Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 7974325, 19 pages
https://doi.org/10.1155/2018/7974325

Research Article

Multiobjective Trajectory Optimization and Adaptive Backstepping Control for Rubber Unstacking Robot Based on RFWNN Method

Le Liang, Yanjie Liu, and Hao Xu

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150080, China

Correspondence should be addressed to Yanjie Liu; yjliu@hit.edu.cn

Received 8 August 2017; Revised 25 November 2017; Accepted 7 December 2017; Published 2 January 2018

Academic Editor: Paolo Boscariol

Multiobjective trajectory optimization and adaptive backstepping control method based on recursive fuzzy wavelet neural network (RFWNN) are proposed to solve the problem of dynamic modeling uncertainties and strong external disturbance of the rubber unstacking robot during recycling process. First, according to the rubber viscoelastic properties, the Hunt-Crossley nonlinear model is used to construct the robot dynamics model. Then, combined with the dynamic model and the recycling process characteristics, the multiobjective trajectory optimization of the rubber unstacking robot is carried out for the operational efficiency, the running trajectory smoothness, and the energy consumption. Based on the trajectory optimization results, the adaptive backstepping control method based on RFWNN is adopted. The RFWNN method is applied in the main controller to cope with time-varying uncertainties of the robot dynamic system. Simultaneously, an adaptive robust control law is developed to eliminate inevitable approximation errors and unknown disturbances and relax the requirement for prior knowledge of the controlled system. Finally, the validity of the proposed control strategy is verified by experiment.

1. Introduction

Countless rubber in the tire manufacturing and other industries has been applied, such as natural rubber and synthetic rubber. Due to the need for a large number of manual participation and taking into account the labor intensity and low efficiency, it is urgent to use robotic technology instead of artificial one to complete the rubber unstacking process. As the heat refining of the rubber block needs to be operated before the mixer processing, the rubber block will have a certain degree of melting and resulting in uneven surface of the rubber block which leads to the fact that adsorption type fixture cannot be used. Therefore, the way that the robot end effector directly is inserted into the rubber block is utilized to complete the unstacking process. During this process, the robot will be in sharp contact with the rubber, time-varying contact force will seriously interfere with the normal operation of the robot, and conventional trajectory planning and control methods are difficult to meet the system performance requirements.

Efficiency, trajectory smoothness, and energy consumption are important performance indicators of rubber unstacking robot in trajectory planning. Particularly during recycling process, due to the viscoelastic characteristics of rubber, the faster the insertion velocity of the end effector, the greater the reaction force. According to the characteristics of robot operation, performance requirements, and corresponding constraints, multiobjective optimization model is established to obtain the optimal trajectory of the rubber unstacking robot during recycling process. Many researches have been done in this area [1–3]. Piazzi and Visioli developed an approach based on interval analysis to find the global minimum-jerk (MJ) trajectory of a robot manipulator within a joint space scheme using cubic splines [4]. Chettibi et al. discussed the problem of minimum cost trajectory planning for robotic manipulators, which put a few free via points uniformly distributed in time [5]. Korayem et al. converted dynamic load-carrying capacity problem into a trajectory optimization problem of cable-suspended parallel robots which is fundamentally a constrained nonlinear optimization problem [6]. Abu-Dakka et al. addressed an indirect method for trajectory planning for industrial robots operating in workspaces with obstacles using an evolutionary algorithm [7].

In the aspect of control strategy, due to the influence of time-varying nonlinear contact force on the robot during the process of inserting the rubber, the modeling error caused by the stiffness of the robot bar, the friction of the joint, and the external disturbance are highlighted, which seriously affect the robot's optimal trajectory tracking accuracy and control performance. The adaptive robust controllers based on hybrid neural networks can exhibit good properties as an attempt to cope with the problem of modeling uncertainty parameters and external disturbances. The neural networks can deal with the unknown dynamics problems of the rubber unstacking robot control system by exploiting their universal approximation ability [8–10]. At the same time, the approximate error of the neural network and the interference of the external system are estimated and compensated by adding the adaptive robust term in the controller [11–13]. Chairez applied differential neural network approach with activation functions described by wavelets in a state observation problem when the dynamic model of a plant contains uncertainties [14]. Yoo et al. proposed a method for the robust control of flexible-joint robots with model uncertainties which combined the adaptive dynamic surface control technique and the self-recurrent wavelet neural network [15]. Boukattaya et al. studied the trajectory tracking control problem of mobile manipulators subject to nonholonomic constraints, operating in task space, with the presence of external torque disturbances and dynamic uncertainties [16].

The main objective of this work is related to multiobjective trajectory optimization and adaptive robust trajectory tracking control of the rubber unstacking robot during the recycling process. First, according to the nonlinearity, viscoelasticity, and anisotropy of the rubber, the Hunt-Crossley foundation model was used to establish the mechanical model while the end effector is inserted into the rubber block. Then, aimed at the performance indexes such as operation efficiency, running trajectory smoothness, and energy consumption, the problem of motion planning is transformed into multiobjective optimization problem. The position of the floating via points for the B-spline trajectory and the running time between adjacent points are optimized, so the optimal trajectory in joint space is obtained. Finally, a recursive fuzzy wavelet neural network is proposed to estimate the uncertainty model of the system by combining fuzzy and wavelet neural networks. At the same time, adaptive backstepping control method is adopted to eliminate the influence of uncertainties such as estimation error and external disturbance on the tracking accuracy of the system trajectory. The control performance is verified by simulation analysis.

2. Robot System and Dynamic Model

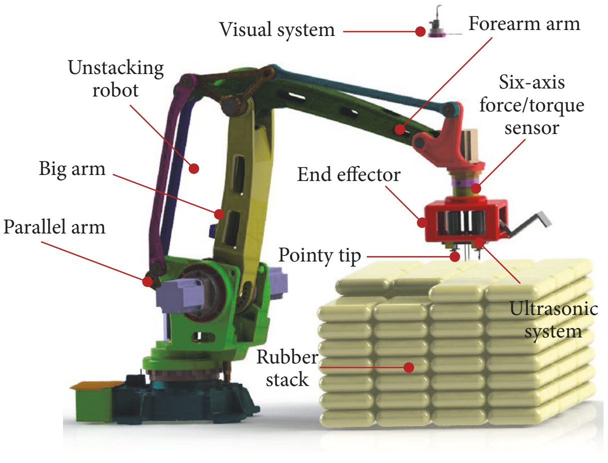


Figure 1: Composition of the robot unstacking system.

Table 1: Definition of main structural physical parameters.

Physical meaning	Symbol	Value
Waist joint angle/°	θ_1	−165~165
Big arm joint angle/°	θ_2	5~130
Forearm joint angle/°	θ_3	60~200
Length of big arm/mm	l_1	945
Length of forearm/mm	l_2	1025
Length of parallel arm/mm	l_3	400

The robot unstacking system is shown in Figure 1 which includes unstacking robot, visual system, ultrasonic system, six-axis force/torque sensor, and end effector with pointy tip. The dynamic model of robot unstacking process is shown as follows:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) + J^{T}(\theta)F_s = \tau, \tag{1}$$

where θ, $\dot{\theta}$, and $\ddot{\theta}$ are the robot joint angle, angular velocity, and angular acceleration, respectively, $M(\theta)$ is the inertia matrix, $C(\theta,\dot{\theta})$ is Coriolis force item and centrifugal force item, $G(\theta)$ is gravity item, τ is the generalized input torque, $J(\theta)$ is Jacobian matrix for the conversion of joint space to Cartesian space, and F_s represents the external force in Cartesian space when the robot is inserted into the rubber. Table 1 shows the definition of main structural physical parameters of the unstacking robot.

As can be seen from (1), $M(\theta)$, $C(\theta,\dot{\theta})$, $G(\theta)$, and $J(\theta)$ can be calculated according to the robot kinematics and dynamics; however, it is difficult to describe F_s with a simple linear mechanics model because the rubber has the characteristics of nonlinearity, viscoelasticity, and anisotropy. To obtain a more accurate viscoelastic model, Hunt and Crossley [17] made a nonlinear combination of spring and damping systems, and a nonlinear model was established as follows:

$$F_e(t) = Kx^{\beta}(t) + \lambda x^{\beta}\dot{x}(t), \tag{2}$$

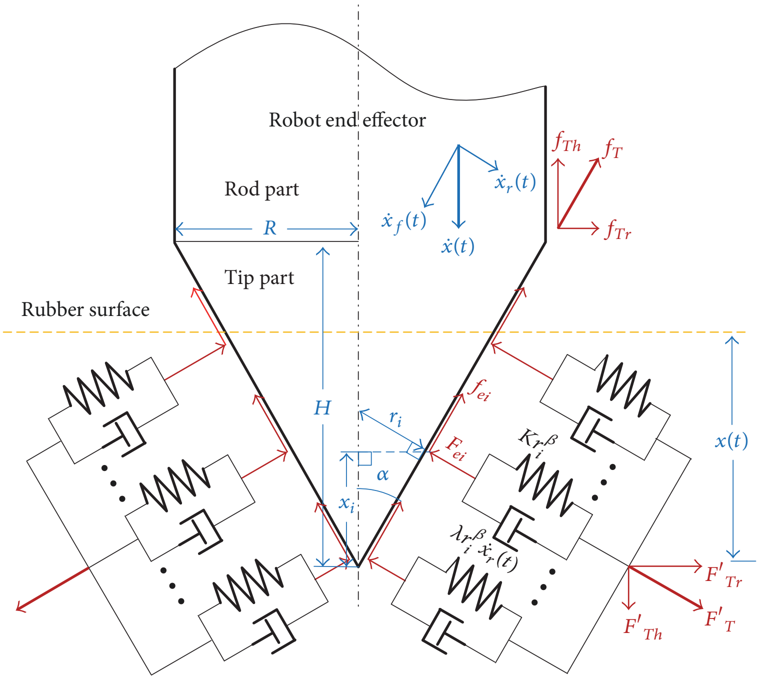


Figure 2: Mechanical model of the tip insertion stage.

where K and λ are the elastic coefficient and the nonlinear damping coefficient, respectively, $x(t)$ is the displacement inserted in the rubber, $\dot{x}(t)$ is insert speed, and β is the positive scalar.

The mechanical model of the robot end effector insertion rubber block process was established on the basis of the selected HC nonlinear basic model and combined with the mechanical characteristics of rubber block. The insertion process is divided into two stages, namely, tip part insertion stage and rod part insertion stage. Figure 2 showed the former mechanical model. Where R is the maximum bottom radius of the tip cone, H is the tip cone height, α is the tip cone angle, $\dot{x}_f(t)$, $\dot{x}_r(t)$ are the velocity component parallel and perpendicular to the direction of the conical bus, respectively, f_T is the tip friction of robot end effector, f_{Th} and f_{Tr} are, respectively, the axial and radial frictional forces, F_T is tip positive pressure of the end effector (F'_T is the reaction force), f_{Th} and f_{Tr} are, respectively, the axial and radial positive pressure components, and μ is the friction coefficient between the rubber and actuator.

According to Figure 2 and HC nonlinear viscoelastic model, the mechanical model equations of the tip part insertion stage ($x(t) \le H$) are as follows:

$$\begin{gathered}
r_i = \frac{x_i \tan\alpha}{\cos\alpha}, \\
\dot{x}_r(t) = \dot{x}(t)\sin\alpha, \\
F_{ei} = Kr_i^{\beta} + \lambda r_i^{\beta}\dot{x}_r(t), \\
F_T = \int_0^{x(t)} F_{ei}\frac{2\pi x_i \tan\alpha}{\cos\alpha}dx_i \\
= \frac{2\pi}{\beta+2}\left(K + \lambda\dot{x}(t)\sin\alpha\right)\left(\frac{\tan\alpha}{\cos\alpha}\right)^{\beta+1} x(t)^{\beta+2}, \\
f_T = \mu F_T, \\
F_s(t) = f_T\cos\alpha + F_T\sin\alpha.
\end{gathered} \tag{3}$$

Figure 3 showed the mechanical model of the rod part insertion stage, in which f_R is the rod friction of robot end effector, and F_R is the rod positive pressure of the end effector (F'_R is the reaction force).

According to Figure 3 and HC nonlinear viscoelastic model, the mechanical model equations of the tip insertion stage ($x(t) > H$) are as follows:

$$\begin{gathered}
F_T = \frac{2\pi}{\beta+2}\left(K + \lambda\dot{x}(t)\sin\alpha\right)\left(\frac{\tan\alpha}{\cos\alpha}\right)^{\beta+1} H^{\beta+2}, \\
f_T = \mu F_T, \\
F_{Ri} = KR^{\beta}, \\
F_R = \int_0^{x(t)-H} F_{Ri}2\pi R dx_i = 2\pi KR^{\beta+1}\left(x(t) - H\right), \\
f_R = \mu F_R, \\
F_s(t) = f_T\cos\alpha + F_T\sin\alpha + f_R.
\end{gathered} \tag{4}$$

The mechanical model of the robot end effector inserted into the rubber process can be obtained after solving (3) and (4):

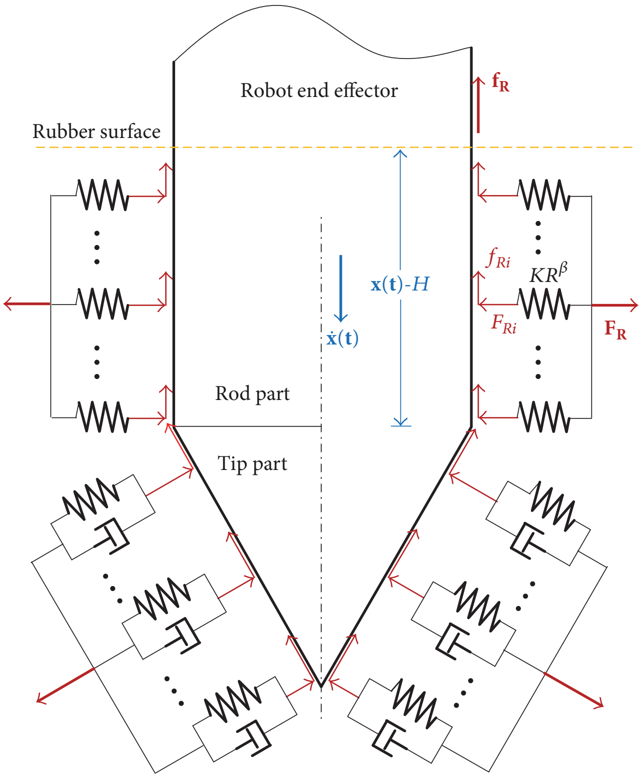


Figure 3: Mechanical model of the rod insertion stage.

$$F_s(t)=\begin{cases}\dfrac{2\pi}{\beta+2}\left(\dfrac{\tan\alpha}{\cos\alpha}\right)^{\beta+1}(\mu\cos\alpha+\sin\alpha)\cdot(K+\lambda\sin\alpha\dot{x}(t))\,x(t)^{\beta+2}, & (x(t)\le H);\\ \dfrac{2\pi}{\beta+2}\left(\dfrac{\tan\alpha}{\cos\alpha}\right)^{\beta+1}(\mu\cos\alpha+\sin\alpha)\cdot(K+\lambda\sin\alpha\dot{x}(t))\,H^{\beta+2}+2\pi\mu KR^{\beta+1}(x(t)-H), & (x(t)>H),\end{cases}\tag{5}$$

where α, H, R, and μ are known quantities and their values are $\alpha = 20^{\circ}$, $H = 48\,\text{mm}$, $R = 17.5\,\text{mm}$, and $\mu = 0.45$, $x(t)$ and $\dot{x}(t)$ are input variables, K, β, and λ can be obtained by experimental parameter identification, and their values are $K = 16.32\,\text{KN/m}$, $\beta = 1.15$, and $\lambda = 7.36\,\text{KN}*\text{s/m}^2$.

3. Multiobjective Trajectory Optimization Algorithm

3.1. B-Spline Trajectory Planning in Joint Space. Robot joint space trajectory planning is based on a series of key constraint points to plan robot motion trajectory which are $\theta_{mi} = \{\theta_{mi}(t_{i+1}),\ i = 0,1,\ldots,n,\ j = 1,2,\ldots,M\}$, where $\theta_{mi}(t)$ represents the ith trajectory segment of joint m. θ_{mi} is the angle at the $i+1$ via point of joint m that is the angle at t_{i+1}, and M is the number of robot joints. The number of constraint points for any joint trajectory curve is $n+1$, which are starting point, end point, and $n-1$ via points, respectively.

The unstacking robot joint space trajectory is described using the nonuniform B-spline curve. Based on determining the order k and the node vector $U = [u_0, u_1, \ldots, u_{n+k+1}]$, B-spline curve recursive equation using the de Boer algorithm is achieved as follows [18]:

$$\begin{gathered}p(u)=\sum_{j=i-k+1}^{i}d_j^lN_{j,k-1}(u)=\cdots=d_i^k,\quad u_i\le u<u_{i+1},\\ d_j^l\\ =\begin{cases}d_j, & l=0\\ \left(1-\alpha_j^l\right)d_{j-1}^{l-1}+\alpha_j^ld_j^{l-1}, & l=1,2,\ldots,k;\ j=i-k+l,\ldots,i,\end{cases}\\ \alpha_j^l=\frac{u-u_j}{u_{j+k+1-l}-u_j},\end{gathered}\tag{6}$$

where $d_i \in \mathbf{R}^{M\times1}$ is the control vertices of the joint spline trajectory, $u \in [u_i, u_{i+1}]$ is the normalized time node vector of spline curve in segment i, and $N_{i,k}(u)$ is k-order specification B-spline basis function. According to the local

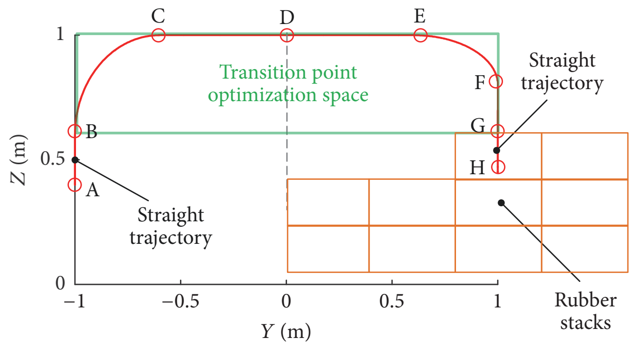


FIGURE 4: Recycling process trajectory before optimization.

support character of the B-spline curve, the r-order derivative of $p(u)$ is obtained

$$p^r(u) = \sum_{j=i-k+r}^{i} d_j^r N_{j,k-r}(u), \quad u_i \le u < u_{i+1},$$

$$d_j^l = \begin{cases} d_j, & l = 0 \\ (k+1-l)\dfrac{d_j^{l-1} - d_{j-1}^{l-1}}{u_{j+k+1-l} - u_j}, & l = 1, 2, \ldots, r;\; j = i-k+l, \ldots, i. \end{cases} \tag{7}$$

According to (7), robot joint space velocity, acceleration, and jerk can be obtained as follows:

$$\begin{aligned} \dot{\theta}_{mi}(t) &= p'(u) = \sum_{j=i-k+1}^{i} d_j^1 N_{j,k-1}(u), \\ \ddot{\theta}_{mi}(t) &= p''(u) = \sum_{j=i-k+2}^{i} d_j^2 N_{j,k-2}(u), \\ \dddot{\theta}_{mi}(t) &= p'''(u) = \sum_{j=i-k+3}^{i} d_j^3 N_{j,k-3}(u). \end{aligned} \tag{8}$$

3.2. Optimization Model. During the robot recycling process, the trajectories of each joint are designed as segment B-spline curves. By using the same time interval parameter h_i to couple the joint tracks to each other, where $\theta_{mi}(t)$ represents the ith trajectory of joint m, θ_{mi}, $\dot{\theta}_{mi}$, and $\ddot{\theta}_{mi}$ are, respectively, the angle, angle velocity, and acceleration at the i via point of joint m. The main constraints of the optimization process are set as follows.

(1) Position Constraint. According to the recycling characteristics, the rubber unstacking robot usually adopts the "door-shaped" trajectory to operate as shown in Figure 4. In order to avoid interference, the AB, CE, and GE segments are set as linear trajectories. In some former researches [1–7], the trajectory is usually optimized according to a number of fixed teaching points and the way of inserting fixed via points. The optimization method overreliance on the selection of the initial trajectory is mainly to optimize the segmentation trajectory between the fixed via points. However, since all the via points have been set in advance, the optimization range of the Cartesian space trajectory is limited and it is difficult to obtain the optimal trajectory.

The trajectory optimization method proposed in this paper contains two movement forms: linear trajectories AB and GH which are limited by obstacle avoidance and recycling form; the free trajectory passes through the barrier point B, the highest point D, and the insertion point G which are optimized by the rectangular space composed of B, D, and G points; meanwhile, to avoid interference, it is necessary to ensure that the robot end effector is always moving in the space. The positional coordinates of the end effector in the Cartesian space along the y- and z-axes can be solved by positive kinematics. It is only necessary to verify whether the limit position of each trajectory satisfies the constraint condition as compared with the coordinate value of the whole motion time period. When the velocity of the rubber unstacking robot along y- and z-axes is zero, the corresponding position is the limit position of the B-spline segment, which can be calculated as follows:

$$\begin{aligned} \dot{y}_i(t) &= \cos\theta_{1i}(t)\left(l_1\cos\theta_{2i}(t) - l_3\cos\theta_{3i}(t) + S_1\right)\dot{\theta}_{1i}(t) \\ &\quad - l_1\sin\theta_{1i}(t)\sin\theta_{2i}(t)\,\dot{\theta}_{2i}(t) \\ &\quad + l_3\sin\theta_{1i}(t)\sin\theta_{3i}(t)\,\dot{\theta}_{3i}(t), \\ \dot{z}_i(t) &= l_1\cos\theta_{2i}(t)\,\dot{\theta}_{2i}(t) - l_3\cos\theta_{3i}(t)\,\dot{\theta}_{3i}(t). \end{aligned} \tag{9}$$

The limit position of $y_i{}^*$ and $z_i{}^*$ of the spline trajectory of the ith segment is obtained by the above equations, and the spatial position constraints are as follows:

$$\begin{aligned} y^{\mathrm{MIN}} &\le y_i{}^* \le y^{\mathrm{MAX}}, \quad i = b, \ldots, g, \\ z^{\mathrm{MIN}} &\le z_i{}^* \le z^{\mathrm{MAX}}, \quad i = b, \ldots, g, \end{aligned} \tag{10}$$

where b and g are the spline segments of the transition point B and G, respectively. Since the linear trajectory segments AB and GE are in the z-axis direction, the position constraint is as follows:

$$\left| y_i{}^* - y^{\mathrm{STRAIGHT}} \right| \le \delta^{\mathrm{MAX}}, \quad i = 1, \ldots, b \;\&\; g, \ldots, n. \tag{11}$$

(2) Velocity, Acceleration, and Jerk Constraints. Under the condition of the drive system which has been set, the joint velocity and acceleration of the rubber unstacking robot are constrained to a certain extreme range and, at the same time, due to excessive acceleration, can cause the robot mechanical system resonance and increase the wear of the mechanical system, so the jerk is also used as one of the constraints. Any

section trajectory of any joint should satisfy the constraint conditions as follows:

$$\begin{gathered}\dot{\theta}_m^{\text{LB}} \le \dot{\theta}_{mi}(t) \le \dot{\theta}_m^{\text{UB}},\\ \ddot{\theta}_m^{\text{LB}} \le \ddot{\theta}_{mi}(t) \le \ddot{\theta}_m^{\text{UB}},\\ \left|\dddot{\theta}_{mi}(t)\right| \le J^{\text{MAX}},\\ i = 0,\dots,n,\ \ j = 1,\dots,m,\end{gathered} \tag{12}$$

where $\dot{\theta}_m^{\text{LB}}$, $\dot{\theta}_m^{\text{UB}}$, $\ddot{\theta}_m^{\text{LB}}$, and $\ddot{\theta}_m^{\text{UB}}$ are, respectively, the velocity and acceleration limit value of joint m and J^{MAX} is the maximum jerk. In the process of optimization, the constraint will be transformed into a constraint on the first three derivatives of the B-spline control vertex $\mathbf{d}_m$ as shown in (7).

(3) Torque Constraints. The joint output torque is limited by the robot drive system and needs to meet the corresponding constraints. According to (1) one can get

$$|\tau| = \left|M(\theta)\ddot{\theta} + C\left(\theta,\dot{\theta}\right)\dot{\theta} + G(\theta) + J^T(\theta)F_s\right| \le \tau^{\text{UB}}, \tag{13}$$

where τ^{UB} is the output torque limit, which is defined as the 80% of the maximum torque of the joint motor.

(4) Normalized Torque Constraints. In order to ensure the smoothness of the robot motion, the normalized torque is taken as the constraint condition as follows:

$$\begin{aligned}&\sum_{m=1}^{M} \frac{\sqrt{(1/H)\int_0^H \left(\tau_{mi}(t)\right)^2 dt}}{\sqrt{(1/N)\sum_{k=1}^{N}\left(\tau_{mi}(k\Delta T) - \tau_{mi}((k-1)\Delta T)\right)^2}}\\ &\quad \le M^{\text{UP}},\end{aligned} \tag{14}$$

where N is the number of servo control cycles, ΔT is the robot servo cycles, and M^{UP} is the maximum value.

In order to improve the working efficiency, take the work time of the rubber unstacking robot during recycling process as the optimization goal. At the same time, taking into account the trajectory smoothness and energy consumption, the motion planning problem is transformed into multiobjective optimization problem. Select the running time (h_i) between the via points and the joint angle θ_{mi} at the floating via points is used as an optimization variable. The optimization model is established as shown in (15), where $\mathbf{X}$ is the optimized variables, S_J is the joint average acceleration which is used to measure the smoothness of the robot motion trajectory, and S_q is the joint total energy consumption. The relevant parameters of the optimization model are set as follows: $y^{\text{MIN}} = -1\,\text{m}$, $y^{\text{MAX}} = 1\,\text{m}$, $z^{\text{MIN}} = 0.6\,\text{m}$, $z^{\text{MAX}} = 1\,\text{m}$, $\delta^{\text{MAX}} = 1\,\text{mm}$, $\dot{\theta}_m^{\text{LB}} = -30°/\text{s}$, $\dot{\theta}_m^{\text{UB}} = 30°/\text{s}$, $\ddot{\theta}_m^{\text{LB}} = -150°/\text{s}^2$, $\ddot{\theta}_m^{\text{UB}} = 150°/\text{s}^2$, $J^{\text{MAX}} = 500°/\text{s}^3$, $\tau^{\text{UB}} = 8000\,\text{Nm}$, and $M^{\text{UP}} = 100$.

$$\begin{aligned}\min_{X}\quad & H = \sum_{i=1}^{n} h_i\\ & S_J = \sum_{m=1}^{M}\sqrt{\frac{1}{H}\int_0^H \left(\dddot{\theta}_{mi}(t)\right)^2 dt}\\ & S_q = \sum_{i=0}^{n}\sum_{m=1}^{M}\left(\int_{h_i}^{h_{i+1}} \tau_{mi}(t)\, d\theta_{mi}\right)\\ \text{s.t.}\quad & y^{\text{MIN}} \le y_i^{*} \le y^{\text{MAX}};\\ & z^{\text{MIN}} \le z_i^{*} \le z^{\text{MAX}};\\ & \left|y_i^{*} - y^{\text{STRAIGHT}}\right| \le \delta^{\text{MAX}};\\ & \dot{\theta}_m^{\text{LB}} \le \dot{\theta}_{mi}(t) \le \dot{\theta}_m^{\text{UB}};\\ & \ddot{\theta}_m^{\text{LB}} \le \ddot{\theta}_{mi}(t) \le \ddot{\theta}_m^{\text{UB}};\\ & \left|\dddot{\theta}_{mi}(t)\right| \le J^{\text{MAX}};\\ & |\tau| \le \tau^{\text{UB}};\\ & \sum_{m=1}^{M} \frac{\sqrt{(1/H)\int_0^H \left(\tau_{mi}(t)\right)^2 dt}}{\sqrt{(1/N)\sum_{k=1}^{N}\left(\tau_{mi}(k\Delta T) - \tau_{mi}((k-1)\Delta T)\right)^2}}\\ & \le M^{\text{UP}}.\end{aligned} \tag{15}$$

3.3. Analysis of Optimization Results. The unstacking robot is recycled in accordance with the standard trajectory before the optimization, the total time-consuming is 3.5 seconds, the robot joint trajectory is constructed with 7 order B-spline curve, and the velocity, acceleration, and jerk are set to zero at the start and stop times. For multiobjective optimization problem, genetic algorithm is one of the most mature methods, and this paper chose NSGA II multiobjective evolutionary algorithm [19]. According to the actual experience, the number of population is 200, the number of iterations is 1000, the probability of parent crossing is 0.5, and the probability of individual population variation is 0.01. The Pareto frontier of the optimization model is shown in Figure 5.

It can be seen in Figure 5 that robot efficiency index and track smoothness indicators and energy consumption indicators conflict with each other; the higher the efficiency, the greater the energy consumption and the less smooth the trajectory. In order to shorten the no-load running time of the robot during recycling process, take the efficiency as the main objective and energy consumption and smoothness as reference index. To get the best results from the Pareto frontier surface of the optimization model, add constraints to the reference index as $S_J \le 800°/\text{s}^2$ and $S_q \le 3000\,\text{J}$. So the optimal results can be obtained as shown in Table 2.

According to the optimization results, the Cartesian spatial trajectory comparison and the corresponding optimal

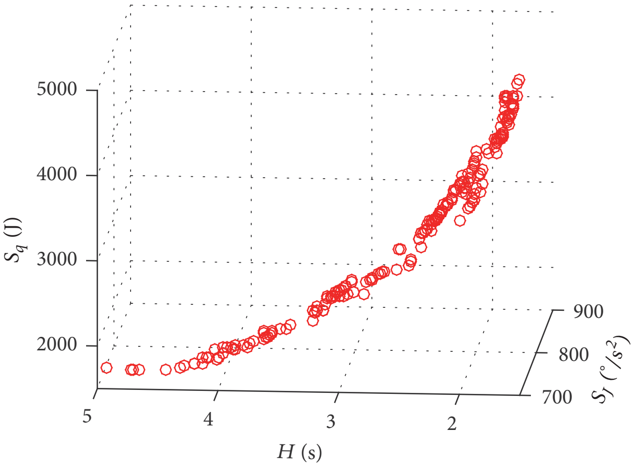
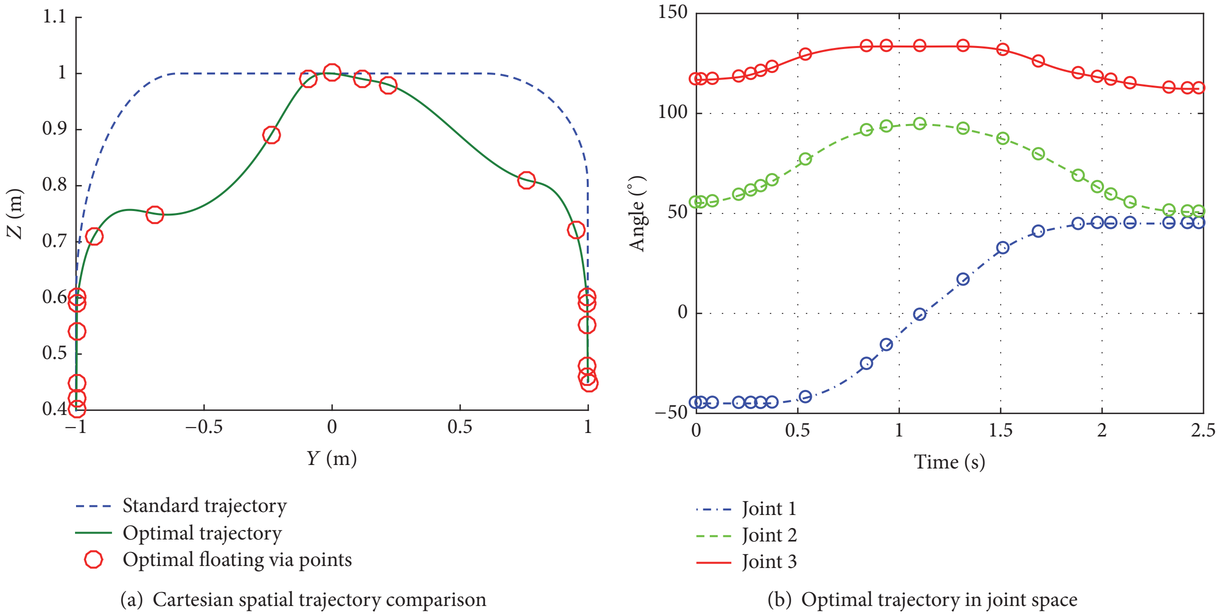


FIGURE 5: The Pareto frontier of the optimization model.



(a) Cartesian spatial trajectory comparison

(b) Optimal trajectory in joint space

FIGURE 6: Cartesian spatial trajectory comparison and the corresponding optimal trajectory in joint space.

trajectory in joint space are obtained in Figure 6. The velocity, acceleration, and output torque of each joint are shown in Figure 7.

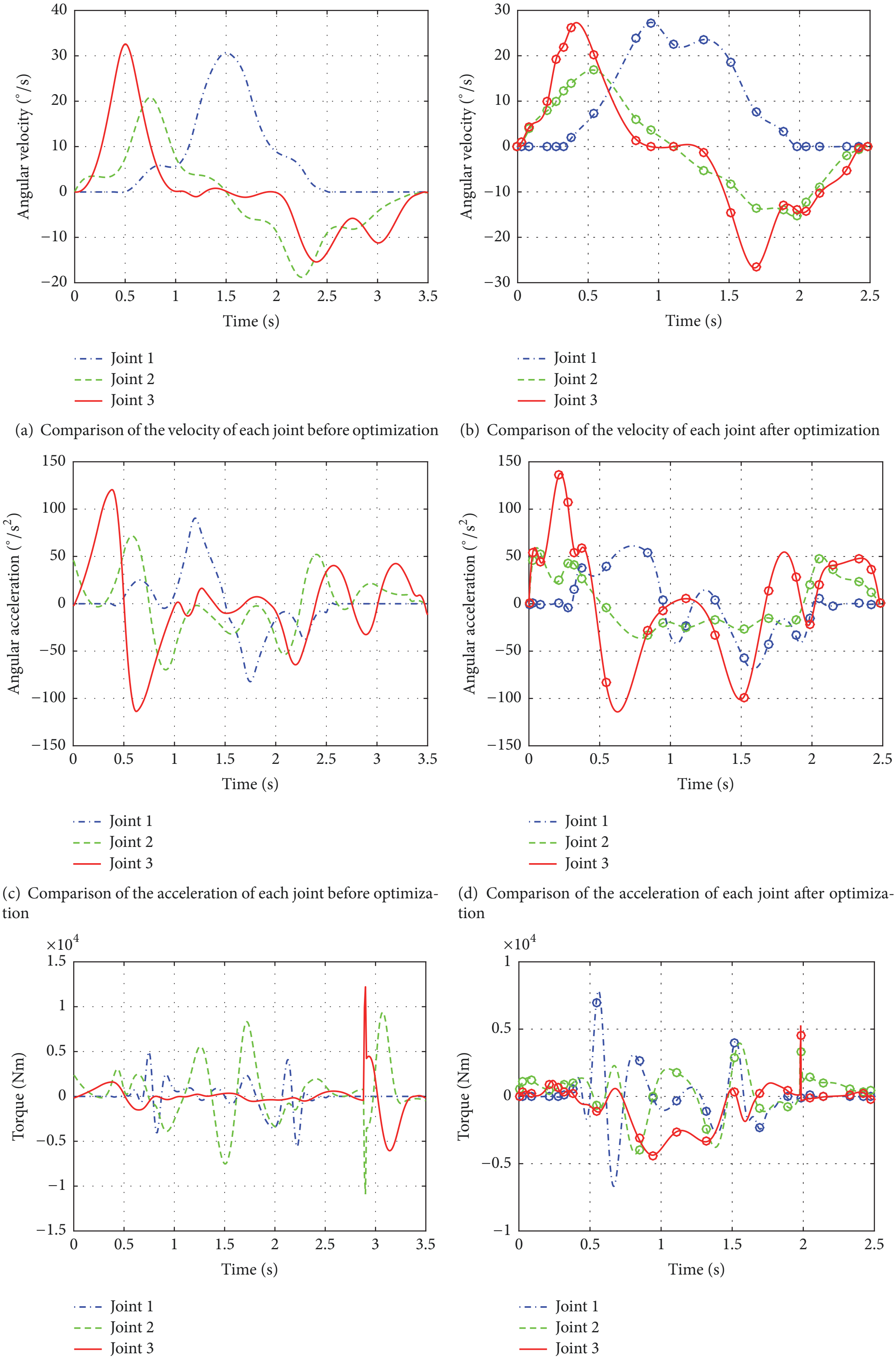
It can be seen in Figure 6 that time required for the robot recycling process is reduced from 3.5 s to 2.479 s, and the production efficiency is improved by 29.17%. The total energy consumption is reduced from 3186.15 J to 2785.64 J, and the joint average acceleration is reduced from 943.71°/s^2 to 794.14°/s^2; that is, the robot energy consumption and track smoothness were reduced by 12.57% and 15.85%. It can be seen from Figures 7(a)–7(d) that although the time required for the recycling process is greatly shortened by optimization, the angular velocity and acceleration of each joint are not significantly increased, and the acceleration of the starting point and the ending point are zero, so as to avoid causing infinite jerk and the impact on the system. It can be seen in Figures 7(e)–7(f) that the joint torque values are within the allowable range after optimization, especially when the robot end effector is inserted into the rubber. Joints 2 and 3 are subjected to significantly impact torque and the resistance is large in the process before optimization. Through the optimization of the floating via points, the time of end effector inserted into the rubber process is shortened, and at the same time the impact force and resistance during the insertion process are significantly reduced.

4. Adaptive Backstepping Control Strategy Based on RFWNN Method

The adaptive backstepping control is a systematic and recursive design methodology for the feedback control of nonlinear systems with parametric uncertainties. Unlike the

Table 2: Trajectory optimization results.

	Joint 1	Joint 2	Joint 3
Optimal position sequence (°)	[−45 −45 −45 −45 −45 −45 −44.852 −42.154 −25.503 −16.098 −0.967 16.616 32.361 40.608 44.449 45 45 45 45 45 45]	[55.2078 55.253 55.803 59.071 61.206 63.373 66.314 76.713 91.353 93.209 94.417 92.086 87.013 79.314 68.553 62.888 59.210 55.221 51.230 50.733 50.644]	[116.812 116.827 117.050 118.214 119.520 121.034 123.033 129.159 133.419 133.504 133.464 133.490 131.504 125.722 119.897 118.047 116.636 114.876 112.666 112.302 112.314]
Optimal time series (s)	[0 0.028 0.084 0.214 0.273 0.322 0.378 0.543 0.844 0.942 1.106 1.320 1.516 1.691 1.887 1.982 2.048 2.143 2.332 2.423 2.479]		
Optimal object	$H = 2.479\,\text{s}$, $S_J = 794.14^{\circ}/\text{s}^2$, $S_q = 2785.64\,\text{J}$		



(a) Comparison of the velocity of each joint before optimization

(b) Comparison of the velocity of each joint after optimization

(c) Comparison of the acceleration of each joint before optimization

(d) Comparison of the acceleration of each joint after optimization

(e) Comparison of the output torque of each joint before optimization

(f) Comparison of the output torque of each joint after optimization

FIGURE 7: Comparison of each joint before and after optimization.

feedback linearization method with the problems such as the precise model requirement and the cancellation of useful nonlinear terms, the backstepping approach offers a choice of design tools for accommodation of uncertainties and nonlinearities and can avoid wasteful cancellations.

4.1. Backstepping Control System Planning. The key idea of the backstepping design is to select recursively some appropriate state variables as virtual inputs for lower dimension subsystems of the overall system and the Lyapunov functions are designed for each stable virtual controller. Therefore, the finally designed actual control law can guarantee the stability of total control system. Combined with (1), the dynamic model of the rubber unstacking robot during recycling process which considered the robot joint friction and the system unknown disturbance is shown as follows:

$$\begin{aligned} M(\theta)\ddot{\theta} + C\left(\theta,\dot{\theta}\right)\dot{\theta} + G(\theta) + f\left(\dot{\theta}\right) + J^T(\theta)F_s + \tau_d \\ = \tau, \end{aligned} \tag{16}$$

where $f(\dot{\theta}) \in R^{M\times 1}$ is the joint friction term and $\tau_d \in R^{M\times 1}$ is the system unknown disturbance term.

$\theta_d(t)$, $\dot{\theta}_d(t)$, and $\ddot{\theta}_d(t)$ are the optimal trajectory parameters calculated in Section 3. Define $\overline{M}(\theta)$, $\overline{C}(\theta,\dot{\theta})$, $\overline{G}(\theta)$, and $\overline{f}(\dot{\theta})$ as the actual values with uncertainties in the nominal values $M(\theta)$, $C(\theta,\dot{\theta})$, $G(\theta)$, and $f(\dot{\theta})$, respectively. According to the above definition, the robot dynamics model is reconstructed as follows:

$$M(\theta)\ddot{\theta} + y\left(\theta,\dot{\theta}\right) + J^T(\theta)F_s + \tau_d = \tau, \tag{17}$$

where $y(\theta,\dot{\theta})$ is the uncertainty term of the robot dynamics system which can be expressed as follows:

$$\begin{aligned} & y\left(\theta,\dot{\theta}\right) \\ & \quad = M(\theta)\overline{M}^{-1}(\theta)\left[\overline{C}\left(\theta,\dot{\theta}\right)\dot{\theta} + \overline{G}(\theta) + f\left(\dot{\theta}\right)\right]. \end{aligned} \tag{18}$$

Define state variables $X_1 = \theta$ and $X_2 = \dot{\theta}$; dynamics model (17) is reconstructed as follows:

$$\begin{aligned} \dot{X}_1 &= X_2, \\ \dot{X}_2 &= M^{-1}(X_1)\left[\tau - J^T(\theta)F_s - y\left(\theta,\dot{\theta}\right) - \tau_d\right]. \end{aligned} \tag{19}$$

According to (19), the goal of the control system is to design an adaptive control law so that the state vector X_1 can precisely track the desired trajectory θ_d. Define tracking error $E_1(t)$ as

$$E_1(t) = X_1(t) - \theta_d(t) \tag{20}$$

and its derivative is

$$\dot{E}_1(t) = \dot{X}_1(t) - \dot{\theta}_d(t) = v(t) - \dot{\theta}_d(t), \tag{21}$$

where $v(t) = \dot{X}_1(t)$ is called the virtual control function and the stabilizing function $s(t)$ is defined as

$$s(t) = -\varsigma_1 E_1(t) + \dot{\theta}_d(t), \tag{22}$$

where ς_1 is a positive definite diagonal matrix. The first Lyapunov function is chosen as

$$V_1(t) = \frac{1}{2}E_1^T(t)E_1(t) \tag{23}$$

and its derivative is

$$\begin{aligned} \dot{V}_1(t) &= E_1^T(t)\dot{E}_1(t) = E_1^T(t)\left(\dot{X}_1(t) - \dot{\theta}_d(t)\right) \\ &= E_1^T(t)\left(v(t) - s(t) - \varsigma E_1(t)\right). \end{aligned} \tag{24}$$

According to the Lyapunov stability criterion, the position tracking error $E_1(t)$ converges asymptotically when the virtual control function $v(t)$ is equal to the stability function $s(t)$. Define $E_2(t) = v(t) - s(t) - \dot{E}_1(t)$, and then the derivative of $E_2(t)$ is expressed as

$$\begin{aligned} \dot{E}_2(t) &= \dot{v}(t) - \dot{s}(t) = \dot{X}_2(t) + \varsigma_1\dot{E}_1(t) - \ddot{\theta}_d(t) \\ &= M^{-1}(X_1)\left[\tau - J^T(X_1)F_s - y(X_1,X_2) - \tau_d\right] \\ &\quad + \varsigma_1\dot{E}_1(t) - \ddot{\theta}(t). \end{aligned} \tag{25}$$

To design the backstepping control system, the Lyapunov function is defined as

$$V_2(t) = V_1(t) + \frac{1}{2}E_2^T(t)E_2(t). \tag{26}$$

And its derivative can be derived as follows:

$$\begin{aligned} \dot{V}_2(t) &= \dot{V}_1(t) + E_2^T(t)\dot{E}_2(t) = E_1^T(t)\left(E_2(t)\right. \\ &\quad \left. - \varsigma_1 E_1(t)\right) + E_2^T(t) \\ &\quad \times\left\{M^{-1}(X_1)\left[\tau - J^T(X_1)F_s - y(X_1,X_2) - \tau_d\right]\right. \\ &\quad \left. + \varsigma_1\dot{E}_1(t) - \ddot{\theta}(t)\right\}. \end{aligned} \tag{27}$$

From (27), the backstepping control law that satisfies the Lyapunov stability condition is designed as follows:

$$\begin{aligned} \tau &= M(X_1)\left(\ddot{\theta}(t) - \Lambda(t)\right) + J^T(X_1)F_s + y(X_1,X_2) \\ &\quad + \tau_d, \end{aligned} \tag{28}$$

where $\Lambda(t) = [\varsigma_1\dot{E}_1(t) + \varsigma_2 E_2(t) + E_1(t)]$ is the defined system synthesis error and ς_2 is a positive definite diagonal matrix. The backstepping control law shown in (28) is difficult to achieve due to the effect of the uncertainty term of the robot dynamic model $y(X_1,X_2)$ and the time-varying disturbance τ_d. Therefore, adaptive robust control technology is needed, which can not only identify the uncertain dynamic of the robot but also compensate for the external time-varying disturbance to ensure the tracking accuracy of the robot trajectory. An adaptive robust online learning control strategy based on recursive fuzzy wavelet neural network (RFWNN) is proposed. RFWNN is used to estimate the uncertainty term of the robot dynamics system. The robust compensator compensates for the external disturbance and the estimation error of the former.

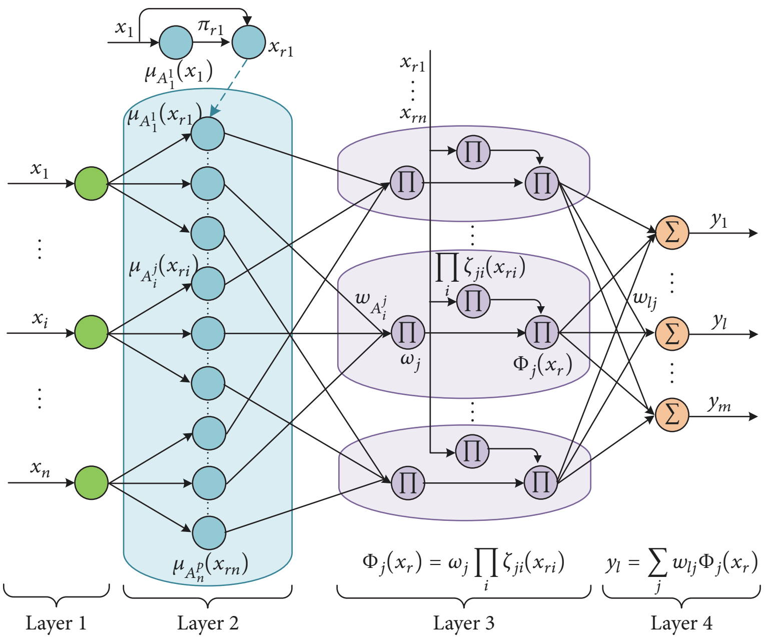


Figure 8: RFWNN structure.

4.2. RFWNN Structure. The RFWNN structure consists of a recursive structure and an FWNN structure as shown in Figure 8, FWNN structure also includes TSK fuzzy system based on fuzzy rules and wavelet neural network (WNN) structure. Each fuzzy rule is considered as a sub-WNN space, and each sub-WNN contains a wavelet with a specified dilation value (resolution). Therefore, sub-WNNs of different resolution levels can capture different behaviors of the approximate functions, and the role of fuzzy sets is to determine the contribution of the sub-FWNN space to input/output of the FWNNs. Therefore, with the main feature stated above, the RFWNN can achieve the same approximation accuracy with a smaller size in comparison with the FNNs/NNs. In addition, feedback loop with the time delay at the FWNN fuzzy layer will make the system become a dynamic structure that have sufficient capability to handle the high dynamics of the robot control system.

The RFWNN structure consists of four layers, the first layer is the input layer, input signals $x_1, x_2, \ldots, x_n$ are transported to the fuzzy layer by neurons, each of which is represented by its corresponding fuzzy membership function $\mu_{A_i^j}(x_i)$, and the equation is as follows:

$$\mu_{A_i^j}(x_i) = e^{-d_{ji}^2(x_i - c_{ji})^2}, \tag{29}$$

where d_{ji} is scale parameters, c_{ji} is translation parameter, p is rule number, and $j = 1, \ldots, p$, $i = 1, \ldots, n$, $(p, n) \in N$. A local feedback unit with a real-time delay section is added to the layer, so the recursive input of the layer is represented as follows:

$$x_{ri}(t) = x_i(t) + \pi_{ri}\mu_{A_i^j}(x_i(t - T)), \tag{30}$$

where the membership function $\mu_{A_i^j}(x_i(t-T))$ represents the delay value of $\mu_{A_i^j}(x_i)$ within the time interval T and π_{ri} is the recursive weight vector of the feedback unit.

The third layer is a fuzzy rule layer, each neuron in the layer is represented as a rule, the prerequisite for the execution of the neuron is matched with the relevant rule, and the output of the layer is multiplied by

$$\omega_j = \prod_i w_{A_i^j}\mu_{A_i^j}(x_{ri}), \tag{31}$$

where $w_{A_i^j}$ is the weight between the fuzzy layer and the rule layer and the multiplication operator $\prod$ is used to determine the excitation intensity in the fuzzy reasoning mechanism. The excitation intensity of rule j is simplified by a combination of ω_j and $\prod_i \zeta_{ji}(x_{ri})$. The fuzzy wavelet basis function is as follows:

$$\Phi_j(x_r) = \omega_j \prod_i \zeta_{ji}(x_{ri}), \tag{32}$$

where $\zeta_{ji}(x_{ri}) = 1 - d_{ji}{}^2(x_{ri} - c_{ji})^2$, $x_{ri} = [x_{r1}, \ldots, x_{rn}]^T \in R^{pn\times 1}$, and $r = 1, \ldots, p$. The wavelet basis function $\Phi_j(x_r)$ is expressed as a multidimensional wavelet function as follows:

$$\Phi_j(x_r) = \prod_i \phi_j(x_{ri}), \tag{33}$$

where $\phi_j(x_{ri}) = [1 - d_{ji}{}^2(x_{ri} - c_{ji})^2]e^{-d_{ji}{}^2(x_{ri}-c_{ji})^2}$ is the Mexican Cap Wavelet function.

The fourth layer is a fuzzy output layer, where each node represents the output variable and is calculated by summing all the input signals

$$y_l = \sum_j w_{lj}\Phi_j(x_r), \tag{34}$$

where w_{lj} is the weight between the rule layer and the output layer, $l = 1, \ldots, m$, $m \in N$, m is RFWNN number of output variables, and the above equation is expressed as a vector form as follows:

$$\begin{gathered}
y(x, d, c, \pi, W) = W^T \Phi(x_r, d, c, \pi); \\
\Phi(x_r, d, c, \pi) = \left[\Phi_1(x_r), \Phi_2(x_r), \ldots, \Phi_p(x_r)\right]^T \in R^{p\times 1}; \\
W^T = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1p} \\ w_{21} & w_{22} & \cdots & w_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mp} \end{bmatrix} \in R^{m\times p},
\end{gathered} \tag{35}$$

where $d \in R^{np\times 1}$, $x \in R^{n\times 1}$, and $y \in R^{m\times 1}$ are scaling parameter vector, $c \in R^{np\times 1}$ is translation parameter vector, and $\pi \in R^{np\times 1}$ is recursive weight parameter vector.

The RFWNN structure is mainly used to online estimate the uncertain dynamics model of the robot. Based on the approximate error analysis, there is an ideal RFWNN structure with the optimal parameters [20]

$$y(x(t)) = W^{*T} \Phi^*(x_r(t), d^*, c^*, \pi^*) + \Delta(x_r(t)), \tag{36}$$

where W^*, d^*, c^*, and π^* are the optimal parameters corresponding to $W, d, c,$ and π and $\Delta(xr(t))$ is the estimation error vector.

4.3. Adaptive Backstepping Controller Design. In (28), the y part is approximated by the RFWNN network, the process will inevitably produce approximate error, and the τ_d part contains the unstructured uncertainties interference of the mechanical model established by the robot insertion process. Thus, this section estimates and compensates for the approximate error and τ_d by adding robust terms to the controller. System torque control law is

$$\tau = M\left(\ddot{\theta} - \Lambda\right) + J^T F_s + \hat{y} + \hat{\tau}_d, \tag{37}$$

where $\hat{\tau}_d$ is a robust term used to compensate for uncertainties, such as approximate error, unknown interference, and unspecified parts of the robot control system, and $\hat{y}$ is an approximate term for the equation y, expressed as

$$\hat{y}(x(t)) = \widehat{W}^T \Phi\left(x_r(t), \hat{d}, \hat{c}, \hat{\pi}\right), \tag{38}$$

where $x = [\theta^T, \dot{\theta}^T, {\theta_d}^T, {\dot{\theta}_d}^T, {\ddot{\theta}_d}^T]^T$ and $\widehat{W}$, $\hat{d}$, $\hat{c}$, and $\hat{\pi}$ are, respectively, the approximate value of W^*, d^*, c^*, and π^*.

The trajectory tracking control rate shown in (37) is introduced into the robot dynamics equation shown in (17) to obtain the closed-loop control system of the rubber unstacking robot

$$-M\Lambda = \tau_d + \tilde{y} - \hat{\tau}_d, \tag{39}$$

where $\tilde{y}$ is as follows:

$$\tilde{y} = W^{*T} \widetilde{\Phi} + \widetilde{W}^T \widehat{\Phi} + \Delta, \tag{40}$$

where $\widetilde{W} = W^* - \widehat{W}$, $\widehat{\Phi} \equiv \Phi(x_r, \hat{d}, \hat{c}, \hat{\pi})$, $\Phi^* \equiv \Phi(x_r, d^*, c^*, \pi^*)$, and $\widetilde{\Phi} = \Phi^* - \Phi$.

In order to achieve good tracking performance of the controller, the parameters such as weight, scaling, and translation in the RFWNN neural network need to be adjusted online. The nonlinear output of the RFWNN is transformed into a partial linear form by using the linearization technique, which can be extended by applying the Lyapunov theorem. Thus, $\widetilde{\Phi}$ is expanded in the Taylor series to get the following form:

$$\begin{aligned}
\widetilde{\Phi} = {} & P^T\left(d^* - \hat{d}\right) + K^T\left(c^* - \hat{c}\right) + H^T\left(\pi^* - \hat{\pi}\right) \\
& + \Theta\left(d^* - \hat{d}, c^* - \hat{c}, \pi^* - \hat{\pi}\right),
\end{aligned} \tag{41}$$

where vector Θ is the higher order term in the Taylor series expansion, $P = [\partial\Phi_1/\partial d, \ldots, \partial\Phi_p/\partial d]|_{d=\hat{d}}$, $K = [\partial\Phi_1/\partial c, \ldots, \partial\Phi_p/\partial c]|_{c=\hat{c}}$, $H = [\partial\Phi_1/\partial\pi, \ldots, \partial\Phi_p/\partial\pi]|_{\pi=\hat{\pi}}$ are bounded normal vectors, and

$$\begin{gathered}
\left[\frac{\partial\Phi_j}{\partial d}\right] = \left[\underbrace{0, \ldots, 0}_{(j-1)n}, \frac{\partial\Phi_j}{\partial d_{j1}}, \ldots, \frac{\partial\Phi_j}{\partial d_{jn}}, \underbrace{0, \ldots, 0}_{(p-j)n}\right]^T, \\
\left[\frac{\partial\Phi_j}{\partial c}\right] = \left[\underbrace{0, \ldots, 0}_{(j-1)n}, \frac{\partial\Phi_j}{\partial c_{j1}}, \ldots, \frac{\partial\Phi_j}{\partial c_{jn}}, \underbrace{0, \ldots, 0}_{(p-j)n}\right]^T, \\
\left[\frac{\partial\Phi_j}{\partial \pi}\right] = \left[\underbrace{0, \ldots, 0}_{(j-1)n}, \frac{\partial\Phi_j}{\partial \pi_{j1}}, \ldots, \frac{\partial\Phi_j}{\partial \pi_{jn}}, \underbrace{0, \ldots, 0}_{(p-j)n}\right]^T.
\end{gathered} \tag{42}$$

Define $\tilde{d} = d^* - \hat{d}$, $\tilde{c} = c^* - \hat{c}$, and $\tilde{\pi} = \pi^* - \hat{\pi}$, and (41) is as follows:

$$\widetilde{\Phi} = P^T\tilde{d} + K^T\tilde{c} + H^T\tilde{\pi} + \Theta\left(\tilde{d}, \tilde{c}, \tilde{\pi}\right). \tag{43}$$

According to (40) and (43) one can obtain the following:

$$\begin{aligned}
\tilde{y} + \tau_d = {} & \widetilde{W}^T\left(\widehat{\Phi} - P^T\hat{d} - K^T\hat{c} - H^T\hat{\pi}\right) \\
& + \widehat{W}^T\left(P^T\tilde{d} + K^T\tilde{c} + H^T\tilde{\pi}\right) + \Gamma_u,
\end{aligned} \tag{44}$$

where $\Gamma_u = W^{*T}(\widetilde{\Phi} + P^T\hat{d} + K^T\hat{c} + H^T\hat{\pi}) - \widehat{W}^T(P^T d^* + K^T c^* + H^T\pi^*) + \Delta + \tau_d$.

Substituting (44) into (39),

$$\begin{aligned}
-M\Lambda = {} & \widetilde{W}^T\left(\widehat{\Phi} - P^T\hat{d} - K^T\hat{c} - H^T\hat{\pi}\right) - \hat{\tau}_d + \Gamma_u \\
& + \widehat{W}^T\left(P^T\tilde{d} + K^T\tilde{c} + H^T\tilde{\pi}\right).
\end{aligned} \tag{45}$$

Set the positive real number W^B, d^B, c^B, π^B, Δ^B, and ${\tau_d}^B$ to satisfy the condition $\|W^*\| \le W^B$, $\|d^*\| \le d^B$, $\|c^*\| \le c^B$,

$\|\pi^*\| \le \pi^B$, $\|\Delta\| \le \Delta^B$, and $\|\tau_d\| \le {\tau_d}^B$, and it can be obtained as follows:

$$\begin{aligned}
\|\Gamma_u\| &= \Big\| W^{*T}\left(\widetilde{\Phi} + I^T\widehat{d} + K^T\widehat{c} + H^T\widehat{\pi}\right) \\
&\quad - \widehat{W}^T\left(I^T d^* + K^T c^* + H^T\pi^*\right) + \Delta + \tau_d \Big\| \\
&\le \left\| W^{*T}\widetilde{\Phi} + \Delta + \tau_d \right\| + \left\| W^{*T} I^T \right\| \|\widehat{d}\| + \left\| W^{*T} K^T \right\| \\
&\quad \cdot \|\widehat{c}\| + \left\| W^{*T} H^T \right\| \|\widehat{\pi}\| \\
&\quad + \left\| \left(I^T d^* + K^T c^* + H^T \pi^*\right) \right\| \|\widehat{W}\| .
\end{aligned} \tag{46}$$

Add $\alpha(W^B)^2/4 + \alpha(d^B)^2/4 + \alpha(c^B)^2/4 + \alpha(\pi^B)^2/4$ to both ends of inequality (46),

$$\begin{aligned}
&\|\Gamma_u\| + \frac{\alpha\left(W^B\right)^2}{4} + \frac{\alpha\left(d^B\right)^2}{4} + \frac{\alpha\left(c^B\right)^2}{4} + \frac{\alpha\left(\pi^B\right)^2}{4} \\
&\quad \le \beta^{*T}\upsilon,
\end{aligned} \tag{47}$$

where $\upsilon^T = [1, \|\widehat{d}\|, \|\widehat{c}\|, \|\widehat{\pi}\|, \|\widehat{W}\|]$, α is positive constant, $\beta^{*T} = [\beta_1, \beta_2, \beta_3, \beta_4, \beta_5]$, β_1, β_2, β_3, β_4, and β_5 are positive constant and the uncertainty of the following term: $\|W^{*T}\widetilde{\Phi} + \Delta + \tau_d\| + \alpha(W^B)^2/4 + \alpha(d^B)^2/4 + \alpha(c^B)^2/4 + \alpha(\pi^B)^2/4$, $\|W^{*T}P^T\|$, $\|W^{*T}K^T\|$, $\|W^{*T}H^T\|$, and $\|(P^T d^* + K^T c^* + H^T \pi^*)\|$.

In order to ensure that the dynamic closed-loop system shown in (35) is stable, the robust term $\widehat{\tau}_d$ must cancel the indeterminate term Γ_u. Therefore, using $\widehat{\tau}_d$ to estimate the uncertainty limit $\beta^{*T}\upsilon$, the expression of the robust term $\widehat{\tau}_d$ is given as follows:

$$\widehat{\tau}_d = \frac{E_1}{\|E_1\|}\widehat{\beta}^T\upsilon + b_s\frac{E_1}{\|E_1\|^2 + \delta}, \tag{48}$$

where b_s and δ are positive constants and $\widehat{\beta}$ is the online estimate of β^*. According to the above analysis, an adaptive online learning algorithm based on RFWNN control system and a robust term expression are proposed as follows:

$$\begin{aligned}
\dot{\widehat{W}} &= K_W\left(\widehat{\Phi} - P^T\widehat{d} - K^T\widehat{c} - H^T\widehat{\pi}\right){E_1}^T \\
&\quad - \alpha K_W \|E_1\| \widehat{W}, \\
\dot{\widehat{d}} &= K_d P\widehat{W}E_1 - \alpha K_d \|E_1\| \widehat{d}; \\
\dot{\widehat{c}} &= K_c K\widehat{W}E_1 - \alpha K_c \|E_1\| \widehat{c}, \\
\dot{\widehat{\pi}} &= K_\pi H\widehat{W}E_1 - \alpha K_\pi \|E_1\| \widehat{\pi}; \\
\dot{\widehat{\beta}} &= \|E_1\| K_\beta \upsilon,
\end{aligned} \tag{49}$$

where K_W, K_d, K_c, K_π, and K_β are diagonal normal matrix. Based on the above analysis, the controller structure is shown in Figure 9.

Combined with the control block diagram, the design procedure of the proposed control system is detailed in the following:

(1) Specify the structure of the RFWNN (n, p, m), and set constant parameters $(\varsigma_1, \varsigma_2, K_W, K_d, K_c, K_\pi, K_\beta$ and $\alpha)$.

(2) Initial the parameters of the RFWNN $(\widehat{W}, \widehat{c}, \widehat{d}, \widehat{\pi}, \mu_{A_i^j}(x_i(0)))$.

(3) Update the RFWNN inputs $(x = [\theta^T, \dot{\theta}^T, \theta_d^T, \dot{\theta}_d^T, \ddot{\theta}_d^T]^T)$ and memory data $(\widehat{W}, \widehat{c}, \widehat{d}, \widehat{\pi}, \mu_{A_i^j}(x_i(t-T)))$.

(4) Compute tracking error $E_1(t)$ and system synthesis error $\mathbf{\Lambda}(t)$, compute the recurrent inputs $x_{ri}(t)$ via (30), and compute the membership functions $\mu_{A_i^j}(x_{ri}(t))$ via (29).

(5) Compute the outputs ω_j of the fuzzy rule layer via (31), and then compute fuzzy wavelet basic functions $\Phi_j(x_r)$ via (32) and (33).

(6) Compute the outputs of the RFWNN $\widehat{y}(x(t))$ via (34), compute the robust term $\widehat{\tau}_d$ via (48), and then construct the control input τ via (37).

(7) Adjust the weights, dilations, and translations and the recurrent values of the RFWNNs via (49).

(8) Save these data in the memory and return to step (3).

5. Experimental Verification

A servo control system based on EtherCAT bus and 4-DOF unstacking robot body experimental platform were built in this work. Beckhoff IPC was used as industrial computer which included TwinCAT3 programming software. The RS2 series servo drives were applied in this work which supported EtherCAT bus communications. The actual output torque and actual position of the robot joints are fed back in real time by the NCtoPLC function of TwinCAT3 software. So real-time communication and robot control could be realized via high-speed EtherCAT bus. Figure 10 shows the entire experimental system.

The model of the unstacking robot is described by (16) with its main dynamics given as

$$M(\theta) = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix},$$

$$C\left(\theta, \dot{\theta}\right) = \begin{bmatrix} 0 & C_{12}\dot{\theta}_2 & C_{13}\dot{\theta}_3 \\ C_{21}\dot{\theta}_1 & 0 & C_{23}\dot{\theta}_3 \\ C_{31}\dot{\theta}_1 & C_{32}\dot{\theta}_2 & 0 \end{bmatrix},$$

$$\begin{aligned}
m_{11} &= 4.4\cos^2\left(\theta_3 + 0.26\right) \\
&\quad - 45.1\cos\left(\theta_3 + 0.26\right)\cos\theta_2 \\
&\quad + 234.4\cos^2\theta_3 + 345.9\cos^2\theta_2
\end{aligned}$$

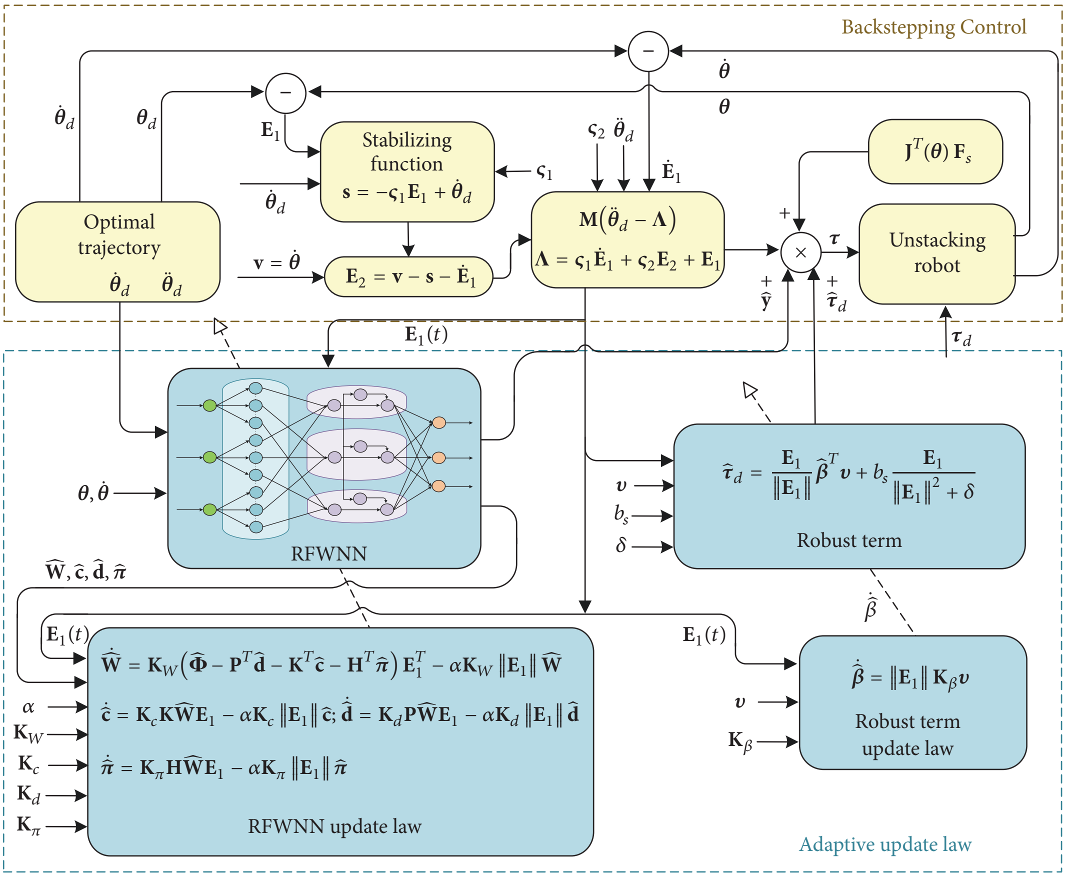

Backstepping Control
$\dot{\theta}$
θ
$\dot{\theta}_d$
θ_d
$\mathbf{E}_1$
Stabilizing function
$\mathbf{s} = -\varsigma_1\mathbf{E}_1 + \dot{\theta}_d$
ς_1
ς_2 $\ddot{\theta}_d$
$\dot{\mathbf{E}}_1$
$\mathbf{J}^T(\theta)\,\mathbf{F}_s$
$+$
τ
Unstacking robot
$\mathbf{M}(\ddot{\theta}_d - \Lambda)$
$\Lambda = \varsigma_1\dot{\mathbf{E}}_1 + \varsigma_2\mathbf{E}_2 + \mathbf{E}_1$
Optimal trajectory
$\dot{\theta}_d$ $\ddot{\theta}_d$
$\mathbf{v} = \dot{\theta}$
$\mathbf{E}_2 = \mathbf{v} - \mathbf{s} - \dot{\mathbf{E}}_1$
$+$ $\hat{\mathbf{y}}$
$+$ $\hat{\tau}_d$
τ_d
$\mathbf{E}_1(t)$
$\theta, \dot{\theta}$
RFWNN
$\hat{\tau}_d = \frac{\mathbf{E}_1}{\|\mathbf{E}_1\|}\hat{\beta}^T v + b_s\frac{\mathbf{E}_1}{\|\mathbf{E}_1\|^2 + \delta}$
v
b_s
δ
Robust term
$\widehat{\mathbf{W}}, \hat{\mathbf{c}}, \hat{\mathbf{d}}, \hat{\pi}$
$\mathbf{E}_1(t)$
$\dot{\widehat{\mathbf{W}}} = \mathbf{K}_W(\widehat{\mathbf{\Phi}} - \mathbf{P}^T\hat{\mathbf{d}} - \mathbf{K}^T\hat{\mathbf{c}} - \mathbf{H}^T\hat{\pi})\mathbf{E}_1^T - \alpha\mathbf{K}_W\|\mathbf{E}_1\|\widehat{\mathbf{W}}$
$\dot{\hat{\mathbf{c}}} = \mathbf{K}_c\mathbf{K}\widehat{\mathbf{W}}\mathbf{E}_1 - \alpha\mathbf{K}_c\|\mathbf{E}_1\|\hat{\mathbf{c}}$; $\dot{\hat{\mathbf{d}}} = \mathbf{K}_d\mathbf{P}\widehat{\mathbf{W}}\mathbf{E}_1 - \alpha\mathbf{K}_d\|\mathbf{E}_1\|\hat{\mathbf{d}}$
$\dot{\hat{\pi}} = \mathbf{K}_\pi\mathbf{H}\widehat{\mathbf{W}}\mathbf{E}_1 - \alpha\mathbf{K}_\pi\|\mathbf{E}_1\|\hat{\pi}$
α
$\mathbf{K}_W$
$\mathbf{K}_c$
$\mathbf{K}_d$
$\mathbf{K}_\pi$
RFWNN update law
$\dot{\hat{\beta}}$
$\mathbf{E}_1(t)$
v
$\mathbf{K}_\beta$
$\dot{\hat{\beta}} = \|\mathbf{E}_1\|\,\mathbf{K}_\beta v$
Robust term update law
Adaptive update law

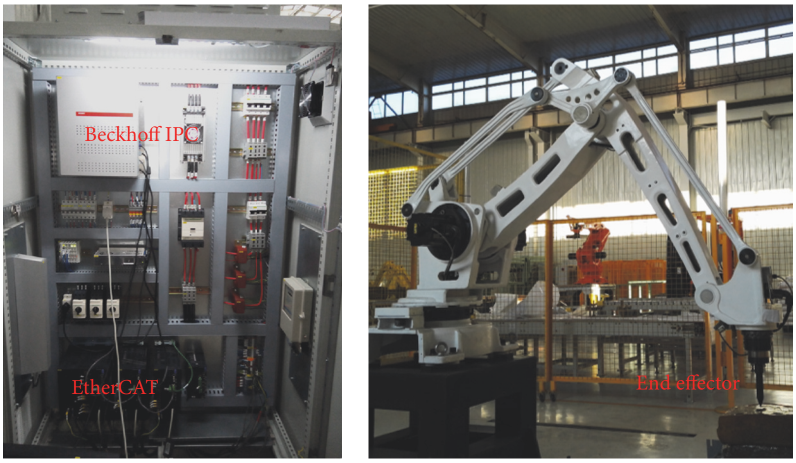
Figure 9: Control block diagram of the adaptive backstepping controller.

$$
\begin{aligned}
&\quad - 394.6\cos\theta_2\cos\theta_3 + 192.5\cos\theta_2\\
&\quad - 94.6\cos\theta_3 - 10.8\cos\left(\theta_3 + 0.26\right)\\
&\quad + 52.6,\\
m_{22} &= 365.37,\\
m_{33} &= 242.26,\\
m_{23} &= m_{32}\\
&= 22.6\cos\left(\theta_2 - \theta_3 - 0.26\right)\\
&\quad - 194.2\cos\left(\theta_2 - \theta_3\right),\\
C_{12} &= -2C_{21}\\
&= 45.1\sin\theta_2\cos\left(\theta_3 + 0.26\right) - 345.9\sin\theta_2\\
&\quad + 394.6\sin\theta_2\cos\theta_3 - 192.5\sin\theta_3,\\
C_{13} &= -2C_{31}\\
&= 10.8\sin\left(\theta_3 + 0.26\right) - 4.4\sin\left(2\theta_3 + 0.53\right)\\
&\quad + 394.6\cos\theta_2\sin\theta_3\\
&\quad + 45.1\sin\left(\theta_3 + 0.26\right)\cos\theta_2 + 94.6\sin\theta_3\\
&\quad - 235.4\sin 2\theta_3,\\
C_{23} &= -C_{32}\\
&= 22.6\sin\left(\theta_2 - \theta_3 - 0.26\right)\\
&\quad - 194.2\sin\left(\theta_2 - \theta_3\right).
\end{aligned}
\tag{50}
$$

The RFWNN structure can be characterized by $n = 15$, $p = 5$, and $m = 3$ nodes and the relevant parameters of the designed controller are set as follows:

$$
\begin{aligned}
\sigma &= \operatorname{diag}\left(120, 60, 25\right),\\
b_s &= 0.01,\\
\delta &= 0.001,\\
K_c &= K_d = K_\pi = \operatorname{diag}\left(35\right),\\
K_W &= \operatorname{diag}\left(60\right),\\
K_s &= \operatorname{diag}\left(180, 180, 180\right),\\
K_\beta &= \operatorname{diag}\left(0.002, 0.002, 0.002, 0.002, 0.002\right).
\end{aligned}
\tag{51}
$$

The control strategy is compared with PID control and PD synchronic control. The PID parameters are designed by a



(a) Control system

(b) Unstacking robot

Figure 10: Control system of unstacking robot.

Table 3: NMSE comparison of each joint in different control strategies.

NMSE ($\times 10^{-3}$)	Case 1 (rad)			Case 2 (rad)		
	Joint 1	Joint 2	Joint 3	Joint 1	Joint 2	Joint 3
PID	0.842	4.387	5.224	2.517	7.848	6.042
PD synchronic	0.955	2.282	2.897	1.338	3.635	4.055
Adaptive backstepping	0.225	1.512	2.199	0.301	2.167	2.552

NMSE = $\sqrt{\sum_{k=0}^{K}(\theta_{(k+1)} - \theta_{d(k+1)})^2/K}$ and K is the total sampling instant.

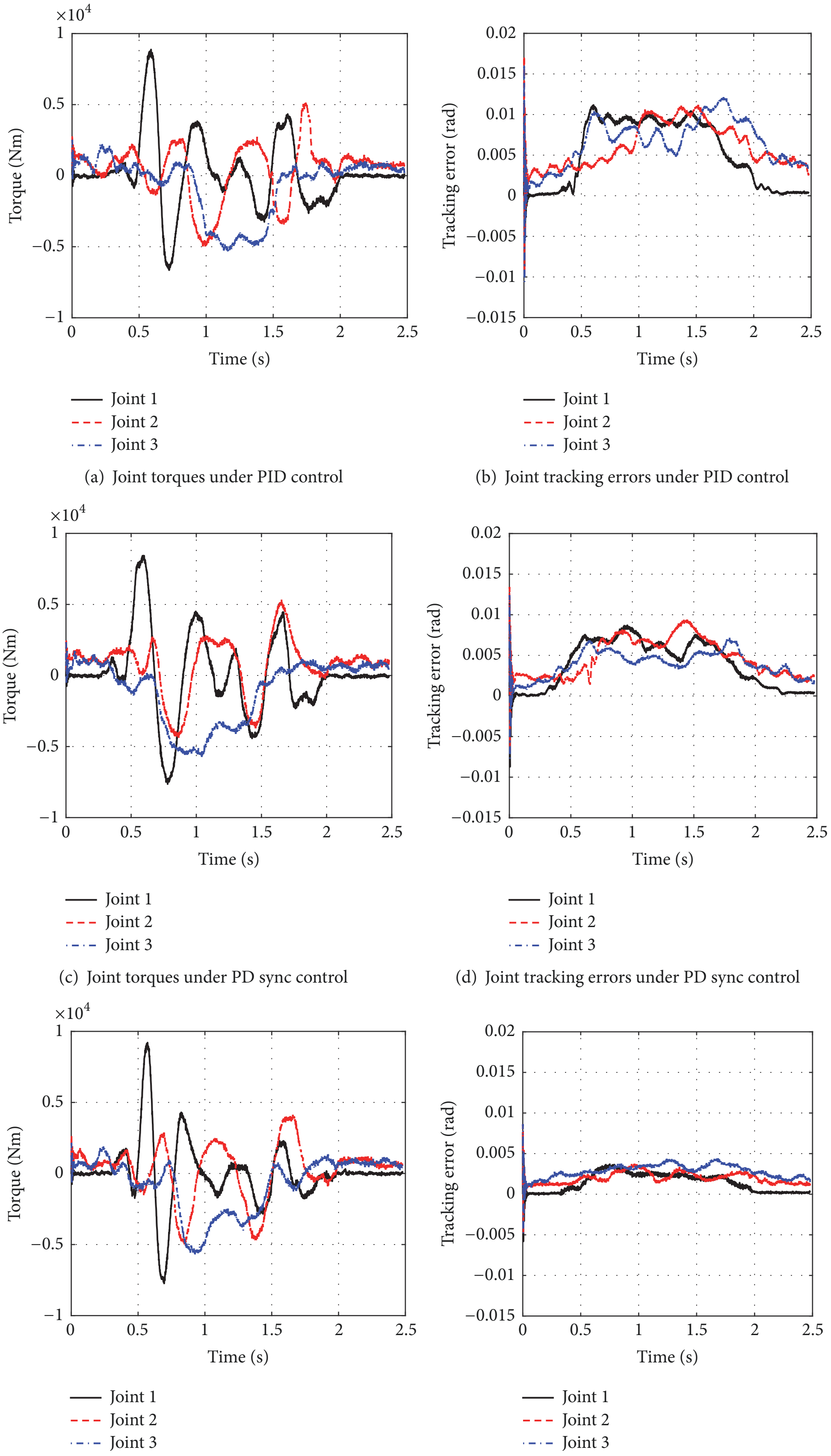
compromise between the superiority of control performance and the magnitude of control effort, and they are given as $K_{p1} = \mathrm{diag}(110, 70, 50)$, $K_{i1} = \mathrm{diag}(0.6, 0.4, 0.5)$, and $K_{d1} = \mathrm{diag}(1200, 900, 600)$. And the parameters of the PD synchronic controller are set as $K_{p2} = \mathrm{diag}(120, 50, 80)$ and $K_{d2} = \mathrm{diag}(1500, 1200, 800)$.

In order to verify the effect of the viscoelastic contact between the robot and the rubber on the control performance of the system and the ability of the control strategy proposed in this paper to effectively overcome this effect, the experiment in this section is validated in two cases. In Case 1, the unstacking robot tracks only the optimal trajectory without physical contact with the rubber. In Case 2, the unstacking robot inserts the rubber with the optimal trajectory according to the actual operation requirements.

In the first case, Figure 11 represents the joint torques and tracking errors of the robot under PID control and PD synchronic control and the proposed control strategy. In the second case, the torque and tracking error of each robot joint under different control strategies are shown in Figure 12. Since the viscoelastic contact between the robot and the rubber occurs within the time $1.982\,\mathrm{s} \leq t \leq 2.479\,\mathrm{s}$, the normalized mean square error (NMSE) of each joint trajectory is compared with the above two experimental cases during this time, as shown in Table 3.

It can be seen from Figure 11 that because the robot does not directly contact with the rubber in Case 1, the robot runs smoothly without large torque ripple and the tracking error is within an acceptable range. Compared with PID control, both PD synchronic control and the proposed control strategy in this paper have better performance and can control the trajectory tracking error to a smaller extent.

According to Figure 12 and Table 3, it can be seen that, under the control of PID, the joint torque of the robot fluctuates greatly, and the impact is more intense especially during the robot contact with rubber in Case 2. In addition the trajectory tracking error increases significantly, which will seriously affect the normal operation of the robot. Although the PD synchronization control can relatively improve the tracking accuracy of the robot in Case 1, the controller parameters cannot be adjusted online adaptively, and the robustness to interference is not strong enough so that it still cannot meet the system requirements in Case 2. Compared with the above controllers, the RFWNN-based adaptive backstepping controller proposed in this paper can reduce the trajectory tracking error significantly when the robot is inserted into the rubber. In addition the torque peak and fluctuations of each joint are effectively suppressed, so the robot can precisely and accurately perform the rubber recycling process according to the optimal trajectory. According to Table 3, it can be concluded that compared with Case 1 the NMSE value increases significantly in Case 2. And compared with the PID and PD synchronization control, the control method proposed in this paper can decrease the NMSE value of the robot joints by more than 72% and 40%, respectively, in the impact process. In Case 2, the robust term of the control



(a) Joint torques under PID control

(b) Joint tracking errors under PID control

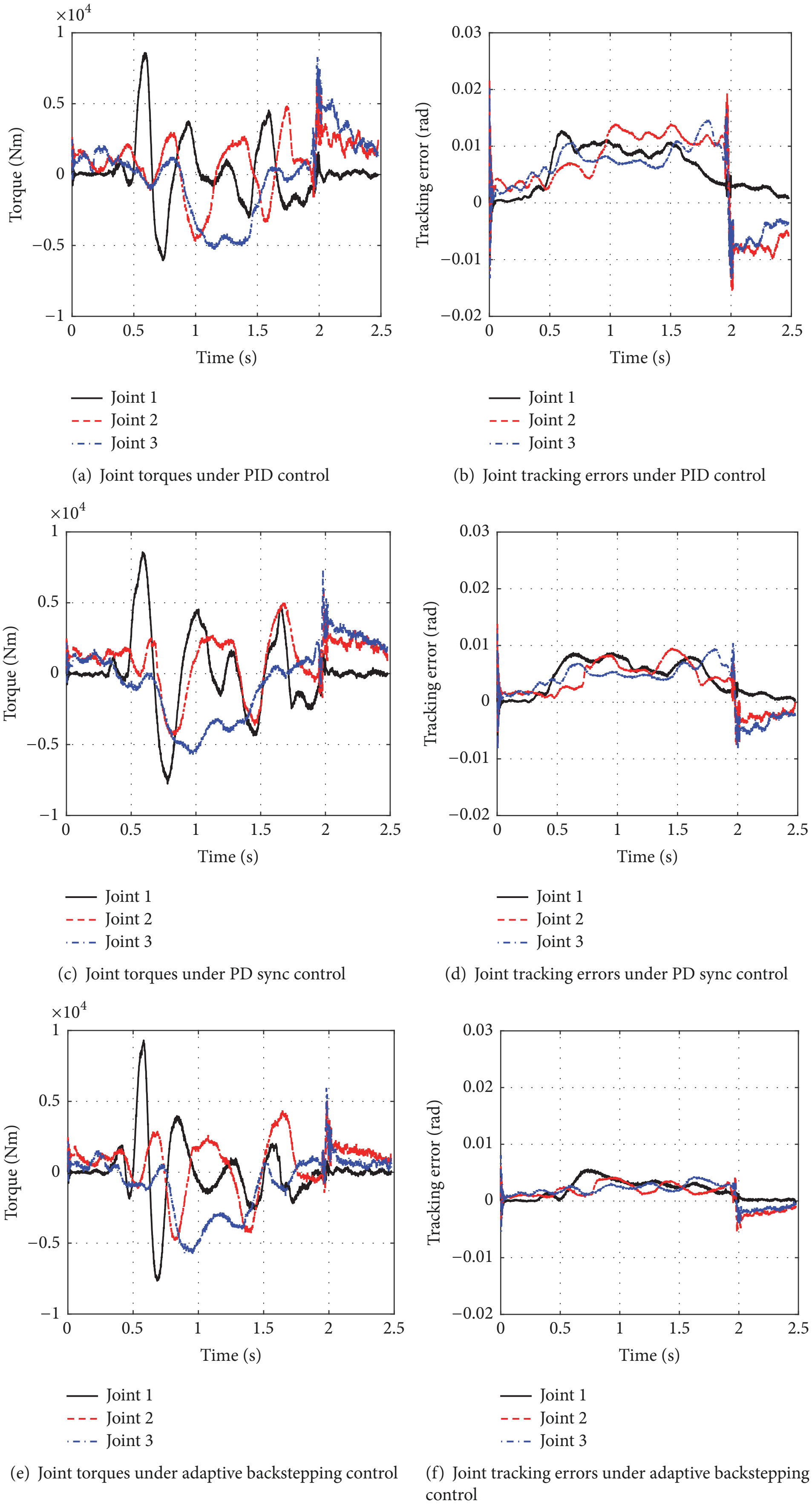
(c) Joint torques under PD sync control

(d) Joint tracking errors under PD sync control

(e) Joint torques under adaptive backstepping control

(f) Joint tracking errors under adaptive backstepping control

Figure 11: Experimental results of different control strategies in Case 1.



(a) Joint torques under PID control

(b) Joint tracking errors under PID control

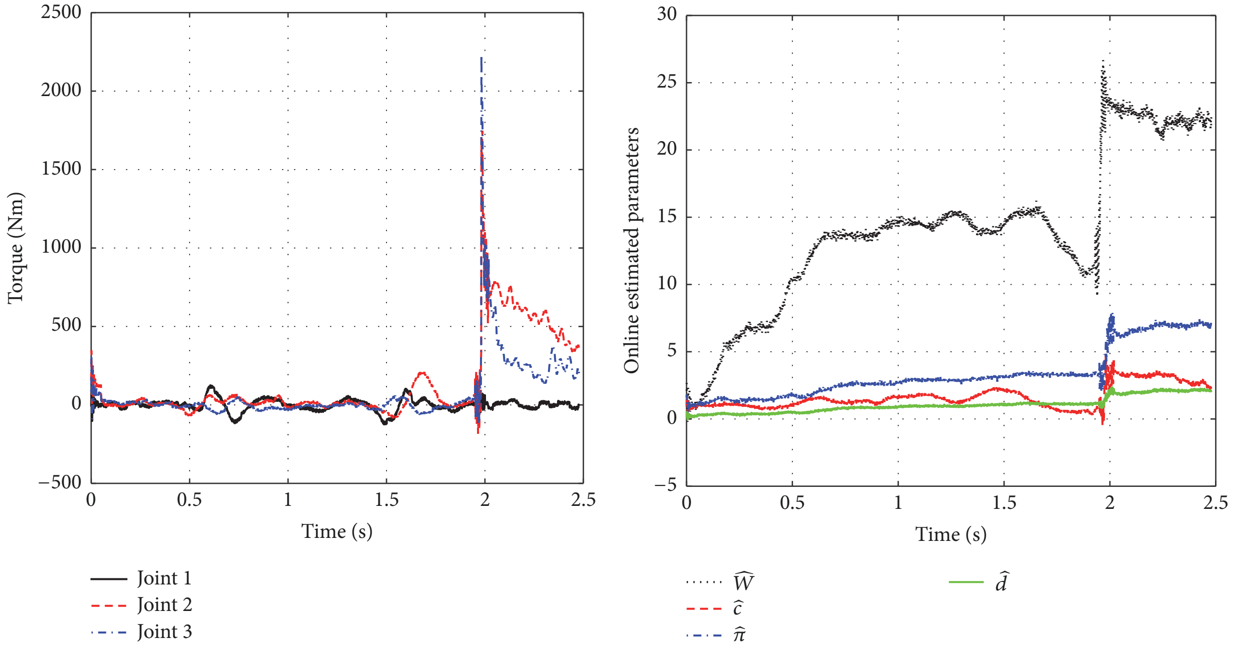
(c) Joint torques under PD sync control

(d) Joint tracking errors under PD sync control

(e) Joint torques under adaptive backstepping control

(f) Joint tracking errors under adaptive backstepping control

Figure 12: Experimental results of different control strategies in Case 2.



(a) Robust term $\hat{\tau}_d$

(b) Online estimated parameters

FIGURE 13: Robust term and online estimated parameters.

strategy and the online update adjustment of the estimated parameters are shown in Figure 13. It can be seen that the robust term $\hat{\tau}_d$ and the estimated parameters $\widehat{W}$, $\hat{c}$, $\hat{\pi}$, and $\hat{d}$ can be adaptively adjusted online according to the modeling error of the robot dynamics and the external disturbances to significantly enhance the system robustness.

6. Conclusion

This work deals with multiobjective trajectory optimization and adaptive backstepping control strategy of the rubber unstacking robot during the recycling process; the main contents of the research include the following aspects:

(1) Considering the mechanical analysis during the inserting process, the Hunt-Crossley nonlinear model based dynamic model of the rubber unstacking robot is deduced, which can be used for trajectory planning and controller design.

(2) Aiming at the performance indicators of operational efficiency, trajectory smoothness, and energy consumption, the motion planning problem is transformed into multiobjective optimization problem. The position of the floating via points of the B-spline trajectory and the running time of each point are optimized, and the NSGA genetic algorithm is used to solve the problem. The optimization results show that compared to the standard trajectory the above indicators were raised by 29.17%, 15.85%, and 12.57%, respectively.

(3) The proposed adaptive backstepping control method based on RFWNN is robust to external torque disturbances and can overcome the effects of the unknown dynamic parameters and approximation error. The backstepping strategy is designed based on Lyapunov synthesis, which can guarantee the stability of the control system. Through experimental analysis, compared with other control methods, the validity of the proposed control strategy is verified.

(4) As the rubber in the heat refining process, adhesion phenomenon will be produced between the rubber blocks due to local melting, which would result in rubbers difficult to be separated. So the future work will be extended to deal with the control problem of the unknown viscoelastic force between rubber blocks during the rubber separation process.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the financial support from the Application Technology Research and Development Program of Heilongjiang Province (GY2016ZB0068) and National High Technology Research and Development Program of China (2015AA043003).

References

[1] A. Gasparetto, A. Lanzutti, R. Vidoni, and V. Zanotto, “Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 164–181, 2012.

[2] V. Zanotto, A. Gasparetto, A. Lanzutti, P. Boscariol, and R. Vidoni, "Experimental validation of minimum time-jerk algorithms for industrial robots," *Journal of Intelligent & Robotic Systems*, vol. 64, no. 2, pp. 197–219, 2011.

[3] Y. Liu, L. Liang, H. Han, and S. Zhang, "A Method of Energy-Optimal Trajectory Planning for Palletizing Robot," *Mathematical Problems in Engineering*, vol. 2017, pp. 1–10, 2017.

[4] A. Piazzi and A. Visioli, "Global minimum-jerk trajectory planning of robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 140–149, 2000.

[5] T. Chettibi, H. E. Lehtihet, M. Haddad, and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *European Journal of Mechanics - A/Solids*, vol. 23, no. 4, pp. 703–715, 2004.

[6] M. H. Korayem, M. Bamdad, H. Tourajizadeh, A. H. Korayem, and S. Bayat, "Analytical design of optimal trajectory with dynamic load-carrying capacity for cable-suspended manipulator," *The International Journal of Advanced Manufacturing Technology*, vol. 60, no. 1-4, pp. 317–327, 2012.

[7] F. J. Abu-Dakka, F. Rubio, F. Valero, and V. Mata, "Evolutionary indirect approach to solving trajectory planning problem for industrial robots operating in workspaces with obstacles," *European Journal of Mechanics - A/Solids*, vol. 42, pp. 210–218, 2013.

[8] F. L. Lewis, A. Yesildirak, and S. Jagannathan, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, 1998.

[9] L. Y. Wang, T. Y. Chai, and L. F. Zhai, "Neural-network-based terminal sliding-mode control of robotic manipulators including actuator dynamics," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3296–3304, 2009.

[10] J. I. Mulero-Martinez, "Robust GRBF static neurocontroller with switch logic for control of robot manipulators," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1053–1064, 2012.

[11] F.-J. Lin, H.-J. Shieh, and P.-K. Huang, "Adaptive wavelet neural network control with hysteresis estimation for piezo-positioning mechanism," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 2, pp. 432–444, 2006.

[12] G. Tong and M. Lin, "Alignment condition-based robust adaptive iterative learning control of uncertain robot system," *Advances in Mechanical Engineering*, vol. 2014, Article ID 201317, pp. 1–8, 2014.

[13] V. Panwar and N. Sukavanam, "Design of optimal hybrid position/force controller for a robot manipulator using neural networks," *Mathematical Problems in Engineering*, vol. 2007, Article ID 65028, pp. 32–34, 2007.

[14] I. Chairez, "Wavelet differential neural network observer," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 20, no. 9, pp. 1439–1449, 2009.

[15] S. J. Yoo, J. B. Park, and Y. H. Choi, "Adaptive dynamic surface control of flexible-joint robots using self-recurrent wavelet neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 6, pp. 1342–1355, 2006.

[16] M. Boukattaya, M. Jallouli, and T. Damak, "On trajectory tracking control for nonholonomic mobile manipulators with dynamic uncertainties and external torque disturbances," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1640–1647, 2012.

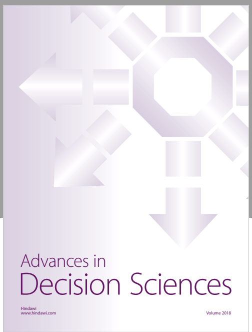
[17] K. H. Hunt and F. R. E. Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *Journal of Applied Mechanics*, vol. 42, no. 2, pp. 440–445, 1975.

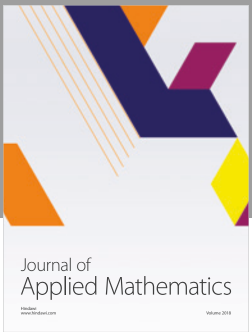
[18] A. Gasparetto and V. Zanotto, "Optimal trajectory planning for industrial robots," *Advances in Engineering Software*, vol. 41, no. 4, pp. 548–556, 2010.

[19] W. Wang, W. Wang, W. Dong, H. Yu, Z. Yan, and Z. Du, "Dimensional optimization of a minimally invasive surgical robot system based on NSGA-II algorithm," *Advances in Mechanical Engineering*, vol. 7, no. 2, pp. 1–11, 2015.

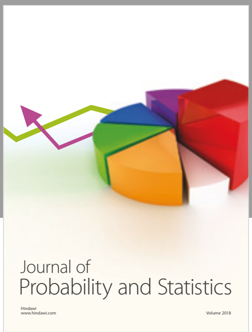
[20] F. L. Lewis, A. Yeşildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 7, no. 2, pp. 388–399, 1996.

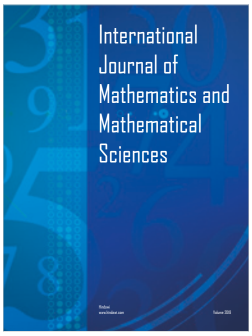
Advances in
Operations Research


Advances in
Decision Sciences

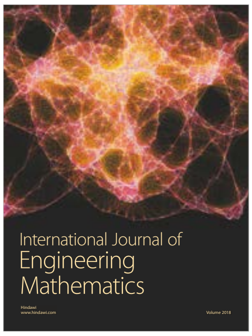

Journal of
Applied Mathematics

The Scientific
World Journal

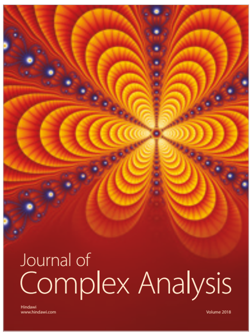

Journal of
Probability and Statistics

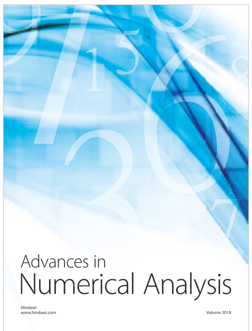

International
Journal of
Mathematics and
Mathematical
Sciences

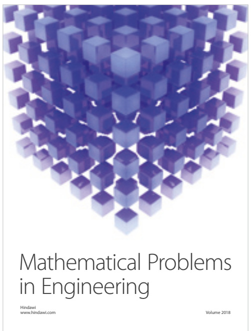
Journal of
Optimization

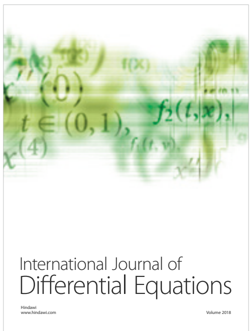

International Journal of
Engineering
Mathematics

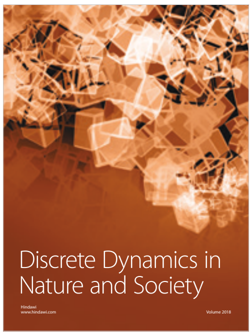
International Journal of
Analysis

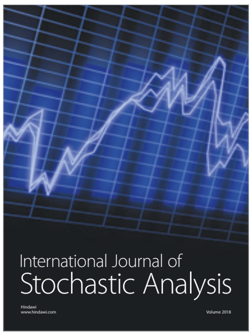

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

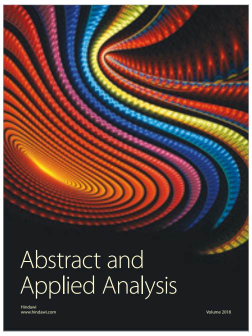

International Journal of
Differential Equations

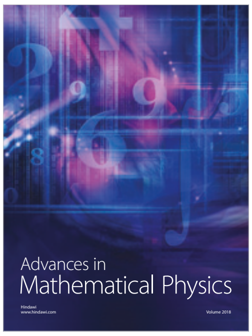

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics