Controlling Pivoting Gait using Graph Model Predictive Control

Ang Zhang 1, Keisuke Koyama 1, Weiwei Wan 1, Kensuke Harada 1

Abstract—Pivoting gait is efficient for manipulating a big and heavy object with relatively small manipulating force, in which a robot iteratively tilts the object, rotates it around the vertex, and then puts it down to the floor. However, pivoting gait can easily fail even with a small external disturbance due to its instability in nature. To cope with this problem, we propose a controller to robustly control the object motion during the pivoting gait by introducing two gait modes, i.e., one is the double-support mode, which can manipulate a relatively light object with faster speed, and the other is the quadruple-support mode, which can manipulate a relatively heavy object with lower speed. To control the pivoting gait, a graph model predictive control is applied taking into account of these two gait modes. By adaptively switching the gait mode according to the applied external disturbance, a robot can stably perform the pivoting gait even if the external disturbance is applied to the object.

Index Terms—Feedback control, graph search, model predictive control, pivoting manipulation, nonprehensile manipulation.

I. INTRODUCTION

Although current robots mostly manipulate objects by once picking them up [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 up. On the other hand, a human can effectively select an adequate manipulation strategy taking into account both features of the task and physical parameters of the object. For example, when a human moves a large and heavy object like furniture, a human may once tilt the object, rotates it around an axis through the contact point between the object and the floor, and then puts it down to the floor. Such manipulation style of a large object is called the pivoting manipulation. Since the manipulated object is not completely constrained by the robot, the pivoting manipulation is classified as a style of the nonprehensile manipulation [4]. So far, several different styles of nonprehensile manipulation has been proposed, such as rolling [5], pushing [6]–[8], and pivoting [9]–[11].

If pivoting manipulation involves the change of rotation vertices, it is called the pivoting gait [12] due to the correspondence between the feet of a legged robot and the vertices of a manipulated object. 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]. Though there are some research works about feedback control of pivoting manipulation [14], [15], there is no research about improving the robustness of pivoting motion by designing the gait mode.

1Department of Engineering Science, Osaka University, Osaka, Japan zhang at hlab.sys.es.osaka-u.ac.jp

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

(b) The pivoting gait runs in the DS mode in (1). A 1kg bottle is placed on the top of the object in (2). The pivoting gait in the DS mode fails after the placement in (3). A switch of gait mode to the QS mode enables the pivoting gait against the disturbance from a 2kg bottle in (4).

Fig. 1: Gait modes and the robot performs the pivoting gait.

Similar to bipedal walking, pivoting gait includes single support (SS) where the object rotates around a supporting vertex which corresponds to the pivoting manipulation, and double support (DS) where to change the supporting vertex, the object once contacts the floor with a supporting edge including two vertices. In addition, we define quadruple support (QS), where, to change its supporting vertex, the object once contacts the floor with a supporting surface including more than three vertices. The way of changing supporting vertices results in different behaviors of pivoting gait and we propose two gait modes: the DS and the QS modes. If a robot pivots an object with the QS mode, a robot can manipulate relatively heavy object with 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 with the DS mode, a robot can manipulate an object with moving relatively fast. One difficulty of controlling pivoting
gait lies in the change of contact modes.

Although adaptively using multiple gait modes can probably enable a robot to realize the robust pivoting gait, there is a challenge on how 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 in a graph. The graph selects the gait mode and outputs a reference trajectory of the object to MPC. Then, MPC tracks the reference and generates the desired position and force of the end effector. Besides, 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 the ability to switch gait modes improves the robustness of the control system and the robot can successfully achieve a stable pivoting gait under unexpected event like uncertainty in object mass and perturbation.

The contributions of this work are:

- MPC is proposed to predict the future dynamic of the object during the pivoting gait.
- Two gait modes are designed to robustly perform the pivoting gait according to different purposes, i.e., the DS mode for fast walking and the QS mode for walking stably.
- A graph MPC is developed to select the sequence of gait modes. The merit of 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 works in Section II, we formulate the pivoting gait and predictive controller in Section III. Section IV provides the design of graph. In Section V, 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, robust against external perturbations and uncertainty in object’s weight. Section VI summarizes the results and describes future work.

II. RELATED WORKS

A. Nonprehensile Manipulation

Nonprehensile manipulation attracts more and more attentions nowadays as it enables manipulating the object with fewer degrees of freedom [16]–[18] where it includes the manipulation styles like throwing [19], catching [20], [21], batting [22], [23], pushing [24], [25], sliding [26], rolling [27] and pivoting [28]–[30]. In nonprehensile manipulation, the object is manipulated without satisfying the form or force closure [31] which indicates that the object is sensitive to the environmental dynamics.

Among several styles of nonprehensile manipulation, pivoting is a promising strategy to manipulate a heavy object. Aiyama et al. [32] originally proposed the pivoting gait in which they show 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 robot to achieve pivoting task and explored simultaneous planning and estimation for humanoid pivoting tasks [35]. Shi et al. [36] proposed an aerial pivoting framework to pivot 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 swing motion.

Pivoting gait is a process of repeating pivoting manipulation around the left and right rotation vertices. We design two gait modes for pivoting gait. To best of our knowledge, there is no research about feedback control of pivoting gait with the ability to switch gait modes.

B. Model Predictive Control

Recently, MPC has been widely used in chemistry process [38], power system [39], solar technology [40], [41] and flight control [42]. In robotics research, MPC is frequently used in bipedal walking [43] as 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 et al. [48] introduce integer variables to represent the active contact modes and compute the mode sequences using mixed-integer nonlinear programming. Graph-based method also provides a way to plan a sequence of motion. Woodruff et al. [49] proposed a graph search algorithm to plan through a sequence of manipulation primitives describing different contact states. Tazaki et al. [50] proposed a graph-based MPC which draws possible modes in a graph and simulates bipedal walking at different speeds. Murooka et al. [51] proposed locomotion planning for humanoid robots based on graph search.

While this research inspires from the similarity between the pivoting gait manipulation and humanoid’s bipedal gait, they are essentially different due to the following two factors. Firstly, the formulation of pivoting gait is more complex than biped gait since a force-controlled dual-arm manipulator controls the contact mode of the grasped object. Secondly, the change of gait mode can be used to control the stability of the grasped object under gravity. This research applies the MPC to the pivoting gait where an impedance-controlled dual-arm manipulator is controlled to change the gait mode with predicting the object’s future dynamics. Different from bipedal walking, we newly design a 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 real-time control method for pivoting gait by considering two gait modes. A graph MPC is proposed to select the proper gait modes and 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:

- $\Sigma_*$: Reference frames at $* = W, B, H_i, P_i$ and $FC_i$ representing the coordinate frames attached to the ground, object, center of $i$-th spherical hand ($i = 1, 2$), contact point on the object surface with the $i$-th hand, friction cone constraint at the contact point, respectively, where $n$ and $t, o$ denote the axes in the normal and two tangent directions, respectively.
- $p_{\cdot}, R_{\cdot}, \Psi_{\cdot}, \omega_{\cdot}$: The position vector, rotation matrix, Euler angle vector and angular velocity vector of $\Sigma_*$ with respect to $\Sigma_W$, respectively, where $* = B, H_i, P_i, FC_i$ and $\Psi_* = [\phi_*, \psi_*, \theta_*]^T$.
- $X_k, Y_k, U_k$: The states, outputs and free variables in MPC taking place in the prediction horizon $n_p$ where $X_k = [x_k, \ldots, x_{k+n_p-1}]$ and the definitions of $Y_k$ and $U_k$ are similar with $X_k$.
- $p_{BP_i}$: The vector directing from the origin of $\Sigma_B$ to the origin of $\Sigma_{P_i}$.
- $f_i$: Force applied at the $i$-th contact point.
- $m_o, I_o$: object’s mass and inertia matrix.
- $g$: Gravity force vector.

For simplicity, we rewrite $p_{BP_i}$ 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

Pivoting gait is a manipulation style of an object iterating the steps where once 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

$$SD_{B0} \begin{bmatrix} \dot{p}_B \\ \omega_B \end{bmatrix} = S \begin{bmatrix} p_0 \\ \omega_0 \end{bmatrix} = 0,$$  \hspace{1cm} (1)

where $S$ is a selection matrix which selects the linear velocity and $D_{Bi}$ transforms the linear/angular velocity from $\Sigma_B$ to $\Sigma_W$.

$$S = [I_3 \hspace{0.5cm} O_3],$$  \hspace{1cm} (2)

$$D_{Bi} = \begin{bmatrix} I_3 & -[R_B \times p_{BP_i}] \end{bmatrix},$$  \hspace{1cm} (3)

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

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

$$SD_{Bi} \begin{bmatrix} \dot{p}_B \\ \omega_B \end{bmatrix} = SD_{Hi} \begin{bmatrix} \dot{p}_{Hi} \\ \omega_{Hi} \end{bmatrix} = \dot{p}_i.$$  \hspace{1cm} (4)

Given a reference trajectory of the object’s rotation, we apply the MPC to generate the motion of EEF and the target force applied by EEF, see Fig. 3. With the 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 where 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 $x = [\Psi_B^T, \omega_B^T]^T$. The relationship between angular velocity $\omega_B$ and the velocity of Euler angles $\Psi_B$ is given by

$$\dot{\Psi}_B = W \omega_B,$$  \hspace{1cm} (5)

where

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

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

$$[\Psi_B^T]_{k+1} = [I_3 \hspace{0.5cm} W \hspace{0.5cm} \Psi_B^T]_{k} + [WT^2/2 \hspace{0.5cm} T] [\omega_B^T]_{k},$$  \hspace{1cm} (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

$$r_1 \times f_1 + r_2 \times f_2 + r_{com} \times m_o g = I_\omega \dot{\omega}_B + \omega_B \times I_\omega \omega_B,$$  \hspace{1cm} (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 object’s CoG. Substituting (8) into (7) and define the force applied by EEF to be the input $u_k = [f_1 \hspace{0.5cm} f_2]^T_k$, we obtain the state equation as follows:

$$x_{k+1} = Ax_k + Bu_k + D,$$  \hspace{1cm} (9)
where \( A, B, D \) are coefficient matrices defined as,

\[
A = \begin{bmatrix} I_3 & W \\ O_3 & I_3 \end{bmatrix},
\]

\[
B = \begin{bmatrix} T_0^{-1}W_1T_2^2/2[r_1x] \\ T_0^{-1}W_1T_2^2/2[r_2x] \\ T_0^{-1}T_2[r_2x] \end{bmatrix},
\]

\[
D = \begin{bmatrix} T_0^{-1}W_1T_2^2/2[r_{com}\times]mg \\ T_0^{-1}T_2[r_{com}\times]mg \end{bmatrix}.
\]

The coefficient matrices \( B \) and \( D \), which include \( m_o \) and \( I_o \), 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 \( X_k = [x_k, x_{k+1}, \ldots, x_{k+n_p-1}]^T \) and \( U_k = [u_k, u_{k+1}, \ldots, u_{k+n_p-1}]^T \), respectively. According to (9), we have,

\[
X_{k+1} = A'X_k + B'U_k + D',
\]

where the coefficient matrices are,

\[
A' = [A, A^2, \ldots, A^{n_u}],
\]

\[
B' = \begin{bmatrix} A^0B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{n_u-1}B & \cdots & A^0B \end{bmatrix},
\]

\[
D' = \begin{bmatrix} A^0D & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{n_u-1}D & \cdots & A^0D \end{bmatrix}.
\]

**D. Output Equation**

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

\[
\dot{p}_i = SD_{Bi}[(R_bp_{Bi})\times]I_qI_4\omega_B.
\]

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

\[
\dot{p}_{Hi} = S(SD_{Hi})^T\dot{p}_i + (I_3 - S(SD_{Hi})^TSD_{Hi})k_v,
\]

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

\[
Y_k = CX_k + E_
\]

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

**E. Cost Function**

The cost function used in MPC is defined as

\[
J_{mpc} = \frac{\alpha}{2} \| X_{k+1} - X_{o}^{ref} \|^2 + \frac{\beta}{2} \| U_k \|^2,
\]

where \( \alpha, \beta \) are the weights. \( X_o^{ref} \) is the reference trajectory of the object along the prediction horizon and it is provided by the graph which we will discuss in Section [V]. 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,

\[
J_{mpc} = \frac{\alpha}{2} \| A'X_k + B'U_k + D' - X_{o}^{ref} \|^2 + \frac{\beta}{2} \| U_k \|^2.
\]

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

\[
\min_{U_k} \frac{1}{2} U_k^T Q U_k + r^T U_k + s,
\]

where

\[
Q = \frac{\alpha}{2} B'^T B' + \frac{\beta}{2} I_{2n_p},
\]

\[
r = \alpha B'^T (A'X_k + D' - X_{o}^{ref}),
\]

\[
s = \frac{\alpha}{2} (A'X_k + D' - X_{o}^{ref})^T (A'X_k + D' - X_{o}^{ref}).
\]

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

\[ f_i \in FC_i = \{ f_i : |f_i^n| \leq \mu f_i^n, \]
\[ |f_i^r| \leq \mu f_i^r, \]
\[ 0 \leq f_i^u \leq f_{\text{max}}, \]

(26)

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

\[-L_{\text{max}} \leq H_c u_k \leq L_{\text{max}},\]

(27)

where

\[ L_{\text{max}} = \begin{bmatrix} \mu f_{\text{max}}^n & \mu f_{\text{max}}^r & \mu f_{\text{max}}^n & \mu f_{\text{max}}^r \\ f_{\text{max}}^n & f_{\text{max}}^r & f_{\text{max}}^n & f_{\text{max}}^r \end{bmatrix}^T \]

(28)

denotes the bound on the normal force component and

\[ H_{ck} = \begin{bmatrix} R_{FC1}^T \\ O_3 \\ R_{FC2}^T \end{bmatrix}. \]

(29)

Considering the free variables taking place in the prediction horizon, we have

\[-L_{\text{max}}^t \leq H_{c}^t U_k \leq L_{\text{max}}^t,\]

(30)

where \( L_{\text{max}}^t \) is a matrix with \( n_p \) elements, \( L_{\text{max}}^t = [L_{\text{max}}, \cdots, L_{\text{max}}]^T \) and \( H_{c}^t = [H_{c1}, \cdots, H_{ck}] \)

G. Impedance Control

After applying MPC, we obtain the desired interaction force as an input \( u_k \) and output EEF trajectory by \( [19] \). This subsection shows how to apply the desired interaction force and the EEF trajectory to the real robot. We apply the impedance control \([53]\) where the EEF is controlled to behave like a mechanical impedance. The desired impedance can be defined as

\[ M_{\text{imp}} \ddot{x}_{H1} + D_{\text{imp}} \dot{x}_{H1} + K_{\text{imp}} (x_{H1} - x_{ref}^H_{H1}) = f_i - f_i^{ref}, \]

(31)

\[ D_{\text{imp}} \dot{x}_{H1} = f_i - f_i^{ref}, \]

(32)

where \( M_{\text{imp}}, D_{\text{imp}} \) and \( K_{\text{imp}} \) denote the mass, damping and stiffness matrices, respectively. \( x_{ref}^H_{H1} \) and \( f_i^{ref} \) indicate the reference position and force of the \( i \)-th hand. These references are provided by MPC where \( x_{ref}^H_{H1} \) is calculated by \( [19] \) and \( f_i^{ref} \) are provided by \( u_k \), the input of MPC. How to control a robot to realize the desired impedance depends on the types of joint servo controllers. In our case, we use a velocity-controlled industrial robot equipped with a function of impedance control.

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 of contact states between the object and ground while 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 and outputs a sequence of key poses as a reference to 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 where 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 QS phases coming after the left-foot SS phase, and vise versa. On the other hand, the switching motion is a transient motion from one principal motion to the other.

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

B. Weights of edges: Cost Functions

Each edge of the graph includes two pieces of information, where one is the transient time to move from one node to the other node, and the other piece is the weight. In this
subsubsection, 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 which includes $n$ edges. It is composed of a cost $J_s$ which is related to the states, $J_t$ related to time, and $J_{dst}$ related to external disturbance,

$$J_{path} = \alpha_s J_s + \beta_g J_t + \gamma_d J_{dst},$$  
(33)

where $\alpha_s, \beta_g$ and $\gamma_d$ denote the weights of each cost function. Weights can be set free under different purposes, for example, if a fast walking motion is wanted, we can increase the value of $\beta_g$. 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.

$$J_s = \sum_{i=0}^{n-1} \|x_i - x_{i+1}\|^2,$$  
(34)

where $x_i$ is the reference state of the object saved in node $i$, and the state in each node is designed with respect to the way of support, for example, single support, double support, and quadruple support. $n$ 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.

$$J_t = \sum_{i=0}^{n-1} t_i,$$  
(35)

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 for the DS mode is selected smaller than that of the QS mode because the DS mode is designed to move faster. Note that the time duration in graph between $x_k$ and $x_{k+1}$ is $t_i$, which is different from sampling time $T$ between $x_k$ and $x_{k+1}$ in MPC. A cost function of disturbance $J_{dst}$ is defined as

$$J_{dst} = \begin{cases} \delta & \text{if disturbance is detected} \\ 0 & \text{otherwise.} \end{cases}$$  
(36)

$\delta$ is a positive value indicating the existence of external disturbance.

$$\delta = \begin{cases} \delta_{ds} & \text{for DS gait mode} \\ \delta_{qs} & \text{for QS gait mode.} \end{cases}$$  
(37)

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

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

$$|\Psi^{cur} - \Psi^{pre}| \geq \Psi^{thr},$$  
(38)

$$|f_i^{cur} - f_i^{pre}| \geq f_i^{thr},$$  
(39)

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

### C. Selection of gait modes in graph

By comparing the cost function of paths, we can select an optimal path that contains information on gait modes. Fig. 6 showed the procedures of finding the optimal path: The top node is selected as a starting node, see Fig 6(a). Suppose $n = 4$ in a candidate path, 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 candidate paths, only the path with the lowest cost is remained and 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 paths in (33) will be updated because of the change of $J_{dst}$ in (36). This change can be reflected in the graph, for example, see Fig 6(c). $\gamma_d \delta_{qs}$ and $\gamma_d \delta_{ds}$ are added to the costs of the inner path and outer path, respectively. After comparing the costs of paths, the inner path is selected as the optimal path, see Fig 6(d). A transition from the outer path to the inner path implies 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 gait modes and generates a reference trajectory of the object corresponding to the selected gait modes. The reference trajectory is then provided to MPC to track. If there is a disturbance during motion, the robot will stop its motion and wait for the new command. Meanwhile, 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 MPC. Then, based on the current state and the reference, the MPC generates the trajectory of EEFs and robot moves.
V. SIMULATION AND EXPERIMENT

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 fast which involves a change of gait mode. This can be simply done by modifying weights in (33). 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 MPC is shown where the support feet are marked as blue squares, the edge connecting two support feet is drawn in red lines where the wider line indicates the QS mode and the thinner line indicates the DS mode. We can observe that gait mode changed from QS to DS after the first step. The rotation around the support foot is shown in black dashed arrows and the rotation indicates the yaw angle of the object.

We simulate the walking motion in RVIZ. In Fig 8(b) the object rotates around the vertex (we define it as 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 shows that a switch of gait mode can be realized during the motion of pivoting gait.

B. Experiment 1: DS and QS mode

We evaluate the performance of the controller design in test scenarios through experiments. Two ball-shaped EEF 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 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 find the characteristic of the gait modes, the first experiment is running pivoting gait in a single gait mode. The step length in the DS mode (Fig 11) and that in the QS mode (Fig 12) are same. In the experiment of the DS mode, Fig 9 shows the Euler angles of the object during walking and Fig 11 shows the motion. The object is firstly rotated from the initial pose (Fig 11(a)) to a DS pose (Fig 11(b) and at time 1.2 s in Fig 9). Then the object walks forward (Fig 11(c), changes support foot from right to left (Fig 11(d)) and at time 2.1 s in Fig 9), and finally goes to a rest pose in Fig 11(f).

In the experiment of the QS mode, see Fig 10 and Fig 12, robot starts to rotate the box around the right vertex and lift 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 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 I, 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 DS gait mode is achieved by selecting \( t \) in the DS mode smaller than that in QS mode. The reasons to design a quicker motion of the DS gait mode are shown below: The QS pose requires all four bottom vertices are on the floor (see the zeros in value of roll and pitch at time 3.5 s in Fig 10) while in DS pose, two vertices are on the floor and two vertices are on the fly (see the values of roll and pitch).
Fig. 8: Simulation of pivoting gait. The walking motion starts from QS mode for the first step and then switches into DS mode for 3 steps.

Fig. 9: Euler angles of the object which walks in the DS mode. At time 2.1s, only the Roll angle goes to zero which indicates the DS pose.

Table I: Time for making two steps

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

Fig. 10: Euler angles of the object which walks in the QS mode. At time 3.5s, both Roll and Pitch angles go to zero which indicates the QS pose.

pitch at time 2.1s in Fig 9. What’s more, the peak value of roll angle in QS is bigger than that of DS which also slows down the motion of QS mode. The reason for designing a rather big value of roll angle in QS is that we leave some space in roll angle to absorb perturbation so that to avoid scuffing and walk stably.

In experiment 1, motions of pivoting gait based on both the DS and QS gait mode are achieved. Pivoting gait in the DS mode is faster than that of the QS mode. Though the QS mode is slow, we believe it has a potential to keep stable walking and test the potential by following experiments.

C. Experiment 2: Uncertainty in object’s mass

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

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

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

We firstly test the pivoting gaits in the DS gait mode by putting a bottle on the manipulated object during motion. Bottles in weight of 0.35kg, 0.5kg and 1kg are placed, see Fig 13, Fig 14, and Fig 15 respectively. In all three experiments, the robot starts to pivot the object in the DS mode, see Fig 13(a), Fig 14(a), and Fig 15(a). In Fig 13(c) and Fig 15(c), the bottle in weight of 0.35kg, 0.5kg and 1kg is placed on the top of the box, respectively, during the walking motions which results in a change of the object 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 placements of bottles in weight of 0.35kg and 0.5kg. In the case of placing the 1kg bottle, the object lands in a DS pose after the placement of bottle, see Fig. 15(d). However, the robot fails to lift the object after switching the rotation vertex from the right to the left. What’s more, unfortunately, a collision is found between the table and the font right foot of the object, see the red rectangle in Fig. 15(e).

Since the failure arises during the switching of rotation vertices, we improve pivoting gait by realizing the switching gait mode to the QS mode where 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 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 firstly manipulates the object in the DS mode, see Fig. 19(a) and Fig. 19(b). A 2kg bottle is placed on the top of the box, see Fig. 19(c). The placement is detected by the force sensors in the wrist of the robot and the graph switch the gait mode from the DS mode to the QS mode. After the placement, the object lands in QS pose in Fig. 19(d). Then the robot pivots the object to walk for two more steps in QS mode and successfully finishes the pivoting gait, see Fig. 19(e)-19(h). This experiment shows that 1. The QS mode is more stable than the DS mode when disturbance occurs. 2. Switching of gait mode improves the robustness of the control system.

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

### Experiment 3: External perturbation

In pivoting gait, if the robot continues manipulation without feedback, the error of the position of the object will accumulate. As a solution to these problem, we use 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 visual system and 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 put in front of the table, which are shown in the red rectangle in Fig. 20(a). 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(b)-20(d). An unexpected push is acted to the box during walking by a metal stick, see Fig. 20(e). 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 QS mode is selected as gait mode because it provides stable walking. In
Fig. 13: Experiment of placing a 0.35kg bottle on the manipulated object during the motion of pivoting, the robot successfully pivots the object in the DS mode.

Fig. 14: Experiment of placing a 0.5kg bottle on the manipulated object during the motion of pivoting, the robot successfully pivots the object in the DS mode.

Fig. 15: Experiment of placing a 1kg 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).

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

Fig. 18 to 20: The new motion, the robot firstly lifts the box into a SS pose which is to avoid the box from scuffing the table, see Fig. 18 from time 12.4-14.4s and Fig. 20(g). Then, the robot lands the box into QS pose to firmly contact with the table, see Fig. 18 at time 24.2s and Fig. 20(h). After landing, the robot pivots the object to walk for two more steps, Fig. 20(i)-20(l).

The 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 the object to walk. Two gait modes are designed for adapting to an environment where the DS mode for fast walking and the QS mode for stable walking. Vision systems and force sensors are implemented to perceive the environment. A graph MPC is proposed where the graph selects the gait mode based on the information collected from both force sensors and the vision system and outputs a reference trajectory which is tracked by the MPC.

Experiments show that the gait mode influences the robustness of the system, especially when disturbance occurs.
Fig. 17: Footprints of experiment 3. A perturbation is detected by the vision system and shown by the red dash line which is close to the first step. Though perturbation occurs, the MPC still finds a solution to follow the planned trajectory.

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

The DS gait mode is fast while the QS mode is stable way because it provides a form contact between the object and the table. The proposed graph MPC can select gait mode adaptively and control the pivoting gait in real-time while being free to switch gait modes, robust against external perturbations and uncertainty in the object’s weight.

The extension is to apply our pivoting approach to objects in different shapes. It requires identifying appropriate model, selection of supporting vertices, design of grasp positions, and design of new gait modes.

REFERENCES

[1] Lars Berscheid, Pascal Meißner, and Torsten Kröger. Self-supervised learning for precise pick-and-place without object model. IEEE Robotics and Automation Letters, 5(3):4828–4835, 2020.

[2] Marcus Gualtieri, Andreas ten Pas, and Robert Platt. Pick and place without geometric object models. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 7433–7440. IEEE, 2018.

[3] Jingren Xu, Kensuke Harada, Weiwei Tan, Toshio Ueshiba, and Yukiyasu Domae. Planning an efficient and robust base sequence for a mobile manipulator performing multiple pick-and-place tasks. arXiv preprint arXiv:2001.08042, 2020.

[4] Kevin M Lynch and Matthew T Mason. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. The International Journal of Robotics Research, 18(1):64–92, 1999.

[5] Zexiang Li and John Canny. Motion of two rigid bodies with rolling constraint. IEEE Transactions on Robotics and Automation, 6(1):62–72, 1990.

[6] Kevin M Lynch and Matthew T Mason. Stable pushing: Mechanics, controllability, and planning. The International Journal of Robotics Research, 15(6):533–556, 1996.

[7] Kensuke Harada, Shinji Kagita, Fumio Kanohe, Kiyoshi Fujisawa, Kenji Kaneko, and Hiroshi Hirukawa. Real-time planning of humanoid robot’s gait for force-controlled manipulation. IEEE/ASME Transactions on Mechatronics, 12(1):53–62, 2007.

[8] Jiaju Zhou, Yifan Hou, and Matthew T Mason. Pushing revisited: Differential flatness, trajectory planning, and stabilization. The International Journal of Robotics Research, 38(12-13):1477–1489, 2019.

[9] Neel Doshi, Francois R Hogan, and Alberto Rodriguez. Hybrid differential dynamic programming for planar manipulation primitive. arXiv preprint arXiv:1911.00175, 2019.

[10] Kevin M Lynch. Toppling manipulation. In Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C), volume 4, pages 2551–2557. IEEE, 1999.

[11] Matthew T Mason and Kevin M Lynch. Dynamic manipulation. In Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’93), volume 1, pages 152–159. IEEE, 1993.

[12] S. ChayUSHI and DA Winter. A mathematical model for the dynamics of human locomotion. Journal of biomechanics, 13(4):361–368, 1980.

[13] Eichi Yoshiida, Claudia Esteves, OuSSama Kanoun, Mathieu Pourier, Anthony Mallet, Jean-Paul Laumond, and Kazuhiro Yokoi. Planning whole-body humanoid locomotion, reaching, and manipulation. In Motion planning for humanoid robots, pages 99–128. Springer, 2010.

[14] Tsuneo Yosihara and Tetsuyuki Watanabe. Dynamic control of soft fingers for pivoting an object in contact with the environment. In Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113), volume 1, pages 324–329. IEEE, 2000.

[15] Yifan Hou, Zhenzhong Jia, and Matthew T Mason. Reorienting objects in 3d space using pivoting. arXiv preprint arXiv:1912.02752, 2019.

[16] Sudharrtha S Sarma, Michael A Erdmann, and Matthew T Mason. Control synthesis for nonprehensile contact manipulation. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pages 2523–2528. IEEE, 2005.

[17] Wald Amanhoud, Mahdi Khoramshahi, and Aude Billard. A dynamical system approach to motion and force generation in contact tasks. Robotics: Science and Systems (RSS), 2019.

[18] Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. Nonprehensile dynamic manipulation: A survey. IEEE Robotics and Automation Letters, 3(3):1711–1718, 2018.

[19] Aykut CÖSUNCU, Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. A coordinate-free framework for robotic pizza tossing and catching. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 3932–3939. IEEE, 2016.

[20] Georg Bätz, Arhan Yaqub, Huiyan Wu, Kolja Kühlmenz, Dirk Wollherr, and Martin Buss. Dynamic manipulation: Nonprehensile ball catching. In 18th Mediterranean Conference on Control and Automation, MED’10, pages 365–370. IEEE, 2010.

[21] Pierluigi Cigliano, Vincenzo Lippiello, Fabio Ruggiero, and Bruno Siciliano. Robotic ball catching with an eye-in-hand single-camera system. IEEE Transactions on Control Systems Technology, 23(5):1657–1671, 2015.

[22] Yanlong Huang, De Xu, Min Tan, and Hu Su. Adding active learning to lwr for ping-pong playing robot. IEEE Transactions on Control Systems Technology, 21(4):1489–1494, 2012.

[23] Katharina Mülling, Jens Kober, Oliver Kromer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. The International Journal of Robotics Research, 32(3):263–279, 2013.
Fig. 19: Experiment of placing a 2kg bottle on the manipulated object during the motion of pivoting. A 2kg bottle is put on the top of the box during walking, see Fig.19(c). After the placement, the robot changes gait mode from DS to the QS mode(Fig.19(d)) and pivots the box to walk.

Fig. 20: Cameras are used to track the motion of the box, see the red rectangle in Fig.20(a). An unexpected push is acted to the box at Fig.20(e). The robot pivots the object to walk and makes reactions to resist disturbance.
Nathan Slegers, Jason Kyle, and Mark Costello. Nonlinear model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, 1978.

Tobias Geyer, Georgios Papafotiou, and Manfred Morari. Hybrid model predictive control of the step-down dc–dc converter. *IEEE Transactions on Control Systems Technology*, 16(6):1122–1124, 2008.

E Camacho. Hybrid model predictive control of a solar refrigeration plant: Editorial of the special section. *European Journal of Control*, 14(6):448–449, 2008.

Winston Garcia-Gabin, Darine Zambrano, and Eduardo F Camacho. Sliding mode predictive control of a solar air conditioning plant. *Control Engineering Practice*, 17(6):652–663, 2009.

Nathan Slegers, Jason Kyle, and Mark Costello. Nonlinear model predictive control technique for unmanned air vehicles. *Journal of guidance, control, and dynamics*, 29(5):1179–1188, 2006.

Salman Faraji, Soha Pouya, Christopher G Atkeson, and Auke Jan Ijspeert. Versatile and robust 3d walking with a simulated humanoid robot (atlas): A model predictive control approach. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1943–1950. IEEE, 2014.

Liuping Wang. *Model predictive control system design and implementation using MATLAB*. Springer Science & Business Media, 2009.

Ang Zhang, Ischel G Ramirez-Alpizar, Kevin Giraud Elsasse, Olivier Stasse, and Kensuke Harada. Humanoid walking pattern generation based on model predictive control approximated with basis functions. *Advanced Robotics*, 33(9):454–468, 2019.

C Neves and Rodrigo Ventura. Energy efficient mpc for biped semi-passive locomotion. In *Robot 2015: Second Iberian Robotics Conference*, pages 145–156. Springer, 2016.

Maximilien Naveau, Manuel Kudruss, Olivier Stasse, Christian Kirches, Katja Mombaur, and Philippe Souères. A reactive walking pattern generator based on nonlinear model predictive control. *IEEE Robotics and Automation Letters*, 2(1):10–17, 2016.

Andrés Klee Valenzuela. Mixed-integral convex optimization for planning aggressive motions of legged robots over rough terrain. PhD thesis, Massachusetts Institute of Technology, 2016.

J Zachary Woodruff and Kevin M Lynch. Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4066–4073. IEEE, 2017.

Yuichi Tazaki and Jun-ichi Imura. Graph based model predictive control of a planar bipedal robot. *Journal of the Robotics Society of Japan*, 24(3):663–671, 2006.

Masaki Murooka, Iori Kumagai, Mitsuharu Morisawa, Fumio Kanehiro, and Abderrahmane Kheddar. Humanoid loco-manipulation planning based on graph search and reachability maps. *IEEE Robotics and Automation Letters*, 6(2):1840–1847, 2021.

H.J. Ferrez, C. Kirchler, A. Potshka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

Yasuhiro Fukumoto and Kensuke Harada. Force control law selection for elastic part assembly from human data and parameter optimization. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–7. IEEE, 2018.

Neville Hogan. Impedance control: An approach to manipulation. In *1984 American control conference*, pages 304–313. IEEE, 1984.

Oussama Koutmane, Florent Lamiraux, Rachid Alami, and Kazuhito Yokoi. Pivoting: In-hand manipulation using three-fingered grasps. In *Proceedings of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 200–207. IEEE, 2017.

Chia-Hung Tsai, and Imin Kao. Nonprehensile dynamic manipulation of a sheet-like viscoelastic object. In *2011 IEEE International Conference on Robotics and Automation*, pages 5103–5108. IEEE, 2011.