IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 18, 2020, accepted June 4, 2020, date of publication June 9, 2020, date of current version June 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001044

Model Predictive Control With Integral Compensation for Motion Control of Robot Manipulator in Joint and Task Spaces

YUHAN CHEN[1], **XIAO LUO**[2], **BAOLING HAN**[1], **QINGSHENG LUO**[3], **AND LIJUN QIAO**[3]
[1]School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China
[2]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[3]School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Xiao Luo (luox@bit.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2016YFC0803000 and Grant 2016YFC0803005.

ABSTRACT The motion control of robot manipulators is a crucial problem concerning automatically controlled robots. In this work, the model predictive control method with an integral compensation (MPC-I), which compensates for the matched uncertainties due to unmodeled dynamics, is proposed to solve the trajectory tracking problem of robot manipulators in joint and task spaces. First, this paper decouples the joint variables of the robot manipulator using a computed torque control method. The MPC-I method is, thereafter, derived to realize the motion control of the robot manipulators in joint space. To realize the motion control of the robot manipulator in task space, the task space is, thereafter, converted into the joint space, in which the MPC-I method is executed, afterward, to control the robot. Furthermore, an MPC-I variation, in which the inverse kinematics is calculated indirectly, is proposed to achieve the motion control in task space. The novelty of this paper is to propose the MPC-I method and the method of converting task space to joint space with indirect inverse kinematics calculation. The former is suitable for the dynamic control of the robot manipulators in the joint space, and the latter can extend the MPC-I method to dynamic control in task space. To evaluate the performance of the proposed control method, motion control simulations are performed in the task and joint spaces, respectively. Simulation results and comparisons verify the effectiveness of the proposed control approach for the dynamic control of the UR5 robot manipulator.

INDEX TERMS Model predictive control (MPC), motion control, robot manipulators, trajectory tracking.

I. INTRODUCTION

Due to existing demands to improve reliability, accuracy, repeatability, and productivity of industrial processes, the significance of robot technology has been highlighted [1]. Furthermore, robots are being used for different purposes, such as surgery [2], handling hazardous materials [3], and motion assistance [4], and in different environments, such as space [5], underwater [6] etc. In all these cases, robot manipulators are employed; therefore, understanding their motion control is crucial. The conventional control methods, such as proportional integral derivative (PID) [7] and linear quadratic regulator [8], can guarantee the reliability and safety of the system, realizing a stable and effective motion control of the manipulator. Their main advantage is their simplicity, as linear controllers are perfectly suited for numerous applications. However, some disadvantages, such as cumbersome gain tuning and adjustment requirements for implementation on a robot manipulator, are still not addressed. Furthermore, obtaining optimal parameters of classical PID controllers may be difficult, especially for systems comprising of high nonlinearities and couplings [9]. As a result, in the past few decades, advanced control technologies [10]–[16] have attracted more attention. The control techniques like Adaptive Control [10], [13], Sliding Mode Control [11], Robust Control [14]–[16] are very promising for implementation on a robot manipulator. Among them, the most representative [15] dealt with the robust task space control of electrically driven robot manipulators using voltage control strategy, which is simpler, less computational and requires less feedbacks than

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

the conventional robust controllers. However, model predictive control (MPC) is not only robust enough to eliminate the effects of nonlinear behavior, complexity and coupling effects in the dynamics, but also complies to possibly enforced constraints on the state and input variables, which is what the above control methods are not good at.

The MPC, also named receding horizon control, is recognized as the most efficient and application potential method, considering the advanced process control methodology [17]. In MPC, a process model is used to predict and optimize the future behavior of the system [18]. Future control inputs in certain control horizon and future plant responses in certain prediction horizon are both predicted using a system model and optimized at regular intervals with respect to a performance index. MPC has generated huge economic profits on thousands of industrial control systems over the world [19]. This kind of control has been first employed in large chemical factories; however, in recent years, it has been widely promoted in other industrial fields. For example, Paluszczyszyn *et al.* [20] has recently proposed this method to adjust the patient support system position. Meanwhile, Tadokoro *et al.* [21] uses this technology to adjust the motion control for rigid-body dynamical systems, such as spacecraft or aerial vehicles. Furthermore, other interesting results on MPC can be found in [22]–[26].

In recent years, MPC application in robot manipulator motion control has also been very popular; however, additional difficulties due to the nonlinearities of the manipulator occur. The most common solution of the problem is the linearization method, as illustrated in [27] and [28], in which the feedback of the inverse dynamics of the manipulator and the Taylor series are, respectively, used in the linearization. Furthermore, boundedness and stability convergence of the state tracking have also been proved in [28]. However, the common disadvantage between [27] and [28] is that compensation measures after linearization were not considered. The other solution is to combine MPC with other advanced control methods to realize the motion control of the manipulator. Reference [29] proposes a combination of MPC and H-infinity control, which is also a nonlinear design technique for a robot manipulator. The result of this combination, however, could not combine the respective advantages of these schemes while trying to avoid their shortcomings. Meanwhile, a robust hierarchical multiloop control scheme design to solve motion control problems for robot manipulators was proposed in [30]. The inverse dynamics-based feedback linearized robotic system and the combination of MPC with an integral sliding mode controller are the key elements of this control approach. The internal control loop was used to compensate for the matched uncertainties due to unmodeled dynamics. This method was verified and validated by simulations; however, the predictive control method is only applicable to the motion control in joint space but not to the motion control in task space that is the workspace of the end-effector.

Based on prior research, MPC variation methods that can be applied to both joint and task spaces of robot manipulators have been rarely found. As a result, this paper proposes an model predictive control with integral compensation (MPC-I) for the motion control of the robot manipulator that can be used in the joint and task spaces. Motion control in joint space was first decoupled by a computed torque control method [31]. Afterward, an approximate prediction was performed using Taylor expansion. Due to the truncation error of the Taylor expansion, an integral compensation is introduced to realize the motion control. To realize motion control in task space, the task space is first transformed into the joint space and, thereafter, the MPC-I in joint space is executed. The task space can be transformed into the joint space by two different procedures. In the first one, the inverse kinematics are calculated directly, whereas, in the second one, which is to calculate inverse kinematics indirectly. The latter has a low computational complexity and is not affected by the configuration of the robot manipulator. Both procedures were compared in this work with simulations.

The rest of the paper is organized in the following manner. In Section II, preliminary knowledge, including the robotic dynamics in joint space and exponential coordinate representation of rigid-body motions are discussed. In Section III, the MPC-I methods that realizes the motion control of the robot manipulator in the joint and task spaces are introduced. In Section IV, the simulation results, which were obtained from the co-simulation of the UR5 manipulator on MATLAB and V-rep [32] platforms, are exhibited. Finally, in Section V, conclusions and future work are informed.

II. PRELIMINARIES

In this section, the preliminary knowledge used in this article was introduced. First, the robotic dynamics in joint space was described. The exponential coordinate representation of rigid-body motions was, thereafter, described, which lays the foundation for the conversion of the task space to the joint space in the Section III-B2, and facilitates the definition of rotation and translation errors in the Section IV-B.

A. THE ROBOTIC DYNAMICS IN JOINT SPACE

The dynamic equation for n degree of freedom robot manipulators can be written as:

$$\boldsymbol{M}(\theta)\ddot{\theta} + \boldsymbol{C}(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau, \tag{1}$$

where $\boldsymbol{M}(\theta) \in \mathbb{R}^{n\times n}$ is the positive definite mass matrix, $\boldsymbol{C}(\theta, \dot{\theta}) \in \mathbb{R}^{n\times n}$ is called the Coriolis matrix, $g(\theta) \in \mathbb{R}^{n\times 1}$ is the vector specifying the effects due to gravity, $\tau \in \mathbb{R}^{n\times 1}$ is the vector of torques (forces) of each joint actuator, and $\theta = [\theta_1\ \theta_2\ \ldots\ \theta_n]^T \in \mathbb{R}^{n\times 1}$ is the vector of joint variables. Furthermore, $\ddot{\theta}$ can be expressed as the following to calculate the acceleration of the joint variables:

$$\ddot{\theta} = \boldsymbol{M}(\theta)^{-1}\left(\tau - h(\theta, \dot{\theta})\right), \tag{2}$$

where $h(\theta, \dot{\theta})$ is defined as:

$$h(\theta, \dot{\theta}) = \boldsymbol{C}(\theta, \dot{\theta})\dot{\theta} + g(\theta). \tag{3}$$

B. EXPONENTIAL COORDINATE REPRESENTATION OF RIGID-BODY MOTIONS

The product-of-exponentials formula used to describe the kinematics of robot manipulators is introduced by [33]. Meanwhile, the Chasles-Mozzi theorem states that every rigid-body displacement can be expressed as a displacement along a fixed screw axis in space [34]. By combining this theorem with the product of the exponential formula, a six-dimensional exponential coordinate (S) of a homogeneous transformation ($\boldsymbol{T}_h$) can be defined as:

$$S = \begin{bmatrix} S_a \\ S_p \end{bmatrix}, \tag{4}$$

where

$$S_a = [S_{a1}, S_{a2}, S_{a3}]^T, \tag{5}$$

$$S_p = [S_{p1}, S_{p2}, S_{p3}]^T, \tag{6}$$

where S_a and S_p represent the rotation and translation, respectively. The matrix are therefore defined by:

$$\boldsymbol{T}_h = \exp([S]), \tag{7}$$

$$[S] = \log(\boldsymbol{T}_h), \tag{8}$$

where $\boldsymbol{T}_h$ belongs to Special Euclidean Group, named $SE(3)$, and $[S]$, which belongs to $se(3)$, the Lie algebra of the Lie group $SE(3)$, is defined as:

$$[S] = \begin{bmatrix} 0 & -a_3 & a_2 & p_1 \\ a_3 & 0 & -a_1 & p_2 \\ -a_2 & a_1 & 0 & p_3 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{9}$$

The specific calculation process of (7) and (8) is described by [33].

III. MOTION CONTROL WITH TORQUE OR FORCE INPUTS

Due to the nonlinearities of the robot manipulator dynamics, motion control with torque or force inputs is more difficult than motion control with velocity inputs. However, the latter is generally limited to applications with low or predictable force/torque requirements, the former is suitable for more applications. In this section, therefore, a control method that generates joint torques or forces to try to track the desired trajectory in joint and task spaces is discussed.

A. MPC-I METHOD OF ROBOT MANIPULATORS IN JOINT SPACE

In this section, the MPC-I method for the motion control of the robot manipulators in joint space is introduced, and the prediction equation required by this method is easily expressed in this space. Trajectories are also naturally described by the joint variables, and there are no issues of singularities or redundancy [33].

First, the joint variables are decoupled by a computed torque method. Afterward, an equation that predicts the behavior of the joint variables is obtained. The objective function is, thereafter, calculated, and the optimization problem is solved. Moreover, the stability analysis of the control method is performed. Finally, integral compensation is introduced to eliminate the steady state error.

The following assumptions were made before calculating the appropriate solution:

1) The MPC controller used in this paper is based on the solution of the so-called finite horizon optimal control problem. Therefore, the prediction and control horizon were defined within p and m steps (where $p \geq m$), respectively, and the time interval of each step is Δt.
2) For discrete systems, the acceleration of joint variables between steps k and $k+1$ was considered constant.
3) Beyond the control horizon, the control quantity was preserved.

$$\ddot{\theta}(k+i-1|k) = \ddot{\theta}(k+i|k),\ i = m, m+1, \ldots, p-1,$$

where $k+i|k$ represents the prediction of the $k+i$ step at k step.

1) PREDICTION OF JOINT VARIABLES

According to these assumptions, the prediction formula of joint variables could be written as

$$\Theta(k+1|k) = \boldsymbol{I}_a\theta(k) + \boldsymbol{I}_b\dot{\theta}(k) + \boldsymbol{A}\ddot{\Theta}(k), \tag{10}$$

where

$$\Theta(k+1|k) \triangleq [\theta(k+1|k)^T\ \theta(k+2|k)^T\ \ldots\ \theta(k+p|k)^T]^T_{p\times 1}, \tag{11}$$

$$\ddot{\Theta}(k) \triangleq [\ddot{\theta}(k)^T\quad \ddot{\theta}(k+1)^T\quad \ldots\quad \ddot{\theta}(k+m-1)^T]^T_{m\times 1}, \tag{12}$$

$$\boldsymbol{I}_a \triangleq [\boldsymbol{I}\quad \boldsymbol{I}\quad \ldots\quad \boldsymbol{I}]^T_{p\times 1}, \tag{13}$$

$$\boldsymbol{I}_b \triangleq [\boldsymbol{I}\quad 2\boldsymbol{I}\quad \ldots\quad p\boldsymbol{I}]^T_{p\times 1}\Delta t, \tag{14}$$

$$\boldsymbol{A} \triangleq \begin{bmatrix} \frac{1}{2}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \frac{3}{2}\boldsymbol{I} & \frac{1}{2}\boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{2m-1}{2}\boldsymbol{I} & \frac{2m-3}{2}\boldsymbol{I} & \frac{2m-5}{2}\boldsymbol{I} & \ldots & \frac{1}{2}\boldsymbol{I} \\ \frac{2m+1}{2}\boldsymbol{I} & \frac{2m-1}{2}\boldsymbol{I} & \frac{2m-3}{2}\boldsymbol{I} & \ldots & \frac{3}{2}\boldsymbol{I} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{2p-1}{2}\boldsymbol{I} & \frac{2p-3}{2}\boldsymbol{I} & \frac{2p-5}{2}\boldsymbol{I} & \ldots & \frac{2p-2m+1}{2}\boldsymbol{I} \end{bmatrix}_{p\times m} \Delta t^2, \tag{15}$$

where $\boldsymbol{I}$ and $\boldsymbol{0}$ denote the $(n \times n)$ identity matrix and $(n \times n)$ null matrix, respectively, and the matrix subscript represents the number of elements rather than the dimension of the matrix.

According to (2), $\ddot{\theta}$ is coupled with τ. These variables were decoupled with a computed torque control method. As a result, the torque or force at each step was defined as:

$$\tau(k+i|k) = \boldsymbol{M}(k+i|k)w(k+i|k) + h(k+i|k). \tag{16}$$

By combining (16) and (2), the following expression can be obtained:

$$\ddot{\theta}(k+i|k) = w(k+i|k). \tag{17}$$

As a result, (10) can be written as:

$$\Theta(k+1|k) = \boldsymbol{I}_a\theta(k) + \boldsymbol{I}_b\dot{\theta}(k) + \boldsymbol{A}W(k), \tag{18}$$

where

$$W(k) \triangleq [w(k)^T \ w(k+1)^T \ \ldots \ w(k+m-1)^T]^T_{m\times 1}. \tag{19}$$

2) CLOSED LOOP SOLUTION OF MPC

The optimization problem can be described as

$$W(k)^* = \arg\min_{W(k)} J(\theta(k), \dot{\theta}(k), W(k), m, p), \tag{20}$$

where the objective function can be defined as

$$J(\theta(k),\dot{\theta}(k),W(k),m,p) = ||\boldsymbol{\Gamma}_\theta\left(\Theta(k+1|k) - \Theta_d(k+1)\right)||^2 + ||\boldsymbol{\Gamma}_w W(k)||^2, \tag{21}$$

where nonnegative weighting matrix and the expected target joint vector were evaluated by equations (22), (23), and (24), respectively.

$$\boldsymbol{\Gamma}_\theta = \mathrm{diag}(\boldsymbol{\Gamma}_{\theta,1}, \boldsymbol{\Gamma}_{\theta,2}, \ldots, \boldsymbol{\Gamma}_{\theta,p}) \tag{22}$$

$$\boldsymbol{\Gamma}_w = \mathrm{diag}(\boldsymbol{\Gamma}_{w,1}, \boldsymbol{\Gamma}_{w,2}, \ldots, \boldsymbol{\Gamma}_{w,m}) \tag{23}$$

$$\Theta_d \triangleq [\theta_d(k+1)^T \ \theta_d(k+2)^T \ \ldots \ \theta_d(k+p)^T]^T_{p\times 1} \tag{24}$$

Injecting (18) in (21), the objective function was modified to:

$$J = W(k)^T\boldsymbol{H}W(k) - F(k+1|k)^T W(k) + Y(k+1|k), \tag{25}$$

where

$$\boldsymbol{H} = \boldsymbol{A}^T\boldsymbol{\Gamma}_\theta^T\boldsymbol{\Gamma}_\theta\boldsymbol{A} + \boldsymbol{\Gamma}_w^T\boldsymbol{\Gamma}_w, \tag{26}$$

$$F(k+1|k) = 2\boldsymbol{A}^T\boldsymbol{\Gamma}_\theta^T\boldsymbol{\Gamma}_\theta E_\theta(k+1|k), \tag{27}$$

$$Y(k+1|k) = E_\theta(k+1|k)^T\boldsymbol{\Gamma}_\theta^T\boldsymbol{\Gamma}_\theta E_\theta(k+1|k), \tag{28}$$

where

$$E_\theta(k+1|k) \triangleq \Theta_d(k+1) - \boldsymbol{I}_a\theta(k) - \boldsymbol{I}_b\dot{\theta}(k) \tag{29}$$

To solve (20), we calculated the derivative of (25) concerning $W(k)$.

$$\frac{\partial J}{\partial W(k)} = 2\boldsymbol{H}W(k) - F(k+1|k) = 0 \tag{30}$$

$$\frac{\partial^2 J}{\partial W(k)^2} = 2\boldsymbol{H} > 0. \tag{31}$$

As a result, the solution of (20) was obtained:

$$W(k)^* = \frac{1}{2}\boldsymbol{H}^{-1}F(k+1|k), \tag{32}$$

where $\boldsymbol{H}$ and $F(k+1|k)$ were defined by (26) and (27), respectively.

According to the MPC method, the first element of $W(k)^*$ should be applied to the system:

$$w(k)^* = \boldsymbol{K}_{\mathrm{mpc}}E_\theta(k+1|k), \tag{33}$$

where $E_\theta(k+1|k)$ is defined by (29) and

$$\boldsymbol{K}_{\mathrm{mpc}} = \boldsymbol{I}_e(\boldsymbol{A}^T\boldsymbol{\Gamma}_\theta^T\boldsymbol{\Gamma}_\theta\boldsymbol{A} + \boldsymbol{\Gamma}_w^T\boldsymbol{\Gamma}_w)^{-1}\boldsymbol{A}^T\boldsymbol{\Gamma}_\theta^T\boldsymbol{\Gamma}_\theta, \tag{34}$$

where

$$\boldsymbol{I}_e = [\boldsymbol{I} \quad \boldsymbol{0} \quad \ldots \quad \boldsymbol{0}]_{1\times m}. \tag{35}$$

As a result, according to (16), the control torque/force was

$$\tau^*(k) = \boldsymbol{M}(k)w(k)^* + h(k), \tag{36}$$

where $w(k)^*$ and $h(k)$ are defined as (33) and (3), respectively.

3) STABILITY ANALYSIS OF THE CONTROL METHOD IN JOINT SPACE

This section discusses the stability of the methods proposed above. The authors adopted a similar concept as reported by [28] to conduct this analysis.

Using (36) and (29) in (2), the acceleration of the joint variables could be calculated as:

$$\ddot{\theta}(k) = \boldsymbol{K}_{\mathrm{mpc}}(\Theta_d(k+1) - \boldsymbol{I}_a\theta(k) - \boldsymbol{I}_b\dot{\theta}(k)), \tag{37}$$

where $\Theta_d(k+1)$ can be written as

$$\Theta_d(k+1) = \boldsymbol{I}_a\theta_d(k) + \boldsymbol{I}_b\dot{\theta}_d(k) + \boldsymbol{A}\ddot{\theta}_d(k) + O(\Delta t^3), \tag{38}$$

where $\boldsymbol{I}_a$, $\boldsymbol{I}_b$, and $\boldsymbol{A}$ are defined as (13), (14), and (15), respectively, and $O(\Delta t^3)$ is truncation error. Furthermore, the joint variable error θ_e is defined as

$$\theta_e = \theta_d - \theta. \tag{39}$$

Therefore, the joint variable acceleration error ($\ddot{\theta}_e$) can be calculated as

$$\begin{aligned}\ddot{\theta}_e(k) &= \ddot{\theta}_d(k) - \ddot{\theta}(k)\\ &= \ddot{\theta}_d(k) - \boldsymbol{K}_{\mathrm{mpc}}(\Theta_d(k+1) - \boldsymbol{I}_a\theta(k) - \boldsymbol{I}_b\dot{\theta}(k)).\end{aligned} \tag{40}$$

By combining (38) with (40), the joint variable error dynamic could be obtained:

$$\begin{cases}\dot{e}_1 = e_2\\ \dot{e}_2 = -\boldsymbol{K}_{\mathrm{mpc}}\boldsymbol{I}_a e_1 - \boldsymbol{K}_{\mathrm{mpc}}\boldsymbol{I}_b e_2\\ \qquad -(\boldsymbol{K}_{\mathrm{mpc}}\boldsymbol{A} - \boldsymbol{I})\ddot{\theta}_d - \boldsymbol{K}_{\mathrm{mpc}}O(\Delta t^3),\end{cases} \tag{41}$$

where $e_1 = \theta_e$ and $e_2 = \dot{\theta}_e$. Moreover, the error state (e) can be defined as:

$$e = [e_1^T \ e_2^T]^T. \tag{42}$$

As a result, the tracking problem of θ_e and $\dot{\theta}_e$ is reduced to the regulation problem of e that was treated as a linear system with non vanishing perturbation [35], which are described by the following equations:

$$\dot{e}(t) = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\\ -\boldsymbol{K}_{\mathrm{mpc}}\boldsymbol{I}_a & -\boldsymbol{K}_{\mathrm{mpc}}\boldsymbol{I}_b\end{bmatrix}e(t) + C(\ddot{\theta}_d, \boldsymbol{K}_{\mathrm{mpc}}, \Delta t), \tag{43}$$

where the perturbed term is

$$C(\ddot{\theta}_d, \boldsymbol{K}_{\text{mpc}}, \Delta t) = \begin{bmatrix} 0 \\ -(\boldsymbol{K}_{\text{mpc}}\boldsymbol{A} - \boldsymbol{I})\ddot{\theta}_d - \boldsymbol{K}_{\text{mpc}}O(\Delta t^3) \end{bmatrix}. \tag{44}$$

The origin may not be an equilibrium point of system (41); therefore, the stability of the system at the origin was not analyzed. Furthermore, the solution of the perturbed system may not approach the origin when $t \rightarrow \infty$. However, the boundedness of the system (41) could be analyzed.

Lyapunov function candidate is defined as

$$V = \frac{1}{2} e_1^T \boldsymbol{Q}\boldsymbol{K}_{mpc} I_a e_1 + \frac{1}{2} e_2^T \boldsymbol{Q} e_2, \tag{45}$$

where both $\boldsymbol{Q}\boldsymbol{K}_{mpc}\boldsymbol{I}_a$ and $\boldsymbol{Q}$ are positive definite matrixes. By deriving the Lyapunov function concerning time, (45) was transformed into:

$$\dot{V} = e_1^T \boldsymbol{Q}\boldsymbol{K}_{\text{mpc}}\boldsymbol{I}_a e_2 + e_2^T \boldsymbol{Q} \dot{e}_2. \tag{46}$$

After combining (41) with (46) the following expression was obtained:

$$\dot{V} = -e_2^T \boldsymbol{Q}\boldsymbol{K}_{\text{mpc}}\boldsymbol{I}_b e_2 - e_2^T \boldsymbol{Q} \left((\boldsymbol{K}_{\text{mpc}}\boldsymbol{A} - \boldsymbol{I})\ddot{\theta}_d + \boldsymbol{K}_{\text{mpc}} O(\Delta t^3) \right) \tag{47}$$

There exist an lower bound $\underline{\delta}_1(\ddot{\theta}_d)$, $\underline{\delta}_2(\Delta t)$, and positive reals t_{p1}, t_{p2} for all e_2:

$$\left|\left|(\boldsymbol{K}_{\text{mpc}}\boldsymbol{A} - \boldsymbol{I})\ddot{\theta}_d\right|\right| \geq \bar{\delta}(\ddot{\theta}_d) = t_{p1} \, ||e_2||^2 \tag{48}$$

$$\left|\left|\boldsymbol{K}_{\text{mpc}} O(\Delta t^3)\right|\right| \geq \bar{\delta}(\Delta t) = t_{p2} \, ||e_2||^2 \tag{49}$$

The time derivative of V is bounded as follows:

$$\dot{V} \leq -\lambda_{\min}(\boldsymbol{Q}\boldsymbol{K}_{\text{mpc}}\boldsymbol{I}_b)||e_2||^2 - (t_{p1} + t_{p2})\lambda_{\min}(\boldsymbol{Q})||e_2||^2 . \tag{50}$$

As a result, if $\lambda_{\min}(\boldsymbol{Q}\boldsymbol{K}_{\text{mpc}}\boldsymbol{I}_b) + (t_{p1} + t_{p2})\lambda_{\min}(\boldsymbol{Q}) > 0$, the solutions of the system expressed by (41) are uniformly bounded. The dynamic of the tracking error (41) is specified by the matrix $\boldsymbol{Q}$ and parameters $\Delta t, \ddot{\theta}_d$.

Because this bound could be made small by reducing the penalty of the control signal $\boldsymbol{\Gamma}_w \rightarrow 0$ in (21). With $\boldsymbol{\Gamma}_w = 0$, that will lead to

$$\boldsymbol{K}_{\text{mpc}}\boldsymbol{A} - \boldsymbol{I} = \boldsymbol{0}. \tag{51}$$

As a result, $t_{p1} = 0$ and the truncation error $O(\Delta t^3)$ can be eliminated by introducing integral compensation, which will be introduced in the next section. Considering this equation, the origin is, therefore, the equilibrium point of the system (41) and the time derivative of the Lyapunov function could be modified to:

$$\dot{V} = -e_2^T \boldsymbol{Q}\boldsymbol{K}_{\text{mpc}} I_b e_2 \tag{52}$$

As $\boldsymbol{Q}\boldsymbol{K}_{\text{mpc}}\boldsymbol{I}_b$ is positive definite matrix, the equilibrium point $e^o = [0 \ 0]^T$ is globally asymptotically stable, according to La Salle's invariant theorem [35].

4) INTEGRAL COMPENSATION

In the previous sections, the MPC method was introduced and its stability was proved. It is obvious that the truncation error is caused by the matched uncertainties due to unmodeled dynamics. To eliminate the truncation error, this section demonstrates the necessity of introducing integral compensation in the previously proposed MPC method.

To introduce the integral compensation, (29) is modified as follow,

$$E_\theta^*(k+1|k) \triangleq \Theta_d(k+1) - \boldsymbol{I}_a\theta(k) - \boldsymbol{I}_b\dot{\theta}(k) + \boldsymbol{K}_I \int_0^k \Theta_e dt, \tag{53}$$

where the integral gain $\boldsymbol{K}_I$ is a positive definite matrix:

$$\boldsymbol{K}_I \triangleq \text{diag}(\boldsymbol{K}_{I,1}, \boldsymbol{K}_{I,2}, \ldots, \boldsymbol{K}_{I,p})/\Delta t \tag{54}$$

and $\Theta_e(i)$ can be calculated as

$$\Theta_e(i) = \begin{bmatrix} 3\theta_e(i)^T & 5\theta_e(i)^T & \ldots & (2p+1)\theta_e(i)^T \end{bmatrix}_{p \times 1}^T . \tag{55}$$

Therefore, the $\ddot{\theta}_e$ can be calculated as

$$\begin{aligned} \ddot{\theta}_e(k) = \ddot{\theta}_d(k) - \boldsymbol{K}_{\text{mpc}}(&\boldsymbol{I}_a\theta_e(k) + \boldsymbol{I}_b\dot{\theta}_e(k) \\ &+ \boldsymbol{A}\ddot{\theta}_d(k) + \boldsymbol{K}_I \int_0^k \Theta_e dt + O(\Delta t^3)). \end{aligned} \tag{56}$$

Then we calculated the derivative of (56) concerning time, the joint variable error dynamic could be obtained:

$$\begin{cases} \dot{e}_1 = e_2 \\ \dot{e}_2 = e_3 \\ \dot{e}_3 = -\boldsymbol{K}_{\text{mpc}}\boldsymbol{K}_I e_1 - \boldsymbol{K}_{\text{mpc}}\boldsymbol{I}_a e_2 - \boldsymbol{K}_{\text{mpc}}\boldsymbol{I}_b e_3 \\ \qquad -(\boldsymbol{K}_{\text{mpc}}\boldsymbol{A} - \boldsymbol{I})\dddot{\theta}_d - \boldsymbol{K}_{\text{mpc}}O(\Delta t^4), \end{cases} \tag{57}$$

where $e_3 = \ddot{\theta}_e$. Moreover, the error state with integral compensation (e_I) can be defined as:

$$e_I = [e_1^T \ e_2^T \ e_3^T]^T . \tag{58}$$

Then (43) can be modified as

$$\begin{aligned} \dot{e}_I(t) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K}_{\text{mpc}}\boldsymbol{K}_I & -\boldsymbol{K}_{\text{mpc}}\boldsymbol{I}_a & -\boldsymbol{K}_{\text{mpc}}\boldsymbol{I}_b \end{bmatrix} e_I(t) \\ + C_I(\ddot{\theta}_d, \boldsymbol{K}_{\text{mpc}}, \Delta t), \end{aligned} \tag{59}$$

where the perturbed term is

$$C_I(\ddot{\theta}_d, \boldsymbol{K}_{\text{mpc}}, \Delta t) = \begin{bmatrix} 0 \\ -(\boldsymbol{K}_{\text{mpc}}\boldsymbol{A} - \boldsymbol{I})\dddot{\theta}_d - \boldsymbol{K}_{\text{mpc}}O(\Delta t^4) \end{bmatrix}. \tag{60}$$

The proof of the stability of (59) is similar to the method in Section III-A3, which will not be repeated here. Below we will mainly analyze the perturbation term (60). When $\ddot{\theta}_d$ is a constant, we can get $\dddot{\theta}_d = 0$. When $\ddot{\theta}_d$ is a variable, we can set $\boldsymbol{\Gamma}_w = 0$, that will lead to $\boldsymbol{K}_{\text{mpc}}\boldsymbol{A} - \boldsymbol{I} = \boldsymbol{0}$. Δt is very small, $O(\Delta t^4)$ in (60) is much smaller than $O(\Delta t^3)$ in (44), so $O(\Delta t^4)$ can even be ignored. Therefore, the integral compensation makes the perturbed term (60) much smaller than (44), which effectively eliminates the truncation error.

Therefore, the final equation that represents the MPC-I method can be obtained as:

$$\tau_I^*(k) = \boldsymbol{M}(k)w_I^*(k) + h(k), \tag{61}$$

where $h(k)$ is defined as (3) and

$$w_I^*(k) = \boldsymbol{K}_{\text{mpc}}E_\theta^*(k+1|k), \tag{62}$$

where $\boldsymbol{K}_{\text{mpc}}$ and $E_\theta^*(k+1|k)$ are defined as (34) and (53), respectively.

5) CONTROL METHOD OF MOTION IN JOINT SPACE

In this section, the detailed process of the MPC-I method to realize the motion control of the robot manipulators in joint space is described.

The MPC-I controller block diagram is illustrated in Fig. 1. It should be noted that $\boldsymbol{K}$ is to transform θ_e into Θ_e. Furthermore, the detailed control process is shown in Algorithm 1. In line 1, the integral compensation S_{θ_e} is the integral of joint variables errors. In line 2, N is the maximum number of control steps. In lines 5 and 6, the joint variables error $\theta_e(k)$ and $\Theta_e(k)$ are calculated as (39) and (55), respectively. From line 8 to 10, $E_\theta^*(k+1|k)$, $\boldsymbol{K}_{mpc}$, and $w_I^*(k)$ are calculated as (53), (34), and (62), respectively. In line 11, the dynamics parameters $h(k)$ is calculated as (3), but in order to reduce the inaccuracy of the estimation, the θ and $\dot{\theta}$ used to calculate $\boldsymbol{M}(k)$ and $h(k)$ are replaced by the follows

$$\theta = \theta(k) + \dot{\theta}(k)\frac{\Delta t}{2} + \frac{1}{2}w_I^*(k)\left(\frac{\Delta t}{2}\right)^2 \tag{63}$$

$$\dot{\theta} = \dot{\theta}(k) + w_I^*(k)\frac{\Delta t}{2} \tag{64}$$

In line 12, the control torque/force $\tau_I^*(k)$ is calculated as (61).

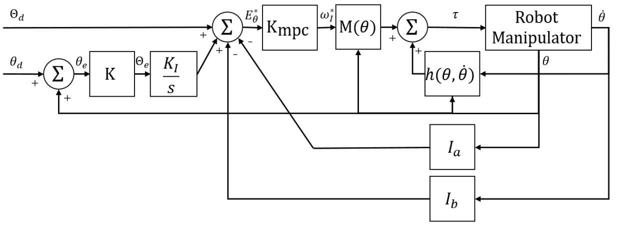


FIGURE 1. The block diagram of the MPC-I controller that produces a commanded torque or force as input to the robot manipulator.

B. MPC-I METHOD OF ROBOT MANIPULATORS IN TASK SPACE

As the robot interacts with the external environment and multiple objects, expressing the motion of the robot as a trajectory of the end-effector in task space is more convenient. To perform this task, the trajectory in task space must be converted to joint space.

Provided the corresponding trajectory in joint space is feasible, this conversion may be done by direct or indirect inverse kinematics calculation. Afterward, the control method in Section III-A5 be used to realize the motion of the robot.

Algorithm 1 The MPC-I Method to Control the Robot Manipulator in Joint Space

```
1:  Initialization: Determine the time step $\Delta t$, the prediction horizon $p$, the control horizon $m$, the nonnegative weighting matrix $\boldsymbol{\Gamma}_\theta$, $\boldsymbol{\Gamma}_w$, $\boldsymbol{K}_I$, and set the integral compensation $S_{\theta_e} = 0$
2:  for each $k \in [0, N]$ do
3:      Read the current the robot manipulator states $\theta(k)$ and $\dot{\theta}(k)$
4:      Determine the expected target joint variables vector $\Theta_d(k+1)$
5:      Calculate the joint variables error $\theta_e(k)$
6:      Determine $\Theta_e(k)$
7:      Calculate the error integral $S_{\theta_e} = S_{\theta_e} + \Theta_e(k)\Delta t$
8:      Determine $E_\theta^*(k+1|k)$
9:      Calculate $\boldsymbol{K}_{\text{mpc}}$
10:     Determine $w_I^*(k)$
11:     Calculate the dynamics parameters $\boldsymbol{M}(k)$ and $h(k)$
12:     Determine the control torque/force $\tau_I^*(k)$ and act on the robot manipulator
13: end for
```

1) THE CONTROL METHOD OF DIRECTLY CALCULATING INVERSE KINEMATICS

The end-effector trajectory is specified by $\boldsymbol{T}_d$, where $\boldsymbol{T}_d \in SE(3)$. The inverse kinematics can transform the end-effector trajectory into a trajectory based on joint variables by the following expression,

$$\theta_d = f^{-1}(\boldsymbol{T}_d), \tag{65}$$

where f^{-1} represents the inverse kinematics calculation. Then proceed with control as in Section III-A5.

The MPC-I method in which directly calculates the inverse kinematics (MPC-I_IK), not only needs significant computing power to calculate the inverse kinematics, but also is affected by the configuration of the robot manipulator.

2) THE CONTROL METHOD INDIRECTLY CALCULATING INVERSE KINEMATICS

To overcome the shortcomings of the method described in the previous section, the MPC-I method in which the inverse kinematics are indirectly calculated (MPC-I_NIK) is introduced in this section.

By assuming that the end-effector frame can be expressed with a coordinate vector (x) governed by the kinematics $x_d = g(\theta_d)$, the forward kinematics can be expressed as the Taylor expansion

$$\begin{aligned} x_d(k+1) &= g(\theta_d(k+1)) \\ &= g(\theta_d(k)) + \underbrace{\left.\frac{\partial g}{\partial \theta_d}\right|_{\theta_d(k)}}_{\boldsymbol{J}(\theta_d(k))}(\theta_d(k+1) - \theta_d(k)) + \text{h.o.t.}, \end{aligned} \tag{66}$$

where $\boldsymbol{J}(\theta_d(k))$ is the coordinate Jacobian evaluated at $\theta_d(k)$. After performing the first-order truncation, (66) can be approximated as:

$$\theta_d(k+1) = \theta_d(k) + \boldsymbol{J}(\theta_d(k))^{\dagger}(x_d(k+1) - g(\theta_d(k))), \quad (67)$$

where the $\boldsymbol{J}(\theta_d(k))^{\dagger}$ denotes the pseudo-inverse of the Jacobian.

To modify (67) to work with a desired end-effector configuration represented as $\boldsymbol{T}_d \in SE(3)$ instead of as a coordinate vector x_d, the coordinate Jacobian J was replaced with the end-effector body Jacobian $\boldsymbol{J}_b \in \mathbb{R}^{6\times n}$ and $x_d(k+i) - g(\theta_d(k+i-1))$ was replaced with a body twist V_b, which would generate the motion from $\boldsymbol{T}_d(k+i-1)$ to the desired configuration $\boldsymbol{T}_d(k+i)$. Furthermore, $V_b(k+i-1)$ is determined by the matrix logarithm introduced in Section II-B:

$$[V_b(k+i-1)] = \log(\boldsymbol{T}_d(k+i-1)^{-1}\boldsymbol{T}_d(k+i)). \quad (68)$$

As a result, (67) can be modified to:

$$\theta_d(k+i) = \theta_d(k+i-1) + \boldsymbol{J}_b(\theta_d(k+i-1))^{\dagger} V_b(k+i-1) \quad (69)$$

The method proposed in this section is fully described by Algorithm 2. Lines 1 to 3 of Algorithm 2 are similar to Algorithm 1. From line 4 to 12, the methods to convert the desired pose of the task space $\boldsymbol{T}_d$ into joint variables θ_d without calculating kinematics are used. In lines 8, 9, and 12, the body twist V_b, the expected joint variables of the next step $\theta_d(k+i)$, and the expected target joint vector are calculated with (68), (69), and (24), respectively. In line 13, the expected joint variables at the current moment $\theta_d(k)$ are approximately calculated with (70) and (71), as a preliminary step for the calculation of the integral error. As a result, the estimated value of the expected joint variable for step $k+1$ at step k is the expected joint variable for the next step. From line 14 to 20, the MPC-I method of a robot manipulator in joint space is executed.

IV. SIMULATION AND DISCUSSION

To illustrate the effectiveness of the MPC-I method proposed in this article, the motion control of the robot manipulators in joint and task spaces is individually simulated. Considering a UR5 robot manipulator as an example, the co-simulation is conducted on the MATLAB 2015b and V-rep 4.0 platforms. The product-of-exponentials parameters of the UR5 with the current configurations relative to the base frame are shown in Table 1, where $[\omega_i^T \; \nu_i^T]^T$ are the screw axes expressed in the fixed space frame. $\boldsymbol{T}_{h0}$ is the end-effector configuration when the UR5 is at its home position. All simulations are performed on a desktop computer (Core i5 3.40GHz, 16GB RAM, MATLAB 2019b and V-rep 4.0 software platforms).

A. COMPARISON OF MOTION CONTROL BETWEEN MPC AND MPC-I METHODS IN JOINT SPACE

To prove the benefits of the integral compensation, the trajectory tracking of the UR5 robot manipulator in joint space was implemented. The UR5 robot manipulator will take $8s$ to go from the initial angle θ_{init} to the end angle θ_{end}. The specific values of the initial and the end angle of UR5 are shown in Table 2. The expected motion trajectory is, thereafter,

Algorithm 2 The MPC-I Method to Control the Robot Manipulator in Task Space

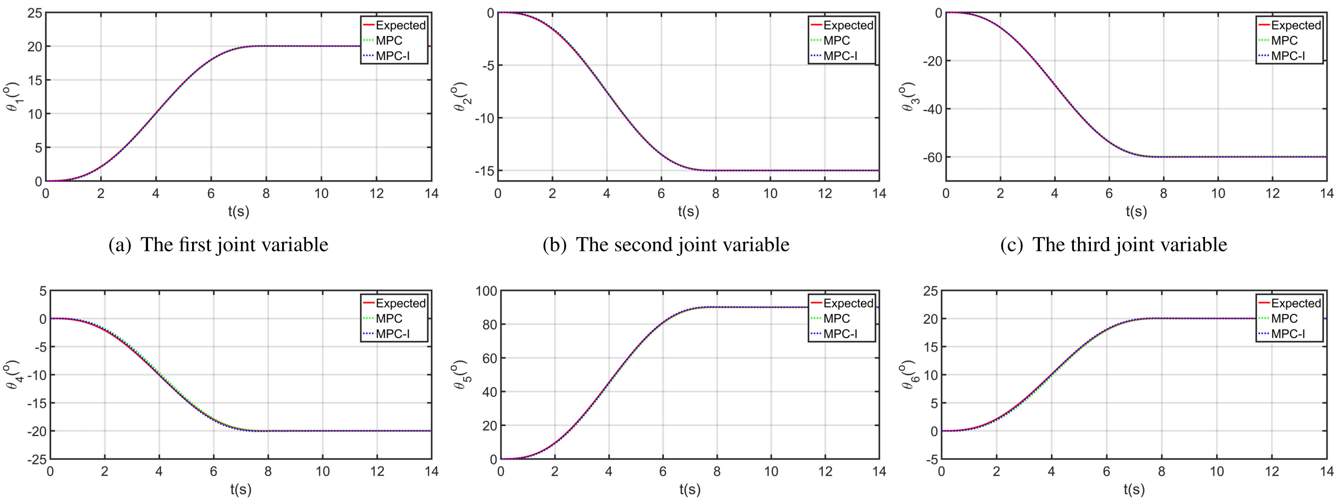
```
1:  Initialization: Determine the time step $\Delta t$, the prediction
    horizon $p$, the control horizon $m$, the nonnegative weighting
    matrix $\boldsymbol{\Gamma}_\theta$, $\boldsymbol{\Gamma}_w$, $\boldsymbol{K}_I$, and set the integral compensation
    $S_{\theta_e} = 0$ and $\theta_{d0} = \theta(0)$
2:  for each $k \in [0, N]$ do
3:      Read the current robot manipulator states
        $\theta(k)$ and $\dot{\theta}(k)$
4:      Calculate the pose transformation matrix $\boldsymbol{T}_h(k)$ using
        forward kinematics and set $\boldsymbol{T}_d(k) = \boldsymbol{T}_h(k)$
5:      Determine the end-effector body Jacobian $\boldsymbol{J}_b(k|k)$
6:      for each $i \in [1, p]$ do
7:          Read expected pose transformation matrix $\boldsymbol{T}_d(k+i)$
8:          Calculate the body twist $[V_b(k+i-1)]$
9:          Determine the expected joint variables $\theta_d(k+i)$
10:         Calculate the end-effector body Jacobian $\boldsymbol{J}_b(k+i|k)$
11:     end for
12:     Determine the expected target joint vector $\Theta_d(k+1)$
13:     Calculate the expected joint variables at the current
        moment
            $\theta_d(k) = \theta_{d0},$                          (70)
        then set
            $\theta_{d0} = \theta_d(k+1)$                        (71)
14:     Determine $\Theta_e(k)$
15:     Calculate the error integral $S_{\theta_e} = S_{\theta_e} + \Theta_e(k)\Delta t$
16:     Determine $E_\theta^*(k+1|k)$
17:     Calculate $\boldsymbol{K}_{\text{mpc}}$
18:     Determine $w_I^*(k)$
19:     Calculate the dynamics parameters $\boldsymbol{M}(k)$ and $h(k)$
20:     Determine the control torque/force $\tau_I^*(k)$ and act on
        the robot manipulator
21: end for
```

TABLE 1. The product-of-exponentials parameters of the UR5.

i	ω_i^T	ν_i^T	$\boldsymbol{T}_{h0}$
1	$[0 \;\; 0 \;\; 1]$	$[0 \;\; 0 \;\; 0]$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0.3819 \\ 0 & 0 & 1 & 0.9866 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
2	$[0 \;\; -1 \;\; 0]$	$[0.0746 \;\; 0 \;\; 0]$	
3	$[0 \;\; -1 \;\; 0]$	$[0.4996 \;\; 0 \;\; 0]$	
4	$[0 \;\; -1 \;\; 0]$	$[0.8919 \;\; 0 \;\; 0]$	
5	$[0 \;\; 0 \;\; 1]$	$[-0.1100 \;\; 0 \;\; 0]$	
6	$[0 \;\; -1 \;\; 0]$	$[0.9866 \;\; 0 \;\; 0]$	

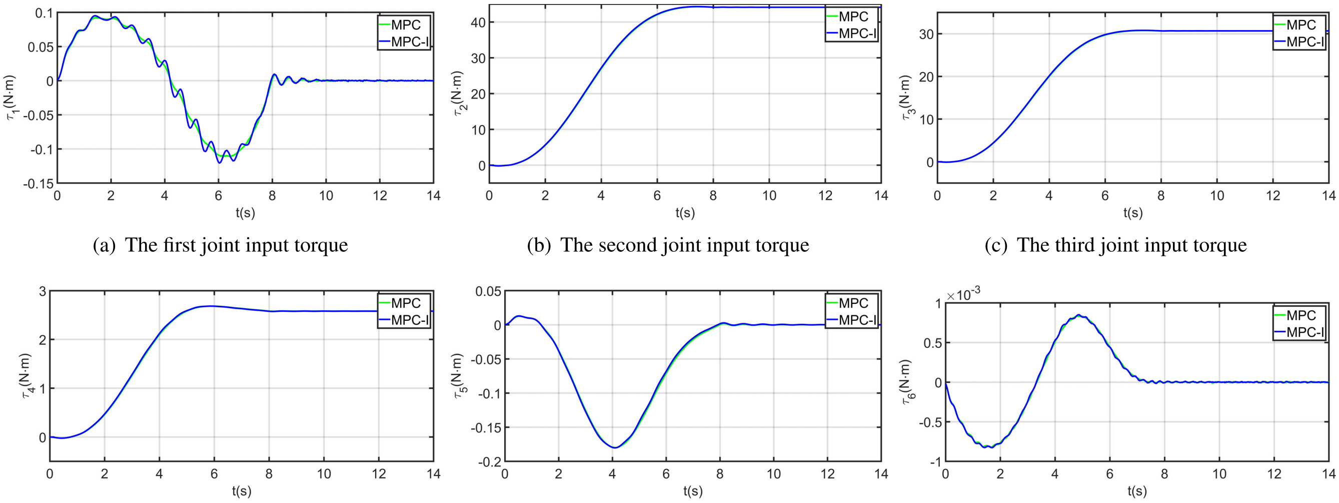
TABLE 2. The specific values of the initial and end joint angle of UR5.

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
$\theta_{\text{init}}(^\circ)$	0	0	0	0	0	0
$\theta_{\text{end}}(^\circ)$	20	-15	-60	-20	90	20



(a) The first joint variable
(b) The second joint variable
(c) The third joint variable
(d) The fourth joint variable
(e) The fifth joint variable
(f) The sixth joint variable

FIGURE 2. The actual and expected joint angles for motion control of the UR5, calculated by the MPC and MPC-I methods, versus time. (a) The first joint variable. (b)The second joint variable. (c) The third joint variable. (d) The fourth joint variable. (e) The fifth joint variable. (f) The sixth joint variable.



(a) The first joint input torque
(b) The second joint input torque
(c) The third joint input torque
(d) The fourth joint input torque
(e) The fifth joint input torque
(f) The sixth joint input torque

FIGURE 3. The joint input torque for motion control of the UR5, calculated by the MPC and MPC-I methods, versus time. (a) The first joint input torque. (b)The second joint input torque. (c) The third joint input torque. (d) The fourth joint input torque. (e) The fifth joint input torque. (f) The sixth joint input torque.

obtained by the fifth-order polynomial interpolation of θ_{init} and θ_{end}. Furthermore, the time interval was defined as $\Delta t = 0.025s$, and the prediction and control horizons were defined as $p = m = 5$. The other parameters in the MPC and MPC-I motion control methods were

$$\boldsymbol{\Gamma}_\theta = \text{diag}(150\boldsymbol{I}_6, 120\boldsymbol{I}_6, 90\boldsymbol{I}_6, 60\boldsymbol{I}_6, 30\boldsymbol{I}_6), \qquad (72)$$

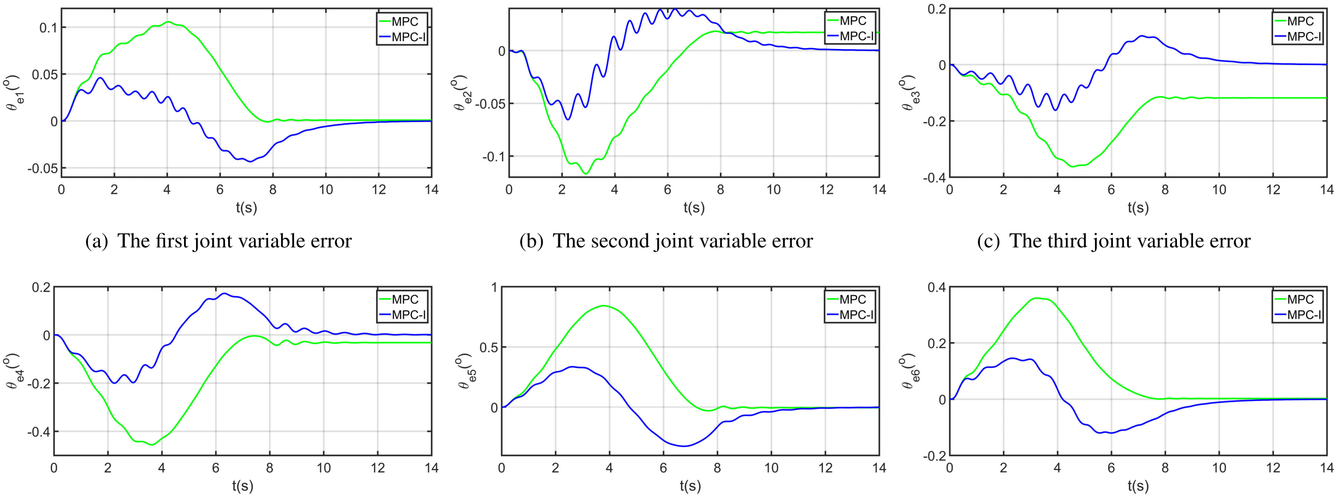
$$\boldsymbol{\Gamma}_w = \text{diag}(0.5\boldsymbol{I}_6, 0.4\boldsymbol{I}_6, 0.3\boldsymbol{I}_6, 0.2\boldsymbol{I}_6, 0.1\boldsymbol{I}_6), \qquad (73)$$

$$\boldsymbol{K}_I = 0.003\ \text{diag}(\boldsymbol{I}_6, \boldsymbol{I}_6, \boldsymbol{I}_6, \boldsymbol{I}_6, \boldsymbol{I}_6), \qquad (74)$$

where $\boldsymbol{I}_66$ denotes the (6×6) identity matrix.

The actual and expected joint angles for motion control of the UR5, calculated by the MPC and MPC-I methods, versus time are shown in Fig. 2. These six figures respectively correspond to the six joint angles of UR5. In addition, the joint input torque for motion control of the UR5, calculated by the MPC and MPC-I methods, versus time are shown in Fig. 3. The differences between MPC and MPC-I methods are shown by driven motions. It can be seen from the Fig. 2 and 3 that the six joints controlled by the MPC and MPC-I methods all follow very accurately and the fluctuation of joint input torque of the MPC-I method is more serious than the MPC method, which can be relieved by adjusting $\boldsymbol{K}_I$.

However, to further understand the behavior of these steady-state errors, the joint errors of the UR5 controlled by the MPC and MPC-I methods versus time are shown in Fig. 4. The steady-state errors, especially those from the second, third, and fourth joints, were significant when the six joints were controlled by this MPC method, which are 0.0172°, −0.1183°, and −0.0323° respectively. Whereas,



(a) The first joint variable error

(b) The second joint variable error

(c) The third joint variable error

(d) The fourth joint variable error

(e) The fifth joint variable error

(f) The sixth joint variable error

FIGURE 4. The joint errors of UR5 controlled by MPC and MPC-I methods versus time. (a) The error of the first joint variable. (b)The error of the second joint variable. (c) The error of the third joint variable. (d) The error of the fourth joint variable. (e) The error of the fifth joint variable. (f) The error of the sixth joint variable.

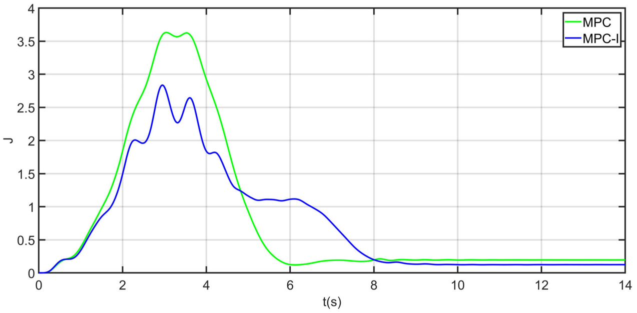


FIGURE 5. The objective function values of MPC and MPC-I methods versus time.

for the MPC-I method, these values converged to 0.0002°, 0.0009°, and 0.0011°, considering the same simulation time. The errors of the MPC-I method were acceptable and were expected to further decrease over time, as analyzed in Section III-A4. To further illustrate the advantages of the MPC-I method, the objective function values (Equation (21)), calculated by the MPC and MPC-I methods, versus time is shown in Fig. 5. As a result, this simulation proves the advantage of the MPC-I method, in which an integral compensation was introduced.

B. COMPARISON OF BETWEEN THE MPC-I_NIK AND MPC-I_IK METHODS IN TASK SPACE

In this section, the reliability of the MPC-I_NIK is verified by comparing its results with those of an MPC-I_IK. The inverse kinematics was calculated directly by the method proposed by [36]. In this experiment, a trajectory that will allow the UR5 to draw the letter "F" in 20s is expected to be obtained. The UR5 model of the V-rep simulation platform is shown in Fig. 6(a) and the desired configuration trajectory of the end-effector of the UR5 is shown in Fig. 6(b).

The initial configuration ($\boldsymbol{T}_{\text{init}}$) and end configuration ($\boldsymbol{T}_{\text{end}}$) of end-effector are defined as:

$$\boldsymbol{T}_{\text{init}} = \begin{bmatrix} -0.0012 & 0.0819 & 0.9966 & 0.5634 \\ 0.9996 & 0.0298 & -0.0012 & 0.0880 \\ -0.0298 & 0.9962 & -0.0819 & 0.3075 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \tag{75}$$

$$\boldsymbol{T}_{\text{end}} = \begin{bmatrix} 0.8780 & -0.3517 & 0.3248 & 0.6634 \\ 0.4102 & 0.9024 & -0.1319 & 0.1880 \\ -0.2467 & 0.2490 & 0.9366 & 0.3075 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \tag{76}$$

To illustrate the configuration trajectory and error analysis of the end-effector, the matrix logarithm was used to convert the configuration to six-dimensional exponential coordinates (S) according to Section II-B. When the expected (S_d) and actual (S_c) exponential coordinates were previously determined, as indicated in (77) and (78), respectively,

$$S_d = [S_{ad}^T, S_{pd}^T]^T \tag{77}$$

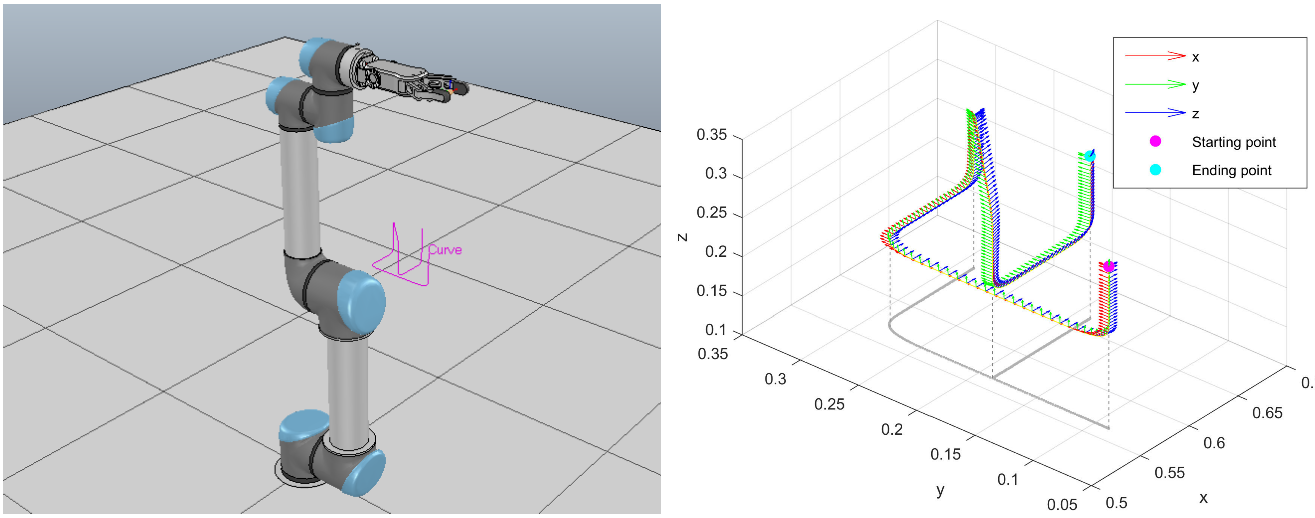
$$S_c = [S_{ac}^T, S_{pc}^T]^T \tag{78}$$

the rotation (E_a) and translation (E_p) errors in task space are defined by (79) and (80), respectively.

$$E_a = ||S_{ad} - S_{ac}||^2 \tag{79}$$

$$E_p = ||S_{pd} - S_{pc}||^2 \tag{80}$$

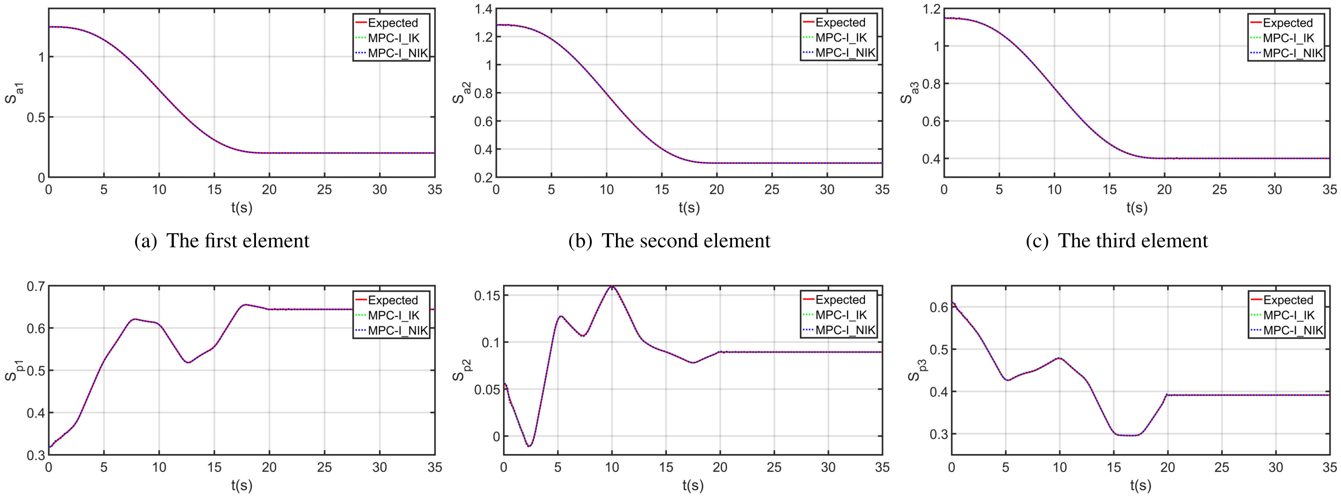
The six components of S_d and S_c versus time are shown in Fig. 7. The actual tracking trajectory was separately obtained by the MPC-I_NIK and MPC-I_IK methods. From Fig. 7, the two methods are satisfactory for the accuracy of the trajectory tracking. For further analysis, the rotation and translation errors of the trajectory tracking expressed by the exponential coordinate versus time are shown in Fig. 8. Both methods were able to obtain results whose rotation and



(a) The UR5 model of V-rep simulation platform

(b) The desired configuration trajectory of the end-effector of UR5

FIGURE 6. The UR5 model of the V-rep simulation platform and the desired configuration trajectory of the end-effector of the UR5. (a) The UR5 model of the V-rep simulation platform. (b) The desired configuration trajectory of the end-effector of the UR5.



(a) The first element

(b) The second element

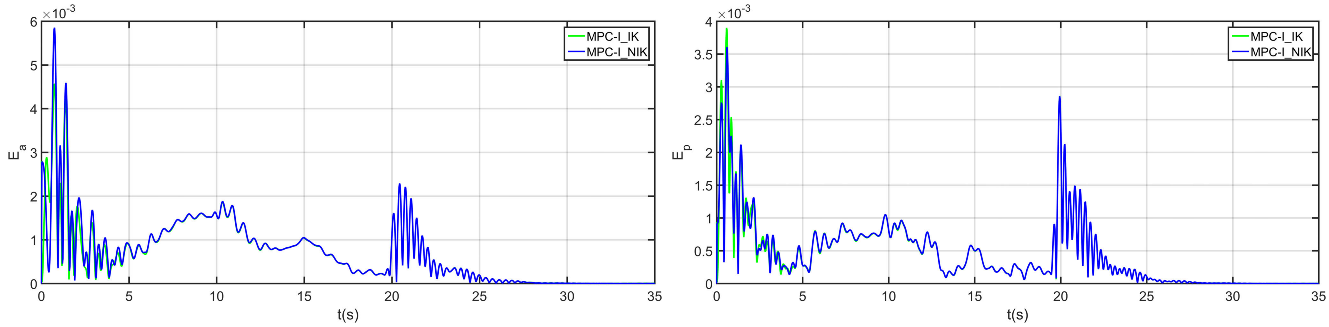
(c) The third element

(d) The fourth element

(e) The fifth element

(f) The sixth element

FIGURE 7. The six components of the expected and actual trajectory exponential coordinate versus time. (a) The first component. (b)The second component. (c) The third component. (d) The fourth element component. (e) The fifth component. (f) The sixth component.

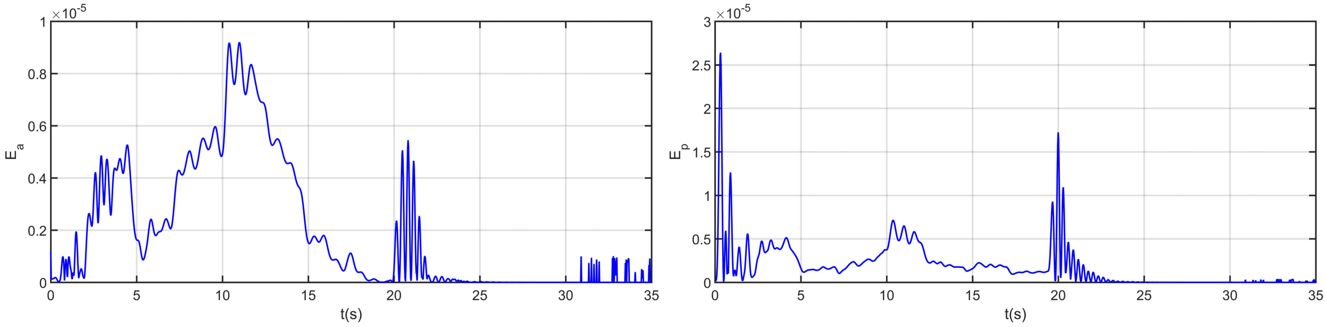


(a) The rotation error

(b) The translation error

FIGURE 8. The rotation and translation errors of trajectory tracking expressed by the exponential coordinate versus time. (a) The rotation error. (b) The translation error.

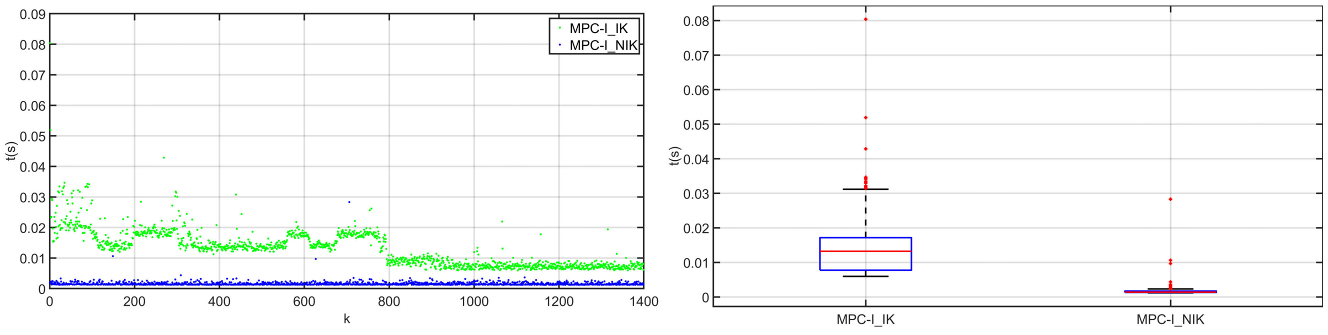
translation errors, in exponential coordinates, were within an acceptable range (order of magnitude of 10^{-6}). To evaluate the accuracy of the space conversion method proposed in this paper, the joint angle estimated by (69) was used to calculate the configuration of the end-effector through forward kinematics. The configuration of the end-effector was, thereafter,



(a) The rotation error

(b) The translation error

FIGURE 9. The rotation and translation errors of spatial transformation. (a) The rotation error. (b)The translation error.



(a) The calculation time versus steps

(b) The calculation time box plots

FIGURE 10. The calculation time of the inverse kinematics of methods MPC-I_NIK and MPC-I_IK. (a) The calculation time versus steps. (b) The calculation time box plots.

converted into exponential coordinates and compared with the exponential coordinates of the desired pose. The rotation and translation errors of the spatial transformation are illustrated in Fig. 9. The rotation error was within 1×10^{-5} and the translation error was within 3×10^{-5}. Finally, to illustrate the computational complexity of the MPC-I_NIK method proposed in this paper, the calculation time of the inverse kinematics of methods MPC-I_NIK and MPC-I_IK is shown in Fig. 10. From Fig. 10(a), the advantages of the MPC-I_NIK method to the MPC-I_IK method can be easily verified. From Fig. 10(b), the medians of the time for calculating inverse kinematics for methods MPC-I_NIK and MPC-I_IK are $0.0014s$ and $0.0132s$, respectively. It can be concluded that the former has a great advantage over the latter in terms of computational complexity.

V. CONCLUSION

This article proposes an MPC-I method for motion control of a robot manipulator considering the joint space and analyzes the advantages of introducing an integral control term. To adapt the method to enable the motion control in task space, a space conversion method for indirect calculation of inverse kinematics was proposed. Two simulations were conducted to illustrate the effectiveness of the proposed method. In the first one, the effect of the MPC and MPC-I methods was compared, considering the motion control in joint space. The results indicated that the steady-state errors of the six joints controlled by the MPC method were 0.0008°, 0.0172°, −0.1183°, −0.0323°, −0.0042°, and 0.0022°, respectively; however, with the proposed MPC-I method, the steady-state errors converged to lower values (−0.0003°, 0.0002°, 0.0009°, 0.0011°, 0.0002°, and −0.0005°, respectively), at the same simulation time. In the second simulation, the MPC-I_NIK and MPC-I_IK methods were compared considering the motion control in task space. Both methods could achieve results with rotation and translation errors in exponential coordinates within the order of magnitude of 10^{-6}. Meanwhile, the rotation and translation errors of the space conversion method proposed in this paper were 1×10^{-5} and 3×10^{-5}, respectively, which shows that the method is effective. And the median time of the MPC-I_NIK method to calculate the inverse kinematics is $0.0014s$, which is a great improvement over the calculation time of MPC-I_IK method.

In future works, actual constraints, such as maximum speed and maximum acceleration, will be added to the MPC-I method, resulting in a more practical method. Furthermore, the motion control method in joint and task spaces proposed in this paper will be applied to practical applications, such as handling of hazardous materials and motion assistance. Moreover, the MPC-I method will be extended to hybrid motion-force control.

ACKNOWLEDGMENT

The authors thank the Shuo Wang for helping with the language of this manuscript and all their colleagues for providing different types of help during the preparation of this manuscript.

REFERENCES

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*. London, U.K.: Springer, 2009.

[2] F. Alambeigi, M. Bakhtiarinejad, S. Sefati, R. Hegeman, I. Iordachita, H. Khanuja, and M. Armand, "On the use of a continuum manipulator and a bendable medical screw for minimally invasive interventions in orthopedic surgery," *IEEE Trans. Med. Robot. Bionics*, vol. 1, no. 1, pp. 14–21, Feb. 2019.

[3] J. Vertut, *Teleoperation and Robotics: Applications and Technology*. New York, NY, USA: Springer, 2013.

[4] M. B. Hong, S. J. Kim, Y. S. Ihn, G.-C. Jeong, and K. Kim, "KULEX-hand: An underactuated wearable hand for grasping power assistance," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 420–432, Apr. 2019.

[5] F. Aghili, "Robust impedance-matching of manipulators interacting with uncertain environments: Application to task verification of the space station's dexterous manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 4, pp. 1565–1576, Aug. 2019.

[6] E. Simetti and G. Casalino, "Manipulation and transportation with cooperative underwater vehicle manipulator systems," *IEEE J. Ocean. Eng.*, vol. 42, no. 4, pp. 782–799, Oct. 2017.

[7] T. Shibata and T. Murakami, "Null space motion control by PID control considering passivity in redundant manipulator," *IEEE Trans. Ind. Informat.*, vol. 4, no. 4, pp. 261–270, Nov. 2008.

[8] M. Alhaddad, B. Shaukifeh, and A. Joukhadar, "Adaptive LQ-based computed-torque controller for robotic manipulator," in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (EIConRus)*, Jan. 2019, pp. 2169–2173.

[9] M. I. Ullah, S. A. Ajwad, M. Irfan, and J. Iqbal, "Non-linear control law for articulated serial manipulators: Simulation augmented with hardware implementation," *Elektronika Ir Elektrotechnika*, vol. 22, no. 1, pp. 3–7, Feb. 2016.

[10] J. Baek, S. Cho, and S. Han, "Practical time-delay control with adaptive gains for trajectory tracking of robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5682–5692, Jul. 2018.

[11] 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.

[12] T. L. Costa, F. A. Lara-Molina, A. A. C. Junior, and E. Taketa, "Robust H∞ computed torque control for manipulators," *IEEE Latin Amer. Trans.*, vol. 16, no. 2, pp. 398–407, Feb. 2018.

[13] G. Yang, Y. Liu, M. Jin, and H. Liu, "A robust and adaptive control method for flexible-joint manipulator capturing a tumbling satellite," *IEEE Access*, vol. 7, pp. 159971–159985, 2019.

[14] M. Sato and M. Toda, "Robust motion control of an oscillatory-base manipulator in a global coordinate system," *IEEE Trans. Ind. Electron.*, vol. 62, no. 2, pp. 1163–1174, Feb. 2015.

[15] Khorashadizadeh, Saeed, "Robust task-space control of robot manipulators using Legendre polynomials for uncertainty estimation," *Nonlinear Dyn.*, vol. 79, no. 2, pp. 1161–1511, 2014.

[16] M. M. Fateh and S. Khorashadizadeh, "Robust control of electrically driven robots by adaptive fuzzy estimation of uncertainty," *Nonlinear Dyn.*, vol. 69, no. 3, pp. 1465–1477, Aug. 2012.

[17] H. Shi, P. Li, L. Wang, C. Su, J. Yu, and J. Cao, "Delay-range-dependent robust constrained model predictive control for industrial processes with uncertainties and unknown disturbances," *Complexity*, vol. 2019, pp. 1–15, Jan. 2019.

[18] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. New York, NY, USA: Springer-Verlag, 2011.

[19] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Pract.*, vol. 11, no. 7, pp. 733–764, Jul. 2003.

[20] D. Paluszczyszyn, P. Skworcow, O. Haas, K. J. Burnham, and J. A. Mills, "Model predictive control for real-time tumor motion compensation in adaptive radiotherapy," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 2, pp. 635–651, Mar. 2014.

[21] Y. Tadokoro, Y. Taya, T. Ibuki, and M. Sampei, "Real-time model predictive control of rigid body motion via discretization using the Cayley map," *IEEE Access*, vol. 8, pp. 17149–17159, 2020.

[22] M. Chipofya, D. J. Lee, and K. T. Chong, "Mobile robot motion control using Laguerre-based model predictive control," *Appl. Mech. Mater.*, vol. 776, pp. 403–410, Jul. 2015.

[23] M. B. Shadmand, R. S. Balog, and H. Abu-Rub, "Model predictive control of PV sources in a smart DC distribution system: Maximum power point tracking and droop control," *IEEE Trans. Energy Convers.*, vol. 29, no. 4, pp. 913–921, Dec. 2014.

[24] M. Hajiahmadi, G. S. van de Weg, C. M. J. Tampere, R. Corthout, A. Hegyi, B. De Schutter, and H. Hellendoorn, "Integrated predictive control of freeway networks using the extended link transmission model," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 65–78, Jan. 2016.

[25] R. Krug and D. Dimitrov, "Model predictive motion control based on generalized dynamical movement primitives," *J. Intell. Robotic Syst.*, vol. 77, no. 1, pp. 17–35, Jan. 2015.

[26] R. A. McCourt and C. W. de Silva, "Autonomous robotic capture of a satellite using constrained predictive control," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 6, pp. 699–708, Dec. 2006.

[27] P. Poignet and M. Gautier, "Nonlinear model predictive control of a robot manipulator," in *Proc. 6th Int. Workshop Adv. Motion Control*, Mar. 2000, pp. 401–406.

[28] R. Hedjar, R. Toumi, P. Boucher, and D. Dumur, "Feedback nonlinear predictive control of rigid link robot manipulators," in *Proc. Amer. Control Conf.*, May 2002, pp. 3594–3599.

[29] M. I. Ullah, S. A. Ajwad, M. Irfan, and J. Iqbal, "MPC and H-infinity based feedback control of non-linear robotic manipulator," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2016, pp. 136–141.

[30] G. P. Incremona, A. Ferrara, and L. Magni, "MPC for robot manipulators with integral sliding modes generation," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1299–1307, Jun. 2017.

[31] M. I. Ullah, S. A. Ajwad, R. U. Islam, U. Iqbal, and J. Iqbal, "Modeling and computed torque control of a 6 degree of freedom robotic arm," in *Proc. Int. Conf. Robot. Emerg. Allied Technol. Eng. (iCREATE)*, Apr. 2014, pp. 133–138.

[32] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1321–1326.

[33] K. Lynch and F. Park, *Modern Robotics, Mechanics, Planning, and Control*. Cambridge, U.K., 2017.

[34] P. E. B Jourdain, *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies; With an Introduction to the Problem of Three Bodies*, vol. 9, no. 131, E. T. Whittaker, Ed, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1917, pp. 145–146.

[35] H. K. Khalil, *Nonlinear Systems*, 3rd ed. New York, NY, USA: Macmillan Publishing Company, 2002.

[36] Y. Chen, X. Luo, B. Han, Y. Jia, G. Liang, and X. Wang, "A general approach based on Newton's method and cyclic coordinate descent method for solving the inverse kinematics," *Appl. Sci.*, vol. 9, no. 24, p. 5461, Dec. 2019.

YUHAN CHEN received the B.S. degree in mechanical engineering from the Taiyuan University of Technology, Taiyuan, China, in 2016, and the master's degree and the M.Sc.-Ph.D. based program in mechanical engineering from the Beijing Institute of Technology. His research interests include motion planning, mobile manipulation, and robot dynamics.

XIAO LUO received the bachelor's degree in information and electronics and the master's and Ph.D. degrees in mechatronical engineering from the Beijing Institute of Technology, Beijing, China, in 2006, 2008, and 2012, respectively. She is currently a Lecturer with the Beijing Institute of Technology. Her current research interests include image processing, computer vision, simultaneous localization and mapping (SLAM), and automate robot.

BAOLING HAN received the B.S. degree in precision instrument design from the Beijing Institute of Technology, Beijing, China, in 1982, the M.S. degree in precision instrument from Tsinghua University, Beijing, in 1995, and the Ph.D. degree in mechanical engineering from the Beijing Institute of Technology, in 2002. She is currently pursuing the Doctor of Engineering degree with the Beijing University of Technology. She is also a Professor/Ph.D. Supervisor with the Beijing Institute of Technology. Her main research interests include intelligent robot, optical electromechanical integration technology, and so on.

QINGSHENG LUO received the B.S. degree in precision instrument design, the M.S. degree in mechanical optimization design, and the Ph.D. degree in mechanical engineering from the Beijing Institute of Technology, Beijing, China, in 1982, 1987, and 2001, respectively. His research interests include specialized robot technology, mechatronical product innovative design, mechatronical system control technology, and mechatronical device integration technology.

LIJUN QIAO received the B.S. degree in measurement and control technology and instrument from the Taiyuan University of Science and Technology, Taiyuan, China, in 2014, and the M.S. degree in instrument science and technology from the North University of China, Taiyuan, in 2017. He is currently pursuing the Ph.D. degree in mechanical engineering with the Beijing Institute of Technology. His major research interests include dual-arm robot, robot dynamics control, and robot stability.