Robotica (2024), **42**, pp. 2690–2712
doi:10.1017/S026357472400105X

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

An online payload identification method based on parameter difference for industrial robots

Tian Xu[1,2], Hua Tuo[1], Qianqian Fang[1], Jie Chen[3], Jizhuang Fan[1], Debin Shan[2] and Jie Zhao[1]

[1]State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China
[2]School of Materials Science and Engineering, Harbin Institute of Technology, Harbin, China
[3]School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China
Corresponding authors: Jie Chen; Email: chenjie@me.neu.edu.cn; Jizhuang Fan; Email: fanjizhuang@hit.edu.cn; Jie Zhao; Email: jzhao@hit.edu.cn

Received: 7 November 2023; **Revised:** 29 April 2024; **Accepted:** 21 May 2024; **First published online:** 13 September 2024

Keywords: online payload estimation; parameter difference; dynamic identification; nonlinear friction model; UR10 robot

Abstract

Accurate online estimation of the payload parameters benefits robot control. In the existing approaches, however, on the one hand, only the linear friction model was used for online payload identification, which reduced the online estimation accuracy. On the other hand, the estimation models contain much noise because of using actual joint trajectory signals. In this article, a new estimation algorithm based on parameter difference for the payload dynamics is proposed. This method uses a nonlinear friction model for the online payload estimation instead of the traditionally linear one. In addition, it considers the commanded joint trajectory signals as the computation input to reduce the model noise. The main contribution of this article is to derive a symbolic relationship between the parameter difference and the payload parameters and then apply it to the online payload estimation. The robot base parameters without payload were identified offline and regarded as the prior information. The one with payload can be solved online by the recursive least squares method. The dynamics of the payload can be then solved online based on the numerical difference of the two parameter sets. Finally, experimental comparisons and a manual guidance application experiment are shown. The results confirm that our algorithm can improve the online payload estimation accuracy (especially the payload mass) and the manual guidance comfort.

1. Introduction

Accurate payload dynamics benefit robot applications [1]. For instance, accurate payload dynamics can provide strong priors for robot planning control [2, 3], safe interaction [4], and manual guidance [5, 6].

Generally, the payload parameters need to be identified experimentally. In addition, aiming at some applications in industrial environments such as picking different payload frequently [7], the identifications have to be provided online [8]. In this area, Abiko et al. [9] proposed two online identification methods for the payload parameters. One is the vibration motion method, which is to construct the identification model based on the velocity deviation of the base. However, this method needs to measure both linear velocity and angular velocity of the base. The other is the reaction force method, which is to construct the identification model based on the six-dimensional force deviation at the end of the robot. It should be noted that this method requires the installation of an additional force/torque sensor. Kubus et al. [10] developed an online method based on the recursive total least squares (RTLS) method [11]. Different from the recursive least squares (RLS) [12, 13], this method can take into account measurement errors to obtain more robust estimation results. In order to cope with the influence of payloads on the balance control of quadruped robots, Tournois et al. [14] proposed two iterative-based online identification algorithms. However, neither two algorithms consider the friction model. Farsoni et al. [15, 16] proposed an identification method based on Kalman filter [17] and RTLS. Firstly, the linear

© The Author(s), 2024. Published by Cambridge University Press.

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

acceleration, angular velocity, and angular acceleration of the payload are estimated by the Kalman filter based on multi-rate quaternion [18]. Then, the inertial parameters of the payload are identified online and in real time by the RTLS. Later, this team proposed an online payload identification method based on collision-free path planning and motion decoupling [19]. When generating recognition trajectory, it regards the operator as an obstacle and uses a mature obstacle avoidance algorithm [20] to avoid the operator. Additionally, it requires information from visual sensors. Renner et al. [21] proposed an optimization method based on ordinary least squares method for online payload identification. This method constructs the optimization objective function by minimizing the torque errors [22] and uses the RLS to solve the payload parameters online.

Another way to obtain payload parameters is by the dynamic identification method. In order to obtain good dynamic parameter identification, the design of a suitable excitation trajectory [23, 24] is essential to fulfill the continuous excitation conditions for the observation matrix. A classic trajectory is the Fourier series excitation trajectory designed by Swevers et al. [25]. Optimal trajectory parameters are often obtained by optimizing the condition number of the observation matrix [26]. Recently, some studies have demonstrated that incentive effects can be improved by optimizing the sub-matrix condition number [27, 28]. Additionally, considering measurement noise covariance during optimization and conducting the process iteratively can better capture the dynamic characteristics of the robot [29]. Furthermore, Scalera et al. [30] proposed an interval arithmetic method to deal with dynamics uncertainties in alternative to the online identification of dynamic parameters. Swevers et al. [31] used a maximum likelihood method to estimate the payload parameters based on periodic excitation. The limitation of this method is that it introduces recognition errors as it ignores the dynamics of the last step. To solve this problem, a classical method was proposed in refs. [32, 33]. It distinguished the contribution of the robot itself and the payload to the joint torques. The robot link parameters without load were considered as the prior information for the online payload identification. However, the estimation results in this method are biased. Later, this method was summarized in ref. [34]. Salah et al. [35] reproduced the online method in refs. [32, 33] for medical applications. Afterward, Gaz et al. [36] proposed a static method to identify the payload parameters. However, this method cannot obtain the inertia tensor of the payload because it lacks the dynamic excitation. Rodriguez et al. [37] proposed an algebraic method for online payload identification. This method assumes that the total mass of both the link and the payload is concentrated at the end of the link. Then, it transforms the dynamic model to the frequency domain for calculation and identification. Unfortunately, this method is only suitable for a single-link system. In order to achieve accurate motion control, Hu et al. [38] proposed an adaptive control algorithm based on generalized momentum for online payload estimation. However, because it sets physical constraints and eliminates many unnecessary dynamic parameters, the accuracy of the torque reconstruction for the robot is insufficient. Thereby, it can increase the cumulative errors of the payload estimation. Recently, Xu et al. [39] have proposed an accurate offline payload identification method based on double weighting technique, but this method cannot yet be applied to online payload identification because of its computation burden. The methods discussed above are all applicable to serial robots. The dynamic modeling, evaluation, and analysis about parallel robots can be found in refs. [40–42].

The authors presented a table as below to compare the proposed work with similar research and state-of-the-art studies identified in the literature review. As shown in the Table I, it can be seen that the payload mass estimation error of the proposed method is the smallest one with respect to other alternative approaches. Khalil et al. [33] and others [38] used the simple linear friction model. The friction model was ignored or not used in refs. [36], [15], [12], and [43] because the joint torque sensors or F/T sensors were used in these literature. To the best of the authors' knowledge, a nonlinear friction model (2) is applied to online payload estimation for the first time in this article. The nonlinear friction model can effectively improve the accuracy of robot dynamic parameter identification, thereby improving the accuracy of payload identification (especially for the mass). The methods in refs. [15] and [12] need external sensors (such as IMU, F/T sensor) for payload estimation. However, other methods only use the proprioceptive sensors to identify payload parameters, which can reduce the cost. Based on the

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Table I. *Comparisons between the proposed method and other methods.*

Method source	Payload mass estimation error (%)	If use nonlinear friction model	If use proprioceptive sensors only	Convergence times
[33]	4.63% (reproduction accuracy: 4.11%)	No	Yes	More than 10 s
[36]	1.29% (reproduction accuracy: 0.92%)	No	Yes	More than 20 s
[15]	1.57%	No	No (IMU, F/T sensor)	6.89 s–12.37 s
[12]	1.00%	No	No (IMU, F/T sensor etc.)	10 s
[38]	4.40%	No	Yes	8 s
[43]	5.48%	No	Yes	6.6 s–7.8 s
Proposed	**0.04%**	**Yes**	**Yes**	**6.59 s**

convergence time data presented in the literature, the proposed method has the least convergence times which are about 6.59 s.

After the comparisons, the originality, novelty, and the advantage and disadvantage with respect to alternative approaches are summarized here. The originality and novelty of this article is to derive a symbolic relationship between the parameter difference and the payload parameters and then apply it to the online payload estimation. In addition, a nonlinear friction model is used for online payload estimation for the first time. The main advantage of the proposed method is to achieve higher online estimation accuracy of the payload mass m_L. In order to increase the online estimation accuracy, on the one hand, a nonlinear friction model (2) was used instead of the linear one. On the other hand, the commanded joint trajectory signals were introduced as the computation input instead of the actual ones. The main disadvantage of the proposed method is that it cannot identify the accurate payload inertia tensor enough online $[I_{xx,L}, I_{xy,L}, I_{xz,L}, I_{yy,L}, I_{yz,L}, I_{zz,L}]$, which is consistent with the disadvantage of these alternatively online approaches by using proprioceptive sensors only.

Finally, based on the comparisons, the main contributions of this work are summarized as follows:

1) An online payload identification method without using external sensors is proposed, which is based on the symbolic relationship between robot dynamic parameter differences and payload parameters.

2) Compared with other methods, the mass estimation accuracy of a payload can be significantly improved by using the proposed method.

3) A nonlinear friction model (2) is applied to online payload estimation for the first time to increase the identification accuracy in the proposed method.

The remainder of this article is organized as follows. Section 2 presents the preliminaries. Section 3 introduces the proposed method. Section 4 provides the experimental results. Finally, Section 5 concludes this article.

2. Preliminaries

The inverse dynamics of the robot with n rigid links by considering the motor dynamics can be written as follows:

$$\boldsymbol{\tau} = (\boldsymbol{M}(\boldsymbol{q}) + \boldsymbol{I}_a)\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_f, \tag{1}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \Re^{n\times 1}$ are joint position, joint velocity, and joint acceleration of robot, respectively. The n is the degrees of freedom of the robot. $\boldsymbol{M}(\boldsymbol{q}) \in \Re^{n\times n}$ and $\boldsymbol{I}_a \in \Re^{n\times n}$ are the inertia matrix and the equivalent

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

moment of rotor, respectively. The gravity vector is $\boldsymbol{g}(\boldsymbol{q}) \in \Re^{n\times 1}$. The centrifugal and Coriolis matrix is $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \Re^{n\times n}$. The joint torque vector is $\boldsymbol{\tau} \in \Re^{n\times 1}$. The friction torque vector is $\boldsymbol{\tau}_f \in \Re^{n\times 1}$.

Compared with the classic linear friction model $\tau_{f,j} = F_{c,j}\text{sign}(\dot{q}_j) + F_{v,j}\dot{q}_j + B_j$ [44], the nonlinear friction model is more conducive to improving the identification accuracy of the robot dynamics [29, 45]. It is called a linear friction model because the model can be transformed into a linear product of pure velocity terms and friction parameters, such as $\tau_{f,j} = \left[\text{sign}(\dot{q}_j)\ \dot{q}_j\ 1\right] \cdot \left[F_{c,j}\ F_{v,j}\ B_j\right]^{\text{T}}$. However, for a nonlinear friction model, the nonlinear friction parameter will be included in the velocity term. For example, a nonlinear friction model was used in this article as

$$\begin{aligned}\tau_{f,j} &= (F_{c,j} + F_{v,j}|\dot{q}_j|^{\alpha_j})\text{sign}(\dot{q}_j) + B_j \\ &= \left[\text{sign}(\dot{q}_j)\ |\dot{q}_j|^{\alpha_j}\text{sign}(\dot{q}_j)\ 1\right] \cdot \left[F_{c,j}\ F_{v,j}\ B_j\right]^{\text{T}},\end{aligned} \tag{2}$$

where $\tau_{f,j}$ is the friction torque of joint j in $\boldsymbol{\tau}_f$. α_j is the nonlinear friction parameter of joint j in $\boldsymbol{\alpha}$. $F_{c,j}$ is the Coulomb friction parameter of joint j. $F_{v,j}$ is the viscous friction parameter of joint j. B_j is the friction torque offset of joint j. $F_{c,j}$, $F_{v,j}$, and B_j are all the linear friction parameters.

Then (1) can be rewritten by considering (2) as follows:

$$\boldsymbol{\tau} = \boldsymbol{W}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\alpha})\boldsymbol{\varphi}, \tag{3}$$

where $\boldsymbol{\varphi} \in \Re^{14n\times 1}$ is the completed robot dynamic parameters. For j-th joint, $\boldsymbol{\varphi}^j$ is equal to $[\boldsymbol{l}_j, m_j\boldsymbol{r}_j, m_j, I_{aj}, F_{c,j}, F_{v,j}, B_j]^{\text{T}}$. $\boldsymbol{l}_j$ is the inertia tensor vector which is equal to $[I_{xx,j}, I_{xy,j}, I_{xz,j}, I_{yy,j}, I_{yz,j}, I_{zz,j}]$. $I_{xx,j}$, $I_{yy,j}$, and $I_{zz,j}$ are the moment of inertia of link j. $I_{xy,j}$, $I_{xz,j}$, and $I_{yz,j}$ are the products of inertia of link j. It can form the inertia tensor matrix $\boldsymbol{I}_j \in \Re^{3\times 3}$ with respect to frame j. $m_j\boldsymbol{r}_j$ is the first moment vector which is equal to $\left[m_jr_{x,j}, m_jr_{y,j}, m_jr_{z,j}\right]$. The link mass is m_j. $r_{x,j}$, $r_{y,j}$, and $r_{z,j}$ are the position of link center of mass (CoM) on X axis, Y axis, and Z axis, respectively.

According to Gautier and Khalil and others [46–49], $\boldsymbol{\varphi}$ has some unidentifiable parameters, and they can be identified by the combined form. After deleting or regrouping these parameters, (3) can be modified as follows:

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\alpha})\boldsymbol{\pi}, \tag{4}$$

where $\boldsymbol{\pi} \in \Re^{p\times 1}$ is the base parameter set and p is the number of base parameters. $\boldsymbol{W}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\alpha}) \in \Re^{n\times 14n}$ is the rank-deficient regressor because of the linearly correlated columns. $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\alpha}) \in \Re^{n\times p}$ is with full rank after eliminating them. In order to simplify the expression, the regressor can be represented by the first letter. For example, $\boldsymbol{W}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\alpha})$ and $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\alpha})$ can be represented as $\boldsymbol{W}$ and $\boldsymbol{Y}$, respectively.

3. The proposed algorithm

3.1. Algorithm details

Assuming that the payload was rigidly connected to the robot flange, the dynamic equation of robot with payload can then be written by

$$\boldsymbol{\tau}_b = \boldsymbol{Y}_b\boldsymbol{\pi}_b = \boldsymbol{Y}_b\boldsymbol{\pi}_a + \boldsymbol{W}_L\boldsymbol{\varphi}_L + \boldsymbol{W}_f\Delta\boldsymbol{\varphi}_f, \tag{5}$$

where subscript a means the case without payload and subscript b means the case with payload. Therefore, $\boldsymbol{\pi}_a$ and $\boldsymbol{\pi}_b$ mean the base parameters of the robot without payload and that with payload, respectively. $\boldsymbol{Y}_b$ is the regressor matrix corresponding to $\boldsymbol{\pi}_b$. $\boldsymbol{\tau}_b$ is the measured joint torques of the robot with payload. $\boldsymbol{\varphi}_L \in \Re^{10\times 1}$ is the payload parameters, that is, $\boldsymbol{\varphi}_L = [I_{xx,L}, I_{xy,L}, I_{xz,L}, I_{yy,L}, I_{yz,L}, I_{zz,L}, m_Lr_{x,L}, m_Lr_{y,L}, m_Lr_{z,L}, m_L]^{\text{T}}$. m_L is the payload mass. $r_{x,L}$, $r_{y,L}$, and $r_{z,L}$ are the position of payload CoM on X axis, Y axis, and Z axis, respectively. $I_{xx,L}$, $I_{yy,L}$, and $I_{zz,L}$ are the moment of inertia of payload. $I_{xy,L}$, $I_{xz,L}$, and $I_{yz,L}$ are the products of inertia of payload. The corresponding payload regressor matrix is $\boldsymbol{W}_L \in \Re^{n\times 10}$. $\Delta\boldsymbol{\varphi}_f \in \Re^{3n\times 1}$ is the linear friction variations due to the payload dynamics, and the corresponding regressor matrix is $\boldsymbol{W}_f \in \Re^{n\times 3n}$.

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

The difference between the base parameters with payload and the ones without payload can be calculated by

$$\boldsymbol{\varepsilon} = \boldsymbol{\pi}_b - \boldsymbol{\pi}_a, \tag{6}$$

where $\boldsymbol{\varepsilon} \in \Re^{p\times 1}$ is the parameter difference.

The symbolic expressions of $\boldsymbol{\pi}_a$ and $\boldsymbol{\pi}_b$ have been derived in the appendix. Then, the parameter difference $\boldsymbol{\varepsilon}$ can be computed by (6) and the nonzero elements $\boldsymbol{\varepsilon}_{nz}$ in $\boldsymbol{\varepsilon}$ can be then extracted. As shown in the appendix, it can be found that there is a linear relationship between $\boldsymbol{\varphi}_L$ and $\boldsymbol{\varepsilon}_{nz}$. Therefore, once the numerical results of $\boldsymbol{\varepsilon}_{nz}$ were solved, the $\boldsymbol{\varphi}_L$ could be then solved according to this relationship. In order to compute $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}_{nz}$, $\boldsymbol{\pi}_a$ and $\boldsymbol{\pi}_b$ need to be identified. In fact, $\boldsymbol{\pi}_a$ can be identified offline in advance by the least-squares method. Then, the remaining problem is how to solve $\boldsymbol{\pi}_b$ online. In this paper, the RLS method [10] is used to solve $\boldsymbol{\pi}_b$ online. The specific implementation process of the proposed algorithm is shown below.

Firstly, the regressor matrix $\boldsymbol{Y}_b$ can be computed by the commanded joint data at each sampling moment, namely

$$\boldsymbol{Y}_b = \boldsymbol{Y}_b(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d, \boldsymbol{\alpha}_a), \tag{7}$$

where $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d \in \Re^{n\times 1}$ is the commanded joint position, commanded joint velocity, and commanded joint acceleration, respectively. $\boldsymbol{\alpha}_a$ is the nonlinear friction parameters of the robot without payload, which was identified offline in advance with $\boldsymbol{\pi}_a$ together.

Then, the gain matrix $\boldsymbol{K}^* \in \Re^{p\times n}$ at each sampling moment can be computed by

$$\boldsymbol{K}^* = \boldsymbol{P}^* \boldsymbol{Y}_b^{\mathrm{T}} (\xi \mathbf{1} + \boldsymbol{Y}_b \boldsymbol{P}^* \boldsymbol{Y}_b^{\mathrm{T}})^{-1}, \tag{8}$$

where $\boldsymbol{P}^* \in \Re^{p\times p}$ is the covariance matrix. ξ is the forgetting factor which is a positive scalar. $\mathbf{1}$ is a properly sized identity matrix.

The torque residual vector $\boldsymbol{e} \in \Re^{n\times 1}$ at each sampling moment can be computed by

$$\boldsymbol{e} = \boldsymbol{\tau}_b - \boldsymbol{Y}_b \boldsymbol{\pi}_b. \tag{9}$$

Afterward, the base parameters $\boldsymbol{\pi}_b$ can be updated by

$$\boldsymbol{\pi}_b = \boldsymbol{\pi}_b + \boldsymbol{K}^* \boldsymbol{e}, \tag{10}$$

where the initial value of $\boldsymbol{\pi}_b$ is $\boldsymbol{\pi}_a$.

The covariance matrix $\boldsymbol{P}^*$ can be updated by

$$\boldsymbol{P}^* = \boldsymbol{P}^* - \boldsymbol{K}^* \boldsymbol{Y}_b \boldsymbol{P}^*, \tag{11}$$

where the initial value of $\boldsymbol{P}^*$ is an identity matrix.

The final solution of the iterative process is the base parameters of the robot with payload, namely, $\boldsymbol{\pi}_b$. Since the base parameters of the robot without payload $\boldsymbol{\pi}_a$ can be obtained in advance through offline identification, the RLS iteration is mainly to solve $\boldsymbol{\pi}_b$. Once $\boldsymbol{\pi}_b$ converges and is solved, the above iterative calculation process can end. Then, the payload parameters can be estimated by the derived symbolic relationship between the base parameter difference and payload parameters in the appendix. The numerical results of $\boldsymbol{\varepsilon}$ can be solved by (6), and the nonzero elements $\boldsymbol{\varepsilon}_{nz}$ can be extracted from $\boldsymbol{\varepsilon}$. According to the symbolic expressions of $\boldsymbol{\varepsilon}_{nz}$ in the appendix, the following six payload parameters can be directly calculated by

$$\begin{aligned} &I_{xy,L} = \varepsilon_{nz,17},\ I_{xz,L} = \varepsilon_{nz,18},\ I_{yz,L} = \varepsilon_{nz,19}, \\ &I_{zz,L} = \varepsilon_{nz,20},\ m_L r_{x,L} = \varepsilon_{nz,21},\ m_L r_{y,L} = \varepsilon_{nz,22}. \end{aligned} \tag{12}$$

where $\varepsilon_{nz,17}$, $\varepsilon_{nz,18}$, $\varepsilon_{nz,19}$, $\varepsilon_{nz,20}$, $\varepsilon_{nz,21}$, and $\varepsilon_{nz,22}$ are the 17-*th*, 18-*th*, 19-*th*, 20-*th*, 21-*th*, and 22-*th* elements in $\boldsymbol{\varepsilon}_{nz}$, respectively. As explained above, $\boldsymbol{\varepsilon}_{nz}$ is the non-zero element vector of $\boldsymbol{\varepsilon}$ and its symbolic expressions have been shown in the appendix.

Then, the payload mass m_L can be calculated by the following relationship:

$$\left[a_3,\ a_4,\ d_5\right] \cdot m_L = \left[\varepsilon_{nz,5},\ \varepsilon_{nz,9},\ \varepsilon_{nz,12}\right], \tag{13}$$

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

where a_3 and a_4 are the link lengths of link 2 and link 3, respectively. d_5 is the joint distance between joint coordinate 4 and joint coordinate 5. They are all the kinematic parameters of the UR10 robot. $\varepsilon_{nz,5}$, $\varepsilon_{nz,9}$, and $\varepsilon_{nz,12}$ are the 5-*th*, 9-*th*, and 12-*th* elements in $\boldsymbol{\varepsilon}_{nz}$, respectively.

When m_L was solved, the $m_L r_{z,L}$ can be solved by

$$m_L r_{z,L} = -d_6 m_L - \varepsilon_{nz,15}, \tag{14}$$

where d_6 is the joint distance between joint coordinate 5 and joint coordinate 6. $\varepsilon_{nz,15}$ is 15-*th* element in $\boldsymbol{\varepsilon}_{nz}$.

Once $m_L r_{x,L}$, $m_L r_{y,L}$, $m_L r_{z,L}$, and m_L were solved, the CoM ($r_{x,L}$, $r_{y,L}$, $r_{z,L}$) can be obtained by the ratios of them, such as $r_{x,L} = \frac{m_L r_{x,L}}{m_L}$.

After that, $I_{yy,L}$ can be solved by the linear equation as follows:

$$\left[1,\ 1\right] \cdot I_{yy,L} = \left[\varepsilon_{nz,13} - 2d_6 m_L r_{z,L} - d_6^2 m_L,\ \varepsilon_{nz,14} - 2d_6 m_L r_{z,L} - d_6^2 m_L\right], \tag{15}$$

where $\varepsilon_{nz,13}$ and $\varepsilon_{nz,14}$ are the 13-*th* and 14-*th* elements in $\boldsymbol{\varepsilon}_{nz}$, respectively.

Finally, $I_{xx,L}$ can be solved by

$$I_{xx,L} = \varepsilon_{nz,16} + I_{yy,L}, \tag{16}$$

where $\varepsilon_{nz,16}$ is the 16-*th* element in $\boldsymbol{\varepsilon}_{nz}$.

3.2. The overall algorithm

In order to explain the proposed algorithm better, the pseudo-code of the proposed algorithm was shown below. The input of the algorithm include two parts. One part is the base parameters $\boldsymbol{\pi}_a$ and the nonlinear friction parameters $\boldsymbol{\alpha}_a$ of the robot without payload, which have been identified offline in advance. The other is the measured joint torques $\boldsymbol{\tau}_b$, the commanded joint position $\boldsymbol{q}_d$, the commanded joint velocity $\dot{\boldsymbol{q}}_d$, and the commanded joint acceleration $\ddot{\boldsymbol{q}}_d$ of the robot with payload at each sampling moment. The output of the algorithm is the inertial parameters of the payload $\boldsymbol{\varphi}_L$. The proposed algorithm mainly includes three steps. The first step is initialization. The forgetting factor ξ, the covariance matrix $\boldsymbol{P}^*$, and the base parameters $\boldsymbol{\pi}_b$ of the robot with payload should be initialized. The initial value of forgetting factor ξ is set to 100. The initial value of covariance matrix $\boldsymbol{P}^*$ is set to 10 times $p \times p$ identity matrix. The initial value of base parameters with payload $\boldsymbol{\pi}_b$ is set to the one without payload $\boldsymbol{\pi}_a$. The second step is the RLS iteration to solve $\boldsymbol{\pi}_b$. When the robot tracks an excitation trajectory, $\boldsymbol{\pi}_b$ can be computed online in real time by the RLS method (namely, the **while** loop) until it converges. If $\boldsymbol{\pi}_b$ has not converged, the regression matrix $\boldsymbol{Y}_b$ can be first computed according to (7). Then the gain matrix $\boldsymbol{K}^*$ can be computed by (8). Afterward, the joint torque residual vector $\boldsymbol{e}$ can be computed by (9). Finally, the base parameters $\boldsymbol{\pi}_b$ and covariance matrix $\boldsymbol{P}^*$ can be updated through (10) and (11), respectively. Once $\boldsymbol{\pi}_b$ has reached the convergence condition, the iteration process ends. The convergence condition is that the consecutive times for $\Delta\pi_{b,max} < 0.005$ are more than 300. $\Delta\pi_{b,max}$ is the maximum absolute difference of the estimated inertial base parameters such as $\pi_{b,31} \sim \pi_{b,36}$ at adjacent sampling moments. At the k-th sampling, $\Delta\pi_{b,max_k}$ can be computed by $\Delta\pi_{b,max_k} = \max(\Delta\pi_{b,31_k}, \Delta\pi_{b,32_k}, \Delta\pi_{b,33_k}, \Delta\pi_{b,34_k}, \Delta\pi_{b,35_k}, \Delta\pi_{b,36_k})$. $\max(\cdot)$ is a function to find the maximum value among the input. For example, $\Delta\pi_{b,31_k}$ can be computed by $\Delta\pi_{b,31_k} = |\pi_{b,31_k} - \pi_{b,31_{k-1}}|$, where $\pi_{b,31_k}$ represents the estimated $\pi_{b,31}$ at k-th sampling. The third step is to solve the payload parameters online by using the symbolic relationship between the parameter difference and the payload parameters. Once $\boldsymbol{\pi}_b$ was solved, the numerical results of parameter difference $\boldsymbol{\varepsilon}$ can be solved by (6). According to the derivation in the appendix, it can be found that there exists a linear relationship between parameter difference and the inertial parameters of the payload. Based on this algebraic relationship, the inertial parameters of the load can be quickly solved. After extracting the nonzero elements $\boldsymbol{\varepsilon}_{nz}$ from the parameter difference, the payload parameters $\boldsymbol{\varphi}_L$ can be calculated by (12)–(16). For example, according to the symbolic expressions of the nonzero elements of parameter difference in the appendix, $I_{xy,L}$, $I_{xz,L}$, $I_{yz,L}$, $I_{zz,L}$, $m_L r_{x,L}$, and $m_L r_{y,L}$ can be first solved by (12). The payload

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Algorithm 1. The proposed online payload identification algorithm

Require:

$\boldsymbol{\tau}_b, \boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d, \boldsymbol{\pi}_a, \boldsymbol{\alpha}_a$

Ensure:

$\boldsymbol{\varphi}_L$

Initialize $\xi = 100$;
Initialize $\boldsymbol{P}^* = 10 \cdot \text{eye}(p, p)$;
Initialize $\boldsymbol{\pi}_b = \boldsymbol{\pi}_a$;
while $\boldsymbol{\pi}_b$ not converge **do**
  Solve $\boldsymbol{Y}_b$ by (7);
  Solve $\boldsymbol{K}^*$ by (8);
  Solve $\boldsymbol{e}$ by (9);
  Update $\boldsymbol{\pi}_b$ by (10);
  Update $\boldsymbol{P}^*$ by (11);
end while
Solve $\boldsymbol{\varepsilon}$ by (6) and then extract $\boldsymbol{\varepsilon}_{nz}$;
Solve $I_{xy,L}$, $I_{xz,L}$, $I_{yz,L}$, $I_{zz,L}$, $m_L r_{x,L}$, and $m_L r_{y,L}$ by (12);
Solve m_L by (13) and solve $m_L r_{z,L}$ by (14);
Solve $I_{yy,L}$ by (15) and solve $I_{xx,L}$ by (16).

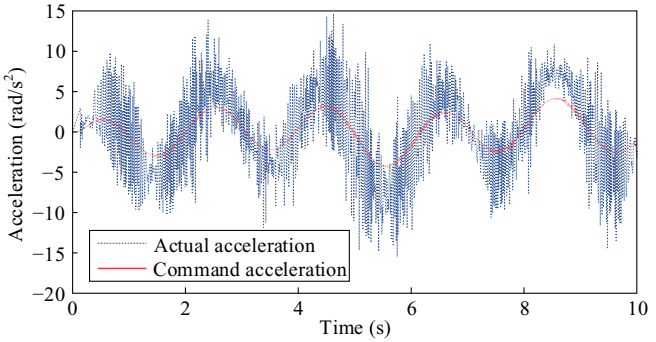


Figure 1. *Actual acceleration and command acceleration.*

mass m_L can be then solved by (13). Afterward, $m_L r_{z,L}$ can be solved by (14). After that, $I_{yy,L}$ can be solved by (15). Finally, $I_{xx,L}$ can be solved by (16).

The proposed algorithm considers the nonlinear friction coefficient $\boldsymbol{\alpha}_a$ when calculating the regression matrix. In fact, the nonlinear friction model (2) is beneficial to improve the identification accuracy of $\boldsymbol{\pi}_a$ [29, 39]. In addition, the proposed algorithm uses commanded joint trajectory data ($\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d$) as computational input rather than the actual joint trajectory data. This is because the commanded joint acceleration signal has low noise. But the actual joint acceleration signals are obtained by performing two differentials on the actual joint position, which will introduce too much noise as shown in Fig. 1 and reduce the identification accuracy of the payload. In fact, the commanded trajectory signals can replace the actual ones as the calculation input because the commercial robots usually have a good trajectory tracking accuracy. In this article, the command acceleration signals can be acquired from the UR10 robot's controller. The joint clearances and compliance of links might affect the accuracy of the proposed method when applying the proposed method to other robot platforms with significant joint flexibility (e.g., robots equipped with both harmonic drives and joint torque sensors) because they have effects on the accuracy of dynamic modeling and identification. However, in this article, due to the high stiffness of joints and links of UR10 robot, the effects of joint clearance and link compliance can be

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

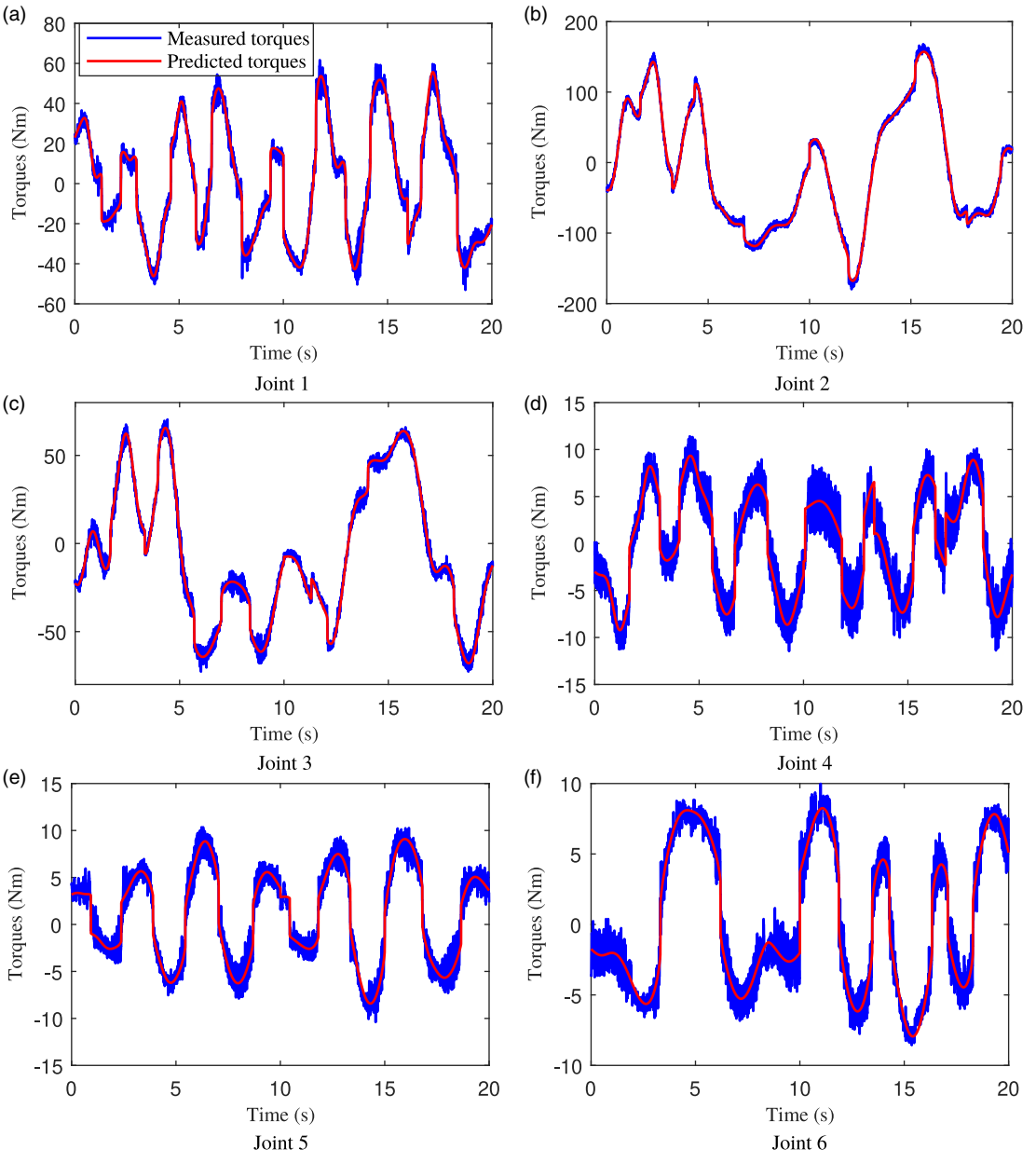


Figure 2. *Joint torque prediction using identified dynamics of UR10 robot.*

usually ignored in dynamic modeling [50]. For example, in this article, the identified dynamics can be used to predict the measured torques accurately as shown in Fig. 2, and the root mean square errors (RMSE) are shown in Table II.

4. Experiments

4.1. Experiment platform

The platform used in this paper is UR10 robot. The kinematic frames of the robot is shown in Fig. 3 and the modified DH (MDH) parameters are listed in Table III. The UR10 robot is controlled by a C++

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Table II. *RMSE of torque errors between predicted torques and measured torques.*

Index	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Unit
RMSE	3.7817	4.1789	3.0339	1.7334	1.0670	0.9896	Nm

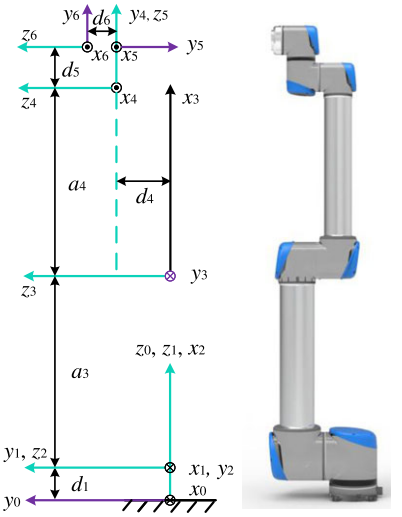


Figure 3. *The kinematic frames of UR10 robot ($n = 6$ and $p = 58$).*

and Python interface *ur_rtde* [51]. It can receive data and control the robot by using the real-time data exchange in the robot controller. In this article, the authors use the Python programming language to code the robot. The host computer that controls the robot is a laptop with a Linux operating system. Its CPU is i5-8250U-1.6 Ghz and the memory is 16 GB.

4.2. Data acquisition and data processing

The excitation trajectory for identification was based on Fourier series [25]. For the j-th joint, the excitation trajectory equation of joint position q_j is as follows:

$$q_j(t) = q_{j,0} + \sum_{l=1}^{5} \left[\frac{a_{j,l}}{\omega_f l} \sin(\omega_f l t) - \frac{b_{j,l}}{\omega_f l} \cos(\omega_f l t) \right] \tag{17}$$

where both $a_{j,l}$ and $b_{j,l}$ are the excitation trajectory coefficients to be optimized, which are shown in Table IV. $\frac{\omega_f}{2\pi}$ is the trajectory fundamental frequency, which is set to 0.1 Hz. The trajectory period is 10 s in this article. $q_{j,0}$ is the offset constant of joint position, which is set to $[0, -\pi/2, 0 - \pi/2, 0, 0]$.

As mentioned in the introduction, incentive effects can be improved by optimizing the sub-matrix condition number of regressor [27, 28]. Additionally, considering measurement noise covariance during optimization and conducting the process iteratively can better capture the dynamic characteristics of the robot [29]. Therefore, the objective function is to minimize the condition numbers of the regressor and its subregressors in this article.

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Table III. *UR10 MDH parameters.*

Joint j	$\boldsymbol{\theta}_j$**(rad)**	$\boldsymbol{d}_j$**(m)**	$\boldsymbol{a}_j$**(m)**	$\boldsymbol{\alpha}_j$**(rad)**
1	0	0.1273	0	0
2	−1.5708	0	0	−1.5708
3	0	0	0.6120	0
4	−1.5708	0.163941	0.5723	0
5	0	0.1157	0	−1.5708
6	0	0.0922	0	1.5708

Table IV. *Trajectory parameters for online payload identification trajectories.*

j	**1**	**2**	**3**	**4**	**5**	**6**
$a_{j,1}$	0.0141	0.5953	0.3104	0.2154	0.1756	−0.9952
$b_{j,1}$	−0.0380	0.1214	0.2227	−0.1627	0.2637	−0.3488
$a_{j,2}$	−0.9433	−0.6218	0.1947	0.2919	0.6207	1.0164
$b_{j,2}$	0.0870	−0.3031	−0.5874	0.7448	−0.5349	0.8982
$a_{j,3}$	0.0123	0.0122	0.3561	−0.3335	−0.0351	0.0161
$b_{j,3}$	0.0007	0.0144	0.3178	−0.5967	−0.0797	−0.1894
$a_{j,4}$	0.0011	0.0253	−0.6303	0.1840	0.1186	−0.0023
$b_{j,4}$	−0.0020	0.0854	−0.3873	−0.3257	−0.1270	−0.0219
$a_{j,5}$	0.9158	−0.0109	−0.2308	−0.3578	−0.8798	−0.0349
$b_{j,5}$	−0.0260	0.0200	0.3096	0.3532	0.3107	−0.1583

$$
\begin{aligned}
\min_{\boldsymbol{\gamma}} \quad & \|10 \cdot \mathrm{cond}(\overline{\boldsymbol{Y}}^*(\boldsymbol{\gamma})) + 5 \cdot \mathrm{cond}(\overline{\boldsymbol{Y}}^*_i(\boldsymbol{\gamma})) + \mathrm{cond}(\overline{\boldsymbol{Y}}^*_g(\boldsymbol{\gamma})) + \mathrm{cond}(\overline{\boldsymbol{Y}}^*_f(\boldsymbol{\gamma}))\|_2 \\
\text{s.t.} \quad & \left[\textstyle\sum_{l=1}^{5} \frac{-b_{j,l}}{\omega_f l}, \sum_{l=1}^{5} a_{j,l}, \sum_{l=1}^{5} \omega_f l b_{j,l}\right] = \mathbf{0} \\
& \sum_{l=1}^{5} \frac{\sqrt{a_{j,l}^2 + b_{j,l}^2}}{\omega_f l} \le q_{j,max} \\
& \sum_{l=1}^{5} \sqrt{a_{j,l}^2 + b_{j,l}^2} \le \dot{q}_{j,max} \\
& \sum_{l=1}^{5} \omega_f l \sqrt{a_{j,l}^2 + b_{j,l}^2} \le \ddot{q}_{j,max}
\end{aligned}
\tag{18}
$$

where $\boldsymbol{\gamma}$ is the optimization variable. $\overline{\boldsymbol{Y}}^*$ is the weighted regressor stacked by N joint trajectory data with $\overline{\boldsymbol{Y}}^* = \boldsymbol{\Lambda}_0^{-\frac{1}{2}} \overline{\boldsymbol{Y}}$. $\overline{\boldsymbol{Y}}$ is the stacked version of $\boldsymbol{Y}$. $\boldsymbol{\Lambda}_0$ is the initial value of initial covariance matrix, which can be computed by $\boldsymbol{\Lambda}_0 = \frac{\overline{\boldsymbol{R}}_0 \cdot \overline{\boldsymbol{R}}_0^{\mathrm{T}}}{N-p}$. $\overline{\boldsymbol{R}}_0$ is the initial torque residual which can be computed by $\overline{\boldsymbol{R}}_0 = \overline{\boldsymbol{\tau}} - \overline{\boldsymbol{Y}} \boldsymbol{\pi}_{ols}$. $\overline{\boldsymbol{\tau}}$ is the stacked version of $\boldsymbol{\tau}$. $\boldsymbol{\pi}_{ols}$ is the ordinary least squares solution of $\boldsymbol{\pi}$. $\overline{\boldsymbol{Y}}^*_i$, $\overline{\boldsymbol{Y}}^*_g$, and $\overline{\boldsymbol{Y}}^*_f$ are the inertia tensor sub-regressor, first moment vector sub-regressor, and linear friction sub-regressor of $\overline{\boldsymbol{Y}}^*$, respectively. $q_{j,max}$, $\dot{q}_{j,max}$, and $\ddot{q}_{j,max}$ are the limit of joint position, limit of joint velocity, and limit of joint acceleration, respectively. cond(·) is a function to solve the condition number of the input.

One can refer to refs. [25, 29] for more details about the excitation trajectory optimization. The optimized excitation trajectory with joint position, joint velocity, joint acceleration, and joint torque are shown in Fig. 4, which can give the condition number of the stacked regressor matrix 79.67. As described in ref. [34], the dynamic characteristics of a robot can be well excited if the condition number of regressor is less than 100. Therefore, the optimized excitation trajectory in this article is well conditioned and

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

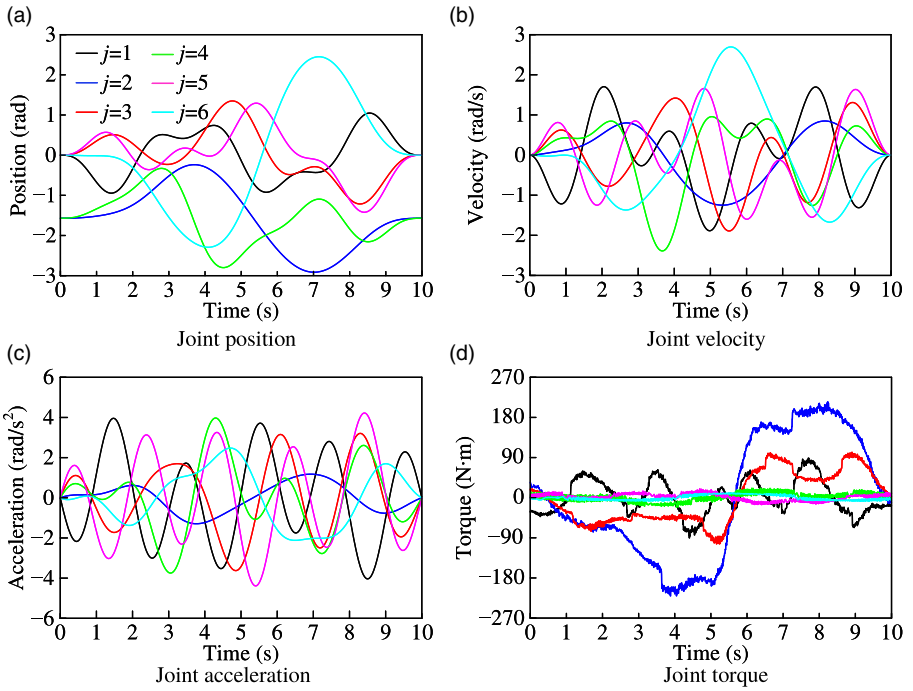


Figure 4. *The optimized excitation trajectory for online payload identification.*

adequate for exciting the whole dynamics of the robot. The rate of the real-time interface can be set to 125 Hz for CB-Series UR robot. For the online payload identification, no filtering is used for the sampled joint trajectory data.

Because the UR10 robot has no joint torque sensors, the joint torques can be calculated by $\boldsymbol{\tau} = \boldsymbol{Ki}$, in which $\boldsymbol{K}$ is the joint drive gains and $\boldsymbol{i}$ is the motor currents. The $\boldsymbol{K}$ of UR10 robot has been identified by the method in [52]. One can refer to our previous work [52] for more details because the identification process of $\boldsymbol{K}$ is not the main contribution in this paper. The identified results of joint drive gains are shown in Table V. In fact, the joint drive gains $\boldsymbol{K}$ refer to the product of the motor torque constant $\boldsymbol{k}$, gear ratio $\boldsymbol{\eta}$, and current amplification factor $\boldsymbol{\psi}$, namely, $\boldsymbol{K} = \boldsymbol{k} \cdot \boldsymbol{\eta} \cdot \boldsymbol{\psi}$ [53]. The identified joint drive gains directly establish the relationship between joint torques and motor currents; therefore, the identified results in Table V include the reduction ratio. For the offline identification of $\boldsymbol{\pi}_a$, the Butterworth filter was used for the estimated joint torques, and the frequency filtering was used for estimating the joint velocities and joint accelerations. The Butterworth filter order is set to 4, and the cutoff frequency is set to 40 Hz.

4.3. Identification results

The $\boldsymbol{\pi}_a$ and the $\boldsymbol{\alpha}_a$ of UR10 robot without payload have been identified offline in ref. [39]. Since the identification process of both $\boldsymbol{\pi}_a$ and $\boldsymbol{\alpha}_a$ is not the main contribution in this paper, only the identification results of them are shown in Table VI, and the identification process is omitted. One can refer to our previous work [39] for more identification details. When the $\boldsymbol{\pi}_a$ and $\boldsymbol{\alpha}_a$ are ready, the online payload identification can start. Firstly, we make the UR10 robot track the designed trajectory with a concentric payload as shown in Fig. 5. The true values of the payload parameters are measured by the CAD software. Then, the proposed algorithm starts running at the same time. Since the identification process is performed online, no filtering is used to improve calculation efficiency. In addition, since the commanded trajectory signal has less noise, the commanded positions, the commanded velocities, and the

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Table V. *Joint drive gains of UR10 robot.*

K_1	K_2	K_3	K_4	K_5	K_6	Unit
14.73	14.33	11.55	11.25	11.50	11.50	Nm/A

Table VI. *$\boldsymbol{\pi}_a$ and $\boldsymbol{\alpha}_a$ of UR10 robot by offline identification*[1].

$\boldsymbol{\pi}_a$	Value	$\boldsymbol{\pi}_a$	Value	$\boldsymbol{\pi}_a$	Value	$\boldsymbol{\pi}_a$	Value
$\pi_{a,1}$	8.9808	$\pi_{a,17}$	0.0395	$\pi_{a,33}$	−0.0028	$\pi_{a,49}$	15.1400
$\pi_{a,2}$	−4.6873	$\pi_{a,18}$	0.0314	$\pi_{a,34}$	0.0096	$\pi_{a,50}$	2.7746
$\pi_{a,3}$	−0.0131	$\pi_{a,19}$	0.0253	$\pi_{a,35}$	−0.0048	$\pi_{a,51}$	3.4578
$\pi_{a,4}$	−1.0510	$\pi_{a,20}$	0.0435	$\pi_{a,36}$	−0.0048	$\pi_{a,52}$	3.5335
$\pi_{a,5}$	0.1458	$\pi_{a,21}$	0.0118	$\pi_{a,37}$	0.2141	$\pi_{a,53}$	−0.3426
$\pi_{a,6}$	6.4621	$\pi_{a,22}$	0.2863	$\pi_{a,38}$	−0.3865	$\pi_{a,54}$	0.0582
$\pi_{a,7}$	8.1662	$\pi_{a,23}$	−0.0047	$\pi_{a,39}$	−0.3508	$\pi_{a,55}$	0.1780
$\pi_{a,8}$	−0.0811	$\pi_{a,24}$	0.0016	$\pi_{a,40}$	−0.2697	$\pi_{a,56}$	0.6453
$\pi_{a,9}$	−2.1302	$\pi_{a,25}$	−0.0008	$\pi_{a,41}$	10.0037	$\pi_{a,57}$	0.6725
$\pi_{a,10}$	−0.0096	$\pi_{a,26}$	−0.0142	$\pi_{a,42}$	−0.1240	$\pi_{a,58}$	0.5685
$\pi_{a,11}$	−0.5125	$\pi_{a,27}$	−0.0057	$\pi_{a,43}$	5.1803	$\alpha_{a,1}$	0.5758
$\pi_{a,12}$	−0.0379	$\pi_{a,28}$	0.0017	$\pi_{a,44}$	2.7800	$\alpha_{a,2}$	0.2723
$\pi_{a,13}$	2.1103	$\pi_{a,29}$	−0.0587	$\pi_{a,45}$	2.0115	$\alpha_{a,3}$	0.6093
$\pi_{a,14}$	3.8882	$\pi_{a,30}$	0.0012	$\pi_{a,46}$	1.7533	$\alpha_{a,4}$	0.9995
$\pi_{a,15}$	−0.0026	$\pi_{a,31}$	−0.0002	$\pi_{a,47}$	21.6084	$\alpha_{a,5}$	0.8031
$\pi_{a,16}$	0.0596	$\pi_{a,32}$	−0.0013	$\pi_{a,48}$	26.4096	$\alpha_{a,6}$	0.7209

[1]All the units are SI unit.

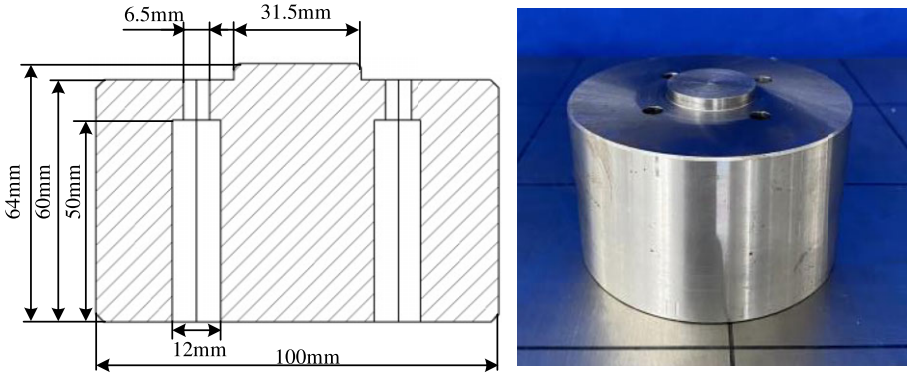


Figure 5. *Concentric payload for online identification.*

commanded accelerations are used to compute $\boldsymbol{Y}_b$. After online iteration calculation by RLS method, the $\boldsymbol{\pi}_b$ converged. In order to save space, the convergence process of part of $\boldsymbol{\pi}_b$ is shown in Fig. 6. From the figure, it can be seen that the base parameters $\pi_{b,31} \sim \pi_{b,36}$ converged at about 6.59 s because the consecutive times for $\Delta\pi_{b,max} < 0.005$ have exceeded 300 at that moment. Once $\boldsymbol{\pi}_b$ converged, the payload parameters can be estimated by using the symbolic relationship between the parameter difference and the payload parameters, namely, (12)–(16). Since the calculation of (12)–(16) is simple and very fast, therefore, the computation times for the online payload identification by using the proposed method are about 6.59 s.

Once $\boldsymbol{\pi}_b$ is solved online, the numerical results of $\boldsymbol{\varepsilon}$ can be computed by (6). Then, the nonzero elements $\boldsymbol{\varepsilon}_{nz}$ can be extracted from $\boldsymbol{\varepsilon}$ according to the appendix. Afterward, the payload parameters can

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

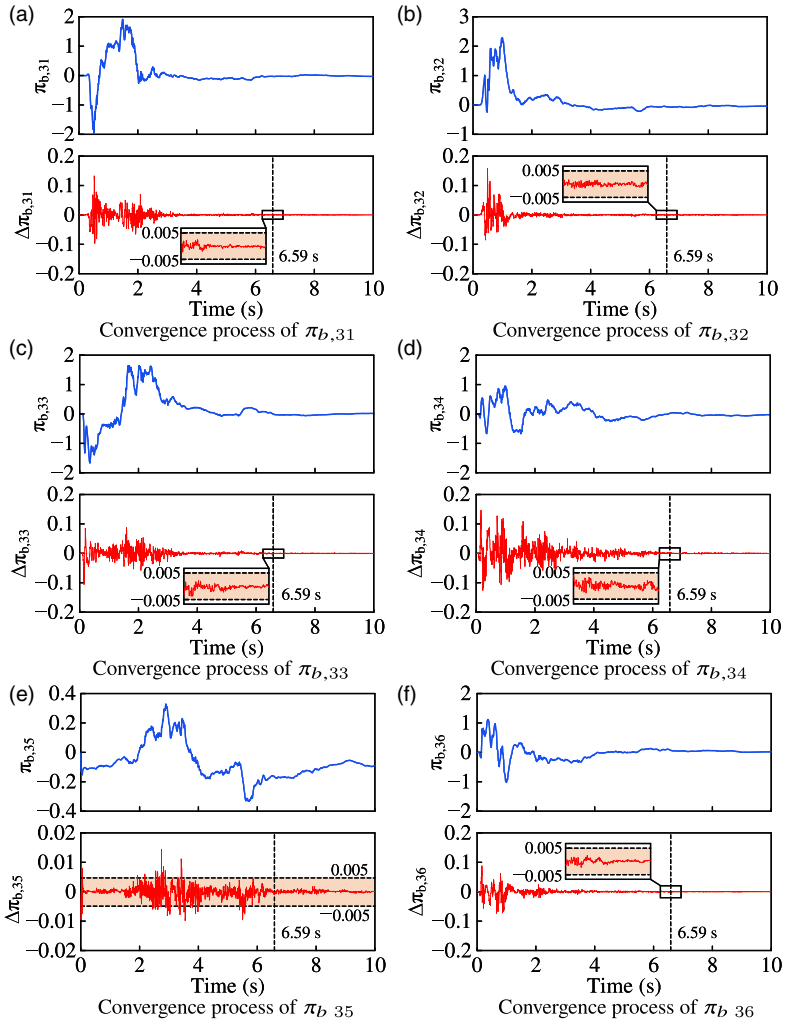


Figure 6. *Convergence process of the part of* $\boldsymbol{\pi}_b$ *during online payload identification.*

be calculated online by (12)–(16). The identified results of $\boldsymbol{\varphi}_L$ by the proposed algorithm are shown in Table VII. In addition, the identified results of $\boldsymbol{\varphi}_L$ by another two classical algorithms are also shown in Table VII as comparison. The algorithm in ref. [33] is named as Algorithm 1 and the other one in ref. [36] is named as Algorithm 2. The principle of Algorithm 1 is to use the changes between the joint torques before and after loading the robot on the same trajectory. It distinguished the contribution of the robot itself and the payload to the joint torques. The robot link parameters without load were considered as the prior information for the online payload identification. However, the estimation results in this method are biased. The principle of Algorithm 2 is based on the relationship between robot static coefficients and inertial parameters of payload. However, this method cannot obtain the inertia tensor of the payload because it lacks the dynamic excitation.

According to these data, it can be found that the identification error of the proposed algorithm is 0.04% for the payload mass, and the ones of Algorithm 1 and Algorithm 2 are 4.11% and 0.92%, respectively. The proposed algorithm can improve the online identification accuracy of the payload mass by about 103 times and 23 times compared with Algorithm 1 and Algorithm 2, respectively. Therefore, the identification accuracy of the proposed algorithm is higher than that of Algorithm 1 and Algorithm 2 for the payload mass. For the CoM in Y axis of the payload, the proposed algorithm has the highest

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Table VII. *Online identification results of the payload by three methods.*

φ_L^1	Truth[2]	Algorithm 1 [33]	Algorithm 2 [36]	Proposed
m_L	3.5256	3.3807	3.5581	3.5242
$r_{x,L}$	0	−0.0145	−0.0069	−0.0213
$r_{y,L}$	0	0.0082	−0.0068	0.0016
$r_{z,L}$	0.0296	0.0577	0.0449	0.0516
$I_{xx,L}$	0.0064	0.1122	-	0.0023
$I_{xy,L}$	0	0.0077	-	−0.0289
$I_{xz,L}$	0	−0.0012	-	−0.0480
$I_{yy,L}$	0.0064	0.0557	-	−0.0958
$I_{yz,L}$	0	0.0126	-	0.0265
$I_{zz,L}$	0.0045	−0.1545	-	−0.0501

[1]The units of mass, CoM, and inertia tensor are kg, m, and kg · m^2, respectively.

[2]The true values were measured by the CAD software, which are relative to the frame $x_6y_6z_6$ in Fig. 3.

identification accuracy. For the CoM in Z axis of the payload, the identification error of the proposed algorithm (74.32%) is smaller than Algorithm 1 (94.93%) and higher than Algorithm 2 (51.69%). It can be found that all the three algorithms cannot accurately identify the CoM of the payload online. This is because the online payload identification uses a small amount of data to improve the computation efficiency. Therefore, it is difficult to obtain the same identification accuracy of CoM with that of the offline payload identification. For the inertia tensor of payload, both the proposed algorithm and Algorithm 1 are not accurate enough. Usually, it is not easy to accurately identify the inertia tensor of the payload neither online payload identification nor offline payload identification, especially when the payload mass is lighter than the robot mass [2, 39].

According to the experimental results, it can be found that the proposed algorithm has the highest online identification accuracy for the payload mass. The CoM online identification accuracy of the proposed algorithm is better than Algorithm 1. Although the CoM online identification accuracy of Algorithm 2 is higher than the proposed algorithm, the Algorithm 2 cannot identify the inertia tensor of the payload online. Consistent with existing algorithms, both the proposed algorithm and Algorithm 1 have deficiencies in the online identification accuracy of payload inertia tensor.

As mentioned above, the nonlinear friction model is beneficial to improve the identification accuracy of $\boldsymbol{\pi}_a$. And then $\boldsymbol{\pi}_a$ affects the online identification accuracy of payload. As shown in Fig. 7, the nonlinear friction model fits the estimated friction torque better than the linear one in both unloaded and loaded conditions. The used linear friction model (green circle) is $\tau_{f,j} = F_{c,j}\text{sign}(\dot{q}_j) + F_{v,j}\dot{q}_j + B_j$ [44]. The parameters of this model for joint 2 is $F_{c,2} = 15.07$, $F_{v,2} = 10.73$, and $B_2 = 8.77 \times 10^{-3}$ at the case without payload. The parameters of this model for joint 2 is $F_{c,2} = 15.87$, $F_{v,2} = 10.77$, and $B_2 = 6.47 \times 10^{-8}$ at the case with payload. The used nonlinear friction model (red dot) is $\tau_{f,j} = (F_{c,j} + F_{v,j}|\dot{q}_j|^{\alpha_j})\text{sign}(\dot{q}_j) + B_j$, namely, (2) [29, 45]. The parameters of this model for joint 2 is $\alpha_2 = 0.2723$, $F_{c,2} = -0.1240$, $F_{v,2} = 26.4096$, and $B_2 = 0.0582$ at the case without payload. The parameters of this model for joint 2 is $\alpha_2 = 0.2603$, $F_{c,2} = -0.5500$, $F_{v,2} = 27.7899$, and $B_2 = -0.1836$ at the case with payload. Especially when the joint velocity is close to zero, the linear friction model will introduce more fitting errors. As shown in Table VIII, it can be found that the RMSE of the nonlinear friction model is smaller than that of the linear one. Therefore, it benefits to improving the accuracy of online payload identification by using a nonlinear friction model. In addition, it is worth explaining that $\boldsymbol{\alpha}_a$ was used as the input for online payload identification. On the one hand, it is difficult to identify the nonlinear friction coefficient $\boldsymbol{\alpha}_b$ with payload in each iteration online. On the other hand, the payload has little effect on the nonlinear friction coefficients except for joint 3 as shown in Table IX. In order to compute the estimated friction torques, the identified friction parameters in $\boldsymbol{\pi}$ are firstly set to zero, then the torques related to the inertial component in $\boldsymbol{\pi}$ can be subtracted from the measured torques [54]. For example, the estimated

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

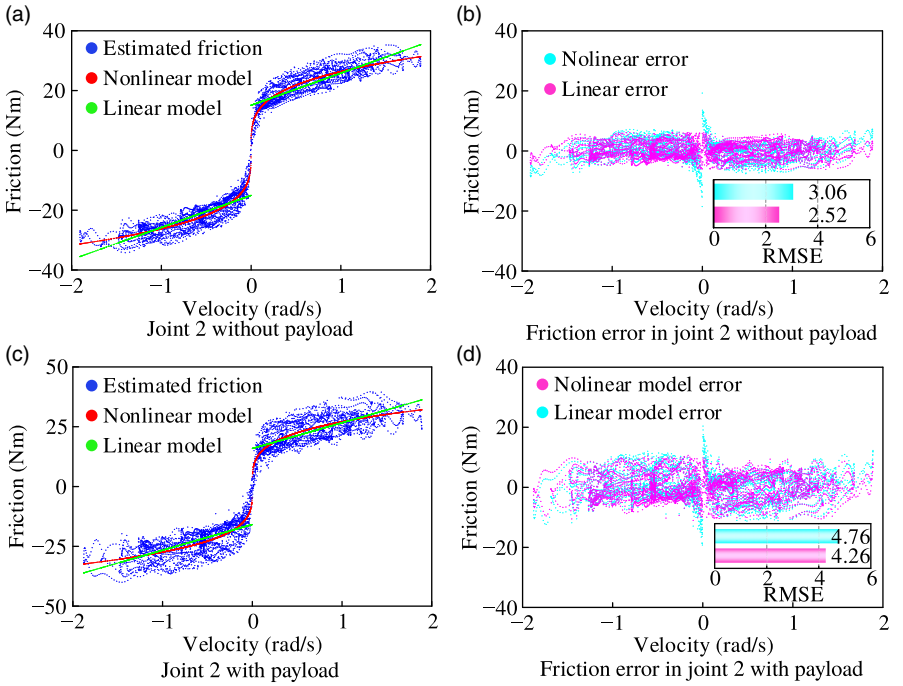


Figure 7. *Comparison between linear friction and nonlinear friction in joint 2 of UR10 robot without and with payload.*

friction torques $\hat{\boldsymbol{\tau}}_f = \boldsymbol{\tau} - \boldsymbol{Y} \cdot \left[\hat{\boldsymbol{\pi}}_i{}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}\right]^{\mathrm{T}}$, in which $\hat{\boldsymbol{\tau}}_f$ is the estimated friction torques, $\boldsymbol{\tau}$ is the measured joint torques, $\hat{\boldsymbol{\pi}}_i$ is the identified inertial parameter component in $\boldsymbol{\pi}$.

The temperature range of the robot joints at which the experiments are performed is shown in Fig. 8. The authors make the robot to execute the online identification excitation trajectory 10 times and collect the dataset from the robot controller. There was an approximate 10 s interval between each movement. Typically, before the online payload identification experiment, the authors first let the robot run 3 times to ensure the robot's status and the code being functioning properly. Therefore, the 10 datasets could fully cover the temperature range of the experiment at that time. From the data in the graph, it can be observed that the mean temperature range of all joints is between 25.5 °C and 30 °C.

4.4. An application of online payload identification

In this subsection, an application experiment of online payload identification is shown. Firstly, one uses the proposed algorithm to identify the inertial parameters of a new payload online. Then, the identified inertia parameters of the payload are compensated to the robot controller. Afterward, the teaching mode of UR10 robot can be activated. In this mode, the user can drag the robot to move freely, and the guidance forces can be recorded by a force/torque sensor. The force/torque sensor used in this work is the ATI MINI45-E. The sensor has a diameter of 45 mm, a height of 15.7 mm, and a mass of 0.0917 kg. The maximum force range is ±580 N in X-axis and Y-axis, ±1160 N in Z-axis. The maximum torque range is ±20 Nm in X-axis, Y-axis, and Z-axis. The force resolution of the sensor is 1/4 N. The torque resolution of the sensor is 1/188 Nm in X-axis and Y-axis and 1/376 Nm in Z-axis. Since the force/torque sensor has a non-detachable connecting wire, it is inconvenient to identify it together with the payload. Therefore, the payload is first identified online to prevent the connecting wire from being torn off when the robot executes the excitation trajectory. Then, the force/torque sensor and the guidance handle are fixed to the

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Table VIII. *RMSE[1] of different friction models without and with payload.*

Payload	Friction	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Without	Linear	1.7499	3.0599	1.4170	0.3913	0.2965	0.3726
	Nonlinear	1.3102	2.5186	1.1437	0.3914	0.2793	0.3325
With	Linear	1.6726	4.7629	2.3821	0.4182	0.3712	0.3782
	Nonlinear	1.1982	4.2579	1.8470	0.4176	0.3603	0.3333

[1]The unit of RMSE is Nm.

Table IX. *The nonlinear friction coefficients $\boldsymbol{\alpha}$ of UR10 robot without and with payload.*

Payload	α_1	α_2	α_3	α_4	α_5	α_6
Without	0.5758	0.2723	0.6093	0.9995	0.8031	0.7209
With	0.5430	0.2603	0.3984	0.9568	0.8151	0.6969

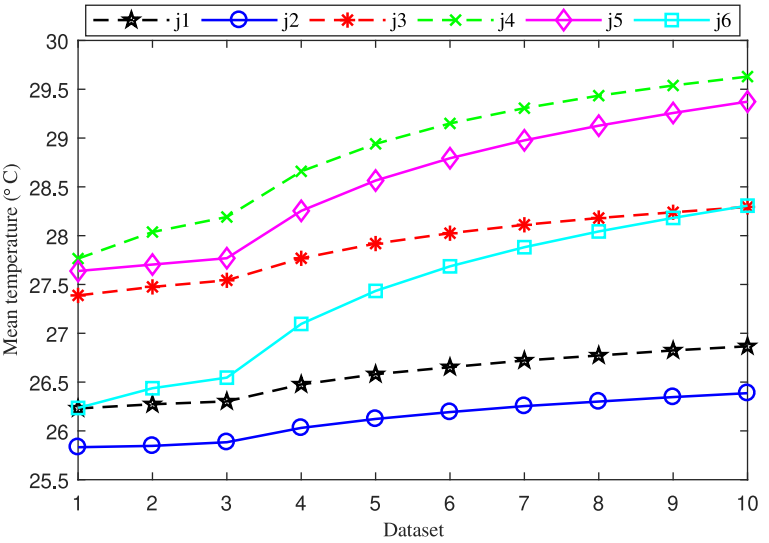


Figure 8. *The mean temperature of each joint of UR10 robot during excitation motions.*

end of the payload. After that, the manual guidance experiment can be carried out. In this application experiment, the designed payload and the assembly scheme are shown in Fig. 9. The mass of both new payload and the bolts is 3.223 kg after measurement. By using the proposed algorithm, the inertial parameters of the payload are identified online. The identified mass is 3.237 kg and the error is 0.43%. The identified CoM of the payload is [−0.024, −0.006, 0.045] m. The identified inertia tensor of the payload is [0.006, −0.012, −0.069, −0.106, 0.031, −0.014] kg · m^2. Since the UR10 robot does not provide a compensation interface for the inertia tensor of payload, the user can only compensate the mass and the CoM of the payload. In fact, the mass of the payload fixed at the end of the robot is usually much smaller than the mass of the robot itself. On the one hand, this will lead to inaccurate estimation of the inertia tensor of the payload. On the other hand, compared with the inertia tensor of the robot itself, the inertia tensor of the payload has little contribution on the dynamics of the robot. It can be speculated that UR robot manufacturer may not provide the compensation interface for payload inertia tensor due to the above reasons.

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

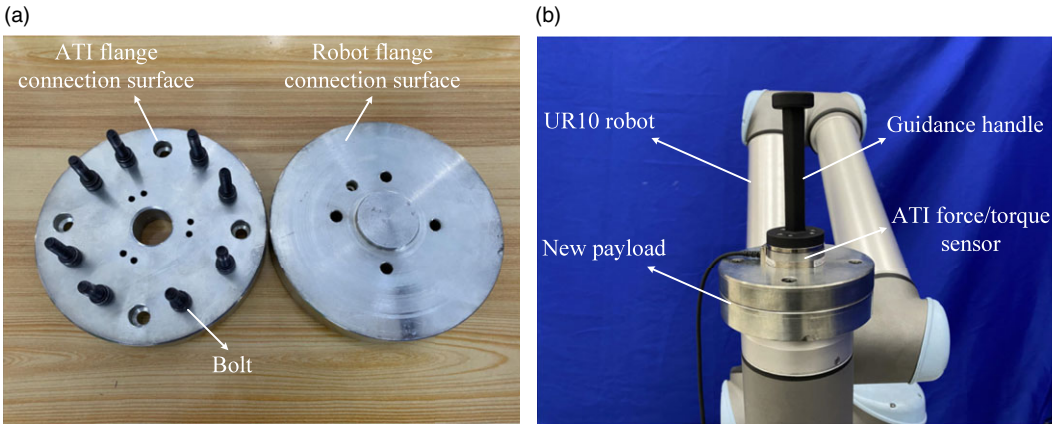


Figure 9. *New payload for the application experiment. (a) Payload. (b) Connection diagram.*

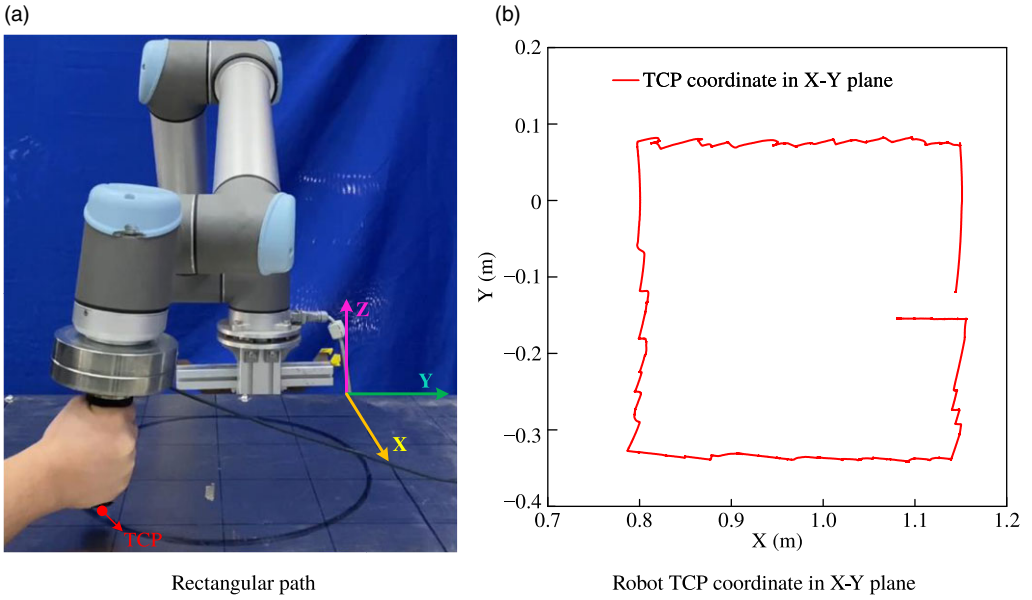


Figure 10. *The rectangular guidance path for the UR10 robot.*

After completing the online identification of the payload, the force/torque sensor and the guidance handle are fixed to the end of the payload. Then, the user can drag the robot to move along the rectangle path as shown in Fig. 10. Figure 10(b) depicts the position of the robot's Tool Center Point (TCP) on the *XY* plane of the experimental platform as shown in Fig. 10(a) when the author manually dragged the robot end effector. The TCP position of robot is sampled from the robot controller. At the same time, the guidance forces/torques and motor currents of robot are recorded as shown in Fig. 11. The mean and variance of the guidance forces without/with payload compensation are plotted in Fig. 12. From the figure, it can be seen that, compared with the ones without payload compensation, the mean of guidance forces with payload compensation can be reduced by 12.99%, 25.75%, and 44.24% in *X*-, *Y*-, and *Z*-axis, respectively. Similarly, after compensating for the payload, the variance of guidance forces can be reduced by 51.84%, 56.83%, and 64.58% in *X*-, *Y*-, and *Z*-axis, respectively. The variance of force in Fig. 12(b) is defined by $\sigma^2 = \frac{1}{M}\sum_{i=1}^{M}(x_i - \mu)^2$, in which M is the data number, x_i is the measured force

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

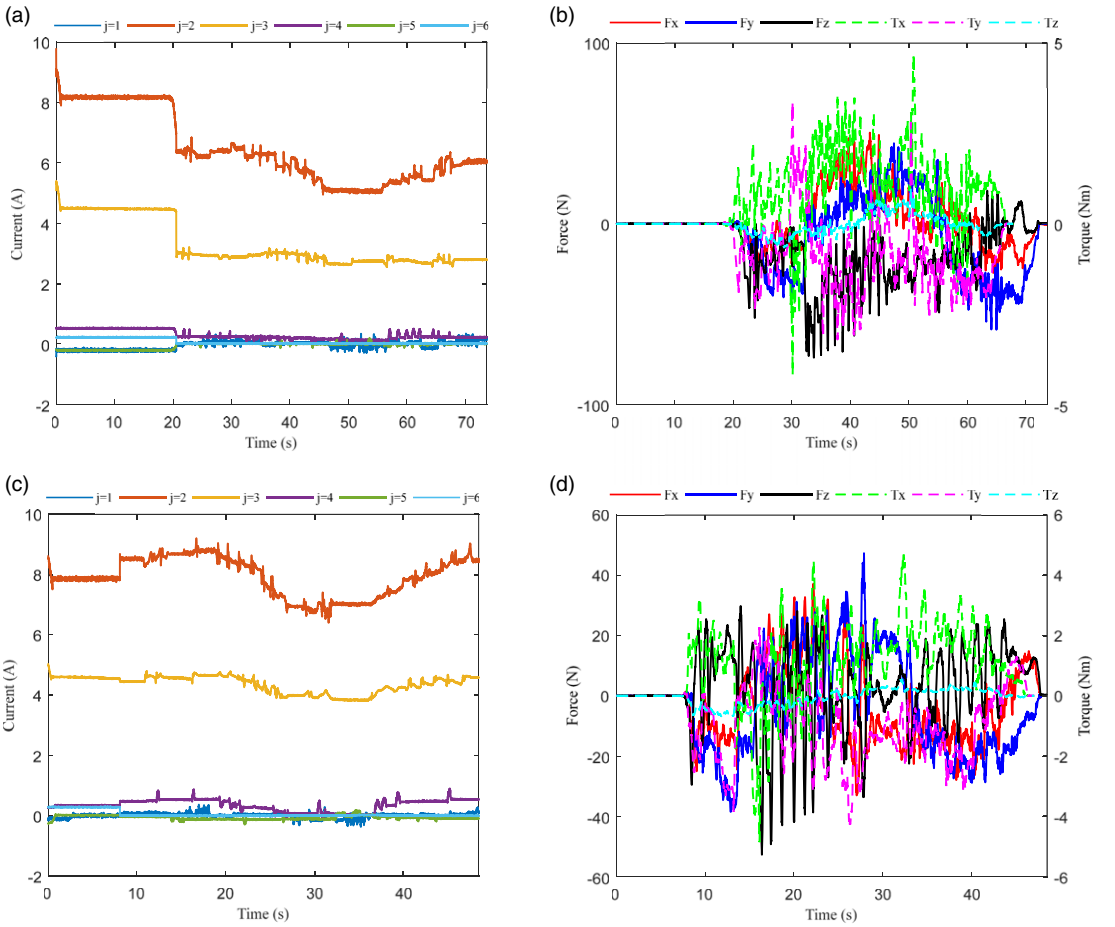
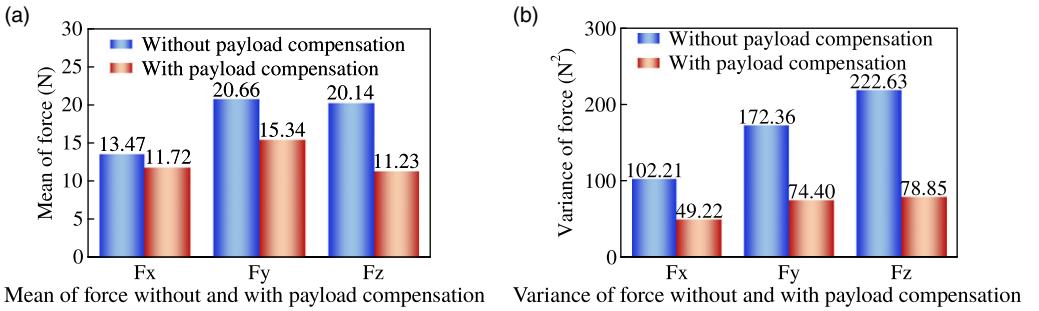


Figure 11. *Measured TCP forces/torques and motor currents of UR10 robot without and with payload compensation during the rectangular guidance path. (a) Measured joint currents without payload compensation. (b) Measured forces/torques without payload compensation. (c) Measured joint currents with payload compensation. (d) Measured forces/torques with payload compensation.*



Mean of force without and with payload compensation

Variance of force without and with payload compensation

Figure 12. *Measured TCP force in manual guidance application experiments along rectangular path.*

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

vector, and μ is the mean value of measured force vector. It can be found that the guidance forces (especially in Z-axis) are significantly reduced with payload compensation. It is because additional forces are needed to balance the payload gravity if the payload parameters are not compensated. However, the robot controller can increase currents to counteract the payload gravity once the payload parameters are compensated. Therefore, the user will feel that the robot becomes light during the guidance process. The experimental results show that the proposed algorithm can identify accurate payload parameters online (especially the payload mass), and the guidance comfort can be improved by compensating the identified payload parameters.

Based on our experience, it is not easy to drag the UR10 robot in teaching mode to make its end-effector track the rectangle in Fig. 10(b). This difficulty is unrelated to compensating for the dynamic of the load. One possible reason is that the UR10 robot lacks joint torque sensors, which makes it less smooth and easy to drag its end effector to track the rectangle in Fig. 10(b). In fact, whether compensating for load dynamics only affects the amount of force applied by the user during the dragging process. For instance, when load dynamics are not compensated for, more force is required for dragging, whereas when load dynamics are compensated for, less force is needed, as indicated by the data in Fig. 12. It is worth noting that the force/torque sensor used in this work was solely employed for quantitative analysis of dragging forces and was not utilized for any algorithm design and integration. In the teaching mode of the UR10 robot, the force/torque sensor was merely employed to record the forces applied by the user while dragging the robot's end effector.

5. Conclusions

In this paper, a symbolic relationship between the parameter difference and the payload parameters was derived. Based on the symbolic relationship, an online method is proposed to identify the payload parameters. In order to increase the online estimation accuracy, on the one hand, a nonlinear friction model (2) was used instead of the linear one. On the other hand, the commanded joint trajectory signals were introduced as the computation input instead of the actual ones. Compared with other methods, the online estimation accuracy of the payload mass m_L can be significantly improved by the proposed method. For example, the accuracy of payload mass estimated by the proposed method can be improved by 103 times and 23 times with respect to Algorithm 1 [33] and Algorithm 2 [36], respectively. According to the experimental results, the computation times by using the proposed method are about 6.59 s. When the identified dynamics of the payload were compensated, the manual guidance forces from the user can be significantly reduced and the guidance comfort can be improved. In this paper, the mean of guidance forces in Z axis was reduced by 44.24% compared with the case without payload dynamics compensation.

How to accurately identify the inertia tensor of a payload online by using proprioceptive sensors only is still a challenge that needs to be solved. One potential avenue for the future research is to reduce the computational complexity of the online identification model. More joint data can be included for online calculation, and then complex online filtering processing can be performed on the joint data. In the future, the authors will continue to develop new online payload estimation methods and further verify the new methods and the proposed one in this article on a domestic robot.

Acknowledgments. The authors would like to thank the editors and reviewers for their professional comments on the improvement of this article.

Author contributions. Conceptualization, T.X. and H.T.; methodology, T.X. and J.C.; formal analysis, T.X. and H.T.; validation, T.X. and D.S.; data curation, T.X. and Q.F.; writing–original draft preparation, T.X.; writing–review and editing, T.X. and J.F.; supervision, J.Z.; project administration, J.F. and J.Z.

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Financial support. This work was supported by the National Key R&D Program of China under Grant No. 2022YFB4701100, 2022YFB4700300, the National Natural Science Foundation of China under Grant No. 92048301, and the Pre-research Task of State Key Laboratory of Robotics and Systems (HIT) under Grant No. SKLRS202411B.

Competing interests. The authors have no relevant financial or non-financial interests to disclose.

References

[1] G. Hirzinger, N. Sporer, M. H. Alin Albu-Schaffer, R. Krenn, A. Pascucci and M. Schedl, "Dlr's Torque-Controlled Light Weight Robot iii-are We Reaching the Technological Limits Now?," **In:** *2002 IEEE International Conference on Robotics and Automation*, (IEEE, 2022) pp. 1710–1716.
[2] N. Mavrakis and R. Stolkin, "Estimation and exploitation of objects' inertial parameters in robotic grasping and manipulation: A survey," *Robot Auton Syst* **124**, 103374 (2020).
[3] C. D. B. Marko Bjelonic, Y. de Viragh, F. D. T. Dhionis Sako, F. Jenelten and M. Hutter, "Keep rollin'–whole-body motion control and planning for wheeled quadrupedal robots," *IEEE Robot Autom Lett* **4**(2), 2116–2123 (2019).
[4] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier and P. Fraisse, "Collaborative manufacturing with physical human–robot interaction," *Robot Comput Integr Manuf* **40**, 1–13 (2016).
[5] D. A. Duque, F. A. Prieto and J. G. Hoyos, "Trajectory generation for robotic assembly operations using learning by demonstration," *Robot Comput Integr Manuf* **57**, 292–302 (2019).
[6] S. L. Canfield, J. S. Owens and S. G. Zuccaro, "Zero moment control for lead-through teach programming and process monitoring of a collaborative welding robot," *J Mech Robot* **13**(3), 031016 (2021).
[7] J. Bai, S. Lian, Z. Liu, K. Wang and D. Liu, "Deep learning based robot for automatically picking up garbage on the grass," *IEEE Trans Consum Electron* **64**(3), 382–389 (2018).
[8] C. Ding, L. Zhou, Y. Li and X. Rong, "Locomotion control of quadruped robots with online center of mass adaptation and payload identification," *IEEE Access* **8**, 224578–224587 (2020).
[9] S. Abiko and K. Yoshida. On-Line Parameter Identification of a Payload Handled by Flexible Based Manipulator. **In:** *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2004) pp. 2930–2935.
[10] D. Kubus, T. Kroger and F. M. Wahl, "On-Line Estimation of Inertial Parameters Using a Recursive Total Least-Squares Approach," **In:** *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (IEEE, 2008) pp. 3845–3852.
[11] D.-Z. Feng, X.-D. Zhang, D.-X. Chang and W. X. Zheng, "A fast recursive total least squares algorithm for adaptive fir filtering," *IEEE Trans Signal Process* **52**(10), 2729–2737 (2004).
[12] M. C. Sánchez, M. Torres-Torriti and F. A. Cheein, "Online inertial parameter estimation for robotic loaders," *IFAC-PapersOnLine* **53**(2), 8763–8770 (2020).
[13] Y. Ma, X. Wu, C. Wang, Z. Yi and G. Liang, "Gait phase classification and assist torque prediction for a lower limb exoskeleton system using kernel recursive least-squares method," *Sensors* **19**(24), 5449 (2019).
[14] G. Tournois, M. Focchi, A. D. Prete, R. Orsolino, D. G. Caldwell and C. Semini, "Online Payload Identification for Quadruped Robots," **In:** *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2017) pp. 4889–4896.
[15] S. Farsoni, C. T. Landi, F. Ferraguti, C. Secchi and M. Bonfè, "Real-Time Identification of Robot Payload Using a Multirate Quaternion-Based Kalman Filter and Recursive Total Least-Squares," **In:** *2018 IEEE International Conference on Robotics and Automation (ICRA)*, (IEEE, 2018) pp. 2103–2109.
[16] S. Farsoni, C. T. Landi, F. Ferraguti, C. Secchi and M. Bonfe, "Compensation of load dynamics for admittance controlled interactive industrial robots using a quaternion-based kalman filter," *IEEE Robot Autom Lett* **2**(2), 672–679 (2017).
[17] J. Liu, P. Wang, F. Zha, W. Guo, Z. Jiang and L. Sun, "A strong tracking mixed-degree cubature kalman filter method and its application in a quadruped robot," *Sensors* **20**(8), 2251 (2020).
[18] L. Armesto, J. Tornero and M. Vincze, "Fast ego-motion estimation with multi-rate fusion of inertial and vision," *Int J Robot Res* **26**(6), 577–589 (2007).
[19] S. Farsoni, F. Ferraguti and M. Bonfè, "Safety-oriented robot payload identification using collision-free path planning and decoupling motions," *Robot Comput Integr Manuf* **59**, 189–200 (2019).
[20] A. H. Khan, S. Li and X. Luo, "Obstacle avoidance and tracking control of redundant robotic manipulator: An rnn-based metaheuristic approach," *IEEE Trans Ind Inform* **16**(7), 4670–4680 (2019).
[21] A. Renner, H. Wind and O. Sawodny, "Online payload estimation for hydraulically actuated manipulators," *Mechatronics* **66**, 102322 (2020).
[22] T. Xu, J. Fan, Y. Chen, X. Ng, M. H. Ang, Q. Fang, Y. Zhu and J. Zhao, "Dynamic identification of the kuka lbr iiwa robot with retrieval of physical parameters using global optimization," *IEEE Access* **8**, 108018–108031 (2020).
[23] X. Chang, H. An and H. Ma, "Modeling and base parameters identification of legged robots," *Robotica* **40**(3), 747–761 (2022).
[24] D. Verdel, S. Bastide, N. Vignais, O. Bruneau and B. Berret, "An identification-based method improving the transparency of a robotic upper limb exoskeleton," *Robotica* **39**(9), 1711–1728 (2021).
[25] J. Swevers, C. Ganseman, D. B. Tukel, J. de Schutter and H. Van Brussel, "Optimal robot excitation and identification," *IEEE Trans Robot Autom* **13**(5), 730–740 (1997).

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

[26] K.-J. Park, "Fourier-based optimal excitation trajectories for the dynamic identification of robots," *Robotica* **24**(5), 625–633 (2006).
[27] G. Venture, K. Ayusawa and Y. Nakamura, "A Numerical Method for Choosing Motions with Optimal Excitation Properties for Identification of Biped Dynamics-an Application to Human," **In:** *2009 IEEE International Conference on Robotics and Automation*, (IEEE, 2009) pp. 1226–1231.
[28] V. Bonnet, P. Fraisse, A.é Crosnier, M. Gautier, A. González and G. Venture, "Optimal exciting dance for identifying inertial parameters of an anthropomorphic structure," *IEEE Trans Robot* **32**(4), 823–836 (2016).
[29] Y. Han, J. Wu, C. Liu and Z. Xiong, "An iterative approach for accurate dynamic model identification of industrial robots," *IEEE Trans Robot* **36**(5), 1577–1594 (2020).
[30] L. Scalera, C. Nainer, A. Giusti and A. Gasparetto, "Robust safety zones for manipulators with uncertain dynamics in collaborative robotics," *Int J Comp Integ Manuf* 1–13 (2023).
[31] J. Swevers, W. Verdonck, B. Naumer, S. Pieters and E. Biber, "An experimental robot load identification method for industrial application," *Int J Robot Res* **21**(8), 701–712 (2002).
[32] J. Swevers, W. Verdonck and J. De Schutter, "Dynamic model identification for industrial robots," *IEEE Cont Syst Mag* **27**(5), 58–71 (2007).
[33] W. Khalil, M. Gautier and P. Lemoine, "Identification of the Payload Inertial Parameters of Industrial Manipulators," **In:** *Proceedings 2007 IEEE International Conference on Robotics and Automation*, (IEEE, 2007) pp. 4943–4948.
[34] J. Hollerbach, W. Khalil and M. Gautier, "Model Identification," **In:** *Springer Handbook of Robotics*, (Springer, 2016) pp. 113–138.
[35] S. Salah, J. Sandoval, M. Ghiss, M. A. Laribi and S. Zeghloul, "Online Payload Identification of a Franka Emika Robot for Medical Applications," **In:** *Proc Mechanisms and Machine Science*, (Springer, 2020) pp. 130–136.
[36] C. Gaz and A. De Luca, "Payload Estimation Based on Identified Coefficients of Robot Dynamics–with an Application to Collision Detection," **In:** *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2017) pp. 3033–3040.
[37] A. S.-M. Rodriguez, J. C. C. Ibanez and V. F. Battle, "Online algebraic identification of the payload changes in a single-link flexible manipulator moving under gravity," *IFAC-PapersOnLine* **47**(3), 8397–8402 (2014).
[38] J. Hu, C. Li, Z. Chen and B. Yao, "Precision motion control of a 6-dofs industrial robot with accurate payload estimation," *IEEE-ASME Trans Mechatron* **25**(4), 1821–1829 (2020).
[39] T. Xu, J. Fan, Q. Fang, Y. Zhu and J. Zhao, "An accurate identification method based on double weighting for inertial parameters of robot payloads," *Robotica* **40**(12), 4358–4374 (2022).
[40] J. Wu, G. Yu, Y. Gao and L. Wang, "Mechatronics modeling and vibration analysis of a 2-DOF parallel manipulator in a 5-DOF hybrid machine tool," *Mech Mach Theory* **121**, 430–445 (2018).
[41] J. Wu, B. B. Zhang, L. P. Wang and G. Yu, "An iterative learning method for realizing accurate dynamic feedforward control of an industrial hybrid robot," *Sci China Technol Sci* **64**(6), 1177–1188 (2021).
[42] J. Wu, H. Ye, G. Yu and T. Huang, "A novel dynamic evaluation method and its application to a 4-DOF parallel manipulator," *Mech Mach Theory* **168**, 104627 (2022).
[43] A. Kurdas, M. Hamad, J. Vorndamme, N. Mansfeld, S. Abdolshah and S. Haddadin, "Online Payload Identification for Tactile Robots using the Momentum Observer," **In:** *2022 International Conference on Robotics and Automation (ICRA)*, (IEEE, 2022) pp. 5953–5959.
[44] P. Zhang and J. Zhang, "Lower limb exoskeleton robots' dynamics parameters identification based on improved beetle swarm optimization algorithm," *Robotica* **40**(8), 2716–2731 (2022).
[45] N. Farhat, V. Mata, A. Page and F. Valero, "Identification of dynamic parameters of a 3-DOF RPS parallel manipulator," *Mech Mach Theory* **43**(1), 1–17 (2008).
[46] M. Gautier and W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Trans Robot Autom* **6**(3), 368–373 (1990).
[47] W. Khalil and F. Bennis, "Comments on "direct calculation of minimum set of inertial parameters of serial robots," *IEEE Trans Robot Autom* **10**(1), 78–79 (1994).
[48] J. M. Ferguson, T. E. Ertop, S. D. Herrell and R. J. Webster, "Unified robot and inertial sensor self-calibration," *Robotica* **41**(5), 1590–1616 (2023).
[49] M. Abedinifar, S. Ertugrul and S. H. Arguz, "Nonlinear model identification and statistical verification using experimental data with a case study of the ur5 manipulator joint parameters," *Robotica* **41**(4), 1348–1370 (2023).
[50] C. Gaz, E. Magrini and A. De Luca, "A model-based residual approach for human-robot collaboration during manual polishing operations," *Mechatronics* **55**, 234–247 (2018).
[51] A. P. Lindvig, Sdu robotics/ ur_rtde (2021). Accessed July 17, 2021. *Available at:* https://gitlab.com/sdurobotics/ur_rtde.
[52] T. Xu, J. Fan, Q. Fang, Y. Zhu and J. Zhao, "Robot dynamic calibration on current level: Modeling, identification and applications," *Nonlinear Dyn* **109**(4), 2595–2613 (2022).
[53] M. Gautier and S. Briot, "Global identification of joint drive gains and dynamic parameters of robots," *J Dynam Syst Measure Cont* **136**(5), 051025 (2014).
[54] Z. Li, H. Wei, C. Liu, Y. He, G. Liu, H. Zhang and W. Li, "An improved iterative approach with a comprehensive friction model for identifying dynamic parameters of collaborative robots," *Robotica* **42**(5), 1–23 (2024).

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Appendix

The symbolic expressions of $\boldsymbol{\pi}_a$ of UR10 robot without payload are shown as follows:

$$\pi_{a,1} = I_{a,1} + I_{zz,1} + \sum_{j=2}^{4} I_{yy,j} + 2d_4 m_4 r_{z,4} + a_3^2 m_3 + (a_3^2 + a_4^2 + d_4^2) \sum_{j=4}^{6} m_j,$$

$$\pi_{a,2} = I_{xx,2} - I_{yy,2} - a_3^2 \sum_{j=3}^{6} m_j,\ \pi_{a,3} = I_{xy,2},$$

$$\pi_{a,4} = I_{xz,2} - a_3[m_3 r_{z,3} + m_4 r_{z,4} + d_4 \sum_{j=4}^{6} m_j],\ \pi_{a,5} = I_{yz,2},$$

$$\pi_{a,6} = I_{a,2} + I_{zz,2} + a_3^2 \sum_{j=3}^{6} m_j,\ \pi_{a,7} = m_2 r_{x,2} + a_3 \sum_{j=3}^{6} m_j,\ \pi_{a,8} = m_2 r_{y,2},$$

$$\pi_{a,9} = I_{xx,3} - I_{yy,3} - a_4^2 \sum_{j=4}^{6} m_j,\ \pi_{a,10} = I_{xy,3},$$

$$\pi_{a,11} = I_{xz,3} - a_4 m_4 r_{z,4} - a_4 d_4 \sum_{j=4}^{6} m_j,\ \pi_{a,12} = I_{yz,3},$$

$$\pi_{a,13} = I_{zz,3} + a_4^2 \sum_{j=4}^{6} m_j,\ \pi_{a,14} = m_3 r_{x,3} + a_4 \sum_{j=4}^{6} m_j,\ \pi_{a,15} = m_3 r_{y,3},$$

$$\pi_{a,16} = I_{xx,4} - I_{yy,4} + I_{yy,5} + 2d_5 m_5 r_{z,5} + d_5^2 \sum_{j=5}^{6} m_j,\ \pi_{a,17} = I_{xy,4},$$

$$\pi_{a,18} = I_{xz,4},\ \pi_{a,19} = I_{yz,4},\ \pi_{a,20} = I_{zz,4} + I_{yy,5} + 2d_5 m_5 r_{z,5} + d_5^2 \sum_{j=5}^{6} m_j,$$

$$\pi_{a,21} = m_4 r_{x,4},\ \pi_{a,22} = m_4 r_{y,4} + m_5 r_{z,5} + d_5 \sum_{j=5}^{6} m_j,$$

$$\pi_{a,23} = I_{xx,5} - I_{yy,5} + I_{yy,6} + 2d_6 m_6 r_{z,6} + d_6^2 m_6,\ \pi_{a,24} = I_{xy,5},$$

$$\pi_{a,25} = I_{xz,5},\ \pi_{a,26} = I_{yz,5},\ \pi_{a,27} = I_{zz,5} + I_{yy,6} + 2d_6 m_6 r_{z,6} + d_6^2 m_6,$$

$$\pi_{a,28} = m_5 r_{x,5},\ \pi_{a,29} = m_5 r_{y,5} - m_6 r_{z,6} - d_6 m_6,\ \pi_{a,30} = I_{xx,6} - I_{yy,6},$$

$$\pi_{a,31} = I_{xy,6},\ \pi_{a,32} = I_{xz,6},\ \pi_{a,33} = I_{yz,6},\ \pi_{a,34} = I_{zz,6},\ \pi_{a,35} = m_6 r_{x,6},$$

$$\pi_{a,36} = m_6 r_{y,6},\ \pi_{a,37} = I_{a,3},\ \pi_{a,38} = I_{a,4},\ \pi_{a,39} = I_{a,5},\ \pi_{a,40} = I_{a,6},$$

$$\pi_{a,41} = F_{c,1},\ \pi_{a,42} = F_{c,2},\ \pi_{a,43} = F_{c,3},\ \pi_{a,44} = F_{c,4},\ \pi_{a,45} = F_{c,5},$$

$$\pi_{a,46} = F_{c,6},\ \pi_{a,47} = F_{v,1},\ \pi_{a,48} = F_{v,2},\ \pi_{a,49} = F_{v,3},\ \pi_{a,50} = F_{v,4},$$

$$\pi_{a,51} = F_{v,5},\ \pi_{a,52} = F_{v,6},\ \pi_{a,53} = B_1,\ \pi_{a,54} = B_2,\ \pi_{a,55} = B_3,$$

$$\pi_{a,56} = B_4,\ \pi_{a,57} = B_5,\ \pi_{a,58} = B_6.$$

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X

Assuming that the payload is rigidly connected to robot flange, the payload parameters are then regarded as a part of the inertia parameters of the link n. Therefore, the inertial parameters of link n with considering the payload can be expressed as follows:

$$m'_n = m_n + m_L,\ m_n\boldsymbol{r}'_n = m_n\boldsymbol{r}_n + m_L\boldsymbol{r}_L,\ \boldsymbol{l}'_n = \boldsymbol{l}_n + \boldsymbol{l}_L,$$

where $\boldsymbol{l}_n, m_n\boldsymbol{r}_n$, and m_n are the inertial parameters of link n without payload. $\boldsymbol{l}_L, m_L\boldsymbol{r}_L$, and m_L are the payload parameters. $\boldsymbol{l}'_n, m_n\boldsymbol{r}'_n$, and m'_n are the inertial parameters of link n with payload. These parameters are all expressed relative to the coordinate $x_n y_n z_n$.

Therefore, when the payload is fixed at the end of the UR10 robot, m_6 can be substituted by $m'_6 = m_6 + m_L$, $m_6 r_{x,6}$ can be substituted by $m_6 r'_{x,6} = m_6 r_{x,6} + m_L r_{x,L}$, $m_6 r_{y,6}$ can be substituted by $m_6 r'_{y,6} = m_6 r_{y,6} + m_L r_{y,L}$, $m_6 r_{z,6}$ can be substituted by $m_6 r'_{z,6} = m_6 r_{z,6} + m_L r_{z,L}$, $I_{xx,6}$ can be substituted by $I'_{xx,6} = I_{xx,6} + I_{xx,L}$, $I_{xy,6}$ can be substituted by $I'_{xy,6} = I_{xy,6} + I_{xy,L}$, $I_{xz,6}$ can be substituted by $I'_{xz,6} = I_{xz,6} + I_{xz,L}$, $I_{yy,6}$ can be substituted by $I'_{yy,6} = I_{yy,6} + I_{yy,L}$, $I_{yz,6}$ can be substituted by $I'_{yz,6} = I_{yz,6} + I_{yz,L}$, and $I_{zz,6}$ can be substituted by $I'_{zz,6} = I_{zz,6} + I_{zz,L}$. Moreover, the linear friction variations caused by the payload need to be also considered. Then, the symbolic expressions of $\boldsymbol{\pi}_b$ of UR10 robot with payload can be obtained. In order to save space, the equations of $\boldsymbol{\pi}_b$ are omitted. According to the symbolic expressions of both $\boldsymbol{\pi}_a$ and $\boldsymbol{\pi}_b$, the difference between them can be easily computed by (6). Then, the nonzero elements $\boldsymbol{\varepsilon}_{nz}$ in $\boldsymbol{\varepsilon}$ can be extracted as follows:

$$\varepsilon_{nz,1} = (d_4^2 + a_4^2 + a_3^2)m_L,\ \varepsilon_{nz,2} = -a_3^2 m_L,\ \varepsilon_{nz,3} = -a_3 d_4 m_L,\ \varepsilon_{nz,4} = a_3^2 m_L,$$

$$\varepsilon_{nz,5} = a_3 m_L,\ \varepsilon_{nz,6} = -a_4^2 m_L,\ \varepsilon_{nz,7} = -a_4 d_4 m_L,\ \varepsilon_{nz,8} = a_4^2 m_L,$$

$$\varepsilon_{nz,9} = a_4 m_L,\ \varepsilon_{nz,10} = d_5^2 m_L,\ \varepsilon_{nz,11} = d_5^2 m_L,\ \varepsilon_{nz,12} = d_5 m_L,$$

$$\varepsilon_{nz,13} = I_{yy,L} + 2d_6 m_L r_{z,L} + d_6^2 m_L,\ \varepsilon_{nz,14} = I_{yy,L} + 2d_6 m_L r_{z,L} + d_6^2 m_L,$$

$$\varepsilon_{nz,15} = -m_L r_{z,L} - d_6 m_L,\ \varepsilon_{nz,16} = I_{xx,L} - I_{yy,L},\ \varepsilon_{nz,17} = I_{xy,L},$$

$$\varepsilon_{nz,18} = I_{xz,L},\ \varepsilon_{nz,19} = I_{yz,L},\ \varepsilon_{nz,20} = I_{zz,L},\ \varepsilon_{nz,21} = m_L r_{x,L},\ \varepsilon_{nz,22} = m_L r_{y,L},$$

$$\varepsilon_{nz,23} = \Delta F_{c,1},\ \varepsilon_{nz,24} = \Delta F_{c,2},\ \varepsilon_{nz,25} = \Delta F_{c,3},\ \varepsilon_{nz,26} = \Delta F_{c,4},\ \varepsilon_{nz,27} = \Delta F_{c,5},$$

$$\varepsilon_{nz,28} = \Delta F_{c,6},\ \varepsilon_{nz,29} = \Delta F_{v,1},\ \varepsilon_{nz,30} = \Delta F_{v,2},\ \varepsilon_{nz,31} = \Delta F_{v,3},\ \varepsilon_{nz,32} = \Delta F_{v,4},$$

$$\varepsilon_{nz,33} = \Delta F_{v,5},\ \varepsilon_{nz,34} = \Delta F_{v,6},\ \varepsilon_{nz,35} = \Delta B_1,\ \varepsilon_{nz,36} = \Delta B_2,\ \varepsilon_{nz,37} = \Delta B_3,$$

$$\varepsilon_{nz,38} = \Delta B_4,\ \varepsilon_{nz,39} = \Delta B_5,\ \varepsilon_{nz,40} = \Delta B_6.$$

In above, a_j and d_j are the MDH parameters in Table III. From the above derivation, it can be seen that $\boldsymbol{\varepsilon}_{nz}$ is made up of $\varepsilon_1, \varepsilon_2, \varepsilon_4, \varepsilon_6, \varepsilon_7, \varepsilon_9, \varepsilon_{11}, \varepsilon_{13}, \varepsilon_{14}, \varepsilon_{16}, \varepsilon_{20}, \varepsilon_{22}, \varepsilon_{23}, \varepsilon_{27}, \varepsilon_{29} \sim \varepsilon_{36}$, and $\varepsilon_{41} \sim \varepsilon_{58}$.

Cite this article: T. Xu, H. Tuo, Q. Fang, J. Chen, J. Fan, D. Shan and J. Zhao (2024). "An online payload identification method based on parameter difference for industrial robots", Robotica **42**, 2690–2712. https://doi.org/10.1017/S026357472400105X

Downloaded from https://www.cambridge.org/core. IP address: 3.137.169.35, on 25 Jan 2025 at 21:47:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S026357472400105X