IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 2, 2021, accepted May 12, 2021, date of publication May 17, 2021, date of current version May 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3081106

Controlling Pivoting Gait Using Graph Model Predictive Control

ANG ZHANG, (Member, IEEE), KEISUKE KOYAMA, (Member, IEEE), WEIWEI WAN, (Senior Member, IEEE), AND KENSUKE HARADA, (Senior Member, IEEE)

Graduate School of Engineering Science, Osaka University, Toyonaka 560-0043, Japan

Corresponding author: Ang Zhang (zhang@hlab.sys.es.osaka-u.ac.jp)

ABSTRACT Pivoting gait, in which a robot iteratively tilts an object, rotates it around a vertex, and then places it down on the floor, is efficient for manipulating a large and heavy object with a relatively small manipulating force. However, pivoting gait can easily fail, even with a small external disturbance, due to its instability. To address this problem, we propose a controller to robustly control the object's motion during pivoting gait by introducing two gait modes, i.e., double-support mode, which can manipulate a relatively light object with higher speed, and quadruple-support mode, which can manipulate a relatively heavy object with slower speed. To control the pivoting gait, a graph model predictive control is applied by considering these two gait modes. The experiments show that by adaptively switching the gait mode according to the applied external disturbance, a robot can stably perform the pivoting gait even when an external disturbance is applied to the object. The experimental results lead us to automate the manipulation of a large and heavy object.

INDEX TERMS Feedback control, graph search, model predictive control, pivoting manipulation, nonprehensile manipulation.

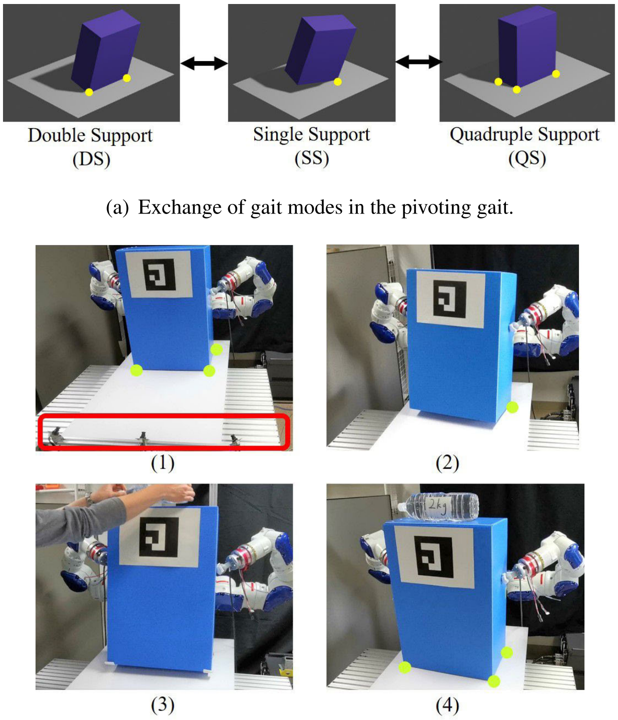
I. INTRODUCTION

Although currently used robots mostly manipulate objects by picking them up once [1]–[3], a pick-and-place manipulation is energy consuming and is not adequate for manipulating a large and heavy object, since the grasped object has to be completely lifted. On the other hand, a human can effectively select an adequate manipulation strategy, taking into account both the features of the task and the physical parameters of the object. For example, when a human moves a large and heavy object such as furniture, a human may tilt the object once, rotate it around an axis through the contact point between the object and the floor, and then place it down on the floor. Such manipulation of a large object is called pivoting manipulation. Since the manipulated object is not completely constrained by the robot, pivoting manipulation is classified as a style of the nonprehensile manipulation [4]. To date, several different styles of nonprehensile manipulation have been proposed, such as rolling [5], pushing [6]–[8], and pivoting [9]–[11].

If pivoting manipulation involves a change in the rotational vertices, it is called a pivoting gait [12] due to the correspondence between the feet of a legged robot and the vertices of a manipulated object. A pivoting gait is effective especially when moving a large and heavy object, since the object's weight is mostly supported by the contact with the floor [13]. Although there has been some research on feedback control of pivoting manipulation [14], [15] , there has been no research about improving the robustness of the pivoting motion by designing the gait mode.

Similar to bipedal walking, the pivoting gait includes single support (SS), in which the object rotates around a supporting vertex, which corresponds to pivoting manipulation, and double support (DS), in which to change the supporting vertex, the object contacts the floor once with a supporting edge that includes two vertices. In addition, we define quadruple support (QS), in which to change its supporting vertex, the object contacts the floor once with a supporting surface that includes more than three vertices. The method of changing the supporting vertices results in different behaviors of the pivoting gait, and we propose two gait modes: the DS and QS modes. If a robot pivots an object in QS mode, a robot can manipulate a relatively heavy object while moving slowly. This is because the face contact is statically stable if the vertical projection of the center of gravity (CoG) is included in the face. On the other hand, if a robot pivots an object in DS mode, the robot can manipulate the object while moving relatively quickly. By

The associate editor coordinating the review of this manuscript and approving it for publication was Yingxiang Liu.



(a) Exchange of gait modes in the pivoting gait.

(b) Cameras are placed in front of the object; see the red rectangle in (1). The robot starts to manipulate the object in DS mode; see (2). A 2 kg bottle is placed on top of the object; see (3). The placement is detected, and the gait mode is switched to QS mode in order to stably manipulate the object; see (4).

FIGURE 1. The designed gait modes and the robot performing the pivoting gait. Once a disturbance is detected, the graph MPC selects a proper gait mode and provides a reference trajectory to the MPC. Then, the MPC tracks the reference trajectory and generates the end effector's motion.

taking into account the characteristics of each mode, we can change contact modes to stabilize the object's motion.

Although adaptively using multiple gait modes may enable a robot to realize a robust pivoting gait, it is challenging to select an adequate gait mode.

In this article, we propose a graph model predictive controller (MPC) to control the pivoting gait. Two gait modes are designed and put into a graph. The graph selects the gait mode and outputs a reference trajectory of the object to the MPC. Then, the MPC tracks the reference and generates the desired position and force of the end effector (EEF). Moreover, we use vision systems and force sensors to monitor the execution of the pivoting gait. If a disturbance is detected, the robot tries to change the gait mode by referring to the graph. The MPC compares the current state with the reference and realizes a feedback control. Simulation and experiments show that the DS mode is fast whereas the QS mode is stable. In addition, the ability to switch gait modes improves the robustness of the control system, and the robot can successfully achieve a stable pivoting gait during unexpected events, such as uncertainty in the object's mass and perturbation.

The contributions of this work are:

- The MPC is proposed to predict the future dynamics of the object during the pivoting gait.
- Two gait modes are designed to robustly perform the pivoting gait according to different purposes, i.e., DS mode for fast walking and QS mode for stable walking.
- A graph MPC is developed to select the sequence of modes. The advantage of the graph MPC is that, by designing cyclic motions in a graph, we can select gait modes and realize the feedback control in real time.
- We strengthen the control system with state feedback by adding vision and force sensors to resist disturbances.

In this paper, after introducing the related work in Section II, we formulate the pivoting gait and predictive controller in Section III. Section IV provides the design of the graph. In Section V, the simulation and experiments show that the proposed feedback control system is able to pivot an object to track a referenced trajectory while being free to switch gait modes, and it is robust against external perturbations and uncertainty in the object's weight. Section VI summarizes the results and describes future work.

II. RELATED WORK

A. NONPREHENSILE MANIPULATION

Nonprehensile manipulation is currently attracting increasing attention, as it enables manipulation of an object with fewer degrees of freedom [16]–[18], including the manipulation styles such as throwing [19], catching [20], [21], batting [22], [23], pushing [24], [25], sliding [26], rolling [27] and pivoting [28]–[30]. In nonprehensile manipulation, an object is manipulated without satisfying the form or force closure [31], which indicates that the object is sensitive to the environmental dynamics.

Among the several styles of nonprehensile manipulation, pivoting is a promising strategy for manipulating a heavy object. Aiyama *et al.* [32] originally proposed the pivoting gait, showing how heavy objects can be effectively manipulated. Yoshida *et al.* [33] proposed a method for planning the pivoting gait by a humanoid robot. More recently, Murooka *et al.* [34] studied whole-body manipulation for humanoid robots to achieve the pivoting task and explored simultaneous planning and estimation for humanoid pivoting tasks [35]. Shi *et al.* [36] proposed an aerial pivoting framework to pivot an object by aerial robots. Hou *et al.* [37] investigated the planar pivoting problem, in which a pinched object is reoriented to a desired pose through a swing motion.

The pivoting gait is a process of repeated pivoting manipulations around the left and right rotational vertices. We design two gait modes for the pivoting gait. To the best of our knowledge, there has been no previous research on feedback control of the pivoting gait with the ability to switch gait modes.

B. MODEL PREDICTIVE CONTROL

Recently, the MPC has been widely used in chemical processes [38], power systems [39], solar technology [40], [41] and flight control [42]. In robotics research, the MPC is frequently used in bipedal walking [43], as the MPC can effectively predict future dynamic behavior and cope with constraints on the state and the input [44]–[46]. Naveau *et al.* [47] modified the MPC schemes formulated as an optimization problem to include various gait modes. Valenzuela [48] introduced integer variables to represent the active contact modes and computed the mode sequences using mixed-integer nonlinear programming. The graph-based method also provides a way to plan a sequence of motions. Woodruff and Lynch [49] proposed a graph search algorithm to plan through a sequence of manipulation primitives describing different contact states. Tazaki and Imura [50] proposed a graph-based MPC that draws possible modes in a graph and simulates bipedal walking at different speeds. Murooka *et al.* [51] proposed loco-manipulation planning for humanoid robots based on the graph search.

While this research is inspired by the similarity between the pivoting gait manipulation and a humanoid's bipedal gait, these gaits are essentially different due to the following two factors. First, the formulation of a pivoting gait is more complex than that of a biped gait since a force-controlled dual-arm manipulator controls the contact mode of the grasped object. Second, a change in gait mode can be used to control the stability of the grasped object under gravity. This research applies the MPC to the pivoting gait in which a damping-controlled dual-arm manipulator is controlled to change the gait mode while predicting the object's future dynamics. Different from bipedal walking, we design a new QS gait mode by changing the object's supporting vertices. The QS mode provides a stable motion of walking especially when perturbation happens.

In this article, we investigate a real-time control method for the pivoting gait by considering two gait modes. A graph MPC is proposed to select the proper gait modes and to realize feedback control by using the vision and force information.

III. FORMULATION OF MODEL PREDICTIVE CONTROL

The target system of this work consists of two robot arms and a rigid object with a polygonal shape; see Fig. 2.

A. NOMENCLATURE

We describe here the notation used in this paper:

- Σ_*: Reference frames at $* = W, B, Hi, Pi$ and FCi representing the coordinate frames attached to the ground, the object, the center of the i-th spherical hand ($i = 1, 2$), the contact point on the object surface with the i-th hand, and the friction cone constraint at the contact point, respectively, where n and t, o denote the axes in the normal and two tangent directions, respectively.
- $p_*, R_*, \Psi_*, \omega_*$: The position vector, the rotation matrix, the Euler angle vector and the angular velocity vector of Σ_* with respect to Σ_W, respectively, where $* = B, Hi, Pi, FCi$ and $\Psi_* = [\phi_*, \psi_*, \theta_*]^{\mathrm{T}}$.
- X_k, Y_k, U_k: The states, outputs and free variables in the MPC taking place in the prediction horizon n_p, where $X_k = [x_k, \cdots, x_{k+n_p-1}]$ and the definitions of Y_k and U_k are similar to X_k.
- p_{BPi}: The vector directing from the origin of Σ_B to the origin of Σ_{Pi}.
- f_i: The force applied at the i-th contact point.
- $m_o, \mathcal{I}_o$: The object's mass and inertia matrix.
- g: The gravity force vector.

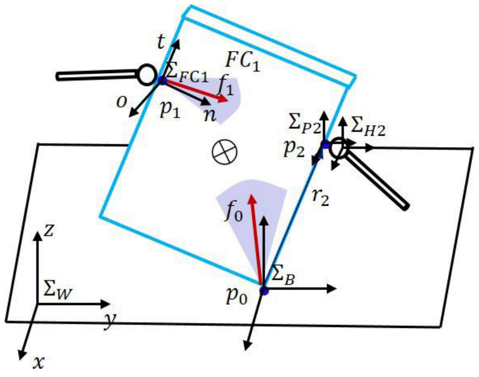


FIGURE 2. Example of frames and contact forces.

For simplicity, we rewrite p_{Pi} as p_i. We express the contact between the object and the ground during the single support phase by $i = 0$. In addition, we assume the point contact with friction at each contact point when an object's vertex contacts the ground.

B. KINEMATICS

The pivoting gait is a style of manipulating an object that iterates the steps of raising the object up on a vertex, rotating it around that vertex and putting it down. Due to the contact with the ground at a fixed vertex, the object motion is constrained as

$$SD_{B0}\begin{bmatrix}\dot{p}_B\\ \omega_B\end{bmatrix} = S\begin{bmatrix}\dot{p}_0\\ \omega_0\end{bmatrix} = o, \tag{1}$$

where S is a selection matrix that selects the linear velocity, and D_{Bi} transforms the linear/angular velocity from Σ_B to Σ_W,

$$S = [I_3 \quad O_3], \tag{2}$$

$$D_{Bi} = \begin{bmatrix} I_3 & -[(R_B{}^Bp_{BPi})\times] \\ O_3 & I_3 \end{bmatrix}, \tag{3}$$

where $[*\times]$ denotes the skew-symmetric matrix of a vector $*$ equivalent to the cross product operation and I_3 and O_3 denote the 3×3 identity and zero matrices, respectively.

Due to the contact with the i-th EEF ($i = 1, 2$), the object motion is constrained as

$$SD_{Bi}\begin{bmatrix}\dot{p}_B\\ \omega_B\end{bmatrix} = SD_{Hi}\begin{bmatrix}\dot{p}_{Hi}\\ \omega_{Hi}\end{bmatrix} = \dot{p}_i. \tag{4}$$

Given a reference trajectory of the object's rotation, we apply the MPC to generate the motion of the EEF and the target force applied by the EEF; see Fig. 3. With a knowledge of future dynamics, the MPC tries to find a way to pivot the object to the reference trajectory by solving an optimization problem. In this research, we consider the dynamics related to the rotation of the object to formulate the controller, since rotational dynamics are dominant to the gravitational stability of the object. In addition, we control the object motion in the single-support phase, whereas the quadruple- and double-support phases are used to connect two single-support phases. Hence, we formulate the state equation in the single-support phase.

Define the state vector as $x = [\Psi_B^T \quad \omega_B^T]^{\mathrm{T}}$. The relationship between the angular velocity ω_B and the velocity of the Euler angles $\dot{\Psi}_B$ is given by

$$\dot{\Psi}_B = W\omega_B, \tag{5}$$

where

$$W = \begin{bmatrix} 1 & \sin\phi_B \tan\theta_B & \cos\phi_B \tan\theta_B \\ 0 & \cos\phi_B & -\sin\phi_B \\ 0 & \sin\phi_B \sec\theta_B & \cos\phi_B \sec\theta_B \end{bmatrix}. \tag{6}$$

From this equation, the $k+1$-th step of the state can be predicted by

$$\begin{bmatrix} \Psi_B \\ \omega_B \end{bmatrix}_{k+1} = \begin{bmatrix} I_3 & W \\ O_3 & I_3 \end{bmatrix} \begin{bmatrix} \Psi_B \\ \omega_B \end{bmatrix}_k + \begin{bmatrix} WT^2/2 \\ T \end{bmatrix} [\dot{\omega}_B]_k, \tag{7}$$

where T denotes the sampling time.

C. DYNAMICS

The object is accelerated by the force applied by two EEFs. Euler's equation of the object's rotational motion can be obtained as

$$r_1 \times f_1 + r_2 \times f_2 + r_{com} \times m_o g = \mathcal{I}_o \dot{\omega}_B + \omega_B \times \mathcal{I}_o \omega_B, \tag{8}$$

where $r_i = p_i - p_0$ $(i = 1, 2)$, $r_{com} = p_{com} - p_0$ and p_{com} denotes the position vector of the object's CoG. Substituting (8) into (7) and defining the force applied by the EEF to be the input $u_k = [f_1 \quad f_2]_k^T$, we obtain the state equation as follows:

$$x_{k+1} = Ax_k + Bu_k + D, \tag{9}$$

where A, B, and D are coefficient matrices defined as

$$A = \begin{bmatrix} I_3 & W \\ O_3 & I_3 \end{bmatrix}, \tag{10}$$

$$B = \begin{bmatrix} \mathcal{I}_o^{-1} WT^2/2[r_1 \times] & \mathcal{I}_o^{-1} WT^2/2[r_2 \times] \\ \mathcal{I}_o^{-1} T[r_2 \times] & \mathcal{I}_o^{-1} T[r_2 \times] \end{bmatrix}, \tag{11}$$

$$D = \begin{bmatrix} \mathcal{I}_o^{-1} WT^2/2[r_{com} \times] mg \\ \mathcal{I}_o^{-1} T[r_{com} \times] mg \end{bmatrix}. \tag{12}$$

The coefficient matrices B and D, which include m_o and $\mathcal{I}_o$, reflect the dynamics of the object. Here, we note that, during the pivoting gait, the supporting vertex of the object expressed by p_o changes between two consecutive single support phases.

Considering the prediction horizon n_p, we define the states and the free variables taking place in the prediction horizon as $X_k = [x_k, x_{k+1}, \cdots, x_{k+n_p-1}]^T$ and $U_k = [u_k, u_{k+1}, \cdots, u_{k+n_p-1}]^T$, respectively. According to (9), we have

$$X_{k+1} = A' X_k + B' U_k + D', \tag{13}$$

where the coefficient matrices are

$$A' = [A, A^2, \ldots, A^{n_p}], \tag{14}$$

$$B' = \begin{bmatrix} A^0 B & \cdots & 0 \\ \ldots & \ddots & \vdots \\ A^{n_p-1} B & \cdots & A^0 B \end{bmatrix}, \tag{15}$$

$$D' = \begin{bmatrix} A^0 D & \cdots & 0 \\ \ldots & \ddots & \vdots \\ A^{n_p-1} D & \cdots & A^0 D \end{bmatrix}. \tag{16}$$

D. OUTPUT EQUATION

The system outputs the velocity of the EEF. From (1) and (3), $\dot{p}_i$ can be obtained as

$$\dot{p}_i = SD_{Bi} \begin{bmatrix} [(R_B{}^B p_{BPi}) \times)] \\ I_3 \end{bmatrix} \omega_B. \tag{17}$$

From this equation, the velocity of the EEF can be obtained as

$$\dot{p}_{Hi} = S(SD_{Hi})^+ \dot{p}_i + (I_3 - S(SD_{Hi})^+ SD_{Hi})k_v, \tag{18}$$

where $*^+$ is the pseudo-inverse of a matrix $*$ and k_v denotes a 6-dimensional vector. By setting $y_k = [\dot{p}_{H1}^T \quad \dot{p}_{H2}^T]_k^T$ and Y_k as the combination of y_k over the prediction horizon, we can define the output equation in the form of

$$Y_k = CX_k + E, \tag{19}$$

where the coefficient matrices C and E can be easily defined from (17) and (18).

E. COST FUNCTION

The cost function used in the MPC is defined as

$$J_{mpc} = \frac{\alpha}{2} \left\| X_{k+1} - X_o^{ref} \right\|^2 + \frac{\beta}{2} \left\| U_k \right\|^2, \tag{20}$$

where α and β are the weights. X_o^{ref} is the reference trajectory of the object along the prediction horizon and it is provided by the graph that we will discuss in Section IV. The first and the second terms in (20) denote the amount of the state error and the amount of the input force of the EEF. Considering (13), the cost function can be formulated in the following form

$$J_{mpc} = \frac{\alpha}{2} \left\| A' X_k + B' U_k + D' - X_o^{ref} \right\|^2 + \frac{\beta}{2} \left\| U_k \right\|^2. \tag{21}$$

From this cost function, we can formulate the quadratic programming (QP) problem as

$$\min_{U_k} \frac{1}{2} U_k^T Q U_k + r^T U_k + s, \tag{22}$$

where

$$Q = \frac{\alpha}{2} B'^T B' + \frac{\beta}{2} I_{2n_p}, \tag{23}$$

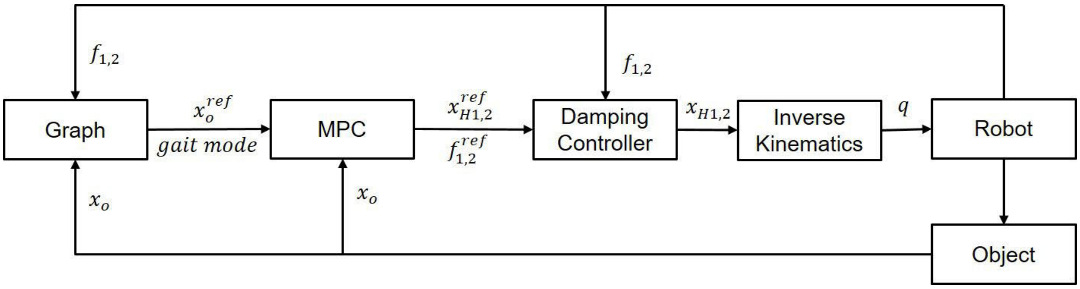


FIGURE 3. This diagram describes the feedback loop used to control the pivoting gait. According to x_0, the state of the object captured by the camera, and $f_{1,2}$, the force of the robot hands that is collected by the force sensors, the graph selects a walking mode and outputs x_o^{ref}, a reference trajectory of the object. Then, the MPC generates both $x_{H1,2}^{ref}$, the desired hand trajectories, and $f_{1,2}^{ref}$, the desired forces. A damping controller is implemented to handle the force control and outputs $x_{H1,2}$, the hand trajectories. After performing inverse kinematics, the joint configuration q is sent to the robot, and then the robot manipulates the object.

$$r = \alpha B'^T (A'X_k + D' - X_o^{ref}), \tag{24}$$

$$s = \frac{\alpha}{2}(A'X_k + D' - X_o^{ref})^T (A'X_k + D' - X_o^{ref}). \tag{25}$$

QP solvers such as qpOASES [52] and quadprog can be implemented to solve this optimization problem.

F. FORCE CONSTRAINT

Suppose the friction coefficient at each contact point is known. The interaction force at the contact location must lie within the friction cone. In this research, the friction cone has been approximately linearized by using the regular 4-sided polygon. We have

$$f_i \in FC_i = \{f_i : |f_i^t| \leq \mu f_i^n, |f_i^o| \leq \mu f_i^n, 0 \leq f_i^n \leq f_{max}^n\}, \tag{26}$$

where f_{max}^n is a designed upper bound of the normal contact force. An example of the frame Σ_{FC1} with axes $\{n, t, o\}$ at contact point p_1 is shown in Fig. 2. Since we set $u_k = [f_1 \quad f_2]_k^T$, (26) acts as constraints to the input,

$$-L_{max} \leq H_c u_k \leq L_{max}, \tag{27}$$

where

$$L_{max} = \left[\mu f_{max}^n \ \mu f_{max}^n \ f_{max}^n \ \mu f_{max}^n \ \mu f_{max}^n \ f_{max}^n\right]^T \tag{28}$$

denotes the bound on the normal force component and

$$H_{ck} = \begin{bmatrix} R_{FC1}^T & O_3 \\ O_3 & R_{FC2}^T \end{bmatrix}. \tag{29}$$

Considering the free variables in the prediction horizon, we have

$$-L'_{max} \leq H'_c U_k \leq L'_{max}, \tag{30}$$

where L'_{max} is a matrix with n_p elements, $L'_{max} = [L_{max}, \cdots, L_{max}]^T$ and $H'_c = [H_{c1}, \cdots, H_{ck}]$.

G. DAMPING CONTROL

After applying the MPC, we obtain the desired interaction force as an input u_k. This subsection describes how to apply the desired interaction force and the EEF trajectory to a real robot. We apply the damping control [53] where the EEF is controlled to behave like a mechanical damper. The damping control can be defined as

$$D_{imp}\dot{x}_{Hi} = f_i - f_i^{ref}, \tag{31}$$

where D_{imp} denotes the damping matrix, and the value of the damping factor is selected as 0.15 in this work. f_i is the force collected from the force sensor on the wrist of the i-th hand. f_i^{ref} indicates the reference force of the i-th hand. The reference is provided by u_k, the input of the MPC. Controlling a robot to realize the desired damping depends on the types of joint servo controllers. In our case, we use a velocity-controlled industrial robot equipped with a damping-control function.

IV. GRAPH MODEL PREDICTIVE CONTROL

To adaptively select the gait mode, we introduce the graph MPC. If a disturbance is detected by the vision and force sensors, we update the weight of the edge. Then, the graph selects the new gait mode and provides a reference trajectory of the object to the MPC. The MPC drives the object from the current state to the reference state and realizes a feedback control.

A. GRAPH

The graph expresses the change in contact states between the object and the ground, whereas the cost function, as a weight of each edge, is designed to select a path. By searching for the graph, we can select a gait mode with the lowest cost that outputs a sequence of key poses as a reference to the MPC.

In the graph, each node includes the object's configuration and a supporting state of the object, i.e., SS, DS, and QS phases. To compose the graph, we define two types of motions: principal motion and switching motion. The principal motion is cyclic as shown in Fig. 4. The inner loop corresponds to a cyclic motion of the QS gait mode in which the SS and QS phases alternate. On the other hand, the outer loop corresponds to a cyclic motion of the DS gait mode. Here, the SS phase includes the right-foot and left-foot support phases. The right-foot SS phase comes after the DS or

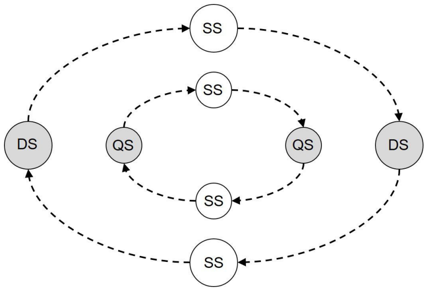


FIGURE 4. A node indicates a supporting state of the object. A gait mode is represented by four nodes. A loop indicates a principal walking motion. The inner and outer loops correspond to the QS and DS gait modes, respectively.

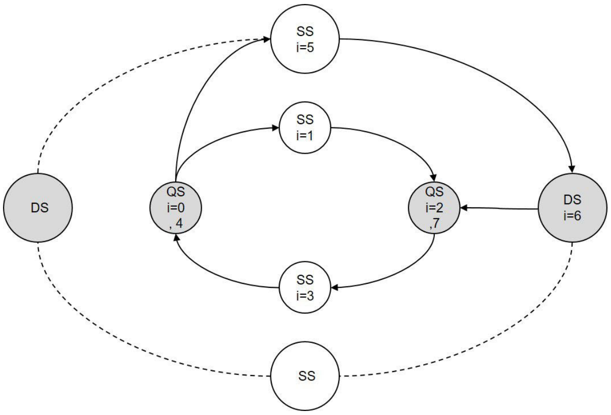


FIGURE 5. Example of a graph. *i* denotes a sequence of motions. The solution path is drawn by solid arrows, whereas the candidate path is drawn by dashed arrows. The object first walks in the QS mode, then walks in the DS mode, and finally rests in a QS pose.

QS phase, which comes after the left-foot SS phase, and vice versa. On the other hand, the switching motion is a transient motion from one principal motion to the other.

An overview of the graph that includes an example of the solution path is shown in Fig. 5. The nodes are connected by edges that are drawn by solid and dashed arrows, in which solid arrows denote an example of the solution path. In addition, the nodes that are included in the solution path include the number i, which indicates the sequence. In the example shown in Fig. 5, the object starts from the initial node ($i = 0$), walks in the QS mode first (from $i = 0$ to 4), then switches to the DS mode ($i = 5, 6$) and finally rests in the QS pose ($i = 7$).

B. WEIGHTS OF EDGES: COST FUNCTIONS

Each edge of the graph includes two pieces of information: the transient time to move from one node to the other node and the weight. In this subsection, we design a cost function as the edge's weight. To evaluate the desirability of the gait mode, we compare the cost of paths and select a gait mode with the lowest cost. We assign the cost J_{path} of a path that includes n edges. It is composed of a cost J_s that is related to the states, J_t that is related to time, and J_{dst} that is related to external disturbance,

$$J_{path} = \alpha_g J_s + \beta_g J_t + \gamma_g J_{dst}, \tag{32}$$

where α_g, β_g and γ_g denote the weights of each cost function. The weights can be set free under different purposes; for example, if a fast walking motion is desired, we can increase the value of β_g. The cost functions are defined as follows. J_s indicates the summation of transition costs between nodes. J_t indicates the summation of times taken to complete the transition. J_{dst} is designed for updating the cost once a disturbance is detected.

$$J_s = \sum_{i=0}^{n-1} \|x_i - x_{i+1}\|^2, \tag{33}$$

where x_i is the reference state of the object saved in node i. The reference state is designed with respect to the type of support, for example, the SS, the DS, and the QS. For all three types of support, the yaw angles are designed according to the step length. Then, in the QS, the roll and pitch angles are zero. In the DS, the roll angle is zero, while the pitch angle can be selected from $(0, -\pi/2)$. In the SS, the pitch angle is selected from $(0, -\pi/2)$, while the roll angle is selected from $(0, \pi/2)$ or $(0, -\pi/2)$ depending on whether the gait is from the left or the right rotation vertex, respectively. By selecting the same pitch angles in the SS and the DS, the DS gait mode behaves more quickly than the QS gait mode because the DS gait mode saves the motion of tracking the pitch angle. n indicates the number of edges along the path. In this work, we select $n = 4$ because four edges are enough to represent a gait mode.

$$J_t = \sum_{i=0}^{n-1} t_i, \tag{34}$$

where t_i is the duration of time taken for the transition from x_i to x_{i+1} along the edge. The duration of time is manually selected, and the duration of time selected for the DS mode is shorter than that of the QS mode because the DS mode is designed to move more quickly. Note that the time duration in the graph between x_i and x_{i+1} is t_i, which is different from the sampling time T between x_k and x_{k+1} in the MPC. A cost function of disturbance J_{dst} is defined as

$$J_{dst} = \begin{cases} \delta & \text{if a disturbance is detected} \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

δ is a positive value indicating the existence of an external disturbance.

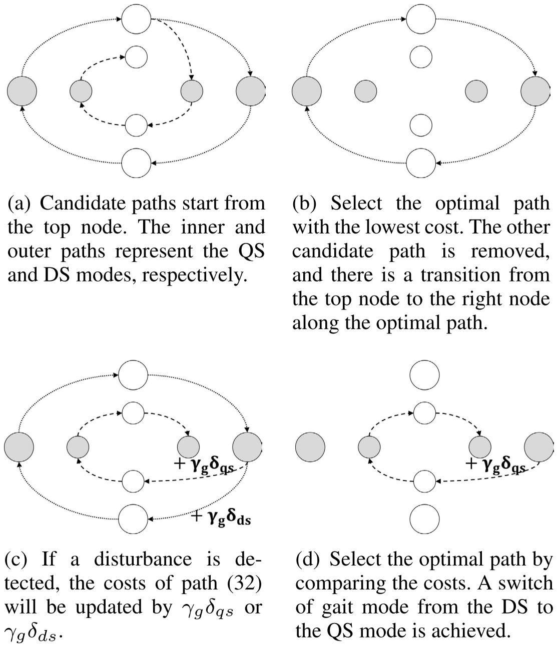
$$\delta = \begin{cases} \delta_{ds} & \text{for DS gait mode} \\ \delta_{qs} & \text{for QS gait mode.} \end{cases} \tag{36}$$

δ includes both δ_{qs} and δ_{ds}. We set $\delta_{ds} > \delta_{qs}$ because a disturbance causes a bigger influence on the DS mode.

Both cameras and force sensors are used to detect the disturbance. Define Ψ^{thr} and f_i^{thr} as the thresholds to detect the occurrence of the disturbance. The external disturbance is detected if the following conditions are met:

$$|\Psi^{cur} - \Psi^{pre}| \geq \Psi^{thr}, \tag{37}$$

$$|f_i^{cur} - f_i^{pre}| \geq f_i^{thr}, \tag{38}$$



(a) Candidate paths start from the top node. The inner and outer paths represent the QS and DS modes, respectively.

(b) Select the optimal path with the lowest cost. The other candidate path is removed, and there is a transition from the top node to the right node along the optimal path.

(c) If a disturbance is detected, the costs of path (32) will be updated by $\gamma_g\delta_{qs}$ or $\gamma_g\delta_{ds}$.

(d) Select the optimal path by comparing the costs. A switch of gait mode from the DS to the QS mode is achieved.

FIGURE 6. The procedure to select a gait mode in the graph.

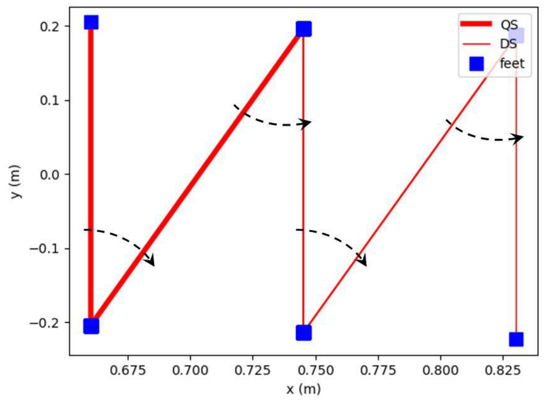


FIGURE 7. Footprints of the walking motion. The blue squares are the hind support feet of the object. The red line indicates the edge connecting the hind feet, where the thicker and thinner lines indicate the QS and DS modes, respectively. The black dashed line indicates the swing motion. The object starts to walk in the QS mode for one step and then walks forward for 3 steps in the DS mode.

where Ψ^{cur} and f_i^{cur} are the current Euler angles detected by the cameras and the force data collected from the force sensor on the i-th arm, respectively. Ψ^{pre} and f_i^{pre} indicate the data at the previous sampling time. After designing the nodes and edges, we can select the gait mode by looking for the optimal path with the minimum cost.

C. SELECTION OF GAIT MODES IN THE GRAPH

By comparing the cost functions of the paths, we can select an optimal path that contains information on the gait modes. Fig. 6 shows the procedure for finding the optimal path.

The top node is selected as a starting node; see Fig. 6(a). Suppose $n = 4$ in a candidate path, and two candidate paths are shown by directed arrows. The inner dashed arrow indicates the QS gait mode, while the outer dot arrow indicates the DS gait mode.

By comparing the cost J_{path} of the candidate paths, only the path with the lowest cost remains and is selected as the optimal path; see Fig. 6(b). Then, a transition happens from the current node to the next node along the optimal path.

If a disturbance is detected, the costs of the paths in (32) will be updated because of the change of J_{dst} in (35). This change can be reflected in the graph; for example, see Fig. 6(c). $\gamma_g\delta_{qs}$ and $\gamma_g\delta_{ds}$ are added to the costs of the inner path and outer path, respectively. After comparing the costs of the paths, the inner path is selected as the optimal path; see Fig. 6(d). A transition from the outer path to the inner path indicates that the gait mode is switched from the DS mode to the QS mode.

In this work, the main task of the graph is to select the gait modes and generate a reference trajectory of the object corresponding to the selected gait modes. The reference trajectory is then provided to the MPC to track. If there is a disturbance during motion, the robot will stop its motion and wait for the new command. Meanwhile, the edges in the graph are updated, where J_{dst} is added to their weights. Considering the change of edges, the graph selects new gait modes and outputs a reference trajectory to the MPC. Then, based on the current state and the reference, the MPC generates the trajectory of the EEFs, and the robot moves.

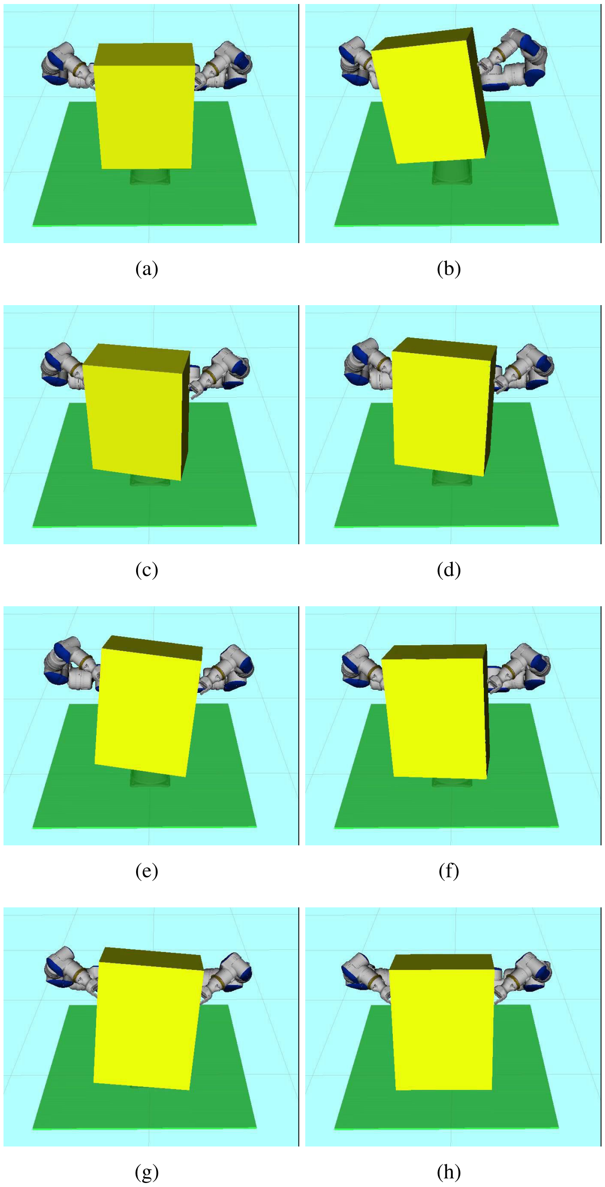
V. SIMULATION AND EXPERIMENTS

The target object is a box in the size of 0.6 × 0.4×0.2 m and its weight is 1.4 kg. We use a dual-arm robot (Yaskawa Motoman SDA5F) to manipulate the box and change the gait mode during motion.

A. SIMULATION

In the simulation, we assume that the object walks stably at the beginning and then walks quickly, which involves a change of the gait mode. This can be done simply by modifying the weights in (32). The obtained sequence of modes is the QS mode for the first step and then the DS mode for three steps. In Fig. 7, the top view of the gaits generated by the MPC is shown, in which the support feet are marked as blue squares, and the edge connecting two support feet is drawn in red lines, where the thicker line indicates the QS mode and the thinner line indicates the DS mode. We can observe that the gait mode changed from QS to DS after the first step. The rotation around the support foot is shown by black dashed arrows, and the rotation indicates the yaw angle of the object.

We simulate the walking motion in the RVIZ. In Fig. 8(b), the object rotates around the vertex (which we define as the right foot) near the right EEF of the robot. Then, the gait mode transforms from QS (Fig. 8(c)) to DS (Fig. 8(d)). The supporting foot changes to the left in Fig. 8(e). Similarly, a foot change happens in Fig. 8(f) and Fig. 8(g), and finally, the object is moved to a rest pose in Fig. 8(h). The simulation



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 8. Simulation of the pivoting gait. The walking motion starts from the QS mode for the first step and then switches into the DS mode for 3 steps.

shows that a change in gait mode can be realized during the motion of the pivoting gait.

B. EXPERIMENT 1: DS AND QS MODES

We evaluate the performance of the controller design in test scenarios through experiments. Two ball-shaped EEFs are designed to keep a point contact between the robot and the box. Force sensors are installed on both sides of the robot's wrists. Impedance control [54], [55] is implemented to control the force between the robot's EEF and the box. The robot in use is Yaskawa Motoman SDA5F, and the controller for the robot is FS100. In front of the robot, a box is placed on a table at a height of 80 cm; see Fig. 11(a).

To determine the characteristics of the gait modes, the first experiment is running the pivoting gait in a single gait mode.

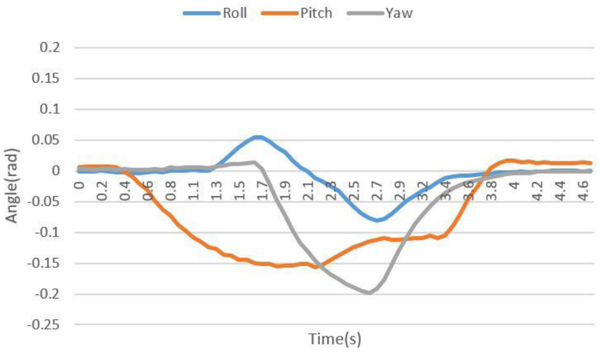


FIGURE 9. The Euler angles of the object that walks in the DS mode. At time 2.1 s, only the roll angle goes to zero, which indicates the DS pose.

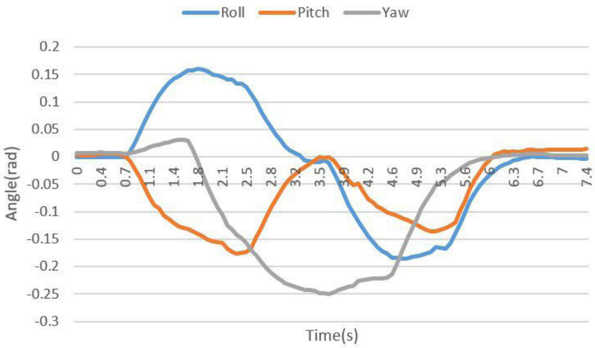


FIGURE 10. The Euler angles of the object that walks in the QS mode. At time 3.5 s, both the roll and pitch angles go to zero, which indicates the QS pose.

TABLE 1. Time for making two steps.

Mode	Time (s)	Step length (m)
DS	2.2	0.085
QS	5.6	0.085

The step length in the DS mode (Fig. 11) and that in the QS mode (Fig. 12) are the same. In the DS mode experiment, Fig. 9 shows the Euler angles of the object during walking, and Fig. 11 shows the motion. First, the object is rotated from the initial pose (Fig. 11(a)) to a DS pose (Fig. 11(b), as indicated at time 1.2 s in Fig. 9). Then, the object walks forward (Fig. 11(c)), changes its support foot from right to left (Fig. 11(d), indicated at time 2.1 s in Fig. 9), and finally goes into a rest pose in Fig. 11(f).

In the QS mode experiment, see Fig. 10 and Fig. 12. The robot starts to rotate the box around the right vertex and lifts it to a SS pose (at time 2.1 s in Fig. 10 and Fig. 12(b)). Then, the box goes into a QS pose to change its support foot (at time 3.6 s in Fig. 10 and Fig. 12(c)) and moves to the target position (Fig. 12(e)).

In Table 1, the time for executing two QS steps takes 5.6 s, while the time for making two steps in DS takes 2.2 s. A quicker motion of the DS gait mode is achieved by selecting t in the DS mode shorter than that in the QS mode. The reasons to design a quicker motion of the DS gait mode

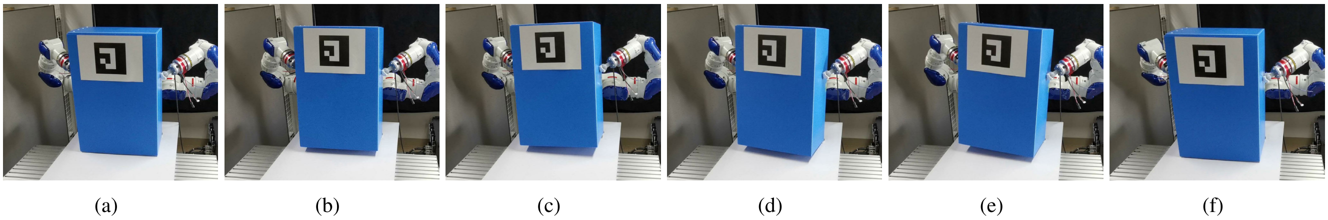


FIGURE 11. The robot pivots the box to walk in the DS mode for two steps.

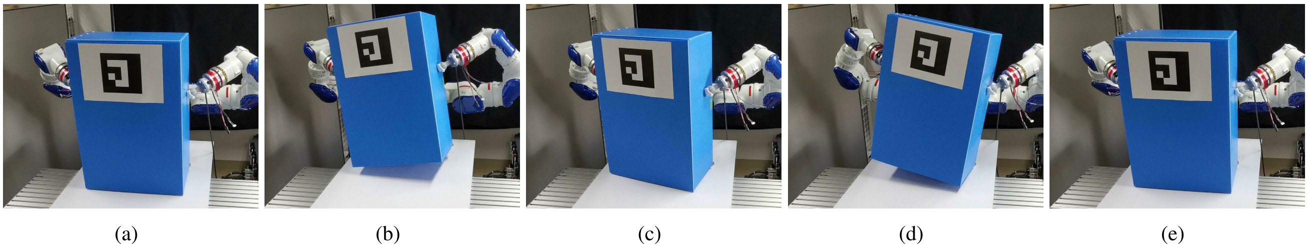


FIGURE 12. The robot pivots the box to walk in the QS mode for two steps.

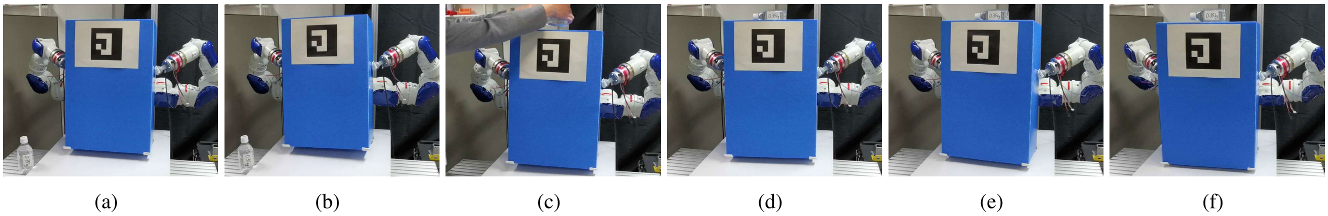


FIGURE 13. The experiment of placing a 0.35 kg bottle on the manipulated object during the motion of pivoting; the robot successfully pivots the object in DS mode.

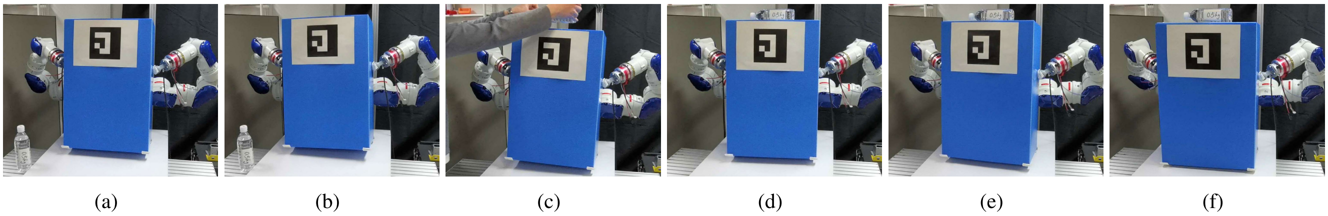


FIGURE 14. The experiment of placing a 0.5 kg bottle on the manipulated object during the motion of pivoting; the robot successfully pivots the object in DS mode.

are shown below. The QS pose requires that all four bottom vertices are on the floor (see the zero values of the roll and pitch at time 3.5 s in Fig. 10), while in the DS pose, two vertices are on the floor and two vertices are on the fly (see the values of roll and pitch at time 2.1 s in Fig. 9). Since the SS pose requires that three vertices are on the fly, the DS gait mode saves a motion of landing the two front vertices compared to the QS gait mode, and it leads to a faster motion in the DS mode. Furthermore, the peak value of the roll angle in QS is larger than that in DS, which also slows down the motion of the QS mode. The reason for designing a rather large value of the roll angle in QS is that we leave some space in the roll angle to absorb perturbation, in order to avoid scuffing and to walk stably.

In experiment 1, the motions of the pivoting gait based on both the DS and the QS gait modes are achieved. The pivoting gait in the DS mode is faster than that in the QS mode. Although the QS mode is slow, we believe it has the potential to maintain stable walking, which we test in the following experiments.

C. EXPERIMENT 2: UNCERTAINTY IN THE OBJECT's MASS

Though the MPC is inherently able to cope with external disturbances, we believe selecting the proper gait mode improves the robustness of the controller. We design experiments regarding the pivot gait with uncertainty in the object's mass to check the following:

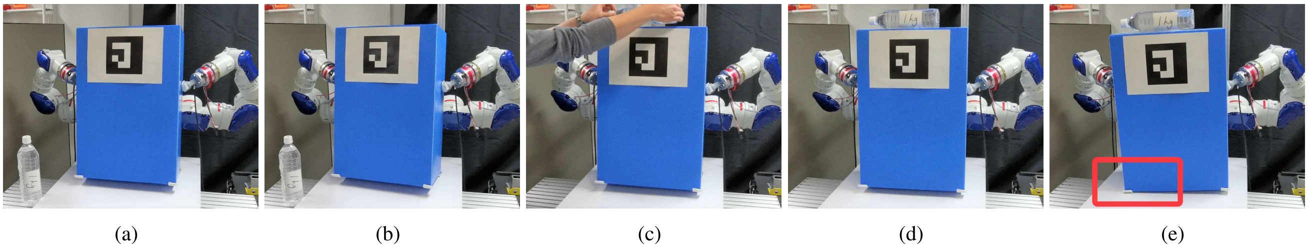


FIGURE 15. The experiment of placing a 1 kg bottle on the manipulated object during the motion of pivoting; the robot fails to pivot the object. After the placement in Fig. 15(c), the object lands in the DS pose; see Fig. 15(d). When lifting the object, the motion fails and a collision between the table and the right front foot can be seen in the red rectangle in Fig. 15(e).

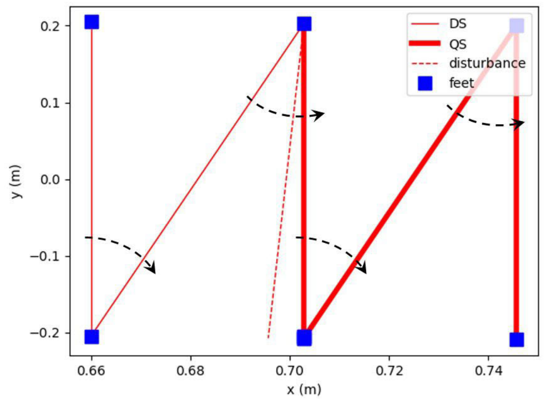


FIGURE 16. Footprints in the experiment of placing a 2 kg bottle. The placement of the bottle is detected and shown by the red dashed line. After detecting the disturbance, the gait mode is transferred from DS mode to QS mode.

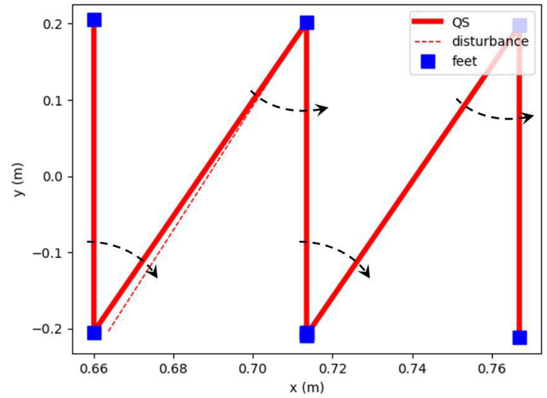


FIGURE 17. Footprints of experiment 3. A perturbation is detected by the vision system, as shown by the red dashed line that is close to the first step. Though perturbation occurs, the MPC still finds a solution to follow the planned trajectory.

- The performance of the two designed gait modes when there is a disturbance.
- If the switching of the gait mode improves the robustness of the system.

The weight of the box is 1.4 kg, and bottles of different weights are placed on the box during the walking process. In the experiments, the weights of the bottles are 0.35 kg, 0.5 kg, 1 kg, and 2 kg.

We first test the pivoting gaits in the DS gait mode by placing a bottle on the manipulated object during motion.

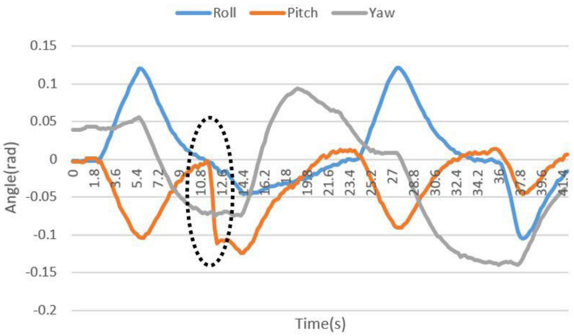
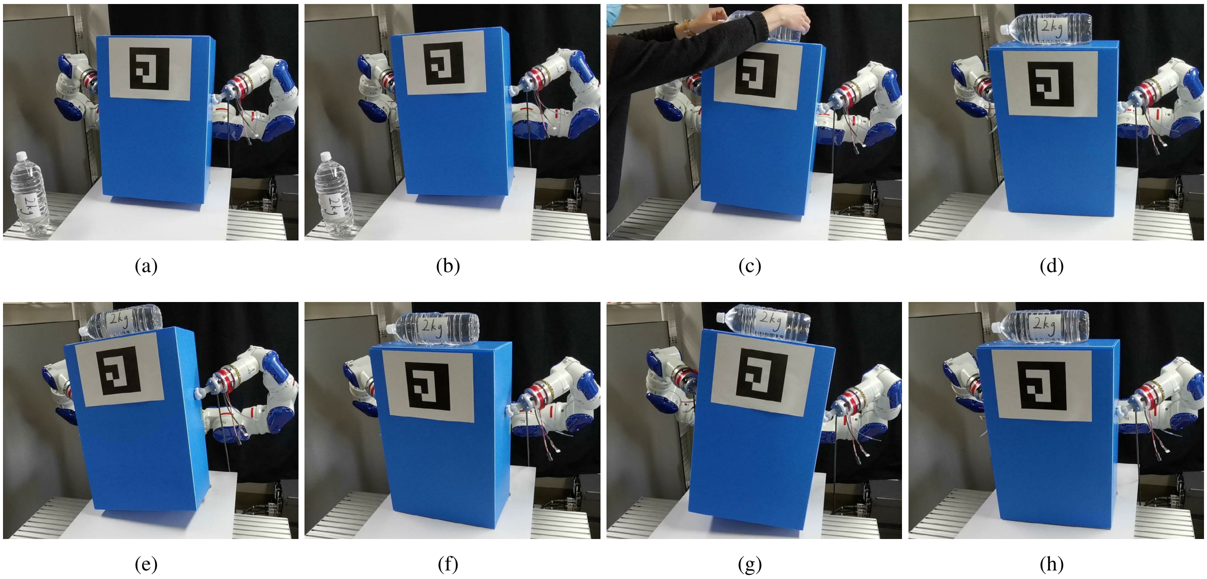


FIGURE 18. The Euler angles of the box in experiment 3. The black dashed circle shows a fluctuation in the pitch angle caused by pushing, detected at time 11.8 s. After the disturbance, the controller first decreases the roll and pitch angles, which moves the box into the SS pose to avoid scuffing during walking. Later, the controller tunes the roll and pitch angles to zero, which indicates a landing pose at time 24.2 s, and then it pivots the box to walk.

Bottles of weights 0.35 kg, 0.5 kg and 1 kg are placed; see Fig. 13, Fig. 14, and Fig. 15, respectively. In all three of the experiments, the robot starts to pivot the object in the DS mode; see Fig. 13(a), Fig. 14(a), and Fig. 11(a). In Fig. 13(c), Fig. 14(c), and Fig. 15(c), the bottles of weights 0.35 kg, 0.5 kg and 1 kg, respectively, are placed on top of the box during the walking motions, which results in a change in the object's mass. In Fig. 13(d) - Fig. 13(f) and Fig. 14(d) - Fig. 14(f), the robot successfully pivots the object to walk after the placement of bottles of weights 0.35 kg and 0.5 kg. In the case of placing the 1 kg bottle, the object lands in a DS pose after the placement of the bottle; see Fig. 15(d). However, the robot fails to lift the object after switching the rotational vertex from the right to the left. Furthermore, unfortunately, a collision occurs between the table and the front right foot of the object; see the red rectangle in Fig. 15(e).

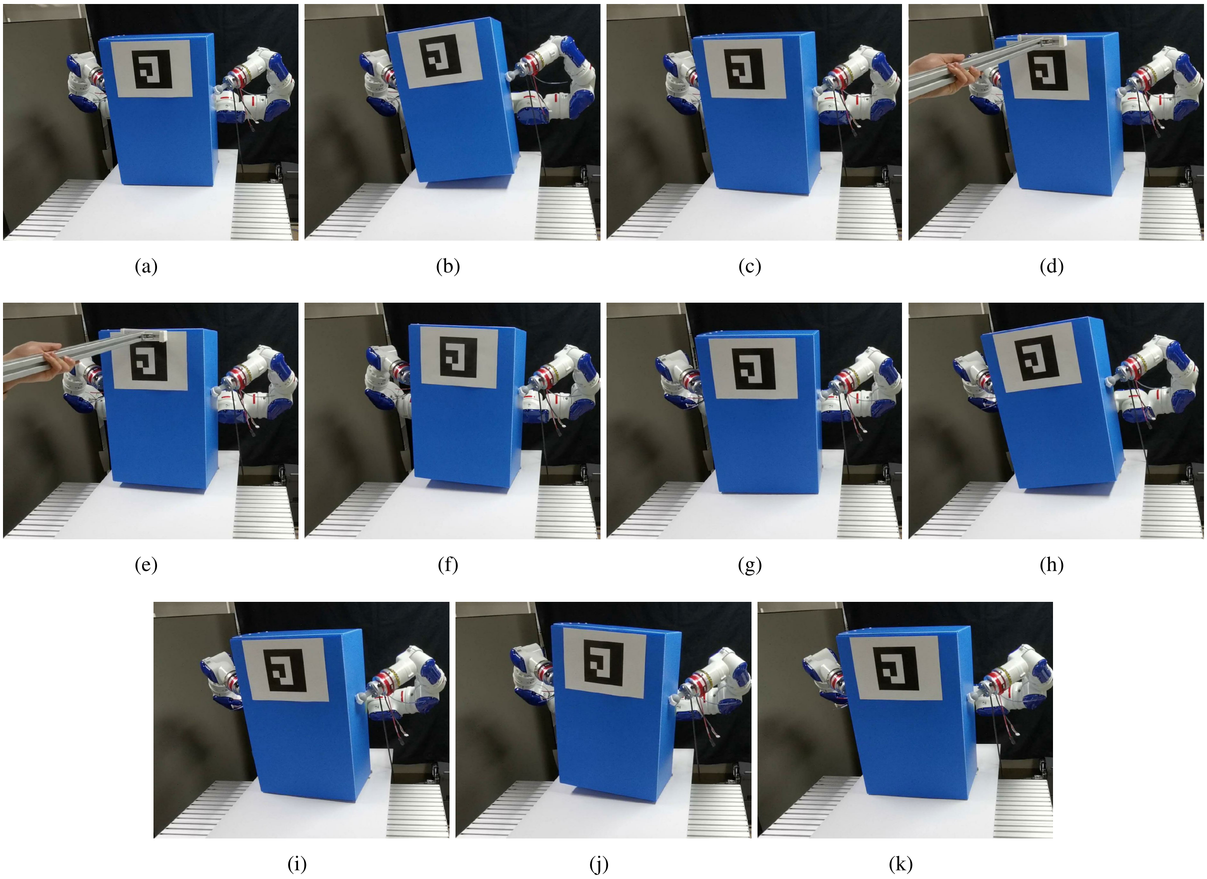
Since the failure arises during the switching of the rotational vertices, we improve the pivoting gait by changing the switching gait mode to the QS mode, in which all four vertices of the object are on the table, which leads to a firm contact between the object and the table. With the help of the graph MPC, we can change the gait mode from the DS to the QS gait mode according to the environment. In the experiment shown in Fig. 19, the robot first manipulates the object in the DS mode; see Fig. 19(a) and Fig. 19(b). A 2 kg bottle is placed on top of the box; see Fig. 19(c). The placement is



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE 19. The experiment of placing a 2 kg bottle on the manipulated object during the motion of pivoting. A 2 kg bottle is placed on top of the box during walking; see Fig. 19(c). After the placement, the robot changes its gait mode from DS mode to QS mode (Fig. 19(d)) and pivots the box to walk.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k)

FIGURE 20. An unexpected push is acted on the box at Fig. 20(d). The robot pivots the object to walk and reacts to resist the disturbance.

detected by the force sensors in the wrist of the robot, and the graph switches the gait mode from the DS mode to the QS mode. After the placement, the object lands in the QS pose in Fig. 19(d). Then, the robot pivots the object to walk for two more steps in the QS mode and successfully finishes the pivoting gait; see Fig. 19(e) - Fig. 19(h). This experiment

TABLE 2. Placements of bottles of different weights.

Weight of bottle (kg)	Gait Mode	Switching of Mode	Result
0.35	DS	No	Succeed
0.5	DS	No	Succeed
1.0	DS	No	Fail
2.0	DS + QS	Yes	Succeed

The robot successfully pivots the object in the DS mode after the placements of 0.35 kg and 0.5 kg bottles, but fails to pivot the object with a 1 kg bottle. A switching of the gait mode from the DS to the QS mode enables the robot to pivot the object after the placement of a 2 kg bottle.

shows that 1.) The QS mode is more stable than the DS mode when a disturbance occurs; 2.) Switching of the gait mode improves the robustness of the control system.

Table 2 shows the results of the experiments in this subsection. The pivoting gait in the DS mode succeeds against little disturbances, for example, disturbances from the placement of bottles of weights 0.35 kg and 0.5 kg. The pivoting gait in DS mode fails to carry a 1 kg bottle. However, after implementing the graph MPC that enables the switching gait mode, the robot successfully pivots the object after the placement of a 2 kg bottle by switching from DS mode to the QS mode. The experiments show that the QS mode is more stable than DS mode, and the ability to switch gait modes improves the robustness of the control system.

D. EXPERIMENT 3: EXTERNAL PERTURBATION

In the pivoting gait, if the robot continues manipulation without feedback, the error of the position of the object will accumulate. As a solution to this problem, we use a visual system to watch the state of the object. When a perturbation causes a relative motion between the EEFs and the object, the combination of the visual system and the graph MPC is useful to compare the tracking data with the desired state of the object. Then, the robot modifies its motion and recovers from the perturbation.

Three cameras are placed in front of the table, as shown in the red rectangle in Fig. 1. The cameras detect the Euler angles and the position of the marker on the box. In the experiment, the robot starts to pivot the object to walk in the QS mode; see Fig. 20(a) - 20(c). An unexpected push is acted on the box during walking by a metal stick; see Fig. 20(d). The states of the object are suddenly changed and captured by the vision system; see Fig. 18. After the disturbance, a new motion is generated, and the QS mode is selected as the gait mode because it provides stable walking. In the new motion, the robot first lifts the box into a SS pose to avoid the box scuffing the table; see Fig. 18 from time 12.4 - 14.4 s and Fig. 20(f). Then, the robot lands the box into the QS pose to firmly contact the table; see Fig. 18 at time 24.2 s and Fig. 20(g). After landing, the robot pivots the object to walk for two more steps; see Fig. 20(h) - Fig. 20(k).

This experiment shows that, with the help of the vision system, the proposed control system is able to recover from the perturbation and track the reference trajectory.

VI. CONCLUSION AND FUTURE WORK

In this article, we use a dual-arm robot to pivot an object to walk. Two gait modes are designed for adapting to the environment: DS mode is for fast walking, and QS mode is for stable walking. Vision systems and force sensors are implemented to perceive the environment. A graph MPC is proposed, in which the graph selects the gait mode based on the information collected from both the force sensors and the vision system and outputs a reference trajectory that is tracked by the MPC.

The experiments show that the gait mode influences the robustness of the system, especially when a disturbance occurs. The DS gait mode is fast, while the QS mode is more stable. By adaptively switching the gait mode according to the applied external disturbance, a robot can stably perform the pivoting gait. With robust control of the pivoting gait, we are able to automate the manipulation of a large and heavy object, such as furniture.

The extension is to apply our pivoting approach to objects of different shapes. This requires identification of the appropriate model, selection of the supporting vertices, design of the grasp positions, and design of new gait modes. Our final goal is to implement the pivoting gait as an ultimate way to manipulate large and heavy objects by robots.

REFERENCES

[1] L. Berscheid, P. Meisner, and T. Kroger, “Self-supervised learning for precise Pick-and-Place without object model,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4828–4835, Jul. 2020.

[2] M. Gualtieri, A. T. Pas, and R. Platt, “Pick and place without geometric object models,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7433–7440.

[3] J. Xu, K. Harada, W. Wan, T. Ueshiba, and Y. Domae, “Planning an efficient and robust base sequence for a mobile manipulator performing multiple Pick-and-place tasks,” 2020, *arXiv:2001.08042*. [Online]. Available: http://arxiv.org/abs/2001.08042

[4] 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, Jan. 1999.

[5] Z. Li and J. Canny, “Motion of two rigid bodies with rolling constraint,” *IEEE Trans. Robot. Autom.*, vol. 6, no. 1, pp. 62–72, Feb. 1990.

[6] K. M. Lynch and M. T. Mason, “Stable pushing: Mechanics, controllability, and planning,” *Int. J. Robot. Res.*, vol. 15, no. 6, pp. 533–556, Dec. 1996.

[7] K. Harada, S. Kajita, F. Kanehiro, K. Fujiwara, K. Kaneko, K. Yokoi, and H. Hirukawa, “Real-time planning of humanoid Robot’s gait for force-controlled manipulation,” *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 1, pp. 53–62, Feb. 2007.

[8] J. Zhou, Y. Hou, and M. T. Mason, “Pushing revisited: Differential flatness, trajectory planning, and stabilization,” *Int. J. Robot. Res.*, vol. 38, nos. 12–13, pp. 1477–1489, Oct. 2019.

[9] N. Doshi, F. R. Hogan, and A. Rodriguez, “Hybrid differential dynamic programming for planar manipulation primitives,” 2019, *arXiv:1911.00175*. [Online]. Available: http://arxiv.org/abs/1911.00175

[10] K. M. Lynch, “Toppling manipulation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, May 1999, pp. 2551–2557.

[11] M. T. Mason and K. M. Lynch, “Dynamic manipulation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 1, Jul. 1993, pp. 152–159.

[12] S. Onyshko and D. A. Winter, “A mathematical model for the dynamics of human locomotion,” *J. Biomechanics*, vol. 13, no. 4, pp. 361–368, Jan. 1980.

[13] E. Yoshida, C. Esteves, O. Kanoun, M. Poirier, A. Mallet, J.-P. Laumond, and K. Yokoi, “Planning whole-body humanoid locomotion, reaching, and manipulation,” in *Motion Planning for Humanoid Robots*. London, U.K.: Springer, 2010, pp. 99–128.

[14] T. Yoshikawa and T. Watanabe, "Dynamic control of soft-finger hands for pivoting an object in contact with the environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 1, Oct./Nov. 2000, pp. 324–329.

[15] Y. Hou, Z. Jia, and M. T. Mason, "Reorienting objects in 3D space using pivoting," 2019, *arXiv:1912.02752*. [Online]. Available: http://arxiv.org/abs/1912.02752

[16] S. S. Srinivasa, M. A. Erdmann, and M. T. Mason, "Control synthesis for dynamic contact manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 2523–2528.

[17] W. Amanhoud, M. Khoramshahi, and A. Billard, "A dynamical system approach to motion and force generation in contact tasks," in *Robotics: Science and Systems (RSS)*. Cambridge, MA, USA: MIT Press, 2019.

[18] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1711–1718, Jul. 2018.

[19] A. C. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, "A coordinate-free framework for robotic pizza tossing and catching," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 3932–3939.

[20] G. Batz, A. Yaqub, H. Wu, K. Kuhnlenz, D. Wollherr, and M. Buss, "Dynamic manipulation: Nonprehensile ball catching," in *Proc. 18th Medit. Conf. Control Autom. (MED)*, Jun. 2010, pp. 365–370.

[21] P. Cigliano, V. Lippiello, F. Ruggiero, and B. Siciliano, "Robotic ball catching with an eye-in-hand single-camera system," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1657–1671, Sep. 2015.

[22] Y. Huang, D. Xu, M. Tan, and H. Su, "Adding active learning to LWR for ping-pong playing robot," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1489–1494, Jul. 2013.

[23] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to select and generalize striking movements in robot table tennis," *Int. J. Robot. Res.*, vol. 32, no. 3, pp. 263–279, Mar. 2013.

[24] K. Harada and M. Kaneko, "Whole body manipulation," in *Proc. IEEE Int. Conf. Robot., Intell. Syst. Signal Process.*, vol. 1, Oct. 2003, pp. 190–195.

[25] M. P. Polverini, A. Laurenzi, E. M. Hoffman, F. Ruscelli, and N. G. Tsagarakis, "Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 859–866, Apr. 2020.

[26] I. G. Ramirez-Alpizar, M. Higashimori, M. Kaneko, C.-H. Tsai, and I. Kao, "Nonprehensile dynamic manipulation of a sheet-like viscoelastic object," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 5103–5108.

[27] A. Donaire, F. Ruggiero, L. R. Buonocore, V. Lippiello, and B. Siciliano, "Passivity-based control for a rolling-balancing system: The nonprehensile disk-on-disk," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 6, pp. 2135–2142, Nov. 2017.

[28] Y. Maeda, T. Nakamura, and T. Arai, "Motion planning of robot fingertips for graspless manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 3, Apr./May 2004, pp. 2951–2956.

[29] Y. Hou, Z. Jia, and M. T. Mason, "Fast planning for 3D any-pose-reorienting using pivoting," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1631–1638.

[30] S. Cruciani and C. Smith, "In-hand manipulation using three-stages open loop pivoting," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1244–1251.

[31] 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 Trans. Robot.*, vol. 35, no. 2, pp. 317–329, Apr. 2019.

[32] Y. Aiyama, M. Inaba, and H. Inoue, "Pivoting: A new method of graspless manipulation of object by robot fingers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 1, Jul. 1993, pp. 136–143.

[33] E. Yoshida, M. Poirier, J.-P. Laumond, O. Kanoun, F. Lamiraux, R. Alami, and K. Yokoi, "Pivoting based manipulation by a humanoid robot," *Auton. Robots*, vol. 28, no. 1, p. 77–78, 2010.

[34] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Feasibility evaluation of object manipulation by a humanoid robot based on recursive estimation of the object's physical properties," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4082–4089.

[35] M. Murooka, S. Nozawa, M. Bando, I. Yanokura, K. Okada, and M. Inaba, "Simultaneous planning and estimation based on physics reasoning in robot manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3137–3144.

[36] F. Shi, M. Zhao, M. Murooka, K. Okada, and M. Inaba, "Aerial regrasping: Pivoting with transformable multilink aerial robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 200–207.

[37] Y. Hou, Z. Jia, A. M. Johnson, and M. T. Mason, "Robust planar dynamic pivoting by regulating inertial and grip forces," in *Algorithmic Foundations of Robotics XII*. Cham, Switzerland: Springer, 2020, pp. 464–479.

[38] J. Testud, J. Richalet, A. Rault, and J. Papon, "Model predictive heuristic control: Applications to industial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[39] T. Geyer, G. Papafotiou, and M. Morari, "Hybrid model predictive control of the step-down DC–DC converter," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1112–1124, Nov. 2008.

[40] E. Camacho, "Hybrid model predictive control of a solar refrigeration plant: Editorial of the special section," *Eur. J. Control*, vol. 14, no. 6, pp. 448–449, 2008.

[41] W. Garcia-Gabin, D. Zambrano, and E. F. Camacho, "Sliding mode predictive control of a solar air conditioning plant," *Control Eng. Pract.*, vol. 17, no. 6, pp. 652–663, Jun. 2009.

[42] N. Slegers, J. Kyle, and M. Costello, "Nonlinear model predictive control technique for unmanned air vehicles," *J. Guid., Control, Dyn.*, vol. 29, no. 5, pp. 1179–1188, Sep. 2006.

[43] S. Faraji, S. Pouya, C. G. Atkeson, and A. J. Ijspeert, "Versatile and robust 3D walking with a simulated humanoid robot (Atlas): A model predictive control approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1943–1950.

[44] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. London, U.K.: Springer, 2009.

[45] A. Zhang, I. G. Ramirez-Alpizar, K. G. Esclasse, O. Stasse, and K. Harada, "Humanoid walking pattern generation based on model predictive control approximated with basis functions," *Adv. Robot.*, vol. 33, no. 9, pp. 454–468, May 2019.

[46] C. Neves and R. Ventura, "Energy efficient MPC for biped semi-passive locomotion," in *Proc. Robot 2015: Second Iberian Robot. Conf.* Cham, Switzerland: Springer, 2016, pp. 145–156.

[47] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Soueres, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robot. Autom. Lett.*, vol. 2, no. 1, pp. 10–17, Jan. 2017.

[48] A. K. Valenzuela, "Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain," Ph.D. dissertation, Dept. Mech. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2016.

[49] J. Z. Woodruff and K. M. Lynch, "Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4066–4073.

[50] Y. Tazaki and J.-I. Imura, "Graph based model predictive control of a planar bipedal robot," *J. Robot. Soc. Jpn.*, vol. 24, no. 5, pp. 663–671, 2006.

[51] M. Murooka, I. Kumagai, M. Morisawa, F. Kanehiro, and A. Kheddar, "Humanoid loco-manipulation planning based on graph search and reachability maps," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1840–1847, Apr. 2021.

[52] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "QpOASES: A parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, no. 4, pp. 327–363, Dec. 2014.

[53] Y. Fukumoto and K. Harada, "Force control law selection for elastic part assembly from human data and parameter optimization," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2018, pp. 1–7.

[54] N. Hogan, "Impedance control: An approach to manipulation," in *Proc. Amer. Control Conf.*, Jul. 1984, pp. 304–313.

[55] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, T. Nishi, S. Kikuchi, T. Matsubara, and K. Harada, "Learning force control for contact-rich manipulation tasks with rigid position-controlled robots," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5709–5716, Oct. 2020.

ANG ZHANG (Member, IEEE) received the B.S. degree in automation engineering from Northeastern University at Qinhuangdao, China, in 2013, and the M.S. degree in electronic engineering from Chonbuk National University, South Korea, in 2016. He is currently pursuing the Ph.D. degree in robotics with the Graduate School of Engineering Science, Osaka University. His research interests include model predictive control, nonprehensile manipulation, and bipedal walking.

KEISUKE KOYAMA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in mechanical engineering and intelligent systems from The University of Electro-Communications (UEC), in 2013, 2015, and 2017, respectively. From 2015 to 2017, he was a Research Fellow with the Japan Society for the Promotion of Science (JSPS), where he researched on high-speed proximity sensor for pre-grasping and integrated control of multi-degree-of-freedom robot hand and arm. From 2017 to 2019, he was a Project Assistant Professor with the Department of Information Physics and Computing, Graduate School of Information Science and Technology, The University of Tokyo. Since 2019, he has been an Assistant Professor with the Department of Systems Innovation, Graduate School of Engineering Science, Osaka University. He has also been a Visiting Researcher with the Department of Information Physics and Computing, Graduate School of Information Science and Technology, The University of Tokyo. His current research interests include high-speed and high precision proximity sensing for high-speed robotic manipulation and assembly.

WEIWEI WAN (Senior Member, IEEE) received the Ph.D. degree in robotics from the Department of Mechano Informatics, The University of Tokyo, Tokyo, Japan, in 2013. From 2013 to 2015, he did his postdoctoral research at Carnegie Mellon University, USA, under the support of the Japanese Society for the Promotion of Science (JSPS). From 2015 to 2017, he was a Tenure-Track Research Scientist with the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. He is currently an Associate Professor working with the Graduate School of Engineering Science, Osaka University, Japan. His research interests include robotic manipulation and smart manufacturing.

KENSUKE HARADA (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in mechanical engineering from Kyoto University, in 1992, 1994, and 1997, respectively. From 1997 to 2002, he worked as a Research Associate with Hiroshima University. Since 2002, he has been working with the National Institute of Advanced Industrial Science and Technology (AIST). From 2005 to 2006, he was a Visiting Scholar with the Department of Computer Science, Stanford University, and the Leader of the Manipulation Research Group, AIST, from 2013 to 2015. He is currently a Professor with the Graduate School of Engineering Science, Osaka University. His research interests include mechanics and control of robot manipulators and robot hands, biped locomotion, and motion planning of robotic systems.