

A Symplectic Instantaneous Optimal Control for Robot Trajectory Tracking with Differential-Algebraic Equation Models

Haijun Peng, Fei Li, Jinguo Liu, *Senior Member, IEEE*,

and Zhaojie Ju, *Senior Member, IEEE*

Abstract—Robot trajectory tracking control based on differential-algebraic equation (DAE) models is still a thorny issue, because the DAEs of such systems are inherently complex and unstable, such as the high index problem. In this paper, based on controlled DAEs, a symplectic instantaneous optimal control (IOC) method for robot trajectory tracking with input saturation is proposed. Based on the discrete variational principle and the canonical transformation, a symplectic discretization form for the controlled DAEs is first constructed. Then, the continuous trajectory tracking problem is approximated for a series of IOC problems at every time step, and the linear complementarity problem (LCP) can be derived for solving the IOC problems. Finally, the control inputs can be obtained by solving the corresponding standard LCP. The proposed method provides a unified framework for solving the trajectory tracking control problems of robot multibody dynamic systems. Numerical simulations and virtual experiments are conducted to verify the robustness and the efficiency of the proposed method, i.e., the input saturation

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. This work was supported in part by National Science Foundation of China (Grant No.11772074, 11761131005, 91748203), National Key R&D Program of China (Grant No. 2018YFB1304600), the Natural Science Foundation of China (Grant No. 51775541), and CAS Interdisciplinary Innovation Team (Grant No. JCTD-2018-11)

constraints are satisfied at the discrete time points, and high accuracy tracking control results can be obtained at low computational cost.

***Index Terms*—Symplectic, instantaneous optimal control, input saturation, robot trajectory tracking, differential-algebraic equation, linear complementarity problem.**

I. INTRODUCTION

WITH wide robotics applications in various fields, new requirements for the dynamics and control performance of robots have been increasingly proposed. Trajectory tracking is one of the typical control problems for robots [1]. In the past few decades, despite various control methods for trajectory tracking problems, most existing methods are established for several specific robots based on the description of ordinary differential equations (ODEs) [2], [3]. Therefore, independent generalized coordinates must be found first to represent the positions and orientations for the robot's objects, and then different controlled dynamic models are built for different types of robots by complex and tedious symbolic operations. This modeling process is difficult, especially in much more complex robotic systems. In fact, a robot is essentially a multibody dynamic system. Thus, a unified multibody dynamic model for a robot can be built based on differential-algebraic equations (DAEs) with nonindependent generalized coordinates. However, robot dynamic systems described by DAEs will bring new challenges for the trajectory tracking control problems. The challenge primarily comes from the fact that numerical integration of the dynamical equations is a difficult problem because of the inherent instability due to the high-index problem for DAEs [4]. Thus, the optimal control law for such DAE systems is difficult to achieve. The aim of this paper is to provide a novel approach to solve the robot trajectory tracking control problem introduced by DAEs.

Numerical methods for DAEs, including both index reduction approaches [5], [6] and direct discrete approaches [7], are popular, but they may be confronted with the issues of constraint violation and computational stability. In other words, the kinematic constraints are not easy to be satisfied, and the system

often “drifts” with time integration. In addition, for many direct discrete approaches, stability of the integration process has been proven mathematically in linear systems but not in nonlinear systems [8]. Thus, to overcome the difficulties associated with the index-3 DAEs, some geometric integration schemes [9]–[11] have been developed. The geometric integration approaches, normally constructed in the Hamilton mechanics framework, are proven to be symplectically preserved. The most fundamental property of the Hamiltonian systems is that the phase flow is a symplectic transformation [12]. Numerical methods preserving the symplectic structure can obtain good performances in the optimal control problem [13]. It is noted that symplectic algorithms can be constructed by appropriate difference schemes. Feng and Qin did some pioneering work and introduced the symplectic geometric theory in [10]. Marsden et al. [11], [13] have developed discrete mechanics and optimal control for constrained systems by the Lagrange-d’Alembert principle, which is a structure-preserving scheme. Peng and Gao et al. [9], [14] have proposed four types of generating functions that allow a symplectic algorithm to be constructed. Hu et al. [15] have proposed the generalized multisymplectic scheme based on the multisymplectic idea, which satisfies the discrete local momentum conservation law exactly. However, the above symplectic algorithm references are mainly applied to deal with ODEs or partial differential equations. In fact, the symplectic idea can be adopted and extended for the index-3 DAEs. Therefore, inspired by the above works, a symplectic method for solving the DAEs of a controlled robot system will be developed in this paper.

To address the robot trajectory tracking control, research efforts have been focused on a variety of control approaches based on several kinds of control design criteria. These approaches include proportional-integral-derivative (PID) control [16], optimal control [17], predictive control [18]–[20], adaptive control [21] and neural network control [22]. These approaches are proven to be valid for a specific controlled multibody system, but they usually do not respect the invariants such as the symplectic structure and energy preserving in a nonlinear dynamic system, which are very important factors for keeping the long-time numerical stability [23]. At present, most of the symplectic-preserving control algorithms are based on the ODEs. Nevertheless, for the DAEs, symplectic algorithms have attracted enormous research

attention, and several new algorithms have been developed. However, a symplectic method of trajectory tracking control for a robot described by DAEs is still very limited.

Thus, in the present work, based on the theory of symplectic algorithms and instantaneous optimal control (IOC), a symplectic IOC method of trajectory tracking control for a robot described in DAEs is first proposed; this follows the work from Yang et al., who introduced the IOC method as it was applied to control seismically excited structures [24]. The performance index for the IOC method is a quadratic function of the responses and the control inputs. It is applicable to control linear, nonlinear and hysteretic structural systems [25]. The structural dynamic system is typically depicted by ODEs. However, the robot multibody dynamic system depicted by DAEs is more complicated because of the high index problem. Furthermore, the DAE kinematic constraints over the entire time horizon will also make it difficult to obtain an optimal control law. To overcome these difficulties in this paper, the original problem will be transformed into a sequence of discrete sub-optimal control problems in every instantaneous time slot with variational and symplectic discretization forms. The instantaneous optimal control inputs can then be obtained by solving such suboptimization problems.

In addition, one of the most important practical problems related to robot trajectory tracking is the control input saturation [26]. In this work, the control input constraints are considered by extending the performance index of the IOC method with Lagrange multipliers, then the input saturation problem is transformed into a linear complementarity problem (LCP), and the Lagrange multipliers can be obtained by a standard LCP solver.

Therefore, the main contributions of this paper are summarized as follows:

- 1) Most existing methods for the trajectory tracking control are based on ODEs, while the proposed method is based on DAEs. Thus, an important advantage of the proposed method is that it provides a unified framework for solving more general tracking control problems of different types of robots.
- 2) The proposed method has merit since the numerical solution at discrete time points can be proven to be symplectically preserved.

The remainder of this paper is organized as follows. In section II, the formulation of the trajectory tracking control problem for a robot based on DAE models with input saturation are introduced. Then, in section III, the controlled robot multibody dynamic system is discretized by a symplectic method. In section IV, based on the theory of IOC and the parametric variational principle, a symplectic IOC method for the robot trajectory tracking multibody dynamic system with input saturation is proposed with details. Subsequently, the simulation results for two types of robots are discussed to demonstrate the effectiveness of the proposed method in section V. Finally, some concluding remarks of this paper are provided in section VI.

II. PROBLEM FORMULATION

First, considering a controlled robot multibody dynamic system that is described by the differential-algebraic equations (DAEs), the basic formulations can be written as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{\Phi}_q^T(\mathbf{q}, t)\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{A}(\mathbf{q})\mathbf{u}(t) \quad (1)$$

$$\mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0} \quad (2)$$

where (1) represents the dynamics of the mechanical system, and (2) represents the kinematic constraints. The vectors $\mathbf{q} \in R^{n \times 1}$, $\dot{\mathbf{q}} \in R^{n \times 1}$ and $\ddot{\mathbf{q}} \in R^{n \times 1}$ denote the generalized coordinates, velocities and accelerations, respectively. $\mathbf{M} \in R^{n \times n}$ is the system mass matrix, $\boldsymbol{\lambda} \in R^{s \times 1}$ is the vector of Lagrange multipliers, and the vector of apparent force $\mathbf{Q} \in R^{n \times 1}$ includes the external, elastic, Coriolis and centrifugal forces. $\mathbf{\Phi} \in R^{s \times 1}$ is the vector of the constraint functions, and $\mathbf{\Phi}_q \in R^{s \times n}$ is the Jacobian matrix of the kinematic constraint equations. The input matrix $\mathbf{A} \in R^{n \times m}$ distributes the m control inputs \mathbf{u} onto the directions of the system generalized coordinates, whereby \mathbf{A} is often a constant matrix. Here it should be noted that for some universal modeling strategies, coordinate transformations may be performed to calculate the contribution of \mathbf{u} to the system's generalized active force; thus, matrix \mathbf{A} will be related to the generalized coordinates \mathbf{q} , which will be reflected in the next section. In addition, n is the number of generalized degree of freedom, and s is the number of holonomic constraints.

Then, the trajectory tracking control problem for the robot multibody dynamic system can be defined. The

goal is to find the optimal control inputs $\mathbf{u}(t)$ with saturation to minimize the performance index J . Based on the DAEs, the formulations can be given as

$$\begin{cases} \text{find } \mathbf{u}(t) \\ \text{min. } J = \int_{t_0}^{t_f} \left[(\mathbf{y}(q,t) - \tilde{\mathbf{y}}(t))^T \cdot \hat{\mathbf{Q}} \cdot (\mathbf{y}(q,t) - \tilde{\mathbf{y}}(t)) + \mathbf{u}^T(t) \cdot \hat{\mathbf{R}} \cdot \mathbf{u}(t) \right] dt \\ \text{s.t. } \mathbf{M}(q) \cdot \ddot{q} + \Phi_q^T(q,t) \cdot \lambda - \mathbf{Q}(q, \dot{q}, t) - \mathbf{A}(q) \cdot \mathbf{u}(t) = \mathbf{0} \\ \Phi(q,t) = \mathbf{0} \\ \mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max} \end{cases} \quad (3)$$

where t_0, t_f are given initial and terminal times, respectively. The operator $\mathbf{y}(q,t)$ computes the outputs of the dynamic system. The function $\tilde{\mathbf{y}}(t)$ represents the known desired path, and it is to be tracked. A typical example for the system outputs is the position coordinates of the end-effector of a manipulator. $\hat{\mathbf{Q}} \in R^{r \times r}$ and $\hat{\mathbf{R}} \in R^{m \times m}$ are the nonnegative and positive definite symmetric weighting matrices, respectively. The control inputs $\mathbf{u}(t)$ are not free but are constrained as the inequality formula in (3). In addition, r is the number of the generalized output coordinates for the control.

So far, the robot tracking control problem described in index-3 DAEs has been given, as shown in (3).

III. DISCRETIZATION OF THE ROBOT CONTROLLED DAEs BY A SYMPLECTIC METHOD

In this section, based on the discrete variational principle and the requirements of a canonical transformation [11], [14], a symplectic numerical algorithm is first developed to discretize the controlled DAEs of the robot multibody system.

A. Discretization by a Symplectic Method

First, to construct the symplectic method, the apparent force \mathbf{Q} in (1) needs to be derived by the following forced Lagrange equation

$$\frac{d}{dt} \left(\frac{\partial L^c(q, \dot{q})}{\partial \dot{q}} \right) - \frac{\partial L^c(q, \dot{q})}{\partial q} = \mathbf{Q}_a(q, \dot{q}, t) + \mathbf{A}(q) \mathbf{u}(t) \quad (4)$$

where \mathbf{Q}_a is the external force and L^c is the augmented Lagrangian function for constrained system, which can be defined as

$$L^c(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} - U(\mathbf{q}) - \Phi^T(\mathbf{q}) \lambda \quad (5)$$

Here, it should be noted that both the gravity and deformation elastic potential energy are contained in $U(\mathbf{q})$. Then, substituting (5) into (4), the dynamic equation for the robot multibody system can be derived and is given as

$$\begin{aligned} \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}) \lambda = & \mathbf{Q}_a(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{A}(\mathbf{q}) \mathbf{u}(t) - U_q(\mathbf{q}) \\ & + \frac{1}{2} \frac{\partial(\dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}))}{\partial \mathbf{q}} \dot{\mathbf{q}} - \frac{\partial(\mathbf{M}(\mathbf{q}) \dot{\mathbf{q}})}{\partial \mathbf{q}} \dot{\mathbf{q}} \end{aligned} \quad (6)$$

By comparing (6) and the standard form of (1), the external force can be obtained:

$$\begin{aligned} \mathbf{Q}_a(\mathbf{q}, \dot{\mathbf{q}}, t) = & \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) + U_q(\mathbf{q}) \\ & - \frac{1}{2} \frac{\partial(\dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}))}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial(\mathbf{M}(\mathbf{q}) \dot{\mathbf{q}})}{\partial \mathbf{q}} \dot{\mathbf{q}} \end{aligned} \quad (7)$$

Second, in order to carry out numerical integrations, we assume that the whole continuous time domain T is divided into N time intervals $[t_k, t_{k+1}]$, $0 \leq k \leq N$ with equal time step length $\eta = T/N$. Then, from the Lagrange-d'Alembert principle, the action variation in $[t_k, t_{k+1}]$ can be defined as

$$\delta S^*(\mathbf{q}, \dot{\mathbf{q}}) = \delta \int_{t_k}^{t_{k+1}} L^c dt + \int_{t_k}^{t_{k+1}} \delta \mathbf{q}^T [\mathbf{Q}_a(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{A}(\mathbf{q}) \mathbf{u}(t)] dt = 0 \quad (8)$$

where the δ represents variations vanishing at the endpoints.

Then, using linear interpolation technology to discrete the generalized coordinates, the interpolation approximation of \mathbf{q} and $\dot{\mathbf{q}}$ in the time interval (t_k, t_{k+1}) can be given as

$$\mathbf{q} = (1 - \alpha) \mathbf{q}_k + \alpha \mathbf{q}_{k+1}, \quad \dot{\mathbf{q}} = (\mathbf{q}_{k+1} - \mathbf{q}_k) / \eta \quad (9)$$

where \mathbf{q}_k and \mathbf{q}_{k+1} denote the generalized coordinates at two ends of the time steps, and the coefficient $\alpha \in [0, 1]$, respectively. In this paper, let $\alpha = 0.5$. In addition, assume that the control inputs $\mathbf{u}(t)$ keep constant between any adjacent sampling points by using a zero-order holder. In other words, the control item $\mathbf{A}(\mathbf{q}) \mathbf{u}(t)$ of (8) in the time interval $[t_k, t_{k+1}]$ is approximately equal to $\mathbf{A}(\mathbf{q}_k) \mathbf{u}_{k+1}$. Then substituting (9) into (8), we have

$$\delta S^* = \delta \int_{t_k}^{t_{k+1}} L^c(\tilde{\mathbf{q}}, \tilde{\mathbf{v}}) dt + \int_{t_k}^{t_{k+1}} \delta \tilde{\mathbf{q}}^T [\mathbf{Q}_a(\tilde{\mathbf{q}}, \tilde{\mathbf{v}}, t) + \mathbf{A}(\mathbf{q}_k) \mathbf{u}_{k+1}] dt = 0 \quad (10)$$

where

$$\tilde{\mathbf{q}} = (\mathbf{q}_k + \mathbf{q}_{k+1})/2, \quad \tilde{\mathbf{v}} = (\mathbf{q}_{k+1} - \mathbf{q}_k)/\eta \quad (11)$$

It should be noted that according to the discrete variational principle, the action S^* , which only depends on \mathbf{q}_k and \mathbf{q}_{k+1} , can be chosen as the first kind of generating function [14] for the discrete dynamical system. Its corresponding canonical transformation between the states of the time steps can be derived as follows

$$\mathbf{p}_k = -\frac{\partial S^*(\mathbf{q}_k, \mathbf{q}_{k+1})}{\partial \mathbf{q}_k}, \quad \mathbf{p}_{k+1} = \frac{\partial S^*(\mathbf{q}_k, \mathbf{q}_{k+1})}{\partial \mathbf{q}_{k+1}} \quad (12)$$

where \mathbf{p}_k and \mathbf{p}_{k+1} are the dual variables at two ends of the time step, which actually are the momentums at time t_k and t_{k+1} , respectively, i.e., $\mathbf{p}_k = \mathbf{M}(\mathbf{q}_k)\dot{\mathbf{q}}_k$, $\mathbf{p}_{k+1} = \mathbf{M}(\mathbf{q}_{k+1})\dot{\mathbf{q}}_{k+1}$.

Finally, substituting (5), (7) and (10) into (12), the discrete forms of (1) can be constructed and given as

$$\hat{\mathbf{f}}_1 = \mathbf{f}_1 + 0.5\eta\mathbf{A}(\mathbf{q}_k)\mathbf{u}_{k+1} = \mathbf{0} \quad (13)$$

$$\hat{\mathbf{f}}_2 = \mathbf{f}_2 + 0.5\eta\mathbf{A}(\mathbf{q}_k)\mathbf{u}_{k+1} = \mathbf{0} \quad (14)$$

where

$$\begin{aligned} \mathbf{f}_1 &= \mathbf{p}_k - \mathbf{M}(\tilde{\mathbf{q}}) \cdot \tilde{\mathbf{v}} + \frac{\eta}{2} \cdot \mathbf{G}(\tilde{\mathbf{q}}, \tilde{\mathbf{v}}) \cdot \tilde{\mathbf{v}} \\ &\quad - \frac{\eta}{2} \cdot \Phi_q^T(\tilde{\mathbf{q}}) \cdot \lambda_{k+1} + \frac{\eta}{2} \cdot \mathbf{Q}(\tilde{\mathbf{q}}, \tilde{\mathbf{v}}) = \mathbf{0} \end{aligned} \quad (15)$$

$$\begin{aligned} \mathbf{f}_2 &= -\mathbf{p}_{k+1} + \mathbf{M}(\tilde{\mathbf{q}}) \cdot \tilde{\mathbf{v}} + \frac{\eta}{2} \cdot \mathbf{G}(\tilde{\mathbf{q}}, \tilde{\mathbf{v}}) \cdot \tilde{\mathbf{v}} \\ &\quad - \frac{\eta}{2} \cdot \Phi_q^T(\tilde{\mathbf{q}}) \cdot \lambda_{k+1} + \frac{\eta}{2} \cdot \mathbf{Q}(\tilde{\mathbf{q}}, \tilde{\mathbf{v}}) = \mathbf{0} \end{aligned} \quad (16)$$

$$\mathbf{G}(\tilde{\mathbf{q}}, \tilde{\mathbf{v}}) = \left. \frac{\partial (\mathbf{M}(\mathbf{q})\dot{\mathbf{q}})}{\partial \mathbf{q}} \right|_{\mathbf{q}=\tilde{\mathbf{q}}, \dot{\mathbf{q}}=\tilde{\mathbf{v}}} \quad (17)$$

In addition, the kinematic constraints (2) should be satisfied at every discrete time node; therefore,

$$\mathbf{f}_3 = \Phi(\mathbf{q}_{k+1}, t_{k+1}) = \mathbf{0} \quad (18)$$

So far, the controlled DAEs have been discretized, as shown in (13), (14) and (18). Here, it should be noted that (12) derived by the canonical transformation is symplectically preserved in the absence of external forces, and the proof of preservation for the symplectic property will be given in the next subsection.

B. Preservation of Symplectic Property

For numerical algorithms, assume that the relationship of solution vectors on two discrete times is

$$\bar{\mathbf{z}}_{j+1} = \phi(\bar{\mathbf{z}}_j) \quad (19)$$

where $\bar{\mathbf{z}}_j = \{\mathbf{q}_j^\top, \mathbf{p}_j^\top\}^\top$, and ϕ is a mapping function in the state space. A numerical algorithm is symplectic-preserving if and only if the Jacobi matrix of the mapping function ϕ is a symplectic matrix [12].

The Jacobi matrix of the mapping function ϕ can be given by

$$\bar{\mathbf{S}} = \partial \bar{\mathbf{z}}_{k+1} / \partial \bar{\mathbf{z}}_k \quad (20)$$

Therefore, if the matrix $\bar{\mathbf{S}}$ satisfies

$$\bar{\mathbf{S}}^\top \cdot \bar{\mathbf{J}} \cdot \bar{\mathbf{S}} = \bar{\mathbf{J}}, \quad \bar{\mathbf{J}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \quad (21)$$

then the corresponding mapping $\phi: \bar{\mathbf{z}}_j \mapsto \bar{\mathbf{z}}_{j+1}$ is symplectic.

Remark 1: To the best of the authors' knowledge, the current research on the symplectic-preserving method is almost about conservative systems [9], [14]. But in this work, the symplectic property will be extended into nonconservative systems.

Theorem: Assuming there is no velocity item in the external forces of a nonconservative system, the discretization of (12) derived via the discrete Lagrange-d'Alembert principle is symplectic.

Proof: The specific expressions for (12) in the absence of velocity item in external forces can be derived as follows

$$\begin{aligned} \bar{\mathbf{f}}_1 &= \mathbf{p}_k - \mathbf{M}(\tilde{\mathbf{q}}) \cdot \tilde{\mathbf{v}} + \frac{\eta}{4} \cdot \left[\dot{\mathbf{q}}^\top \frac{\partial \mathbf{M}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} \right]_{\mathbf{q}=\tilde{\mathbf{q}}, \dot{\mathbf{q}}=\tilde{\mathbf{v}}} - \frac{\eta}{2} \cdot \mathbf{U}_q(\tilde{\mathbf{q}}) \\ &\quad - \frac{\eta}{2} \cdot \Phi_q^\top(\tilde{\mathbf{q}}) \cdot \boldsymbol{\lambda}_{k+1} + \frac{\eta}{2} \cdot \mathbf{Q}_a(\tilde{\mathbf{q}}, t) + \frac{\eta}{2} \cdot \mathbf{A}(\mathbf{q}_k) \mathbf{u}_{k+1} = \mathbf{0} \end{aligned} \quad (22)$$

$$\begin{aligned} \bar{\mathbf{f}}_2 &= \mathbf{p}_{k+1} - \mathbf{M}(\tilde{\mathbf{q}}) \cdot \tilde{\mathbf{v}} - \frac{\eta}{4} \cdot \left[\dot{\mathbf{q}}^\top \frac{\partial \mathbf{M}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} \right]_{\mathbf{q}=\tilde{\mathbf{q}}, \dot{\mathbf{q}}=\tilde{\mathbf{v}}} + \frac{\eta}{2} \cdot \mathbf{U}_q(\tilde{\mathbf{q}}) \\ &\quad + \frac{\eta}{2} \cdot \Phi_q^\top(\tilde{\mathbf{q}}) \cdot \boldsymbol{\lambda}_{k+1} - \frac{\eta}{2} \cdot \mathbf{Q}_a(\tilde{\mathbf{q}}, t) - \frac{\eta}{2} \cdot \mathbf{A}(\mathbf{q}_k) \mathbf{u}_{k+1} = \mathbf{0} \end{aligned} \quad (23)$$

The matrix $\bar{\mathbf{S}}$ can then be obtained, given by

$$\bar{\mathbf{S}} = \frac{\partial \bar{\mathbf{z}}_{k+1}}{\partial \bar{\mathbf{z}}_k} = \frac{\partial \bar{\mathbf{z}}_{k+1}}{\partial \bar{\mathbf{f}}} \cdot \frac{\partial \bar{\mathbf{f}}}{\partial \bar{\mathbf{z}}_k} = \begin{bmatrix} \frac{\partial \bar{\mathbf{f}}_1}{\partial \mathbf{q}_{k+1}} & \frac{\partial \bar{\mathbf{f}}_1}{\partial \mathbf{p}_{k+1}} \\ \frac{\partial \bar{\mathbf{f}}_2}{\partial \mathbf{q}_{k+1}} & \frac{\partial \bar{\mathbf{f}}_2}{\partial \mathbf{p}_{k+1}} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \frac{\partial \bar{\mathbf{f}}_1}{\partial \mathbf{q}_k} & \frac{\partial \bar{\mathbf{f}}_1}{\partial \mathbf{p}_k} \\ \frac{\partial \bar{\mathbf{f}}_2}{\partial \mathbf{q}_k} & \frac{\partial \bar{\mathbf{f}}_2}{\partial \mathbf{p}_k} \end{bmatrix} \quad (24)$$

where $\bar{\mathbf{f}} = \{\bar{\mathbf{f}}_1^\top, \bar{\mathbf{f}}_2^\top\}^\top$ and

$$\frac{\partial \bar{\mathbf{f}}_1}{\partial \mathbf{p}_k} = \mathbf{I}, \quad \frac{\partial \bar{\mathbf{f}}_1}{\partial \mathbf{p}_{k+1}} = \mathbf{0}, \quad \frac{\partial \bar{\mathbf{f}}_2}{\partial \mathbf{p}_k} = \mathbf{0}, \quad \frac{\partial \bar{\mathbf{f}}_2}{\partial \mathbf{p}_{k+1}} = \mathbf{I} \quad (25)$$

$$\begin{aligned} \frac{\partial \bar{\mathbf{f}}_1}{\partial \mathbf{q}_k} = & - \left[\frac{\partial \mathbf{M}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} \right]_{q=\tilde{q}, \dot{q}=\tilde{v}} + \frac{\eta}{8} \cdot \left[\dot{\mathbf{q}}^\top \frac{\partial^2 \mathbf{M}(\mathbf{q})}{\partial \mathbf{q}^2} \dot{\mathbf{q}} \right]_{q=\tilde{q}, \dot{q}=\tilde{v}} \\ & + \frac{1}{\eta} \cdot \mathbf{M}(\tilde{\mathbf{q}}) - \frac{\eta}{4} \cdot \frac{\partial^2 \mathbf{U}(\tilde{\mathbf{q}})}{\partial \mathbf{q}^2} - \frac{\eta}{4} \cdot \frac{\partial [\Phi_q^\top(\tilde{\mathbf{q}}) \cdot \boldsymbol{\lambda}_{k+1}]}{\partial \mathbf{q}} \quad (26) \\ & + \frac{\eta}{4} \cdot \frac{\partial \mathbf{Q}_a(\tilde{\mathbf{q}}, \tilde{\mathbf{v}}, t)}{\partial \mathbf{q}} + \frac{\eta}{2} \cdot \frac{\partial [\mathbf{A}(\mathbf{q}) \mathbf{u}_{k+1}]}{\partial \mathbf{q}} \Big|_{q=q_k} \end{aligned}$$

$$\begin{aligned} \frac{\partial \bar{\mathbf{f}}_1}{\partial \mathbf{q}_{k+1}} = & - \frac{1}{\eta} \cdot \mathbf{M}(\tilde{\mathbf{q}}) - \frac{\eta}{4} \cdot \frac{\partial^2 \mathbf{U}(\tilde{\mathbf{q}})}{\partial \mathbf{q}^2} - \frac{\eta}{4} \cdot \frac{\partial [\Phi_q^\top(\tilde{\mathbf{q}}) \cdot \boldsymbol{\lambda}_{k+1}]}{\partial \mathbf{q}} \quad (27) \\ & + \frac{\eta}{8} \cdot \left[\dot{\mathbf{q}}^\top \frac{\partial^2 \mathbf{M}(\mathbf{q})}{\partial \mathbf{q}^2} \dot{\mathbf{q}} \right]_{q=\tilde{q}, \dot{q}=\tilde{v}} + \frac{\eta}{4} \cdot \frac{\partial \mathbf{Q}_a(\tilde{\mathbf{q}}, t)}{\partial \mathbf{q}} \end{aligned}$$

$$\begin{aligned} \frac{\partial \bar{\mathbf{f}}_2}{\partial \mathbf{q}_k} = & \frac{1}{\eta} \cdot \mathbf{M}(\tilde{\mathbf{q}}) - \frac{\eta}{8} \cdot \left[\dot{\mathbf{q}}^\top \frac{\partial^2 \mathbf{M}(\mathbf{q})}{\partial \mathbf{q}^2} \dot{\mathbf{q}} \right]_{q=\tilde{q}, \dot{q}=\tilde{v}} \quad (28) \\ & + \frac{\eta}{4} \cdot \frac{\partial^2 \mathbf{U}(\tilde{\mathbf{q}})}{\partial \mathbf{q}^2} + \frac{\eta}{4} \cdot \frac{\partial [\Phi_q^\top(\tilde{\mathbf{q}}) \cdot \boldsymbol{\lambda}_{k+1}]}{\partial \mathbf{q}} \\ & - \frac{\eta}{4} \cdot \frac{\partial \mathbf{Q}_a(\tilde{\mathbf{q}}, t)}{\partial \mathbf{q}} - \frac{\eta}{2} \cdot \frac{\partial [\mathbf{A}(\mathbf{q}) \mathbf{u}_{k+1}]}{\partial \mathbf{q}} \Big|_{q=q_k} \end{aligned}$$

$$\begin{aligned} \frac{\partial \bar{\mathbf{f}}_2}{\partial \mathbf{q}_{k+1}} = & - \left[\frac{\partial \mathbf{M}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} \right]_{q=\tilde{q}, \dot{q}=\tilde{v}} - \frac{1}{\eta} \cdot \mathbf{M}(\tilde{\mathbf{q}}) \\ & - \frac{\eta}{8} \cdot \left[\dot{\mathbf{q}}^\top \frac{\partial^2 \mathbf{M}(\mathbf{q})}{\partial \mathbf{q}^2} \dot{\mathbf{q}} \right]_{q=\tilde{q}, \dot{q}=\tilde{v}} + \frac{\eta}{4} \cdot \frac{\partial^2 \mathbf{U}(\tilde{\mathbf{q}})}{\partial \mathbf{q}^2} \quad (29) \\ & + \frac{\eta}{4} \cdot \frac{\partial [\Phi_q^\top(\tilde{\mathbf{q}}) \cdot \boldsymbol{\lambda}_{k+1}]}{\partial \mathbf{q}} - \frac{\eta}{4} \cdot \frac{\partial \mathbf{Q}_a(\tilde{\mathbf{q}}, t)}{\partial \mathbf{q}} \end{aligned}$$

So far, substituting (24) into (21), the criteria of (21) is satisfied. Therefore, it has been proved that (12) defines a symplectic-preserving update map $\{q_j^\top, p_j^\top\}^\top \mapsto \{q_{j+1}^\top, p_{j+1}^\top\}^\top$ in the absence of a velocity item in the external forces. Then, in the following two sub-sections, based on the above discretization equations, an

instantaneous optimal control method will be presented to solve the tracking control problem with input saturation for the robot multibody dynamic system.

IV. THE SYMPLECTIC IOC METHOD FOR ROBOT TRAJECTORY TRACKING WITH INPUT SATURATION

In this section, a symplectic IOC method for the robot trajectory tracking multibody dynamic system with input saturation is proposed. The depiction and formula derivation can be found in the following sub-sections.

A. Construction of the Iterative Scheme

Equations (13) and (14) along with (18) constitute a nonlinear algebraic equation, which depends on the variables \mathbf{q}_{k+1} , $\boldsymbol{\lambda}_{k+1}$, \mathbf{p}_{k+1} and \mathbf{u}_{k+1} . Combining (13) and (18), the generalized coordinates \mathbf{q}_{k+1} and the Lagrange multipliers $\boldsymbol{\lambda}_{k+1}$ can be obtained by applying the Newton-Raphson iteration method. We have $\mathbf{x}_k = (\mathbf{q}_k, \boldsymbol{\lambda}_k)$, $\mathbf{x}_{k+1} = (\mathbf{q}_{k+1}, \boldsymbol{\lambda}_{k+1})$, and

$$\mathbf{F} = [\mathbf{f}_1^T, (-\eta/2) \cdot \mathbf{f}_3^T]^T \quad (30)$$

$$\hat{\mathbf{F}} = \mathbf{F} + \mathbf{B}(\mathbf{q}_k) \cdot \mathbf{u}_{k+1} \quad (31)$$

where \mathbf{F} and $\hat{\mathbf{F}}$ represent the iteration function for uncontrolled and controlled DAE systems, respectively. It is noted here that the function \mathbf{f}_3 is multiplied by a scaling factor $-\eta/2$ to increase the stability of the algorithm.

The matrix $\mathbf{B}(\mathbf{q}_k)$ can be given as

$$\mathbf{B}(\mathbf{q}_k) = 0.5\eta \cdot \begin{bmatrix} \mathbf{I}_{n \times n} \\ \mathbf{0}_{s \times n} \end{bmatrix} \cdot \mathbf{A}(\mathbf{q}_k) \quad (32)$$

Equation (31) is a nonlinear algebraic equation, which usually can be solved by the Newton-Raphson iteration algorithm [27]. The iteration scheme with control inputs \mathbf{u}_{k+1} can be given as

$$\begin{aligned} \mathbf{x}_{k+1}^{(j+1)} &= \mathbf{x}_{k+1}^{(j)} - \hat{\mathbf{F}}_{\mathbf{x}_{k+1}}'^{-1}(\mathbf{x}_{k+1}^{(j)}) \cdot \hat{\mathbf{F}}(\mathbf{x}_{k+1}^{(j)}) \\ &= \mathbf{x}_{k+1}^{(j)} - \mathbf{F}_{\mathbf{x}_{k+1}}'^{-1}(\mathbf{x}_{k+1}^{(j)}) \cdot \mathbf{F}(\mathbf{x}_{k+1}^{(j)}) - \mathbf{F}_{\mathbf{x}_{k+1}}'^{-1}(\mathbf{x}_{k+1}^{(j)}) \cdot \mathbf{B}(\mathbf{q}_k) \cdot \mathbf{u}_{k+1} \end{aligned} \quad (33)$$

where the superscript in (33) denotes the iteration index, the symbol $\mathbf{x}_{k+1}^{(j+1)}$ is the variable vector of the current $(j+1)^{\text{th}}$ iteration at time t_{k+1} , and $\mathbf{x}_{k+1}^{(j)}$ is the variable vector from the previous j^{th} iteration, which is taken as the

reference for the current computation. The initial value of \mathbf{x}_{k+1} for the iteration is $\mathbf{x}_{k+1}^{(0)} = \mathbf{x}_k$. The operator $\mathbf{F}'_{\mathbf{x}_{k+1}}$ (or $\hat{\mathbf{F}}'_{\mathbf{x}_{k+1}}$) computes the Jacobian matrix of \mathbf{F} (or $\hat{\mathbf{F}}$), which can be given as

$$\mathbf{F}'_{\mathbf{x}_{k+1}} = \begin{bmatrix} \frac{\partial f_1}{\partial \mathbf{q}_{k+1}} & \frac{\partial f_1}{\partial \boldsymbol{\lambda}_{k+1}} \\ -\frac{\eta}{2} \cdot \frac{\partial f_3}{\partial \mathbf{q}_{k+1}} & \mathbf{0}_{s \times s} \end{bmatrix} \quad (34)$$

To simplify the presentation, (33) can be rewritten as

$$\mathbf{x}_{k+1}^{(j+1)} = \boldsymbol{\xi}_1^{(j)} - \boldsymbol{\xi}_2^{(j)} \mathbf{u}_{k+1} \quad (35)$$

where

$$\begin{cases} \boldsymbol{\xi}_1^{(j)} = \mathbf{x}_{k+1}^{(j)} - \mathbf{F}'_{\mathbf{x}_{k+1}}^{-1}(\mathbf{x}_{k+1}^{(j)}) \cdot \mathbf{F}(\mathbf{x}_{k+1}^{(j)}) \\ \boldsymbol{\xi}_2^{(j)} = \mathbf{F}'_{\mathbf{x}_{k+1}}^{-1}(\mathbf{x}_{k+1}^{(j)}) \cdot \mathbf{B}(\mathbf{q}_k) \end{cases} \quad (36)$$

So far, the current iteration variables $\mathbf{x}_{k+1}^{(j+1)}$ have been derived and expressed by the control inputs \mathbf{u}_{k+1} .

B. Formulation and Solution of the Symplectic IOC Method

In this sub-section, a symplectic IOC method with input saturation will be proposed to minimize the tracking error as much as possible. As shown in (3) of section II, the tracking problem is described as a nonlinear optimal control problem, subject to the constraints of DAEs and input saturation. An optimal control law for such systems is hard to obtain, since these complex constraints are imposed continuously over the entire time horizon. Thus, we will find a suboptimal control law for this problem. It is noted that the IOC method is applicable to the control for linear, nonlinear and hysteretic structural systems [25]. As presented in [24], the time-dependent performance index of the IOC method is just minimized at every time instant t for all $t_0 \leq t \leq t_f$. Therefore, compared with the performance index J given by (3), which is the integral of the quadratic functions over the time interval (t_0, t_f) , the control law can be designed easily and very efficiently by the IOC method.

Combined with the symplectic discretization form of the controlled DAEs in section III, we can transcribe the nonlinear optimal control problem of (3) into a discrete form as follows

$$\begin{cases} \text{find } \mathbf{u}_{k+1} \\ \text{min. } \hat{J} = (\hat{\mathbf{y}}_{k+1} - \tilde{\mathbf{y}}_{k+1})^T \cdot \hat{\mathbf{Q}} \cdot (\hat{\mathbf{y}}_{k+1} - \tilde{\mathbf{y}}_{k+1}) + \mathbf{u}_{k+1}^T \cdot \hat{\mathbf{R}} \cdot \mathbf{u}_{k+1} \\ \text{s.t. } \mathbf{x}_{k+1}^{(j+1)} = \boldsymbol{\xi}_1^{(j)} - \boldsymbol{\xi}_2^{(j)} \mathbf{u}_{k+1} \\ \mathbf{u}_{\min} \leq \mathbf{u}_{k+1} \leq \mathbf{u}_{\max} \end{cases} \quad (37)$$

where $\hat{\mathbf{y}}_{k+1}$ is the output variable vector, which is equivalent to the operator $\mathbf{y}(\mathbf{q}, t)$ in (3). It can be obtained by the following output equation

$$\hat{\mathbf{y}}_{k+1} = \mathbf{C} \cdot \mathbf{x}_{k+1}^{(j+1)} \quad (38)$$

where $\mathbf{C} \in \mathbb{R}^{r \times (n+s)}$ is the target output matrix and r is the number of generalized output coordinates. Generally, the output values can include the positions, velocities and accelerations of the tracked points. In this work, the outputs are the position coordinates of a point to be tracked.

To find the suboptimal control inputs \mathbf{u}_{k+1} , the constrained optimization problem in (37) will be transformed into an unconstrained optimization problem.

First, for the equality constraints, substitute the iterative scheme of (35) into the output equation (38); then, the output vector $\hat{\mathbf{y}}_{k+1}$ in the performance index \hat{J} can be expressed by the control inputs \mathbf{u}_{k+1} explicitly, given as

$$\hat{J} = (\hat{\boldsymbol{\xi}}_1 - \hat{\boldsymbol{\xi}}_2 \mathbf{u}_{k+1} - \tilde{\mathbf{y}}_{k+1})^T \cdot \hat{\mathbf{Q}} \cdot (\hat{\boldsymbol{\xi}}_1 - \hat{\boldsymbol{\xi}}_2 \mathbf{u}_{k+1} - \tilde{\mathbf{y}}_{k+1}) + \mathbf{u}_{k+1}^T \hat{\mathbf{R}} \mathbf{u}_{k+1} \quad (39)$$

where

$$\hat{\boldsymbol{\xi}}_1 = \mathbf{C} \boldsymbol{\xi}_1^{(j)}, \quad \hat{\boldsymbol{\xi}}_2 = \mathbf{C} \boldsymbol{\xi}_2^{(j)} \quad (40)$$

Second, for the inequality constraints, introducing variables $\bar{\boldsymbol{\alpha}}_{k+1} \geq \mathbf{0}$ and $\underline{\boldsymbol{\alpha}}_{k+1} \geq \mathbf{0}$, the inequality constraints in (37) can be transformed into the equality constraints as follows

$$\begin{cases} \mathbf{u}_{k+1} - \mathbf{u}_{\max} + \bar{\boldsymbol{\alpha}}_{k+1} = \mathbf{0} \\ \mathbf{u}_{\min} - \mathbf{u}_{k+1} + \underline{\boldsymbol{\alpha}}_{k+1} = \mathbf{0} \end{cases} \quad (41)$$

Then, based on the theory of instantaneous optimal control and the parametric variational principle [31], an expanded performance index \hat{J}_c can be defined as

$$\begin{aligned} \hat{J}_c = & (\hat{\boldsymbol{\xi}}_1 - \hat{\boldsymbol{\xi}}_2 \mathbf{u}_{k+1} - \tilde{\mathbf{y}}_{k+1})^T \cdot \hat{\mathbf{Q}} \cdot (\hat{\boldsymbol{\xi}}_1 - \hat{\boldsymbol{\xi}}_2 \mathbf{u}_{k+1} - \tilde{\mathbf{y}}_{k+1}) + \mathbf{u}_{k+1}^T \hat{\mathbf{R}} \mathbf{u}_{k+1} \\ & + \bar{\boldsymbol{\beta}}_{k+1}^T (\mathbf{u}_{k+1} - \mathbf{u}_{\max} + \bar{\boldsymbol{\alpha}}_{k+1}) + \underline{\boldsymbol{\beta}}_{k+1}^T (\mathbf{u}_{\min} - \mathbf{u}_{k+1} + \underline{\boldsymbol{\alpha}}_{k+1}) \end{aligned} \quad (42)$$

where $\bar{\boldsymbol{\beta}}_{k+1}, \underline{\boldsymbol{\beta}}_{k+1} \in R^{m \times 1}$ are the parameter variables at time t_{k+1} .

So far, the problem has been changed into an unconstrained optimization problem. Then, performing the calculus of variations with respect to the control inputs \mathbf{u}_{k+1} is given as

$$\partial \hat{J}_c / \partial \mathbf{u}_{k+1} = \mathbf{0} \quad (43)$$

The suboptimal control inputs \mathbf{u}_{k+1} can be explicitly expressed, which are only dependent on $\bar{\boldsymbol{\beta}}_{k+1}$ and $\underline{\boldsymbol{\beta}}_{k+1}$, given as

$$\mathbf{u}_{k+1} = [\hat{\boldsymbol{\zeta}}_2^T \hat{\mathbf{Q}} \hat{\boldsymbol{\zeta}}_2 + \hat{\mathbf{R}}]^{-1} [\hat{\boldsymbol{\zeta}}_2^T \hat{\mathbf{Q}} (\hat{\boldsymbol{\zeta}}_1 - \tilde{\mathbf{y}}_{k+1}) - (\bar{\boldsymbol{\beta}}_{k+1} - \underline{\boldsymbol{\beta}}_{k+1}) / 2] \quad (44)$$

To simplify the presentation and avoid the reduplicative computation of the matrix, we have

$$\hat{\mathbf{K}}_1 = (\hat{\boldsymbol{\zeta}}_1^T \hat{\mathbf{Q}} \hat{\boldsymbol{\zeta}}_1 + \hat{\mathbf{R}})^{-1} / 2, \quad \hat{\mathbf{K}}_2 = 2\hat{\mathbf{K}}_1 \hat{\boldsymbol{\zeta}}_2^T \hat{\mathbf{Q}} \quad (45)$$

Then, (44) can be rewritten as

$$\mathbf{u}_{k+1} = -\hat{\mathbf{K}}_1 (\bar{\boldsymbol{\beta}}_{k+1} - \underline{\boldsymbol{\beta}}_{k+1}) + \hat{\mathbf{K}}_2 (\hat{\boldsymbol{\zeta}}_1 - \tilde{\mathbf{y}}_{k+1}) \quad (46)$$

Substituting (46) into (41), the constraint conditions can be transformed as follows:

$$\begin{cases} -\hat{\mathbf{K}}_1 (\bar{\boldsymbol{\beta}}_{k+1} - \underline{\boldsymbol{\beta}}_{k+1}) + \hat{\mathbf{K}}_2 (\hat{\boldsymbol{\zeta}}_1 - \tilde{\mathbf{y}}_{k+1}) - \mathbf{u}_{\max} + \bar{\boldsymbol{\alpha}}_{k+1} = \mathbf{0} \\ \mathbf{u}_{\min} + \hat{\mathbf{K}}_1 (\bar{\boldsymbol{\beta}}_{k+1} - \underline{\boldsymbol{\beta}}_{k+1}) - \hat{\mathbf{K}}_2 (\hat{\boldsymbol{\zeta}}_1 - \tilde{\mathbf{y}}_{k+1}) + \underline{\boldsymbol{\alpha}}_{k+1} = \mathbf{0} \end{cases} \quad (47)$$

Rewriting (47) in matrix form, yields

$$\begin{cases} \hat{\mathbf{K}}_2 (\hat{\boldsymbol{\zeta}}_1 - \tilde{\mathbf{y}}_{k+1}) - \mathbf{u}_{\max} \\ \mathbf{u}_{\min} - \hat{\mathbf{K}}_2 (\hat{\boldsymbol{\zeta}}_1 - \tilde{\mathbf{y}}_{k+1}) \end{cases} + \begin{bmatrix} -\hat{\mathbf{K}}_1 & \hat{\mathbf{K}}_1 \\ \hat{\mathbf{K}}_1 & -\hat{\mathbf{K}}_1 \end{bmatrix} \begin{cases} \bar{\boldsymbol{\beta}}_{k+1} \\ \underline{\boldsymbol{\beta}}_{k+1} \end{cases} + \begin{cases} \bar{\boldsymbol{\alpha}}_{k+1} \\ \underline{\boldsymbol{\alpha}}_{k+1} \end{cases} = \mathbf{0} \quad (48)$$

Consequently, the constraint IOC problem described by (37) is transformed into a linear complementarity problem (LCP). $\bar{\boldsymbol{\beta}}_{k+1}$ and $\underline{\boldsymbol{\beta}}_{k+1}$ can be calculated by solving the LCP, given as

$$\begin{cases} \hat{\mathbf{q}} + \hat{\mathbf{H}} \hat{\mathbf{z}} + \hat{\boldsymbol{w}} = \mathbf{0} \\ \hat{\boldsymbol{w}} \geq \mathbf{0}, \hat{\mathbf{z}} \geq \mathbf{0}, \hat{\mathbf{z}}^T \hat{\boldsymbol{w}} = \mathbf{0} \end{cases} \quad (49)$$

where

$$\hat{\mathbf{q}} = \begin{cases} \hat{\mathbf{K}}_2 (\hat{\boldsymbol{\zeta}}_1 - \tilde{\mathbf{y}}_{k+1}) - \mathbf{u}_{\max} \\ \mathbf{u}_{\min} - \hat{\mathbf{K}}_2 (\hat{\boldsymbol{\zeta}}_1 - \tilde{\mathbf{y}}_{k+1}) \end{cases}, \quad \hat{\mathbf{H}} = \begin{bmatrix} -\hat{\mathbf{K}}_1 & \hat{\mathbf{K}}_1 \\ \hat{\mathbf{K}}_1 & -\hat{\mathbf{K}}_1 \end{bmatrix} \quad (50)$$

$$\hat{\mathbf{z}} = \begin{cases} \bar{\boldsymbol{\beta}}_{k+1} \\ \underline{\boldsymbol{\beta}}_{k+1} \end{cases}, \quad \hat{\boldsymbol{w}} = \begin{cases} \bar{\boldsymbol{\alpha}}_{k+1} \\ \underline{\boldsymbol{\alpha}}_{k+1} \end{cases}, \quad \begin{cases} \hat{\mathbf{K}}_1 = (\hat{\boldsymbol{\zeta}}_1^T \hat{\mathbf{Q}} \hat{\boldsymbol{\zeta}}_1 + \hat{\mathbf{R}})^{-1} / 2 \\ \hat{\mathbf{K}}_2 = 2\hat{\mathbf{K}}_1 \hat{\boldsymbol{\zeta}}_2^T \hat{\mathbf{Q}} \end{cases}$$

The LCP has been well investigated in the last several decades; it can be solved by several methods, including interior-point methods [28], the pivotal methods [29] and noninterior continuation methods [30]. In this paper, the Lemke's algorithm, which belongs to the pivotal method category, is adopted to solve the standard LCP of (49). One of the advantages of the proposed method is that the input saturation constraints can be satisfied directly just by solving the LCP, rather than adjusting the control parameters such as the weighting matrices to guarantee that the actuators are operating below their physical limits indirectly. Therefore, one can solve the LCP to obtain the parameters $\bar{\beta}_{k+1}$ and $\underline{\beta}_{k+1}$ firstly; then, the control inputs \mathbf{u}_{k+1} and the current iteration variables $\mathbf{x}_{k+1}^{(j+1)}$ can be obtained by (44) and (35), respectively. So far, one time numerical iteration of the nonlinear algebraic equations for (13) and (18) has been completed. Thus, the variables \mathbf{x}_{k+1} , i.e., the generalized coordinates \mathbf{q}_{k+1} and the Lagrange multipliers $\boldsymbol{\lambda}_{k+1}$, and the suboptimal control inputs \mathbf{u}_{k+1}^* for the current time step can be solved successively until an acceptable convergence error is achieved in few iterations. The convergence criteria can be defined by the variables \mathbf{x}_{k+1} or the performance index \hat{J}_c , given as follows, and the former is used in this work.

$$\|\mathbf{x}_{k+1}^{(j+1)} - \mathbf{x}_{k+1}^{(j)}\| / \|\mathbf{x}_{k+1}^{(j+1)}\| < \varepsilon \text{ or } \|\hat{J}_c^{(j+1)} - \hat{J}_c^{(j)}\| / \|\hat{J}_c^{(j+1)}\| < \varepsilon \quad (51)$$

Finally, the dual variables \mathbf{p}_{k+1} can be successively obtained by substituting \mathbf{q}_{k+1} , $\boldsymbol{\lambda}_{k+1}$ and \mathbf{u}_{k+1}^* into (14). Also it can be derived by $\hat{\mathbf{f}}_1 - \hat{\mathbf{f}}_2 = \mathbf{0}$, i.e., $\mathbf{p}_{k+1} = 2\mathbf{M}(\tilde{\mathbf{q}}) \cdot \tilde{\mathbf{v}} - \mathbf{p}_k$.

Therefore, the variables in time interval $[t_k, t_{k+1}]$ can be obtained, then computing step by step, the numerical simulation can be completed successfully. Before the end of this section, two beneficial remarks about the stability and optimality of the proposed method have been given as follows.

Remark 2 (local stability): The controlled DAEs system described by (31) implies a nonlinear discrete dynamical system. Substituting the suboptimal control inputs \mathbf{u}_{k+1} of (46) into the iteration equation (35), a closed-loop control system, which is a map at time t_{k+1} , can then be constructed by a vector function $\psi(\mathbf{x})$, which maps a vector $\mathbf{x}^{(j)}$ onto a vector $\mathbf{x}^{(j+1)}$, i.e.

$$\mathbf{x}^{(j+1)} = \psi(\mathbf{x}^{(j)}) \quad (52)$$

where

$$\psi(\mathbf{x}^{(j)}) = \hat{\boldsymbol{\xi}}_1^{(j)} + \hat{\boldsymbol{\xi}}_2^{(j)} [\hat{\mathbf{K}}_1(\bar{\boldsymbol{\beta}}_{k+1} - \underline{\boldsymbol{\beta}}_{k+1}) - \hat{\mathbf{K}}_2(\hat{\boldsymbol{\xi}}_1 - \tilde{\mathbf{y}}_{k+1})] \quad (53)$$

Then, the solution \mathbf{x}^* solved by $\mathbf{x}^* = \psi(\mathbf{x}^*)$ can be treated as a fixed point. According to the stability theory of fixed points [32], the convergence $\mathbf{x}^{(j)} \mapsto \mathbf{x}^*$ can only take place for $j \rightarrow \infty$ when $|\mu_i| < 1$ for all $i=1, 2, \dots, n+s$. Here, μ_i denotes all the eigenvalues of the Jacobian matrix $\psi(\mathbf{x})$, evaluated at \mathbf{x}^* . If the fixed points of all the time moments are locally stable, then the controlled trajectory will obviously converge to a region around the target path in the whole time domain. The local stability of the proposed method can also be verified by numerical analysis as demonstrated in the next section.

Remark 3 (optimality): The global optimal formula (3) for the entire time domain is transformed into a local optimal formula (37), which is an IOC problem. The Lagrange multiplier method is applied for solving this problem. The control inputs at the current moment are locally optimal since they satisfy the necessary optimality conditions (43). Then, based on the theory of IOC, a suboptimal control law can be achieved by combining the control input sequences at all time points. Therefore, the proposed symplectic IOC method is a global suboptimal control algorithm with local optimality.

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, two robot trajectory tracking control problems will be investigated to illustrate the effectiveness of the proposed method. It is noted here that the convergence precision ε in the criterion of (51) and the gravity acceleration are set to be $1.0\text{e-}7$ and -9.8N/kg , respectively. And all computations were performed in MATLAB (R2014a) on a personal computer with an Intel(R) core(TM) i7-6700 CPU (3.40 GHz) processor and 8 GB RAM, running on Win.7 64bit.

A. A Plane Serial Manipulator



Fig. 1 Two types of robot multibody models. (a) The model of plane serial manipulators. (b) The model of a space parallel delta robot.

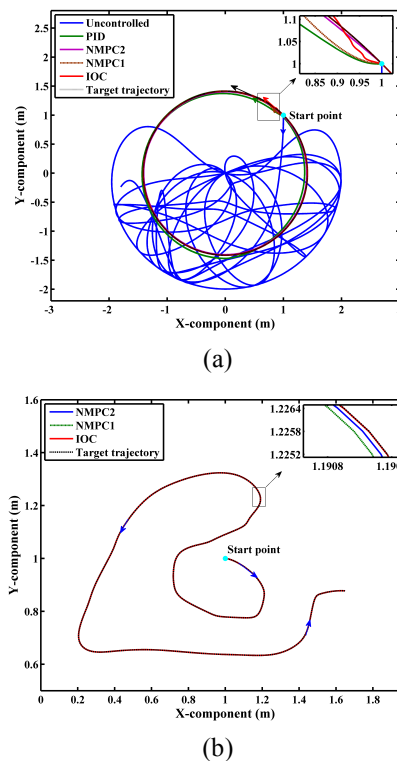


Fig. 2 Comparisons for different trajectories of the controlled point P_r . (a) The circular motion trajectory. (b) The random motion trajectory.

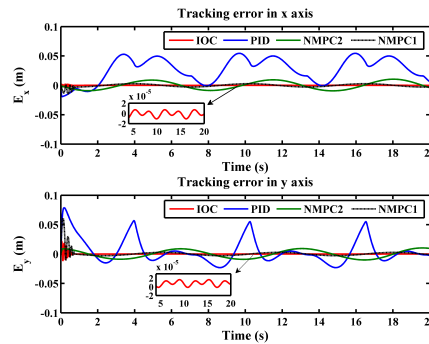
As shown in Figs. 1(a–b), serial and parallel are two typical geometric models of industrial robots. Their various closed-loop robot trajectory tracking control methods based on ODEs have been discussed extensively in the existing literatures, for instance, the nonlinear model predictive control (NMPC) algorithm [18], [19]. However, to the best knowledge of the authors, the closed-loop trajectory tracking control methods based on index-3 DAEs can rarely be found. Considering that in practical control engineering, there

are many model-independent control methods, which are convenient for code realization, such as the PID control. Thus, in this sub-section, for the serial manipulator trajectory tracking control problem, the PID control and the NMPC will be compared to demonstrate the validity and the robustness of the proposed symplectic IOC method.

For the serial manipulators shown in Fig. 1(a), assuming that both bars are homogeneous and rigid, their lengths and masses are $l_1 = l_2 = 1$ m and $m_1 = m_2 = 2$ kg, respectively. The initial generalized coordinates $\mathbf{q}_0 = (0.5, 0, 0, -0.7071, 0, 0.7071, 0, 1, 0.5, 0, -0.5, 0.5, 0.5, 0.5)^T$, and the corresponding velocities $\dot{\mathbf{q}}_0 = \mathbf{0}$. The trajectory tracking control problem can be defined so that the controlled end point P_r is forced to follow a given trajectory, which is defined by a circular path or a feasible random path from a given start point $(X_0, Y_0) = (1, 1)$. The control inputs $\mathbf{u}(t)$ are the torques of M_1 and M_2 , which are applied on these two joints. The circular trajectory is given as

$$\tilde{\mathbf{y}}(t) = \begin{Bmatrix} \tilde{X}_d \\ \tilde{Y}_d \end{Bmatrix} = \begin{Bmatrix} \sqrt{2} \times \cos(t + 0.25\pi) \\ \sqrt{2} \times \sin(t + 0.25\pi) \end{Bmatrix} \quad (54)$$

The random trajectory is designed by the spline interpolation of selecting some discrete points. In addition, the control inputs of the system are subject to the inequality constraint $|\mathbf{u}(t)| \leq 50$ Nm.



(a)

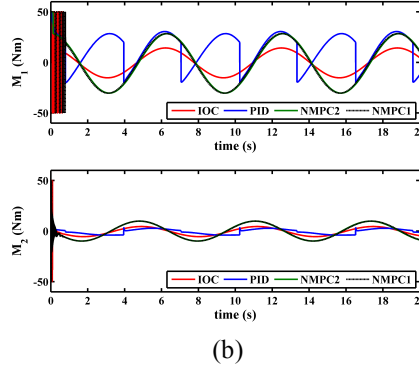


Fig. 3 Comparisons for the tracking errors and control inputs of the serial manipulators with disturbances. (a) Time history curves of the tracking errors. (b) Time history curves of the control inputs.

Because that disturbance rejection is one of the most important objectives to any control methodology, it reflects to some extent the robustness of a control approach. Therefore, the control simulations are performed not only in a gravity field environment but also with two disturbance torques that are set on the joints, which are given as

$$\boldsymbol{\sigma}(t) = \{\sigma_1, \sigma_2\}^T, \sigma_1 = \sin(t) \text{ and } \sigma_2 = 1 - \cos(t) \quad (55)$$

The above disturbance torques can be treated as the generalized external forces, which can be added to the controlled differential equations as follows

$$\mathbf{M}(q)\ddot{q} + \Phi_q^T(q, t)\lambda = \mathbf{Q}(q, \dot{q}, t) + \mathbf{A}(q)u(t) + \mathbf{W}(q)\boldsymbol{\sigma}(t) \quad (56)$$

where $\mathbf{W}(q)$ distributes the disturbance torques $\boldsymbol{\sigma}(t)$ onto the directions of the system generalized coordinates. In addition, for the proposed method, the weighting matrices in the expanded performance index \hat{J}_c of (42) are chosen to be $\hat{\mathbf{Q}} = 1.0 \times 10^{12} \times I_3$ and $\hat{\mathbf{R}} = 1.0 \times I_2$, where $I_2 \in R^{2 \times 2}$ and $I_3 \in R^{3 \times 3}$ are unit matrices. All the results are computed with a constant step length $\eta = 0.002s$. For the PID method, referring to the literature [34], the controller is designed by the PD compensator with a feedforward transfer function. The gain coefficient matrices of the proportional and derivative items are both set to be $30.0 \times I_2$. For the NMPC method, referring to the literature [18], the ODE-based continuous-time state equation of the serial manipulators can be discretized by the backward Euler's method, and the Resilient Propagation (RPROP) algorithm is adopted to minimize the cost function. In addition, referring to the literature [19], the index-1 DAEs of the serial

manipulators is derived, and the system is discretized via the BDF integration method, then the NMPC problem is solved by the software toolkit Automatic Control and Dynamic Optimization (ACADO). Here, the above two NMPC methods are named NMPC1 and NMPC2, respectively. The weighting values in their cost functions are chosen the same as those of the proposed symplectic IOC method. The prediction horizon time is 0.01s, and the number of prediction points is 5. The control results for these four methods with disturbances are presented in Figs. 2 and 3.

As shown in Fig. 2(a), if there are no control inputs for this robot manipulator system, then the position of the end point P_r will move in a disorderly fashion. This phenomenon shows a very complex nonlinear characteristic of the motion for the double pendulum system, in which chaos can even exist. While controlled by the PID method, the NMPC1 method, the NMPC2 method or the proposed symplectic IOC method, the target trajectory can be tracked effectively. The average online CPU time of the above controllers for one step are approximately 1.20×10^{-5} s, 2.86×10^{-4} s, 8.11×10^{-4} s and 4.48×10^{-4} s, which are all less than the sampling period of 0.002 s. However, the details between these four methods can be compared. For the PID method, deviations will obviously occur at some moments. The control effect is improved to some extent by both the NMPC1 and NMPC2 method, while the target point P_r can be well tracked by the proposed symplectic IOC method. The tracking errors in the X-direction and Y-direction are presented clearly in Fig. 3(a) and the specific data are listed in Table I. It is noted that the results of the NMPC1 and the NMPC2 methods in the tables are marked by ‘(a)’ and ‘(b)’, respectively. For the PID method, the maximum absolute (Max. Abs.) values increase to approximately 0.05 m. And those are approximately 0.003 m for the NMPC1 method and 0.008 m for the NMPC2 method. However, for the proposed method, tracking errors can be kept at the magnitude order of 10^{-5} m as shown in the enlarged drawing of the IOC method. Furthermore, as shown in Fig. 3(b), the control inputs for the proposed method can be well satisfied with the saturation constraints, and the values are limited between -50 Nm and 50 Nm. Detailed data comparisons for control inputs in the time range 4–20 s are made in Table II. The root-mean-square (RMS) in Table I and Table II can be expressed as

$$RMS(\Omega_i) = \sqrt{\sum_{k=1}^N \Omega_i^2(t_k)} / N, \Omega = E \text{ or } M \quad (57)$$

TABLE I
COMPARISONS OF TRACKING ERRORS IN THE TIME RANGE 4-20 s

Tracking error	E_x ($\times 10^{-3}$ m)			E_y ($\times 10^{-3}$ m)		
	NMPC	PID	IOC	NMPC	PID	IOC
RMS	2.0 (a)	34.2	0.005	2.0 (a)	21.2	0.010
	8.1 (b)			4.7 (b)		
Max. Abs.	2.8 (a)	53.0	0.011	2.8 (a)	55.0	0.016
	8.8 (b)			7.1 (b)		

TABLE II
COMPARISONS OF CONTROL INPUTS IN THE TIME RANGE 4-20 s

Control input	M_1 (Nm)			M_2 (Nm)		
	NMPC	PID	IOC	NMPC	PID	IOC
RMS	20.59 (a)	20.94	10.31	6.98 (a)	2.47	3.52
	20.47 (b)			7.04 (b)		
Max.	28.44 (a)	30.49	14.24	9.86 (a)	3.85	4.51
	28.39 (b)			9.84 (b)		
Min.	-30.44 (a)	-20.18	-15.24	-9.86 (a)	-3.96	-5.51
	-30.40 (b)			-9.83 (b)		

From Table II, it can be found that for the proposed method, the RMS, maximum (Max.) and minimum (Min.) of the control input M_1 are about 10.31 Nm, 14.24 Nm and -15.24 Nm, respectively. However, for the PID method, the corresponding data will increase by approximately 103.1%, 114.1% and 32.4%, respectively. Those will also increase by approximately 99.7% for the NMPC1 method, and approximately 98.5%, 99.4% and 99.5%, respectively, for the NMPC2 method. The control input M_2 of the proposed method is slightly larger than that of the PID method and smaller than the two NMPC methods. The RMS values of M_2 for both the four methods are smaller than those of M_1 , and they are only approximately 34.1%, 11.8%, 33.9% and 34.4%. Thus, overall, these results demonstrate that the proposed method needs less control inputs than the other three methods to perform the same circular trajectory tracking task. In addition, the good control performance of the proposed method can be further illustrated with the results of a random motion tracking task, which has been presented in Fig. 2(b).

Therefore, compared with the PID method, which belongs to the model-independent control, the NMPC1 method, which is ODE-based and solved by the RPROP algorithm, or the NMPC2 method, which is index-1

DAE-based and solved by the software toolkit ACADO, the above numerical simulations illustrate that the proposed symplectic IOC method has a higher tracking precision and better robustness, which is mainly owed to the symplectic property of the proposed method.

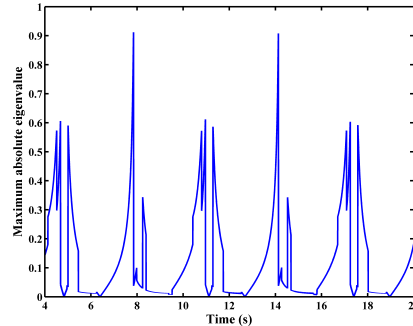


Fig. 4 The Max. Abs. eigenvalues for each step in the time range 4-20 s.

Furthermore, to illustrate the control stability of the proposed algorithm mentioned in Remark 1 of Section IV, the stability of all the fixed points for the nonlinear discrete map $\psi(\mathbf{x})$ at each time step has been analyzed in this example. Fig. 4 gives the maximum absolute eigenvalues of the Jacobian matrix $\psi_{\mathbf{x}}$, evaluated at each solution point, i.e., the fixed point. From Fig. 4, it can be deduced that all the absolute eigenvalues of $\psi_{\mathbf{x}}$ at each time step will lie in an unit circle, since all the corresponding maximum values are between 0 and 1. Based on the stability theory of fixed points [32], the states controlled by the proposed method are stable and reliable.

B. A Space Parallel Delta Robot

For the parallel robots, the kinematics and dynamics are more difficult to derive than those of serial robots if modeling by the ODEs. However, as a model that can describe a wider and more complex robot multibody system, DAEs have a great advantage in the field of numerical simulations for robot multibody dynamics. Thus, to demonstrate the convenience of the modeling process and the ability of solving more complex robot trajectory tracking control problem based on DAEs by the proposed method, the parallel delta robot [33], as shown in Fig. 1(b), will be discussed, not only by a numerical simulation but also by a virtual experiment platform that is set up using V-rep and MATLAB. The implementation strategy of the virtual experiment platform is presented in Fig. 5.

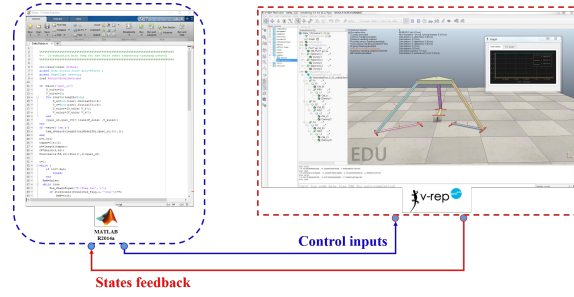


Fig. 5 The virtual experiment platform constructed by V-rep and Matlab

As shown in Fig. 5, the dynamic model of a delta robot should first be built. Then, for the k^{th} cosimulation step, the state values \mathbf{x}_k can be calculated by V-rep. Next, the controller in MATLAB will provide the data \mathbf{x}_k as states feedback and compute the local optimal control forces \mathbf{u}_k by solving the LCP of (49) and some matrix operations of (44). Finally, the control forces \mathbf{u}_k will be applied to the dynamic system in V-rep as external forces and update the states for next time step. In this way, the virtual experiment can be successfully performed step by step.

Then, some DAE-based model information and controller parameters will be given. As shown in Fig. 1(b), the distance between the fixed base and the workspace is 0.52 m, and for the workspace, its thickness is 0.05 m, and its section is an equilateral triangle whose circumradius is 0.02 m. Then, three bars are linked to the fixed base by revolute joints, and the lengths of these bars are $l_1 = 0.5$ m, $l_2 = 0.4$ m and $l_3 = 0.3$ m. In addition, another three bars are linked to the workspace by spherical joints. The material density of all bodies of the system is 2.74×10^3 kg/m³, and the section area of all bars is 8×10^{-4} m². Thus, the trajectory tracking control problem can be defined such that the tracked point P_r is forced to follow a given trajectory, which is defined by a helix path from a given start point $(X_0, Y_0, Z_0) = (0, 0, 0.52)$. The control inputs $\mathbf{u}(t)$ are the torques of M_1 , M_2 and M_3 , which are applied on the revolute joints. The inequality control constraint is $|\mathbf{u}(t)| \leq 200$ Nm, and the helix trajectory is given as

$$\tilde{\mathbf{y}}(t) = \begin{Bmatrix} \tilde{X}_d \\ \tilde{Y}_d \\ \tilde{Z}_d \end{Bmatrix} = \begin{Bmatrix} 0.015 \times \sin(t) \\ 0.015 - 0.015 \times \cos(t) \\ 0.02 \times t + 0.5196 \end{Bmatrix} \quad (58)$$

Then, assume that the model is a rigid multibody system, and the proposed symplectic IOC method, the

NMPC2 method, and the PID method are used to solve this example. For the IOC and the NMPC2 methods, the weighting matrices are set to be $\hat{\mathbf{Q}}=1.0 \times 10^{13} \times I_3$ and $\hat{\mathbf{R}}=1.0 \times I_3$. The initial generalized coordinates \mathbf{q}_0 are given in Table III, and the velocities $\dot{\mathbf{q}}_0 = \mathbf{0}$. For the PD compensator, the gain coefficient matrices of the derivative and proportional items are set to be $30.0 \times I_3$ and $1.3 \times 10^4 \times \text{diag}(5,1,1)$, respectively.

TABLE III
INITIAL GENERALIZED VELOCITIES

ID	Centroid Coordinates (m)			Attitude Angle Described by the Quaternion			
	X	Y	Z	e_0	e_1	e_2	e_3
1	-0.417	-0.240	0	-0.683	-0.183	0.683	-0.183
2	0.373	-0.215	0	-0.683	0.183	0.683	0.183
3	0	0.381	0	-0.5	0.5	0.5	0.5
4	0	0	0.52	-0.5	0.5	0.5	0.5
5	-0.417	-0.240	0.23	-0.078	0.689	0.621	-0.366
6	0.373	-0.215	0.26	-0.309	-0.686	-0.652	-0.091
7	0	0.381	0.26	-0.176	0.659	0.706	0.189

Finally, the number of iterations of the proposed IOC method is given in Fig. 6. The control results of the virtual experiment by the proposed IOC method, the NMPC2 method and the PID method are presented in Fig. 7.

From Fig. 6, it can be seen that an acceptable convergence error by the convergence criteria of (51) can be achieved with 3 or 5 iterations for a short time at the beginning and that only 3 iterations is needed after 1 s. As shown in Fig. 7(a), the controlled point P_r , i.e., the centroid of the workspace, can be tracked by all three methods with different accuracies. The configuration at times $t = 0$ s, 4.02 s and 15 s are presented to observe the tracking process directly. For the proposed IOC method, the stable tracking errors are at the magnitude of order of 10^{-3} m. Compared with the numerical simulation results, the tracking accuracy decreases by the virtual experiment may come from the fact that some inaccurate physical engine methods are adopted for dynamic analysis by V-rep. For the NMPC2 method, somewhat larger errors will occur at some points. However, obvious deviations can be found for the PID method. Besides, the control inputs of all the three methods are limited between -200 Nm and 200 Nm as shown in Fig. 7(b).

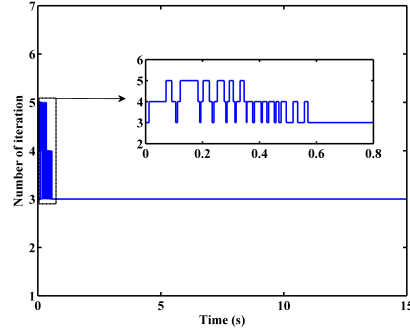
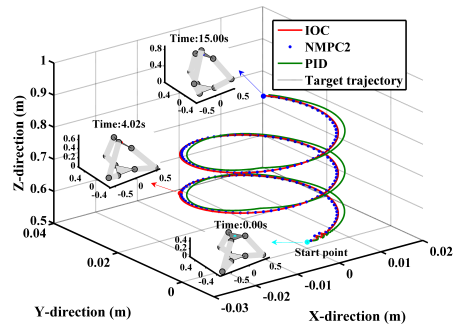
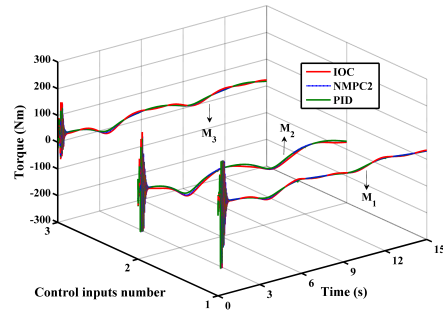


Fig. 6 The number of iteration for the parallel delta robot.



(a)



(b)

Fig. 7 The control results of the parallel delta robot. (a) The trajectories of the controlled point P_r . (b) Time history curves of the control inputs.

The virtual experiment of this example demonstrates that the DAE-based modeling of the robot multibody systems is simple and clear, and it can unify the modeling processes for different types of robots, such as serial robots and parallel robots. The proposed method based on DAEs is effective to solve the more complex robot trajectory tracking control problems.

VI. CONCLUSIONS

In this paper, a symplectic IOC method is proposed to solve the nonlinear trajectory tracking control problem of a robot multibody dynamic system with input saturation. First, based on the continuous controlled

DAEs, the trajectory tracking control problem is described as a nonlinear optimal control problem with constraints. Then, to overcome the difficulties for solving this problem, the continuous nonlinear optimal control problem is transformed into a sequence of IOC problems at all time points with a variational and symplectic discretization scheme. Finally, the numerical solutions for the control inputs can be obtained by solving these IOC problems. The proposed method is a symplectic structure-preserving algorithm and provides a unified framework for solving more general robot tracking control problems. In addition, the control input saturation constraints can be guaranteed directly by the linear complementarity solver. The numerical simulations and virtual experiments demonstrate that the proposed method is effective, easy to implement, and able to provide accurate tracking results. For future work, based on the current algorithm framework, another symplectic discrete strategy with the velocity item in the external forces will be studied to extend the proposed method for further applications. A distributed symplectic IOC method can be further developed to adapt for the larger-scale robot tracking control systems.

REFERENCES

- [1] A. Benosman, and G. Le Vey, "Control of flexible manipulators: A survey," *Robotica*, vol. 22, pp. 533–545, Aug. 2004.
- [2] A. Visioli, and G. Legnani, "On the trajectory tracking control of industrial SCARA robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 49, no. 1, pp. 224-232, Feb. 2002.
- [3] M. Jin, J. Lee, P. H. Chang, and C. Choi, "Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control," *IEEE Trans. Ind. Electron.*, vol. 56, no. 9, pp. 3593-3601, Sep. 2009.
- [4] W. Blaher, "Index of differential-algebraic equations governing the dynamics of constrained mechanical systems," *Appl. Math. Model*, vol. 16, no. 2, pp. 70-77, Feb. 1992.
- [5] C. W. Gear, "Differential-algebraic equation index transformations," *SIAM J. Sci. and Stat. Comput.*, vol. 9, no. 1, PP. 39–47, Jan. 1988.

- [6] R. Altmann, P. Betsch, and Y. P. Yang, "Index reduction by minimal extension for the inverse dynamics simulation of cranes," *Multibody Syst. Dyn.*, 36(3), pp. 295-321, Mar. 2016.
- [7] M. Arnold, and O. Bruels, "Convergence of the generalized- α scheme for constrained mechanical systems," *Multibody Syst. Dyn.*, vol. 18, no. 2, pp. 185-202, Mar. 2007.
- [8] T. Belytschko, and T. J. R. Hughes, *Computational Methods for Transient Analysis*. Amsterdam, North-Holland, 1983.
- [9] H. J. Peng, Q. Gao, Z. G. Wu, and W. X. Zhong, "Symplectic approaches for solving two-point boundary-value problems," *J. Guid. Control Dyn.*, vol. 35, no. 2, pp. 653-658, Mar. 2012.
- [10] K. Feng, and M. Qin, *Symplectic Geometric Algorithms for Hamiltonian Systems*, Springer, New York, 2010.
- [11] J. E. Marsden, and M. West, "Discrete mechanics and variational integrators," *Acta Numer.*, vol. 10, no. 1, pp. 357-514, May. 2001.
- [12] V. I. Arnold, *Mathematical Methods of Classical Mechanics*. Springer-Verlag, New York, 1989.
- [13] S. Leyendecker, J. E. Marsden, S. Ober-Blöbaum, and O. Junge, "Discrete mechanics and optimal control for constrained systems," *Optim. Control Appl. Methods*, vol. 31, no. 6, pp. 505-528, Nov. 2010.
- [14] Q. Gao, S. J. Tan, H. W. Zhang, and W. X. Zhong, "Symplectic algorithms based on the principle of least action and generating functions," *Int. J. Numer. Methods Eng.*, vol. 89, no. 4, pp. 438-508, Jan. 2012.
- [15] W. P. Hu, Z. C. Deng, S. M. Han, and W. R. Zhang, "Generalized multi-symplectic integrators for a class of Hamiltonian nonlinear wave PDEs," *J. Comput. Phys.*, vol. 235, pp. 394-406, Feb. 2013.
- [16] W. W. Shang, S. Cong, and Y. Ge, "Coordination motion control in the task space for parallel manipulators with actuation redundancy," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 665-673, Jul. 2013.
- [17] G. Bastos, R. Seifried, and O. Bruls, "Inverse dynamics of serial and parallel under actuated multibody systems using a DAE optimal control approach," *Multibody Syst. Dyn.*, vol. 30, no. 3, pp. 359-376, Oct. 2013.

- [18] T. P. Nascimento, C. E. T. Dórea, and L. M. G. Gonçalves, "Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots: A modified approach," *Int. J. Adv. Robot. Syst.*, vol. 15, no. (1-2), pp. 1-14, Feb. 2018.
- [19] S. Gros, M. Zanon, Vukob, and M. Diehl, "Nonlinear MPC and MHE for mechanical multi-body systems with application to fast tethered airplanes," in *Proc. the 4th IFAC Conf. on Nonlinear Model Predictive Control*, Noordwijkerhout, NL, Aug. 2012.
- [20] A. Tatulea, Codrean, D. Haßkerl, M. Urselmann, and S. Engell, "Steady-state optimization and nonlinear model-predictive control of a reactive distillation process using the software platform do-mpc," in *Proc. IEEE Conf. on Control Applications (CCA)*, Buenos Aires, Argentina, Sep. 2016, pp. 1513-1518.
- [21] J. Baek, M. Jin, and S. Han, "A new adaptive sliding-mode control scheme for application to robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3628-3637, Jun. 2016.
- [22] W. He, Y. H. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE T. Cybern.*, vol. 46, no. 3, pp. 620-629, Mar. 2016.
- [23] A. Iserles, "Multistep methods on manifolds," *IMA J. Numer. Anal.*, vol. 21, no.1, pp. 407-419, Jan. 2001.
- [24] J. N. Yang, A. Akbarpour, and P. Ghaemmaghami, "New optimal control algorithms for structural control," *J. Eng. Mech.*, vol. 113, no. 9, pp. 1369-1386, Sep. 1987.
- [25] J. N. Yang, and S. C. Liu, "Stable controllers for instantaneous optimal control," *J. Eng. Mech.*, vol. 118, no. 8, pp. 1612-1630, Aug. 1992.
- [26] W. He, Y. T. Dong, and C. Y. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Trans. Syst. Man Cybern. -Syst.*, vol. 46, no. 3, pp. 334-344, Mar. 2016.
- [27] C. F. Chen, and N. R. Strader, "A normalized multidimensional Newton-Raphson method" *Int. J. Control*, vol. 12, no. 2, pp. 273-279, Mar. 1970.
- [28] S. J. Wright, *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 1997.

- [29] A. C. Billups, and K. G. Murty, "Complementarity problems," *J. Comput. Appl. Math.*, vol. 124, no. 1, pp. 303-318, Dec.2000.
- [30] C. Kanzow, "Some noninteriorcontinuation methods for linear complementarity problems," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 4, pp. 851-868, Oct. 1996.
- [31] W. X. Zhong, and R. L. Zhang, "Parametric variational-principles and their quadratic-programming solutions in plasticity," *Comput. Struct.*, vol. 30, no. 4, pp. 887-896, 1988.
- [32] R. Seydel, *Practical Bifurcation and Stability Analysis*, 3rd ed., Springer-Verlag, NewYork, 2010.
- [33] Z. Wang, Q. Tian, and H. Hu, "Dynamics of flexible multibody systems with hybrid uncertain parameters," *Mech. Mach. Theory*, vol. 121, pp. 128-147, 2018.
- [34] Q. J. Chen, H. T. Chen, Y. J. Wang, and P. Y. Woo, "Global stability analysis for some trajectory-tracking control schemes of robotic manipulators," *J. Robot. Syst.*, vol. 18, no. 2, pp. 69-75, 2001.