Real-time collision detection based on external torque mutation suppression and time series analysis

Tianyi Zhang
Department of Intelligent Equipment, Gree Electric Appliances Inc of Zhuhai, Zhuhai, China and School of Intelligent Systems Science and Engineering/JNU-Industry School of Artificial Intelligence, Jinan University, Zhuhai, China

Haowu Luo
Department of Electric, College of Information Science and Technology, Jinan University, Guangzhou, China and Industrial Robot Intelligent Control Engineering Technology Research Center of Guangdong Province, Guangzhou, China

Ning Liu
Department of Electric, College of Information Science and Technology, Jinan University, Guangzhou, China; Industrial Robot Intelligent Control Engineering Technology Research Center of Guangdong Province, Guangzhou, China and School of Intelligent Systems Science and Engineering/JNU-Industry School of Artificial Intelligence, Jinan University, Zhuhai, China, and

Feiyan Min, Zhixin Liang and Gao Wang
Department of Electric, College of Information Science and Technology, Jinan University, Guangzhou, China and Industrial Robot Intelligent Control Engineering Technology Research Center of Guangdong Province, Guangzhou, China

Abstract

Purpose – As the demand for human–robot collaboration in manufacturing applications grows, the necessity for collision detection functions in robots becomes increasingly paramount for safety. Hence, this paper aims to improve the existing method to achieve efficient, accurate and sensitive robot collision detection.

Design/methodology/approach – The external torque is estimated by momentum observers based on the robot dynamics model. Because the state of the joints is more accessible to distinguish under the action of the suppression operator proposed in this paper, the mutated external torque caused by joint reversal can be accurately attenuated. Finally, time series analysis (TSA) methods can continuously generate dynamic thresholds based on external torques.

Findings – Compared with the collision detection method based only on TSA, the invalid time of the proposed method is less during joint reversal. Although the soft-collision detection accuracy of this method is lower than that of the symmetric threshold method, it is superior in terms of detection delay and has a higher hard-collision detection accuracy.

Originality/value – Owing to the mutated external torque caused by joint reversal, which seriously affects the stability of time series models, the collision detection method based only on TSA cannot detect continuously. The consequences are disastrous if the robot collides with people or the environment during joint reversal. After multiple experimental verifications, the proposed method still exhibits detection capabilities during joint reversal and can implement real-time collision detection. Therefore, it is suitable for various engineering applications.

Keywords Real-time collision detection, Suppression operator, Mutated external torque, Time series analysis, Dynamic threshold

Paper type Research paper

1. Introduction

As the demand for automatic production increases, collaborative robots are being gradually applied to production lines to assist employees in making products. In human–machine interaction scenarios, workers may accidentally collide with collaborative

The current issue and full text archive of this journal is available on Emerald Insight at: **https://www.emerald.com/insight/0143-991X.htm**

Industrial Robot: the international journal of robotics research and application
51/4 (2024) 640–657
Emerald Publishing Limited [ISSN 0143-991X]
[DOI 10.1108/IR-12-2023-0338]

The first institution that completed work with the article [10.1108/IR-12-2023-0338] is College of Information Science and Technology, Jinan University, Guangzhou, China. And the authors gratefully acknowledge the financial support for this work from the Shaoguan Science and Technology Project under Grant 230615178031046, the Zhuhai Science and Technology Project under Grant 2220004002542, 2220004002352.

Received 18 December 2023
Revised 23 February 2024
Accepted 14 March 2024

robots. To reduce the degree of damage to all parties after a collision, it is necessary for the robot to detect the collisions in time.

Collisions in robots can be divided into two types according to the intensity of the collision, namely, hard collision and soft collision. The difference between them is the collision duration and characteristics of external torque changes (Park *et al.*, 2020). When a robot is hit by a rigid object, it is usually called a hard collision due to its short duration. When the robot comes into contact with an elastic/flexible/soft object or is subjected to a slowly increasing external force, it is usually called a soft collision due to its long duration. Concerning hard collisions, the external torque shows a trend of rapid increase and is accompanied by a rigid impact. When the object is struck, it undergoes an elastic, undamped impact, rapidly dissipating energy, which may lead to deformation or damage. As for soft collisions, the robot cannot escape the collision state quickly and is subject to continuous resistance. It keeps moving synchronously with the object being hit or absorbs the resistance. At this time, the external torque will slowly increase, and no rigid impact will occur.

By analyzing how the robot obtains information, robot collision detection methods can be divided into two categories: those with external sensors and those without external sensors (Heo *et al.*, 2019). External sensors provide contact information for robots to interact with the outside world, but they also increase structural complexity and introduce data consistency issues (Liu *et al.*, 2021). The external sensors commonly used for implementing robot collision detection are force/torque sensors.

The physical identification method for applying arm surface materials is also limited by the behavioral cognition of the collision subject. In a nonstructurally constrained environment, there is an imbalance between coverage integrity and random collision response.

According to threshold characteristics, conventional robot collision detection methods without external sensors can be divided into two types: static threshold method and dynamic threshold method. Based on these types, further classification can be made according to the dependence on intelligent algorithms, as shown in Table 1. Each method in the table corresponds to a representative literature and will be introduced later.

The static threshold method generally sets a fixed threshold, and its range is larger than the statistical peak value of previous experimental values. For external torque acquisition, besides the direct estimation method (Suita *et al.*, 1995) and the estimation method based on harmonic reducer deformation (Wang *et al.*, 2023), numerous observer-based estimation methods have been proposed (De Luca *et al.*, 2006; Haddadin, 2013; De Luca and Mattone, 2005; Duan *et al.*, 2023; Ren *et al.*, 2018). However, collision detection based on these external torque estimation methods needs to be more sensitive.

For the range limitation problem of static threshold methods, neural networks (NNs) and deep learning can effectively extend the statistical characteristics of time series to the accumulation and prediction of data patterns, and a high collision detection performance can be obtained by using limited and fixed training targets. The deep NN CollisionNet eliminates the cumbersome heuristic decision-making process proposed to detect collisions (Heo *et al.*, 2019). In addition, a multilayer NN is used to fit the robot dynamics data model to supplement or replace the physical model, which plays an important role in collision detection (Sharkawy and Aspragathos, 2018). To set the collision detection threshold closer to the changing trend, using the convolutional NN (CNN) to compensate for the uncertain data characteristics based on the existing dynamic model (Park *et al.*, 2020). After training on the collision data set, the fuzzy system achieved fast and accurate collision detection (Dimeas *et al.*, 2015). Intelligent algorithms can be used to design not only static thresholds but also dynamic thresholds, such as the long short-term memory network, which is used to design high-fit thresholds (Zhang *et al.*, 2023).

However, the training resource consumption of intelligent algorithms and the iterative nested regression of heuristic algorithms all put forward computing power requirements for real-time robot control frameworks. To address this problem, the dynamic threshold method, which usually models the external torque representing the uncertainty of the model and sets the output threshold according to the law, is recommended (Zhang and Hong, 2019; Sotoudehnejad and Kermani, 2014; Li *et al.*, 2021; Indri *et al.*, 2015). Dynamic thresholds generated by methods lacking statistical tools may exceed symmetric static thresholds or be subject to certain application conditions, making them ineffective.

For the above problems of collision detection methods based on dynamic thresholds, time series analysis (TSA) in statistics provides some ideas. Although statistical methods are rarely used in the field of robotics, TSA has shown excellent results in fault detection problems in different fields (Feng *et al.*, 2023; Lin *et al.*, 2019; Liu *et al.*, 2019). Collisions during robot movement can also be regarded as system anomalies. Therefore, applying TSA to robot collision detection is feasible. The sensorless collision-detection method based only on TSA generates dynamic thresholds

Table 1 Collision detection method based on the force/torque sensorless

Collision detection method	Advantage	Disadvantage
1. The static threshold method		
Based on Nonintelligent algorithms (De Luca and Mattone, 2005)	– Simple threshold	– Limited range – Invalid occasion
Based on intelligent algorithms (Park *et al.*, 2020)	– Detection performance moderate	– Data set need – Much calculation
2. The dynamic threshold method		
Based on nonintelligent algorithms (Zhang *et al.*, 2021)	– Detection performance high – Real-time	– Invalid occasion
Based on intelligent algorithms (Zhang *et al.*, 2023)	– Detection performance high	– Data set need – Much calculation

Source: Authors' own work

based on historical information of external torques, conforms to the regular control of data feature statistics, and can achieve more accurate and sensitive collision detection (Zhang *et al.*, 2021). However, there is a blind spot in the detection of joint reversals, which affects collision detection performance.

In this paper, we propose a method that suppresses abnormal external torque estimates to solve the problem of short-term identification failure of collisions when robot joints change their direction. This method is integrated with TSA to build a data model that continuously generates high-fit dynamic thresholds after the initialization conditions are met, and achieves sensitive and real-time collision detection. The new collision detection method improves the short-term identification failure of collisions during joint reversal and reduces the detection delay. When it operates in a robot control system, the block diagram is shown in Figure 1.

The remainder of this paper is organized as follows. Section 2 introduces the external torque estimation based on momentum observers and the problem of the collision-detection method based only on TSA. The proposed method is described in Section 3. Section 4 presents experimental verification and data analysis.

2. Problem description of the collision detection method based only on TSA

2.1 Dynamic model construction

When the robot is running in the position-control mode, the angle difference between the motor and link sides can be ignored. If the robot comes in contact with the external environment under this condition, the dynamic model is given by equation (1):

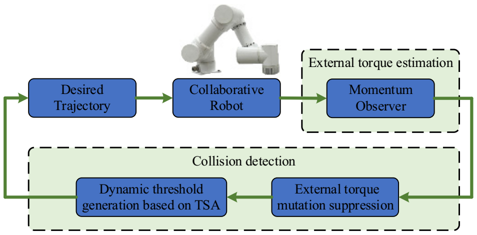
$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{f}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau}_m + \boldsymbol{\tau}_{ext} \tag{1}$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ are the joint position, joint velocity and joint acceleration, respectively; $\boldsymbol{M}(\boldsymbol{q})$ is the inertial matrix; $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is a matrix of the centrifugal and Coriolis torques; $\boldsymbol{G}(\boldsymbol{q})$ is the gravitational torque vector; $\boldsymbol{f}(\dot{\boldsymbol{q}})$ is the friction torque vector obtained by the Coulomb-viscous friction model; $\boldsymbol{\tau}_m$ is the output torque vector of the joint motor; $\boldsymbol{\tau}_{ext}$ is the external torque vector.

2.2 External torque estimation

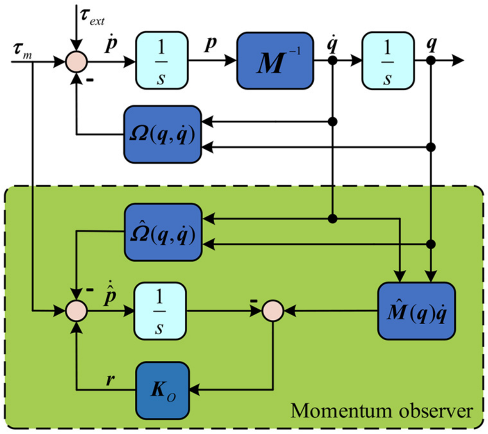
The dynamic model shown in equation (1) requires the joint acceleration $\ddot{\boldsymbol{q}}$ to calculate the external torque, but the acquisition of $\ddot{\boldsymbol{q}}$ depends on the sensor and cannot be calculated directly from the velocity difference. Therefore, the observer method can be used to obtain the external torque (Haddadin *et al.*, 2017). The generalized momentum observer based on the dynamic model is shown in Figure 2.

Figure 1 Block diagram of the closed-loop control system


Desired Trajectory
Collaborative Robot
External torque estimation
Momentum Observer
Collision detection
Dynamic threshold generation based on TSA
External torque mutation suppression

Source: Authors' own work

Figure 2 Block diagram of momentum observer



Source: Authors'own work

Because $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is an antisymmetric matrix, matrix $\dot{\boldsymbol{M}}(\boldsymbol{q})$ can be simplified as shown in equation (2):

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) = C(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{C}^T(\boldsymbol{q},\dot{\boldsymbol{q}}) \tag{2}$$

To facilitate the subsequent derivation of equations, we define the vector $\boldsymbol{\Omega}(\boldsymbol{q},\dot{\boldsymbol{q}})$ as shown in equation (3):

$$\begin{aligned}\boldsymbol{\Omega}(\boldsymbol{q},\dot{\boldsymbol{q}}) &:= \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{f}(\dot{\boldsymbol{q}}) - \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ &= \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{f}(\dot{\boldsymbol{q}}) - \boldsymbol{C}^T(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\end{aligned} \tag{3}$$

The obtained generalized momentum p is shown in equation (4):

$$\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4}$$

Combining equations (1) and (4), the derivative of the generalized momentum $\dot{p}$ is given by equation (5):

$$\dot{\boldsymbol{p}} = \boldsymbol{C}^T(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{f}(\dot{\boldsymbol{q}}) + \boldsymbol{\tau}_m + \boldsymbol{\tau}_{ext} \tag{5}$$

The symbol $\boldsymbol{r}$ is replaced with $\boldsymbol{\tau}_{ext}$ and combined with equation (3) to obtain the dynamic response of the momentum observer, as shown in equation (6):

$$\begin{cases}\dot{\hat{\boldsymbol{p}}} = \boldsymbol{\tau}_m - \hat{\boldsymbol{\Omega}}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{r} \\ \dot{\boldsymbol{r}} = \boldsymbol{K}_O\left(\dot{\boldsymbol{p}} - \dot{\hat{\boldsymbol{p}}}\right)\end{cases} \tag{6}$$

where $\boldsymbol{K}_o$ is the diagonal gain matrix of the observer; $\boldsymbol{r}(t)$ is the observer output, which can be obtained by integrating $\dot{\boldsymbol{r}}$, as shown in equation (7):

$$\boldsymbol{r} = \boldsymbol{K}_O\left(\boldsymbol{p}(t) - \int_0^t \dot{\hat{\boldsymbol{p}}}(s)ds - \boldsymbol{p}(0)\right) \tag{7}$$

Because $\boldsymbol{p} = \hat{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}}$, and under ideal conditions, $\hat{\boldsymbol{M}} = \boldsymbol{M}$ and $\hat{\boldsymbol{\Omega}} = \boldsymbol{\Omega}$, the dynamic relationship between $\boldsymbol{\tau}_{ext}$ and $\boldsymbol{r}$ is given by equation (8):

$$\dot{\boldsymbol{r}} = \boldsymbol{K}_O(\boldsymbol{\tau}_{ext} - \boldsymbol{r}) \tag{8}$$

The transient response of component $\boldsymbol{r}$ is related to the same component of $\boldsymbol{\tau}_{ext}$. In the limit case ($\boldsymbol{K}_O \to \infty$), $\boldsymbol{r}$ and $\boldsymbol{\tau}_{ext}$ satisfy the relationship in equation (9):

$$\boldsymbol{r} \approx \boldsymbol{\tau}_{ext} \tag{9}$$

2.3 Short-term failure during the joint reversal

The collision detection method based only on TSA (Zhang *et al.*, 2021) uses the autoregressive model (AR) as the external torque time-series model. Although collisions can be detected without adjusting the motion trajectory parameters, short-term recognition failure still occurs during the joint reversal. The reason for suspending the method is that the mutated external torque caused by joint reversal causes abnormal statistical information in the time series model, which makes the model unstable. The short-term invalid time of this method is $t_{pause} = t_{recover} + t_{init}$, where $t_{recover}$ is the time required for the external torque to return from the abnormal value to the normal value and t_{init} is the time required to initialize the model coefficients once.

Joint velocity is an important feature for judging joint reversals. To maintain the stability of collision detection, the condition for the judgment of joint reversal is chosen to be $\dot{\boldsymbol{q}} \leq 0.09\text{rad/s}$ in this paper. That is, lower than the velocity of joint creep $\dot{\boldsymbol{q}} = 0.1\text{rad/s}$ can be regarded as a sign of clockwise/counterclockwise. The external torque of Joint 1 when the robot runs a sinusoidal trajectory and the dynamic threshold generated by this method are shown in Figure 3. Notably, there is an absence of dynamic thresholds within the region highlighted by the arrow in Figure 3. However, this period lasted for 536 ms. If a collision occurred at this time, the method would be invalid.

3. Collision detection based on external torque mutation suppression and TSA

Generally, when a joint changes direction, it goes through three states in sequence: the motion state, transition state between motion and rest and rest state. To solve the problem of the short-term failure of collision detection during joint reversal, it is necessary to handle the mutated external torque during this period.

3.1 Improvement based on external torque mutation suppression

Inspired by the approach that compensates for model uncertainty (Cao *et al.*, 2019), the operator to suppress mutated external torques is constructed using attenuation exponential and cosine functions based on joint velocity characteristics. The suppressed external torque is obtained by multiplying the external torque and the suppression operator. The suppression operator O_{sp} is given by equation (10):

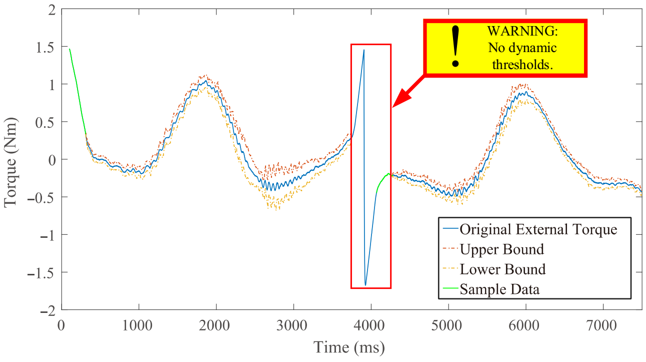
$$\begin{cases} \mathrm{O}_{sp} = \left(\cos\left(e^{-\rho\cdot\dot{\boldsymbol{q}}^2}\right)\right)^n \\ \boldsymbol{\tau}_{sp} = \boldsymbol{\tau}_{ext} \cdot O_{sp} \end{cases} \tag{10}$$

where ρ is used to adjust the degree of external torque tracking when the joint does not change its direction; n is used to adjust the suppression effect of the external torque during joint reversal, which directly affects the stability of the timing model; $\boldsymbol{\tau}_{sp}$ is the suppressed external torque.

When the robot runs a sinusoidal trajectory, the results of the external torque of Joint 1 before and after the suppression process under different ρ and n are shown in Figures 4 and 5, respectively. When $\rho = 150$ and $n = 16$, the suppressed external torque changed smoothly during joint reversal.

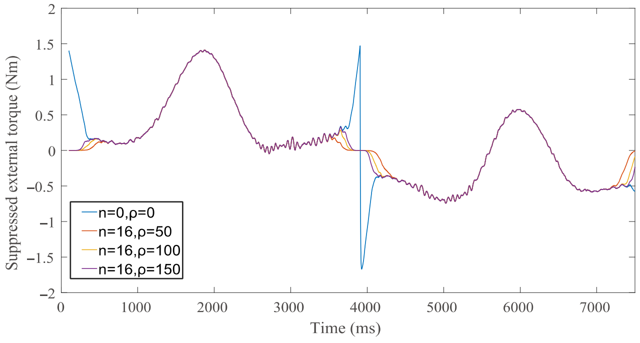
From equation (10), the suppressed external torque is directly affected by the value of the suppression operator. When $\mathrm{O_{sp}} = 0$, the suppressed external torque is 0. When $\mathrm{O_{sp}} \geq 0.1$, the suppressed external torque caused by collisions can still be reflected. When $\mathrm{O_{sp}} < 0.1$, the suppressed external torque

Figure 3 The defect of the collision detection method based only on TSA



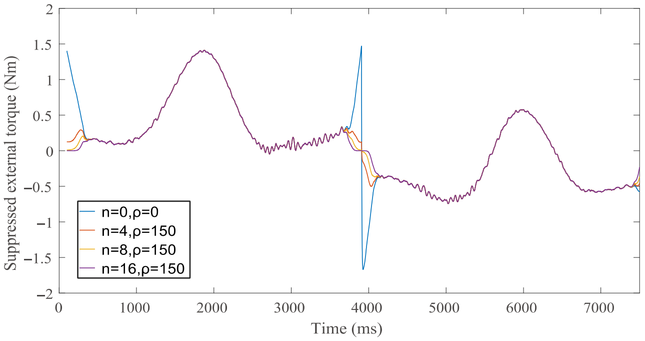
Source: Authors' own work

Figure 4 The comparison of external torques before and after suppression with different ρ



Source: Authors' own work

Figure 5 The comparison of external torques before and after suppression with different *n*



Source: Authors' own work

caused by the collision is difficult to reflect, and the method proposed in this paper can be regarded as invalid. The values of the suppression operator for different ρ and n are shown in Figure 6.

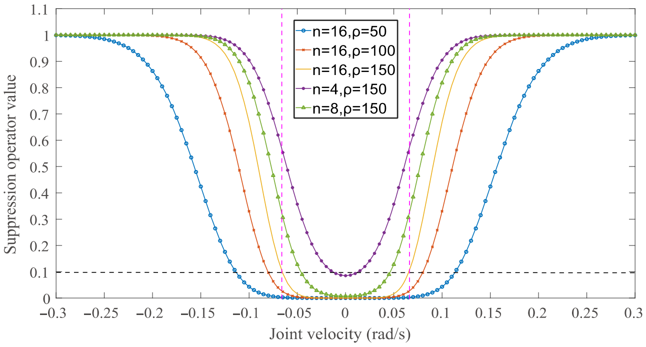
O_{sp} is symmetrical about the vertical line where the joint velocity is 0. As shown in Figure 6, when $\rho = 150$, $n = 16$ and $O_{sp} = 0.1$, the corresponding joint speed value was approximately ±0.0657 rad/s.

In this paper, we combined external torque mutation suppression and TSA to obtain a novel collision detection method. The process of this method is illustrated in Figure 7, where Processes 1 and 2 are used for offline simulation, and Process 3 is used for online robot control. However, as shown in Figure 3, the collision detection method based only on TSA still needs to wait for the re-initialization, which can resume detection after the completion of joint reversal. According to the above analysis, the short-term failure duration of the method used in this paper is significantly shorter than that of the method based only on TSA.

The pseudocode corresponding to Figure 7 is as follows:

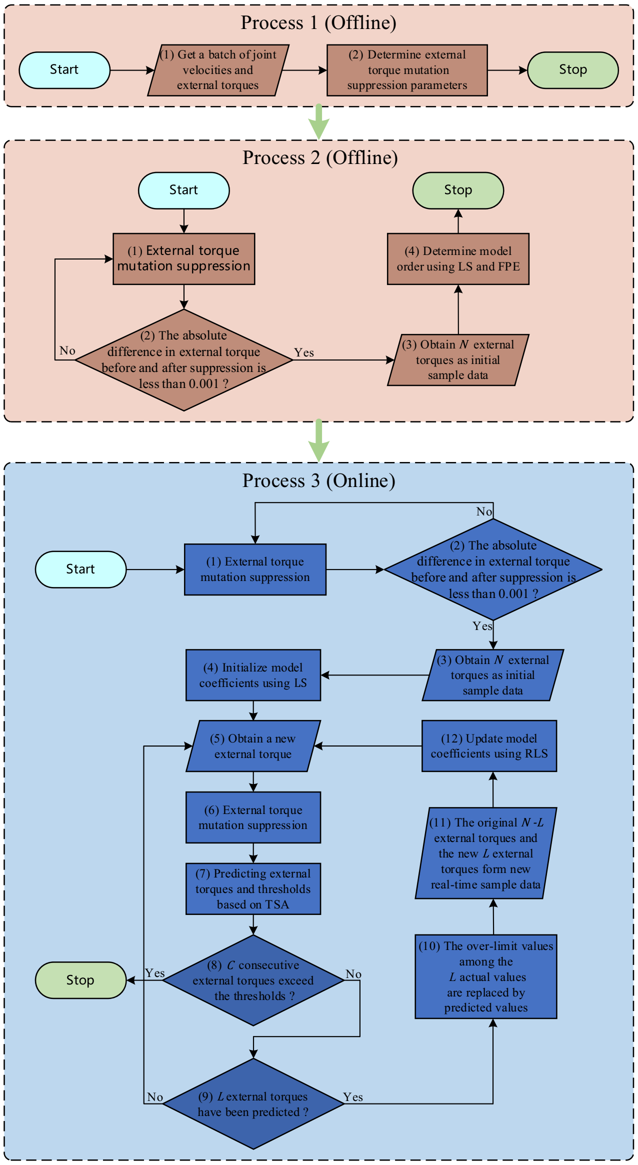
```
Collision detection method based on external
  torque mutation suppression and TSA.

  Input: Suppression parameters 1, $\rho$
         Suppression parameters 2, $n$
         Model order, $u$
         Prediction length, $L$
         Confidence level, $\gamma$
         Forgetting factor, $\lambda$
         Threshold Margin, $\psi$
         The total amount of sample data, $N$
         Number of external torque exceeding the threshold, C
         Original external torque, $\boldsymbol{\tau}_{ext}$
         Joint velocity, $\dot{\boldsymbol{q}}$
         Running time, $run\_time$

  Output: Collision detection result, $result$
  while $t \le run\_time$ do
    $t \leftarrow t + 1$;
    $O_{sp} \leftarrow (\cos(\exp(-\rho * \dot{\boldsymbol{q}}^2)))n$;  //Suppression operator initialization.
    $\boldsymbol{\tau}_{sp} \leftarrow \boldsymbol{\tau}_{ext}{}^{*} O_{sp}$; //External torque mutation suppression.
    if abs $(\boldsymbol{\tau}_{ext} - \boldsymbol{\tau}_{sp}) \ge$ 1e-3 && firstinit == false //Determine whether the initialization conditions are met.
      continue;
    else if abs $(\boldsymbol{\tau}_{ext} - \boldsymbol{\tau}_{sp}) <$ 1e-3 && firstinit == false
      $\boldsymbol{\tau}_{SP} \leftarrow [\boldsymbol{\tau}_{SP} \cdot \boldsymbol{\tau}_{sp}]$;
    else
      $\boldsymbol{\tau}_{SP} \leftarrow [\boldsymbol{\tau}_{SP}(2:\mathrm{end})\ \boldsymbol{\tau}_{sp}]$;
    end if
    len ← len + 1;
    if len == $N$ && firstinit == false
      $\boldsymbol{\theta} \leftarrow$ LS_init $(\boldsymbol{\tau}_{SP})$; //Using LS algorithm to Initialize model coefficients.
      firstinit ← true;
      continue;
    end if
    if fistinit == true
      $\boldsymbol{\tau}_{pd}, \boldsymbol{\eta}_{up}, \boldsymbol{\eta}_{down} \leftarrow$ TSA_generate_dyth $(\boldsymbol{\tau}_{SP}$,
```

Figure 6 Suppression operator O_{sp} under different ρ and n



Source: Authors' own work

Figure 7 Flow chart of the method proposed in this paper


Process 1 (Offline)
Start
(1) Get a batch of joint velocities and external torques
(2) Determine external torque mutation suppression parameters
Stop
Process 2 (Offline)
Start
Stop
(1) External torque mutation suppression
(4) Determine model order using LS and FPE
No
(2) The absolute difference in external torque before and after suppression is less than 0.001 ?
Yes
(3) Obtain N external torques as initial sample data
Process 3 (Online)
No
Start
(1) External torque mutation suppression
(2) The absolute difference in external torque before and after suppression is less than 0.001 ?
Yes
(4) Initialize model coefficients using LS
(3) Obtain N external torques as initial sample data
(5) Obtain a new external torque
(12) Update model coefficients using RLS
(6) External torque mutation suppression
(11) The original N -L external torques and the new L external torques form new real-time sample data
(7) Predicting external torques and thresholds based on TSA
(10) The over-limit values among the L actual values are replaced by predicted values
Stop
Yes
(8) C consecutive external torques exceed the thresholds ?
No
No
(9) L external torques have been predicted ?
Yes

Source: Authors' own work

```
$\psi$, u, $\gamma$, $\theta$); //Using TSA to generate dynamic thresholds.
        $\tau_{PD} \leftarrow [\tau_{PD}\tau_{pd}]$;
        $result \leftarrow$ consecutive_values_out_of_dyth($C$, $\tau_{SP}$, $\eta_{up}$, $\eta_{down}$); //Check whether there are C consecutive external torque exceedances
        if $result$ == false && len % $L$ == 0
        $\tau_{SP}$(end−L+1:end) ← replace_values_out_of_Dyth $\tau_{SP}$ (end−L+ 1: end), $\tau_{PD}$); //The exceeding limit value among the L actual external torque values is replaced by the predicted value.
        $\tau_{PD} \leftarrow []$;
        $\theta$ ← RLS_update ($\tau_{SP}$, $\lambda$); //Using RLS algorithm to update model coefficients.
    endif
    if $result$ == true
        return $result$;
    endif
  endif
end while
```

3.2 Model coefficient identification based on least squares

The u-order autoregressive model $AR(u)$ of the suppressed external torque time series $\{\tau_{sp}(t)\}$ is expressed in equation (11) (Box *et al.*, 2015):

$$\begin{cases} \tau_{sp}(t) = \boldsymbol{x}(t)\boldsymbol{\theta} + \varepsilon(t) \\ \boldsymbol{x}(t) = \left[\tau_{sp}(t-1)\,\tau_{sp}(t-2)\ldots\tau_{sp}(t-u)\right] \\ \boldsymbol{\theta} = [\alpha_1 \quad \alpha_2\ldots\alpha_u]^T \end{cases} \tag{11}$$

where $\boldsymbol{x}$ is the historical external torque vector, $\boldsymbol{\theta}$ is the autoregressive coefficient vector and $\varepsilon(t)$ is the white noise with a mean of 0 and variance of $\boldsymbol{\sigma}_\varepsilon^2$. The total amount of sample data is assumed to be N. When $\{t = u + 1, u + 2, \ldots, N\}$, equation (11) can form the following linear equation (12):

$$\begin{cases} \boldsymbol{Y} = \boldsymbol{X}\cdot\boldsymbol{\theta} + \boldsymbol{E}_r \\ \boldsymbol{Y} = \left[\tau_{sp}(u+1) \quad \tau_{sp}(u+2)\ldots\tau_{sp}(N)\right]^T \\ \boldsymbol{X} = \left[\boldsymbol{x}^T(u+1)\,\boldsymbol{x}^T(u+2)\ldots\boldsymbol{x}^T(N)\right]^T \\ \boldsymbol{E}_r = \left[\varepsilon(u+1)\,\varepsilon(u+2)\ldots\varepsilon(N)\right]^T \end{cases} \tag{12}$$

where $\boldsymbol{Y}$ is the external torque vector, $\boldsymbol{X}$ is the observation matrix of the autoregressive coefficients and $\boldsymbol{E}_r$ is the error vector.

Therefore, the least squares (LS) estimate of $\boldsymbol{\theta}$ is given by equation (13):

$$\hat{\boldsymbol{\theta}}_{LS} = \left(\boldsymbol{X}^T\cdot\boldsymbol{X}\right)^{-1}\cdot\boldsymbol{X}^T\cdot\boldsymbol{Y} \tag{13}$$

3.3 Model order determination

After obtaining the parameter identification result $\hat{\boldsymbol{\theta}}_{LS}$ based on the time series, the prediction error of the model $\hat{\boldsymbol{E}}_r$ can be expressed as shown in equation (14):

$$\hat{\boldsymbol{E}}_r = \boldsymbol{Y} - \boldsymbol{X}\cdot\hat{\boldsymbol{\theta}}_{LS} \tag{14}$$

An appropriate model order minimizes the error between the prediction results of the model and the actual external torque. The final prediction error (FPE) criterion effectively balances errors arising from an excessive number of parameters or missing parameters. This is achieved by determining model order that minimizes the variance of the one-step prediction error.

In this paper, the one-step prediction error sequence during the offline analysis was assumed to be $\{\tilde{\tau}_{sp}(j), j = \{u+1, u+2, \ldots, N\}\}$, where $\tilde{\tau}_{sp}(j)$ is the one-step prediction error of the model, and its expression is given by equation (15):

$$\tilde{\tau}_{sp}(j) = \tau_{sp}(j) - \hat{\tau}_{sp}(j) \tag{15}$$

where $\tau_{sp}(j)$ is the true value of the external torque at moment j. $\hat{\tau}_{sp}(j)$ is the predicted external torque at moment j, and its expression is given by equation (16):

$$\hat{\tau}_{sp}(j) = \boldsymbol{x}(j)\hat{\boldsymbol{\theta}}_{LS} \tag{16}$$

$\hat{\boldsymbol{\sigma}}_\varepsilon^2$ is the variance estimate of white noise $\varepsilon(t)$, and its expression is shown in equation (17).

$$\hat{\boldsymbol{\sigma}}_\varepsilon^2 = \frac{1}{N-u}\cdot\hat{\boldsymbol{E}}_r^T\cdot\hat{\boldsymbol{E}}_r \tag{17}$$

The FPE is the variance of the one-step forecast error sequence, a function of the model order u. Therefore, it can be assumed to be FPE(u), and its expression is shown in equation (18):

$$\text{FPE}(u) = \frac{N+u}{N}\cdot\hat{\boldsymbol{\sigma}}_\varepsilon^2 = \frac{N+u}{N(N-u)}\cdot\hat{\boldsymbol{E}}_r^T\cdot\hat{\boldsymbol{E}}_r \tag{18}$$

Therefore, the minimum FPE simplifies the model order and can achieve better prediction. The appropriate model order u_0 is selected according to these conditions, and the initial coefficient $\hat{\boldsymbol{\theta}}_{LS}$ of the model can be simultaneously obtained.

3.4 Dynamic threshold generation based on prediction results

The different predictive targets cause the different online/offline analysis equation.

During the online analysis, for the time serie $\{\tau_{sp}(t)\}$, the value at moment t and before can be used to predict the value of L times in the future. The suppressed external torque τ_{sp} predicted at moment $\{j = t + l, l = 1,2, \ldots, L\}$ can be assumed as $\tau_{pd}(l)$, where L is the prediction length. The one-step prediction error $\xi_{pd}(l)$ of $AR(u)$ is given by equation (19):

$$\xi_{pd}(l) = \tau_{sp}(t+l) - \hat{\tau}_{pd}(l) \tag{19}$$

where $\tau_{pd}(l)$ is the predicted values of $AR(u)$ at moment $\{j = t + l, l = 1,2, \ldots, L\}$ and L is the prediction length.

After determining the model coefficient $\boldsymbol{\theta}$, the recursive form of the predicted value $\tau_{pd}(l)$ based on $AR(u)$ is given by equation (20):

$$\begin{aligned}\tau_{pd}(l) = \alpha_1 \cdot \tau_{pd}(l-1) + \alpha_2 \cdot \tau_{pd}(l-2) + \\ \ldots + \alpha_u \cdot \tau_{pd}(l-u)\end{aligned} \tag{20}$$

where$\{\tau_{pd}(l-1), \tau_{pd}(l-2), \ldots, \tau_{pd}(l-u)\}$ represent the predicted values of $AR(u)$ at moment $\{t+l-1, t+l-2, \ldots, t+l-u\}$, respectively. If $l-h < 0 (h = 1,2, \ldots, u)$, then $\tau_{pd}(l-h)$ is replaced by $\tau_{sp}(t+l-h)$.

During the online analysis, the variance of the one-step prediction error $Var[\xi_{pd}(l)]$ of $AR(u)$ is given by equation (21):

$$Var\left[\xi_{pd}(l)\right] = E\left[\left(\tau_{\mathbf{sp}}(\mathbf{t+1}) - \tau_{\mathbf{pd}}(\mathbf{1})\right)^2\right] \tag{21}$$

In the TSA, the u-order autoregressive operator $\alpha(B)$ can be defined based on equation (11), as shown in equation (22):

$$\begin{cases}\alpha(B) = 1 - \alpha_1 \cdot B - \alpha_2 \cdot B^2 - \ldots - \alpha_u \cdot B^u \\ B\tau_{sp}(t) = \tau_{sp}(t-1)\end{cases} \tag{22}$$

where B is the delay operator that defines the relationship between the external torque at the current and previous moments. Therefore, $AR(u)$ shown in equation (11) can be rewritten as equation (23):

$$\tau_{sp}(t) = \alpha^{-1}(B)\varepsilon(t) \tag{23}$$

The s-order moving average model $MA(s)$ of the external torque time series $\{\tau_{sp}(t)\}$ is given by equation (24):

$$\begin{aligned}\tau_{sp}(t) = \varepsilon(t) + \beta_1 \cdot \varepsilon(t-1) + \beta_2 \cdot \varepsilon(t-2) \\ + \ldots + \beta_s \cdot \varepsilon(t-s)\end{aligned} \tag{24}$$

where $\beta_i(i = 1,2, \ldots, s)$ is the moving-average coefficient. $\beta(B)$ is defined as the s-order moving-average operator, as shown in equation (25):

$$\beta(B) = 1 + \beta_1 \cdot B + \beta_2 \cdot B^2 + \ldots + \beta_s \cdot B^s \tag{25}$$

During a stable motion process, the suppressed external torque fluctuates smoothly around zero, which satisfies the stability requirements of the time series model (Zhang *et al.*, 2021).

Under the condition that the process is stationary, the autoregressive process can be transformed into a moving average process (Chan and Cryer, 2008). Therefore, $\beta(B) = \alpha^{-1}(B)$ and $AR(u)$ shown in equation (11) is converted into $MA(\infty)$. Therefore, $MA(\infty)$ shown in equation (24) can be rewritten as equation (26):

$$\tau_{sp}(t+l) = \beta(B)\varepsilon(t+l) \tag{26}$$

where $(t+l)$ is the lth moment in the future.

During the conversion process, the expression for $\beta_i(i = 1,2, \ldots, \infty)$ can be obtained by combining equations (22) and (25), as shown in equation (27):

$$\begin{cases}\beta_1 = \alpha_1 \\ \beta_2 = \alpha_1 \cdot \beta_1 + \alpha_2 \\ \vdots \\ \beta_l = \alpha_1 \cdot \beta_{l-1} + \alpha_2 \cdot \beta_{l-2} + \ldots + \alpha_u \cdot \beta_{l-u} \\ \vdots\end{cases} \tag{27}$$

$\xi_{pd}(l)$ of the model can also be expressed as equation (28):

$$\begin{aligned}\xi_{pd}(l) = \varepsilon(t+l) + \beta_1 \cdot \varepsilon(t+l-1) + \ldots \\ + \beta_{l-1} \cdot \varepsilon(t+1)\end{aligned} \tag{28}$$

The variance of the one-step prediction error $Var[\xi_{pd}(l)]$ in equation (21) can be rewritten as in equation (29):

$$\begin{aligned}Var\left[\xi_{pd}(l)\right] &= \left(1 + \beta_1^2 + \beta_2^2 + \ldots + \beta_{l-1}^2\right)\boldsymbol{\sigma}_\varepsilon^2 \\ &= \left(1 + \sum_{i=1}^{l-1} \beta_i^2\right)\boldsymbol{\sigma}_\varepsilon^2\end{aligned} \tag{29}$$

The prediction error is normal white noise with a mean of 0 and variance of $Var[\xi_{pd}(l)]$. Consequently, the prediction result of $AR(u)$ at the lth moment in the future should also obey the normal distribution, with a mean of $\tau_{pd}(l)$ and variance of $Var[\xi_{pd}(l)]$. Therefore, when the confidence level is γ, the detection threshold at the lth moment in the future is given by equation (30):

$$\eta(l) = \tau_{pd}(l) \pm \left(\mu_{\frac{\gamma}{2}} \cdot \sqrt{\left(1 + \sum_{i=1}^{l-1} \beta_i^2\right)\hat{\boldsymbol{\sigma}}_\varepsilon^2} + \psi\right) \tag{30}$$

where $\mu_{\frac{\gamma}{2}}$ is the deviation coefficient and the standard normal distribution exceeds $\mu_{\frac{\gamma}{2}}$ with a probability of $\frac{\gamma}{2}$. Parameter ψ was used to adjust the threshold margin.

3.5 Model coefficient update based on recursive least squares

To reduce the computational complexity of generating dynamic thresholds based on prediction results, this paper uses the recursive least squares (RLS) algorithm to update the model coefficients (Song *et al.*, 2019), as shown in equation (31):

$$\begin{cases}\boldsymbol{K}_w = \boldsymbol{P}_{w-1} \cdot \boldsymbol{x}^T(w)\left[\lambda + \boldsymbol{x}(w)\boldsymbol{P}_{w-1} \cdot \boldsymbol{x}^T(w)\right]^{-1} \\ \boldsymbol{P}_w = \lambda^{-1} \cdot \left[\boldsymbol{I} - \boldsymbol{K}_w \cdot \boldsymbol{x}(w)\right] \cdot \boldsymbol{P}_{w-1} \\ \hat{\boldsymbol{\theta}}_w = \hat{\boldsymbol{\theta}}_{w-1} + \boldsymbol{K}_w \cdot \left[\tau_{sp}(w) - \boldsymbol{x}(w)\hat{\boldsymbol{\theta}}_{w-1}\right]\end{cases} \tag{31}$$

where $\boldsymbol{K}_w$ is the gain matrix, $\boldsymbol{P}_w$ is the covariance matrix and λ is the forgetting factor.

If no collision is detected in the $(w-1)$th detection, L-suppressed external torques obtained in this detection cycle are used to reconstruct $\tau_{sp}(w)$ and $\boldsymbol{x}(w)$. The RLS algorithm is then used to update $\boldsymbol{K}_w$, $\boldsymbol{P}_w$ and $\hat{\boldsymbol{\theta}}_w$, which are used for the wth prediction and threshold generation.

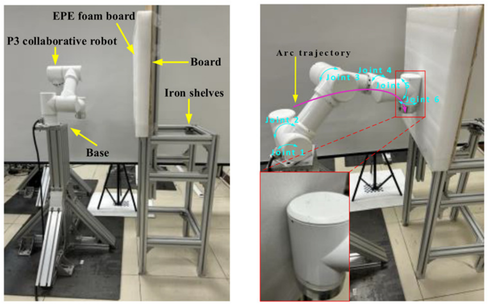
4. Experiment validation

4.1 Experiment platform

The 6DOF robot was modeled, and Joint 1 was used as an example to analyze the collision detection data.

The robot controller communicates with a universal collaborative robot based on the EtherCAT protocol, which has a payload of 3 kg and arm span of 0.54 m. The master station control architecture is Igh-EtherCAT +PREEMPT-RT, the control cycle is 1 ms, and the servo slaves are integrated modules. The setup is shown in Figure 8(a).

Figure 8

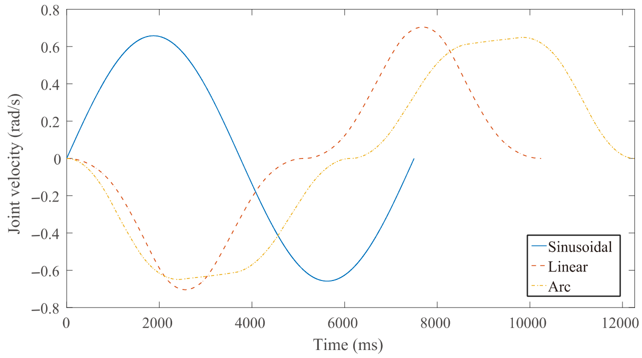


Notes: (a) Setup; (b) experiment with arc trajectory and soft collision
Source: Authors' own work

4.2 Experiment program

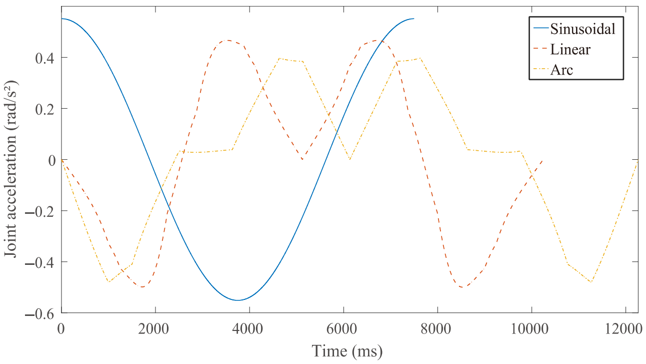
In this paper, we designed three trajectories: sinusoidal trajectory in joint space, linear trajectory in Cartesian space and arc trajectory in Cartesian space. The target motion data of Joint 1, when the robot runs these three trajectories, are shown in Figures 9 and 10.

Figure 9 The target velocity of Joint 1 when the robot runs the test trajectory



Source: Authors' own work

Figure 10 The target acceleration of Joint 1 when the robot runs the test trajectory



Source: Authors' own work

To verify the performance of collision detection methods, two main aspects need to be assessed: accuracy and detection delay. Repeated experiments were conducted to obtain results regarding these two aspects under hard or soft collision conditions, comparing the proposed method with the symmetric threshold method.

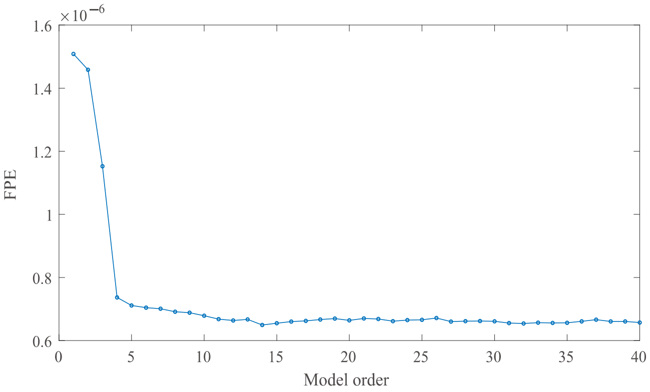
4.3 Parameters setting

Setting parameters of the method in this paper according to the process shown in Figure 7. When the robot runs the sinusoidal trajectory for the first time, ρ and n can be determined after performing in Process 1. The suppression effect is shown in Figures 4 and 5. When the robot runs the sinusoidal trajectory for the second time, set the order range of the model $AR(u)$ to $u \in [1{,}40]$. After executing Process 2, not only N and u can be determined, but also FPE with different orders can be obtained, which are shown in Figure 11. Starting from the third time the robot runs the sinusoidal trajectory, the other parameters in this method except ρ, n, N and u can be determined after executing Process 3 in a loop. For γ, the universal value is 0.05 or 0.01 (Chan and Cryer, 2008). When repeatedly adjusting all the parameters that need to be set, the following two conditions need to be met. First, the method will not cause a false collision alarm. Second, the method can accurately and sensitively detect real collisions.

It can be seen from Figure 11 that starting from the model order $u = 12$, even if the model order increases, the reduction in FPE is no longer noticeable. Due to the need for selecting the smaller model order under small FPE, u is set to 12.

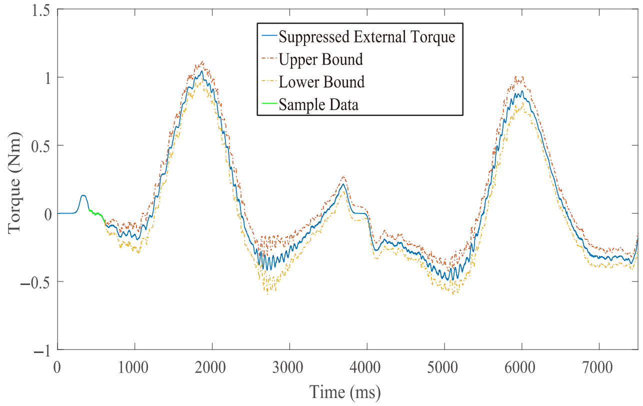
When the robot runs the sinusoidal trajectory, and no collision occurs, the original external torque of Joint 1 and the dynamic threshold generated by the collision detection method based only on TSA are shown in Figure 3. The suppressed external torque of Joint 1 and the dynamic threshold generated by the method in this paper are shown in Figure 12. When $t =$ 406 ms, the initialization condition shown in Step 2 of Process 3 in Figure 7 is met. Therefore, the external torque when $t =$ 406 ms–615 ms is intercepted to construct the initial sample data. Then, the dynamic thresholds generated by the method in this paper can tightly envelop the external torque, and no false collision alarm occurs during robot motion.

Figure 11 FPE with different orders



Source: Authors' own work

Figure 12 The suppressed external torque of Joint 1 and the dynamic threshold generated by the method in this paper when the robot runs the sinusoidal trajectory without collision



Source: Authors' own work

The final parameter values with repeated adjustments are listed in Tables 2 and 3. Method 1 is the collision detection method based only on TSA, and Method 2 is the method proposed in this paper. To further verify the versatility of the parameters, make the robot run the linear trajectory or the arc trajectory without collision. Multiple tests have found that neither the method in this paper nor the collision detection method based only on TSA has produced any false collision alarm.

4.4 Collision detection experiment

The robot runs the linear trajectory and bears an artificial hard collision when Joint 1 changes direction. The suppressed external torque of Joint 1, the dynamic threshold generated by the method in this paper, and the static threshold set by the symmetric threshold method are shown in Figure 13. Besides, the original external torque of Joint 1 and the dynamic threshold generated by the collision detection method based only on TSA are shown in Figure 14.

Figure 13 shows that the suppressed external torque undergoes a rapid increase starting approximately at $t = 5{,}983$ ms, breaking through the lower bound at $t = 6{,}002$ ms. By $t = 6{,}005$ ms, the method proposed in this paper detects that the lower bound has been exceeded for four consecutive time steps, indicating a collision occurrence. Thus, the detection delay of the method proposed in this paper is approximately 22 ms. However, the symmetric threshold method fails to detect this hard collision.

Table 2 The values of N, ρ, n and λ in methods 1 and 2

Parameters	N	ρ	n	λ
Method 1	210	–	–	0.999
Method 2	210	150	16	0.999

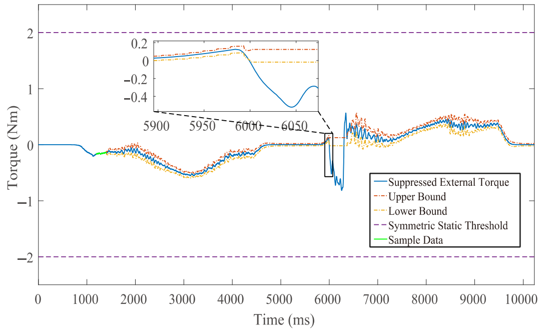
Source: Authors' own work

Table 3 The values of L, C, u, γ and λ in methods 1 and 2

Parameters	L	C	u	γ	ψ
Method 1	15	4	12	0.01	–
Method 2	15	4	12	0.01	0.02

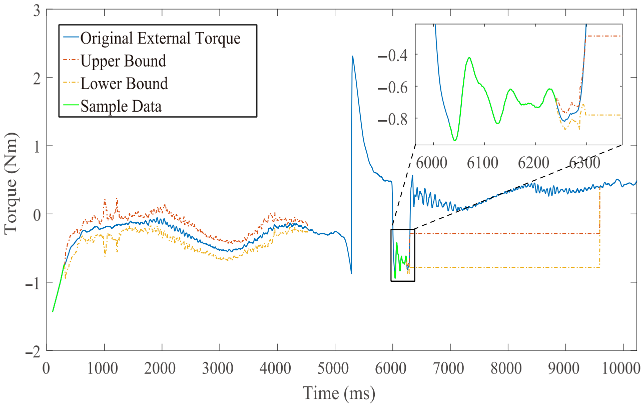
Source: Authors' own work

Figure 13 The suppressed external torque of Joint 1 and the dynamic threshold generated by the method in this paper when the robot runs the linear trajectory and bears a hard collision



Source: Authors' own work

Figure 14 The original external torque of Joint 1 and the dynamic threshold generated by the collision detection method based only on TSA when the robot runs the linear trajectory and bears a hard collision



Source: Authors' own work

As shown in Figure 14, when $t = 4{,}522$ ms–6,030 ms, $|\dot{q}| < 0.09\text{rad/s}$, the collision detection method based only on TSA stops generating dynamic thresholds. Starting from $t = 6{,}031$ ms, $|\dot{q}| \geq 0.09\text{rad/s}$, the external torque for $t = 6{,}031$ms–6,240 ms is intercepted to reinitialize the model coefficients and then the collision detection method based only on TSA continues to generate dynamic thresholds. The collision is detected at $t = 6{,}298$ ms, and the detection delay of the collision detection method based only on TSA is recorded to be approximately 315 ms.

Therefore, the method in this paper detects this hard collision about 296 ms faster than the collision detection method based only on TSA.

To obtain hard collision statistics, we made the robot run linear and arc trajectories multiple times and then applied 100 hard collisions when the robot ran each trajectory; that is, 200 hard collisions were applied. Then, we collected and analyzed all experiment data. In terms of accuracy, the number of detected collisions divided by the number of real collisions is calculated. In terms of average delay, the sum of collision detection delay divided by the number of successful detections is calculated. The statistical results are shown in Table 4.

When the robot runs the arc trajectory, it comes into contact with the foam board, as shown in Figure 8(b). At this time, the robot has a soft collision. The experiment data is shown in Figure 15.

From Figure 15, we know that the soft collision occurs at approximately $t = 2{,}645$ ms and the suppressed external torque breaks through the upper bound at $t = 2{,}680$ ms. After 4 ms, the exceeded upper bound leads to the collision alarm of the method in this paper at $t = 2{,}683$ ms. However, the symmetric threshold method detects this collision at $t = 2{,}732$ ms.

Obviously, in this collision detection, the delay of the method in this paper is about 38 ms, and the delay of the symmetric threshold method is about 87 ms. That is, the method in this paper detects the collision about 49 ms faster than the symmetric threshold method.

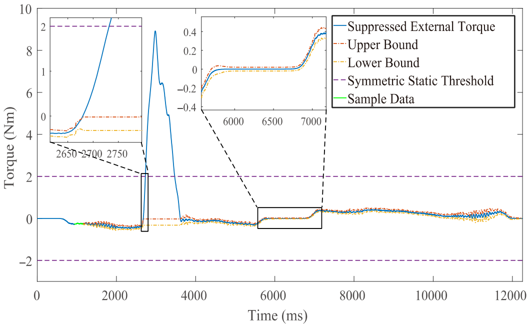
To obtain soft collision statistics, we made the robot run linear and arc trajectories multiple times and had 100 soft collisions with the foam board when running each trajectory. That is, 200 soft collisions occurred between the robot and the foam board. The statistical method of repeated experiments, used for hard collision cases, is also used to analyze the results of repeated experiments for soft collisions. The statistical results are shown in Table 5.

Table 4 The accuracy and average delay of the method in this paper and the symmetric threshold method when the robot bears hard collisions

Detection method	Accuracy (%)	Average delay (ms)
The method in this paper	97.5	10.0
The symmetric threshold method	43.5	71.3

Source: Authors' own work

Figure 15 The suppressed external torque of Joint 1 and the dynamic threshold generated by the method in this paper when the robot runs the arc trajectory and collides with the foam board



Source: Authors' own work

Table 5 The accuracy and average delay of the method in this paper and the symmetric threshold method when the robot collides with the foam board

Detection method	Accuracy (%)	Average delay (ms)
The method in this paper	81.0	33.4
The symmetric threshold method	100.0	84.0

Source: Authors' own work

5. Conclusion

The method proposed in this paper calculates the external torque using the current/torque data of the servo slave. Then, it performs mutation suppression processing on the external torque. Finally, it uses the TSA method to obtain several steps of predicted response and effectively prevents mutated external torques caused by joint reversals from false collision alarms. Compared with the collision detection method based only on TSA, its invalid time is less. If a collision occurs while the joint changes direction, it saves at least one time required to initialize the model coefficients. Three test trajectories were used to obtain the differences between this method and the symmetric threshold method. The comparison results show that it has higher accuracy and lower detection delay. In addition, it meets the real-time requirements of robot control systems and does not rely on external sensors. Therefore, it is so superior and practical that it can be used to improve the safety of human–machine collaborative production.

References

Box, G., Jenkins, G. and Reinsel, G. (2015), *Time Series Analysis: Forecasting and Control*, John Wiley & Sons, New York, NY.

Cao, P., Gan, Y. and Dai, X. (2019), "Model-based sensorless robot collision detection under model uncertainties with a fast dynamics identification", *International Journal of Advanced Robotic Systems*, Vol. 16 No. 3, pp. 1-15.

Chan, K. and Cryer, J. (2008), *Time Series Analysis with Applications in R*, Springer, New York, NY.

De Luca, A. and Mattone, R. (2005), "Sensorless robot collision detection and hybrid force/motion control", *IEEE international Conference on Robotics and Automation*, pp. 999-1004.

De Luca, A., Albu-Schaffer, A. and Haddadin, S. (2006), "Collision detection and safe reaction with the DLR-III lightweight manipulator arm", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1623-1630.

Dimeas, F., Avendaño-Valencia, L. and Aspragathos, N. (2015), "Human-robot collision detection and identification based on fuzzy and time series modelling", *Robotica*, Vol. 33 No. 9, pp. 1886-1898.

Duan, Z., Luo, Z. and Liu, Y. (2023), "A faster and more robust momentum observer for robot collision detection based on loop shaping techniques", *International Conference on Intelligent Robotics and Applications*, pp. 108-120.

Feng, K., Ji, J. and Ni, Q. (2023), "A review of vibration-based gear wear monitoring and prediction techniques", *Mechanical Systems and Signal Processing*, Vol. 182, p. 109605.

Haddadin, S. (2013), *Towards Safe Robots: Approaching Asimov's 1st Law*, Springer, Heidelberg.

Haddadin, S., De Luca, A. and Albu-Schäffer, A. (2017), "A survey on detection, isolation, and identification", *IEEE Transactions on Robotics*, Vol. 33 No. 6, pp. 1292-1312.

Heo, Y., Kim, D. and Lee, W. (2019), "Collision detection for industrial collaborative robots: a deep learning approach", *IEEE Robotics and Automation Letters*, Vol. 4 No. 2, pp. 740-746.

Indri, M., Trapani, S. and Lazzero, I. (2015), "A general procedure for collision detection between an industrial robot and the environment", IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), pp. 1-8.

Li, Y., Li, Y. and Zhu, M. (2021), "A nonlinear momentum observer for sensorless robot collision detection under model uncertainties", *Mechatronics*, Vol. 78

Lin, C., Hsieh, Y. and Cheng, F. (2019), "Time series prediction algorithm for intelligent predictive maintenance", *IEEE Robotics and Automation Letters*, Vol. 4 No. 3, pp. 2807-2814.

Liu, Y., Zhi, T. and Xi, H. (2019), "A novel content popularity prediction algorithm based on auto regressive model in information-centric IoT", *IEEE Access*, Vol. 7, pp. 27555-27564.

Liu, S., Wang, L. and Wang, X. (2021), "Sensorless haptic control for human-robot collaborative assembly", *CIRP Journal of Manufacturing Science and Technology*, Vol. 32 No. 138, pp. 132-144.

Park, K., Kim, J. and Park, J. (2020), "Learning-based real-time detection of robot collisions without joint torque sensors", *IEEE Robotics and Automation Letters*, Vol. 6 No. 1, pp. 103-110.

Ren, T., Dong, Y. and Wu, D. (2018), "Collision detection and identification for robot manipulators based on extended state observer", *Control Engineering Practice*, Vol. 79 No. 10, pp. 144-153.

Sharkawy, A. and Aspragathos, N. (2018), "Human-robot collision detection based on neural networks", *International Journal of Mechanical Engineering and Robotics Research*, Vol. 7 No. 2, pp. 150-157.

Song, Q., Mi, Y. and Lai, W. (2019), "A novel variable forgetting factor recursive least square algorithm to improve the anti-interference ability of battery model parameters identification", *IEEE Access*, Vol. 7, pp. 61548-61557.

Sotoudehnejad, V. and Kermani, M. (2014), "Velocity-based variable thresholds for improving collision detection in manipulators", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3364-3369.

Suita, K., Yamada, Y. and Tsuchida, N. (1995), "A failure-to-safety 'Kyozon' system with simple contact detection and stop capabilities for safe human-autonomous robot coexistence", *IEEE International Conference on Robotics and Automation*, pp. 3089-3096.

Wang, G., Wang, Z. and Huang, B. (2023), "Active compliance control based on EKF torque fusion for robot manipulators", *IEEE Robotics and Automation Letters*, Vol. 8 No. 5, pp. 2668-2675.

Zhang, T. and Hong, J. (2019), "Collision detection method for industrial robot based on envelope-like lines", *Industrial Robot: The International Journal of Robotics Research and Application*, Vol. 46 No. 4, pp. 510-517.

Zhang, T., Ge, P. and Zou, Y. (2021), "Robot collision detection without external sensors based on time-series analysis", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 143 No. 4, pp. 41005-41012.

Zhang, T., Chen, Y. and Ge, P. (2023), "LSTM-based external torque prediction for 6-DOF robot collision detection", *Journal of Mechanical Science and Technology*, Vol. 37 No. 9, pp. 4847-4855.

Appendix

When the robot runs the test trajectory, the target motion data of the first three joints are shown in Figures A1–A3.

From Figure A4, the FPE of the three joint timing models at different orders can be seen. For Joint 2, even if the model order continues to increase starting from the model order $u = 12$, the reduction in FPE is no longer noticeable. For Joint 1 or 3, when the model order $u \geq 12$, the FPE decreases as the order increases. The model order should be as small as possible to satisfy the small FPE, and it can make the method in this paper meet the requirements of high real-time and convenient programming.

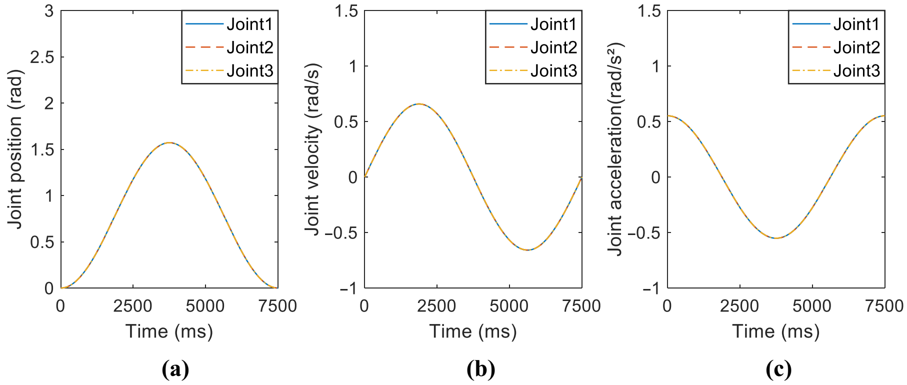
When the robot runs the sinusoidal trajectory, and no collision occurs, the suppressed external torque of Joints 1 to 3 and the dynamic threshold generated by the method in this paper are shown in Figures A5–A7.

The robot runs the linear trajectory and bears an artificial hard collision when Joint 1 changes direction. However, there was no significant change in the external torques of Joints 2 and 3. The original external torque of Joints 1 to 3 and the dynamic threshold generated by the collision detection method based only on TSA are shown in Figures A8–A10. Besides, the suppressed external torque of Joint 1–3 and the dynamic threshold generated by the method in this paper are shown in Figures A11–A13.

As shown in Figure A8, when t = 4,523 ms–6,024 ms, $|\dot{q}| < 0.09\text{rad/s}$, the collision detection method based only on TSA stops generating dynamic thresholds. Starting from t = 6,025 ms, $|\dot{q}| \geq 0.09\text{rad/s}$, the external torque for t = 6,025 ms–6,234 ms is intercepted to reinitialize the model coefficients and then the collision detection method based only on TSA continues to generate dynamic thresholds. However, the intercepted data is an abnormal external torque caused by collisions, so the predicted values obtained after reidentifying the model coefficients are also considered outliers. Finally, the generated threshold cannot be exceeded by the external torque caused by the collision.

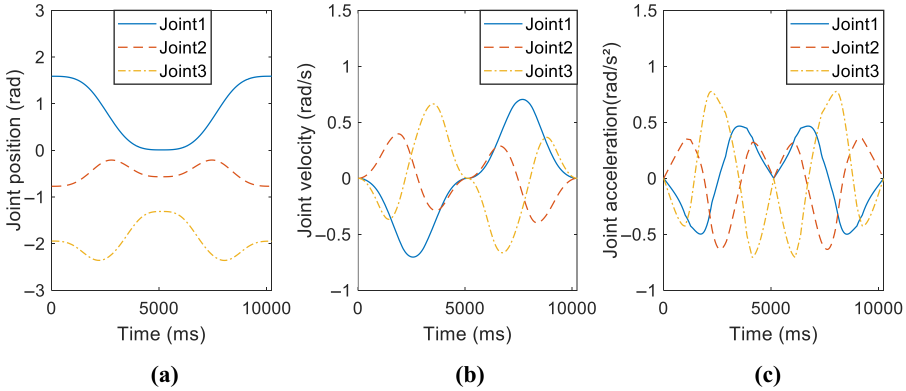
Figure A11 shows that the suppressed external torque rapidly increases approximately from t = 5,894 ms and exceeds the lower bound at t = 5,925 ms. When t = 5,928 ms, the method in this paper detects that the lower bound for four consecutive moments has been exceeded and determines that the collision occurred. However, the static threshold is exceeded by the suppressed external torque at t = 6,145 ms. Obviously, in this collision detection, the delay of the method in this paper is about 34 ms, and the delay of the symmetric

Figure A1 The target motion data when the robot runs the sinusoidal trajectory



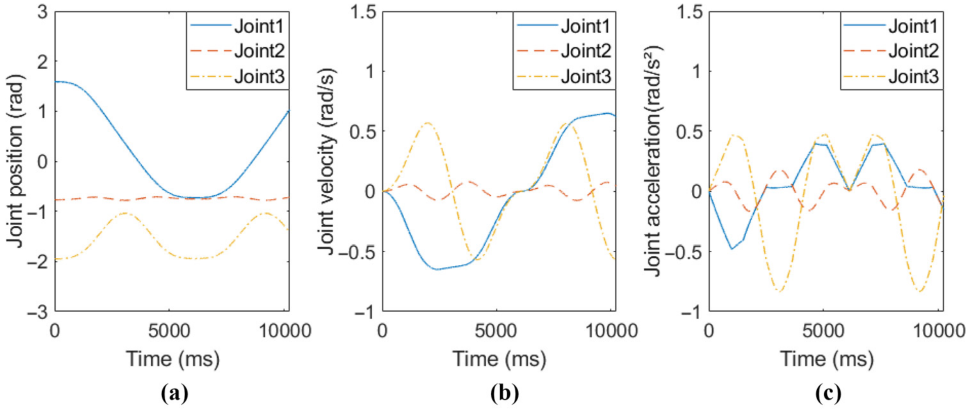
Notes: (a) Joint position; (b) joint velocity; (c) joint acceleration
Source: Authors' own work

Figure A2 The target motion data when the robot runs the sinusoidal trajectory



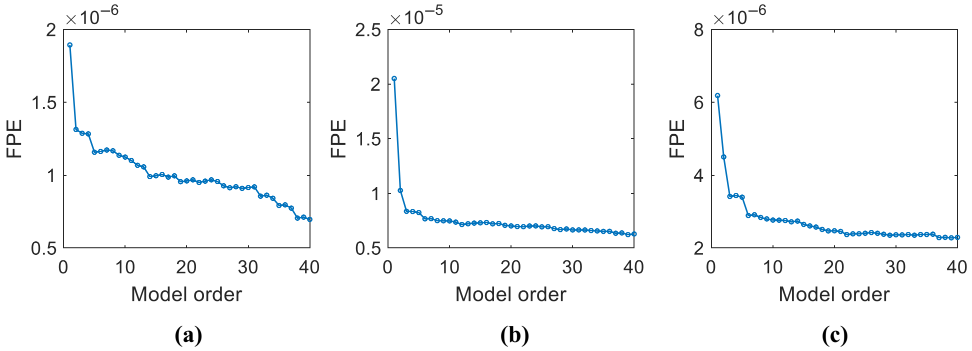
Notes: (a) Joint position; (b) joint velocity; (c) joint acceleration
Source: Authors' own work

Figure A3 The target motion data when the robot runs the linear trajectory



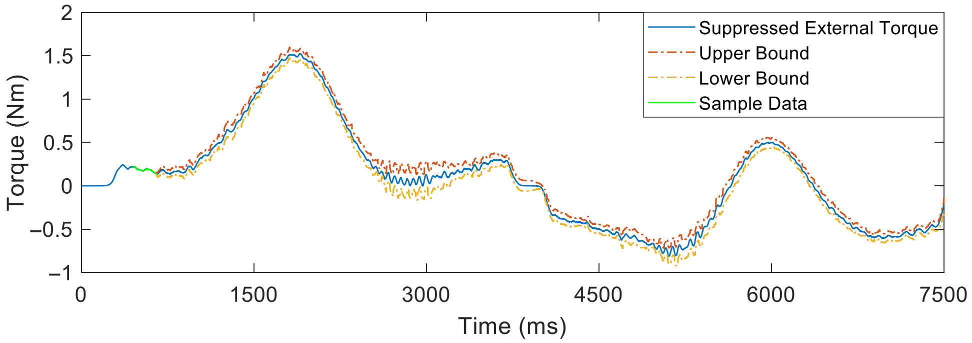
Notes: (a) Joint position; (b) joint velocity; (c) joint acceleration
Source: Authors'own work

Figure A4 The target motion data when the robot runs the arc trajectory



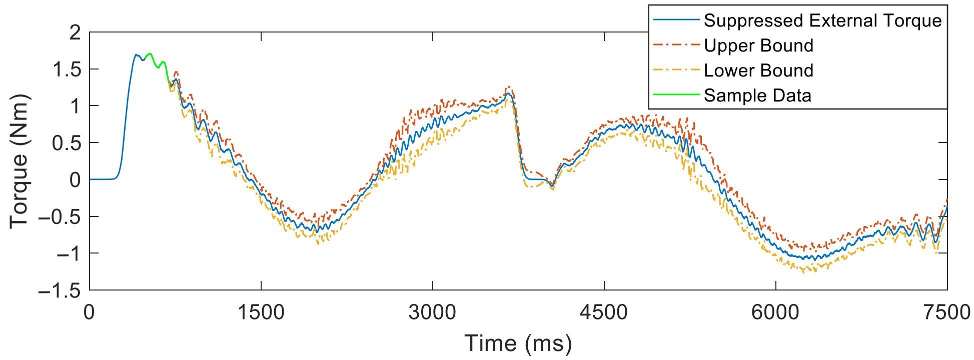
Notes: (a) The FPE of Joint 1's timing model; (b) the FPE of Joint 2's timing model; (c) the FPE of Joint 3's timing model
Source: Authors' own work

Figure A5 FPE with different orders



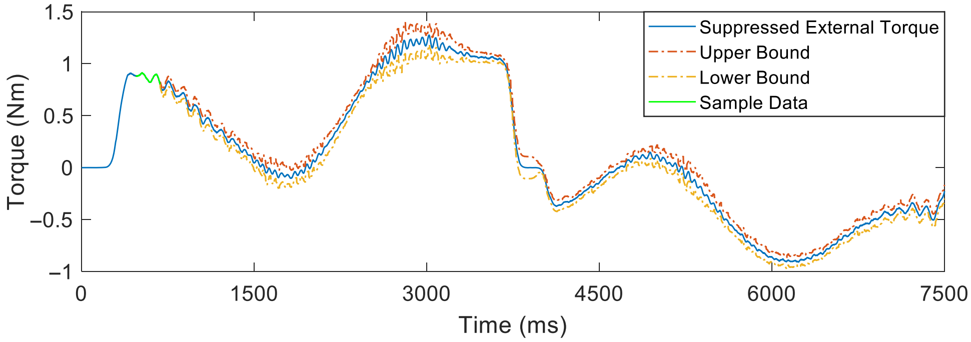
Source: Authors' own work

Figure A6 The suppressed external torque of Joint 1 and the dynamic threshold generated by the method in this paper when the robot runs the sinusoidal trajectory without collision



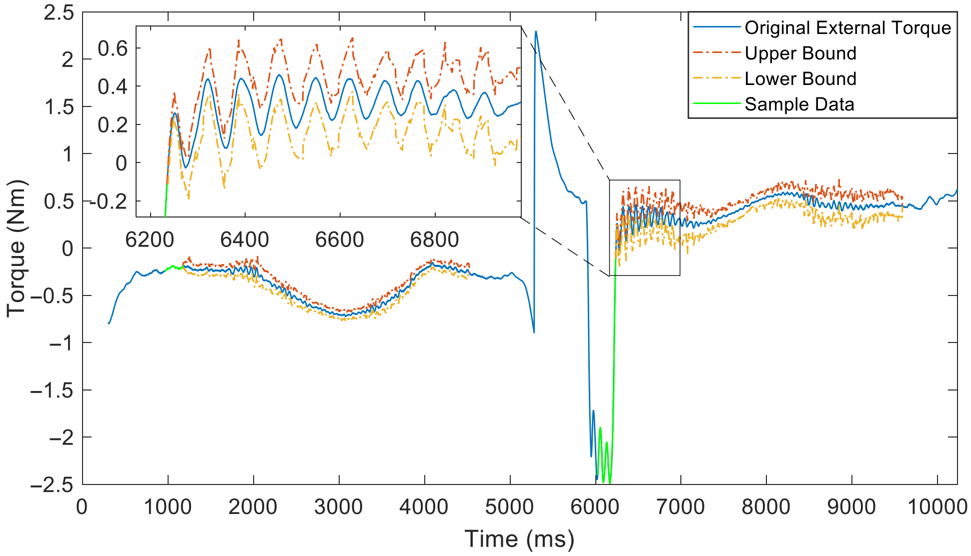
Source: Authors' own work

Figure A7 The suppressed external torque of Joint 2 and the dynamic threshold generated by the method in this paper when the robot runs the sinusoidal trajectory without collision



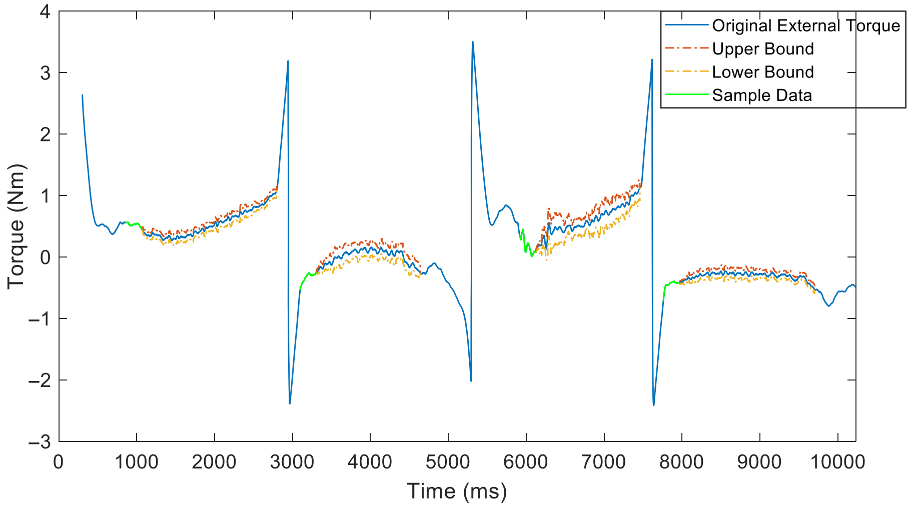
Source: Authors’ own work

Figure A8 The suppressed external torque of Joint 3 and the dynamic threshold generated by the method in this paper when the robot runs the sinusoidal trajectory without collision



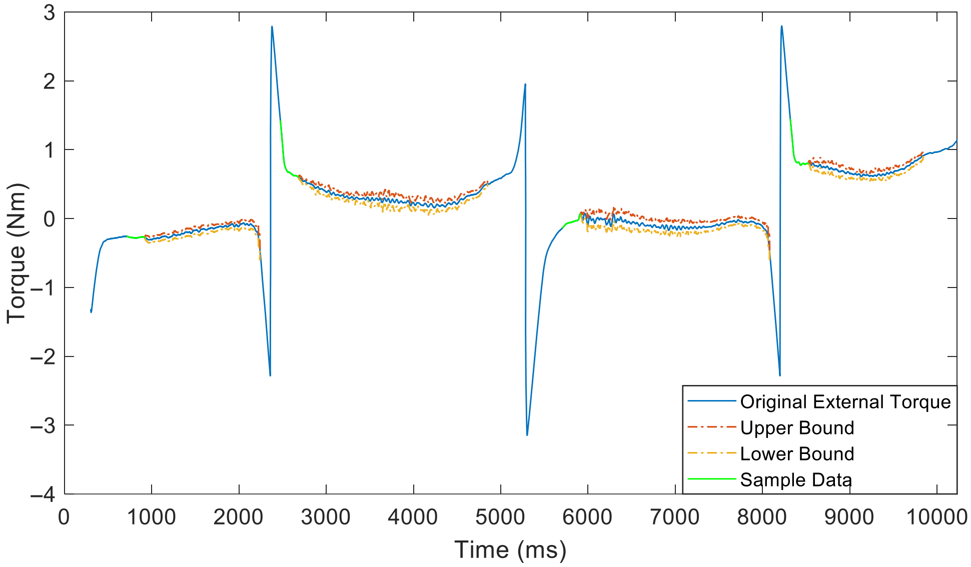
Source: Authors’ own work

Figure A9 The original external torque of Joint 1 and the dynamic threshold generated by the collision detection method based only on TSA when the robot runs the linear trajectory and bears a hard collision



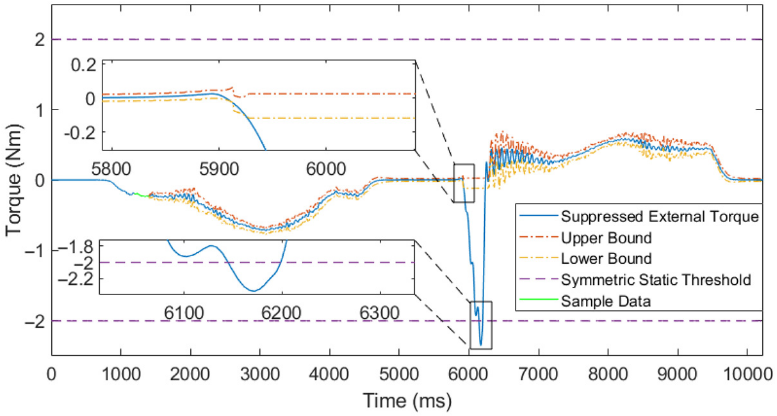
Source: Authors' own work

Figure A10 The original external torque of Joint 2 and the dynamic threshold generated by the collision detection method based only on TSA when the robot runs the linear trajectory and bears a hard collision



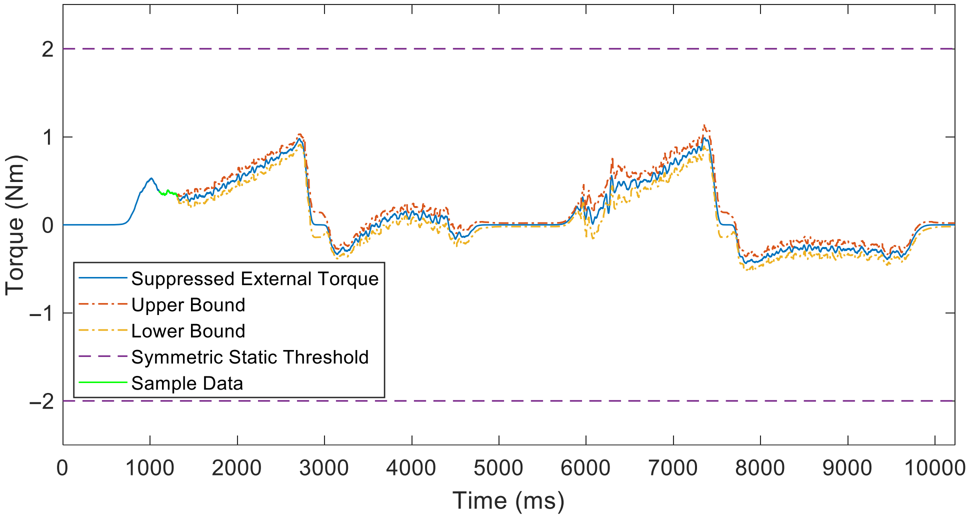
Source: Authors' own work

Figure A11 The original external torque of Joint 3 and the dynamic threshold generated by the collision detection method based only on TSA when the robot runs the linear trajectory and bears a hard collision



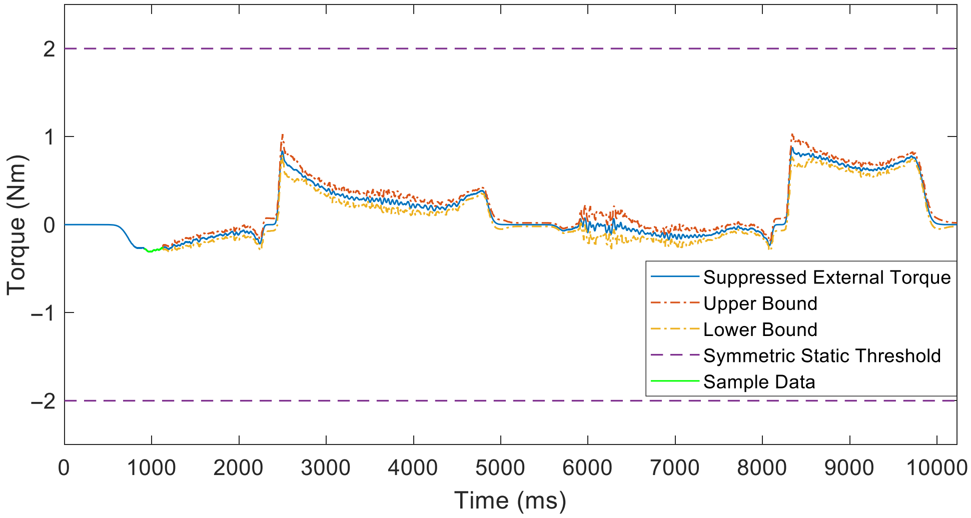
Source: Authors'own work

Figure A12 The suppressed external torque of Joint 1 and the dynamic threshold generated by the method in this paper when the robot runs the linear trajectory and bears a hard collision



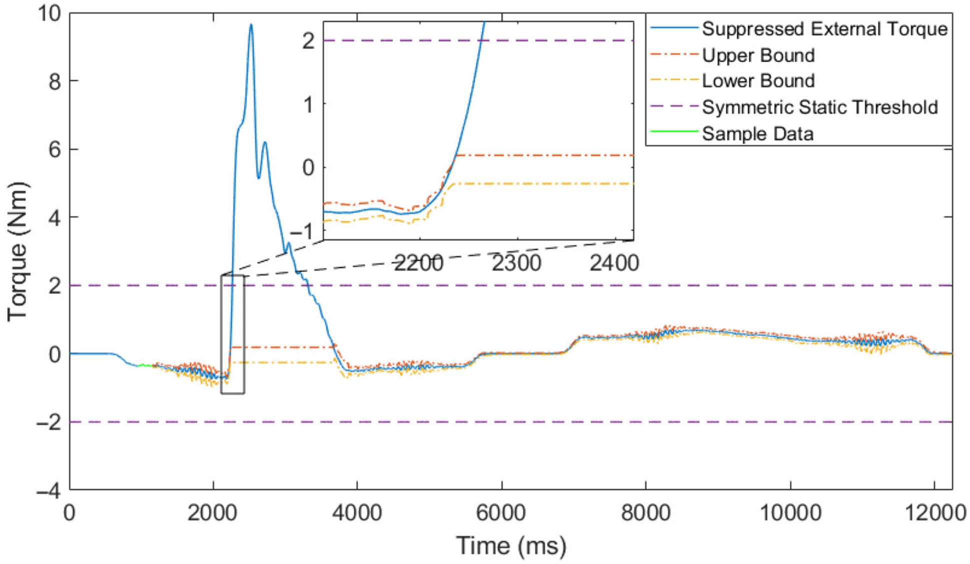
Source: Authors' own work

Figure A13 The suppressed external torque of Joint 2 and the dynamic threshold generated by the method in this paper when the robot runs the linear trajectory and bears a hard collision



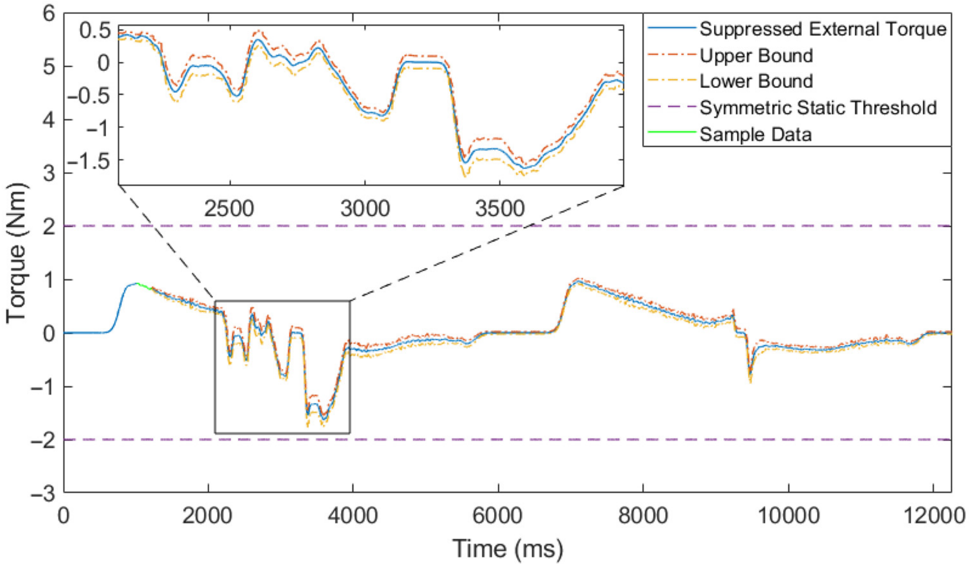
Source: Authors' own work

Figure A14 The suppressed external torque of Joint 3 and the dynamic threshold generated by the method in this paper when the robot runs the linear trajectory and bears a hard collision



Source: Authors'own work

Figure A15 The suppressed external torque of Joint 1 and the dynamic threshold generated by the method in this paper when the robot runs the arc trajectory and collides with the foam board



Source: Authors'own work

threshold method is about 251 ms. This information indicates that the method in this paper detects the collision about 217 ms faster than the symmetric threshold method.

When the robot runs the arc trajectory, it has a soft collision with the foam board, as shown in Figure 8(b). Due to the small velocity of Joint 2, the initialization condition shown in Step (2) of Process 3 in Figure 7 has yet to be satisfied. Therefore, the dynamic threshold related to the suppressed external torque of Joint 2 has not been generated. For Joints 1 and 3, the suppressed external torque and related thresholds are shown in Figures A14 and A15, respectively.

From Figure A15, it can be seen that the suppressed external torque of Joint 3 does not exceed the dynamic threshold generated by the method in this paper and the static threshold set by the symmetric static threshold method. However, Figure A14 shows that the suppressed external torque of Joint 1 exceeds the above two thresholds. For Joint 1, the following details can be obtained from Figure A14. The suppressed external torque slowly increases approximately from $t = 2{,}200$ ms and exceeds the upper bound at $t = 2{,}235$ ms. After 4 ms, the exceeded upper bound leads to the collision alarm of the method in this paper at $t = 2{,}238$ ms. In addition, the static threshold is exceeded by the suppressed external torque at $t = 2{,}264$ ms. Therefore, in this collision detection, the delay of the method in this paper is about 38 ms and the delay of the symmetric threshold method is about 64 ms. That is, the method in this paper detects the collision about 26 ms faster than the symmetric threshold method.

Corresponding author

Gao Wang can be contacted at: twangg@jnu.edu.cn