

Dynamic Modeling and Torque Feedforward based Optimal Fuzzy PD control of a High-Speed Parallel Manipulator

Zirong Lin, Chuangchuan Cui, Guanglei Wu*

School of Mechanical Engineering, Dalian University of Technology, Dalian, China

Email: gwu@dlut.edu.cn

*Corresponding Author

Abstract—Dynamic modeling and control of high-speed parallel manipulators are of importance due to their industrial applications deployed in production lines. However, there are still a number of open problems, such as the development of a precise dynamic model to be used in the model-based control design. This paper presents a four-limb parallel manipulator with Schönflies motion and its simplified dynamic modeling process. Then, in order to fix the issue that computed torque method control (CTC) will spend a lot of time to calculate dynamic parameters in real-time, offline torque feedforward-based PD (TFPD) control law is adopted in the control system. At the same time, fuzzy logic is also used to tune the gains of PD controller to adapt to the variation of external disturbance and compensate the un-modeled uncertainty. Additionally, bottom widths of membership functions of fuzzy controller are optimized by bat algorithm. Finally, three controllers of CTC, TFPD and bat algorithm-based torque feedforward fuzzy PD controller (BA-TFFPD) are compared in trajectory tracking simulation. From the result, compared with TFPD and CTC, BA-TFFPD can lead faster transient response and lower tracking error, which prove the validity of BA-TFFPD.

Keywords—High-Speed parallel manipulator; Dynamic model; Torque feedforward; PD control; Bat algorithm; Fuzzy logic

I. INTRODUCTION

DELTA robot and SCARA robots dedicate to pick-and-place tasks in the industrial production line [1]. On the one hand, the translational DELTA robot, whose mobile platform can only realize translations in three directions, is difficult to meet more complex task with object realignment. DELTA robot with an additional actuator on the end-effector or the fourth RPUR(R, P and U stand for revolute, prismatic and universal joints, respectively) limb to connect the centers of the base and mobile platform still has some drawbacks as revealed in [2]. On the other hand, the serial SCARA robots cannot work in high speed due to its large dynamic inertia, though it can produce three independent translations and one rotation around an axis of fixed direction, namely, Schönflies motion. Thanks to its high speed, low inertia and high accuracy, parallel Schönflies motion manipulator very popular in industrial applications.

Up to date, many types of Schönflies motion generators have been proposed, such as H4 [3], I4 [4], Par4 [5]. And amongst others [6]-[9], the Adept Quattro robot and the Veloce. Robot are famous for their commercial application. Compared to the parallel manipulators with only single

platform [10][11], the four-limb robots, whose mobile platform consist of two sub-platforms, can realize more large range of rotation motion, thanks to well-configuration of sub-platforms. Moreover, some pick-and-place tasks of those objects with irregular shapes require full-circle rotation. Rotational motion can be realized by the relative movement in parallel between the two sub-platforms of the robot, such as H4, Quattro and Par4. As usual, pulley-belt mechanism [4], rack-pinion mechanism [5], gears and screw mechanism [12] can be adopted to enhance the rotational capability of manipulator. For this article, helix joint is employed to enhance the rotational capability of end effector.

To apply the manipulator into manufacture, designing a control system, with high tracking accuracy and well effected error compensation, will be a crucial work. For this goal, a number of control laws are designed by researchers. Model-free control technique is usually employed in trajectory tracking task. PID as a classical model-free controller is used widely in industrial manufacturing thanks its simple structure. In fact, conventional PID cannot adapt to variation of external disturbance and un-modeled uncertainty. So other type of PID controllers, such as fractional-order PID [13]-[18], is investigated. W. A. Shutnan and T. Y. Abdalla [19] presented the fractional-order PID controller for path tracking task of robot manipulator to obtain better performance than classical PID controller. J. Viola and L. Angel [20] used fractional-order PID controller with the feedback linearization technique to design the tracking system of parallel manipulator, which get better tracking performance than integer order PID. On the other hand, combining other algorithm also can enhance the tracking performance of PID controller [21]-[23]. V. Nguyen et al. [24] proposed radial basic function neural networks based PID (APID-RBFNs), which is used in tracking task of two degree of freedom parallel manipulator. P. C. Sau [25] compared fuzzy fractional-order PID (FFOPID) controller and fractional-order PID (FOPID) for the control system of three-arm manipulator. And the better tracking performance can be obtained by FFOPID.

Differ from model-free control technique, model-based control technique needs dynamic model of manipulator to design the controller. As a classical method, computed torque control (CTC) is widely employed in industrial robot and parallel manipulator [26]. But classical CTC has many



drawbacks, such as higher computational cost and the difficulty of getting precise dynamic model [27][28]. For those reason, many modified algorithms. J. Viola and L. Angel [29] based computed torque control strategy design a fraction-order PID controller for two degree of freedom manipulator to reduce the error generated during tracking task. M. Alhaddad et al. [30] presented an adaptive LQ-based computed torque controller (LQ-CTC), which can improve the accuracy of robotic manipulator carrying unknown load.

An excellent controller needs well parameter tuning. And a lot of metaheuristic parameter searching algorithm, such as particle swarm optimization (PSO) algorithm [31], bat algorithm (BA) [32], grey wolf optimization algorithm (GWO) [34], genetic algorithms (GA) [35][36] and cuckoo search (CS) algorithm [37] are adopted in optimization of parameter of controller for its powerful performance for tough optimization problems. T. M. Ahmed et al. [16] used PSO to optimize the parameters of fractional-order PID controller. Liu et al. [38] designed fuzzy controller, whose bottom width of membership functions were tuning by improved PSO, to adjust PID gains. And the result of its simulation can significantly reduce tracking error and contour error of the three-axis machine tool. F. Liu et al. [39] use bat algorithm to optimize the center and width vectors of membership functions of fuzzy control system adopted in autonomous mobile robots and lead a better tracking performance.

In this paper, a parallel manipulator with SCARA motion [8] is presented together with the dynamic modeling and control design. Architecture of the manipulator is introduced at first. Then, principle of virtual work is adopted for dynamic modeling and evaluated by simulating in MATLAB/Simulink. For the next part, the torque feedforward-based PD controller is proposed. At the same time, to fix the issue of parameters tuning, fuzzy controller, whose bottom width of membership functions are optimized by bat algorithm, is adopted to adjust PD gains in real-time. For simulation, a numerical manipulator model is built and the normal trajectory is used to evaluate the performance of trajectory tracking. Finally, the conclusion will be summarized by compared performance of the bat algorithm-based torque feedforward fuzzy PD (BA-TFFPD) controller to computed torque controller (CTC) and torque feedforward PD (TFPD) controller.

II. ROBOT ARCHITECTURE

Fig. 1(a) depicts the simplified CAD model of the parallel robot under study. The robot has four limbs, to connect the base frame and mobile platform, of which each limb consists of an inner arm and an outer arm (parallelogram, a.k.a., Π joint). The mobile platform, depicted in Fig. 1(b), can be split into two subparts, namely, the upper and lower platforms that are connected by helical pair to convert the relative translation along vertical axis of two subparts into rotational movement about the axis. The rotational capability can be enhanced by selecting an appropriate screw lead that is denoted as h .

As shown in Fig. 1(a), the origin of global coordinate frame (x, y, z) is located at the geometric center of the base frame. The z -axis is in vertical direction, and x -axis is parallel

to segment A_1A_2 that is the line between actuated joint 1 and 2. The moving coordinate frame (X, Y, Z) is located at geometric center of the lower platform and the Z -axis and X -axis are also parallel to z and x -axis, respectively. Moreover, \mathbf{i} , \mathbf{j} and \mathbf{k} are the unit vectors of x , y and z -axis, respectively.

The i th limb, depicted in Fig. 2, consists of inner arm and outer arm. A_i stand for the center of actuated joint. B_i and C_i represent the center of two spherical joints at two ends. In addition, length of inner and outer arms can be written in b and l respectively. The axis of rotation of the i th actuated joint is parallel to unit vector $\mathbf{u}_i = \mathbf{R}_i \mathbf{k}$, where \mathbf{R}_i stands for the rotation matrix from global coordinate frame to i th actuated joint.

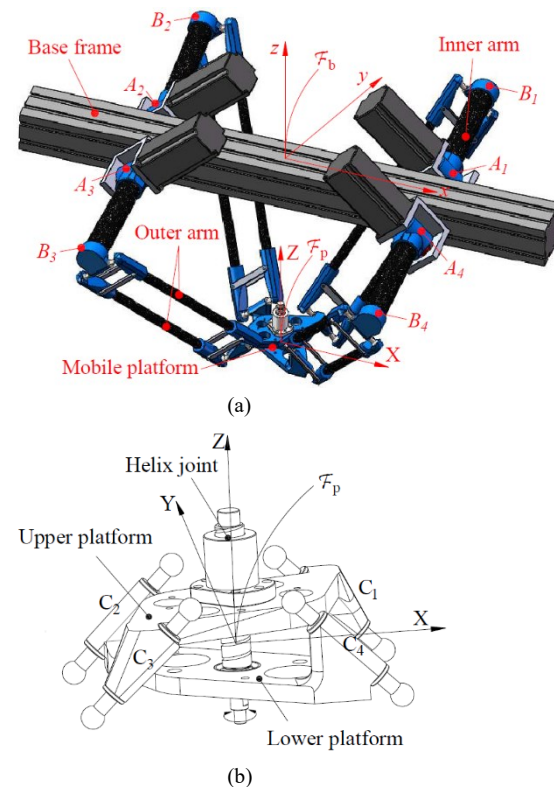


Fig. 1. Architecture of the manipulator: (a) CAD model; (b) Architecture of mobile platform

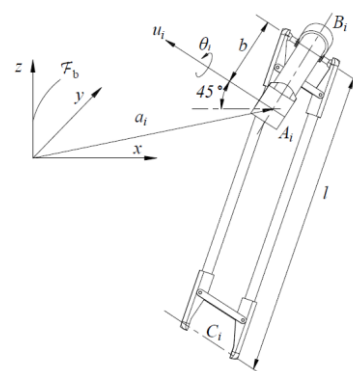


Fig. 2. The parameterization of i th limb.

III. MANIPULATOR DYNAMICS

Dynamic modeling of the robotic systems has been extensively studied, wherein the main modeling approaches include the Lagrangian formulation, the Newton–Euler

equations and the principle of virtual work. Here, the principle of virtual work is adopted to model the dynamic model of the parallel manipulator.

A. Kinematic Jacobian

The closed-form solution [9] of the robot geometry has been extensively studied in the literature, which will be skipped in this paper. For each limb, depicted in Fig. 2, geometric constraints as below must be satisfied.

$$\|c_i - b_i\|^2 = l^2, \quad i = 1, \dots, 4 \quad (1)$$

where, c_i and b_i are the position vectors in the global coordinate frame $oxyz$.

Differentiating (1) with respect to time yields

$$J_\chi \dot{\chi} = J_\theta \dot{\theta} \quad (2)$$

with $\chi = [x \ y \ z \ \phi]$, $\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]$

$$\begin{cases} J_\chi = [J_1 \ J_2 \ J_3 \ J_4]^T \\ J_i = \left[(c_i - b_i)^T \quad (c_i - b_i)^T \text{mod}(i,2) \frac{h}{2\pi} k \right]^T \end{cases} \quad (3)$$

$$\begin{cases} J_\theta = \text{diag}([w_1 \ w_2 \ w_3 \ w_4]) \\ w_i = (c_i - b_i)^T u_i \times (b_i - a_i) \end{cases} \quad (4)$$

where, J_χ and J_θ are the forward and backward Jacobian matrices, respectively.

As long as the forward kinematic Jacobian is nonsingular, the kinematic Jacobian matrix is obtained as:

$$J = J_\chi^{-1} J_\theta \quad (5)$$

Because of the mobile platform consists of two sub-platforms, the upper platform can translate in vertical direction with respect to the lower one, which introduces different velocities between the end-effector and sub-platforms. Besides, the Jacobian matrix between upper platform and actuated joints is also different from that of the lower one.

The velocities of the upper and lower platforms can be expressed as

$$\dot{\chi}_{up} = T_{up} \dot{\chi}; \quad T_{up} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \frac{h}{2\pi} \\ & & & 0 \end{bmatrix} \quad (6)$$

$$\dot{\chi}_{low} = T_{low} \dot{\chi}; \quad T_{low} = \text{diag}([1 \ 1 \ 1 \ 0]) \quad (7)$$

where $\dot{\chi}_{up}$, $\dot{\chi}_{low}$ and $\dot{\chi}$ represent the velocity vector of upper platform, lower platform and end-effector, respectively. And the Jacobian matrices of the two sub-platforms can be written below

$$J_{\chi,up} \dot{\chi}_{up} = J_\theta \dot{\theta}; \quad J_{\chi,low} \dot{\chi}_{low} = J_\theta \dot{\theta} \quad (8)$$

Where

$$\begin{cases} J_{\chi,up} = [j_{up,1} \ j_{up,2} \ j_{up,3} \ j_{up,4}]^T \\ j_{up,i} = \left[(c_i - b_i)^T \quad -\text{mod}(i+1,2)(c_i - b_i)^T \frac{h}{2\pi} k \right]^T \end{cases} \quad (9)$$

(10)

$$\begin{cases} J_{\chi,low} = [j_{low,1} \ j_{low,2} \ j_{low,3} \ j_{low,4}]^T \\ j_{low,i} = \left[(c_i - b_i)^T \quad \text{mod}(i,2)(c_i - b_i)^T \frac{h}{2\pi} k \right]^T \end{cases} \quad (10)$$

Sequentially, the kinematic Jacobian matrices of the two sub-platforms are obtained as

$$J_{up} = J_{\chi,up}^{-1} J_\theta; \quad J_{low} = J_{\chi,low}^{-1} J_\theta \quad (11)$$

Finally, the relationship between the velocities of sub-platforms and actuated joints are derived as

$$\dot{\chi}_{up} = J_{up} \dot{\theta}; \quad \dot{\chi}_{low} = J_{low} \dot{\theta} \quad (12)$$

B. Dynamic Model

Before modeling, some assumptions are made, where the torque caused by rod inertia of the outer arm should be small enough to be ignored. Moreover, their mass is simplified as two lumped masses located at the two ends, as shown in Fig. 3.

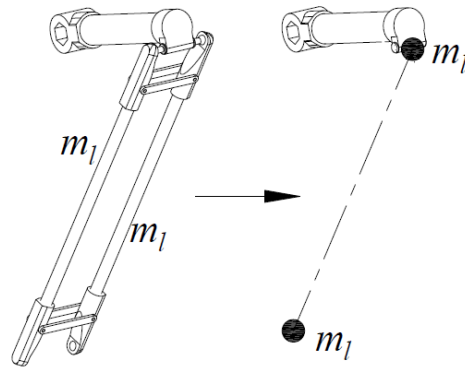


Fig. 3. Simplification of the rod.

According to the Principle of virtual work, sum of the work done by the external and internal forces should be zero. The external forces include actuating torque τ and the gravities M_b , $M_{p,up}$ and $M_{p,low}$ of the inner arm, upper platform and lower platform. On the other hand, F_b , $F_{p,up}$ and $F_{p,low}$ are the inertia forces of inner arm and sub-platforms, proportional to the internal forces. Then, the equation of dynamics can be formed as

$$\begin{aligned} \tau^T \delta_\theta + M_b^T \delta_\theta + M_{p,up}^T \delta_{\chi,up} + M_{p,low}^T \delta_{\chi,low} \\ - F_b^T \delta_\theta - F_{p,up}^T \delta_{\chi,up} - F_{p,low}^T \delta_{\chi,low} = 0 \end{aligned} \quad (13)$$

where $\tau = [\tau_1 \ \tau_2 \ \tau_3 \ \tau_4]^T$ is the torques vector exerted on the actuated joints. $\delta_\theta = [\delta_{\theta_1} \ \delta_{\theta_2} \ \delta_{\theta_3} \ \delta_{\theta_4}]^T$ and

$\delta_{\chi,up/low} = [\delta_x \ \delta_y \ \delta_z \ \delta_\phi]^T$ represent virtual displacement of actuated joints and sub-platforms, respectively. Moreover,

$$M_b = \left(\frac{1}{2}m_b + m_l\right)gb \cdot I_4 \cdot [\cos\theta_1 \ \cos\theta_2 \ \cos\theta_3 \ \cos\theta_4]^T \quad (14)$$

$$M_{p,up/low} = \begin{bmatrix} 0 & 0 & -(m_{p,up/low} + 2m_l)g & 0 \end{bmatrix}^T \quad (15)$$

$$F_b = I_b \cdot \ddot{\theta}; \quad I_b = \left(\frac{1}{3}m_b b^2 + m_l b^2\right) \cdot I_4 \quad (16)$$

$$\begin{cases} F_{p,up/low} = M_{p,up/low} \cdot \ddot{\chi} \\ M_{p,up/low} = (m_{p,up/low} + 2m_l) \cdot \text{diag}([1 \ 1 \ 1 \ 0]) \end{cases} \quad (17)$$

for which the mass properties can be listed in TABLE I.

TABLE I. THE MASS PROPERTIES OF THE ROBOT IN (14)-(17)

Symbol	Description	Value
m_b	Mass of inner arm	0.073 (kg)
m_l	Mass of rod	0.421 (kg)
$M_{p,up}$	Mass of upper platform	0.955 (kg)
$M_{p,low}$	Mass of lower platform	1.180 (kg)
I_4	4x4 identity matrix	$\text{diag}([1 \ 1 \ 1 \ 1])$

Differentiating (12) with respect to time yields

$$\ddot{\chi}_{up} = \dot{J}_{up} \dot{\theta} + J_{up} \ddot{\theta}; \quad \ddot{\chi}_{low} = \dot{J}_{low} \dot{\theta} + J_{low} \ddot{\theta} \quad (18)$$

Analogue to the time differential equations of motion, small-amplitude displacements can deduce as following

$$\delta_{\chi,up} = J_{up} \delta_\theta; \quad \delta_{\chi,low} = J_{low} \delta_\theta \quad (19)$$

Combing (13) - (19), (13) can be rewritten as

$$\begin{aligned} (\tau^T + M_b^T + M_{p,up}^T J_{up} + M_{p,low}^T J_{low} \\ - F_b^T - F_{p,up}^T J_{up} - F_{p,low}^T J_{low}) \delta_\theta = 0 \end{aligned} \quad (20)$$

For any virtual displacement δ_θ , Eq. (16) should be satisfied

$$\begin{aligned} \tau = -M_b - J_{up}^T M_{p,up} - J_{low}^T M_{p,low} \\ + F_b + J_{up}^T F_{p,up} + J_{low}^T F_{p,low} \end{aligned} \quad (21)$$

After rearranging (21), the simplified dynamic model can be obtained as

$$\tau = M(\theta) \ddot{\theta} + C(\theta, \dot{\theta}) \dot{\theta} + G(\theta) \quad (22)$$

with

$$M(\theta) = J_{low}^T M_{p,low} J_{low} + J_{up}^T M_{p,up} J_{up} + I_b \quad (23)$$

$$C(\theta, \dot{\theta}) = J_{low}^T M_{p,low} \dot{J}_{low} + J_{up}^T M_{p,up} \dot{J}_{up} \quad (24)$$

$$G(\theta) = -J_{low}^T M_{p,low} g - J_{up}^T M_{p,up} g - M_b g \quad (25)$$

C. Evaluation of the dynamic model

The dynamic model described by (22) can be evaluated by Simscape module of MATLAB/Simulink. Through comparing the actuating torques calculating by simplified model and simulated by Simscape model built completely

and without any simplification, the difference between simplified model and complete model can be observed for assessment.

For the simulation, as shown in Fig. 4, the trajectory with horizontal displacement of 305 mm, vertical displacement of 25mm and rotational motion of 360 degree is adopted. Then, 4-5-6-7 polynomial is used in trajectory design. Fig. 5 depicts the displacement, velocity and acceleration of x , y , z -axis and rotational motion. The comparison between the torque of analytical model (solid line) and simulation model (dotted line) is shown in Fig. 6, from which it can be seen that the computation accuracy of the simplified model is very close to that of the complete model, despite the acceptable errors. From Fig. 7, which show the absolute torque error of two models, it can be seen that the max error of 0.8 N*m for joint 1, 1.7 N*m for joint 2, 0.9 N*m for joint 3 and 2.3 N*m for joint 4. And due to the inertia of the rods are ignored, those max errors occur at the point of max acceleration of each joint.

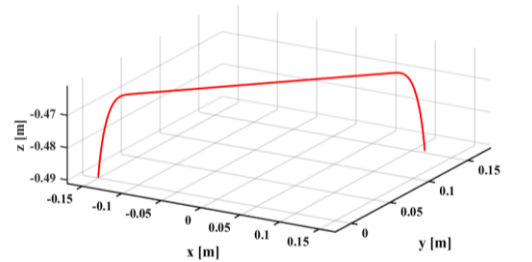


Fig. 4. Simulation trajectory.

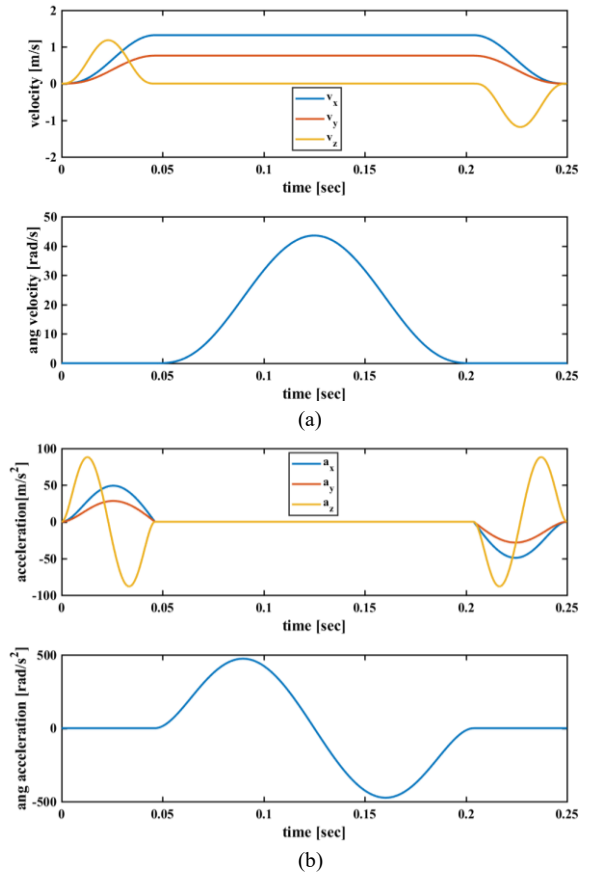


Fig. 5. Motion profiles of the trajectory: (a) Velocity in three-axis and rotational direction. (b) Acceleration in three-axis and rotational direction.

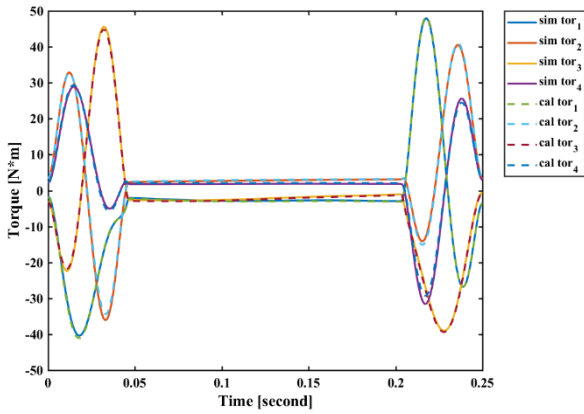


Fig. 6. Actuated torque of each joint

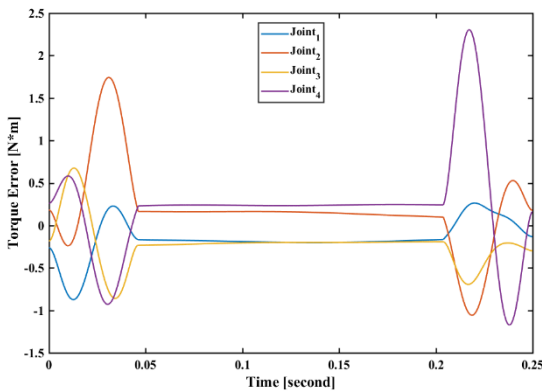


Fig. 7. Torque error of each joints.

IV. CONTROLLER DESIGN

In this section, bat algorithm-based torque feedforward PD (BA-TFFPD) controller will be introduced. An independent PD controller for each will be employed for main controller. Then, the torque calculated offline will be introduced into the PD control system to improve the tracking performance. Moreover, in order to adapt to time-varying disturbance, fuzzy logic will be used to tune the gains of PD controller in real-time. And, bat algorithm also will be used to optimize bottom widths of membership functions of the fuzzy controller.

A. Torque feedforward based PD control

The complete dynamic model should be the sum of simplified model and un-modeled uncertainty $f(t)$, namely,

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) + f(t) \quad (26)$$

The tracking error can be defined as

$$e = \theta - \theta_d \quad (27)$$

where $\theta_d = [\theta_{d1} \ \theta_{d2} \ \theta_{d3} \ \theta_{d4}]^T$ represent the desired angle vector on the trajectory. And $e = [e_1 \ e_2 \ e_3 \ e_4]^T$.

Sequentially, ignoring the coupling of each joints, it can be designed an independent PD controller for each joint.

$$u_i = k_{p_i} e_i + k_{d_i} \dot{e}_i ; i = 1, \dots, 4 \quad (28)$$

For convenience, (28) can be reformed to matrix form:

$$u = K_p e + K_d \dot{e} \quad (29)$$

With

$$\begin{cases} K_p = \text{diag}([k_{p1} \ k_{p2} \ k_{p3} \ k_{p4}]) \\ K_d = \text{diag}([k_{d1} \ k_{d2} \ k_{d3} \ k_{d4}]) \end{cases} \quad (30)$$

where K_p and K_d are the proportional and derivative gains matrices, respectively.

Introduce the torque calculated offline, the control law can be summarized as

$$\begin{cases} \tau = \tau_d - u \\ \tau_d = M(\theta_d)\ddot{\theta}_d + C(\theta_d, \dot{\theta}_d)\dot{\theta}_d + G(\theta_d) \end{cases} \quad (31)$$

Where, $\tau_d = [\tau_{d1} \ \tau_{d2} \ \tau_{d3} \ \tau_{d4}]^T$ is the offline torque vector.

According to (30) and (31), the system equation can be obtained as

$$K_p e + K_d \dot{e} - \Delta = 0 \quad (32)$$

Where, $\Delta = \tau_d - \tau_a - f(t)$, is un-modeled uncertainty vector, and $\tau_a = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta)$ represent actual torque during tracking task.

From (32), obviously, the tracking error e can converge to zero in finite time by choose appropriate K_p and K_d . And the method of trial and error can be used in gains selection.

B. Parameter tuning strategy of PD controller

Tracking error cannot maintain in a low level, when the gains of PD controller are fixed. Fuzzy logic is effective way to adaptively tune parameter online. But, it is a hard work to design fuzzy controller appropriately, in which bottom widths should be designed according to the experiential knowledge summarized by proficient or trial and error method. However, for trial and error method, it is not a sensible way to spend lot of time on these repetitive work that is more suitable for computer. Therefore, optimization algorithm, such as bat algorithm (BA), particle swarm optimization (PSO) algorithm and cuckoo search (CS) algorithm, can be considered to employ in parameter tuning. For those reasons above, fuzzy logic based parameter tuning strategy will be presented in this part. At the same time, bat algorithm will be employed to optimize bottom widths of the fuzzy controller.

During trajectory tracking task, gains of PD controller should be adjusted in real-time to adapt to different situations. The strategy of parameters adjustment can be summarize as

$$K'_p = K_p + \text{diag}([\Delta k_{p1} \ \Delta k_{p2} \ \Delta k_{p3} \ \Delta k_{p4}]) \quad (33)$$

$$K'_d = K_d + \text{diag}([\Delta k_{d1} \ \Delta k_{d2} \ \Delta k_{d3} \ \Delta k_{d4}]) \quad (34)$$

Where K_p, K_d defined in (29) are base gain and set to be constant, while Δk_{p_i} and Δk_{d_i} are adjustable gains produced by fuzzy controllers. The sum of base gains and adjustable

gains are composite gains K'_p and K'_d , which can be adjusted in real-time with tracking error changes. Generally, four fuzzy controllers are needed to tune the parameter of PD controller for each joint. In fact, because of the symmetry of the manipulator, only two fuzzy controllers need to design for four joints: a fuzzy controller for joint 1 and 3 and another controller for joint 2 and 4. Then we will introduce the design process of one among them in this part.

For fuzzy controller, as shown in Fig. 8, it can be summarized as 4 parts: input fuzzification, fuzzy rule base, inference engine and output defuzzification. And the design process can be summarized as follow:

First of all, for each joint, it needs a fuzzy controller to tuning both adjustable gains Δk_{pi} and Δk_{di} in real-time. Tracking error e_i and its differential signal \dot{e}_i can be considered as inputs of the fuzzy controller, because they can reflect the tracking performance. Then, adjustable gain Δk_{pi} and Δk_{di} are outputs of fuzzy controller.

The fuzzy set of input and output variables are defined as

$$A = B = D = \{NB \quad NM \quad NS \quad Z \quad PS \quad PM \quad PB\} \quad (35)$$

$$C = \{Z \quad PS \quad PM \quad PB\} \quad (36)$$

Where A , B are the fuzzy sets of input variables e and \dot{e} , respectively. C , D are the fuzzy sets of output variables Δk_p and Δk_d . And NB: negative big; NM: negative medium; NS: negative small; Z: zero; PB: positive big; PM: positive medium; PS: positive small;

Then, before the inference engine, inputs of the controller, which have crisp value, need to transform to membership function expression, which is call fuzzification. And, in this paper, the triangle membership function, which can be expressed in (37), is adopted for input and output variables. Fig. 9 (a) and (b) show the degree of membership about different value of input and output variables.

$$\mu(x) = \begin{cases} 0, & x < q + \Delta q \\ \frac{x - q + \Delta q}{\Delta q}, & q - \Delta q \leq x \leq q \\ \frac{q + \Delta q - x}{\Delta q}, & q \leq x \leq q + \Delta q \\ 0, & q + \Delta q < x \end{cases} \quad (37)$$

Where, x stands for those input and output variables, it can be e , \dot{e} , Δk_p and Δk_d . And q and Δq represent vertex and half the length of the base of the triangle, respectively. For example, as depicted in Fig. 9, $q = 0$ and $\Delta q = 0.1$ for fuzzy linguistic terms "Z".

For the next step, fuzzy rules, the most important process, are designed. Generally, it should set different rules for different control system. For the PD controller of i th joint, four situations, as shown in Fig. 10, possibly meet during tracking. And the laws of gain tuning can be summarized as follow case:

Case 1 (point A): when e_i is big and \dot{e}_i is in the direction close to zero, we should increase k_{pi} and decrease k_{di} to enforce tracking error close to zero in a short time.

Case 2 (point B): when e_i is small and \dot{e}_i is in the direction close to zero, decrease k_{pi} and increase k_{di} to enforce tracking error close to zero in a slow speed.

Case 3 (point C): when e_i is small and \dot{e}_i is in the direction away from zero, increase k_{pi} and decrease k_{di} to enforce tracking error return to zero in a short time.

Case 4 (point D): when e_i is big and \dot{e}_i is in the direction away from zero, increase k_{pi} and decrease k_{di} to enforce tracking error return to zero in a short time.

Based on above tuning law, fuzzy rule of the fuzzy controllers, whose output are Δk_{pi} and Δk_{di} , can be obtain as TABLE II. What's more, every rule can be written as "IF-THEN" expression. For instance,

$$\begin{aligned} \text{IF } e_i \text{ is } A_j \text{ \& } \dot{e}_i \text{ is } B_j \\ \text{THEN } \Delta k_{pi} \text{ is } C_j \text{ \& } \Delta k_{di} \text{ is } D_j \end{aligned} \quad (38)$$

Where, A_j , B_j , C_j and D_j are subsets of fuzzy set A , B , C and D .

The last step is that according to the fuzzy rules to infer and defuzzification the output variables Δk_{pi} and Δk_{di} .

Then, for outputs, according to centroid method, it can be expressed as

$$\Delta k_{pi} = \frac{\sum_{i=1}^m h_C^i \mu_C^i(\Delta k_{pi})}{\sum_{i=1}^m \mu_C^i(\Delta k_{pi})} \quad (39)$$

$$\Delta k_{di} = \frac{\sum_{i=1}^m h_D^i \mu_D^i(\Delta k_{di})}{\sum_{i=1}^m \mu_D^i(\Delta k_{di})} \quad (40)$$

Where $\mu_C^i(\Delta k_{pi})$ and $\mu_D^i(\Delta k_{di})$ represent membership function of Δk_{pi} and Δk_{di} for i th fuzzy rule. And $\mu_C^i(\Delta k_{pi}) = \mu_A^i(e_i) \wedge \mu_B^i(\dot{e}_i)$, $\mu_D^i(\Delta k_{di}) = \mu_A^i(e_i) \wedge \mu_B^i(\dot{e}_i)$. Symbol of " \wedge " denote minimum operator. h_C^i and h_D^i stand for the center value of linguistic terms of output variables obtained in i th fuzzy rule. For example, for the first fuzzy rule, as show in TABLE II, when e_i is NB and \dot{e}_i is NB, then Δk_{pi} is PB and Δk_{di} is PB. In this fuzzy rule, $h_C^i = 0.3$ and $h_D^i = 0.3$.

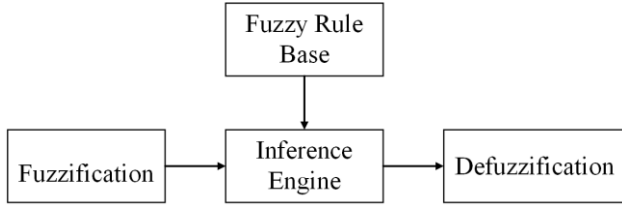


Fig. 8. The composition of the fuzzy controller

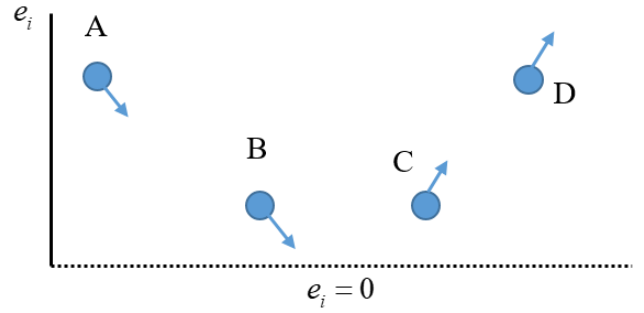
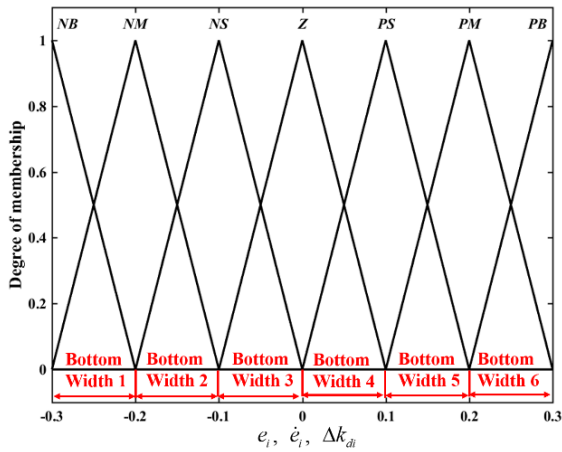
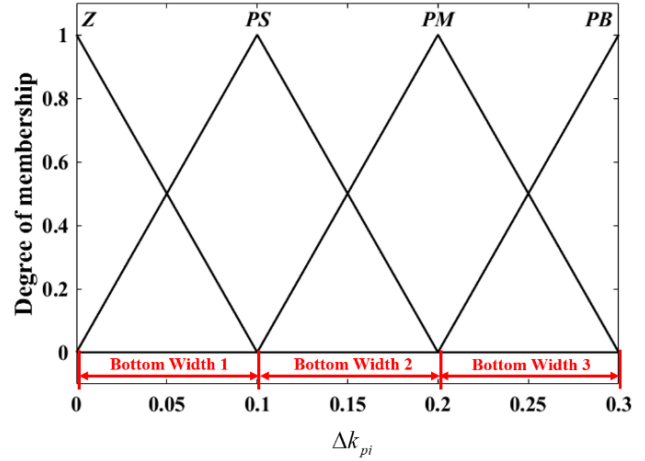


Fig. 9. Four situations of tracking error



(a)



(b)

Fig. 10. Degree of membership of fuzzy controller: (a) e_i , \dot{e}_i and Δk_{di} . (b) Δk_{pi} .

TABLE II. FUZZY LOGIC RULES

$(\Delta k_{pi}, \Delta k_{di})$	\dot{e}_i						
	<i>NB</i>	<i>NM</i>	<i>NS</i>	<i>Z</i>	<i>PS</i>	<i>PM</i>	<i>PB</i>
<i>NB</i>	(<i>PB,PB</i>)	(<i>PB,PM</i>)	(<i>PB,PS</i>)	(<i>PB,Z</i>)	(<i>PB,NS</i>)	(<i>PB,NM</i>)	(<i>PB,NB</i>)
<i>NM</i>	(<i>PM,PB</i>)	(<i>PM,PM</i>)	(<i>PM,PS</i>)	(<i>PM,Z</i>)	(<i>PM,NS</i>)	(<i>PM,NM</i>)	(<i>PM,NB</i>)
<i>NS</i>	(<i>PS,PB</i>)	(<i>PS,PM</i>)	(<i>PS,PS</i>)	(<i>PS,Z</i>)	(<i>PS,NS</i>)	(<i>PS,NM</i>)	(<i>PS,NB</i>)
<i>Z</i>	(<i>Z,PB</i>)	(<i>Z,PM</i>)	(<i>Z,PS</i>)	(<i>Z,Z</i>)	(<i>Z,NS</i>)	(<i>Z,NM</i>)	(<i>Z,NB</i>)
<i>PS</i>	(<i>PS,NB</i>)	(<i>PS,NM</i>)	(<i>PS,NS</i>)	(<i>PS,Z</i>)	(<i>PS,PS</i>)	(<i>PS,PM</i>)	(<i>PS,PB</i>)
<i>PM</i>	(<i>PM,NB</i>)	(<i>PM,NM</i>)	(<i>PM,NS</i>)	(<i>PM,Z</i>)	(<i>PM,PS</i>)	(<i>PM,PM</i>)	(<i>PM,PB</i>)
<i>PB</i>	(<i>PB,NB</i>)	(<i>PB,NM</i>)	(<i>PB,NS</i>)	(<i>PB,Z</i>)	(<i>PB,PS</i>)	(<i>PB,PM</i>)	(<i>PB,PB</i>)

C. Bat algorithm based optimal parameters of fuzzy controller

Bat algorithm (BA), was presented by Yang [40], and was usually used for parameters optimization thanks to that it combines the advantages of existing algorithms.

For BA based optimization of parameters, the update rules of characteristic of virtual bats can be denoted as

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (41)$$

$$\mathbf{v}_i^n = \mathbf{v}_i^{n-1} + (\mathbf{x}_i^n - \mathbf{x}_*)f_i \quad (42)$$

$$\mathbf{x}_i^n = \mathbf{x}_i^{n-1} - \mathbf{v}_i^n \quad (43)$$

Where \mathbf{x}_i^n and \mathbf{v}_i^n are positions and velocities of i th virtual bat at step n in d -dimension space. And \mathbf{x}_* is the current global best solution. Then, f_{\max} and f_{\min} represent

maximum and minimum of the frequency, respectively, and $\beta \in [0,1]$ is a random vector.

For local search, a new solution for each bat can updated by

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \varepsilon A^n \quad (44)$$

Where $\varepsilon \in [-1,1]$ is a random number, while $A^n = \langle A_i^n \rangle$ is the average loudness of all bats at n step. Then the loudness A_i and pulse emission rate r_i are updated by

$$A_i^{n+1} = \alpha A_i^n \quad (45)$$

$$r_i^{n+1} = r_i^0 [1 - e^{(-\gamma n)}] \quad (46)$$

Where $0 < \alpha < 1$, $r_i^0 \in [0,1]$ and $\gamma > 0$ and they are constant. With iteration step increase to infinity, A_i^n and r_i^n approach to zero and r_i^0 , respectively.

Combine fuzzy logic and bat algorithm, it is easy to select bottom widths L_i of membership functions of fuzzy inputs and outputs as optimization variable \mathbf{x} . According to Fig. 9, dimension d of \mathbf{x} , designed as 21, include two subsets of input variables e and \dot{e} , while two subset of output variables Δk_{pi} and Δk_{di} . the arrangement of \mathbf{x} is

$$\mathbf{x}_i = [\underbrace{L(i,1), \dots, L(i,6)}_{e_i}, \underbrace{L(i,7), \dots, L(i,12)}_{\dot{e}_i}, \underbrace{L(i,13), \dots, L(i,15)}_{\Delta k_{pi}}, \underbrace{L(i,16), \dots, L(i,21)}_{\Delta k_{di}}]^T \quad (47)$$

At last, optimization object of the whole system is designed as

$$\min g(e) = \|\max(e)\|_2 \quad (48)$$

Where, symbol “ $\|y\|_2$ ” denote the 2-Norm of vector y .

According to literature [32][33][40], Pseudo code of BA can be summarized in Fig. 11.

```

Initialize the bat population  $\mathbf{x}_i = [L(i,1), \dots, L(i,36)]^T$  and  $\mathbf{v}_i, (i = 1, 2, \dots, m)$ 
Define pulse frequency  $f_i$  and  $\mathbf{x}_i$ 
Initialize pulse rates  $r_i$  and the loudness  $A_i$ 
While ( $n < \text{Max number of iterations}$ )
    Generate new solutions by adjusting frequency according to (41)
    Run the simulation of control system and capture the error signal  $e$ 
    Evaluate object function  $g(e)$  for each solutions
    Updating velocities and solutions by (42), (43)
    if ( $\text{rand} > r_i$ )
        Select a solution among the best solutions
        Generate a local solution around the selected best solution
    end if
    Generate a new solution  $\mathbf{x}_{new}$  by flaying randomly according to (44)
    Run the simulation of control system and capture the error signal  $e$ 
    Evaluate object function  $g(e)$  for each solutions
    if ( $\text{rand} < A_i$  &  $g(e) < g(e_*)$ )
        Accept the new solutions as the new best solution  $\mathbf{x}_* = \mathbf{x}_{new}$ 
        Increase  $r_i$  and reduce  $A_i$  by (45), (46)
    end if
    Rank the bats and find the current best  $\mathbf{x}_*$ 
end while
Post process results and visualization

```

Fig. 11. Pseudo code of the bat algorithm

V. SIMULATION RESULT

The control design is numerically illustrated with the torque calculating by simplified model, along the test trajectory, as displayed in Fig. 4. The corresponding designed control scheme is depicted in Fig. 12, in which the saturation can limit the torques less than its maximum value of 100 N*m.

In this paper, classical computed torque control (CTC) method is compared. According to [20], the control law of CTC can be summarized as:

$$\begin{cases} \tau = M(\theta)u + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \\ \tau_d = \ddot{\theta}_d + K_{cd}\dot{e} + K_{cp}e \end{cases} \quad (49)$$

Where, K_{cd} and K_{cp} are diagonal matrices differ from K_d and K_p . And choose $K_{cd} = \text{diag}([4 \ 4 \ 4 \ 4]) \times 10^2$, $K_{cp} = \text{diag}([2 \ 2 \ 2 \ 2]) \times 10^4$ in the simulation.

For the simulation, $f(\theta, \dot{\theta}, t)$ is a function to simulate un-modeled uncertainty $f(t)$ in (26), which include friction and external disturbance and so on. Then, function $f(\theta, \dot{\theta}, t)$ in this article can be described by

$$f_i(\theta_i, \dot{\theta}_i, t) = 2 \sin(\theta_i) - 2 \sin(\dot{\theta}_i); i = 1 \dots 4 \quad (50)$$

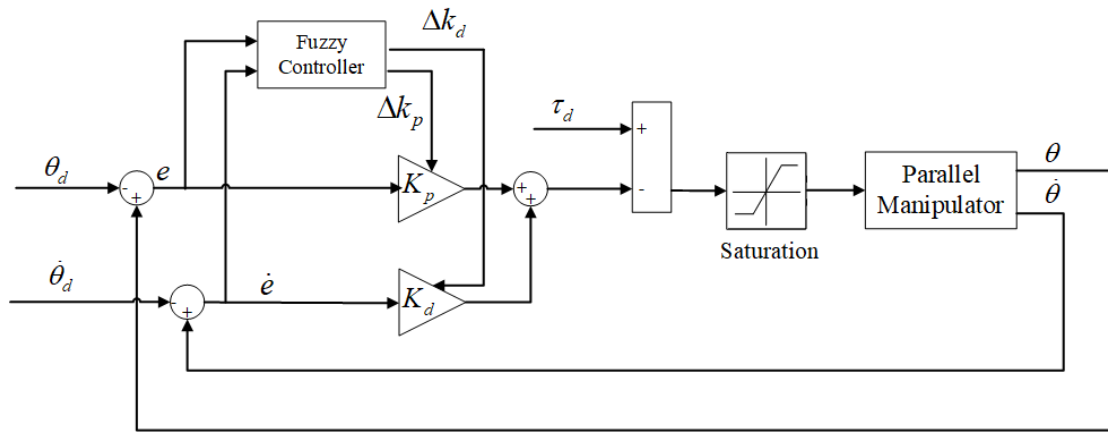


Fig. 12. Scheme of torque-based feedforward fuzzy PD control system

And according to characteristics of normal pick-and-place trajectory, as shown in Fig. 5, and (50), the un-modeled uncertainty can be displayed in Fig. 13.

About the setting of the PD controller, base gains $K_p = \text{diag}([2 \ 2 \ 2 \ 2]) \times 10^4$ and $K_d = \text{diag}([2 \ 2 \ 2 \ 2]) \times 10^2$. Then, the range of adjustable gains set as $\Delta k_{pi} \in [-4000, 4000]$ and $\Delta k_{di} \in [-40, 40]$. In order to obtain a good optimization result, the parameters of bat algorithm can be set referenced to [32][33][40], i.e. use $i = 20$ bats, $\alpha = \gamma = 0.9$, $A_i = 0.25$ and $r_i^0 = 0.5$. The range of frequency set as $[f_{\min}, f_{\max}] = [0, 2]$.

Sampling time set at 10^{-4} s. then, set initial condition of the manipulator model as $\theta^{(0)} = \theta_d^{(0)} - \theta_e$, $\dot{\theta}^{(0)} = 0$ and $\ddot{\theta}^{(0)} = 0$, which $\theta_e = 0.2$ is initial position error.

In the case of un-modeled uncertainty existing, Fig. 14 shows the tracking performance of each joints under bat algorithm-based torque feedforward fuzzy PD (BA-TFFPD) controller, torque feedforward PD (TFPD) controller and computed torqued controller (CTC). And angular error of each joints is shown in Fig. 15. TABLE III is the mean tracking error of each joints during tracking task.

Compared with CTC, as shown in Fig. 14, BA-TFFPD can reduce the tracking error in joint space (e.g. the mean tracking error of TFFPD of joint 1 is $e = -0.41 \times 10^{-3}$ rad, while $e = -0.67 \times 10^{-3}$ rad for CTC), even if un-modeled uncertainty is existing. What's more, it can be seen in Fig.

15 and Fig. 5, the larger torque error between simplified model and real model of manipulator will lead the larger tracking error. Obviously, in the case of that the precise mathematical model cannot be obtained, tracking performance under the effort of BA-TFFPD is better than CTC.

On the other hand, with the help of fuzzy logic, gains of PD controller can be tuned in real-time to adapt to different rang of tracking error and external disturbance. In fact, as show in Fig. 14 and Fig. 15, faster transient response can be obtained by using BA-TFFPD rather than TFFPD and CTC. Such as the convergence time of BA-TFFPD of joint 2 is approximately $t = 0.0056$ s, but it will increase to 0.0089s and 0.0147 for CTC and TFFPD.

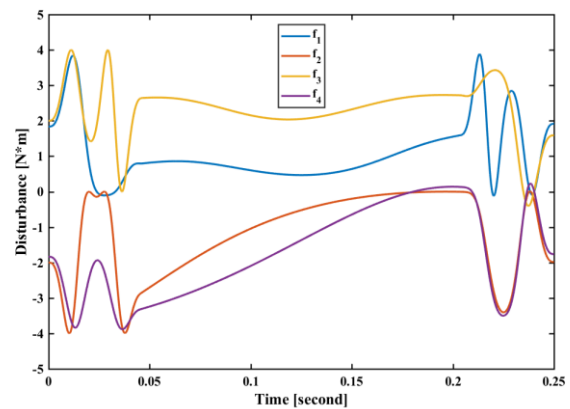


Fig. 13. Un-modeled uncertainty of each joints

TABLE III. THE MEAN TRACKING ERROR OF EACH JOINTS

Controller Joint	BA-TFFPD ($\times 10^{-3}$ rad)	TFPD ($\times 10^{-3}$ rad)	CTC ($\times 10^{-3}$ rad)
Joint 1	-0.41	-0.90	-0.67
Joint 2	0.41	0.94	0.63
Joint 3	-0.44	-0.98	-0.81
Joint 4	0.48	0.98	0.70

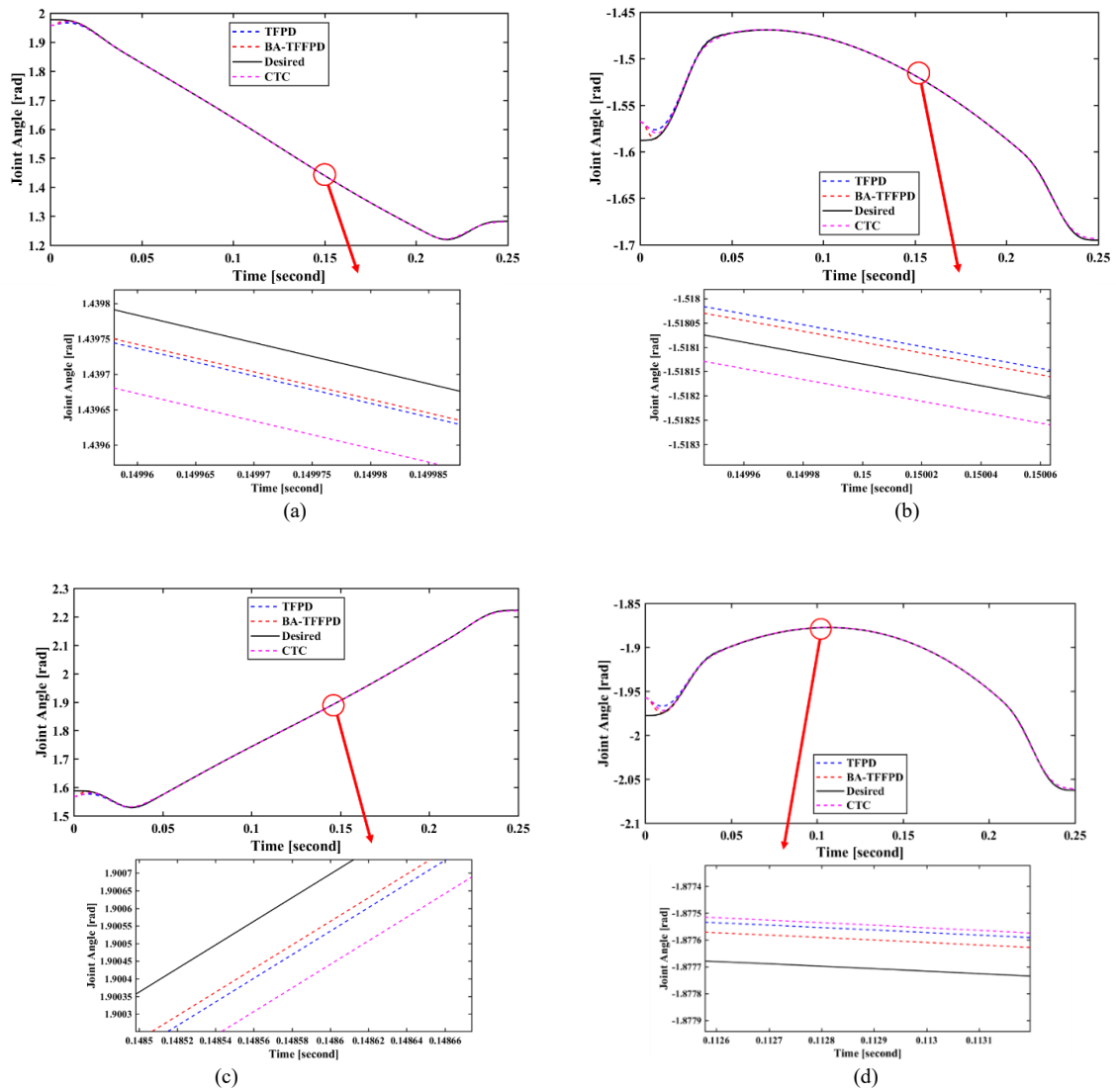


Fig. 14. Tracking performance of TFPD and BA-TFFPD: (a) Joint 1; (b) Joint 2; (c) Joint 3; (d) Joint 4.

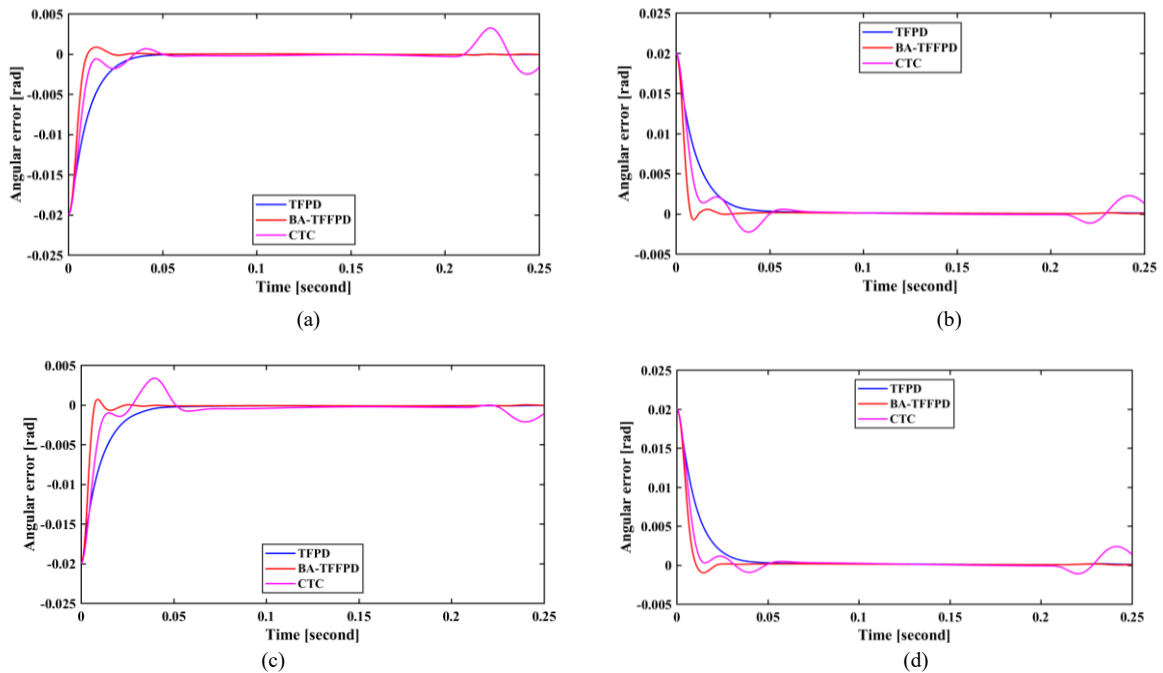


Fig. 15. Tracking error of each joints: (a) Joint 1; (b) Joint 2; (c) Joint 3; (d) Joint 4.

VI. CONCLUSION

In this paper, a fast-parallel manipulator for pick-and-place operation, whose rotational capability was enhanced by helical joint, has been presented. And the simplified model was built by ignoring the inertia of the rods. Because traditional PD controller without a priori knowledge of manipulator cannot lead a well tracking performance, and in fact, it is difficult to build a precise dynamic model to employ the computed torque method controller (CTC). For those reason, torque feedforward fuzzy PD controller, whose bottom widths of membership function of fuzzy controller were optimized by bat algorithm, was investigated. The tracking performance of BA-TFFPD was evaluated by simulating in MATLAB/Simulink.

The main contribution of this study is that bat algorithm-based torque feedforward fuzzy PD (BA-TFFPD) can be easily employed in the manipulator, whose dynamic parameters are difficult to identify precisely. By calculating the actuation torque offline, BA-TFFPD has the lower computational cost than CTC. Then BA-TFFPD introduce the fuzzy logic to tune the gains of PD controller, which obtain a faster transient response than torque feedforward PD controller (TFPD) without adjustable gains. Finally, after the optimization by bat algorithm, BA-TFFPD get a better tracking performance than CTC and TFPD.

REFERENCES

- [1] S. Suri, A. Jain, N. Verma and N. Prasertpoj, "SCARA Industrial Automation Robot," 2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC), Greater Noida, India, pp. 173-177, 2018.
- [2] G. Wu, Z. Lin, W. Zhao, S. Zhang, H. Shen, S. Caro "A four-limb parallel Schönflies motion generator with full-circle end-effector rotation," *Mechanism and Machine Theory*, vol.146, pp.425-435, 2020.
- [3] F. Pierrot and O. Company, "H4: a new family of 4-DOF parallel robots," 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399), Atlanta, GA, USA, pp. 508-513, 1999.
- [4] S. Krut, M. Benoit, H. Ota and F. Pierrot, "I4: A new parallel mechanism for Scara motions," 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), Taipei, Taiwan, vol. 2, pp. 1875-1880, 2003.
- [5] V. Nabat, M. de la O Rodriguez, O. Company, S. Krut and F. Pierrot, "Par4: very high speed parallel robot for pick-and-place," 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, pp. 553-558, 2005.
- [6] O. Altuzarra, B. Sandru, C. Pinto, V. Petuya, "A symmetric parallel Schönflies motion manipulator for pick-and-place operations," *Robotica*, vol.29, pp.853-862, 2011.
- [7] G. Wu, "Kinematic analysis and optimal design of a wall-mounted four-limb parallel Schönflies-motion robot for pick-and-place operations," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 85, pp. 663-677, 2017.
- [8] G. Wu, S. Bai, P. Hjønet, "Architecture Optimization of a Parallel Schönflies motion Robot for Pick-and-Place Applications in a Predefined Workspace," *Mechanism and Machine Theory*, vol. 106, pp. 148-165, 2016.
- [9] G. Wu, S. Bai, S. Caro, "A transmission quality index for a class of four-limb parallel Schönflies motion generators," *Journal of Mechanisms and Robotics*, vol.10, pp. 051014-051014, 2018.
- [10] S. M. Kim, W. Kim and B. Yi, "Kinematic analysis and optimal design of a 3T1R type parallel mechanism," 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, pp. 2199-2204, 2009.
- [11] F. Xie, X. Liu, "Design and development of a high-speed and high-rotation robot with four identical arms and a single platform," *Journal of Mechanisms and Robotics*, vol.7, pp.41015-41015, 2015.
- [12] D. Corbel, M. Gouttefarde, O. Company, F. Pierrot, "Actuation redundancy as a way to improve the acceleration capabilities of 3T and 3T1R pick-and-place parallel manipulators," *Journal of Mechanisms and Robotics*, vol. 2, pp. 041002-041002, 2010.
- [13] F. Pierrot, V. Nabat, O. Company, S. Krut and P. Poignet, "Optimal Design of a 4-DOF Parallel Manipulator: From Academia to Industry," in *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 213-224, April 2009.
- [14] A. T. Azar, F. E. Serrano and A. Koubaa, "Adaptive Fuzzy Type-2 Fractional Order Proportional Integral Derivative Sliding Mode Controller for Trajectory Tracking of Robotic Manipulators," 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, pp. 183-187, 2020.
- [15] J. Wan, B. He, D. Wang, T. Yan and Y. Shen, "Fractional-Order PID Motion Control for AUV Using Cloud-Model-Based Quantum Genetic Algorithm," in *IEEE Access*, vol. 7, pp. 124828-124843, 2019.
- [16] T. M. Ahmed, A. N. A. E. Gaber, R. Hamdy and A. S. Abdel-Khalik, "Position Control of Arm Manipulator Within Fractional Order PID Utilizing Particle Swarm Optimization Algorithm," 2019 IEEE Conference on Power Electronics and Renewable Energy (CPERE), Aswan, Egypt, pp. 135-139, 2019.
- [17] R. Sharma, D. Joshi and P. Gaur, "A study on different configurations of fractional order fuzzy logic controller scheme for robotic manipulator using NSGA-II," 2018 8th IEEE India International Conference on Power Electronics (IICPE), Jaipur, India, pp. 1-6, 2018.
- [18] J. Zhang et al., "Design and Implementation of Novel Fractional-Order Controllers for Stabilized Platforms," in *IEEE Access*, vol. 8, pp. 93133-93144, 2020.
- [19] W. A. Shutman and T. Y. Abdalla, "Artificial immune system based optimal fractional order PID control scheme for path tracking of robot manipulator," 2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA), Wasit Kut, Iraq, pp. 19-24, 2018.
- [20] J. Viola and L. Angel, "Delta Parallel Robotic Manipulator Tracking Control using Fractional Order Controllers," in *IEEE Latin America Transactions*, vol. 17, no. 03, pp. 393-400, March 2019.
- [21] M. Elsis, K. Mahmoud, M. Lehtonen and M. M. F. Darwish, "An Improved Neural Network Algorithm to Efficiently Track Various Trajectories of Robot Manipulator Arms," in *IEEE Access*, vol. 9, pp. 11911-11920, 2021.
- [22] E. O. Freire, F. G. Rossomando and C. M. Soria, "Self-tuning of a Neuro-Adaptive PID Controller for a SCARA Robot Based on Neural Network," in *IEEE Latin America Transactions*, vol. 16, no. 5, pp. 1364-1374, May 2018.
- [23] F. Liu, H. Wang, Q. Shi, H. Wang, M. Zhang and H. Zhao, "Comparison of an ANFIS and Fuzzy PID Control Model for Performance in a Two-Axis Inertial Stabilized Platform," in *IEEE Access*, vol. 5, pp. 12951-12962, 2017.
- [24] V. Nguyen, C. Lin, S. Su and Q. Tran, "Adaptive PID tracking control based radial basic function networks for a 2-DOF parallel manipulator," 2017 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, pp. 309-312, 2017.
- [25] P. C. Sau, "Comparative study of Fractional Order Controllers for Three-link Robotic Manipulator System," 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, pp. 528-533, 2020.
- [26] Q. T. Dao and S. i. Yamamoto, "Modified Computed Torque Control of a Robotic Orthosis for Gait Rehabilitation," 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, pp. 1719-1722, 2018.
- [27] T. Lima Costa, F. A. Lara-Molina, A. A. Cavalini Junior and E. Taketa, "Robust H_∞ Computed torque Control for Manipulators," in *IEEE Latin America Transactions*, vol. 16, no. 2, pp. 398-407, Feb. 2018.
- [28] L. Santos and R. Cortesão, "Computed-Torque Control for Robotic-Assisted Tele-Echography Based on Perceived Stiffness Estimation," in *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1337-1354, July 2018.
- [29] J. Viola and L. Angel, "Tracking Control for Robotic Manipulators using Fractional Order Controllers with Computed Torque Control," in

- IEEE Latin America Transactions, vol. 16, no. 7, pp. 1884-1891, July 2018.
- [30] M. Alhaddad, B. Shaukifeh and A. Joukhadar, "Adaptive LQ-Based Computed-Torque Controller for Robotic Manipulator," 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus), Saint Petersburg and Moscow, Russia, pp. 2169-2173, 2019.
- [31] X. Gao, Z. Ren, L. Zhai, Q. Jia and H. Liu, "Two-Stage Switching Hybrid Control Method Based on Improved PSO for Planar Three-Link Under-Actuated Manipulator," in IEEE Access, vol. 7, pp. 76263-76273, 2019.
- [32] S. Chen, B. Yang, T. Pu, C. Chang and R. Lin, "Active Current Sharing of a Parallel DC-DC Converters System Using Bat Algorithm Optimized Two-DOF PID Control," in IEEE Access, vol. 7, pp. 84757-84769, 2019.
- [33] H. Huang, "Fusion of Modified Bat Algorithm Soft Computing and Dynamic Model Hard Computing to Online Self-Adaptive Fuzzy Control of Autonomous Mobile Robots," in IEEE Transactions on Industrial Informatics, vol. 12, no. 3, pp. 972-979, June 2016.
- [34] S. Han, "Modified Grey-Wolf Algorithm Optimized Fractional-Order Sliding Mode Control for Unknown Manipulators With a Fractional-Order Disturbance Observer," in IEEE Access, vol. 8, pp. 18337-18349, 2020.
- [35] X. -K. Dang, V. -D. Do and X. -P. Nguyen, "Robust Adaptive Fuzzy Control Using Genetic Algorithm for Dynamic Positioning System," in IEEE Access, vol. 8, pp. 222077-222092, 2020.
- [36] M. Elsisy, K. Mahmoud, M. Lehtonen and M. M. F. Darwish, "An Improved Neural Network Algorithm to Efficiently Track Various Trajectories of Robot Manipulator Arms," in IEEE Access, vol. 9, pp. 11911-11920, 2021.
- [37] J. Chroua, W. Chakchouk, A. Zaafouri and M. Jemli, "Modeling and Control of an Irrigation Station Process Using Heterogeneous Cuckoo Search Algorithm and Fuzzy Logic Controller," in IEEE Transactions on Industry Applications, vol. 55, no. 1, pp. 976-990, Jan.-Feb. 2019.
- [38] W. Liu, Y. Sun, X. Yuan, M. Chen, "A new approach to the pre-compensation of contour errors for three-axis machine tools using an adaptive cross-coupled controller," International Journal of Advanced Manufacturing Technology, vol. 90, pp. 3711-3725, 2017.
- [39] F. Liu, H. Wang, Q. Shi, H. Wang, M. Zhang and H. Zhao, "Comparison of an ANFIS and Fuzzy PID Control Model for Performance in a Two-Axis Inertial Stabilized Platform," in IEEE Access, vol. 5, pp. 12951-12962, 2017.
- [40] X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), vol. 284, pp. 65-74, 2010.