Differential Dynamic Programming based Hybrid Manipulation Strategy for Dynamic Grasping

Cheng Zhou, Yanbo Long, Lei Shi, Longfei Zhao, Yu Zheng

***Abstract*— To fully explore the potential of robots for dexterous manipulation, this paper presents a whole dynamic grasping process to achieve fluent grasping of a target object by the robot end-effector. The process starts from the phase of approaching the object over the phases of colliding with the object and letting it roll about the colliding point to the final phase of catching it by the palm or grasping it by the fingers of the end-effector. We derive a unified model for this hybrid dynamic manipulation process embodied as approaching-colliding-rolling-catching/grasping from the spatial vector based articulated body dynamics. Then, the whole process is formulated as a free-terminal constrained multi-phase optimal control problem (OCP). We extend the traditional differential dynamic programming (DDP) to solving this free-terminal OCP, where the backward pass of DDP involves constrained quadratic programming (QP) problems and we solve them by the primal-dual Augmented Lagrangian (PDAL) method. Simulations and real experiments are conducted to show the effectiveness of the proposed method for robotic dynamic grasping.**

I. Introduction

Dynamic manipulation is a quite challenging form of robot manipulation, which can thoroughly demonstrate the body dexterity of robots, the robustness of control algorithms, and the ultimate capability of robots [1], [2], [3]. The effective usage of friction, centrifugal force or inertial force is of great importance for successful dynamic manipulation [4], [5], [6]. Conventional manipulation research refers to a single manipulation primitive, such as grasping, putting, sliding, throwing, catching, casting, and pushing [7], [8], [9]. Indeed, a single manipulation primitive can achieve some common tasks, such as object handling, object transportation, pick-and-place, etc. However, combining multiple manipulation primitives can be a better way to tackle certain complex tasks, e.g., door opening, throw-catch, push-grasp [10], [11].

This article mainly studies about how to enable robots to dynamically catch objects on the table without finger closure, which cannot be accomplished by traditional static or quasi-static manipulation [12], [13]. We consider the dynamic manipulation process consisting of four phases:

a) Approaching: the robot is guided to the target object and be ready to grasp;

b) Colliding: it allows the object to obtain a reasonable initial velocity;

c) Rolling: the object is adjusted to an appropriate grasping position and attitude for the robot palm.

d) Catching or Grasping: the object is halted in the robot palm or transferred to the desired position.

The first three phases are common, while the last one uses different actions and defines the so-called strategy I (catching) or II (grasping). Strategy I makes full use of the robot palm, which means in this case it can complete the dynamic grasping of objects without any finger closure. Strategy II is generally used in situations like pouring water or other subsequent dexterous manipulation after dynamic grasping. These hybrid manipulations with fixed sequence can be boiled down to the multi-phase optimal control problem (OCP).

Furthermore, the contributions of this paper include:

- Two hybrid dynamic manipulation strategies are proposed to achieve efficient dynamic grasping;
- The unified modeling achieved by articulated body algorithm (ABA) is utilized to build the hybrid system;
- The free-terminal constrained differential dynamic programming (FT-CDDP) is expanded to generate dynamic feasible trajectory to achieve the task.

This paper is organized as follows: Section II reviews works on dynamic manipulation and hybrid system. The hybrid dynamic manipulation and manipulation primitives are presented in Section III, Section IV respectively. DDP based multi-phase optimization is shown in the Section V. The simulations and experiments for the task are shown in the Section VI. Section VII concludes the paper.

II. Related Work

A. Dynamic Manipulation

The key factor in researching the dynamic manipulation is the dynamic characteristic of objects. For dynamic prehensile manipulation, [14] and [15] designed an object-level grasp controller considering the dynamic effects of fast motion. A dynamic system approach for softly catching a flying object with arm-hand collaborative control was proposed in [16]. Besides, typical dynamic nonprehensile manipulation family including throwing, catching, batting, sliding, rolling and so on [17], [1]. Some nonprehensile experimental studies focused on special types of robots, such as: butterfly robot [18], plannar robot [10], mobile robot [19], manipulator [15].

B. Optimal Control for Hybrid System

The study of hybrid dynamic systems exists in many research fields, namely locomotion, manipulation, and so on [20], [21]. The hybrid system modeling and optimization

The authors are with Tencent Robotics X, Shenzhen, Guangdong, China. {chowchzhou, yanbolong, timothyshi,longfeizhao, petezheng} @tencent.com

Yanbo Long is also with University of Bristol, Bristol, UK. Lei Shi is also with Johns Hopkins University, Baltimore, MD, US. The work is done when they were interns at Tencent Robotics X.

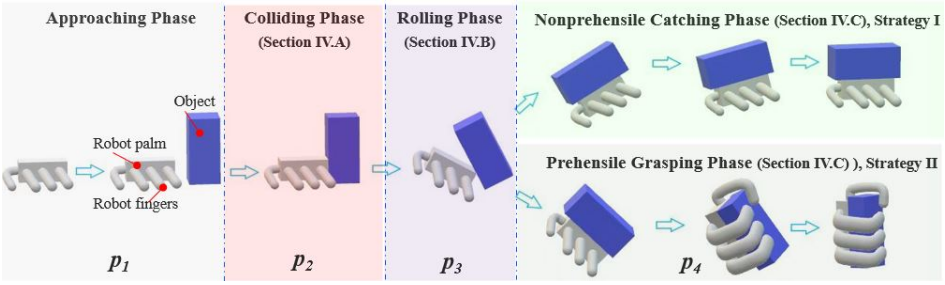


Fig. 1: Hybrid manipulation strategy. p_1, p_2, p_3, p_4 denote the phases of approaching, colliding, rolling, catching(strategy I)/grasping(strategy II) respectively.

is made for quadrupted bounding gait by whole-body motion planning [22]. The quasi-static plannar hybrid pushing and pivoting manipulation with frictional contact switches is studied in [23]. The piecewise-smooth hybrid dynamic system optimization with unknown sequence and timing for quadcopter perching is also studied in [22]. [10] studied the planning and control for nonprehensile picking and throwing of a block. To some extent, the solution of the hybrid system eventually resorts to the optimal control algorithm, such as direct and indirect method, shooting and collocation method [24]. Owing to advantages of the smaller dimension matrix calculation of subproblems and fast convergence characteristics, the differential dynamic programming [25], [26], [27], [28] based algorithms have become prevailing. The constrained nonlinear programming in the DDP based algorithms can be dealt with penalty, sequential quadratic programming(SQP), interior points, or augmented lagrangian(AL) [29], [30]. Other properties, like multi-shooting [31], free-terminal [32], and parallel calculation [33] have also been investigated successively.

All the studies reviewed above proved the fact that the optimal control theory is sufficient enough to solve the hybrid system problem in dynamic manipulation. However, the object grasping without slowing down robot has not been studied yet, which can be implemented by the hybrid manipulation strategy, with the modelling and optimization of the hybrid system. Besides, DDP based method combined with multi-constraint, free-terminal, and multi-phase has not been comprehensively studied, let alone applied to such complex and highly dynamic tasks.

III. Hybrid Manipulation Strategy

A. Hybrid Manipulation Strategies for Dynamic Grasping

We conclude hybrid dynamic manipulation (shown in Fig.1) as approaching-colliding-rolling-catching(strategy I) and approaching-colliding-rolling-grasping(strategy II). Under both strategies, the object can be grasped successfully.

Strategy I belongs to nonprehensile hybrid manipulation purely. The robot equipped with a palm shaped end-effector needs to quickly catch the object. So the dynamic catching of objects is achieved by the palm edge and surface, which greatly expands the robot dexterous manipulation ability.

Strategy II belongs to nonprehensile & prehensile hybrid manipulation. When the object is upside down on the palm, such as pouring water, it is necessary to close the fingers to enable the object to move arbitrarily with the robot. So hybrid manipulation is no longer limited to nonprehensile primitives. Blending nonprehensile primitives with prehensile primitives can fully utilize the advantages of both.

B. Two-Body Hybrid System

The two-body system developed from articulated body algorithm(ABA) [34] is described in Fig.2(a). The pose of hybrid system is defined as $\boldsymbol{x}_{ro} = [\boldsymbol{x}_r \ \boldsymbol{x}_o], \boldsymbol{x}_r = [\boldsymbol{r}_{B_r}^T \ \boldsymbol{\Omega}_{B_r}^T], \boldsymbol{x}_o = [\boldsymbol{r}_{B_o}^T \ \boldsymbol{\Omega}_{B_o}^T]$, where $\boldsymbol{x}_r, \boldsymbol{x}_o \in \mathbb{R}^6, \boldsymbol{x}_{ro} \in \mathbb{R}^{12}$. Furthermore, we denote $\boldsymbol{x} = [\boldsymbol{x}_{ro} \ \dot{\boldsymbol{x}}_{ro}]$ and $\boldsymbol{u} = [\hat{\boldsymbol{f}}_r \ \hat{\boldsymbol{f}}_o]$ the general state and input of the hybrid system [35]. And the vector shaped like $'\hat{\imath}'$ means that $'\iota'$ is a spatial vector. The time frames of N_s phases are $\boldsymbol{t}_m = [t_m^{[1]}, \cdots, t_m^{[N_s]}]$, the subscript $m =' 0'$ or $'f'$ denote the start or terminal time. So the unified state equation can be derived as

$$\begin{cases} \boldsymbol{x}_{n+1} = \boldsymbol{f}(\boldsymbol{x}_n, \boldsymbol{u}_n, \Delta t), \quad n = 0, 1, 2, \cdots, N-1 \\ \boldsymbol{h}_n(\boldsymbol{x}_n, \boldsymbol{u}_n) \leq 0, \quad n = 0, 1, 2, \cdots, N \\ \boldsymbol{g}_n(\boldsymbol{x}_n, \boldsymbol{u}_n) = 0, \quad n = 0, 1, 2, \cdots, N \end{cases} \tag{1}$$

where $N = \sum_{i=1}^{N_s} N^{[i]}$. $N^{[i]}$ is the discretized value of $i-$th phase. $\boldsymbol{x}_{n+1} = \boldsymbol{f}(\boldsymbol{x}_n, \boldsymbol{u}_n, \Delta t)$ is the discrete form of state space equation, and Δt is the time step. $\boldsymbol{h}_n(\boldsymbol{x}_n, \boldsymbol{u}_n)$ denotes the inequality constraint, namely friction cone constraint at the sticking contact points. The motion trajectory $\boldsymbol{\zeta}_\kappa(t)$ of hybrid system can be described by

$$\begin{cases} \boldsymbol{\zeta}_\kappa(t) = [\boldsymbol{\zeta}_\kappa^P(t) \ \boldsymbol{\zeta}_\kappa^\Omega(t)]^T \in \mathbb{R}^{18} \\ \boldsymbol{\zeta}_\kappa^P(t) = [\boldsymbol{r}_\kappa^T(t) \ \dot{\boldsymbol{r}}_\kappa^T(t) \ \ddot{\boldsymbol{r}}_\kappa^T(t)]^T \in \mathbb{R}^9 \\ \boldsymbol{\zeta}_\kappa^\Omega(t) = [\boldsymbol{\Omega}_\kappa^T(t) \ \boldsymbol{\omega}_\kappa^T(t) \ \dot{\boldsymbol{\omega}}_\kappa^T(t)]^T \in \mathbb{R}^9 \end{cases} \tag{2}$$

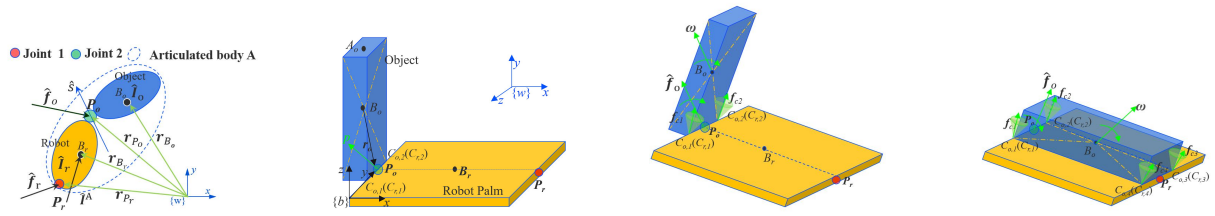
where superscript P and Ω denote position and attitude component. κ is the points B_r, C_r, B_o, or C_o shown in Fig.2. $\boldsymbol{g}_n(\boldsymbol{x}_n, \boldsymbol{u}_n)$ denotes the equality linkage constraint, consisting of motion linkage condition and force linkage condition. The motion linkage condition is the relationship between $\boldsymbol{\zeta}_{C_r}$ and $\boldsymbol{\zeta}_{C_o}$, while the force linkage condition is the relationship between $\hat{\boldsymbol{f}}_r$ and $\hat{\boldsymbol{f}}_o$.

More details of the two-body hybrid system can be found in Appendix.

IV. Manipulation Primitives

A. Colliding Manipulation

The colliding configuration is illustrated in Fig.2(b). Since it is implemented by the controlled manipulator, the mass of robot m_r and the inertia of robot σ_r are assumed as ∞. So the velocity of robot $\dot{\boldsymbol{r}}_{C_r}$ will not mutate in a very short time. We denote $\boldsymbol{\rho}$ the real exchanged momentum considering the



(a) Two-body system. (b) Colliding model (c) Rolling model. (d) Nonprehensile catching.

Fig. 2: Hybrid system. B_j is the centroid of body j, and $j = r$ or o denotes robot palm or object. $\hat{\boldsymbol{I}}^A$ is the spatial inertia of articulated robot-object (simplified as articulated body A). m_j, σ_j are the mass and inertia of body j. $\hat{\boldsymbol{I}}_j \in \mathbb{R}^6$ and $\hat{\boldsymbol{f}}_j \in \mathbb{R}^6$ are the spatial inertia and the spatial force vector of body j, respectively. $\boldsymbol{r}_{B_j}$ is the position vector of point B_j. $\hat{\boldsymbol{s}}$ is joint axis. In the Fig.2(b), (c), (d), $C_{r,i}$ and $C_{o,i}$, $i = 1, 2, 3, 4$ are possible contact point pairs. P_j is the equivalent contact point mounted by the virtual joint, and it is also the action point of $\hat{\boldsymbol{f}}_j$. $\boldsymbol{r}_o$ is vector from B_o to P_o. $\hat{\boldsymbol{f}}_{C,i} \in \mathbb{R}^3$ are the contact force acted at point $C_{o,i}$. $\{w\}$ is inertial frame. $\{b\}$ is body frame.

energy loss, as the general collision consists of compression phase and restitution phase [36], [37], [38] . The changes of contact velocity $\boldsymbol{v} = \dot{\boldsymbol{r}}_{P_o} - \dot{\boldsymbol{r}}_{Pr}$ can be defined as

$$\Delta \boldsymbol{v} = \Delta \dot{\boldsymbol{r}}_{P_o} - \Delta \dot{\boldsymbol{r}}_{P_r} = W(\boldsymbol{m_o}, \boldsymbol{\sigma}_o)\boldsymbol{\rho} = W\boldsymbol{\rho} \tag{3}$$

where W is referred to as the *inverse inertia matrix* [39]. So we have the object velocity at the post-colliding stage

$$\hat{\boldsymbol{v}}_o^+ = \hat{\boldsymbol{v}}_o^- + \Delta \hat{\boldsymbol{v}}_o = \hat{\boldsymbol{v}}_o^- + \boldsymbol{M}\boldsymbol{\rho} \tag{4}$$

where $\boldsymbol{M}$=$[1/\boldsymbol{m}_o; \boldsymbol{r}_o \times /\boldsymbol{\sigma}_o]$ is the dynamic parameter related matrix. The superscript $'+'$ and $'-'$ denote the post-colliding and pre-colliding, respectively. We denote $\hat{\boldsymbol{a}}_{r,\max}$ as the maximum acceleration of the robot, then $\hat{\boldsymbol{v}}_r^+ - \hat{\boldsymbol{v}}_r^- = \hat{\boldsymbol{a}}_{r,\max}\delta t$. If $\hat{\boldsymbol{v}}_o^+ < \hat{\boldsymbol{v}}_r^-$, this two bodies will be keep sticking. If $\hat{\boldsymbol{v}}_r^- < \hat{\boldsymbol{v}}_o^+ < \hat{\boldsymbol{v}}_r^+$, the robot needs to accelerate to keep sticking . If $\hat{\boldsymbol{v}}_o^+ > \hat{\boldsymbol{v}}_r^+$, this two bodies will be separated. The choice of collision point can be roughly divided into A_o, B_o, C_o. The palm edge collides with the point C_o so that the object can be separated from the table in time, reducing the uncertainty of friction. The proper impact point selection makes dynamic grasping task more deterministic.

B. Rolling Manipulation

The rolling configuration is illustrated in Fig.2(c). N_c denotes the number of contact points. In the rolling manipulation, $N_c = 2$, $C_r = C_{r,1} \cup C_{r,2}$. This 2-point contact cannot fully constrain the object, allowing the object to rotate. So we can describe the equality constraints of motion linage and force linage condition as:

$$\boldsymbol{\zeta}_{C_{r,i}}^P = \boldsymbol{R}_{B_r}^{C_{r,i}} \boldsymbol{\zeta}_{B_r}^P = \boldsymbol{\zeta}_{C_{o,i}}^P = \boldsymbol{R}_{B_o}^{C_{o,i}} \boldsymbol{\zeta}_{B_o}^P, i = 1, \cdots, N_c \tag{5}$$

$$(\acute{\boldsymbol{E}} - \hat{\boldsymbol{I}}_r \hat{\boldsymbol{I}}^{A,-1})\hat{\boldsymbol{f}}_r - \hat{\boldsymbol{f}}_o + \hat{\boldsymbol{I}}_r \hat{\boldsymbol{I}}^{A,-1} \hat{\boldsymbol{p}}^A + \hat{\boldsymbol{p}}_r = 0 \tag{6}$$

where $\boldsymbol{R}_{B_j}^{C_{j,i}}$ is the transformation matrix from $\boldsymbol{\zeta}_{B_j}^P$ to $\boldsymbol{\zeta}_{C_{j,i}}^P$, $j = r$ or o. And $\acute{\boldsymbol{E}}$ is the identity matrix. In general, this equation can be formulated as the equality constraint of $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$. Furthermore, for the manipulated object, the body wrench $\boldsymbol{w} \in \mathbb{R}^6$, net force vector $\hat{\boldsymbol{f}}_o \in \mathbb{R}^6$ and the contact force $\boldsymbol{f}_C \in \mathbb{R}^{N_c \times 3}$ have the following relationship:

$$\boldsymbol{w} = \boldsymbol{T}\hat{\boldsymbol{f}}_o = \boldsymbol{G}_r \boldsymbol{f}_C, \quad j = 1, \cdots, N_c \tag{7}$$

where $\boldsymbol{T}$ is the spatial transformation matrix. $\boldsymbol{G}_r$ is the grasp transformation matrix. Besides, the sticking-rolling can be guaranteed by the friction cone constraint $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$ as follows

$$\boldsymbol{N}_i^T \boldsymbol{f}_{C,i} \leq \boldsymbol{b}_i \tag{8}$$

where $\boldsymbol{N}_i = -[\mu_i \boldsymbol{n}_i - \boldsymbol{o}_i, \mu_i \boldsymbol{n}_i \boldsymbol{o}_i, \mu_i \boldsymbol{n}_i - \boldsymbol{t}_i, \mu_i \boldsymbol{n}_i \boldsymbol{t}_i, \boldsymbol{n}_i, -\boldsymbol{n}_i]$, $\boldsymbol{N}_i \in \mathbb{R}^{3\times 6}$. μ_i is the Coulomb friction coefficient. $\boldsymbol{b}_i^j = [0, 0, 0, 0, -f_{C,i}^L, -f_{C,i}^R]$. $\boldsymbol{n}_i$ is unit normal vector, and $\boldsymbol{o}_i, \boldsymbol{t}_i \in \mathbb{R}^3$ are two orthonormal tangent vectors described with respect to the object frame. $f_{C,i}^L$, $f_{C,i}^R$ are the nonnegative lower and upper bounds on the normal contact force, respectively. The dynamic underactuated rolling manipulation is under unilateral constraints, which will make the energy of the object keep increasing. So the boundary conditions of the object need to meet this requirement.

C. Nonprehensile Catching or Prehensile Grasping

The constraints in this phase are almost the same as the previous rolling manipulation, except for the number of contact points $N_c = 4$, $C_r = C_{r,1} \cup C_{r,2} \cup C_{r,3} \cup C_{r,4}$. This 4-point surface contact fully constrains the object, allowing the object to rest on the palm. So the nonprehensile catching and prehensile grasping share the same equality constraint $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$ of motion linage and force linage. However, the main differences between these two strategies are the friction-constrained or friction-free.

Prehensile Grasping: The grasping of objects requires the robot fingers to close and form an envelope relationship with the palm. After the fingers are closed, the two bodies have the same linear velocity and angular velocity, where the velocity of the object needs to be eliminated. After that, the object is transferred to a specific position and attitude.

Nonprehensile Catching: The catching configuration is illustrated in Fig.2(d). Different from the prehensile manipulation, the non-prehensile manipulation requires the utilization of dynamic balance during the movement of the object transportation, so that the object is in a sticking contact on the palm [15], [6]. The contact force between the palm and the object must always be within the friction cone [40], and the inequality constraints $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$ should be considered.

V. DDP Based Multi-phase Optimal Control

With unified modeling, the hybrid dynamic system is described by the following formula

$$\begin{cases} \min J = \sum_{i=1}^{N_s} \sum_{k=0}^{N^{[i]}-1} l^{[i]}(\boldsymbol{x}_k^{[i]}, \boldsymbol{u}_k^{[i]}) + l_f^{[i]}(\boldsymbol{x}_N^{[i]}, \boldsymbol{t}_N^{[i]}) \\ \text{s.t}: \ \boldsymbol{x}_{k+1}^{[i]} = f(\boldsymbol{x}_k^{[i]}, \boldsymbol{u}_k^{[i]}), \quad \boldsymbol{u}_{min}^{[i]} \le \boldsymbol{u}^{[i]} \le \boldsymbol{u}_{max}^{[i]} \\ \quad \boldsymbol{g}^{[i]}(\boldsymbol{x}_k^{[i]}, \boldsymbol{u}_k^{[i]}) = 0 \\ \quad \boldsymbol{h}^{[i]}(\boldsymbol{x}_k^{[i]}, \boldsymbol{u}_k^{[i]}) \le 0 \\ \quad t_N^{[i]} \text{ is free}, t_{min}^{[i]} \le t^{[i]} \le t_{max}^{[i]} \quad i = 1, 2, \ldots, N_i. \end{cases} \tag{9}$$

where J is the total cost function. $\boldsymbol{t}_N^{[i]} = \boldsymbol{t}_f^{[i]} - \boldsymbol{t}_0^{[i]}$ is the terminal time of i-th phase. $\boldsymbol{t}_N = [t_N^{[1]}, \cdots, t_N^{[N_s]}]$. And $l^{[i]} = \widetilde{\boldsymbol{x}}_k^{[i],T} \boldsymbol{Q} \widetilde{\boldsymbol{x}}_k^{[i]} + \boldsymbol{u}_k^{[i],T} \boldsymbol{R} \boldsymbol{u}_k^{[i]}$, $l_f^{[i]}(\boldsymbol{x}_N^{[i]}) = \widetilde{\boldsymbol{x}}_N^{[i],T} \boldsymbol{Q}_f \widetilde{\boldsymbol{x}}_N^{[i]}$ are the trajectory cost and terminal cost respectively. And $\widetilde{\boldsymbol{x}}_k^{[i]} = \boldsymbol{x}_k^{[i]} - \boldsymbol{x}_g^{[i]}$ denotes state errors, $\boldsymbol{x}_g^{[i]}(t_m^{[i]})$ is the desired value. $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{Q}_f$ are the weight matrices. $\boldsymbol{u}_k$ is derived backward sequentially in DDP based methods. As the multi-constraints are solved by the primal-dual Augmented Lagrangian (PDAL) approach [29], [30], [41], [42], [43], we can rewrite the Bellman recursion equation as

$$\begin{aligned} \widehat{V}(\boldsymbol{x}_n) &= \min_{\boldsymbol{u}_n, \lambda_E, \lambda_I} \widehat{Q}(\boldsymbol{x}_n, \boldsymbol{u}_n, \lambda_E, \lambda_I) \\ &= \min_{\boldsymbol{u}_n, \lambda_E, \lambda_I} (\widehat{l}(\boldsymbol{x}_n, \boldsymbol{u}_n, \lambda_E, \lambda_I) + \widehat{V}(\boldsymbol{x}_{n+1})) \end{aligned} \tag{10}$$

where $n = 1, 2, \ldots, N$. λ_E and λ_I are the Lagrangian multipliers. $V(\boldsymbol{x}_n)$ and $Q(\boldsymbol{x}_n, \boldsymbol{u}_n)$ denote value function and action-state value function. The subscripts E and I denote the equality and inequality part respectively. $\widehat{Q}$ and $\widehat{V}$ are the modified value. What's more, we denote $l_{g_n}(\lambda_E)$ and $l_{h_n}(\lambda_I)$ as the trajectory cost related with constraints. So, $\widehat{l}(\boldsymbol{x}_n, \boldsymbol{u}_n, \lambda_E, \lambda_I)$ will be shown as

$$\widehat{l}(\boldsymbol{x}_n, \boldsymbol{u}_n, \lambda_E, \lambda_I) = l_n + \mu_E l_{g_n}(\lambda_E)/2 + \mu_I l_{h_n}(\lambda_I)/2 \tag{11}$$

$$l_{g_n}(\lambda_E) = \left\| \boldsymbol{g}_n/\mu_E + \lambda_E^{'} \right\|^2 + \left\| \boldsymbol{g}_n/\mu_E + \lambda_E^{'} - \lambda_E \right\|^2 \tag{12}$$

$$l_{h_n}(\lambda_I) = \left\| [\boldsymbol{h}_n/\mu_I + \lambda_l^{'}]_+ \right\|^2 + \left\| \boldsymbol{h}_n/\mu_I + \lambda_l^{'} - \lambda_l]_+ \right\|^2 \tag{13}$$

where μ_E, μ_I are the positive scalar. λ_E and λ_I denote the Lagrange multipliers. The superscript $'$ denotes the updated value. $[.]_+$ denotes the positive orthogonal projection operators. $[.]_A$ denotes the active set inequality constraints. The optimal descent direction for $\widehat{Q}$ can be obtained with

$$K \begin{bmatrix} \delta u \\ \delta x \\ \delta \lambda \\ [\delta \nu]_A \end{bmatrix} = \begin{bmatrix} \widehat{Q}_u + \widehat{Q}_{ux} \delta x \\ \widehat{Q}_x + \widehat{Q}_{xx} \delta x \\ f + f_x \delta x + \mu_E(\lambda_l - \lambda) \\ [h + h_x \delta x + \mu_I(\lambda_l - \lambda)]_A \end{bmatrix} \tag{14}$$

and

$$K = \begin{bmatrix} \widehat{Q}_{uu} & \widehat{Q}_{ux} & f_u^T & [h_u^T]_A \\ \widehat{Q}_{xu} & \widehat{Q}_{xx} & f_x^T & [h_x^T]_A \\ f_u & f_x^T & -\mu_E \acute{E} & 0 \\ [h_u]_A & [h_x]_A & 0 & -\mu_I \acute{E} \end{bmatrix} \tag{15}$$

where $\xi_{\varrho,\zeta}$ are the Hessian matrix of ξ with respective to ϱ and ζ. ξ_ϱ is the Jacobian matrix of ξ with respective to ϱ. K is the KKT matrix. Then the optimal incremental value, δu, δx, $\delta \lambda$, $\delta \mu$ can be accessed. And the second order Taylor expansion is utilized to estimate the value function [32].

$$\begin{aligned} \Delta \widehat{V}(\boldsymbol{x}_N, \boldsymbol{t}_N) \approx \widehat{V}_x \delta \boldsymbol{x} + \widehat{V}_t \delta \boldsymbol{t} + \frac{1}{2} \delta \boldsymbol{x}^T \widehat{V}_{xx} \delta \boldsymbol{x} \\ + \frac{1}{2} \delta \boldsymbol{x}^T \widehat{V}_{xt} \delta \boldsymbol{t} + \frac{1}{2} \delta \boldsymbol{t}^T \widehat{V}_{tx} \delta \boldsymbol{x} + \frac{1}{2} \delta \boldsymbol{t}^T \widehat{V}_{tt} \delta \boldsymbol{t} \end{aligned} \tag{16}$$

The terminal optimal time δt_N^* can be derived by the first-order optimality condition

$$\delta t_N^* = -\widehat{V}_{t_N t_N}^{-1} (\widehat{V}_{t_N} + \widehat{V}_{t_N x_N} \delta x_N) \tag{17}$$

DDP based optimal framework consists of two main parts: forward and backward passes. The backward pass is formulated as an constrained QP issue solved by PDAL method. The modified constrained DDP is shown in the following Algorithm 1.

Algorithm 1 Free Terminal CDDP

Given

$\boldsymbol{x}_0$, $\boldsymbol{x}_g$, $\boldsymbol{x}_{n+1} = \boldsymbol{f}(\boldsymbol{x}_n, \boldsymbol{u}_n)$, $\boldsymbol{g}(\boldsymbol{x}_n, \boldsymbol{u}_n)$, $\boldsymbol{h}(\boldsymbol{x}_n, \boldsymbol{u}_n)$

Initialization

$\boldsymbol{U} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots, \boldsymbol{u}_{N-1}\}$, $\boldsymbol{X} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$

$n_n = 0$, // iterations counter

while $(|J_n - J_{n-1}| \ge \epsilon)$ **do**

$\widehat{V}_{x_N}, \widehat{V}_{x_N x_N}, \widehat{V}_{t_N}, \widehat{V}_{t_N x_N}, \widehat{V}_{t_N t_N}$

for $k = N-1$ to 0 **do**

Calculation of Jacobian and Hessian matrix

Then the optimal incremental value, δu, δx, $\delta \lambda$, $\delta \mu$

end for

for $n = 0$ to $N-1$ **do**

$\boldsymbol{u'}_n = \boldsymbol{u}_n + \delta \boldsymbol{u}_n$

$\boldsymbol{x}_{n+1} = f(\boldsymbol{x}_n, \boldsymbol{u'}_n)$

end for

$\delta t_N \leftarrow Eq.17$

$\boldsymbol{t}_N = \min(\max(\boldsymbol{t}_N + \delta t, \boldsymbol{t}_{min}), \boldsymbol{t}_{max})$

$N^{'} = N + \delta t_N / \Delta t$

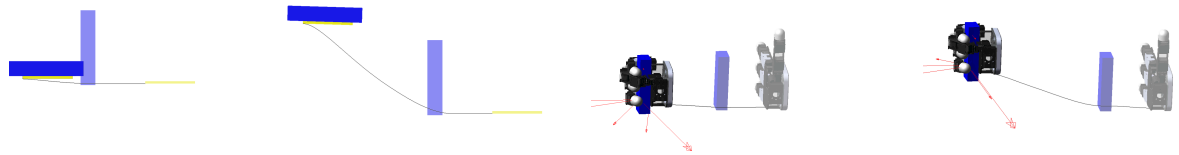
$J \leftarrow J^{'}$, $\boldsymbol{X} \leftarrow \boldsymbol{X}^{'}$, $\boldsymbol{U} \leftarrow \boldsymbol{U}^{'}$, $n_n \leftarrow n_n + 1$, $N \leftarrow N^{'}$.

end while=0

where ϵ is the given small number. The superscript $'$ denote the updated value. $\boldsymbol{X}$ is the state sequence. $\boldsymbol{U}$ is the control input sequence.

VI. Simulations and Experiments

The simulations are performed by co-simulation in $Recurdyn$ and $Matlab$. The experiments are relied on dual UR16e robots. All the results are also shown in the accompanying video. The simulations and experiments are conducted in the plane gravitational field. Now, we define $\boldsymbol{x}_r = [x_r \ y_r \ \gamma_r]$, $\boldsymbol{u}_r = [f_{x_r} \ f_{y_r} \ \tau_{\gamma_r}]$, $\boldsymbol{x}_o = [x_o \ y_o \ \gamma_o]$, $\boldsymbol{x}_{ro} = [\boldsymbol{x}_r \ \boldsymbol{x}_o]$, and $\boldsymbol{u}_o = [f_{x_o} \ f_{y_o} \ \tau_{\gamma_o}]$. Since the object x_o, y_o in this task is either fixed or constrained by motion linage, only the simplified general coordinate $\overline{\boldsymbol{x}} = [x_r \ y_r \ \gamma_r \ \gamma_o] \in \mathbb{R}^4$ is listed for simplicity as below.



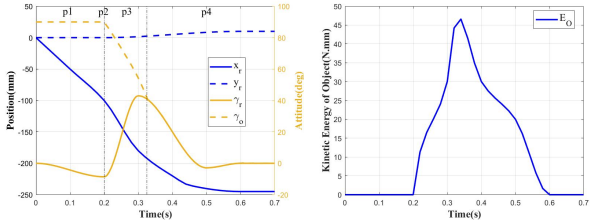
(a) Low velocity by strategy I. (b) High velocity by strategy I. (c) Low velocity by strategy II. (d) High velocity by strategy II.

Fig. 3: Four simulation cases. The blue and yellow rectangle represent the object and the robotic palm or hand, respectively.

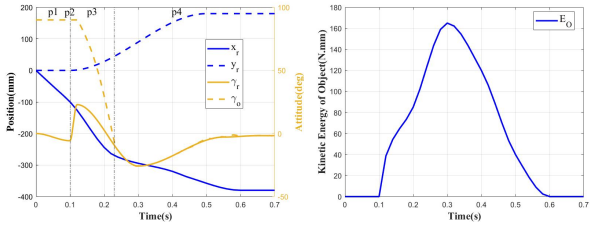
TABLE I: Desired boundary conditions at different stages in different tasks.

	Simulation I(low)	Simulation I(high)	Simulation II(low)	Simulation II(high)	Experiment I	Experiment II
$\boldsymbol{s}_N^{[1]}$	$[-0.1, 0, -0.15, 1.57]$	$[-0.1, 0, -0.15, 1.57]$	$[-0.1, 0, 0.785, 1.57]$	$[-0.1, 0, 0.87, 1.57]$	$[-0.1, 0, -0.35, 1.57]$	$[-0.1, 0, 1.13, 1.57]$
$\boldsymbol{s}_N^{[2]}$	•	•	•	•	•	•
$\boldsymbol{s}_N^{[3]}$	$[\diamond, \diamond, 0.7, 0.7]$	$[\diamond, \diamond, -0.17, -0.17]$	$[\diamond, \diamond, 1.05, 1.05]$	$[\diamond, \diamond, 0.28, 0.28]$	$[\diamond, \diamond, -0.35, -0.35]$	$[\diamond, \diamond, 0.7, 0.7]$
$\boldsymbol{s}_N^{[4]}$	$[\diamond, \diamond, 0, 0]$	$[\diamond, \diamond, 0, 0]$	$[\diamond, \diamond, 1.57, 1.57]$	$[\diamond, \diamond, 1.57, 1.57]$	$[\diamond, \diamond, 0, 0]$	$[\diamond, \diamond, 1.57, 1.57]$
$\dot{\boldsymbol{s}}_N^{[1]}$	$[-0.5, 0, 0, 0]$	$[-1, 0, 0, 0]$	$[-0.5, 0, 0, 0]$	$[-1, 0, 0, 0]$	$[-0.6, 0, 0, 0]$	$[-0.8, 0, 0, 0]$
$\dot{\boldsymbol{s}}_N^{[2]}$	$[-0.5, 0, 0, -3]$	$[-1, 0, 0, -5]$	$[-0.5, 0, 0, -3]$	$[-1, 0, 0, -5]$	$[-0.6, 0, 0, -3.6]$	$[-0.8, 0, 0, -4.8]$
$\dot{\boldsymbol{s}}_N^{[3]}$	$[-0.8, 0.1, -5.5, -5.5]$	$[-1.4, 0.4, -8, -8]$	$[-0.8, -0.05, -5, -5]$	$[-1.4, 0.2, -7, -7]$	$[-0.8, 0.22, -5.6, -5.6]$	$[-1.3, 0.2, -6.8, -6.8]$
$\dot{\boldsymbol{s}}_N^{[4]}$	$[0, 0, 0, 0]$	$[0, 0, 0, 0]$	$[0, 0, 0, 0]$	$[0, 0, 0, 0]$	$[0, 0, 0, 0]$	$[0, 0, 0, 0]$

• denotes no change from the previous stage. $\diamond$ denotes no constraint, that is free terminal

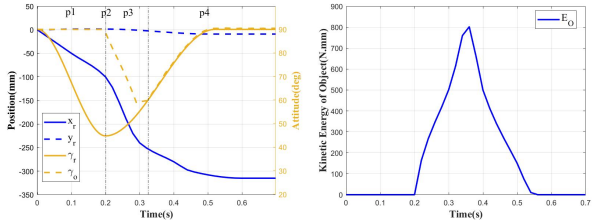


(a) Pose curves in case a. (b) Energy curve in case a.

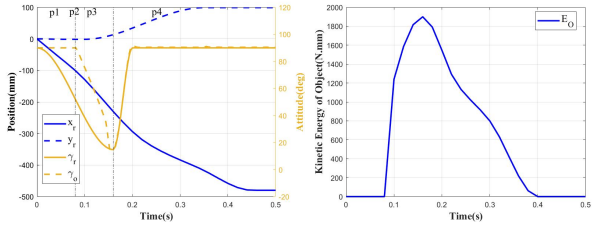


(c) Pose curves in case b. (d) Energy curve in case b.

Fig. 4: The simulation results of strategy I.



(a) Pose curves in case a. (b) Energy curve in case a.



(c) Pose curves in case b. (d) Energy curve in case b.

Fig. 5: The simulation results of strategy II.

In the four simulations(shown in Fig.3), the weight of cuboid is 0.3kg and the size of the cuboid is $12\text{cm} \times 2.5\text{cm} \times 5\text{cm}$. The real friction coefficient between the palm and cuboid is 0.55. In simulation, case a and b correspond to the object with low and high collision velocity, respectively. Curves for desired pose and energy are shown as Fig.4 and Fig.5, respectively. Due to the short time cost during collision (denoted as $(\boldsymbol{t}_f^{[2]} - \boldsymbol{t}_f^{[1]}) \to 0$), the pose of the object does not change, but the object acquires an instantaneous velocity.

A. Simulations of Strategy I at Different Speeds

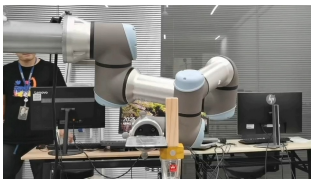
The robot end-effector uses a flat plate, which means the object is grasped dynamically with no finger closed. This section presents the simulation results of dynamic nonprehensile catching for both low-speed and high-speed collision cases (differentiated by the contact velocity $\boldsymbol{v}$). The corresponding switching time vector is derived as $\boldsymbol{t}_f = [0.2, 0.2, 0.32, 0.6]\text{s}$ and $\boldsymbol{t}_f = [0.1, 0.1, 0.22, 0.57]\text{s}$ respectively. Constraints in the rolling and catching phases ensure stable contact between the palm and the object. According to the simulation results, the objects can be successfully grasped under different colliding velocities.

B. Simulations of Strategy II at Different Speeds

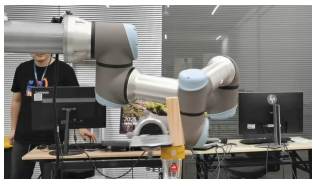
Allegro Hand is used in this part to implement prehensile and nonprehensile hybrid manipulation. This section presents the simulation results of dynamic grasping for both low-speed and high-speed collision cases. The corresponding switching time vector is derived as $\boldsymbol{t}_f = [0.2, 0.2, 0.33, 0.55]\text{s}$ and $\boldsymbol{t}_f = [0.08, 0.08, 0.16, 0.39]\text{s}$ respectively. According to simulation results, the objects can be successfully grasped under different collision speeds, and the entire robotic arm does not stop.

C. Robot Experiment of Strategy I

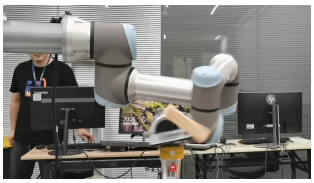
In this experiment, the grasped object is a cuboid. The weight of cuboid is 0.3kg and the size of the cuboid is



(a) Approaching phase in strategy I.

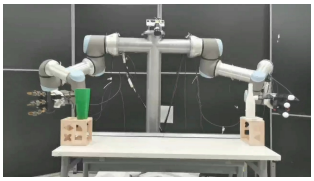


(b) Colliding phase in strategy I.

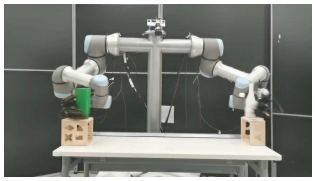


(c) Rolling phase in strategy I.

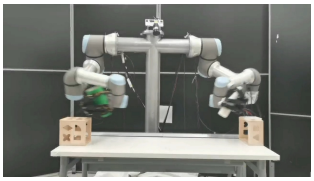
(d) Catching phase in strategy I.



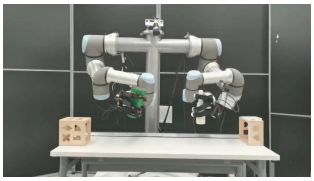
(e) Approaching phase in strategy II.



(f) Colliding phase in strategy II.



(g) Rolling phase in strategy II.



(h) Catching phase in strategy II.

Fig. 6: The experiment setup of strategy I (a-d) and strategy II (e-h) .

$12\text{cm} \times 2.5\text{cm} \times 5\text{cm}$. The centroid of the objects is set at their geometric center. The real friction coefficient between the palm and cuboid is 0.55. The corresponding switching time vector $\boldsymbol{t}_f = [0.15, 0.15, 0.3, 0.5]\text{s}$. The robot end-effector uses a flat plate, which does not decelerate when grabbing the target object. By effectively mixing the three non-prehensile dynamic manipulation primitives, objects are finally dynamically grasped by the palm without finger closure. The snapshots are shown as Fig.6. The success rate of the experiment is 90% (9/10).

D. Robot Experiment of Strategy II

The dual UR16es and Allegro Hands are used in this part to implement prehensile and nonprehensile hybrid manipulation for dynamic grasping of large shaker and bottle. The weight of these two bodies are 0.5kg and 0.3kg respectively. The real friction coefficient between the Allegro Hand and these two bodies are 0.45 and 0.5. The centroid of the objects is assumed almost at their geometric center. The robot trajectory can be calculated, and corresponding switching time vector $\boldsymbol{t}_f = [0.2, 0.2, 0.35, 0.52]\text{s}$. The object can be grasped by the fingers, and then tasks such as pouring water can be realized. The snapshots are shown as Fig.6. The success rate of the experiment is 100% (10/10).

VII. Conclusions

The dynamic hybrid manipulation strategy with colliding-rolling-catching and colliding-rolling-grasping for grasping the target object during the rapid movement of a robot is firstly proposed in this article. The hybrid manipulation is regarded as robot-object hybrid system which is unifiedly modelled as the spatial vector based articulated body dynamics. So the diverse states of hybrid system can be described by the motion linage and force linage. Besides, the modified free-terminal CDDP method is utilized to solve this fixed sequence multi-phase optimal control issue for hybrid dynamic manipulation. The simulation and experiment results show that the model, strategy and algorithm proposed in this paper can effectively achieve the dynamic grasping of objects.

Appendix

Assuming two bodies (shown in Fig.2(a)) are in contact, the spatial vector based dynamics can be derived as

$$\hat{\boldsymbol{f}}_r = \hat{\boldsymbol{I}}^A \hat{\boldsymbol{a}}_r + \hat{\boldsymbol{p}}^A = \hat{\boldsymbol{I}}_r \hat{\boldsymbol{a}}_r + \hat{\boldsymbol{p}}_r + \hat{\boldsymbol{f}}_o \tag{18}$$

$$\hat{\boldsymbol{f}}_o = \hat{\boldsymbol{I}}_o \hat{\boldsymbol{a}}_o + \hat{\boldsymbol{p}}_o \tag{19}$$

$$\hat{\boldsymbol{a}}_o = \hat{\boldsymbol{a}}_r + \Delta\hat{\boldsymbol{a}} \tag{20}$$

where $\hat{\boldsymbol{a}}_j \in \mathbb{R}^6$, $\hat{\boldsymbol{p}}_j \in \mathbb{R}^6$ be the acceleration and the bias force of body j. $\hat{\boldsymbol{p}}^A \in \mathbb{R}^6$ is the bias force of articulated body A. $\Delta\hat{\boldsymbol{a}}$ is the relative acceleration.

According to the contact and motion state of the two-body system, it can be divided into the following three cases:

Case I: body r and body o are separated. $\hat{\boldsymbol{I}}^A = \hat{\boldsymbol{I}}_r$, $\hat{\boldsymbol{p}}^A = \hat{\boldsymbol{p}}_r$. $\hat{\boldsymbol{f}}_r$ and $\hat{\boldsymbol{f}}_o$ are independent of each other, and $\hat{\boldsymbol{f}}_o = \boldsymbol{0}$.

Case II: body r and body o are contact coupling, and with relative motion. $\hat{\boldsymbol{I}}^A = \hat{\boldsymbol{I}}^A(\hat{\boldsymbol{I}}_r, \hat{\boldsymbol{p}}_r, \hat{\boldsymbol{s}}, \hat{\boldsymbol{I}}_o, \hat{\boldsymbol{p}}_o)$, $\hat{\boldsymbol{p}}^A = \hat{\boldsymbol{p}}^A(\hat{\boldsymbol{I}}_r, \hat{\boldsymbol{p}}_r, \hat{\boldsymbol{s}}, \hat{\boldsymbol{I}}_o, \hat{\boldsymbol{p}}_o)$. $\hat{\boldsymbol{f}}_r$ and $\hat{\boldsymbol{f}}_o$ are dependent, and $\hat{\boldsymbol{f}}_o$ is the coupling force vector. $\hat{\boldsymbol{a}}_r$ and $\hat{\boldsymbol{a}}_o$ are also dependent. $\Delta\hat{\boldsymbol{a}} \neq \boldsymbol{0}$.

Case III: body r and body o are contact coupling, and with no relative motion(sticking contact). The kinematics and dynamics of the two bodies in this case are the same as the case II, except that $\Delta\hat{\boldsymbol{a}} = \boldsymbol{0}$.

Assuming $\ddot{\boldsymbol{x}}_{ro} = [\hat{\boldsymbol{a}}_r, \hat{\boldsymbol{a}}_o]$, $\boldsymbol{x} = [\boldsymbol{x}_{ro}, \dot{\boldsymbol{x}}_{ro}]$, $\boldsymbol{u} = [\hat{\boldsymbol{f}}_r, \hat{\boldsymbol{f}}_o]$, we can get the unified differential equation

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{C} \tag{21}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are the system matrix. $\boldsymbol{A} = [\boldsymbol{0}, \boldsymbol{E}; \boldsymbol{0}, \boldsymbol{0}]$, $\boldsymbol{B} = [\boldsymbol{0}, \boldsymbol{0}; \hat{\boldsymbol{I}}^{A,-1}, \boldsymbol{0}; \boldsymbol{0}, \hat{\boldsymbol{I}}_o^{-1}]$, $\boldsymbol{C} = [\boldsymbol{0}; \hat{\boldsymbol{I}}^{A,-1}\hat{\boldsymbol{p}}^A; \hat{\boldsymbol{I}}_o^{-1}\hat{\boldsymbol{p}}_o]$.

In this paper, the approaching phase belongs to **Case I**. **Case II** is adapted to rolling manipulation. And the **Case III** can be applied to prehensile grasping or nonprehenisle catching. Compared with ordinary 3D vectors, 6D vectors based hybrid system model is more concise.

REFERENCES

[1] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1711–1718, 2018.

[2] K. M. Lynch, N. Shiroma, H. Arai, and K. Tanie, "The roles of shape and motion in dynamic manipulation: The butterfly example," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 3. IEEE, 1998, pp. 1958–1963.

[3] M. T. Mason and K. M. Lynch, "Dynamic manipulation," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 1. IEEE, 1993, pp. 152–159.

[4] R. R. Ma and A. M. Dollar, "On dexterity and dexterous manipulation," in *2011 15th International Conference on Advanced Robotics (ICAR)*. IEEE, 2011, pp. 1–7.

[5] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, 2019.

[6] C. Zhou, Y. Long, Y. Cao, L. Zhao, B. Huang, and Y. Zheng, "Optimal nonprehensile interception strategy for objects in flight," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3130–3136.

[7] J. Stüber, C. Zito, and R. Stolkin, "Let's push things forward: a survey on robot pushing," *Frontiers in Robotics and AI*, vol. 7, p. 8, 2020.

[8] D. Serra, F. Ruggiero, A. Donaire, L. R. Buonocore, V. Lippiello, and B. Siciliano, "Control of nonprehensile planar rolling manipulation: A passivity-based approach," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 317–329, 2019.

[9] H. Arisumi, K. Yokoi, and K. Komoriya, "Casting manipulation—midair control of a gripper by impulsive force," *IEEE transactions on robotics*, vol. 24, no. 2, pp. 402–415, 2008.

[10] J. Z. Woodruff and K. M. Lynch, "Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4066–4073.

[11] Z. Pan and K. Hauser, "Decision making in joint push-grasp action space for large-scale object sorting," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6199–6205.

[12] J. Wu, S. Fan, M. Jin, K. Sun, C. Zhou, and H. Liu, "Design and experiment of a universal space-saving end-effector for multi-task operations," *Industrial Robot: An International Journal*, vol. 43, no. 2, pp. 193–203, 2016.

[13] Z. Liu, L. Jiang, B. Yang, C. Li, M. Cheng, S. Fan, and D. Yang, "A model-free synchronous control of humanoid robot finger," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8380–8385.

[14] T. Wimböck, C. Ott, A. Albu-Schäffer, and G. Hirzinger, "Comparison of object-level grasp controllers for dynamic dexterous manipulation," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 3–23, 2012.

[15] C. Zhou, M. Lei, L. Zhao, Z. Wang, and Y. Zheng, "Topp-mpc-based dual-arm dynamic collaborative manipulation for multi-object nonprehensile transportation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 999–1005.

[16] S. S. Mirrazavi Salehian, M. Khoramshahi, and A. Billard, "A dynamical system approach for catching softly a flying object: Theory and experiment," *IEEE Transactions on Robotics*, vol. 32, no. ARTICLE, pp. 462–471, 2016.

[17] Y. Farid, B. Siciliano, and F. Ruggiero, "Review and descriptive investigation of the connection between bipedal locomotion and nonprehensile manipulation," *Annual Reviews in Control*, 2022.

[18] M. Surov, A. Shiriaev, L. Freidovich, S. Gusev, and L. Paramonov, "Case study in non-prehensile manipulation: planning and orbital stabilization of one-directional rollings for the "butterfly" robot," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1484–1489.

[19] F. Bertoncelli, F. Ruggiero, and L. Sabattini, "Linear time-varying mpc for nonprehensile object manipulation with a nonholonomic mobile robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 11 032–11 038.

[20] A. M. Johnson, S. A. Burden, and D. E. Koditschek, "A hybrid systems model for simple manipulation and self-manipulation systems," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1354–1392, 2016.

[21] F. Zhu and P. J. Antsaklis, "Optimal control of hybrid switched systems: A brief survey," *Discrete Event Dynamic Systems*, vol. 25, no. 3, pp. 345–364, 2015.

[22] H. Li and P. M. Wensing, "Hybrid systems differential dynamic programming for whole-body motion planning of legged robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5448–5455, 2020.

[23] N. Doshi, F. R. Hogan, and A. Rodriguez, "Hybrid differential dynamic programming for planar manipulation primitives," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6759–6765.

[24] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[25] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1168–1175.

[26] Y. Pan and E. Theodorou, "Probabilistic differential dynamic programming," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

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

[28] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential dynamic programming for multi-phase rigid contact dynamics," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.

[29] W. Jallet, A. Bambade, N. Mansard, and J. Carpentier, "Constrained differential dynamic programming: A primal-dual augmented lagrangian approach," 2022.

[30] ——, "Proxnlp: a primal-dual augmented lagrangian solver for nonlinear programming in robotics and beyond," in *6th Legged Robots Workshop*, 2022.

[31] E. Pellegrini and R. P. Russell, "A multiple-shooting differential dynamic programming algorithm," in *AAS/AIAA Space Flight Mechanics Meeting*, vol. 2, 2017.

[32] X. Zheng, S. He, and D. Lin, "Constrained trajectory optimization with flexible final time for autonomous vehicles," *IEEE Transactions on Aerospace and Electronic Systems*, 2021.

[33] B. Plancher and S. Kuindersma, "A performance analysis of parallel differential dynamic programming on a gpu," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2018, pp. 656–672.

[34] J. N. Nganga and P. M. Wensing, "Accelerating second-order differential dynamic programming for rigid-body systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7659–7666, 2021.

[35] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE control systems magazine*, vol. 29, no. 2, pp. 28–93, 2009.

[36] Y.-B. Jia, M. Gardner, and X. Mu, "Batting an in-flight object to the target," *The International Journal of Robotics Research*, vol. 38, no. 4, pp. 451–485, 2019.

[37] Y.-B. Jia, "Three-dimensional impact: energy-based modeling of tangential compliance," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 56–83, 2013.

[38] Y.-B. Jia, M. T. Mason, and M. A. Erdmann, "Multiple impacts: A state transition diagram approach," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 84–114, 2013.

[39] M. Gardner, Y.-B. Jia, and H. Lin, "Batting flying objects to the target in 2d," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3225–3232.

[40] C. Zhou, Y. Long, Y. Cao, L. Zhao, B. Huang, and Y. Zheng, "Optimal nonprehensile interception strategy for objects in flight."

[41] P. E. Gill and D. P. Robinson, "A primal-dual augmented lagrangian," *Computational Optimization and Applications*, vol. 51, no. 1, pp. 1–25, 2012.

[42] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "Prox-qp: Yet another quadratic programming solver for robotics and beyond," in *RSS 2022-Robotics: Science and Systems*, 2022.

[43] J.-P. Sleiman, F. Farshidian, and M. Hutter, "Constraint handling in continuous-time ddp-based model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8209–8215.