




Article

# Active Disturbance Rejection Control of Multi-Joint Industrial Robots Based on Dynamic Feedforward

Xin Cheng <sup>1,2</sup> , Xiao Tu <sup>3</sup> , Yunfei Zhou <sup>3</sup>  and Rougang Zhou <sup>4,\*</sup>

<sup>1</sup> School of Mechanical & Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China; chengx@whut.edu.cn

<sup>2</sup> Hubei Maglev Engineering Technology Research Center, Wuhan University of Technology, Wuhan 430070, China

<sup>3</sup> School of Mechanical Science & Engineering, Huazhong University of Science & Technology, Wuhan 430070, China; tuxiao90321@hust.edu.cn (X.T.); yfzhou@hust.edu.cn (Y.Z.)

<sup>4</sup> School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310002, China

\* Correspondence: zhourg@hdu.edu.cn

Received: 3 May 2019; Accepted: 23 May 2019; Published: 27 May 2019



**Abstract:** In this paper, the dynamics-based high-performance robot motion control technology has been mainly studied, and the overall structure is controlled via dynamics forward, given the nonlinearity, strong coupling and time-variability of robots. Considering the unavailability of precise robot model parameters and the uncertain disturbance in real operation, we put forward an active disturbance rejection control (ADRC) strategy based on dynamic feedforward, aiming to improve the control robustness and combining the simple structure, strong anti-disturbance ability, and no restriction from the control model of ADRC. Given the multi-joint coupling of robots, controlled decoupling is conducted by using dynamic characteristics. The ADRC cascade control structure and algorithm based on dynamic feedforward have been studied and the closed-loop stability of the system is investigated by analyzing the system dynamic linearization compensation and the anti-disturbance ability of the extended state observer. Experiments have shown the new strategy is more robust over uncertain disturbance than the conventional proportional-integral-derivative control strategy.

**Keywords:** industrial robots; active disturbance rejection control; dynamic feedforward; cascade control structure; closed-loop stability

## 1. Introduction

Industrial robot systems are featured with nonlinearity, strong coupling and multivariate time-variability [1]. High-quality motion control refers to high-speed and high-precision motion control, that is the robot in the high-speed motion process, can have fast dynamics response, high tracking accuracy, smooth start and stop stage. Although the robot ontology parameters are fixed, the changing load and uncertain disturbances can affect the robustness of the control, and even significant mechanical vibration. Moreover, the control systems during high-speed motion are largely challenged by the severe multi-joint torque coupling, large inertia change and significant nonlinear effect. The attitude variation stable refers to the robot maintaining a smooth, non-jitter throughout the movement. The stability index needs to be determined by the actual requirements, such as in the air blade grinding process the trajectory error is required to be less than 0.05 mm. To keep the position and attitude variation stable and achieve high-quality motion control, researchers have to first solve the nonlinear time-variability due to the characteristics of robots. Namely, the control system should provide real-time control characteristics that match with the dynamics of robots and the multi-source disturbance characteristics, thereby achieving multi-joint nonlinear decoupling control.

The conventional proportional-integral-derivative (PID) controllers are structurally simple and achievable, but demand high power consumption for large-inertia changes and can ensure only local stability. The main reason is that this control strategy is usually applicable to linear stationary systems, and its conservative control parameters limit the gain bandwidth and thereby affect dynamic characteristics, leading to large delayed errors. Moreover, the large inertia variation may cause vibration in the system and destroy the stability. Thus, this strategy does not meet the requirement of robot systems for nonlinear time-variability. Later, nonlinear PID [2], blur PID [3] and neural network PID [4,5] controllers have been put forward and relatively outperform traditional PID controllers. These new strategies match with the nonlinear time-variability of robot systems by similarly optimizing the PID structure and improving its performance, but they are little dependent on the information of dynamics models and thus fail to meet the requirements of multi-joint high-quality motion control.

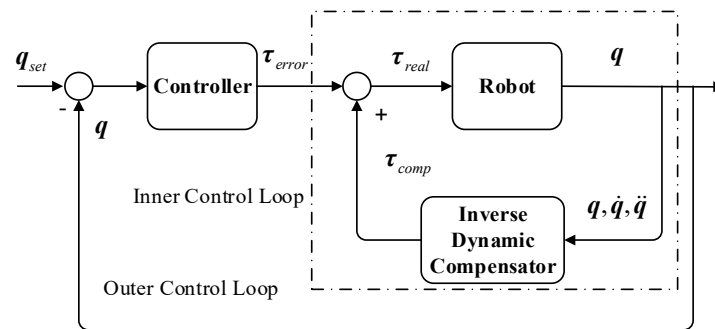
To promote further the motion performances of robots and solve the nonlinear time-variability and multi-source disturbance of robots, researchers have proposed finer nonlinear control algorithms, such as adaptive control, robust control and active disturbance rejection control (ADRC).

The adaptive control algorithms, which are structurally diverse, essentially extract the information of system states from closed-loop error feedback models, and regulate the control parameters via adaptive algorithms, so as to compensate the time-variability of the systems [6]. However, adaptive control has a strict requirement for real-time performance and is too complex to be realized, and its parameter mutations often destroy the overall stability; the parameter convergence is dependent on sufficient continual excitation, which is unachievable in real life. Thus, the system robustness of adaptive control should also be considered. For this reason, robust adaptive controlling algorithms have been put forward, which adopt the modified adaptive rate to make the system robust to the nonparametric uncertain disturbance in the models [7,8]. Typically, Hu et al. theoretically verified the two merits of comprehensive adaptive control and robust control of adaptive robust control [9,10]. However, adaptive robust control algorithms are yet based on the system feedback status, and control movement through the estimation of system characteristics, which raise the requirements for the amount and timeliness of state information. In fact, no enough state information can be acquired in advance at the initial stage, which leads to severe control lag and heavy following errors. Moreover, its structural complexity and diversity complicate practical applications.

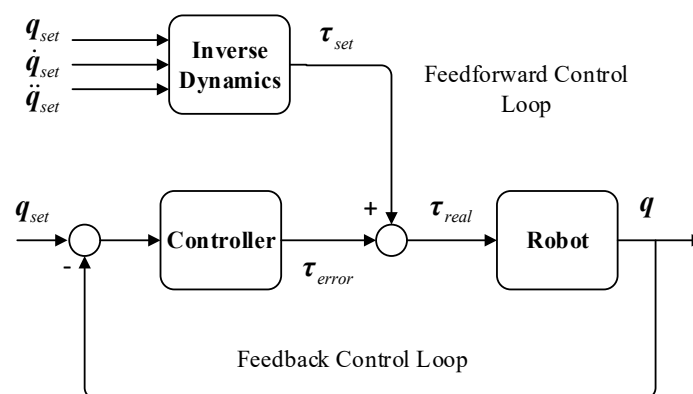
ADRC [11], put forward in 1989, has been theoretically verified and continually improved [12–15]. ADRC mainly consists of an extended state observer (ESO), a tracking differentiator (TD) and error feedback control laws, and focuses on solving the uncertain external disturbance. ADRC controllers are structurally simple, efficient and excellent at anti-interference, and have been applied into industrial robot control [16,17]. The anti-disturbance core of ADRC controllers is dependent on ESO and largely affected by the unknown disturbance amplitude. If the whole dynamic characteristics of robots are estimated and compensated by regarding them as systematic disturbance, the fast convergence and systematic stability of ESO cannot be ensured, which limit the use of ADRC. The advanced control algorithms above have modestly improved the control performance of robots, but are unable to really describe the dynamic characteristics of robots and thus are limited in use. Nevertheless, along with the development of electronic technique and computer science recently, the controlling algorithms based on dynamic models have gained growing attention and are considered as the most efficient way to enhance the dynamic characteristics and tracking precision of robots. Their aims are to compensate according to the dynamic characteristics of robots and thereby simplify the complex robot system into a controllable system, such as decoupling it into a linear time-invariant system [18].

Generally, there are two schemes of model-based control: (1) Robot dynamic feedback compensation control (Figure 1a) and robot dynamic feedforward control (Figure 1b). For the first structure, when the dynamic model is precise enough, it can well solve the nonlinear time-variability of robots through feedback compensation, and reduces the power consumption in regulation, due to external feedback loop errors, improving the dynamic response and following the performance of

controllers. However, it faces three difficulties. (1) No precise model parameters can be acquired, due to the load variation and sampling/computation errors. (2) This control structure raises the requirement for computation timeliness and has to overcome the time restraint of the torque loop control algorithm. (3) The real-time information of joint acceleration is needed, but in fact, the high-order differential signals are low-qualified, due to the magnifying effect of differential noise. Thus, model-based dynamic compensation control strategies are difficult to implement in practice.



(a)



(b)

**Figure 1.** Model-based control structure. (a) dynamic feedback compensation; (b) dynamic feedforward.

Different from the first structure, the dynamic feedforward model inputs the expected joint movement order and offers the far-higher-quality signals than the sampled values. Moreover, despite the complexity of dynamic computation, the problem of dynamic feedforward timeliness can be solved through pre-storing or setting background programs, which are favorable for the implementation of control algorithms. Its ideas are to rapidly compensate the system characteristics through a precise dynamic model and thus to precisely control the movement. In fact, however, due to the diversity of model parameters and the severe effects of external loading and interference, the model parameters are hard to acquire precisely, which reduce the feedforward control performance. For this reason, an observer is usually used to pre-estimate and compensate for the uncertain disturbance, which would improve the robustness of feedforward controllers. The model uncertain disturbance observer is usually based on the construction of a system model, but a complicated robot system involves the computation of robot forward dynamics, which interfere with the timeliness of motion control. For this reason and regarding the simple structure, strong robustness and no constraint from the control model of ADRC, we put forward an ADRC strategy based on dynamic feedforward.

As for the nonlinearity and strong coupling time-variability of robots, we use dynamic feedforward compensation and uncertainty disturbance compensation into multi-joint control decoupling, or namely

to regard the joints as independent control units. To reduce the order of disturbance observation and improve the observation precision, we study the ADRC cascade control structure and algorithm based on dynamic feedforward. We analyze the system dynamic linearization compensation and the anti-disturbance ability of ESO, and the stability of closed-loop systems. On this basis, the robot ADRC based on dynamic feedforward has been realized.

## 2. Robot System Decoupling Control

### 2.1. Analysis of System Disturbance Compensation

The basic ways of multi-joint decoupling control are to use dynamic feedforward compensation and uncertainty disturbance compensation. To solve disturbance via ADRC, we had to clarify the input, output, relative order number and disturbance compensation of the controlled system. According to the dynamic feedforward control structure, the current instruction is the input and the motor rolling angle is the output of the system. For the motor, the robot dynamic characteristics can be regarded as a result of the motor output torque  $\tau_m$  overcoming friction torque  $\tau_{mf}$  and inertia torque (in Figure 2):

$$\tau_m = J_m \ddot{q}_m + \tau_{ml} + \tau_{mf}, \tag{1}$$

where

$$J_m = \text{diag}(J_{m1}, J_{m2}, \dots, J_{mn}), \tag{2}$$

is the rotary inertia of the motor shaft;  $\ddot{q}_m$  is the angular acceleration of the motor shaft;  $\tau_{ml}$  is the loaded torque vector of the motor to overcome the joint dynamic characteristics, including joint inertia torque, friction torque, gravity torque and joint loading. To build the relation between motor input current  $I_q$  and output position  $q_m$ , firstly, we had the machinery transmission relation  $\eta_i$  to express the gear kinematic relations at both ends:

$$\begin{cases} \Lambda = \text{diag}(1/\eta_1, 1/\eta_2, \dots, 1/\eta_n) \\ q = \Lambda q_m, \dot{q} = \Lambda \dot{q}_m, \ddot{q} = \Lambda \ddot{q}_m \\ \tau_{ml} = \Lambda \tau \end{cases}, \tag{3}$$

where  $q, \dot{q}, \ddot{q}$  express the kinestate of robot joints,  $\tau$  is the output torque vector at the joint end:

$$\tau = M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) + \tau_{load}, \tag{4}$$

where  $M(q)$  is the inertia matrix of Joints;  $V(q, \dot{q})$  is the centrifugal force and Coriolis force vectors;  $G(q)$ ,  $F(q, \dot{q})$  and  $\tau_{load}$  are the gravitational torque, friction torque, joint loading torque vectors, respectively. Combining with the electromagnetic torque equation:

$$\tau_m = K_m I_q, \tag{5}$$

where  $K_m = \text{diag}(k_{m1}, k_{m2}, \dots, k_{mn})$  is the torque constant matrix,  $I_q = [i_{q1}, i_{q2}, \dots, i_{qn}]^T$  is the current order vector, we can determine the motor shaft motion equation:

$$\ddot{q}_m = J_m^{-1}(K_m I_q - \Lambda \tau - \tau_{mf}) \tag{6}$$

Let  $f(q, \dot{q}, t) = -J_m^{-1}(\Lambda \tau + \tau_{mf})$ , we have

$$\ddot{q}_m = J_m^{-1} K_m I_q + f(q, \dot{q}, t). \tag{7}$$

During ADRC construction, usually  $f(q, \dot{q}, t)$  is regarded as the comprehensive acceleration disturbance of the system, including the dynamic characteristic disturbance of the robot and other uncertain disturbances. In the high-speed large-inertia changing the movement of robots, the real

change  $f(q, \dot{q}, t)$  is very large, and the large-amplitude disturbance variation may induce the slow convergence or even instability of ESO, thereby leading to large following errors or even system vibration. Thus, it needs to solve the problems to decrease the observed values of acceleration and increase the robustness of ESO.

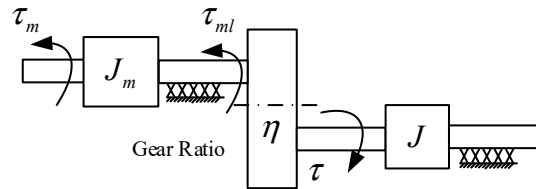


Figure 2. Transmission structure from robot motor end to joint end.

The analysis showed there were two ways: (1) Designing stable ESO ensured it could effectively predict all the dynamic disturbance; (2) fully utilizing the known theoretical dynamic models and conducting uncertainty compensation would ensure ESO only observed the relatively small error part except the model. The real output torque  $\tau$  consists of the model-estimated torque  $\tau_{pre}$  and torque errors  $\Delta\tau$ .

$$\tau = \tau_{pre} + \Delta\tau. \tag{8}$$

Based on this idea, starting from the motor end to the joint end, the real output torque  $\tau$  can be regarded as the disturbance of the output shaft from the motor, where  $\tau_{pre}$  can be compensated by the identification model, and the uncertainty except the model  $\Delta\tau$  can be observed by the ESO. We have:

$$\begin{cases} B_0 = J_m^{-1}K_m \\ F_1 = -J_m^{-1}\Delta\tau_{pre} \\ F_2 = -J_m^{-1}(\Delta\tau + \tau_{mf}) \\ F = F_1 + F_2 \\ u = I_q \end{cases}, \tag{9}$$

where  $B_0$  is the system gain, and  $F_1$  is the theoretical disturbance related to a known dynamic model and can be used into feedforward compensation;  $F_2$  is the uncertain acceleration disturbance except for the model and can be observed via the ESO. Noticeably,  $F_2$  includes the joint torque errors, which may contain minor time-variable acceleration component  $\ddot{q}_m$  that cannot be observed by ESO, and, theoretically, the corresponding control current should be compensated.  $F$  can be considered as the total disturbance of the motor system. Admittedly, the robot system is a second order multiple-input multiple-output (MIMO) system,  $u$  is the real input of the controlled object or namely the current control quantity of the motor driven system.

Let  $\Delta\tau_{pre}$  be the theoretical output torque at the motor end, and  $I_{pre}$  be the corresponding drive current, then:

$$\Delta\tau_{pre} = K_m I_{pre}, F_1 = -B_0 I_{pre}, \tag{10}$$

Let

$$u_{ff} = I_{pre}, U = B_0 u. \tag{11}$$

Thereby the motor output state equation can be expressed as:

$$\begin{cases} \dot{X}_1 = X_2 \\ \dot{X}_2 = B_0 u + F_1 + F_2 = U + F \\ Y = X_1 \end{cases}, \tag{12}$$

where  $\mathbf{U}$  can be seen as the virtual control quantity of the system;  $\mathbf{Y}$  is the output from the system. The real inertia demand can be met by modifying the system gain  $\mathbf{B}_0$ :

$$\mathbf{B}_0 = \mathbf{J}_m^{-1} \mathbf{K}_m \boldsymbol{\zeta}, \tag{13}$$

where  $\boldsymbol{\zeta}$  is the diagonal matrix of the correction coefficient,  $\zeta_i \geq 1$ . Many simulation studies show during ADRC control, the estimation demand of  $\mathbf{B}_0$  is not high, and the relative error is within 30%, and when it is higher, it does not affect the closed-loop control quality. Thus,  $\boldsymbol{\zeta}$  is usually considered as constant in practical applications [19].

### 2.2. Robot System Decoupling

The decoupling of multi-joints could ease the difficulty of controlling. In the ADRC system, the model part except the system-controlled quantity  $\mathbf{F}$  is called the ‘dynamic coupling part’ and  $\mathbf{U}$  is called the ‘static coupling part’. In the MIMO system, the input and output relation of channel  $i$  is:

$$\begin{cases} \dot{x}_{i1} = x_{i2} \\ \dot{x}_{i2} = U_i + F_i \\ y_i = x_{i1} \end{cases} . \tag{14}$$

It is assumed  $U_i$  is the ‘virtual control quantity’ of channel  $i$ , then it and the output  $y_i$  form a single-input - single-output relation, or namely the controlled output  $y_i$  and the controlled quantity  $U_i$  of channel  $i$  can be completely decoupled:  $F_i$  is considered as the total disturbance of channel  $i$ , including the acceleration disturbance of robot dynamics, and the uncertain disturbance except for the model. Thereby,  $n$  ADRC controllers implemented in parallel between the control quantity  $\mathbf{U}$  and the output vector  $\mathbf{Y}$  can control the multivariate system structure. The decoupling process is shown below (Figure 3):

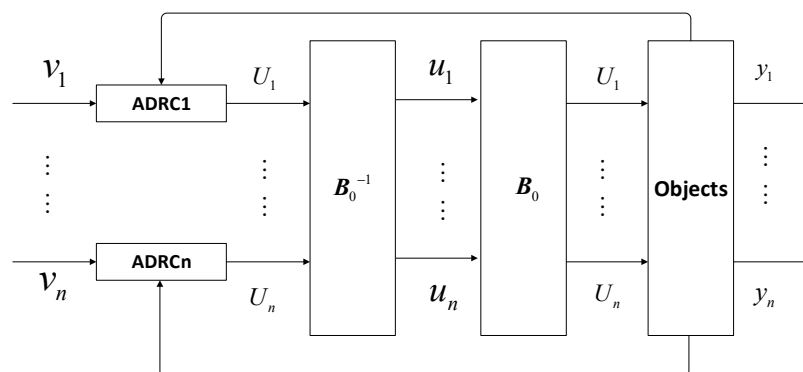


Figure 3. Robot ADRC control decoupling.

The conversion relation between the real control quantity  $\mathbf{u}$  and the virtual control quantity  $\mathbf{U}$  is shown as:

$$\mathbf{u} = \mathbf{B}_0^{-1} \mathbf{U}. \tag{15}$$

Admittedly,  $\mathbf{B}_0$  is reversible, time-variable and bounded, and in ADRC control, its required estimation precision is not high. It should be noted the dynamic coupling part  $\mathbf{F}$  during decoupling can be considered as the total disturbance to be estimated and compensated, and its estimation precision decides the control performance of ADRC. Due to the very large inertia change during robot movement, the corresponding change of acceleration disturbance is also large, so it will intensify the burden on the ESO if it is all observed by ESO. According to the torque feedforward compensated total disturbance, only the unknown small amount is undertaken by ESO, which thereby largely reduces the burden

on ESO and enhanced the system robustness. Then based on the principles of ADRC, we study the structure and algorithm of independent motion controllers.

### 3. Motion Controller Design and Stability Analysis

#### 3.1. Design of Controller

The decoupled robot system is transformed into five independent single-input single-output (SISO) systems, degrees of freedom of which formed independent control channels. A computed torque feedforward control strategy has been adopted to improve the control performance and stability of ADRC. It should be noted the controlled object here is no longer the robot joint position, but is the institution ontology and servo drivers under the torque mode (current mode).

The robot system is second order and usually has two choices in the ADRC structure: A second order system structure (Figure 4) and a cascade system structure (Figure 5). In the second order system, the control variable  $u$  (current) directly drives the output of displacement  $x_1$ , without any transitional loop, and the corresponding uniaxial servo feedforward control structure is shown in Figure 4. This structure consists of two loops—a position loop and a current loop. The position loop directly offers a current command to the current loop implemented in a motor driver. The current feedforward path, as the main path, can offer the motor servo system with real-time theoretical current corresponding to the movement command, while the real-time current errors have been observed by ESO and regulated and compensated by the PID.

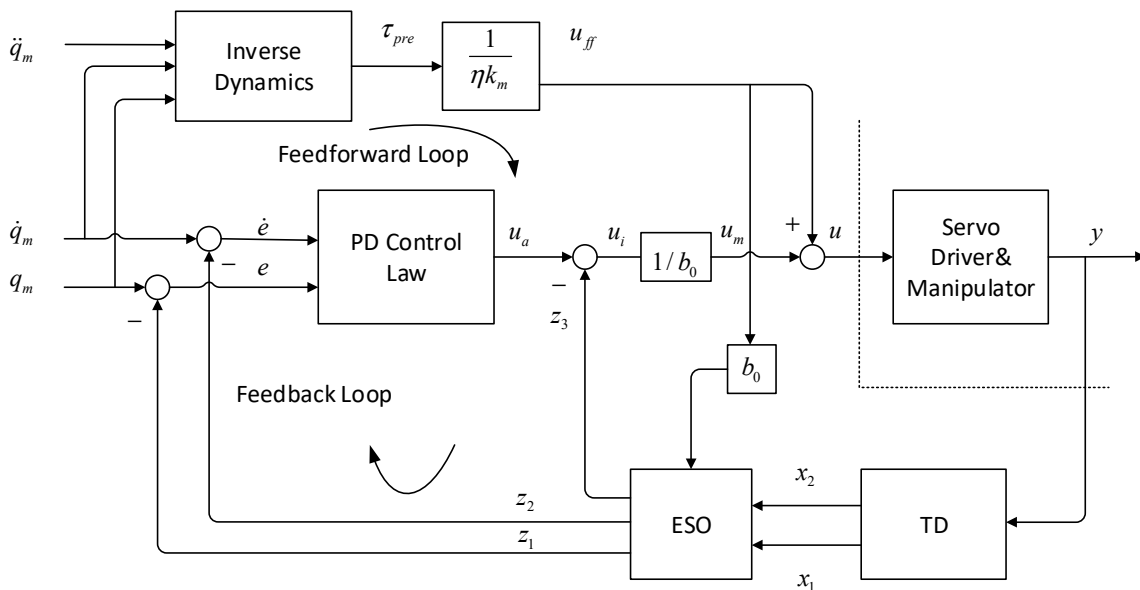


Figure 4. Structure of ADRC feedforward control 2-order system.

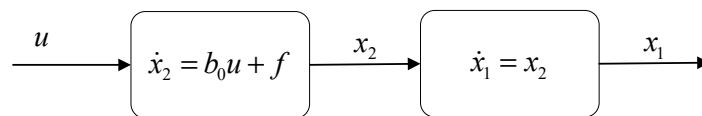


Figure 5. Structure of the cascade system.

Since the current loop bandwidth is large and the current order from the position loop should be rapidly updated, the position order should be more refine and kept continuous, otherwise, the control variable  $u$  would be discontinuous, thereby inducing systematic vibration.

In the structure of the ADRC cascade system, the control variable  $u$  (current) first drives the intermediate variable  $x_2$  (velocity) and then  $x_2$  drives position  $x_1$ , so as to control the target (Figure 6). In real servo control, this structure is the most common. Its idea is similar to the back-stepping method

of modern control theories: The complex nonlinear system is decomposed into a number of subsystems (not larger than the system order); the difference is that back-stepping selects an appropriate Lyapunov function for each subsystem, and builds an assistant virtual control variable, so as to compensate the effects of uncertain factors and thereby loop-wise reverse back to the system control input. On the contrary, through a first-order ADRC method, each loop of ADRC compensates uncertain disturbance and regulates errors, avoiding the construction of Lyapunov function and being relatively simple. Based on the ideas of ADRC cascade structure, we design an ADRC cascade controller based on inertia feedforward (Figure 6).

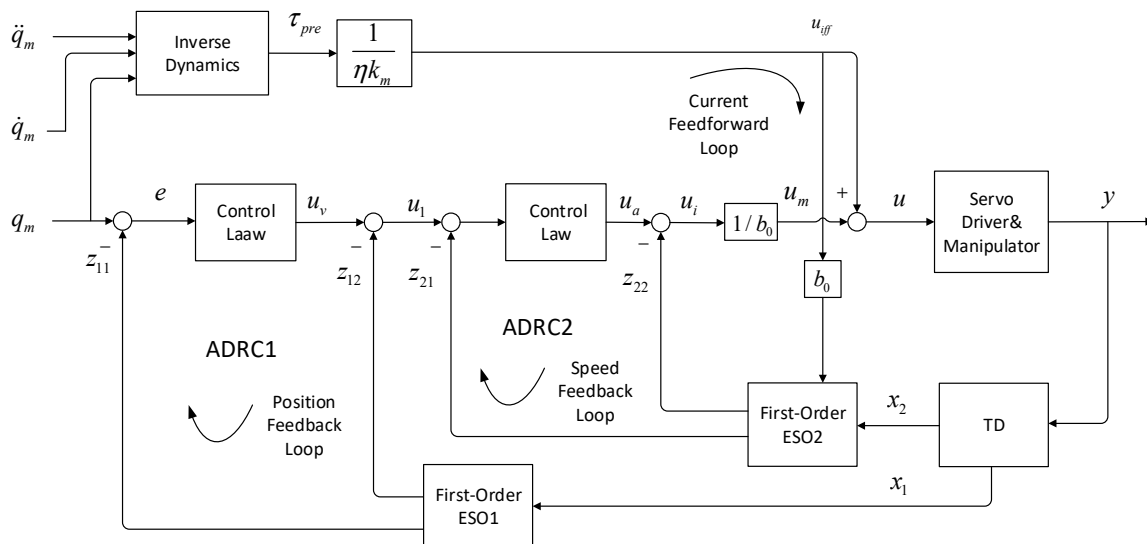


Figure 6. Active disturbance rejection control (ADRC) cascade controller based on inertia feedforward.

This ADRC feedforward controller is composed of a position loop and a velocity loop, and each loop is based on the ADRC feedforward control technology. The interior velocity loop mainly produced comprehensive current command  $u$ , which is  $u_{ff}$  mainly supplied by the current feedforward path. Then the acceleration corresponding to the uncertain errors is partially observed and compensated by the ESO2 and partially regulated by the velocity error feedback control law. The ESO disturbance observation and current feedforward can effectively reduce the power consumption, due to error control, improving the system robustness and response speed. Similarly, the exterior position loop mainly provides velocity order  $u_1$ , which mainly consists of the position error feedback control law output  $u_v$  and the ESO1 velocity disturbance  $z_{12}$ . Moreover, the loop bandwidth can be also improved to some extent by the velocity feedforward method, which ensures the rapid response to the velocity order. As for the feedback, TD has been used to acquire information about robot position and velocity, and its estimation precision modestly affects the controller performance.

It should be noted in a cascade system, the movement change of the velocity loop is faster than the position loop, which is decided by the physical properties of the controlled system. Thus, during digital computation, the velocity loop and position loop are set at different sampling step sizes (sampling period), so as to prevent the system from high-frequency vibration. Usually, the sampling period of the position loop is set as an integral multiple of that of the velocity loop and is at the millisecond level. Such setting of order density avoids the restraint of current bandwidth as faced by second order systems. Moreover, to avoid overshoot and excessive noise in each loop control order, the amplitude limiting, and low-pass filtering can be used, respectively. By combining the analyses of two control structures, we can find that each stage of the cascade controller can be considered as a first order system. Its preliminary design of control algorithm is shown below:



- Basic function

$$fal(e, \alpha, \delta) = \begin{cases} |e|^\alpha \text{sign}(e), & |e| > \delta \\ \frac{e}{\delta^{1-\alpha}}, & |e| \leq \delta \end{cases}; \quad (16)$$

- Tracking differentiator [19]

$$\begin{cases} fsg(x, d) = [\text{sign}(x + d) - \text{sign}(x - d)]/2 \\ d = Rh_0^2 \\ a_0 = h_0 \cdot x_2 \\ y = x_1 + a_0 \\ a_1 = \sqrt{d(d + 8|y|)} \\ a_2 = a_0 + \text{sign}(y)(a_1 - d)/2 \\ a = (a_0 + y)fsg(y, d) + a_2(1 - fsg(y, d)) \\ fhan(x_1, x_2, R, h_0) = -R(a/d)fsg(a, d) - R\text{sign}(a) \cdot (1 - fsg(a, d)) \end{cases}; \quad (17)$$

- Differential state

$$\begin{cases} fh = fhan(x_1 - y, x_2, r, h_0) \\ x_1 = x_1 + h_0 \cdot x_2 \\ x_2 = x_2 + h_0 \cdot fh \end{cases}. \quad (18)$$

ADRC is divided into three parts. The first part is about arranging the transition process, which is implemented in tracking differentiator, as in Equations (17) and (18), used to produce smooth instruction curves and differential curves. The second part is the design of ESO, as the first three equations in Equations (19) and (20), which are used to observe the disturbance in the corresponding loop.

- Position loop control using first order ESO1

$$\begin{cases} e = z_{11} - x_1, fe = fal(e, 0.5, h_1) \\ z_{11} = z_{11} + h_1(z_{12} - \beta_{01} \cdot e + u_1) \\ z_{12} = z_{12} + h_1(-\beta_{02} \cdot fe) \\ e_1 = q_m - z_{11} \\ u_v = \beta_1 fal(e_1, 0.5, 1.0) \\ u_1 = u_v - z_{12} \end{cases}; \quad (19)$$

- Velocity loop control using first order ESO2

$$\begin{cases} e = z_{21} - x_2, fe = fal(e, 0.5, h_2) \\ z_{21} = z_{21} + h_2(z_{22} - \beta_{03} \cdot e + b_0 u_i - b_0 u_{ff}) \\ z_{22} = z_{22} - h_2 \beta_{04} \cdot fe \\ e_2 = u_1 - z_{21} \\ u_a = \beta_2 fal(e_2, 0.5, 1.0) \\ u_i = u_a - z_{22} \\ u_m = u_i / b_0 \\ u = u_m + u_{ff} \end{cases}; \quad (20)$$

where  $h_0, h_1, h_2$  are the sampling step sizes of TD, position loop and velocity loop, respectively;  $z_{11}, z_{12}$  are the values of position and velocity disturbance observed by first-order ESO1, respectively;  $z_{11}, z_{12}$  are the values of velocity and acceleration ‘unknown disturbance’ observed by first-order ESO2, respectively. The third part of ADRC is the feedback controller. The control laws  $u_v$  in Equation (19) and  $u_i$  in Equation (20) adopt the simple P control laws combining basic function  $fal()$ . When the controller structure has been determined, appropriate ESO and PID

control laws could be designed/selected according to real situations, so as to improve the system robustness. More details can be found in [11,12,15,19].

### 3.2. System Stability Analysis

The closed-loop stability is very significant for the application of ADRC. However, the nonlinear structure of ADRC is very complex, and its stability verification is very complicated, which become the concerns in ADRC theoretical research. Guo strictly verified the convergence of high-gain ESO in nonlinear SISO systems mathematically and explored the convergence of nonlinear ESO in uncertain MIMO systems [20]. He used the absolute theory to analyze the stability of the simplified first-order ADRC closed-loop systems [21]. Zhao defined the convergence of ADRC controllers in SISO nonlinear systems and built and verified the convergence conditions of ADRC [22]. For simplification, we will analyze the necessary influence factors on system stability on the basis of system transformation, so as to characterize system stability.

#### 3.2.1. System Dynamic Compensation Linearization

The shafts in the decoupled robot system are mutually independent. A mono-joint system can be described as a nonlinear system:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = bu(t) + f(x, \dot{x}, t) + \omega(x, t) \\ y = x_1 \end{cases}, \quad (21)$$

where  $f(x, \dot{x}, t)$  is the acceleration function corresponding to the theoretical model;  $\omega(x, t)$  is the comprehensive acceleration disturbance except for the model;  $b$  was the system gain;  $u(t)$  is the control variable. The closed-loop controller is expected to have the following dynamic properties:

$$\ddot{x} = g(x, \dot{x}). \quad (22)$$

When the theoretical model  $f(x, \dot{x}, t)$  has been known, the acceleration order could be compensated by the feedforward way, and the disturbance part is compensated by the observation of ESO, or namely  $z(t) \approx \omega(t)$ . Then the system control quantity is:

$$u = \frac{(u_0 - z)}{b} - \frac{f}{b}. \quad (23)$$

At this moment, the closed-loop system became an integrator series system:

$$\ddot{x} \approx u_0(x, \dot{x}). \quad (24)$$

Admittedly, the system disturbance is compensated by the real-time estimation, which would forcefully transform the original nonlinear system into a classic linear system. Then an efficient error feedback control law  $u_0$  (e.g., nonlinear PID control law) could be used to regulate the closed-loop performance and stabilize the system. However, the premise is that the feedback from the closed-loop control system could well estimate the uncertain disturbance and could promptly pre-estimate and compensate for the acceleration disturbance to the system. Thus, ESO is a core component of ADRC, and its convergence and anti-disturbance abilities directly affect the closed-loop performance and stability of ADRC [23]. Here we further analyzed the anti-disturbance ability of ESO.

#### 3.2.2. Anti-disturbance Ability of ESO

**Lemma 1** [21]. For a first-order controlled object,

$$\dot{x} = w(x, t) + u, \quad (25)$$

where  $x$  is a state variable;  $u$  is the controlled amount;  $w(x, t)$  is the bounded disturbance of the system, and  $|w(x, t)| \leq \omega_0$ . When this system is treated with nonlinear state feedback:

$$u(x) = -k|x|^\alpha \text{sign}(x), \alpha > 0, k > 0. \tag{26}$$

The system state  $x$  under disturbance could rapidly converge to a small range of steady-state errors, then the solution domain of the closed-loop system meets:

$$\lim_{t \rightarrow \infty} |x| \leq (\omega_0/k)^{1/\alpha}. \tag{27}$$

For an  $n$ th order nonlinear SISO system:

$$\begin{cases} x^{(n)} = b_0 u(t) + f(x, \dot{x}, \dots, x^{(n-1)}, t) + w(t) \\ y = x(t) \end{cases}, \tag{28}$$

where  $f(\cdot)$  is an unknown function;  $w(t)$  is the unknown disturbance;  $b_0$  is the system gain;  $u(t)$  is the system input;  $y$  is the system output.

Let  $x_1(t) = x(t), \dots, x_n(t) = x^{(n-1)}(t), x_{n+1}(t) = a(t) = f(x, \dot{x}, \dots, x^{(n-1)}, t) + w(t)$ , then the state equation of the above system  $\Sigma_1$  becomes:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dots \\ \dot{x}_{n-1} = x_n \\ \dot{x}_n = x_{n+1} + b_0 u \\ \dot{x}_{n+1} = c(t) \\ y = x_1 \end{cases} \tag{29}$$

where  $c(t) = d[a(t)]/dt$ . For the above system, we can build an  $n$ th order ESO system  $\Sigma_2$ :

$$\begin{cases} e = z_1 - x_1 \\ \dot{z}_1 = z_2 - g_1(e) \\ \dots \\ \dot{z}_n = z_{n+1} - g_n(e) + b_0 u \\ \dot{z}_{n+1} = -g_{n+1}(e) \end{cases}. \tag{30}$$

The state of system  $\Sigma_1$  is followed:  $z_1 \rightarrow x_1, \dots, z_n \rightarrow x_n, z_{n+1} \rightarrow x_{n+1} = a$ . Let  $\delta x_1 = z_1 - x_1, \dots, \delta x_{n+1} = z_{n+1} - x_{n+1}$ , then the following error system  $\Sigma_3$  can be determined.

$$\begin{cases} \delta \dot{x}_1 = \delta x_2 - g_1(\delta x_1) \\ \dots \\ \delta \dot{x}_n = \delta x_{n+1} - g_n(\delta x_1) \\ \delta \dot{x}_{n+1} = -c(t) - g_{n+1}(\delta x_1) \end{cases} \tag{31}$$

For a bounded variation  $c(t)$ , we can select an appropriate nonlinear function  $g_i(\cdot)$  to make system  $\Sigma_3$  always stabilize at the origin and thereby make the state observation variable follow the system status. When the controlled amount coefficient  $b$  is inaccurate, its estimated value  $b_0$  could be used, and the estimation errors  $(b - b_0 u)$  is included in  $z_{n+1}(t)$ . Expression of  $g_i(\cdot)$  decides the anti-disturbance performance of ESO and is unrelated to the system, but is only related to the changing rate  $c(t)$  of the comprehensive acceleration  $a(t)$  and changing range. Thus,  $g_i(\cdot)$  could be built by using different forms of nonlinear function. Usually the following function has been selected:

$$g(\varepsilon) = \beta_{0i} \text{fal}(\varepsilon, \alpha_i, \delta_i) \quad (i = 1, 2, \dots, n + 1), \tag{32}$$

where  $\beta_{0i}$  is the gain factor, and  $\alpha_i$  designed here meets  $0 < \alpha_i \leq 1$ ,  $\delta_i = \delta \geq 0$ . From the above equation, it is the function  $fal(\cdot)$  consisted of a small-range linear interval and a large-range  $|x|^\alpha \text{sign}(x)$ . For convenience of analysis, we set:

$$g(\delta x_1) = \beta_{0i} |\delta x_1|^{\alpha_i} \text{sign}(\delta x_1). \tag{33}$$

Then the error system  $\Sigma_3$  becomes:

$$\begin{cases} \delta \dot{x}_1 = \delta x_2 - \beta_{01} |\delta x_1|^{\alpha_1} \text{sign}(\delta x_1) \\ \dots \\ \delta \dot{x}_n = \delta x_{n+1} - \beta_{0n} |\delta x_1|^{\alpha_n} \text{sign}(\delta x_1) \\ \delta \dot{x}_{n+1} = -c(t) - \beta_{0(n+1)} |\delta x_1|^{\alpha_{n+1}} \text{sign}(\delta x_1) \end{cases} \tag{34}$$

Let the changing function of the comprehensive acceleration  $c(t)$  be bounded and satisfy  $c(t) \leq \omega_0$ ; according to the prerequisite of ESO convergence, when  $t \rightarrow \infty$ , then  $\delta \dot{x}_i \rightarrow 0$ , and according to Lemma 1, the range of system steady-state error is:

$$\begin{cases} \lim_{t \rightarrow \infty} |\delta x_1| = \lim_{t \rightarrow \infty} |z_1 - x_1| < (\omega_0 / \beta_{0(n+1)})^{\frac{1}{\alpha_{n+1}}} \\ \dots \\ \lim_{t \rightarrow \infty} |\delta x_n| = \lim_{t \rightarrow \infty} |z_n - x_n| < \beta_{0(n-1)} (\omega_0 / \beta_{0(n+1)})^{\frac{\alpha_{n-1}}{\alpha_{n+1}}} \\ \lim_{t \rightarrow \infty} |\delta x_{n+1}| = \lim_{t \rightarrow \infty} |z_{n+1} - x_{n+1}| < \beta_{0n} (\omega_0 / \beta_{0(n+1)})^{\frac{\alpha_n}{\alpha_{n+1}}} \end{cases} \tag{35}$$

The right size of the above inequality presented the limits of steady-state errors in a closed-loop system. Clearly, the anti-disturbance ability of ESO is decided by the parameter setting. Let  $\sigma = \beta_{0i} (\omega_0 / \beta_{0(n+1)})^{\frac{\alpha_i}{\alpha_{n+1}}}$ , when  $\beta_{0(n+1)}$  is large enough, the system steady-state error is very small, but easily leads to vibration of ESO system. When  $0 < \alpha_i < \alpha_{i+1} < \dots < \alpha_{n+1} < 1$ , we have  $\sigma < \beta_{0i} (\omega_0 / \beta_{0(n+1)})$ , indicating compared with linear feedback, the nonlinear feedback showed smaller steady-state error and stronger following ability and anti-disturbance ability. Moreover, the high-order system usually adopts a high-order ESO, but the amplitude  $\omega_0$  of highest-order changing rate is very large, leading to larger  $\beta_{0(n+1)}$ , which not only complicates the parameter adjustment, but also easily makes the ESO unable to converge, or even the ADRC system unstable. Thus, ADRC systems at 3rd order or above are usually decomposed by the time scale into cascade systems, while the ESO adopts the low-order series form. Thereby, it is indicated a low-order cascade system contributes to improving the anti-disturbance ability of ESO and thereby improves the stability of ADRC systems.

#### 4. Experimental Results and Analysis

To validate the effectiveness of the new control method, space arc trajectory follow-up experiments were conducted in a six-degree-of-freedom robot platform, in which the 6R industrial robots HSR-JR605-C (its parameters are in Table 1) were used. Cartesian space arc trajectory was selected. The robot load was 3.6 kg, the initial attitude was  $q = [0, -\pi/2, 0, 0, \pi/2, 0]^T$ ; the space arc trajectory was determined from the three space coordinates of the end-point P, where,  $A = [376.5, 0, 331]$ ,  $B = [396.5, 20, 331]$ ,  $C = [416.5, 0, 331]$ , and arc center  $O = [396.5, 20, 331]$  in mm, as shown in Figure 7.

Table 1. Modified D-H parameters of Robot.

Joint $i$	Link Twist $\alpha_i$ ( $^\circ$ )	Link Length $a_i$ (mm)	Link Offset $d_i$ (mm)	Joint Angle
1	0	0	0	$q_1$
2	-90	0	0	$q_2$
3	0	360	0	$q_3$
4	-90	90	376.5	$q_4$
5	90	0	0	$q_5$
6	-90	0	0	$q_6$

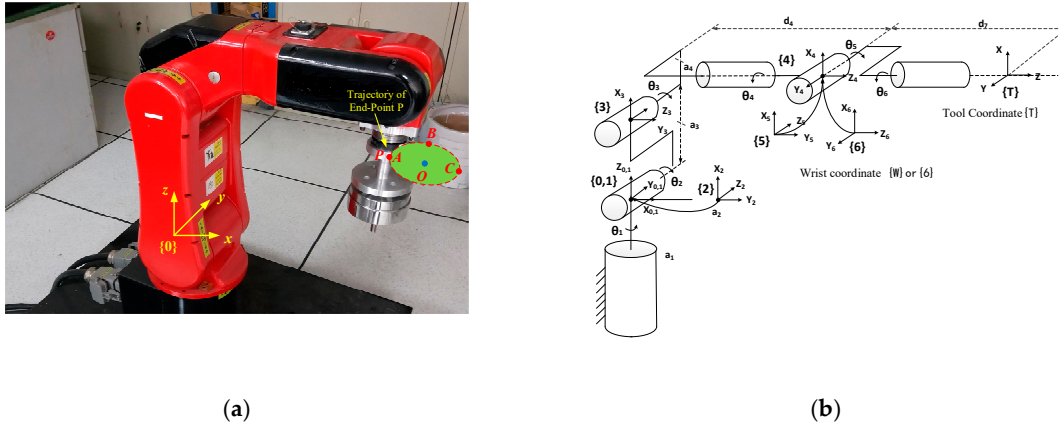


Figure 7. Experimental robot. (a) Robot end-point end P trajectory; (b) coordinate system.

Point P is accelerated and clockwise along an arc from point A to point C, and then returns clockwise along from point C to point A. With the Cartesian space circular interpolation algorithm and the inverse kinematics computation, we could determine the joint motion status order. The input of this order is to convert the transmission ratio into motor end motion. It should be noted the on-line computation of the inverse dynamic model is much time-consuming, which may affect the timeliness of the controlled loops. Thus, here it is set as ‘background program’, and the computation memory is pre-stored and extracted in real-time. To validate the superiority of the new control algorithm in Figure 8, we select the commonly-used PI control algorithm as a comparison:

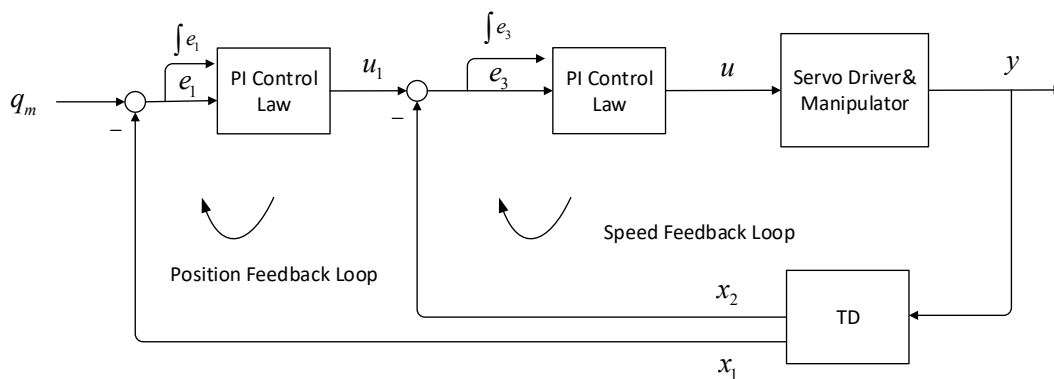


Figure 8. Robot uniaxial PI controller.

Its control algorithm consists of:

- Differential state estimation

$$\begin{cases} fh = fhan(x_1 - y, x_2, R, h_0) \\ x_1 = x_1 + h_0 \cdot x_2 \\ x_2 = x_2 + h_0 \cdot fh \end{cases} ; \tag{36}$$

- Position loop control algorithm

$$\begin{cases} e_1 = q_m - x_1 \\ e_0 = \int e_1 dt \\ u_1 = \beta_0 fal(e_0, 0.25, 1.0) + \beta_1 fal(e_1, 0.5, 1.0) \end{cases} ; \tag{37}$$

- Velocity loop control algorithm

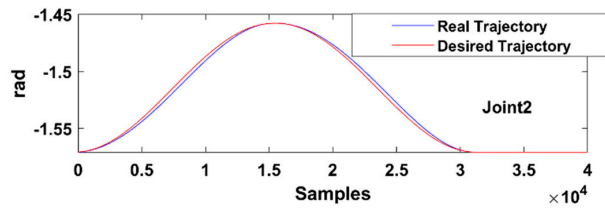
$$\begin{cases} e_3 = u_1 - x_2, e_2 = \int e_3 dt \\ u = \beta_2 fal(e_2, 0.25, 1.0) + \beta_3 fal(e_3, 0.5, 1.0) \end{cases} . \tag{38}$$

As for the indices of response speed and trajectory following precision, we investigate the following trajectories of different joints in the joint coordinate system and the following trajectory of the end-point position in the Cartesian coordinate system. To reasonably evaluate the controller performance, we have to acquire accurate real-time robot joint position and attitude states and the Cartesian coordinates of the end-point ends, and when the joint flexibility has been ignored and for simplification, we recognize that the coder positional information directly reflected the joint positional information. The end-point Cartesian coordinates were acquired from robot forward kinematics, without any external measuring tool. Based on the semi-closed-loop control structure, we regulate the parameters of the two controllers to optimize the following performance. The parameters of the two controllers in Equations (15–20) and (36–38) are listed in Table 2.

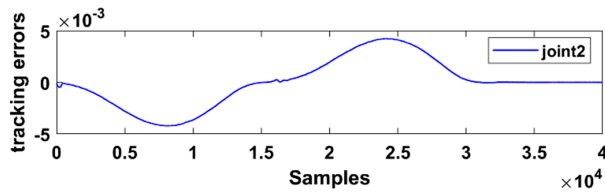
**Table 2.** Robot controller parameters.

<b>Electromechanical Parameters</b>	Torque constant $k_m = [0.354, 0.354, 0.283, 0.276, 0.276, 0.226]$ (Nm/A), Gear ratio $\eta = [81, 100, 80, 76.5, 81, 50]$ , Motor shaft rotary inertia $J_m = [0.130, 0.130, 0.044, 0.027, 0.027, 0.005]$ ( $kgm^2 \times 10^{-3}$ )	
	TD	$R = 40, h_0 = 0.0002$
<b>ADRC feedforward controller (6-shafts)</b>	ESO1 :	$\beta_{01} = [500, 500, 500, 500, 500, 500]$ $\beta_{02} = [5000, 5000, 5000, 5000, 5000, 5000]$ $h_1 = 0.002$
	ESO2 :	$\beta_{03} = [5000, 5000, 5000, 5000, 5000, 5000]$ $\beta_{04} = [50000, 50000, 45000, 40000, 40000, 40000]$ $h_2 = 0.0002$
		$b_0 = J_m^{-1} k_m \zeta, \zeta = [0.950, 0.950, 0.980, 0.980, 0.980, 0.990]$
	P	Position Loop: $\beta_1 = [400, 400, 350, 250, 200, 200]$ Speed Loop: $\beta_2 = [5000, 5000, 4000, 4000, 4000, 4000]$
	$\delta = 1/\eta k_m$	$\delta = [0.03487, 0.02824, 0.04427, 0.04736, 0.04473, 0.08850]$
<b>PI controller (6-shafts)</b>	TD	$R = 40, h_0 = 0.0002$
	PI	Position Loop: $\beta_0 = [0, 0, 0, 0, 0, 0]$ $\beta_1 = [1000, 1000, 800, 500, 500, 500]$
		Speed Loop: $\beta_2 = [0, 0, 0, 0, 0, 0]$ $\beta_3 = [10000, 10000, 8000, 8000, 5000, 5000]$

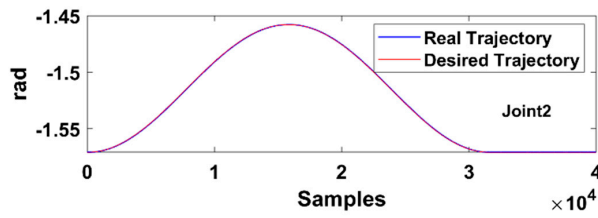
The servo periods of the position loop and the velocity loop were 2 ms and 200  $\mu$ s, respectively, and the sampling period was 200  $\mu$ s. Let the angular velocity of the end-point P of the operating arm revolving around the circle center O be  $w = 2, 4, 6, 8$  (rad/s). The representative joint trajectory follow-up results and end-point Cartesian trajectory follow-up results of joint 2 are shown in Figures 9–11. The end-point trajectory follow-up results at different speeds are shown in Figure 12.



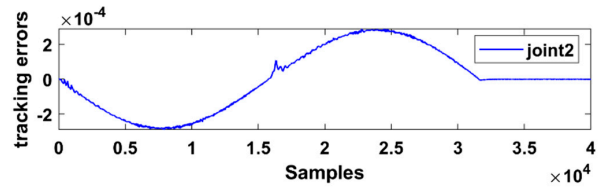
(a)



(b)

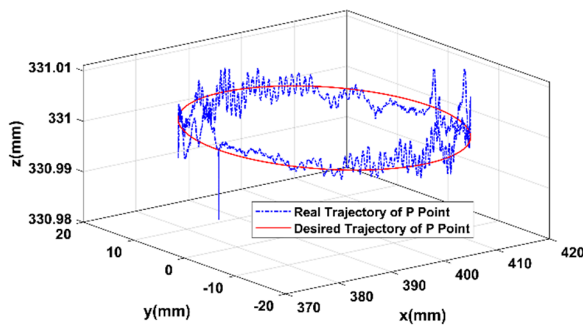


(c)

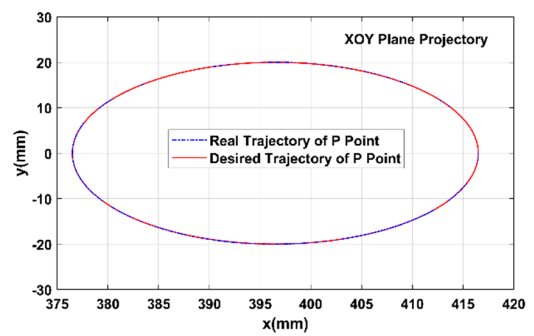


(d)

**Figure 9.** The trajectory follow-up results and error curve of joint 2 in different controllers ( $w = 2$  rad/s). (a) trajectory follow-up curve of PI controller; (b) trajectory following errors of PI controller; (c) trajectory follow-up curve of ADRC controller; (d) trajectory following error of ADRC controller based on dynamic feedforward.

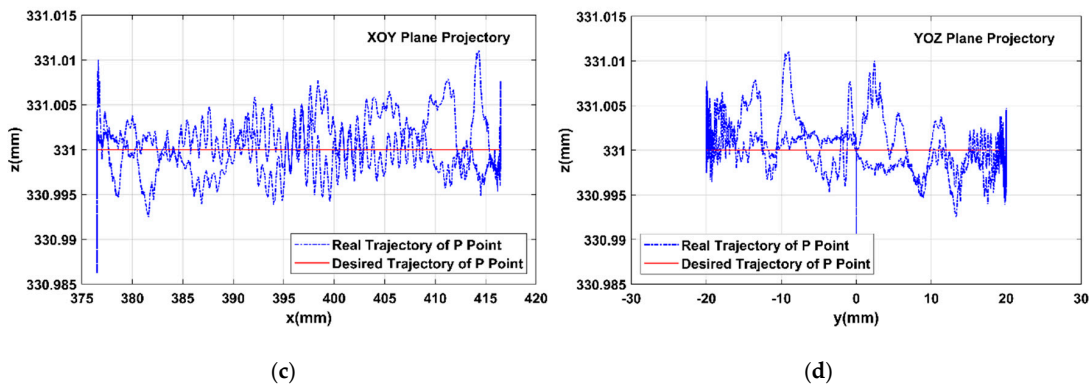


(a)

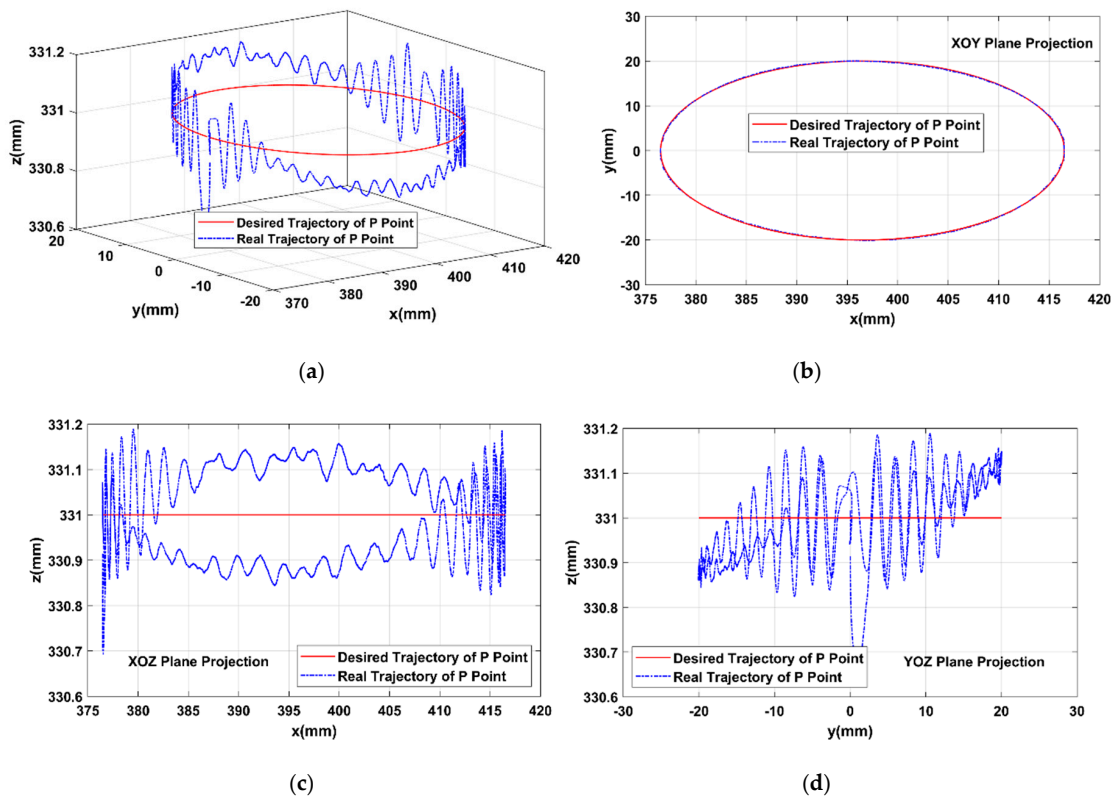


(b)

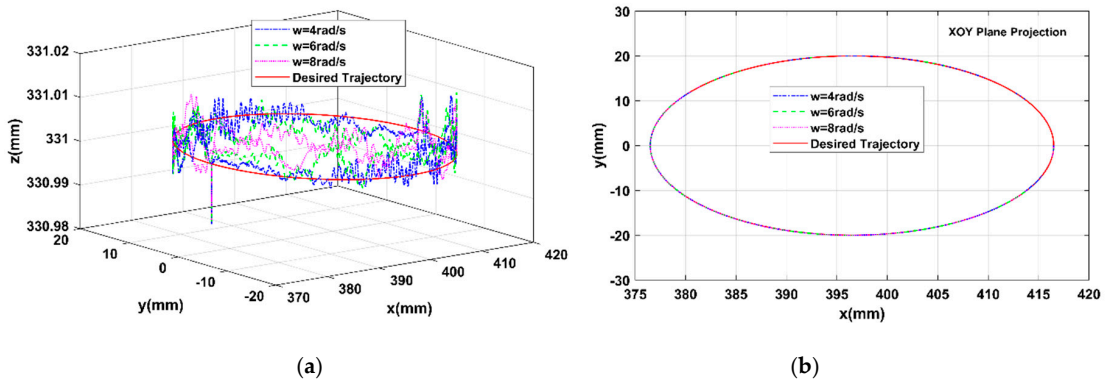
**Figure 10.** Cont.



**Figure 10.** Robot end-point Cartesian follow-up trajectory using ADRC controller ( $w = 2 \text{ rad/s}$ ). (a) 3D image; (b) XOY planar projection; (c) XOZ planar projection; (d) YOZ planar projection.

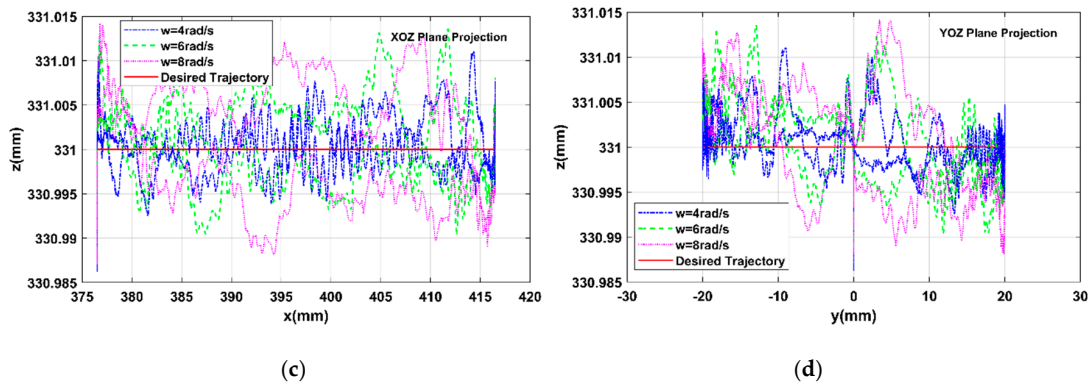


**Figure 11.** Robot end-point Cartesian following trajectory using PI controller ( $w = 2 \text{ rad/s}$ ). (a) 3D image; (b) XOY planar projection; (c) XOZ planar projection; (d) YOZ planar projection.



**Figure 12.** Cont.





**Figure 12.** Robot end-point Cartesian feedforward forward controller at different speeds. (a) 3D image; (b) XOY planar projection; (c) XOZ planar projection; (d) YOZ planar projection.

The above figures show the PI controller and the ADRC controller based on dynamic feedforward (or ADRC controller for short) both could stably follow the trajectory, but the following precision and response speeds both differed. In the joint coordinate system, for the large-inertia joint 2, the trajectory following error of the PI controller was very large (about 0.005 rad) (Figure 9b), which was higher than that of the ADRC controller (Figure 9d). As for the response speed, the new controller also outperformed the PI controller (Figure 9a,c). Clearly, the ADRC controller outperformed the PI controller in both control precision and response speed, but the joint end-point following trajectory was more precise in the Cartesian coordinate system (Figure 10). As for planar projection, the ADRC feedforward showed control precision over the robot end-point end up to 0.015 mm (Figure 10c,d), which was far higher than the PI controller (0.3 mm, Figure 11c,d). As for error distributions, the end-point following error of the robot was very large upon the start-up, which was largely induced by the joint friction. Firstly, the joint static friction after the start of movement was very large, which could not be rapidly adjusted by a pure PI controller; secondly, due to the errors of the friction model, the static friction at the time of mutation was hard to be monitored or compensated promptly. Admittedly, relative to the PI controller, our new ADRC controller could well inhibit the friction uncertain disturbance, and generally, was outstanding in terms of control precision and response speed. Moreover, at different speeds, especially high speed (8 rad/s), the ADRC controller showed both high-quality trajectory follow-up and robustness (Figure 12).

To verify the position repeatability of robot, we set the start/stop point A (376.5, 0, 331), and control the robot end-point to perform a periodic circular motion to stop at point A for 20 times ( $w = 2 \text{ rad/s}$ ). The end-point Cartesian coordinates ( $\bar{x}$ ,  $\bar{y}$  and  $\bar{z}$  in Table 3) were acquired from robot forward kinematics based on the sampled information of encoders.

The  $t_i$  and  $\delta$  are expressed in Equations (39) and (40), respectively.

$$t_i = \sqrt{(x - \bar{x})^2 + (y - \bar{y})^2 + (z - \bar{z})^2}, \tag{39}$$

where  $(x, y, z) = (376.5, 0, 331); i = 1, 2, 3, \dots, 20$ .

$$\delta = \sqrt{\frac{\sum_{i=1}^n (\bar{t} - t_i)^2}{n - 1}}, \tag{40}$$

where  $\bar{t} = \frac{1}{n} \sum_{i=1}^n t_i, n = 20$ .

It can be seen that the repetitive position accuracy is slightly higher than the trajectory following accuracy. However, it should be noted that the actual repetitive position accuracy will be slightly worse

because we acquire the end-point Cartesian coordinates from the encoders and theoretical calculation based on robot forward kinematics.

**Table 3.** Position repeatability of robot end-point.

Samples	$\bar{x}$ (mm)	$\bar{y}$ (mm)	$\bar{z}$ (mm)	$t_i$ (mm)
1	376.505	0.004	331.008	0.010
2	376.497	0.002	331.001	0.003
3	376.495	−0.006	331.004	0.008
4	376.497	0.004	331.004	0.006
5	376.505	0.001	330.994	0.008
6	376.502	0.007	331.005	0.009
7	376.501	−0.005	331.002	0.005
8	376.503	−0.003	330.992	0.009
9	376.492	−0.004	331.003	0.009
10	376.494	−0.004	330.995	0.009
11	376.505	0.007	331.005	0.010
12	376.508	0.002	330.998	0.008
13	376.501	0.009	331.005	0.010
14	376.498	0.005	331.002	0.006
15	376.505	0.005	330.998	0.007
16	376.501	−0.004	330.995	0.006
17	376.502	0.003	331.006	0.007
18	376.503	0.01	330.999	0.010
19	376.498	−0.005	331.001	0.005
20	376.502	0.007	331.003	0.007
$\delta$		0.0079 (mm)		
$3\delta$		0.0237 (mm)		

## 5. Conclusions

To improve the robustness of the dynamic feedforward control system, we put forward an ADRC strategy based on dynamic feedforward starting from the control perspective and combining the simple structure, strong anti-interference ability and no restraint from the control model of ADRC.

(1) Together with the dynamic characteristics of robots, we analyze the system disturbance compensation methods, and by regarding robots as motor output load, achieve multi-joint control decoupling through dynamic characteristic compensation and uncertain disturbance compensation.

(2) As for the uniaxial independent systems after decoupling, we study the ADRC series control structure and algorithm based on dynamic feedforward, and investigate the closed-loop stability by analyzing the dynamic linearization compensation and ESO anti-disturbance ability.

(3) Space arc trajectory following experiments have been conducted to comprehensively evaluate the 6-joint motion performances of robots. The ADRC controller based on dynamic feedforward outperforms traditional PI controllers in terms of trajectory following precision and response speed, and shows high values of practical engineering. It should be noted ADRC controllers are structurally diverse, and this study just proves the high efficiency of the P control law. To further improve the controller performance, other more advanced PD control law and ESO combination may be used, which is not elaborated here.

**Author Contributions:** Methodology, R.Z.; data curation, X.T.; writing—original draft preparation, X.T.; writing—review and editing, X.C.; supervision, Y.Z.

**Funding:** This work was supported by National Natural Science Funds of China under Grants 51575411 and 51375194, the Fundamental Research Funds for the Central Universities under Grants WUT2019III143CG.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Craig, J.J. *Introduction to Robotics: Mechanics and Control*; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2005.
2. Su, Y.X.; Duan, B.Y.; Zheng, C.H. Nonlinear PID control of a six-DOF parallel manipulator. *IEEE Proc. Control Theory Appl.* **2004**, *151*, 95–102. [[CrossRef](#)]
3. Khoury, G.M.; Saad, M.; Kanaan, H.Y.; Asmar, C. Fuzzy PID control of a five DOF robot arm. *J. Intell. Rob. Syst.* **2004**, *40*, 299–320. [[CrossRef](#)]
4. Sharma, R.; Kumar, V.; Gaur, P.; Mittal, A.P. An adaptive PID like controller using mix locally recurrent neural network for robotic manipulator with variable payload. *ISA Trans.* **2016**, *62*, 258–267. [[CrossRef](#)] [[PubMed](#)]
5. Thanh, T.D.C.; Ahn, K.K. Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network. *Mechatronics* **2006**, *16*, 577–587. [[CrossRef](#)]
6. Norrlof, M. An adaptive iterative learning control algorithm with experiments on an industrial robot. *IEEE Trans. Robot. Autom.* **2002**, *18*, 245–251. [[CrossRef](#)]
7. Etxebarria, O.B.V. Robust adaptive control for robot manipulators with unmodelled dynamics. *Cybern. Syst.* **2000**, *31*, 67–86. [[CrossRef](#)]
8. Er, M.J.; Gao, Y. Robust adaptive control of robot manipulators using generalized fuzzy neural networks. *IEEE Trans. Ind. Electron.* **2003**, *50*, 620–628.
9. Hong, Y.; Yao, B. A globally stable high-performance adaptive robust control algorithm with input saturation for precision motion control of linear motor drive systems. *IEEE/ASME Trans. Mechatron.* **2007**, *12*, 198–207. [[CrossRef](#)]
10. Hu, C.; Yao, B.; Wang, Q. Adaptive robust precision motion control of systems with unknown input dead-zones: A case study with comparative experiments. *IEEE Trans. Ind. Electron.* **2011**, *58*, 2454–2464. [[CrossRef](#)]
11. Chen, Z.Q.; Wang, Y.S.; Sun, M.W.; Sun, Q.L. Convergence and stability analysis of active disturbance rejection control for first-order nonlinear dynamic systems. *Trans. I. Meas. Control* **2019**, *41*, 2064–2076. [[CrossRef](#)]
12. Gao, Z. Active disturbance rejection control: A paradigm shift in feedback control system design. In Proceedings of the 2006 American Control Conference (ACC 2006), Minneapolis, MN, USA, 14–16 June 2006.
13. Zheng, Q.; Gao, Z. On practical applications of active disturbance rejection control. In Proceedings of the 29th Chinese Control Conference (CCC 2010), Beijing, China, 29–31 July 2010.
14. Zhao, Z.L.; Guo, B.Z. A novel extended state observer for output tracking of MIMO systems with mismatched uncertainty. *IEEE Trans. Autom. Control* **2017**, *63*, 211–218. [[CrossRef](#)]
15. Guo, B.; Zhao, Z. On convergence of non-linear extended state observer for multi-input multi-output systems with uncertainty. *IET Control Theor. Appl.* **2012**, *6*, 2375–2386. [[CrossRef](#)]
16. Talole, S.E.; Kolhe, J.P.; Phadke, S.B. Extended-state-observer-based control of flexible-joint system with experimental validation. *IEEE Trans. Ind. Electron.* **2010**, *57*, 1411–1419. [[CrossRef](#)]
17. Medjebouri, A.; Mehennaoui, L. Active Disturbance Rejection Control of a SCARA Robot Arm. *Int. J. U E Serv. Sci. Tech.* **2015**, *8*, 435–446. [[CrossRef](#)]
18. Spong, M.W.; Vidyasagar, M. *Robot Dynamics and Control*; John Wiley & Sons: New York, NY, USA, 2008.
19. Tu, X.; Zhou, Y.F.; Zhao, P.; Cheng, X. State estimation of a robot joint by a novel nonlinear tracking differentiator. *Ind. Robot.* **2018**, *45*, 11–22. [[CrossRef](#)]
20. Guo, B.; Zhao, Z. On the convergence of an extended state observer for nonlinear systems with uncertainty. *Syst. Control Lett.* **2011**, *60*, 420–430. [[CrossRef](#)]
21. Lingyun, H. *Auto Disturbance Rejection Control of Maglev System*; National University of Defense Technology: Changsha, China, 2006.
22. Zhao, Z.; Guo, B. On Convergence of Nonlinear Active Disturbance Rejection Control for SISO Nonlinear Systems. *J. Dyn. Control Syst.* **2016**, *22*, 385–412. [[CrossRef](#)]
23. Wu, Z.; Guo, B. On convergence of active disturbance rejection control for a class of uncertain stochastic nonlinear systems. *Int. J. Control* **2017**, 1–14. [[CrossRef](#)]

