($\mathcal{M}^d = 1$).

1) *Object's dynamics*: The object dynamics in free motion are denoted with $\mathcal{M}^d = 0$, while when $\mathcal{M}^d = 1$, the object is in contact. According to this parametrization the dynamics of the object are described by

$$[\mathbf{x}_{n+1}^o, \dot{\mathbf{x}}_{n+1}^o] = \begin{cases} \bar{f}^o(\mathbf{x}_n^o, \Delta \mathbf{T}_n), & \text{if } \mathcal{M}^d = 0, \\ f^o(\mathbf{x}_n^o, \Delta \mathbf{T}_n, \mathbf{F}_n), & \text{if } \mathcal{M}^d = 1, \end{cases} \quad (21)$$

where $\bar{f}^o(\cdot) : \mathbb{R}^{o+1} \rightarrow \mathbb{R}^{2 \cdot o}$, $f^o(\cdot) : \mathbb{R}^{2 \cdot o+1} \rightarrow \mathbb{R}^{2 \cdot o}$ with $f^o(\cdot)$ ((21) for $\mathcal{M}^d = 1$) in detail being

$$\begin{bmatrix} m\mathbf{E}^3 & 0 \\ 0 & \mathbf{I} \end{bmatrix} \ddot{\mathbf{x}}_n^o + \begin{bmatrix} m\mathbf{g} \\ \boldsymbol{\omega}_n \times (\mathbf{I}\boldsymbol{\omega}_n) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_n \\ \mathbf{r}_n \times \mathbf{F}_n \end{bmatrix}. \quad (22)$$

(22) makes the hybrid nature of the system's dynamics evident. With $\mathcal{M}^d = 1$ the right hand side (RHS) of (22) remains while with $\mathcal{M}^d = 0$, RHS of (22) disappears. The orientation of the object is represented with ZYX Euler angles, hence, the angular velocity is first converted to the derivative of Euler angles and then it is integrated to the ZYX Euler angles.

² Knots are the discretization points of the transcribed continuous problem.

2) *Robot's motion*: A key characteristic of motions with velocity jumps—such as impacts—is that the integration from accelerations to velocities needs to be skipped at specific mode transitions [68], [69]. Hence, with the following:

$$\begin{bmatrix} \mathbf{x}_{n+1}^e \\ \dot{\mathbf{x}}_{n+1}^e \end{bmatrix} = \begin{cases} f^e(\mathbf{x}_n^e, \dot{\mathbf{x}}_n^e, \ddot{\mathbf{x}}_n^e, \Delta \mathbf{T}_n), & \text{if } \mathcal{M}_n^d = \mathcal{M}_{n+1}^d = 0 \text{ or } 1 \\ \bar{f}^e(\mathbf{x}_n^e, \dot{\mathbf{x}}_n^e, \Delta \mathbf{T}_n), & \text{if } \mathcal{M}_n^d = 0, \mathcal{M}_{n+1}^d = 1, \\ \underline{f}^e(\mathbf{x}_n^e, \dot{\mathbf{x}}_n^e, \Delta \mathbf{T}_n), & \text{if } \mathcal{M}_n^d = 1, \mathcal{M}_{n+1}^d = 0 \end{cases} \quad (23)$$

we describe the motion of the robots' end-effectors, where $f^e(\cdot) : \mathbb{R}^{3 \cdot o+1} \rightarrow \mathbb{R}^{2 \cdot o}$, $\bar{f}^e(\cdot) : \mathbb{R}^{2 \cdot o+1} \rightarrow \mathbb{R}^{2 \cdot o}$ and both are integration functions. The transitions in (23) occur during the making and the breaking of contact, and they are susceptible to impact. We denote these as $\mathcal{M}_n^d = 0 \rightarrow \mathcal{M}_{n+1}^d = 1$ and $\mathcal{M}_n^d = 1 \rightarrow \mathcal{M}_{n+1}^d = 0$, respectively. By omitting $\ddot{\mathbf{x}}_n^e$ in $f^e(\cdot)$, accelerations are free, the velocity jumps are possible. Hence, the solution space of (17) includes discrete events, typically described as jumps in hybrid systems literature [68].

B. Control Modes during Contact Transitions

Similar to the dynamics modes (\mathcal{M}^d), here, we describe control modes (\mathcal{M}^c), which are relevant to mitigate impact in highly dynamic manipulation that involve contact transitions.

Based on the relation between an impact event and the behaviour of a spring-damper system (see Section III-B and Fig. 5), the characteristics of the physical system during impact, such as the duration of impact and the restitution coefficient³, can be associated to the mass, damping and stiffness parameters of a mechanical system [33]. As it can be seen in Fig. 5, an impact event can be divided into two stages: (i) the compression stage where the relative velocity is negative and the contact force establishes a stable contact, and (ii) the restitution stage where the relative velocity is positive and the contact force yields the desired manipulation task. We encode these two stages with two different control modes

$$\mathcal{M}^c = \begin{cases} -1, & v^- \rightarrow 0 \\ 1, & 0 \rightarrow v^+ \end{cases}. \quad (24)$$

For dynamic motions with contact, during the compression stage (soft mode) the stiffness should be minimized to alleviate impulsive forces, while during the restitution stage (stiff mode) the stiffness should be optimized to achieve the desired motion.

C. Contact Force Transmission during Impact

Based on the control modes introduced above, here, we devise a force transmission model to conform with the different stages during an impact event. The proposed force transition model is used in (17) to model the force evolution during an impact event and is a simplified version of the energy-based 3D impact model [7] (see Section III-B).

We adopt the simplified impact model mainly because of the following reasons. First, our model is practical and can be embedded in a TO formulation like (17), with which its parameters can be optimized (versus only forward integration). Second, our model considers the contact force only along the normal direction of the contact surface (versus all three), as it

³ The restitution coefficient value $\epsilon_r = 1$ represents a perfectly elastic collision, $0 \leq \epsilon_r \leq 1$ represents a real-world inelastic collision. $\epsilon_r = 0$ represents a perfectly inelastic collision.

is the only one that can be actively controlled. Third, it allows us to describe the force evolution as a function of stiffness and time (versus the impulse increment $\Delta\Lambda_z$).

Yet, to devise a force model for impacts that is time dependent and can be used to plan the evolution of the contact force during impact, we make the following assumption.

Assumption 1. *The duration of an impact event can be prolonged by reducing the contact force magnitude during the compression stage.*

1) *Contact force transmission model:* The ideal contact force profile should be smooth without oscillations. From (7), we can formulate the normal force transmission model as a second-order critically damped dynamical system (cd-DS) [3] to enforce the smoothness of contact force. By setting $\alpha = \sqrt{K/M}$ and assuming the magnitude of the force is dominated by the stiffness and displacement components considering $\Delta\dot{x} \approx 0$ and $\Delta\ddot{x} \approx 0$, i.e. $F(t) \approx K\Delta x(t)$, (7) becomes:

$$F(t) = \frac{\alpha^2}{s^2 + 2\alpha s + \alpha^2} F, \quad (25)$$

and the second-order cd-DS in time-domain:

$$\ddot{F}(t) + 2\alpha\dot{F}(t) + \alpha^2 F(t) = \alpha^2 F, \quad (26)$$

where $F(t)$ is continuous, satisfying $F(t) \in [0, F]$ and it is the magnitude of the normal component (with respect to contact surface) of \mathbf{F}_n in (22), while \dot{F} and \ddot{F} are its first and second derivatives. F represents the optimised force at the end of each control mode and it is not an explicitly variable in the optimisation (17). For any $\alpha > 0$, the contact force transmission model is critically damped, therefore it will converge to its equilibrium point asymptotically in the shortest possible time without oscillation.

Given any two out of three from; the stiffness parameter K , the mass parameter M , and the cd-DS parameter α , we can determine the other one through $\alpha = \sqrt{K/M}$ and the damping parameter via $B = 2\sqrt{MK}$. This leads to:

Remark 3. *Stiffness parameter K is related to parameter α .*

In addition to the smoothness enforced by the cd-DS (26) on the contact force transmission model, from the differential equation (26) we can derive a relationship between the settling time t_s and α (within 5%, $t_s \approx \frac{3.0}{\omega\zeta} = \frac{3.0}{\alpha}$). This is the same as settling time of the spring-damper system (6), which reveals:

Remark 4. *Impact duration is related to parameter α .*

2) *Hybrid contact force constraint:* The forces of the MMTO (see (17)) are generated based on the contact force transmission model described above. Yet, as it has been discussed in Section V-B there can be two different control modes (corresponding to two stages of contact, see (24)) during an impact event. Hence, to encode the two different control modes we consider two different contact force transmission models with different α parameters, denoted as $\alpha_{\mathcal{M}^c}$, which are respectively (i) for the compression stage with $\mathcal{M}^c = -1$ and (ii) for the restitution stage with $\mathcal{M}_c = 1$. This leads to:

$$\begin{bmatrix} \dot{F} \\ \ddot{F} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\alpha_{\mathcal{M}^c}^2 & -2\alpha_{\mathcal{M}^c} \end{bmatrix} \begin{bmatrix} F \\ \dot{F} \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha_{\mathcal{M}^c}^2 \end{bmatrix} F_{\mathcal{M}^c}, \quad (27)$$

that describes the contact force profile through two different control modes during an impact event. Note that each control mode (i.e. compression and restitution) has different $F_{\mathcal{M}^c}$. For each control mode, $\alpha_{\mathcal{M}^c}$ is optimized to modulate the contact duration ΔT (see Remark 4) and contact force profile along the surface normal. These are optimised within the MMTO according to the feasible contact duration, which is related to the velocity of object, and the workspace of the robot arms.

3) *Friction cone constraint:* To model the force in the two tangent dimensions of the surface, we use a friction cone constraint. This is activated when the robot switches from free motion ($\mathcal{M}^d = 0$) to contact ($\mathcal{M}^d = 1$) mode, and it is enforced in the MMTO with:

$$\angle(\mathbf{F}_n, \mathbf{n}) \leq \arctan(\mu), \quad (28)$$

where $\angle(\cdot)$ is the operator used to denote the included angle between two vectors and μ is the friction coefficient.

D. Stiffness Modulation during Contact

The main benefit of the proposed force transmission model is that based on the optimized parameter $\alpha_{\mathcal{M}^c}$, the corresponding damping and stiffness parameters of the impedance controller on a robot can be obtained. We can achieve this as the same mass-spring-damper system model (see Section III) which establishes the relationship between force and position is also adopted in the impedance controller to regulate the operational force of a robot manipulator. In this way, in MMTO, the stiffness characteristics of the robots are also optimized in a coherent way through $\alpha_{\mathcal{M}^c}$ —considering both stages of impact (see Section V-B)—without separating the contact scheduling from stiffness modulation into two separate optimisation levels as in [70].

E. Manipulation Planning Modes for Catching

In addition to the dynamics (\mathcal{M}^d) and the control (\mathcal{M}^c) modes, we use manipulation planning modes (\mathcal{M}^p), commonly used in multi-modal planning to enforce geometric constraints, such as the object being coupled to the robots' end-effectors (without modelling any force interaction), dual-arm motion coordination in free-motion, *etc.* Here, we have two different modes; (i) the decoupled mode, where the robots' end-effectors are allowed to move independently, denoted as $\mathcal{M}^p = 0$, and (ii) the coupled mode, where the robots' end-effectors maintain a fixed relative pose to the object and between them, denoted as $\mathcal{M}^p = 1$. According to these two modes we define the following constraint:

$$d(\mathbf{x}^e, \mathbf{p}) \begin{cases} > 0 & \text{for } \mathcal{M}^p = 0 \\ = 0 & \text{for } \mathcal{M}^p = 1 \end{cases}, \quad (29)$$

where $d(\cdot)$ is the signed distance operator (w.r.t the surface) between two elements, one of which can be the desired contact point \mathbf{p} and the other is the end-effector's position \mathbf{x}^e .

Note that for the catching scenario specifically, we assume that for the short time duration of the motion, each end-effector of the robot remains on the same side of contact surface, and both end-effectors make contact with the object at the same time. The mode (\mathcal{M}^p) transition from decoupled to coupled happens when the mode (\mathcal{M}^d) transitions from free-motion to

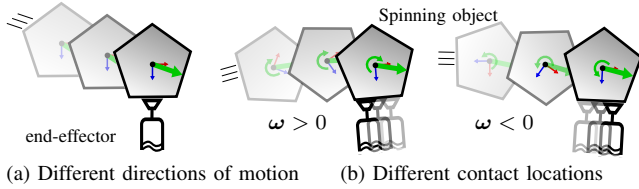


Fig. 7. Making contact with different contact directions and locations. (a) Same velocity magnitude and different inclination angles. (b) Different locations along the normal direction of the motion with positive and negative angular velocity, respectively.

contact, yet for other tasks this might not be the case. Most importantly, once the coupled mode is active the motion of the (two) robot arms needs to be coordinated, as they are constrained on the object. Hence, the MMTO needs to generate a motion that transits from independent motion (decoupled) to coordinated motion (coupled) of the robot arms.

F. Mode-Independent Constraints

Here, we introduce all the mode-independent constraints in the proposed MMTO (see (17)), i.e. the constraints that are not related to the mode sequence \mathcal{M} .

1) *Initial and final conditions*: The initial states of the object and the robot are respectively denoted as $\mathbf{x}_0^o = \mathbf{x}_*^o$, $\dot{\mathbf{x}}_0^o = \dot{\mathbf{x}}_*^o$ and $\mathbf{x}_0^e = \mathbf{x}_*^e$, $\dot{\mathbf{x}}_0^e = \dot{\mathbf{x}}_*^e$, while the final state of the object is defined according to the task requirement. For the catching task without a target position we set the final state of the object as: $\dot{\mathbf{x}}_N^o = \mathbf{0}$ and for catching with a target position we set the final state as $\dot{\mathbf{x}}_N^o = \mathbf{0}$, $\mathbf{x}_N^o = \mathbf{x}_d^o$.

2) *Workspace limit of the bimanual robot*: For each robot manipulator, we use a sphere located at the center of the second joint, denoted as \mathbf{c}_{j2} , to approximately define the workspace: $\|\mathbf{c} - \mathbf{c}_{j2}\| \leq R_w$, with R_w being the radius of the workspace sphere.

3) *Time bound between adjacent knots*: Further, we also set upper and lower bound on the time duration between any two adjacent knots as: $\Delta T_l \leq \Delta T_n \leq \Delta T_u$.

VI. SIMULATION RESULTS

In this section, we analyse and validate the hypotheses (see Section III), models (see Section V-C) and algorithms (see Section IV and Section V) described in the previous sections, using various numerical simulation results. First, we analyse the impulse distribution along normal and tangential directions, for the case of a moving object making contact with the robot at different contact locations and directions. Second, we present the results of the impact-aware contact selection algorithm, on several discontinuous and non-convex shaped objects. Third, we perform an ablation study by numerically comparing our impact-aware method with its impact-agnostic version, on a bimanual catching task. We used a 64-bit Intel Quad-Core i9 3.60GHz computer with 64GB RAM for running all the numerical computations and simulations.

A. Impulse Distribution along Normal-Tangential Directions

Our aim here is to determine the optimal contact direction and location for the robots to make contact with a moving object. We use the 3D compliance impact model (see Section III-B) to analyse the impulse distribution along

normal and tangential directions in different scenarios. Fig. 7 illustrates various scenarios where a static end-effector makes contact with a surface of a moving object. The scenarios considered involve different linear and angular velocities of the object, and different contact locations and directions.

The first scenario, exemplified in Fig. 7(a), considers an object making contact with the robot at a speed of 2 m/s and along different directions. We consider the contact inclinations of 180° , 175° , 160° , 150° , 140° , 130° , 120° , 100° , 95° , and 91° , where 180° corresponds to a contact velocity aligned with the normal vector of the contact surface and 90° corresponds to a contact velocity aligned with the tangential vector. Note that an inclined angle of 90° result in a normal contact velocity of zero, hence the contact will not happen.

The first and second plots of Fig. 8 show the impulse distribution along the normal and tangential directions. As the inclination angle decreases, from a normal direction (180°) to almost tangential (91°), the total impulse magnitude decreases from 3.969 Ns to 0.028 Ns (the normal impulse decreases from 3.969 Ns to 0.025 Ns while the tangential impulse increases from 0.0 Ns to 0.922 Ns, then decreasing to 0.012 Ns). Note that when the contact direction changes from the normal to tangential, there will be sliding contact during the impact evolution between the object and the robot. As a result, the friction cone will limit the impulse magnitude, which is consistent with *Hypothesis 1* in Section III-C.

In the second scenario, we investigate the optimal contact location on a spinning object for mitigating the impulse resulting from the motion uncertainty. In this scenario, the contact between the object and the end-effector occurs when the object travels with linear velocity of 2 m/s, and angular velocity of ± 0.2 rad/s (see Fig. 7(b)). Further, we consider seven contact locations (one at the center, three left and three right) and at the moment of contact we ensure that the surface normal (of the end-effector) is perpendicular to the linear velocity of the contact point. Fig. 8(3) shows the corresponding impulse distribution for different contact locations. For different relative angular velocities, the impulses at the same location are different. The contact location which is closest to the center of mass of the object has moderate impulse value irrespective of the different direction of relative angular velocity, i.e. minimizing the effect of motion uncertainty on the impact, which is consistent with *Hypothesis 2* in Section III-C.

B. Impact-Aware Contact Searching for Moving Objects

In this section, we will demonstrate the effectiveness of the proposed impact-aware contact selection algorithm (see Section IV). Given any kind of mesh that describes the shape of an object and the object velocity (linear and angular), the proposed algorithm can find the optimal contact locations, which are resistant to impact. Without loss of generality, here we consider four objects with different shapes and velocities, i.e. cube (side length $l = 1.0m$), sphere (radius $r = 0.5m$), dodecahedron (side length $l = 0.3m$) and torus (major radius $R = 0.225m$, minor radius $r = 0.075m$). The cost function weights of (12) are $w_1 = w_2 = 2.0$ and $w_3 = 1.0$. The results of impact-aware selection algorithm are shown in Fig. 9. Given the velocity of the object (black arrow) and any arbitrary initial

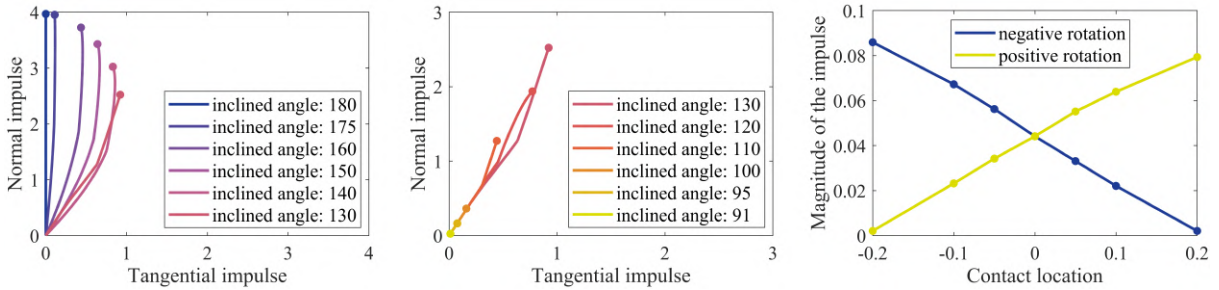


Fig. 8. Impulse distributions of different contact directions and locations (unit: N s). 1) Impulse distributions of the same velocity magnitude and different inclined angles; 2) Impulse distributions of different contact locations when making contact along the normal direction of the motion (positive angular velocity); 3) Impulse distributions of different contact locations when making contact along the normal direction of the motion (negative angular velocity).

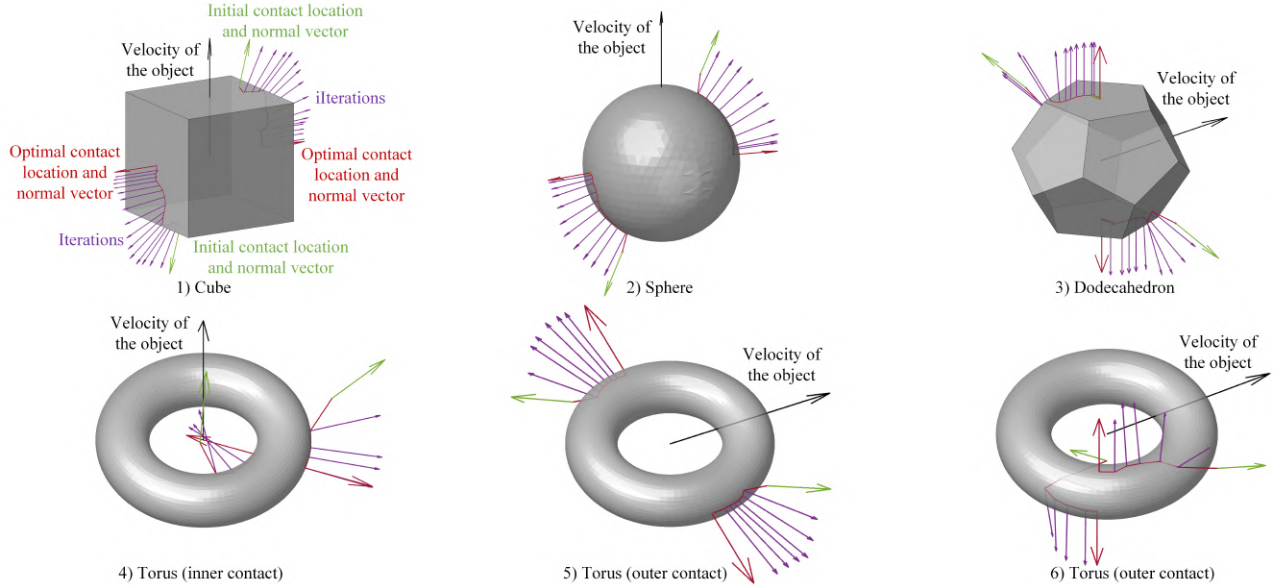


Fig. 9. Contact searching for objects with different kind of shapes (noncontinuous and nonconvex): 1) Cube; 2) Sphere; 3) Dodecahedron; 4/5/6) Torus. Black arrow: the velocity of the object; Green arrow: initial contact location and its corresponding normal vector on the contact surface; Red arrow: optimal contact location and its corresponding normal vector on the contact surface; Purple arrow: the iteration of contact location and normal vector.

contact location (green arrows), the optimal contact locations and their corresponding normal vectors on the surface (red arrow) can be found in $10m.s$ with the convergence threshold of contact locations being $10mm$. The contact points along with their normal vectors during the iterations of the algorithm are shown with the purple arrows. Also, note that the proposed algorithm is able to select the optimal contact locations across discontinuous and non-convex contact surfaces, due to its sequential linearization nature and the smoothing of the normal vectors between different contact surfaces. Especially for the case shown in Fig. 9 (4-6), the proposed algorithm is able to find both the inner and outer contacts for the torus with the non-convex shape.

C. Impact-Agnostic vs Impact-Aware Methods

Here, we compare the planned trajectories generated by the proposed impact-aware MMTO method that uses our contact force transmission model (see Section V-C) against an impact-agnostic method (MMTO without contact force transmission model). The task is specified as catching and stopping a swinging object using two contacts (end-effectors), in which the cost function is defined to minimize the contact force, stiffness and velocity for each robot end-effector. For this

scenario, at the moment of contact ($t \approx 0.55s$) the object has a linear velocity around $1.0m/s$ and an angular velocity around $0.5 rad/s$. Fig. 10 and Fig. 11 show the three dimension trajectories of the object and the end-effectors generated from impact-agnostic method and impact-aware method, respectively, while Fig. 12 shows the corresponding force profiles.

At the moment of contact, the impact-agnostic trajectory optimization plans a large contact force (more than $100 N$) with short contact duration (less than $0.5 s$) for both contacts, which stops the object suddenly (see Fig. 10). However, as shown in Fig. 12, our impact-aware MMTO method minimizes planned contact force at the initial phase of the contact making process, to establish a stable contact, and after that the contact force is increased such that the desired manipulation task (stopping the moving object) is achieved. This corresponds to smooth robot motion and contact force, consequently smooth stopping of the object motion, as shown in Fig. 11.

VII. EXPERIMENT RESULTS

In this section, we describe the experimental setup (both hardware and software) and the scenarios adopted to validate the proposed impact-aware methods in the real world. The goal is to minimize the impulsive force when halting or

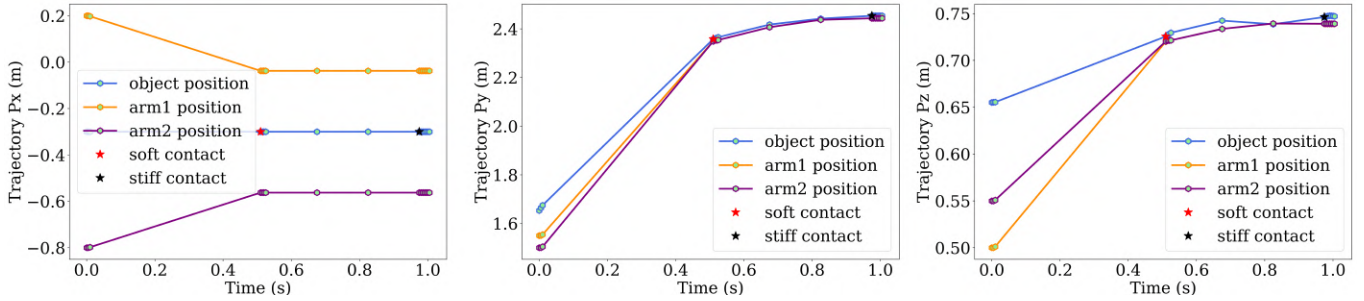


Fig. 10. Trajectory optimization results of impact-agnostic method. The red star and black star belong to the single contact phase of the impact-agnostic method, there is no difference between them.

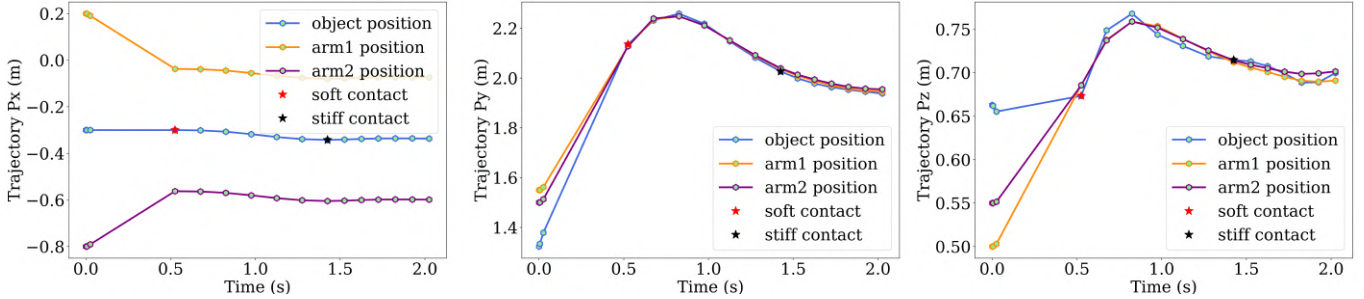


Fig. 11. Trajectory optimization results of impact-aware method. The red star represents the beginning of the first contact phase, while the black star represents the beginning of the second contact phase.

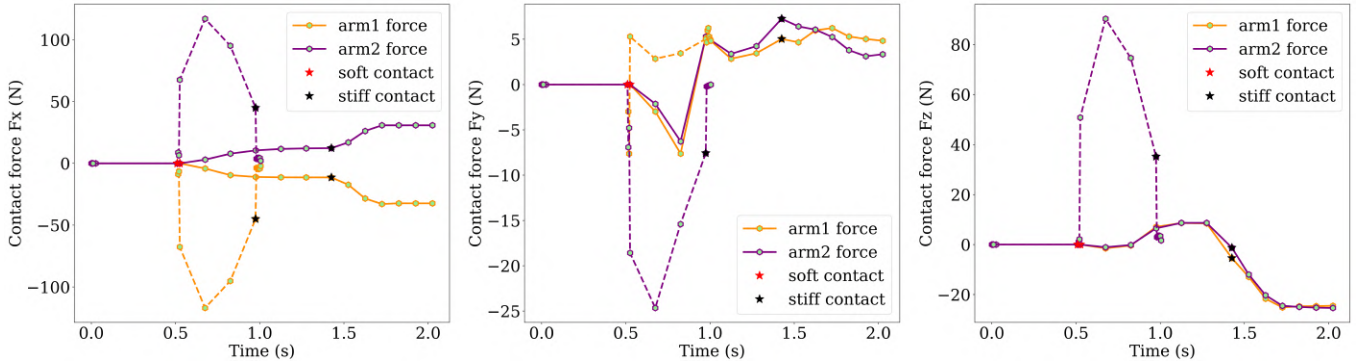


Fig. 12. Desired contact forces resulted from impact-aware (solid line) and impact-agnostic (dashed line) MMTO method. The red and black stars are respectively used to denote the beginning of compression contact phase (soft contact) and restitution contact phase (stiff contact) for impact-aware method in Fig. 11, while for impact-agnostic method in Fig. 10 both of them belong to the single contact phase.

manipulating a moving object given only measurements of the object's pose. As shown in the attached video⁴, this task is realized under different conditions where the object follows: (i) a definite motion on a conveyor belt (see Fig. 13), (ii) a constrained (swinging) motion by being tethered to the ceiling and (iii) a free-flying motion after being thrown.

Next, we will first introduce the components of hardware setup and implementation details of each module in Fig. 3. Then we compare the experiment results of impact-agnostic method and the proposed impact-aware method. We further investigate the benefit of indirect force controller which generates penetration depth from desired contact force and stiffness, compared with the prescribed penetration method which uses a fixed penetration depth regardless of contact force and stiffness, both of them are used to track the desired motion and contact force from MMTO. At last, we validate and demonstrate the capabilities of the proposed system on the scenario of catching a free-flying object.

A. Experiment Setup

The hardware setup includes a 4.2 kg object with dimensions $0.55m \times 0.40m \times 0.42m$, a motion capture system (Vicon) which measures the 3D pose of the object with 250 Hz and two *KUKA-iiwa* compliant robots (LBR iiwa 14 R820) equipped with an ATI F/T sensor that measures force with 100 Hz at each end-effector. All force reported here are raw without any filtering. All the experiments are conducted on a 64-bit Intel Quad-Core i9 3.60GHz computer with 64GB RAM. The proposed system communicates with the *KUKA iiwa* robots through the Fast Research Interface (FRI) which allows us to change the stiffness parameter of the robots on-the-fly. Next, we provide a sequential description of the modules of the system (also see Section I-B).

Estimation: For all the experiments we estimate on-the-fly the velocity (linear and angular) of the object using an EKF [26] based on the *robot_localization* package⁵. The

⁴ https://youtu.be/1__FuHY3-qo

⁵ https://github.com/cra-ros-pkg/robot_localization

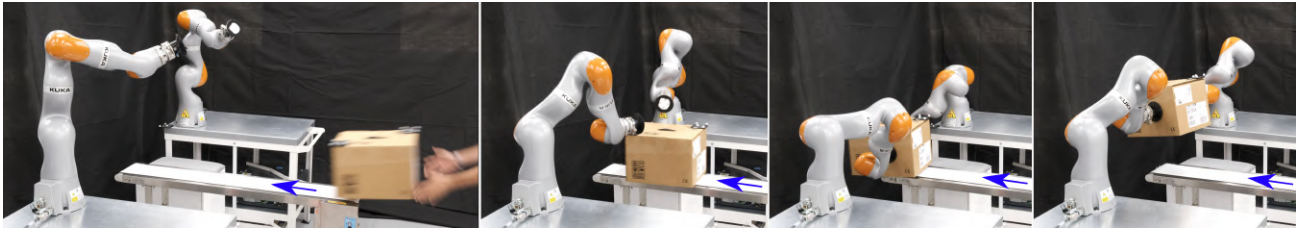


Fig. 13. Two KUKA iiwa arms capture a moving object which is thrown onto the conveyor belt by an operator.

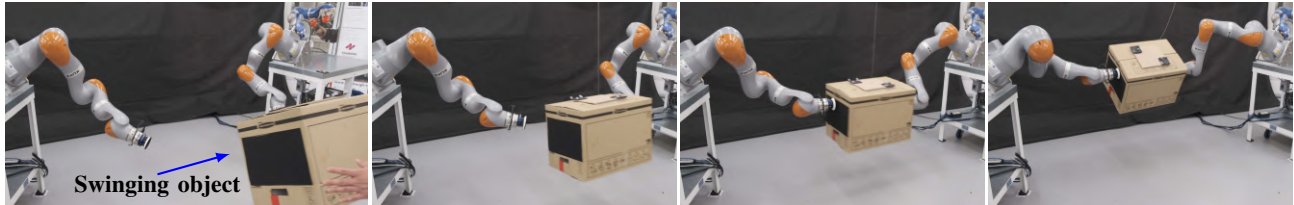


Fig. 14. Snapshots of the two KUKA-iiwa robots catching a swinging object.

tuning of the EKF parameters, such as the frequency of 250Hz, and the queue size of 5 were done using the *pyBullet* physics simulation [71] in which we have access to ground-truth velocities of the object. Also, starting the object with zero or small velocities is key to obtaining accurate velocity estimations for the spinning/flying motions.

Prediction: Given the pose and the estimated velocity of the object, its future trajectory is predicted by solving an IVP. The IVP is a nonlinear program similar in form to (17), but only simpler, as it only considers the nominal dynamics (without modes) and geometric constraints that describe the environment, *e.g.* tether of the swinging object. For the IVP, we selected $N = 60$ knots with $\Delta T_n = 0.03s \forall n = \{0, \dots, N\}$ to predict the object motion going through the dual-arm workspace while it can be solved in less than 25 ms (using pre-stored initial guess).

Contact selection: Once the predicted trajectory of the object is computed, the object pose (time instance of the trajectory) that is closest to the center of the workspace of the bimanual robot system is selected. Given this object pose, the CD-SQP contact selection algorithm is used to determine optimal contact locations for impact-aware catching. This is particularly useful for scenarios where the object is tumbling during the swinging motion, as shown in Fig. 17. For this setup, two robots were used. Hence, $K = 2$; the selected weights are $w_1 = w_2 = 2$ and $w_3 = 1$ to prioritise impact mitigation over distribution of contact locations; and $\Delta p_{min} = 10mm$, $\Delta p_{max} = 50mm$ to ensure convergence of locally optimal contact locations in less than 10 ms.

Multi-mode planning: Given the above as initial conditions (predicted object pose and velocity) and contact information (optimal contact location), the proposed MMTO algorithm is adopted to compute the optimal trajectories in real-time to capture and halt or manipulate the moving object. The MMTO simultaneously generates the optimal end-effector motion, force and stiffness profiles for the two robot arms, before, during and after contact. For all the experiments, we compute hybrid plans with the MMTO in less than 260 ms (using stored initial guess). Note that, to achieve the real-time computation, we use a rough discretization with $N = 12$ collocation knots

(4 for free motion, 4 for soft contact phase and 4 for stiff contact phase) with $0.05 \leq \Delta T_n \leq 0.4$ for free motion phase and $0.01 \leq \Delta T_n \leq 0.1$ for two contact phases. Parameter tuning was performed in simulation with various initial object poses and velocities.

Indirect force-control and IK: As the output of the MMTO are end-effector trajectories in Cartesian space, these are mapped to the robot motion in configuration space using differential inverse kinematics and are streamed to the two position-controlled *KUKA-iiwa* robots along with the prescribed Cartesian impedance (stiffness profiles). Further, in order to track the optimal contact force planned with the MMTO, we use an indirect force control scheme to regulate the end-effector's penetration of the *KUKA-iiwa* robots, which is detailed in Appendix-B. Commanding stiffness enables impact-aware contact transitions, while the indirect force control enables fast contact force tracking without oscillations.

The communication between modules is realized using *ROS*, while the prediction (IVP) and the multi-mode planning (MMTO) are realized using *CasADi* with its automatic differentiation and solved by the *Ipopt* solver.

B. Impact-Agnostic vs Impact-Aware Method

Here, similar to Section VI-C, we compare the proposed impact-aware MMTO that uses our contact force transmission model (see Section V-C) against an impact-agnostic method (MMTO without contact force transmission model). In this scenario (see Fig. 14), the object is tethered to the ceiling (tether length is 3m) and it is released from an initial position (at an angle of 20°) with zero linear and angular velocities. This results in a maximum speed at the contact moment being around 1.8m/s. We perform the catching task with both the impact-agnostic and the impact-aware methods, and we plot the normal contact forces in Fig. 15.

It can be seen that the impact-agnostic method results in a large impulsive contact force at and right after the contact moment, *i.e.* 188N and 213N for two arms, respectively. In contrast, our impact-aware MMTO results in smoothly-increasing contact forces, due to the low stiffness planned for making contact and the high stiffness for post-contact

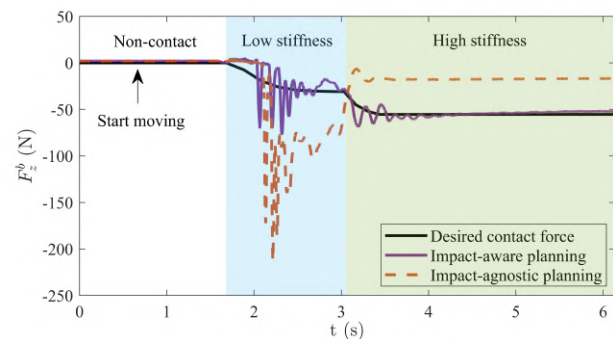
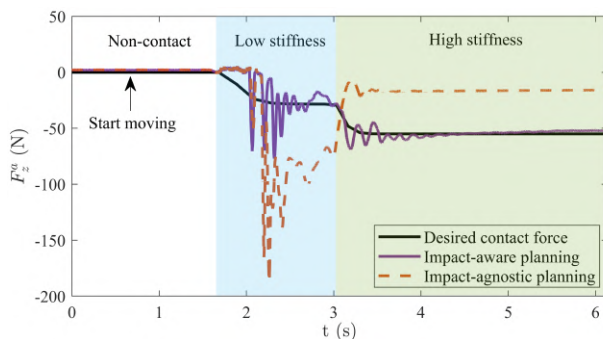


Fig. 15. Normal contact forces of two *KUKA-iiwa* corresponding to impact-aware planning and impact-agnostic planning.

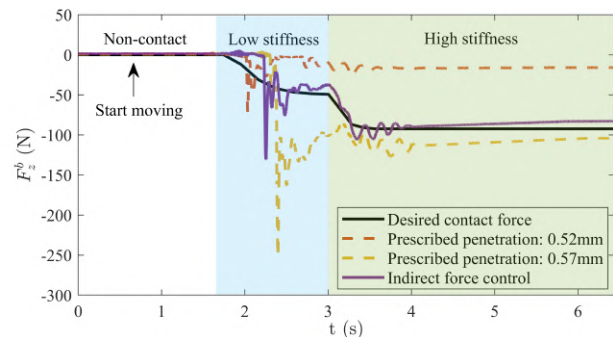
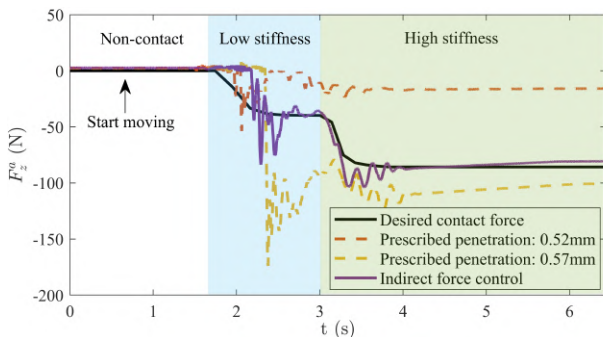


Fig. 16. Normal contact forces of two *KUKA-iiwa* corresponding to indirect force control and prescribed penetration.

manipulation. The maximum contact force is around $77N$ and $119N$ for both arms, which are 59% and 44% smaller than the ones resulted from impact-agnostic method. Further, it is noteworthy that compared with the impact-aware method (16 successful catches in 20 trials), the impact-agnostic method (3 successful catches within 10 trials) is more prone to lose the object (bouncing off) due to the large impulsive force at the moment of contact.

C. Indirect Force Control vs Prescribed Penetration

On the same setup as in Section VII-B, the swinging object is released from a higher initial position which brings the object to a speed of around 2 m/s at the moment of contact. The impact-aware MMTO is used to plan the dual-arm motion and contact force profiles for catching the swinging object. Yet, the joint position controller with Cartesian impedance control mode of the *KUKA-iiwas* does not allow tracking the force profile directly. To be able to exert the desired force at the end-effector of the robot using Cartesian impedance control mode of the *KUKA-iiwas*, we adopted an indirect force control scheme (see Appendix-B). The penetration depth of the robot end-effector is calculated based on the optimal contact force and stiffness, both computed from the impact-aware MMTO, and it is compared against a prescribed penetration approach.

For the prescribed penetration approach, a constant penetration depth (0.52mm and 0.57mm are respectively used) is defined for each robot to generate a contact force and make a stable contact with the swinging object. However, as shown in Fig. 16, this will either result in large impulsive force at the moment of contact (penetration depth of 0.57 mm results in $174N$ and $251N$ for the two arms, respectively) or result in small contact forces which is not able to hold the object (penetration depth of 0.52mm). In contrast, as shown

in Fig. 16, the exerted force using the indirect force control scheme (purple solid line) are $84N$ and $130N$, which are respectively 48% and 52% smaller than the ones resulting from prescribed penetration method (brown dashed line).

D. Stiffness Adjustment when Catching Tumbling Objects

Here, we present a setup where the two *KUKA-iiwa* robots catch a tumbling-swinging object that rotates with angular velocity $\geq 112^\circ/\text{s}$, as shown in Fig. 17. This scenario encompasses the challenge of selecting contact locations on-the-fly, which is addressed using the CD-SQP (see Section IV).

In addition, when respectively considering a positive and a negative tumbling motion as shown in Fig. 18, the dual-arm catching motion generated from the proposed algorithm is asymmetric. This implies that one of the two arms needs to contribute more to stop the motion of the object, which is also reflected in the planned optimal stiffness profiles. For the positive tumbling motion (see Fig. 18(a)), higher stiffness is planned for the right robot which actually decelerates the object, while for the negative tumbling motion the left robot has higher stiffness that is used to decelerate the motion of the object. It is worthy to note that the impact-aware MMTO algorithm is able to autonomously optimize the stiffness for each robot to achieve a soft catching behavior according to different initial conditions.

E. Catching Free-Flying Objects

In order to further validate and demonstrate the capabilities of the proposed system, we consider the scenario of a free-flying object, as shown in Fig. 19. The object is thrown approximately 2.7m away from the robots and close to the moment of contact, travels with linear velocity $\geq 3.5\text{ m/s}$ on y axis and $\geq 2\text{ m/s}$ on z axis of the motion. For this scenario, the

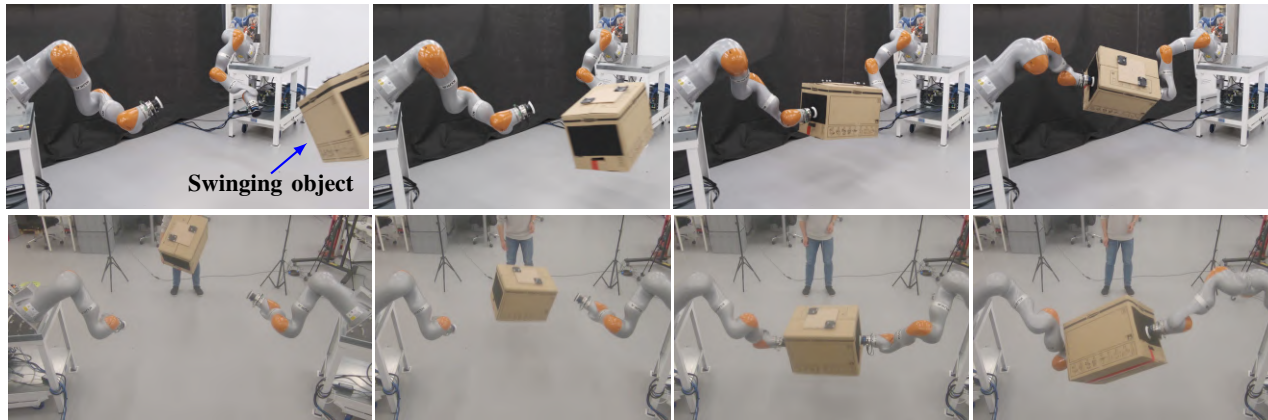


Fig. 17. Snapshots of the two KUKA-iiwa robots catching a swinging and tumbling object; side view camera (above), top-view camera (below).

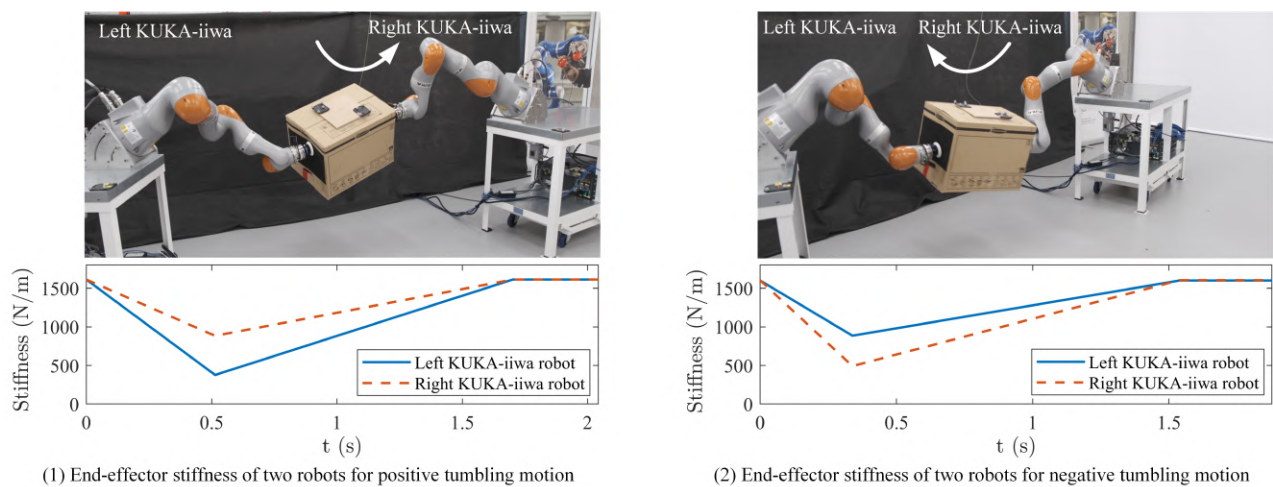


Fig. 18. End-effector stiffness of two robots for positive tumbling motion and negative tumbling motion.

system performs *estimation*, *prediction*, *contact selection* and *hybrid planning* in less than 350ms. The catching motion is truly at the limits of the robot hardware with the end-effectors of the arms reaching speeds higher than 2m/s. Yet, as it is demonstrated, the two *KUKA-iiwa* robots are able to catch a flying object that weighs 4.2kg. Although the two robots can catch the tumbling and flying objects, we found that the robots frequently go to E-stop due to the very high accelerations and speed of the motions. For this kind of extreme manipulation tasks, we encounter limitations that emerge not only from the computation time of the proposed algorithm but also the maximum payload and speed of the robot hardware. We will discuss the limitations in detail in Section VIII.

VIII. DISCUSSION

Here, we discuss practical considerations, some limitations that hinder the performance and capabilities of the proposed system and possible extensions of the proposed work.

1) *IK with joint limit constraints*: Throughout the experiments, we found out that the feasible volume where catching can be realised is very small due to the limited coordinated dual-arm workspace, the joint limits of the robots, singularity configurations and the fast moving object. These made it difficult to demonstrate the more extensive capabilities of

the proposed algorithm, as for each scenario we needed to carefully choose the initial configurations of the robots to avoid the robots stopping in the middle of the motion due to joint limits. An IK algorithm that can avoid joint limits and singularities could further enhance the robustness and success rate of the proposed system.

2) *Tracking the planned motions*: Given the very high speed of the moving object, the two robots need to accelerate with the maximum acceleration to follow the moving object at their maximum speed and catch it. Following the object motion requires accurate tracking of the planned motions (output of MMTO), which deteriorates as: (i) each robot can not immediately reach its maximum velocity, due to its own mass and inertia, and (ii) accurate position tracking with a compliance controller, such as the Cartesian impedance control mode of the *KUKA-iiwas*, cannot be achieved out-of-the-box. As a result, the mismatch between the planned motion and the actual motion of the robots can result in real-world catching failures, due to the mismatch at the moment of contact. One could improve the tracking performance with a Model Predictive Controller that has a prediction horizon and takes into account the hardware specifications, as well as any discrepancies resulting from the impedance controller.

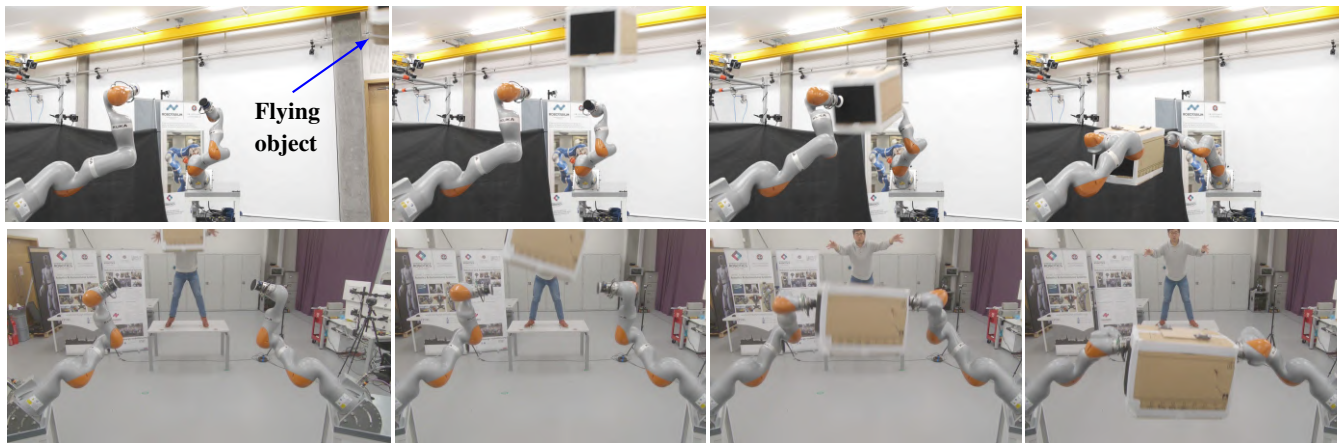


Fig. 19. Snapshots of the two KUKA-iiwa robots catching a flying object thrown by a human; (above) side-view camera. (below) top-view camera.

3) *Computation time*: To catch a large-momentum object, the allowed computation time to estimate and predict the object's motion and plan the motion of the arms is limited. For the scenarios considered in this paper, it is less than 0.5s. Both the IVP and the MMTO are formulated as non-linear programming and are solved with *Ipopt*. Hence, in order to meet the allowed computation time bounds, we need to provide appropriate initial guesses. However, for highly-dynamic manipulation task that involve many variations and uncertainties, such as catching an object that swings or that is thrown by a human, generating an appropriate initial guess might not be easy. Therefore, investigation for more efficient solvers and problem formulations, as well as methods to store and quickly obtain robust initial guesses for warm-starting could be very beneficial towards achieving on-line computation reliably.

4) *Hardware*: Last but not least, the catching task requires high load-to-weight ratio robots. A more lightweight robot would have better performance for such dynamic manipulation tasks, yet it should also have large payload capability. Another useful extension of the system would be to mount each *KUKA-iiwa* onto a mobile base to enlarge their limited workspace.

In summary, the proposed framework is a first attempt to obtain a feasible online optimization approach that can plan and execute catching motions while considering impulsive forces and slippage limits. Further, more detailed impact dynamics modeling is encouraged and required towards achieving more accurate impact predictions.

IX. CONCLUSION

In this paper, we present an optimization framework for catching large-momentum objects whose total movement duration is less than one second and with speeds higher than 3.5 m/s. Further, we developed the first ever system able to realise dynamic bimanual catching of flying objects. This includes motion estimation and prediction, contact selection, multi-mode trajectory optimization, variable-stiffness modulation and motion execution on two *KUKA-iiwa* robots. The executed motions are at the maximum speed limit of the hardware. In our optimization framework, based on the 3D compliance

impact model, we propose impact-aware contact selection and impact-aware motion planning in order to mitigate impulsive forces when the robots make contact with the moving object. A variety of bimanual catching simulations and real-world experiments validate and demonstrate the effectiveness of the proposed methods and system. The scenarios extend from grasping a moving object on a conveyor belt to catching a tethered tumbling object, and even catching a flying object thrown by a human.

Our future work will focus on enabling our MMTO formulation to modify the contact sequence schedule with the object (continuing our work beyond simultaneous contacts for the two arms) as well as to achieve on-line computation with offline knowledge, towards demonstrating different type of dynamic bimanual manipulation behaviours.

APPENDIX

A. Energy-based 3D Impact Model

In this appendix, we will briefly introduce the energy-based impact dynamics model – more details can be found in [31].

Based on the 3D compliance impact model with three spatial springs, the energy stored in these three springs during impact can be calculated. From the three components of the energy, we can determine whether a contact is in slipping mode or sticking mode. Consequently, the impulse and the state of the object are updated till the end of the impact.

1) *Impact modelling with three virtual springs*: By modelling the contact between two bodies with three virtual springs along three unit orthogonal vectors \hat{x} , \hat{y} and \hat{z} , we can decompose the impulse $\mathbf{\Lambda}$, contact force \mathbf{F} , and contact velocity \mathbf{v} along the three directions:

$$\mathbf{\Lambda} = \Lambda_x \hat{x} + \Lambda_y \hat{y} + \Lambda_z \hat{z}, \quad (30)$$

$$\mathbf{F} = F_x \hat{x} + F_y \hat{y} + F_z \hat{z}, \quad (31)$$

$$\mathbf{v} = v_x \hat{x} + v_y \hat{y} + v_z \hat{z}. \quad (32)$$

where \hat{z} is the normal vector of contact tangential plane; \hat{x} is the opposite direction of the tangential component of initial contact velocity \mathbf{v}_{tan} , where $\mathbf{v}_{tan} = \mathbf{v} - v_z \hat{z}$; and $\hat{y} = \hat{z} \times \hat{x}$.

At the same time, the changes in length of the three springs are respectively denoted as u , w , and n . Correspondingly the rates of change of the three springs are denoted as \dot{u} , \dot{w} , \dot{n} . Therefore, the contact forces and energies stored in the three springs can be obtained as:

$$F_x = -K_t u, \quad F_y = -K_t w, \quad F_z = -K_n n, \quad (33)$$

$$E_x = \frac{1}{2} K_t u^2, \quad E_y = \frac{1}{2} K_t w^2, \quad E_z = \frac{1}{2} K_n n^2. \quad (34)$$

where K_t and K_n are the stiffness of the springs along tangential and normal directions.

In order to analyse the impact mechanism, the normal impulse, denoted with Λ_z , is taken as the independent variable instead of time. We can represent the change rate of the normal impulse Λ_z in terms of energy E_z as

$$\dot{\Lambda}_z = \frac{d\Lambda_z}{dt} = F_z = \sqrt{2K_n E_z}, \quad (35)$$

the derivatives of the other two tangential impulses and the normal energy with respect to I_z can be derived as:

$$\Lambda'_x = \frac{\dot{\Lambda}_x}{\dot{\Lambda}_z} = -\sigma_x \sqrt{\frac{2K_t E_x}{2K_n E_z}} = -\frac{\sigma_x}{\eta} \sqrt{\frac{E_x}{E_z}}, \quad (36)$$

$$\Lambda'_y = \frac{\dot{\Lambda}_y}{\dot{\Lambda}_z} = -\sigma_y \sqrt{\frac{2K_t E_y}{2K_n E_z}} = -\frac{\sigma_y}{\eta} \sqrt{\frac{E_y}{E_z}}, \quad (37)$$

$$E'_z = \frac{dE_z}{d\Lambda_z} = \frac{\dot{E}_z}{\dot{\Lambda}_z} = \frac{kn\dot{n}}{-k\dot{n}} = -\dot{n} = -v_z, \quad (38)$$

where σ_x and σ_y are set to 1 if the springs are extended and are set to -1 if they are compressed; $\eta = \eta_0$ during the compression stage and $\eta = \eta_0/e$ during the restitution stage, where $\eta_0^2 = K_n/K_t$ is the constant stiffness ratio and e is the energetic coefficient of restitution.

2) *Slipping and sticking contact modes*: Here, we describe the conditions that specify slipping and sticking contact modes and their transitions from one to the other. According to Coulomb's law, the contact between two bodies sticks if

$$\sqrt{F_x^2 + F_y^2} < \mu F_z. \quad (39)$$

From (33), (34) and (39) we obtain the following

$$E_x + E_y < \mu^2 \eta^2 E_z. \quad (40)$$

Therefore, if a contact is in sticking mode, and $E_x + E_y = \mu^2 \eta^2 E_z$, the contact mode will switch to slipping mode. Otherwise, if contact is in slipping mode and $v_t - \dot{u}\hat{x} - \dot{w}\hat{y} = 0$, the contact mode will switch to sticking mode.

3) *Impulse evolution along normal and tangential directions*: Given the current contact mode, we can determine the change rates of the tangential springs, based on

$$\dot{u} = \begin{cases} v_x, & \text{stick,} \\ \frac{\sigma_x \mu^2 \eta^3 E'_z \sqrt{E_z E_x} + v_x E_y - \sigma_x \sigma_y v_y \sqrt{E_x E_y}}{\mu^2 \eta^2 E_z}, & \text{slip,} \end{cases} \quad (41)$$

$$\dot{w} = \begin{cases} v_y, & \text{stick,} \\ \frac{\sigma_y \mu^2 \eta^3 E'_z \sqrt{E_z E_y} + v_y E_x - \sigma_x \sigma_y v_x \sqrt{E_x E_y}}{\mu^2 \eta^2 E_z}, & \text{slip.} \end{cases} \quad (42)$$

Further, the energy stored in the three springs is obtained by

$$E_x = \frac{G_x^2}{4\eta_0^2}; \quad E_y = \frac{G_y^2}{4\eta_0^2}; \quad \text{and } E_z = E_z + E'_z \Delta\Lambda_z, \quad (43)$$

where G_x and G_y are used to track the changes in length of the two tangential springs whose initial values are 0, and their derivatives with respect to the normal impulse Λ_z are

$$\begin{bmatrix} G'_x \\ G'_y \end{bmatrix} = \begin{cases} G'_x = \frac{\dot{u}}{\sqrt{E_z}}, G'_y = \frac{\dot{w}}{\sqrt{E_z}}, & \text{if compression,} \\ G'_x = e \frac{\dot{u}}{\sqrt{E_z}}, G'_y = e \frac{\dot{w}}{\sqrt{E_z}}, & \text{if restitution.} \end{cases} \quad (44)$$

According to (30) and (38), the impulse along normal and tangential directions are updated in $(i+1)^{th}$ iteration as:

$$\Lambda_{i+1} = \Lambda_i + (\Lambda'_x \hat{x} + \Lambda'_y \hat{y} + \hat{z}) \Delta\Lambda_z \quad (45)$$

4) *Impact state transition*: As shown in Fig. 5, the impact starts with compression phase, then it will switch to restitution stage when the normal contact velocity decrease to zero and the strain energy E_n reaches its maximum value. During the restitution stage, the strain energy will decrease to zero till the impact ends. At the end of the impact the velocity of the object can be updated according to (3) and (4) as:

$$\Delta v = \frac{\Lambda}{M}, \quad \Delta \omega = \mathbf{I}^{-1} \mathbf{r} \times \Lambda, \quad (46)$$

where the manipulator is assumed to be fixed (not moving) at the moment of making contact with the object.

B. Indirect Force Control

In indirect force control (IFC) schemes [64], the physical robot is coupled with a virtual robot through a virtual mechanical relationship, such as mass-spring-damping system. By modulating the set point of the virtual robot, the desired contact force and interaction behavior of the physical robot can be achieved. Here, instead of using a configuration space IFC controller [64], we design a Cartesian IFC controller.

Throughout our experiments, the two *KUKA-iiwa* robots are position-controlled in Cartesian impedance mode, which means that we send the desired joint positions and the end-effector stiffness to the robot controller. In addition, the relationship between the interaction force and end-effector displacement can be written as:

$$\mathbf{F} = \mathbf{K} (\mathbf{x}_v - \mathbf{x}) - \mathbf{D} \dot{\mathbf{x}} \quad (48)$$

where \mathbf{K} and \mathbf{D} are respectively the stiffness and damping matrices of the end-effector of robot, \mathbf{x}_v and \mathbf{x} are respectively the end-effector positions of virtual robot and physical robot, \mathbf{F} is the contact force at the end-effector.

As the penetration speed $\dot{\mathbf{x}}$ is relatively small when the robot's end-effector is in contact with the object (after the compression stage), based on (48) we can obtain the desired end-effector motion \mathbf{x}_d from the following set-point generator:

$$\mathbf{x}_d = \mathbf{x} + \Delta \mathbf{x} = \mathbf{x} + \frac{\mathbf{F}}{\mathbf{K}}. \quad (49)$$

In summary, we will first generate the nominal end-effector motion \mathbf{x} , the desired contact force \mathbf{F} and end-effector stiffness \mathbf{K} from our MMTO algorithm. Then we use the set-point generator (49) to transform the contact force \mathbf{F} to the penetration depth $\Delta \mathbf{x}$ of the robot end-effector. In this way,

we can track the desired force trajectory with specific end-effector stiffness based on joint position control with Cartesian impedance mode.

REFERENCES

- [1] N. C. Daffe, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic dexterity: In-hand manipulation with external forces," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 1578–1585.
- [2] A. Rodriguez, "The unstable queen: Uncertainty, mechanics, and tactile feedback," *Sci. Robot.*, vol. 6, no. 54, pp. 1–2, 2021.
- [3] D. J. Braun, F. Petit, F. Huber, S. Haddadin, P. Van Der Smagt, A. Albu-Schäffer, and S. Vijayakumar, "Robots driven by compliant actuators: Optimal control under actuation constraints," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1085–1101, 2013.
- [4] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1307–1319, 2020.
- [5] M. Jongeneel, A. Bernardino, N. van de Wouw, and A. Saccon, "Model-based 6d visual object tracking with impact collision models," in *Proc. Am. Control Conf.*, 2022, pp. 3850–3856.
- [6] J. Moura, T. Stouraitis, and S. Vijayakumar, "Non-prehensile planar manipulation via trajectory optimization with complementarity constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 970–976.
- [7] Y.-B. Jia, M. Gardner, and X. Mu, "Batting an in-flight object to the target," *Int. J. Robot. Res.*, vol. 38, no. 4, pp. 451–485, 2019.
- [8] S. Kim, A. Shukla, and A. Billard, "Catching objects in flight," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1049–1065, 2014.
- [9] S. S. M. Salehian, M. Khoramshahi, and A. Billard, "A dynamical system approach for softly catching a flying object: Theory and experiment," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 462–471, 2016.
- [10] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, "Trajectory planning for optimal robot catching in real-time," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3719–3726.
- [11] L. Yan, Z. Mu, W. Xu, and B. Yang, "Coordinated compliance control of dual-arm robot for payload manipulation: Master-slave and shared force control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 2697–2702.
- [12] L. Yan, Y. Yang, W. Xu, and S. Vijayakumar, "Dual-arm coordinated motion planning and compliance control for capturing moving objects with large momentum," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7137–7144.
- [13] M. Bombile and A. Billard, "Dual-arm control for coordinated fast grabbing and tossing of an object: Proposing a new approach," *IEEE Robot. Autom. Mag.*, vol. 29, no. 3, pp. 127–138, 2022.
- [14] B. Bäuml, T. Wimböck, and G. Hirzinger, "Kinematically optimal catching a flying ball with a hand-arm-system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 2592–2599.
- [15] C. G. Prevost, A. Desbiens, and E. Gagnon, "Extended kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle," in *Proc. Am. Control Conf.*, 2007, pp. 1805–1810.
- [16] R. Lampariello, "Optimal motion planning for object interception and capture," Ph.D. dissertation, Technical University of Darmstadt, 2021.
- [17] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.
- [18] W. Yang and M. Posa, "Impact invariant control with applications to bipedal locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 5151–5158.
- [19] Y. Wang, N. Dehio, and A. Kheddar, "Predicting impact-induced joint velocity jumps on kinematic-controlled manipulator," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6226–6233, 2022.
- [20] Y. Wang and A. Kheddar, "Impact-friendly robust control design with task-space quadratic optimization," in *Proc. Robot.: Sci. Syst.*, 2019, pp. 1–9.
- [21] T. Stouraitis, L. Yan, J. Moura, M. Gienger, and S. Vijayakumar, "Multi-mode trajectory optimization for impact-aware manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9425–9432.
- [22] K. Dong, K. Pereida, F. Shkurti, and A. P. Schoellig, "Catch the ball: Accurate high-speed motions for mobile manipulators via inverse dynamics learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 6718–6725.
- [23] Y. Masutani, T. Iwatsu, and F. Miyazaki, "Motion estimation of unknown rigid body under no external forces and moments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 1066–1072.
- [24] F. Aghili, "A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 634–649, 2012.
- [25] H. Yu, D. Guo, H. Yin, A. Chen, K. Xu, Z. Chen, M. Wang, Q. Tan, Y. Wang, and R. Xiong, "Neural motion prediction for in-flight uneven object catching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 4662–4669.
- [26] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Proc. Int. Conf. Intell. Auton. Syst.*, 2014, pp. 1–6.
- [27] W. J. Stronge, *Impact mechanics*. Cambridge university press, 2018.
- [28] R. M. Brach, "Rigid Body Collisions," *J. Appl. Mech.*, vol. 56, no. 1, pp. 133–138, 1989.
- [29] J. B. Keller, "Impact With Friction," *J. Appl. Mech.*, vol. 53, no. 1, pp. 1–4, 1986.
- [30] D. E. Stewart, "Rigid-body dynamics with friction and impact," *SIAM Review*, vol. 42, no. 1, pp. 3–39, 2000.
- [31] Y.-B. Jia, "Three-dimensional impact: energy-based modeling of tangential compliance," *Int. J. Robot. Res.*, vol. 32, no. 1, pp. 56–83, 2013.
- [32] G. Gilardi and I. Sharf, "Literature survey of contact dynamics modelling," *Mech. Mach. Theory*, vol. 37, no. 10, pp. 1213–1239, 2002.
- [33] M. Nagurka and S. Huang, "A mass-spring-damper model of a bouncing ball," in *Proc. Am. Control Conf.*, 2004, pp. 499–504.
- [34] A. M. Johnson, S. A. Burden, and D. E. Koditschek, "A hybrid systems model for simple manipulation and self-manipulation systems," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1354–1392, 2016.
- [35] M. T. Mason, "Progress in nonprehensile manipulation," *Int. J. Robot. Res.*, vol. 18, no. 11, pp. 1129–1141, 1999.
- [36] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Auton. Robots*, vol. 38, no. 1, pp. 65–88, 2014.
- [37] F. Lamiroux and J. Mirabel, "Prehensile manipulation planning: Modeling, algorithms and implementation," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2370–2388, 2021.
- [38] X. Markenscoff, L. Ni, and C. H. Papadimitriou, "The geometry of grasping," *Int. J. Robot. Res.*, vol. 9, no. 1, pp. 61–74, 1990.
- [39] A. Bicchi, "On the closure properties of robotic grasping," *Int. J. Robot. Res.*, vol. 14, no. 4, pp. 319–334, 1995.
- [40] K. M. Lynch and M. T. Mason, "Dynamic nonprehensile manipulation: Controllability, planning, and experiments," *Int. J. Robot. Res.*, vol. 18, no. 1, pp. 64–92, 1999.
- [41] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1711–1718, 2018.
- [42] M. Toussaint, K. Allen, K. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," in *Proc. Robot.: Sci. Syst.*, 2018, pp. 1–9.
- [43] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 755–773, 2020.
- [44] Y. Hou and M. T. Mason, "Robust execution of contact-rich motion plans by hybrid force-velocity control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 1933–1939.
- [45] T. Stouraitis, I. Chatzinikolaïdis, M. Gienger, and S. Vijayakumar, "Online hybrid motion planning for dyadic collaborative manipulation via bilevel optimization," *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1452–1471, 2020.
- [46] M. Toussaint, J. Harris, J.-S. Ha, D. Driess, and W. Hönl, "Sequence-of-constraints mpc: Reactive timing-optimal control of sequential manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 13 753–13 760.
- [47] A. Pekarovskiy, F. Stockmann, M. Okada, and M. Buss, "Hierarchical robustness approach for nonprehensile catching of rigid objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3649–3654.
- [48] J. Zhou, Y. Hou, and M. T. Mason, "Pushing revisited: Differential flatness, trajectory planning, and stabilization," *Int. J. Robot. Res.*, vol. 38, no. 12–13, pp. 1477–1489, 2019.
- [49] M. Toussaint, J.-S. Ha, and D. Driess, "Describing physics for physical reasoning: Force-based sequential manipulation planning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6209–6216, 2020.
- [50] T. Stouraitis, I. Chatzinikolaïdis, M. Gienger, and S. Vijayakumar, "Dyadic collaborative manipulation through hybrid trajectory optimization," in *Proc. Conf. Robot Learn.*, 2018, pp. 869–878.
- [51] M. Murooka, K. Okada, and M. Inaba, "Optimization-based posture generation for whole-body contact motion by contact point search on the body surface," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2905–2912, 2020.

- [52] D. Driess, J.-S. Ha, M. Toussaint, and R. Tedrake, "Learning models as functionals of signed-distance fields for manipulation planning," in *Proc. Conf. Robot Learn.*, 2022, pp. 245–255.
- [53] N. Dehio, Y. Wang, and A. Kheddar, "Dual-arm box grabbing with impact-aware mpc utilizing soft deformable end-effector pads," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5647–5654, 2022.
- [54] H. Khurana, M. Bombile, and A. Billard, "Learning to hit: A statistical dynamical system based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 9415–9421.
- [55] Y. Wang, N. Dehio, and A. Kheddar, "On inverse inertia matrix and contact-force model for robotic manipulators at normal impacts," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3648–3655, 2022.
- [56] W. Amanhoud, M. Khoramshahi, and A. Billard, "A dynamical system approach to motion and force generation in contact tasks," in *Proc. Robot.: Sci. Syst.*, 2019, pp. 1–10.
- [57] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *Int. J. Robot. Res.*, vol. 28, no. 11-12, pp. 1507–1527, 2009.
- [58] A. Saccon, N. van de Wouw, and H. Nijmeijer, "Sensitivity analysis of hybrid systems with state jumps with application to trajectory tracking," in *Proc. IEEE Conf. Decis. Control*, 2014, pp. 3065–3070.
- [59] M. Rijnen, A. Saccon, and H. Nijmeijer, "On optimal trajectory tracking for mechanical systems with unilateral constraints," in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 2561–2566.
- [60] J. J. van Steen, N. van de Wouw, and A. Saccon, "Robot control for simultaneous impact tasks via quadratic programming-based reference spreading," in *Proc. Am. Control Conf.*, 2022, pp. 3865–3872.
- [61] A. Aydinoglu, V. M. Preciado, and M. Posa, "Contact-aware controller design for complementarity systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 1525–1531.
- [62] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified impedance and admittance control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 554–561.
- [63] L. Roveda, N. Iannacci, F. Vicentini, N. Pedrocchi, F. Braghin, and L. M. Tosatti, "Optimal impedance force-tracking control design with impact formulation for interaction tasks," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 130–136, 2015.
- [64] E. Lutscher, E. C. Dean-León, and G. Cheng, "Hierarchical force and positioning task specification for indirect force controlled robots," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 280–286, 2017.
- [65] Q. Zhou, "Pymesh—geometry processing library for python," *Software available for download at <https://github.com/PyMesh/PyMesh>*, 2019.
- [66] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.
- [67] S. J. Wright, "Coordinate descent algorithms," *Math. Program.*, vol. 151, no. 1, pp. 3–34, 2015.
- [68] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Syst. Mag.*, vol. 29, no. 2, pp. 28–93, 2009.
- [69] M. Rijnen, E. de Mooij, S. Traversaro, F. Nori, N. van de Wouw, A. Saccon, and H. Nijmeijer, "Control of humanoid robot motions with impacts: Numerical experiments with reference spreading control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 4102–4107.
- [70] J. Nakanishi, A. Radulescu, and S. Vijayakumar, "Spatio-temporal optimization of multi-phase movements: Dealing with contacts and switching dynamics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 5100–5107.
- [71] C. Mower, T. Stouraitis, J. Moura, C. Rauch, L. Yan, N. Z. Behabadi, M. Gienger, T. Vercauteren, C. Bergeles, and S. Vijayakumar, "RO-spybullet interface: A framework for reliable contact simulation and human-robot interaction," in *Proc. Conf. Robot Learn.*, 2022, pp. 1411–1423.



Lei Yan received his Ph.D. in Mechanical Engineering from Harbin Institute of Technology, China in 2019. From 2019 to 2021, he was a Research Associate with the School of Informatics, The University of Edinburgh, Edinburgh, UK.

He is currently an Assistant Professor with the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, China. His research interests include dexterous dynamic manipulation, multi-robot collaboration and physical human-robot interaction.



Theodoros Stouraitis received his Ph.D. in Robotics from the University of Edinburgh in UK and in collaboration with the Honda Research Institute Europe in Germany. From 2012 to 2015, he was research assistant at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR), in 2022, he was a Visiting Scientist at Massachusetts Institute of Technology (MIT), USA and from 2022 till 2023, he was a Guest Scientist at the Honda Research Institute Europe in Germany.

He is currently Control Scientist working on robotics and autonomous vehicles projects. His research interests include motion planning and control, non-linear optimization, contacts and human-robot interaction.



João Moura received his Ph.D. in Robotics and Autonomous Systems jointly awarded by Heriot-Watt University and The University of Edinburgh, UK, in 2021. He is currently a Research Associate in the School of Informatics at The University of Edinburgh, and affiliated with The Alan Turing Institute. His research interests include contact-rich and non-prehensile manipulation, trajectory optimization, model predictive control, and imitation learning.



Wenfu Xu (Senior Member, IEEE) received the B.E. and M.E. degrees in Control Engineering from the Hefei University of Technology, Hefei, China, in 2001 and 2003, respectively, and the Ph.D. degrees in Control Science and Engineering from the Harbin Institute of Technology, Harbin, China, in 2007.

He was a Research Associate with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China. He is currently a Professor with the Department of Mechanical and Automation Engineering, Harbin Institute of Technology, Shenzhen, China. His research interests include space robotics, bionic robotics, and cable-driven manipulators.



Michael Gienger received the Diploma degree in Mechanical Engineering from the Technical University of Munich, Germany, in 1998. From 1998 to 2003, he was research assistant at the Institute of Applied Mechanics of the TUM, addressing issues in design and realization of biped robots. He received his Ph.D. degree with a dissertation on "Design and Realization of a Biped Walking Robot". After this, Michael Gienger joined the Honda Research Institute Europe in Germany in 2003, where he currently holds a position as Chief Scientist. His

research interests include mechatronics, robotics, control systems, imitation learning and human-robot interaction.



Sethu Vijayakumar received his Ph.D. in Computer Science and Engineering from the Tokyo Institute of Technology, Japan in 1998. He is the Professor of Robotics at the University of Edinburgh, an adjunct faculty of the University of Southern California, Los Angeles and the founding Director of the Edinburgh Centre for Robotics. His research interests include statistical machine learning, anthropomorphic robotics, multi objective optimisation and optimal control in autonomous systems as well as the study of human motor control. He is the Programme co-

Director for Artificial Intelligence (AI) at The Alan Turing Institute, the UK's national institute for data science and AI. He is a Fellow of the Royal Society of Edinburgh, and winner of the 2015 Tam Dalyell Prize for excellence in engaging the public with science.