SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2019, Vol. 62 202203:1–202203:15
https://doi.org/10.1007/s11432-018-9726-3

Parameter estimation survey for multi-joint robot dynamic calibration case study

Shaolin ZHANG[1,2], Shuo WANG[1,2,3*], Fengshui JING[1,2] & Min TAN[1,2]

[1] *The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2] *University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[3] *CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai* 200031*, China*

Received 29 September 2018/Revised 25 October 2018/Accepted 12 December 2018/Published online 21 August 2019

Abstract Accurate model parameters are the basis of robot dynamics. Many linear and nonlinear models have been proposed to calibrate the inertial parameters and friction parameters of multi-joint robots. However, methods of choosing a model and calculating its parameters still have few summaries. This paper reviews typical linear/nonlinear models and different calculation methods for robot dynamic calibration. Through simulations, the features of different methods are analyzed, including torque error, parameter error, model adaptability, solution time, and anti-interference ability of the calibration results. Finally, an experiment performed on a six-degree-of-freedom industrial manipulator is used as an example to illustrate how to select the model for a specified robot. These comparisons and experiments provide references for the parameter calibration of multi-joint robots.

Keywords dynamic parameter calibration, friction calibration, robot dynamics, industrial manipulator, dynamic models

Citation Zhang S L, Wang S, Jing F S, et al. Parameter estimation survey for multi-joint robot dynamic calibration case study. Sci China Inf Sci, 2019, 62(10): 202203, https://doi.org/10.1007/s11432-018-9726-3

1 Introduction

Model based force control is widely used in multi-joint robots. Accurate force control requires accurate parameters, including the inertial parameters and friction parameters of each joint. However, these parameters are often not provided by the robot manufacturers and some may change over time. Therefore, calibration is required to be carried out before application.

There have been many studies on the inertial parameter calibration methods of multi-joint robots. One method is to calibrate the parameters of each joint separately [1]. However, the parameter error will accumulate from previously calibrated joints. Another calibration method is based on the inverse dynamic identification model (IDIM) and least-squares method [2]. In this method, the angular position, angular velocity and angular acceleration $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ of each joint are required. However, most multi-joint robots measure only the angular position. The angular velocity and angular acceleration are obtained by differentiation and filtering, which requires appropriate tuning. Ref. [3] proposed to integrate the identification model before calibration, however additional calculations are introduced. Another method (output error method, OE) [4] is to minimize the quadratic error between the actual position and the simulated position. The simulated system takes the same input as the actual system. However, this

[*] Corresponding author (email: shuo.wang@ia.ac.cn)

method is sensitive to initial values and takes dozens of iterations to converge for a single degree-of-freedom robot. The DIDIM method [5] and IDIM-IV method [6] are improvements of the OE method. The DIDIM method chooses to minimize the joint force/torque error between the actual system and simulated system. The IDIM-IV method utilizes the simulated data to build an instrument variable, which is thought to be noise-free and correlated with the observation matrix. The iterative calculation is simplified. The two methods both converge in approximately three iterations and show robustness to the initialization of parameters. In these methods, the control law is supposed to be known. For industrial robots whose control system is unknown, the parameters can be searched by optimization method [7], and constraints can be added [8, 9]. In addition, there are also online parameter calibration methods. The Kalman [10] and Extended Kalman [11] methods are suitable for linear and nonlinear identification models, respectively. However, the points sampled earlier cannot reflect the current parameter conditions when following changing parameters. The exponential forgetting method [3] introduces an exponential forgetting index to solve this problem.

There have also been many studies on the calibration methods of friction parameters. The joint friction consists of Coulomb friction, viscous friction, and static friction [12]. The models of Coulomb friction and viscous friction are consistent in many studies [13, 14]. However, static friction is difficult to estimate and has different models. It is nonlinear and exists at low velocities. A commonly used static friction model is the LuGre model [15, 16]. Some other models [17, 18] can be used to approximate the effects of static friction. However, the nonlinear models are difficult to solve and have a long solution time. Therefore, some linear models [19, 20] have been proposed to approximate the friction curve. The parameters in approximation models may not have physical meanings; they are simply mathematical parameters.

In this paper, we analyze the parameter calibration of multi-joint industrial robots. For these robots, the driving torque is measured by the joint torque sensor [21] or motor current [22, 23]. The measured torque is the sum of the inertial force and friction. Therefore, the inertial parameters and friction parameters are calibrated together in one step [24, 25]. The calculation method has been studied by many researchers. However, the features of these methods have not been compared and summarized in detail. Some calculation methods [26] estimate parameters within feasible ranges, but some others [2] only ensure that the joint torques match with the measured ones. Some calculation methods [27] take a lot of time for solution and are only suitable for offline calculation, while some others [3] can meet the requirements of online calibration. Some calculation methods [6] are only suitable for linear models, while some others [28] can be used for nonlinear models and fit better with static friction. To have an overview of the calibration of inertial parameters and friction parameters, this paper introduces several linear and nonlinear calculation methods for different models. The features are compared through MATLAB simulations. Finally, an experiment on a six-degree-of-freedom industrial robot is performed as an example to illustrate how to select the calculation method and model.

The contributions of the paper are as follows.

(1) This paper compares commonly used inertial parameter and friction parameter calibration models, proposes quantitative evaluation indices, and analyzes their advantages and disadvantages.

(2) Through simulations and experiments, the features of different calculation methods are compared and divided into three levels. The summary of features and an instance of a six-degree-of-freedom industrial robot provide references for the parameter calibration of multi-joint robots.

The rest of this paper is organized as follows. Section 2 introduces typical linear and nonlinear models. Section 3 presents the parameter calculation methods. Section 4 discusses the optimal trajectory. Section 5 describes the features and quantization methods. Section 6 gives the results of simulations and experiments, and finally, Section 7 summarizes the conclusion.

2 Calibration models

This section introduces calibration models for the multi-joint industrial robot. The Coulomb friction and viscous friction models, nonlinear models, and approximation models are introduced. Finally, the inertial force model and friction model are combined for calculation.

The forces applied on the robot can be described as [29]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{V}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_f = \boldsymbol{\tau}_j, \tag{1}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are the angular position, angular velocity and angular acceleration, $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the centrifugal and Coriolis forces, $\boldsymbol{G}(\boldsymbol{q})$ is the gravity vector, $\boldsymbol{\tau}_f$ is the friction torque, and $\boldsymbol{\tau}_j$ is the joint driving torque.

Define the set of inertial parameters to be calibrated as $\boldsymbol{\phi}_d$. Then Eq. (1) can be rewritten as [2]

$$\boldsymbol{K}_d \boldsymbol{\phi}_d + \boldsymbol{\tau}_f = \boldsymbol{\tau}_j, \tag{2}$$

where $\boldsymbol{K}_d$ is only related to the kinematic parameters and $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$, which are supposed to be known.

If only the Coulomb friction and viscous friction are taken into consideration, $\boldsymbol{\tau}_f$ is

$$\boldsymbol{\tau}_f = \boldsymbol{F}_c \mathrm{sgn}(\dot{\boldsymbol{q}}) + \boldsymbol{\beta}\dot{\boldsymbol{q}}, \tag{3}$$

where $\boldsymbol{F}_c$ is the Coulomb friction coefficient and $\boldsymbol{\beta}$ is the viscous friction coefficient.

When the joint angular velocity $\dot{\boldsymbol{q}}$ is smaller than the threshold $\dot{\boldsymbol{q}}_s$, static friction cannot be ignored. In the Lugre model, $\boldsymbol{\tau}_f$ is described as [12]

$$\boldsymbol{\tau}_f = \left[\boldsymbol{F}_c + (\boldsymbol{F}_s - \boldsymbol{F}_c)\mathrm{e}^{-\left|\frac{\dot{\boldsymbol{q}}}{\dot{\boldsymbol{q}}_s}\right|^{\delta_s}}\right] \mathrm{sgn}(\dot{\boldsymbol{q}}) + \boldsymbol{\beta}\dot{\boldsymbol{q}}, \tag{4}$$

where $\boldsymbol{F}_c$ is the Coulomb friction coefficient, $\boldsymbol{F}_s$ is the static friction coefficient, $\dot{\boldsymbol{q}}_s$ is the angular velocity threshold, δ_s can be set to 2, and $\boldsymbol{\beta}$ is the viscous friction coefficient.

There are also studies using different models to calculate the friction $\boldsymbol{\tau}_f$ [17]:

$$\boldsymbol{\tau}_f = \boldsymbol{f}_1 \mathrm{sgn}(\dot{\boldsymbol{q}}) - \boldsymbol{f}_2 \mathrm{sgn}(\dot{\boldsymbol{q}})\mathrm{e}^{-\frac{|\dot{\boldsymbol{q}}|}{\boldsymbol{f}_3}} - \boldsymbol{f}_4 \mathrm{sgn}(\dot{\boldsymbol{q}})\mathrm{e}^{-\frac{1}{\boldsymbol{f}_5|\dot{\boldsymbol{q}}|}} + \beta\dot{\boldsymbol{q}}, \tag{5}$$

where $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4, \boldsymbol{f}_5, \boldsymbol{\beta}$ are the parameters to be calibrated.

Ref. [18] proposed another model:

$$\boldsymbol{\tau}_f = (\boldsymbol{f}_1 + \boldsymbol{f}_2 \mathrm{sech}(\boldsymbol{f}_3\dot{\boldsymbol{q}}))\tanh(\boldsymbol{f}_4\dot{\boldsymbol{q}}) + \beta\dot{\boldsymbol{q}}, \tag{6}$$

where $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4, \boldsymbol{\beta}$ are the parameters to be calibrated.

Eq. (3) can be written in a linear form:

$$\boldsymbol{\tau}_f = \begin{bmatrix} \mathrm{sgn}(\dot{\boldsymbol{q}}) & \dot{\boldsymbol{q}} \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_c \\ \boldsymbol{\beta} \end{bmatrix}. \tag{7}$$

However, Eqs. (4)–(6) cannot be written as a linear form of $\dot{\boldsymbol{q}}$. Therefore, some studies use linear approximations to fit the friction curve. Ref. [19] proposed a method as

$$\boldsymbol{\tau}_f = \boldsymbol{F}_c \mathrm{sgn}(\dot{\boldsymbol{q}}) + \beta\dot{\boldsymbol{q}} + \boldsymbol{f}_1 \dot{\boldsymbol{q}}^{1/3}. \tag{8}$$

Ref. [30] proposed another approximation model:

$$\boldsymbol{\tau}_f = \boldsymbol{F}_c \mathrm{sgn}(\dot{\boldsymbol{q}}) + \beta\dot{\boldsymbol{q}} + \boldsymbol{f}_1 \tau_{\mathrm{load}}, \tag{9}$$

where τ_{load} is the torque caused by the load.

The linear expressions of (7)–(9) can be summarized as

$$\boldsymbol{\tau}_f = \boldsymbol{K}_f \boldsymbol{\phi}_f, \tag{10}$$

where $\boldsymbol{K}_f$ is a known matrix related to the kinematic parameters and $\boldsymbol{\phi}_f$ contains all the friction parameters.

Eqs. (4)–(6) are non-linear expressions and can be summarized as

$$\boldsymbol{\tau}_f = f(\boldsymbol{\phi}_f). \tag{11}$$

Therefore, the parameter calibration problem is to calculate $\boldsymbol{\phi}_d$ and $\boldsymbol{\phi}_f$ according to (2) and (10) or (11).

$$\boldsymbol{\tau}_j = \begin{bmatrix} \boldsymbol{K}_d \ \boldsymbol{K}_f \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_d \\ \boldsymbol{\phi}_f \end{bmatrix}, \tag{12}$$

$$\boldsymbol{\tau}_j = \boldsymbol{K}_d\boldsymbol{\phi}_d + f(\boldsymbol{\phi}_f). \tag{13}$$

Eq. (12) is a general linear model; Eq. (13) is a general nonlinear model. The two equations can be written as

$$\boldsymbol{\tau}_j = \boldsymbol{K}\boldsymbol{\phi}, \tag{14}$$

where $\boldsymbol{K} = [\, \boldsymbol{K}_d \ \boldsymbol{K}_f \,]$ for linear models and $\boldsymbol{K} = [\, \boldsymbol{K}_d \ \partial \boldsymbol{f}/\partial \boldsymbol{\phi}_f \,]$ for nonlinear models.

3 Parameter calculation methods

This section describes the calculation methods for (12) and (13). The typical calculation method for the linear model is the pseudo-inverse method [31], and that for the nonlinear model is the optimization method. Considering the online estimation method, the Kalman filter method [10,11] and other methods [3] will also be introduced. Some methods [32] are variants of these methods and are ignored. Only typical methods are introduced in the paper.

3.1 Pseudo-inverse method

Eq. (12) can be solved by the pseudo-inverse method:

$$\hat{\boldsymbol{\phi}} = \boldsymbol{K}^{+}\boldsymbol{\tau}_j = (\boldsymbol{K}^{\mathrm{T}}\boldsymbol{K})^{-1}\boldsymbol{K}^{\mathrm{T}}\boldsymbol{\tau}_j, \tag{15}$$

where $\hat{\boldsymbol{\phi}}$ is the estimate of $\boldsymbol{\phi}$.

This method is actually a least-squares estimation method. Its objective function is

$$\boldsymbol{J} = \left\| \boldsymbol{\tau}_j - \boldsymbol{K}\hat{\boldsymbol{\phi}} \right\|^2. \tag{16}$$

The pseudo-inverse method calculates all parameters quickly but requires sampling all of the points and calculating them together. It is only suitable for offline calibration.

Remark 1. In this method, $\dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are obtained by the differentiation of $\boldsymbol{q}$, which introduces noise and delay. Filtering and phase adjustment are required to reduce the bias in the result. However, the filters need to be carefully tuned. An alternative is to use the IDIM-IV method [6,33]. An instrumental variable which is not disturbed by the noise is built for calibration:

$$\hat{\boldsymbol{\phi}} = (\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{K})^{-1}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{\tau}_j, \tag{17}$$

where $\boldsymbol{Z} = \boldsymbol{K}(\boldsymbol{q}_{\mathrm{sp}}, \dot{\boldsymbol{q}}_{\mathrm{sp}}, \ddot{\boldsymbol{q}}_{\mathrm{sp}})$ is the instrumental variable calculated with simulated parameters $\boldsymbol{q}_{\mathrm{sp}}, \dot{\boldsymbol{q}}_{\mathrm{sp}}, \ddot{\boldsymbol{q}}_{\mathrm{sp}}$, which are observed by simulation of the control law and direct dynamic model of the robot. The simulated system takes the same reference trajectories and control structure as the actual robot system. The torque is filtered and resampled with a parallel decimation procedure to reduce ripples. Otherwise, the IDIM-IV method will lose its efficiency.

3.2 Kalman filter/extended Kalman filter

Considering the noise in the model and measurement, Eq. (12) can be rewritten as follows:

$$\boldsymbol{\tau}_j = \boldsymbol{K}\boldsymbol{\phi} + \boldsymbol{\omega}, \tag{18}$$

where $\boldsymbol{\omega} \sim \boldsymbol{N}(0, \boldsymbol{R})$ is the noise. To solve this equation using a Kalman filter, the process is as follows:

$$\begin{aligned}
&\hat{\boldsymbol{\phi}}_n^- = \hat{\boldsymbol{\phi}}_{n-1},\\
&\boldsymbol{P}_n^- = \boldsymbol{P}_{n-1},\\
&\boldsymbol{G}_n = \boldsymbol{P}_n^- \boldsymbol{K}^{\mathrm{T}} (\boldsymbol{K}\boldsymbol{P}_n^- \boldsymbol{K}^{\mathrm{T}} + \boldsymbol{R})^{-1},\\
&\hat{\boldsymbol{\phi}}_n = \hat{\boldsymbol{\phi}}_n^- + \boldsymbol{G}_n(\boldsymbol{\tau}_j - \boldsymbol{K}\hat{\boldsymbol{\phi}}_n^-),\\
&\boldsymbol{P}_n = [\boldsymbol{I} - \boldsymbol{G}_n\boldsymbol{K}]\, \boldsymbol{P}_n^-,
\end{aligned} \tag{19}$$

where the subscripts n and $n-1$ represent the nth and $(n-1)$th iterations, respectively, and $\boldsymbol{P}, \boldsymbol{G}$ are intermediate variables.

Compared with the pseudo-inverse method, this method calculates the current parameters from the calibration result of the previous step and does not require pre-sampling, which is suitable for online calibration.

For nonlinear models, the extended Kalman filter can be used. First, Eq. (13) with noise can be rewritten as

$$\boldsymbol{\tau}_j = \boldsymbol{K}_d\boldsymbol{\phi}_d + f(\boldsymbol{\phi}_f) + \boldsymbol{\omega}. \tag{20}$$

The solution to this equation is as follows:

$$\begin{aligned}
&(\hat{\boldsymbol{\phi}}_d)_n^- = (\hat{\boldsymbol{\phi}}_d)_{n-1},\\
&(\hat{\boldsymbol{\phi}}_f)_n^- = (\hat{\boldsymbol{\phi}}_f)_{n-1},\\
&\boldsymbol{P}_n^- = \boldsymbol{P}_{n-1},\\
&\boldsymbol{G}_n = \boldsymbol{P}_n^- \boldsymbol{K}^{\mathrm{T}} (\boldsymbol{K}\boldsymbol{P}_n^- \boldsymbol{K}^{\mathrm{T}} + \boldsymbol{R})^{-1},\\
&\hat{\boldsymbol{\phi}}_n = \hat{\boldsymbol{\phi}}_n^- + \boldsymbol{G}_n(\boldsymbol{\tau}_j - \boldsymbol{K}_d(\hat{\boldsymbol{\phi}}_d)_n^- - f((\hat{\boldsymbol{\phi}}_f)_n^-)),\\
&\boldsymbol{P}_k = [\boldsymbol{I} - \boldsymbol{G}_n\boldsymbol{K}]\, \boldsymbol{P}_k^-.
\end{aligned} \tag{21}$$

The Kalman filter and the extended Kalman filter have few calculations and are suitable for online calibration.

3.3 Exponential forgetting method

In general, the inertial parameters of robot joints are fixed, but the friction parameters will change with the temperature and working time. The points sampled earlier cannot reflect the current parameter conditions. As such, it is necessary to weaken the influence of the previous data. The exponential forgetting method is commonly used to achieve this goal.

Ref. [3] defines the optimization objective as

$$\boldsymbol{J} = \sum_{i=1}^{N} \mu^{N-i} (\boldsymbol{\tau}_j - \boldsymbol{K}\boldsymbol{\phi})^2, \tag{22}$$

where N is the number of samples and $\mu \in [0, 1]$ is a real number representing forgetting speed. A larger μ leads to a slower forgetting speed, and the effect of the previous data will exist longer, and vice versa.

To find the $\boldsymbol{\phi}$ which minimizes J, the solution process is as follows:

$$\begin{aligned}
&\hat{\boldsymbol{\phi}}_n^- = \hat{\boldsymbol{\phi}}_{n-1},\\
&\boldsymbol{P}_n^- = \boldsymbol{P}_{n-1},\\
&\boldsymbol{G}_n = \boldsymbol{P}_n^- \boldsymbol{K}^{\mathrm{T}} (\boldsymbol{K}\boldsymbol{P}_n^- \boldsymbol{K}^{\mathrm{T}} + \mu\boldsymbol{I})^{-1},\\
&\hat{\boldsymbol{\phi}}_n = \hat{\boldsymbol{\phi}}_n^- + \boldsymbol{G}_n(\boldsymbol{\tau}_j - \boldsymbol{K}\hat{\boldsymbol{\phi}}_n^-),\\
&\boldsymbol{P}_n = \frac{1}{\mu} [\boldsymbol{I} - \boldsymbol{G}_n\boldsymbol{K}]\, \boldsymbol{P}_n^-.
\end{aligned} \tag{23}$$

Similarly, the exponential forgetting method for a nonlinear model is as follows:

$$\begin{aligned}
&(\hat{\boldsymbol{\phi}}_d)_n^- = (\hat{\boldsymbol{\phi}}_d)_{n-1},\\
&(\hat{\boldsymbol{\phi}}_f)_n^- = (\hat{\boldsymbol{\phi}}_f)_{n-1},\\
&\boldsymbol{P}_n^- = \boldsymbol{P}_{n-1},\\
&\boldsymbol{G}_n = \boldsymbol{P}_n^- \boldsymbol{K}^{\mathrm{T}} (\boldsymbol{K}\boldsymbol{P}_n^- \boldsymbol{K}^{\mathrm{T}} + \mu \boldsymbol{I})^{-1},\\
&\hat{\boldsymbol{\phi}}_n = \hat{\boldsymbol{\phi}}_n^- + \boldsymbol{G}_n(\boldsymbol{\tau}_j - \boldsymbol{K}_d(\hat{\boldsymbol{\phi}}_d)_n^- - f((\hat{\boldsymbol{\phi}}_f)_n^-)),\\
&\boldsymbol{P}_k = \frac{1}{\mu}\left[\boldsymbol{I} - \boldsymbol{G}_n\boldsymbol{K}\right]\boldsymbol{P}_k^-.
\end{aligned} \tag{24}$$

The exponential forgetting method is similar to the Kalman filter. The calculation is fast and suitable for online calibration. However, the parameter μ needs to be carefully set. A too large or too small μ will cause observation errors.

3.4 Optimization method

To solve (12) and (13) with the optimization method, an objective function needs to be set first. If the objective function is defined as (16), the optimization method is similar to the least-squares method. If the objective function is defined as (22), the optimization method is similar to the exponential forgetting method. A choice is to set the objective function as

$$J = \sum_{i=1}^{n_j} |\bar{e}_i| + k_e \sum_{i=1}^{n_j} \sigma(e_i), \quad \boldsymbol{e} = \boldsymbol{\tau}_j - \boldsymbol{K}\hat{\boldsymbol{\phi}}, \tag{25}$$

where $\bar{e}_i$ is the mean of e_i, k_e is a positive real number, and n_j is the number of joints.

Unlike the other methods, the optimization method can solve the problem within an appropriate parameter range. The initial inertial parameter can be obtained from the three-dimensional model of the robot. The parameter range could be set around the initial parameter. The inertia matrix can be constrained to be positive definite. Other limits can also be added [26, 34]. With these constraints, $\hat{\boldsymbol{\phi}}$ is obtained which minimizes the objective function J. The calibration result is more physically feasible.

To solve the optimization problem, the gradient descent method [35], genetic algorithm [36] and particle filter algorithm [37] can be used.

(1) Gradient descent method. The gradient descent method decreases the objective function by changing the parameters along the reverse gradient direction. However, this method needs initial values of the calibrated parameters, which should be set carefully. Owing to the strong nonlinearity of the objective function, this method may obtain a locally optimal solution.

(2) Genetic algorithm. The genetic algorithm simulates the evolution process by operators such as mutation, crossover, and selection and searches for the optimal solution that satisfies the constraints. This method does not need to specify the initial parameters, and only search ranges are required. However, this method may also obtain a locally optimal solution.

(3) Particle filter algorithm. The particle filter algorithm estimates the sample distribution by resampling and then performs a global search. Additionally, this method does not need to estimate the initial parameters, but instead must collect a large number of samples to approximate the actual distribution probability, especially for high-dimensional systems. The amount of calculation is large.

4 Optimal trajectory

In order to reduce the measurement noise, multiple points can be sampled on a trajectory to form multiple equation groups. Solving these equation groups together can achieve better accuracy. The trajectory should sufficiently excite the effect of the dynamic parameters to make the observation matrix well-conditioned. The excitation characteristic of this trajectory is directly related to the accuracy of the

Table 1 The levels of torque error

	Level A	Level B	Level C
e_T	<0.01	<0.1	>0.1

calculation result [38]. There are several criteria [39] for generating the excitation trajectory. The most prominent is

$$J_q = \mathrm{cond}(\boldsymbol{K}), \quad \boldsymbol{q} = \arg\min(J_q). \tag{26}$$

Based on this criterion, some other criteria have been derived [40–42]. All these criteria are designed to make the dynamic parameters reflected in the joint torques so that the calibration result is more accurate.

Based on the criteria, two schemes can be used to generate the trajectory: (1) Deal with the constraint directly with a genetic algorithm [41] or other methods [43, 44]; (2) Use different curves to generate the trajectory, such as splines, polynomials, sine series, and Fourier series and optimize the parameters based on the constraint. Many applications adopt the Fourier series trajectory because it obtains a smaller condition number [26].

$$q_i = \sum_{l=1}^{L} \frac{a_{i,l}}{\omega_f l}\sin(\omega_f l t) - \frac{b_{i,l}}{\omega_f l}\cos(\omega_f l t), \tag{27}$$

where $a_{i,l}$ and $b_{i,l}$ are trajectory parameters constrained by (26), ω_f is the fundamental frequency of the Fourier series, and $i = 1, 2, \ldots, n_j$.

5 Features and quantization

To compare the features of the different models and calculation methods, the following presents the features and quantization method.

5.1 Features of calibration models

The main difference between the linear models and nonlinear models is the static friction. When the joint velocity is smaller than $\dot{\boldsymbol{q}}_s$, the static friction has a significant effect if the lubrication is poor. Eq. (8) is an approximation of the nonlinear model and has a compromising effect. The linear models and nonlinear models mainly differ in torque error.

Torque error is the difference between the measured torques and simulated torques. Define the evaluation indexes as

$$e_T = \mathrm{Mean}(|\boldsymbol{\tau}_j - \hat{\boldsymbol{\tau}}_j|/\boldsymbol{\tau}_{\max}), \quad e_T\_\mathrm{Std} = \mathrm{Std}(|\boldsymbol{\tau}_j - \hat{\boldsymbol{\tau}}_j|/\boldsymbol{\tau}_{\max}),$$

where $\hat{\boldsymbol{\tau}}_j$ is the torque calculated by (2) with the calibrated parameters, $\boldsymbol{\tau}_{\max}$ is the maximum joint driving torque, Mean() means averaging, and Std() means standard deviation.

Based on the evaluation index e_T, define three levels as Table 1.

5.2 Features of calculation methods

The calculation methods mainly differ in:

(1) Model adaptability. The pseudo-inverse method and Kalman filter apply only to the linear models. The extended Kalman filter is applicable to the nonlinear model. The optimization method can be applied to both linear models and nonlinear models.

(2) Parameter errors. The parameter errors include inertial parameter errors and friction parameter errors. If the parameters change over time, there are also parameter following errors. The evaluations of parameter errors are shown in Table 2.

$$e_{di} = \mathrm{Mean}(|(\boldsymbol{\phi}_{di} - \hat{\boldsymbol{\phi}}_{di})/\boldsymbol{\phi}_{di}|),$$

$$e_{fi} = \mathrm{Mean}(|(\boldsymbol{\phi}_{fi} - \hat{\boldsymbol{\phi}}_{fi})/\boldsymbol{\phi}_{fi}|),$$

Table 2 The levels of parameter errors

	Level A	Level B	Level C
e_d	<0.2	<1	>1
e_f	<0.2	<1	>1
J_e	<1	<10	>10
Follow changing parameter	Slope change	Step change	None

Table 3 The levels of solution time of offline methods

	Level A	Level B	Level C
Solution time	<1 min	<1 h	>1 h

Table 4 The levels of anti-interference ability

	Level A	Level B	Level C
ΔJ_e	<0.1	<1	>1

$$e_d = \mathrm{Mean}(e_{di}),$$

$$e_f = \mathrm{Mean}(e_{fi}),$$

$$J_e = e_d + e_f + e_T + e_{T}\_\mathrm{Std},$$

where e_d is the inertial parameter errors and e_f is the friction parameter errors.

(3) Solution time. The Kalman filter and exponential forgetting method are iteratively calculated once per cycle. The pseudo-inverse method and the optimization method are offline calculation methods. To obtain the result, the pseudo-inverse method calculates in one iteration and the optimization method requires multiple iterations to converge. The levels of solution time are presented in Table 3.

(4) Anti-interference ability. In order to analyze the anti-interference ability of the parameter calculation methods, a Gaussian noise is added to the measured torque $\boldsymbol{\tau}_j$, and different methods are used to calculate again. The evaluation index of anti-interference ability is defined as

$$\Delta J_e = (J_{e2} - J_{e1})/J_{e1},$$

where J_{e1} is J_e measured without noise and J_{e2} is J_e measured with noise. The levels of anti-interference ability are shown in Table 4.

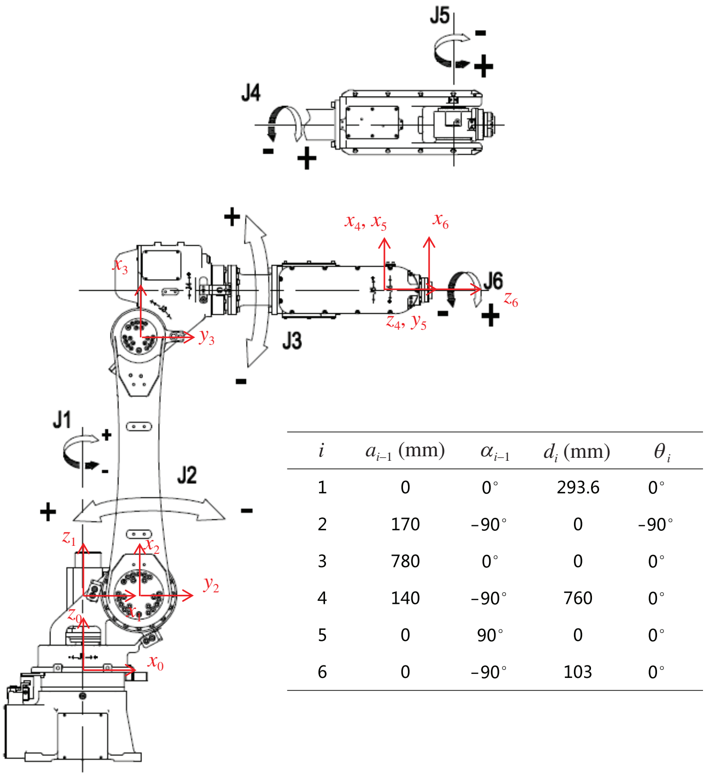
6 Simulations and experiments

6.1 Simulation platform

First, a simulation platform is built in MATLAB to compare the results of the parameter calculation methods. The platform is built based on the six-degree-of-freedom Efort robot ER20-C10 (as shown in Figure 1), having the same kinematic parameters and inertial parameters as the three-dimensional model of the robot. The friction parameters are set to suitable values based on experience.

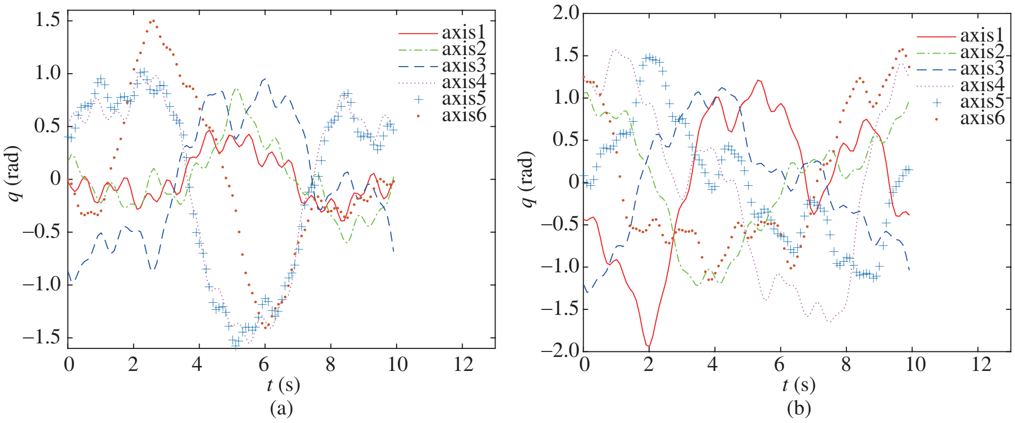
The condition number of $\boldsymbol{K}$ could then be calculated and the trajectory described in (27) could be optimized. According to [26], the parameter L is set to 5 and the fundamental frequency ω_f is set to 0.1 Hz. The optimization problem is solved with a genetic algorithm. The linear calibration method takes (7) as an example and the nonlinear calibration method takes (4) as an example. The minimum J_q of (7) and (4) are 4.6 and 5.7, respectively. The results of position trajectories of the six joints are shown in Figure 2.

The optimal trajectories $(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d)$ are simulated in the platform as drawn in Figure 3. The ER20 robot model consists of a direct dynamic model and a friction model. $\boldsymbol{\phi}_s$ is the set parameter, $\boldsymbol{\tau}_{jc}$ is the control torque, $\boldsymbol{\tau}_{jm}$ is the measured torque, and $\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m, \ddot{\boldsymbol{q}}_m$ are the measured angular position, angular velocity, and angular acceleration. The inverse dynamic and friction model also takes $\boldsymbol{\phi}_s$ for control. $\boldsymbol{\tau}_{jm}$ and $\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m, \ddot{\boldsymbol{q}}_m$ are obtained from the simulation at 100 Hz. The simulation is performed in two steps.



i	a_{i-1} (mm)	α_{i-1}	d_i (mm)	θ_i
1	0	0°	293.6	0°
2	170	-90°	0	-90°
3	780	0°	0	0°
4	140	-90°	760	0°
5	0	90°	0	0°
6	0	-90°	103	0°

Figure 1 (Color online) The Efort ER20-C10 robot and D-H frames.



(a)

(b)

Figure 2 (Color online) Position trajectories of the six joints optimized according to (26) and (27). (a) Linear model (7); (b) nonlinear model (4).

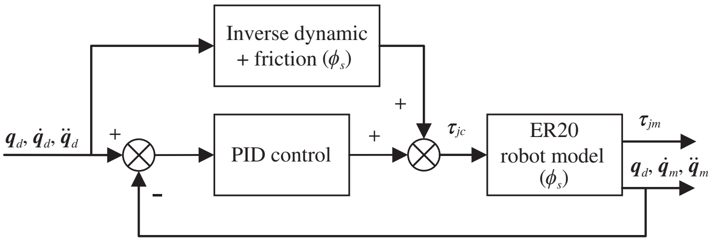

Inverse dynamic + friction (ϕ_s)
$q_d, \dot{q}_d, \ddot{q}_d$
+
−
PID control
+
+
τ_{jc}
ER20 robot model (ϕ_s)
τ_{jm}
$q_d, \dot{q}_m, \ddot{q}_m$

Figure 3 The simulation platform for point sampling.

- Ignore the noise of $\boldsymbol{\tau}_{jm}, \boldsymbol{q}_m, \dot{\boldsymbol{q}}_m, \ddot{\boldsymbol{q}}_m$ and compare the parameter errors of the different calculation methods.
- Take the noise into consideration and analyze the anti-interference ability of the different methods.

Table 5 Linear calibration result of joint 3

Parameter		Set	Pseudo-inverse	Kalman	Index forgetting	Optimization		
						Gradient descent	Genetic algorithm	Particle filter
Mass (kg)	m	25.6083	17.7803	17.7782	17.7793	24.5913	24.5699	24.5250
Center (m)	X	0.0938	−0.1811	−0.1807	−0.1809	0.0672	0.0994	0.1208
	Y	0.1383	−0.1121	−0.1140	−0.1132	0.1107	0.1657	0.1384
	Z	0.1164	0.0085	0.0082	0.0083	0.0970	0.1159	0.1148
Inertia ($\mathrm{kg \cdot m^2}$)	I_{11}	0.1706	3.3342	3.3206	3.3262	0.1287	0.1783	0.2189
	I_{12}	−0.1498	0.5166	0.5146	0.5151	−0.2175	−0.1258	−0.1829
	I_{13}	0.2140	0.6943	0.6915	0.6928	0.1790	0.1615	0.2486
	I_{22}	0.3493	−3.5211	−3.5359	−3.5298	0.4161	0.2925	0.4160
	I_{23}	−0.1320	−0.2403	−0.2808	−0.2641	−0.1994	−0.0933	−0.1646
	I_{33}	0.2363	−0.3693	−0.3691	−0.3693	0.3062	0.3046	0.3063
Friction	F_c	50.0000	50.0000	49.7802	49.8704	50.2227	50.2716	50.4040
	β	40.0000	40.0000	40.0250	40.0161	39.7539	39.6028	39.8216
Torque error (Nm)	Mean		−0.0001	−0.0066	0.0323	0.2568	0.2725	0.0029
	Std		0.0001	0.2903	0.2210	0.6332	0.8545	0.8825

Table 6 Nonlinear calibration result of joint 3

Parameter		Set	Extended Kalman	Index forgetting	Optimization		
					Gradient descent	Genetic algorithm	Particle filter
Mass (kg)	m	25.6083	18.0803	18.0862	25.5425	29.5065	29.1977
Center (m)	X	0.0938	−0.1405	−0.1401	0.075	0.0964	0.1141
	Y	0.1383	−0.0374	−0.0377	0.1107	0.1156	0.1526
	Z	0.1164	0.0179	0.0175	0.0931	0.1143	0.1158
Inertia ($\mathrm{kg \cdot m^2}$)	I_{11}	0.1706	3.9916	4.0685	0.2302	0.1447	0.2406
	I_{12}	−0.1498	0.2146	0.2089	−0.0886	−0.2168	−0.097
	I_{13}	0.214	0.5786	0.595	0.265	0.2368	0.1683
	I_{22}	0.3493	−3.1141	−3.0855	0.4044	0.2933	0.2909
	I_{23}	−0.132	−0.2575	−0.2548	−0.1994	−0.1421	−0.0809
	I_{33}	0.2363	0.5709	0.5297	0.2828	0.302	0.2335
Friction	F_c	50	51.7727	52.183	50.625	49.9266	50.2767
	F_s	60	52.1061	52.2013	60	59.8052	60.0016
	q_s	0.1	5.5823	6.3761	0.08	0.1142	0.0996
	β	40	38.2327	38.1439	39.8125	39.8125	39.7704
Torque error (Nm)	Mean		0.0355	0.0314	2.8462	−0.0209	0.0062
	Std		1.9688	1.9705	1.2125	0.6137	0.2182

6.2 Parameter errors

The calculated parameters are compared to the set parameters. The linear calibration result is present in Table 5 and the nonlinear calibration result is present in Table 6. For simplicity, only the parameters of joint 3 are presented.

As can be seen from Tables 5 and 6, the pseudo-inverse method and the Kalman filter can ensure that the torque error and friction parameter errors are small, but the inertial parameters have large deviations from the set parameters and are partially unreasonable. Because the optimization methods constrain the parameter ranges, the inertial parameters are relatively reasonable and the joint torques are also consistent with the actual torques. In this simulation, the initial parameters of the gradient method are set to ±20% around the set parameters.

The exponential forgetting method is an extension of the Kalman filter and the result is similar. The advantage of the exponential forgetting method is that it can adjust the forgetting factor and follow

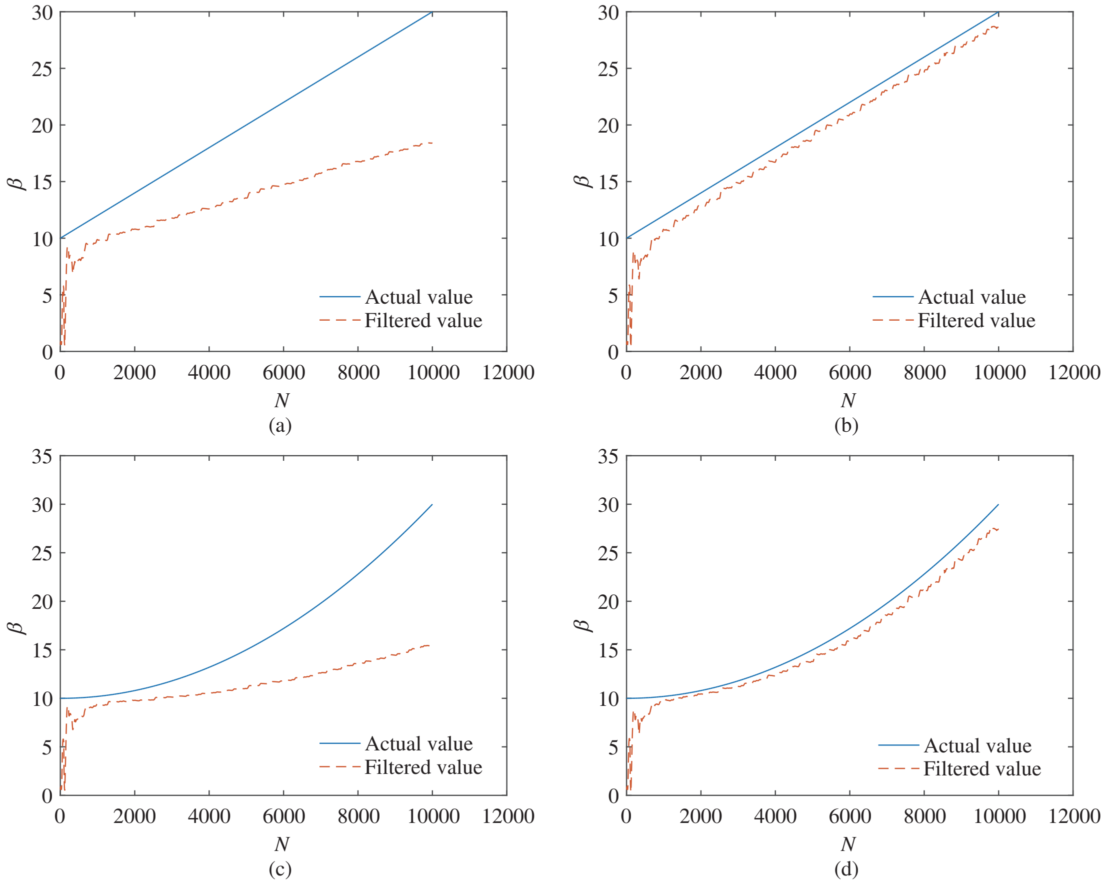


Figure 4 (Color online) A comparison of the Kalman filter and exponential forgetting method. (a) Kalman filter (slope change); (b) exponential forgetting (slope change); (c) Kalman filter (second-order change); (d) exponential forgetting (second-order change).

closer to changing parameters. In Figure 4, the Kalman filter and exponential forgetting method are used to track the coefficient of viscous friction ($\boldsymbol{\beta}$). $\boldsymbol{\beta}$ is changing with the number of iterations (N). In Figure 4(a) and (b), $\boldsymbol{\beta}$ is a ramp function; in Figure 4(c) and (d), $\boldsymbol{\beta}$ is a second-order function. It can be seen that the exponential forgetting method tracks closer to the actual value than the Kalman filter.

6.3 Solution time

The offline calculation time is tested on a computer with an Intel i5-4590 CPU and 8 GB of memory. The pseudo-inverse method gets results within one second. The gradient descent method and particle filter algorithm take approximately 30 min and the genetic algorithm takes more than an hour.

6.4 Anti-interference ability

The torque noise added at each joint is shown in Table 7. The standard deviation of noise is set to the same magnitude as the measurement noise of the robot. The comparisons of e_d, e_f, e_T, J_e, e_T_Std with and without noise are shown in Tables 8 and 9. Under the influence of noise, the errors (e_d, e_f, e_T, J_e) become larger but not significant. Each method shows a certain degree of anti-interference ability. e_T_Std increases obviously, which is related to the standard deviation of the noise. In the optimization method, the gradient descent method is sensitive to the initial parameters. Different initial parameters may lead to different results. The genetic algorithm and particle filter algorithm do not need initial values and have similar results.

Taking the noise of $\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m$, and $\ddot{\boldsymbol{q}}_m$ into consideration, it further affects the stability of the calibration results. The IDIM-IV method [6] is proposed for this problem. To illustrate the anti-interference ability of

Table 7 The standard deviation of noise added to $\boldsymbol{\tau}_j$

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Std (Nm)	10	10	10	1.78	1.78	1.78

Table 8 The comparison of J_e with and without noise (linear methods)

		Pseudo-inverse	Kalman	Exponential forgetting	Optimization		
					Gradient descent	Genetic algorithm	Particle filter
Without noise	e_d	279.8642	256.6049	255.1294	0.3839	0.8657	0.6336
	e_f	0.0000	0.0135	0.0080	0.0032	0.0061	0.0185
	e_T	0.0000	0.0000	0.0001	0.0001	0.0006	0.0001
	e_T_Std	0.0000	0.0007	0.0005	0.0006	0.0012	0.0010
	J_e	279.8642	256.6191	255.1380	0.3879	0.8736	0.6532
With noise	e_d	289.2396	261.2449	260.0939	0.9258	1.0587	0.9754
	e_f	0.0585	0.0289	0.0314	0.1136	0.1147	0.1258
	e_T	0.0016	0.0005	0.0005	0.0013	0.0008	0.0010
	e_T_Std	0.0219	0.0203	0.0203	0.0203	0.0209	0.0204
	J_e	289.3216	261.2945	260.1460	1.0610	1.1951	1.1225
Anti-interference	ΔJ_e	0.0338	0.0182	0.0196	1.7352	0.3680	0.7184

Table 9 The comparison of J_e with and without noise (nonlinear methods)

		Extended Kalman	Exponential forgetting	Optimization		
				Gradient descent	Genetic algorithm	Particle filter
Without noise	e_d	71.7454	72.6643	0.2243	0.8152	0.7340
	e_f	3.0663	3.4116	0.0681	0.0376	0.0182
	e_T	0.0001	0.0001	0.0014	0.0001	0.0001
	e_T_Std	0.0061	0.0060	0.0019	0.0013	0.0007
	J_e	74.8180	76.0820	0.2956	0.8541	0.7530
With noise	e_d	74.0046	74.4194	0.8248	0.8351	0.8438
	e_f	48.4565	54.6694	0.1287	0.1776	0.1493
	e_T	0.0004	0.0004	0.0005	0.0008	0.0006
	e_T_Std	0.0212	0.0211	0.0197	0.0207	0.0197
	J_e	122.4827	129.1102	0.9737	1.0343	1.0135
Anti-interference	ΔJ_e	0.6371	0.6970	2.2936	0.2110	0.3459

Table 10 The comparison between the IDIM-IV method and Pseudo-inverse method for model (7)

	Pseudo-inverse	IDIM-IV
J_e (with filter)	411.5920	412.3433
J_e (without filter)	1911.7927	446.4756
ΔJ_e	3.6449	0.0828

the IDIM-IV method, it is compared with the pseudo-inverse method. First, $\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m$, and $\ddot{\boldsymbol{q}}_m$ are filtered offline with a 20-Hz fourth-order Butterworth filter. J_e of the pseudo-inverse method is calculated with the filtered parameters. The pseudo-inverse method is then performed again without filtering. In both cases, the IDIM-IV method builds the instrument variable with the simulated data to calculate J_e. The change of J_e is shown in Table 10. It can be seen that the IDIM-IV method has a smaller ΔJ_e than the pseudo-inverse method. Therefore, the IDIM-IV method has a better anti-interference ability in this case.

Table 11 The feature levels of different methods

		Pseudo-inverse	Kalman	Extended Kalman	Exponential forgetting	Optimization		
						Gradient descent	Genetic algorithm	Particle filter
Error	e_d	C	C	C	C	B	B	B
	e_f	A	A	B	B	A	A	A
	e_T	A	A	A	A	A	A	A
	J_e (without noise)	C	C	C	C	A	A	A
	Follow changing parameter	C	B	B	A	C	C	C
Solution time		OfflineA	Online	Online	Online	OfflineB	OfflineC	OfflineB
Linear/ nonlinear		Linear	Linear	Nonlinear	Linear/ nonlinear	Linear/ nonlinear	Linear/ nonlinear	Linear/ nonlinear
Anti-interference	ΔJ_e	A	A	B	B	C	B	B

6.5 Summary of calculation methods

In summary, Table 11 presents the feature levels of different methods. The feature is marked ‘A’, ‘B’ and ‘C’ as defined in Section 5. For the methods that apply to both linear and nonlinear models, the evaluation indexes are averaged before marking. It can be seen that different methods adapt to different applications. The features of these methods are summarized as follows.

- The pseudo-inverse method. The pseudo-inverse method is accurate in estimating the Coulomb friction and viscous friction parameters and has a strong anti-interference ability. It is easy to calculate and can be used in offline calculations.
- The Kalman method. The Kalman method has similar features to the pseudo-inverse method but is used in online calculations.
- The extended Kalman method. The extended Kalman method is an online calculation method for nonlinear models. The anti-interference ability is weaker than the Kalman method.
- The exponential forgetting method. The exponential forgetting method is similar to the Kalman filter but has advantages in following changing parameters.
- The IDIM-IV method. The IDIM-IV method has a better anti-interference ability in noisy systems with known robot control laws. It is suitable for linear models.
- The optimization methods. The optimization methods can ensure the results are reasonable, but they require to estimate the parameter ranges and the solution time is long. The friction parameter errors of the optimization methods are relatively large. The gradient descent method requires to estimate the initial values. The genetic algorithm takes the longest time to obtain a solution.

6.6 Instance

In this experiment, the Efort robot ER20-C10 is taken as an example to illustrate how to choose the calculation method and model. For this industrial robot, the requirements for parameter calibration are as follows:

- The parameter error J_e should be as small as possible.
- Under the premise of small J_e, the calculation method should adapt to different models, take less solution time, and have a certain degree of anti-interference ability.
- There are no requirements for online and offline calibration.

Considering the characteristics of each calculation method, the particle filter method is selected for calculation. The robot is then controlled to move along an optimal trajectory and the points are also sampled at 100 Hz. The torque of each joint is calculated from the motor current. The angular position is measured by the encoder. The angular velocity and angular acceleration are obtained from differentiation and filtered with a 20-Hz fourth-order Butterworth filter. As we have no knowledge of the structural characteristics of the robot, it is difficult to choose the model. Therefore, we try different models with

Table 12 The experiment result of torque errors

	Linear models			Nonlinear models		
	Eq. (7)	Eq. (8)	Eq. (9)	Eq. (4)	Eq. (5)	Eq. (6)
Mean (Nm)	0.4478	0.0469	0.4509	0.4085	2.7903	1.2232
Std (Nm)	44.7068	30.8389	44.417	19.1073	51.8087	25.5865

the particle filter method. The Coulomb friction coefficients and viscous friction coefficients of different models are set to the same ranges. The torque errors of different models are compared in Table 12. It can be seen that all the e_T are small. However, the e_T_Std of (4) is the smallest. Thus the Lugre model fits better with this robot than other tested models.

The torque error is related to the robot and trajectory used for testing. If the static friction of the driving system is small and the absolute velocity is high, the difference between the two types of model is small. Because different robots have different driving systems, the best model may not be the same.

7 Conclusion

In this paper, several typical linear/nonlinear calibration models for the inertial parameters and friction parameters of multi-joint robots are introduced. Different calculation methods are compared quantitatively and the features are analyzed. The linear model has fewer parameters and the calculation is easier. However, friction is usually nonlinear in practice. Linear models can be taken as approximations. The pseudo-inverse method and the Kalman filter are commonly used calculation methods for linear models. The extended Kalman filter is a nonlinear online method. The index forgetting method has advantages in following changing parameters. The IDIM-IV method has a better anti-interference ability to the noise of $\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m$, and $\ddot{\boldsymbol{q}}_m$. The optimization methods are suitable for linear/nonlinear models with parameter constraints. In applications, the calculation method is determined first according to the requirements of a specified robot. A series of points are then sampled by experiments. Different models are tested with the selected calculation method to determine the most suitable model. More complex models can also be used for comparison. However, there is a trade-off between complexity and effectiveness.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. U1713222, 61773378, 61421004, U1806204), Beijing Science and Technology Project (Grant No. Z181100003118006), and Youth Innovation Promotion Association CAS.

References

1 Jamisola R. Dynamics identification and control of an industrial robot. In: Proceedings of the 9th International Conference on Advanced Robotics, Kyongju, 1999. 749–754

2 Hollerbach J, Khalil W, Gautier M. Model identification. In: Springer Handbook of Robotics. Berlin: Springer, 2016. 113–138

3 van Damme M, Beyl P, Vanderborght B, et al. Estimating robot end-effector force from noisy actuator torque measurements. In: Proceedings of IEEE International Conference on Robotics and Automation, Shanghai, 2011. 1108–1113

4 Richalet J, Fiani P. The global approach in identification protocol optimization. In: Proceedings of International Conference on Control Applications, Albany, 1995. 423–431

5 Gautier M, Janot A, Vandanjon P O. A new closed-loop output error method for parameter identification of robot dynamics. IEEE Trans Contr Syst Technol, 2013, 21: 428–444

6 Janot A, Vandanjon P O, Gautier M. A generic instrumental variable approach for industrial robot identification. IEEE Trans Contr Syst Technol, 2014, 22: 132–145

7 Montazeri A, West C, Monk S D, et al. Dynamic modelling and parameter estimation of a hydraulic robot manipulator using a multi-objective genetic algorithm. Int J Control, 2017, 90: 661–683

8 Wensing P M, Kim S, Slotine J J E. Linear matrix inequalities for physically consistent inertial parameter identification: a statistical perspective on the mass distribution. IEEE Robot Autom Lett, 2018, 3: 60–67

9 Sousa C D, Cortesão R. Physical feasibility of robot base inertial parameter identification: a linear matrix inequality approach. Int J Robot Res, 2014, 33: 931–944

10 Welch G, Bishop G. An Introduction to the Kalman Filter. Chapel Hill, Technical Report. 1995

11 Gautier M, Poignet P. Extended Kalman filtering and weighted least squares dynamic identification of robot. Control Eng Practice, 2001, 9: 1361–1372

12 Bona B, Indri M. Friction compensation in robotics: an overview. In: Proceedings of the 44th IEEE Conference Decision Control, Seville, 2005. 4360–4367

13 Swevers J, Verdonck W, Schutter J D. Dynamic model identification for industrial robots. IEEE Control Syst, 2007, 27: 58–71

14 Ding L, Wu H, Yao Y, et al. Dynamic model identification for 6-DOF industrial robots. J Robot, 2015, 2015: 1–9

15 Freidovich L, Robertsson A, Shiriaev A, et al. LuGre-model-based friction compensation. IEEE Trans Contr Syst Technol, 2010, 18: 194–200

16 Astrom K J, Carlos C C. Revisiting the LuGre friction model. IEEE Control Syst, 2008, 28: 101–114

17 Bompos N A, Artemiadis P K, Oikonomopoulos A S, et al. Modeling, full identification and control of the mitsubishi PA-10 robot arm. In: Proceedings of 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Zurich, 2007. 1–6

18 Wernholt E, Gunnarsson S. Nonlinear identification of a physically parameterized robot model 1. IFAC Proc Vol, 2006, 39: 143–148

19 Grotjahn M, Daemi M, Heimann B. Friction and rigid body identification of robot dynamics. Int J Solids Struct, 2001, 38: 1889–1902

20 Lee S D, Ahn K H, Song J B. Torque control based sensorless hand guiding for direct robot teaching. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, 2016. 745–750

21 Blumenkranz S J, Prisco G M, DiMaio S P, et al. Force and torque sensing in a surgical robot setup arm. US Patent, 9 895 813, 2018

22 Erden M S, Jonkman J A. Physical human-robot interaction by observing actuator currents. Int J Robot Autom, 2012, 27: 233–243

23 Nagamatsu Y, Shirai T, Suzuki H, et al. Distributed torque estimation toward low-latency variable stiffness control for gear-driven torque sensorless humanoid. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, 2017. 5239–5244

24 Vuong N D, Marcelo A H, Li Y P, et al. Improved dynamic identification of robotic manipulators in the linear region of dynamic friction. IFAC Proc Vol, 2009, 42: 167–172

25 Vuong N D, Marcelo A H. Dynamic model identification for industrial robots. Acta Polytech Hung, 2009, 6: 51–68

26 Stürz Y R, Affolter L M, Smith R S. Parameter identification of the KUKA LBR iiwa robot including constraints on physical feasibility. IFAC-PapersOnLine, 2017, 50: 6863–6868

27 Duan X J, Zhi J H, Chen H M, et al. Two novel robust adaptive parameter estimation methods with prescribed performance and relaxed PE condition. Sci China Inf Sci, 2018, 61: 129203

28 Jahandideh H, Namvar M. Use of pso in parameter estimation of robot dynamics; part one: no need for parameterization. In: Proceedings of the 16th International Conference on System Theory, Control and Computing, Sinaia, 2012. 1–6

29 Craig J. Introduction to Robotics: Mechanics and Control. 3rd ed. Upper Saddle River: Pearson Education, 2005. 165–200

30 Kammerer N, Garrec P. Dry friction modeling in dynamic identification for robot manipulators: theory and experiments. In: Proceedings of 2013 IEEE International Conference on Mechatronics, Kagawa, 2013. 422–429

31 Wu D W, Liu Q, Xu W J, et al. External force detection for physical human-robot interaction using dynamic model identification. In: Proceedings of International Conference on Intelligent Robotics and Applications, 2017. 581–592

32 Wang X M, He X K, Bao Y, et al. Parameter estimates of Heston stochastic volatility model with MLE and consistent EKF algorithm. Sci China Inf Sci, 2018, 61: 042202

33 Janot A, Vandanjon P O, Gautier M. Identification of 6 DOF rigid industrial robots with the instrumental variable method. IFAC Proc Vol, 2012, 45: 1659–1664

34 Sousa C D, Cortesão R. Physically feasible dynamic parameter identification of the 7-DOF WAM robot. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013. 2868–2873

35 Marino I P, Miquez J. Gradient-descent methods for parameter estimation in chaotic systems. In: Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, Zagreb, 2005. 440–445

36 Reeves C R. Genetic Algorithms. Boston: Springer, 2010. 109–139

37 Candy J V. Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods. Hoboken: John Wiley Sons, 2009. 237–293

38 Wu J, Wang J, You Z. An overview of dynamic parameter identification of robots. Robot Comput-Integrated Manuf, 2010, 26: 414–419

39 Sun Y, Hollerbach J M. Observability index selection for robot calibration. In: Proceedings of 2008 IEEE International Conference on Robotics and Automation, London, 2008. 831–836

40 Bargsten V, Zometa P, Findeisen R. Modeling, parameter identification and model-based control of a lightweight robotic manipulator. In: Proceedings of 2013 IEEE International Conference on Control Applications (CCA), Hyderabad, 2013. 134–139

41 Calafiore G, Indri M, Bona B. Robot dynamic calibration: optimal excitation trajectories and experimental parameter estimation. J Robot Syst, 2001, 18: 55–68

42 Swevers J, Ganseman C, Tukel D B, et al. Optimal robot excitation and identification. IEEE Trans Robot Autom, 1997, 13: 730–740

43 Presse C, Gautier M. New criteria of exciting trajectories for robot identification. In: Proceedings of 1993 IEEE International Conference on Robotics and Automation, Atlanta, 1993. 907–912

44 Liu J C, Wu Z X, Yu J Z, et al. Sliding mode fuzzy control-based path-following control for a dolphin robot. Sci China Inf Sci, 2018, 61: 024201