

Article

Eliminating the Effect of Uncertainties of Cutting Forces by Fuzzy Controller for Robots in Milling Process

Khoi Bui Phan ^{1,*} , Hai Thanh Ha ^{1,2} and Sinh Vinh Hoang ¹

¹ School of Mechanical Engineering, Hanoi University of Science and Technology, 01 Dai Co Viet, Hai Ba Trung, Hanoi 100000, Vietnam; hathanhhai1976@gmail.com (H.T.H.); sinh.hoangvinh@hust.edu.vn (S.V.H.)

² College of Urban Works Construction, Yen Thuong, Gia Lam, Ha Noi 100000, Vietnam

* Correspondence: khoi.phanbui@hust.edu.vn; Tel.: +84-913525160

Received: 8 January 2020; Accepted: 26 February 2020; Published: 2 March 2020



Abstract: This study presents a method of controlling robots based on fuzzy logic to eliminate the effect of uncertainties that are generated by the cutting forces in milling process. The common method to control industrial robots is based on the robot dynamic model and the differential equations of motion to compute the control values. The quantities in the differential equations of the motion of robots are complex and difficult to determine fully and accurately. The interaction forces between the cutting tool and the workpiece are the cutting forces, which are generated during the machining process. It is difficult to calculate the cutting force because it depends on many factors such as material of the machining part, depth of cut, feed rate, etc. This article presents the fuzzy rule system and the selection of the physical value domain of input and output variables of the fuzzy controller. The fuzzy rules are applied in this article to allow us to compute the driving forces based on the errors of input and output signals of the joint positions and velocities, thereby avoiding the calculation of cutting forces. This article shows the simulation results of the fuzzy controller and comparison with the results of the conventional controller when the dynamic model is assumed to be correctly determined. The achieved results are reliable and facilitate the research and application of a fuzzy controller to mechanical processing robots in general and milling machining in particular.

Keywords: milling machining robot; fuzzy logic; fuzzy control; fuzzy control law; cutting force

1. Introduction

Robots are increasingly being used in mechanical machining due to many technical and economic advantages [1–5]. Robots have many degrees of freedom and a structure with many links and joints enabling them to move flexibly in a large workspace. They have machine parts with complex geometric surfaces, large dimensions, and a minimum number of machining operations and simple jigs [3]. Machining by industrial robots is more economical than machine tools and according to the calculation of the COMET, the price advantage is up to 30% [6]. However, there are many difficulties and challenges in exploiting the potential of robot applications [2,7]. The structure with many links and joints will give the robot the ability to manipulate flexibly as mentioned, but it is difficult when modeling kinematics and dynamics to control the robots. The expressions of the kinematic and dynamic quantities in the differential equations of motion of the robots are often cumbersome and complex. The machining process has many factors, which are difficult to fully and accurately determine, for example, chatter, vibration, or the fact that the perturbations may appear abnormally [8–10]. In particular, the cutting force generated during machining is an uncertain factor and has a great effect on the determination of

control force based on the dynamic model [11–32]. The mechanical machining processes often require high accuracy, especially when milling to shape complex surfaces, which requires even more rigorous accuracy of shaping motion. In order to solve these problems, there have been many study directions to improve and enhance the machining accuracy of robots. One of the solutions is controlling based on fuzzy logic to overcome the difficulty of determining cutting forces while ensuring control accuracy. Fuzzy logic is an effective tool for solving many technical problems in general [33–43], in mechanical processing and in industrial robot applications in particular.

There have been many studies on the effect of cutting forces based on fuzzy logic [44–47]. In addition, there are also many studies on robot control based on fuzzy logic [48–61]. However, the application of fuzzy logic to control robots to eliminate the effect of cutting forces is still an open problem [62–67].

The paper proposes a fuzzy law system for a controller to calculate and determine the control forces (driving forces/torques of the motors) based on errors between input and output signals on the joint positions and velocities. The computations based on the control law are simple linear algebra calculations that are easier and faster to perform than calculating the generalized force expression of cutting forces in the differential equations of motion of robots.

It is assumed that the dynamic model of robots is precisely determined, including the cutting forces in the differential equations of motion, which is considered to be accurately determined. This article presents the application of a common (conventional and popular) method in controlling industrial robots, meaning the inverse dynamics method combined with the law of derivative proportional control (PD), hereinafter referred to as the Inverse Dynamics + PD method. With the above assumption, the Inverse Dynamics + PD controller is considered an accurate one. To distinguish, the controllers based entirely on the dynamic model are called (hereinafter) explicit (crisp) controllers.

The article shows the results of simulating the fuzzy controller in comparison with the results of simulating the above explicit and accurate controller. With the different assumption of cutting force calculation, that is, choosing the different influenced coefficients in calculating cutting forces, the Inverse Dynamics + PD controller will give different results, meaning that the control errors are significant. With the fuzzy controller, due to the determination of cutting forces not being required, even if the components of the cutting force change, or the occurrence of perturbation, then the controller's performance will still ensure the accuracy of desired motion laws (desired trajectories).

The remains of this article is organized as follows. Section 2 briefly presents robot kinematic modeling based on the authors of [68–72] to indicate necessary computations for receiving the motion trajectory of robots, which is used for the following studies. Section 3 discusses dynamics modeling, presents the derivation of robot's dynamic model, and analyzes the calculation of cutting forces with an empirical formula. Section 4 presents a familiar control method that is used as a basis for comparison. Section 5 presents the steps to build the fuzzy controller, and proposes a fuzzy control law that is suitable for a controller of a robot in milling machining. Section 6 presents the results of simulation, calculation and comparative analysis. Section 7 presents the conclusions, analyzing the possibility of applying the proposed controller, and discusses the directions of future studies.

2. Robot Kinematic Modeling

The system of the robot has a clamping platform with eight degrees of freedom (DOF), including a robot and a platform, which allows expanding the machining space (Figure 1). The robot is designed according to the robot model of Asea Brown Boveri, which is a six DOF serial robot, consisting of six moveable links connected to a fixed base. The cutting tool is mounted on the end effector of the robot. The platform has two DOF, including two moveable links; the workpiece is fixed on the upper link. Names of the robot links and platform links of the system in turn are: L0, L1, . . . , L6, B and D, respectively. Link L0 is the base, link L6 is the end effector, link B connects the prismatic joint to the fixed base, and link D carries the workpiece. The robot and the clamping platform coordinate programmatically to perform the form-shaping motion of the machining surface.

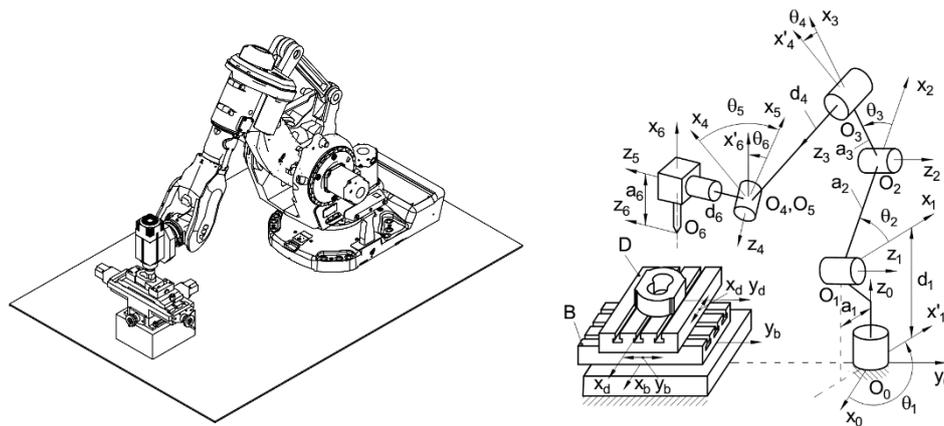


Figure 1. The model of robot; clamping platform has eight degrees of freedom (DOF) in milling process.

Normally, the machining object is clamped on a fixed platform, and a robot with 6 DOF is sufficient to obtain positions and orientations of the end effector for the machining performance. However, for the purpose of expanding the operational space of the robot, or choosing a convenient machining posture, a machining system with a two DOF clamping platform is utilized. As a result, the robot can machine large parts with many complex surfaces, reducing the number of clamping operations. Robots with a combination of two such component robots or more are often referred to Mechanism of Relative Manipulation robots—MRM [73–81]. For a simple presentation, the robot–platform system is hereinafter called the robot system or the robot.

The article applies the Denavit–Hartenberg convention to define the link frames and derive kinematic equations of the robot kinematic chains, using generalized coordinates to represent and calculate the kinematic chain of the platform. The coordinate systems, kinematic parameters and corresponding homogeneous coordinate transformation matrices are shown in Table 1.

In Table 1, $O_i x_i y_i z_i$ ($i = 1, \dots, 6$) is the frame that is attached to the end of link i , and represents the position and orientation of frame $O_i x_i y_i z_i$ with respect to frame $O_{i-1} x_{i-1} y_{i-1} z_{i-1}$ by the Denavit–Hartenberg homogeneous coordinate transformation matrix ${}^{i-1}A_i$, with respect to frame $O_0 x_0 y_0 z_0$ by matrix 0A_i . $O_6 x_6 y_6 z_6$ is called the end-effector frame. $O_0 x_0 y_0 z_0$ is called the base frame for the entire robot–platform system, with matrix E being the 4×4 identity matrix that represents the position and orientation of the base frame with respect to itself. The parameters in the third column of Table 1 represent the relative positions between two consecutive frames, in which θ_i ($i = 1, \dots, 6$), y_b, x_d are the joint coordinates, and the other parameters are constants.

The vector of joint coordinate is denoted by q , represented by Equation (1).

$$q = [q_1, \dots, q_8]^T = [\theta_1, \dots, \theta_6, y_b, x_d]^T \tag{1}$$

$O_0' x_0' y_0' z_0'$ is the fixed frame that has the origin at position O_0' and the axes parallel to the corresponding ones of the base frame, used to location the platform.

$O_b x_b y_b z_b$ is the frame that attached to link B; it is possible to translate in the direction $y_b // y_0$, corresponding to the joint coordinate y_b , the coordinate transformation matrices as shown in Table 1. $O_d x_d y_d z_d$ is the frame that attached to link D; it is possible to translate in the direction $x_d // x_0$, corresponding to the joint coordinate x_d , the coordinate transformation matrices as shown in Table 1.

Frame $O_d x_d y_d z_d$, hereinafter called the platform frame or the jig frame, is used to locate the workpiece. The parameters of the workpiece’s machined surface and toolpath are shown in the jig frame.

In the machining process, the tool performs the relative motion with respect to the machined surface according to the law (trajectory) that is determined by the machining engineering. This trajectory is represented by parameters ${}^d x_E, {}^d y_E, {}^d z_E, {}^d \alpha_E, {}^d \beta_E, {}^d \eta_E$ (Table 1) and their first and second

time derivatives with respect to time. Thereby, the vector of operational coordinates is called p , represented by (2).

$$p = {}^d p_E = [{}^d p_1, {}^d p_2, \dots, {}^d p_6]^T = [{}^d x_E, {}^d y_E, {}^d z_E, {}^d \alpha_E, {}^d \beta_E, {}^d \eta_E]^T \tag{2}$$

Table 1. The frames and homogeneous coordinate transformation matrices.

Links	Frame	Kin. Parameters	${}^{i-1}A_i$	0A_i
The robot kinematic chain				
L_0	$O_0x_0y_0z_0$	$0, \dots, 0$	E	E
L_1	$O_1x_1y_1z_1$	$\theta_1, d_1, a_1, \alpha_1$	${}^0A_1(\theta_1)$	${}^0A_1(\theta_1)$
...
L_6	$O_6x_6y_6z_6$	$\theta_6, d_6, a_6, \alpha_6$	${}^5A_6(\theta_6)$	${}^0A_1(\theta_1) \dots {}^5A_6(\theta_6)$
Cutting tool	$O_Ex_Ey_Ez_E$	${}^6x_E, {}^6y_E, {}^6z_E, {}^6\alpha_E, {}^6\beta_E, {}^6\eta_E$	6A_E	${}^0A_1(\theta_1) \dots {}^5A_6(\theta_6) {}^6A_E$
The platform kinematic chain				
L_0	$O_0x_0y_0z_0$	$0, \dots, 0$	E	E
$L_{0'}$	$O_{0'}x_{0'}y_{0'}z_{0'}$	$x_{0'}, y_{0'}, z_{0'}, \alpha_{0'}, \beta_{0'}, \eta_{0'}$	${}^0A_{0'}$	${}^0A_{0'}$
B	$O_bx_by_bz_b$	$x_b, y_b, z_b, \alpha_b, \beta_b, \eta_b$	${}^{0'}A_b(y_b)$	${}^0A_{0'} {}^{0'}A_b(y_b)$
D	$O_dx_dy_dz_d$	$x_d, y_d, z_d, \alpha_d, \beta_d, \eta_d$	${}^bA_d(x_d)$	${}^0A_{0'} {}^{0'}A_b(y_b) {}^bA_d(x_d)$
Cutting tool	$O_Ex_Ey_Ez_E$	${}^d x_E, {}^d y_E, {}^d z_E, {}^d \alpha_E, {}^d \beta_E, {}^d \eta_E$	${}^d A_E$	${}^0A_{0'} {}^{0'}A_b(y_b) {}^bA_d(x_d) {}^d A_E$

Therefore, the frame $O_dx_dy_dz_d$ is used as the operational frame.

Frame $O_Ex_Ey_Ez_E$ is attached to the cutting edge of the cutter on the end effector, called the tool frame.

According to the robot kinematic chain, matrix 0A_E represents the position and orientation of the tool frame $O_Ex_Ey_Ez_E$ with respect to the base frame $O_0x_0y_0z_0$, determined by (3).

$${}^0A_E = {}^0A_1(\theta_1) {}^1A_2(\theta_2) \dots {}^5A_6(\theta_6) {}^6A_E \tag{3}$$

According to the platform kinematic chain, 0A_E is determined by (4).

$${}^0A_E = {}^0A_{0'} {}^{0'}A_b(y_b) {}^bA_d(x_d) {}^dA_E(p) \tag{4}$$

From (3) and (4), we obtained (5).

$${}^dA_E(p) = {}^bA_d^{-1}(x_d) {}^{0'}A_b^{-1}(y_b) {}^0A_{0'}^{-1} {}^0A_1(\theta_1) \dots {}^5A_6(\theta_6) {}^6A_E \tag{5}$$

The right hand side of (5) is the matrix representing the position and orientation of the cutting tool with respect to the platform frame $O_dx_dy_dz_d$, whose elements are functions of the joint coordinates (1). Meanwhile, the elements of the matrix on the left hand side of (5) are functions of the operational coordinates (2). (5) is rewritten by (6).

$$\begin{bmatrix} {}^dC_p({}^d\alpha_E, {}^d\beta_E, {}^d\eta_E) & {}^d r_E({}^d x_E, {}^d y_E, {}^d z_E) \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} {}^dC_E(q) & {}^d r_E(q) \\ 0^T & 1 \end{bmatrix} \tag{6}$$

Equation (6) is called the kinematic equation of robot in matrix form.

For each specific machining case, the requirements on shaping motion of the tool are given. That is, to the scope of this study, the operational coordinates (2) and its derivatives are known. Applying the accompanying trihedron method [68–72] together with the kinematic Equation (6) allows us to

calculate the motion trajectory of the robot. That is, the joint coordinates (1) and their first and second time derivatives with respect to time are calculated, which are used as the set trajectories (7) for the control problem in the following section.

$$\begin{cases} \mathbf{q}^d = [\theta_1, \theta_2, \dots, \theta_6, y_b, x_d]^T = [q_1^d, \dots, q_8^d]^T \\ \dot{\mathbf{q}}^d = [\dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_6, \dot{y}_b, \dot{x}_d]^T = [\dot{q}_1^d, \dots, \dot{q}_8^d]^T \\ \ddot{\mathbf{q}}^d = [\ddot{\theta}_1, \ddot{\theta}_2, \dots, \ddot{\theta}_6, \ddot{y}_b, \ddot{x}_d]^T = [\ddot{q}_1^d, \dots, \ddot{q}_8^d]^T \end{cases} \quad (7)$$

3. Robot Dynamic Modeling

3.1. Differential Equations of Motion of the Robot

The Lagrange equation in matrix form describing the motion of the robot has form (8).

$$[M(q)]_{(8 \times 8)} \ddot{\mathbf{q}}_{(8 \times 1)} + [C(q, \dot{q})\dot{q}]_{(8 \times 1)} + [G(q)]_{(8 \times 1)} + [Q]_{(8 \times 1)} = [U]_{(8 \times 1)} \quad (8)$$

In which $[M(q)]_{8 \times 8}$ is the mass matrix of the robot system, calculated through mass m_i , translation Jacobian matrix J_{Ti} , rotation Jacobian matrix ${}^{ci}J_{Ri}$ and inertia tensor ${}^{Ci}\Theta_{Ci}$ of link i (9).

$$[M(q)]_{(8 \times 8)} = \left[\sum_{i=1}^8 (J_{Ti}^T m_i J_{Ti} + {}^{ci}J_{Ri}^T {}^{Ci}\Theta_{Ci} {}^{ci}J_{Ri}) \right]_{(8 \times 8)} \quad (9)$$

$[C(q, \dot{q})\dot{q}]_{8 \times 1}$ is the vector of the generalized forces of Coriolis and centrifugal inertial forces, calculated through the partial derivative of the elements of matrix $[M(q)]$ with respect to the generalized coordinates q , and the generalized velocity \dot{q} (10).

$$[C(q, \dot{q})\dot{q}]_{(8 \times 1)} = [c_1, c_2, \dots, c_8]^T; \quad c_j = \sum_{k,l=1}^8 (k, l; j) \dot{q}_k \dot{q}_l; \quad (k, l; j) = \frac{1}{2} \left(\frac{\partial m_{kj}}{\partial q_l} + \frac{\partial m_{lj}}{\partial q_k} - \frac{\partial m_{kl}}{\partial q_j} \right) \quad (10)$$

$(k, l; j)$ is Christoffel notation with 3 indexes of the first kind; m_{kl} ($k, l = 1, \dots, 8$) are the elements of matrix $[M(q)]$.

$[G(q)]_{8 \times 1}$ is the vector of the generalized forces of the conservative forces (11).

$$[G(q)]_{(8 \times 1)} = [G_1, G_2, \dots, G_8]^T; \quad G_j = \frac{\partial \Pi}{\partial q_j} \quad (11)$$

Π is the potential energy of the system.

$[U]_{8 \times 1}$ is the vector of the generalized forces of the driving forces/torques (12).

$$[U]_{(8 \times 1)} = [U_1, U_2, \dots, U_8]^T; \quad U_i = \tau_i \quad (12)$$

τ_i is the driving force at joint i . τ_i is the force or torque depending on the prismatic joint or revolute joint.

$[Q(q)]_{8 \times 1}$ is the vector of the generalized forces of the non-conservative forces, including the friction forces at the contact point between the tool and the workpiece, the forces that separate the chip from the workpiece; the acting forces between the tool and the workpiece are called the cutting forces, collectively. In addition, there are also the friction forces at the joints, the abnormal perturbation from the environment, etc., of which the value of cutting forces and the effect of cutting forces on motion are great. Therefore, this paper limits the investigation of cutting forces and other types of force are ignored. The next subsection introduces the investigation of cutting forces in milling process with the robot.

3.2. Cutting Forces in Milling

The cutting forces are generated at the cutting point of the tool and the workpiece and this is a complex process force system. However, in general, the process force system can be represented as the force system from the workpiece acting on the robot including the force and the torque (force couple) with notation (F_c, M_c) . The notation (F_r, M_r) represents the force and torque vectors from a tool acting on the workpiece (13).

$$F_c = [F_x, F_y, F_z]^T; M_c = [M_x, M_y, M_z]^T; F_r = -F_c; M_r = -M_c \tag{13}$$

The generalized force components of the force F_c and the torque M_c are determined by the robot kinematic chain (14).

$$Q_c = J_{TE}^T F_c + J_{RE}^T M_c = [J_{TE}^T, J_{RE}^T] [F_c, M_c]^T = J_E^T R_c \tag{14}$$

r_E is the position vector of the acting point of the force F_c with respect to the base frame, ω_E is the angular velocity of the end effector, determined by the robot kinematic chain. The Jacobian matrices in (14) are calculated according to (15).

$$J_{TE} = \frac{\partial r_E}{\partial q}; J_{RE} = \frac{\partial \omega_E}{\partial \dot{q}} \tag{15}$$

The generalized force components of the force F_r and the torque M_r are determined according to the platform kinematic chain (16).

$$Q_r = J_{Tr}^T F_r + J_{Rr}^T M_r = [J_{Tr}^T, J_{Rr}^T] [F_r, M_r]^T = J_r^T R_r \tag{16}$$

r_k is the position vector of acting point of the force F_r with respect to the base frame and ω_k is the angular velocity of the workpiece determined by the platform kinematic chain. The Jacobian matrices in (16) are calculated according to (17).

$$J_{Tr} = \frac{\partial r_k}{\partial q}; J_{Rr} = \frac{\partial \omega_k}{\partial \dot{q}} \tag{17}$$

Note that Equation (17) is used for the general case; with the robot system investigated in this paper, the angular velocity ω_k is zero because the platform carrying the workpiece has only two prismatic joints, and only performs translational motion.

It is known that on the robot kinematic chain, there are only joint coordinates q_1, q_2, \dots, q_6 ; while on the platform kinematic chain, there are only joint coordinates q_7, q_8 , therefore we have (18) and (19).

$$Q_c = [Q_{c1}, Q_{c2}, \dots, Q_{c6}, 0, 0]^T \tag{18}$$

$$Q_r = [0, 0, \dots, 0, Q_{r7}, Q_{r8}]^T \tag{19}$$

So, the sum of the generalized force components of the non-conservative forces is expressed as (20).

$$[Q(q)]_{(8 \times 1)} = Q_c + Q_r = [Q_{c1}, Q_{c2}, \dots, Q_{c6}, Q_{r7}, Q_{r8}]^T \tag{20}$$

Recognizing that in order to calculate the generalized forces of cutting forces, the Jacobian matrices calculated according to (15) are quite complex, because the expressions for determining the position of cutting points and the angular velocity of the end effector are often cumbersome and complex.

The cutting forces expressing as (13) appear in milling process depending on the cutting parameters, the geometric parameters of the chip geometry, cutting conditions, etc. The cutting force components are not constants because the materials are heterogeneous, and the depth of cut, the feed rate, etc., may be changed during the machining process. The cutting forces are complex distribution forces on the

contact area between the surface of the cutting edge and the workpiece, with the resultant force placed at the center of the contact area (placed at the upper point of the cutting edge between the cutter entry and exit angles).

According to the authors of [25,26], the approximate method of calculating the milling forces in the x, y, z directions of the Z cutting teeth is represented by (21).

$$F_x = \sum_{i=1}^Z F_{x,i}; \quad F_y = \sum_{i=1}^Z F_{y,i}; \quad F_z = \sum_{i=1}^Z F_{z,i} \quad (21)$$

where $F_{x,i}, F_{y,i}, F_{z,i}$ are the cutting forces acting on the i th cutting edge in the x, y, z directions, respectively, calculated according to (22).

$$F_{f,i}(\theta_i(z)) = \int_{z_{j,1}}^{z_{j,2}} dF_{f,i}(\theta_i(z)) dz; \quad f = x, y, z \quad (22)$$

In which $z_{j,1}(\theta_i(z)), z_{j,2}(\theta_i(z))$ are upper and lower limitations of the cutting area of the i th cutting edge, in axial direction of the cutter shaft.

dF_x^i, dF_y^i, dF_z^i are the differential cutting forces acting on the infinitesimal cutting edge segment of the i th cutting edge, in x, y, z directions, respectively (23).

$$\begin{bmatrix} dF_{xi} \\ dF_{yi} \\ dF_{zi} \end{bmatrix} = \begin{bmatrix} -\cos \theta_i & -\sin \theta_i \sin \kappa & -\sin \theta_i \cos \kappa \\ \sin \theta_i & -\cos \theta_i \sin \kappa & -\cos \theta_i \cos \kappa \\ 0 & -\cos \kappa & -\sin \kappa \end{bmatrix} \begin{bmatrix} dF_{ti} \\ dF_{ri} \\ dF_{ai} \end{bmatrix} \quad (23)$$

$\theta_i(z)$ is the immersion angle for the cutting edge i at axial depth of cut z (24):

$$\theta_i(z) = \theta(t) + \theta_{pi} - \psi(z); \quad i = 1, \dots, Z; \quad \theta(t) = \frac{2\pi n}{60} t; \quad \theta_{pi} = (1 - i) \frac{2\pi}{Z} \quad (24)$$

κ is the angle between dF_{ri} and the cutter axial; Z is number of cutting edge/flute/tooth of the cutter; $\Psi(z)$ is the radial lag angle; n is spindle speed (rpm).

At instant t , at any point P of the cutting area, a segment of the i th cutting edge performs cutting, the differential cutting forces $dF_{ti}, dF_{ri}, dF_{ai}$ in tangential, radial and axial directions, respectively, acting on point P , represented in Figure 2 and determined by (25).

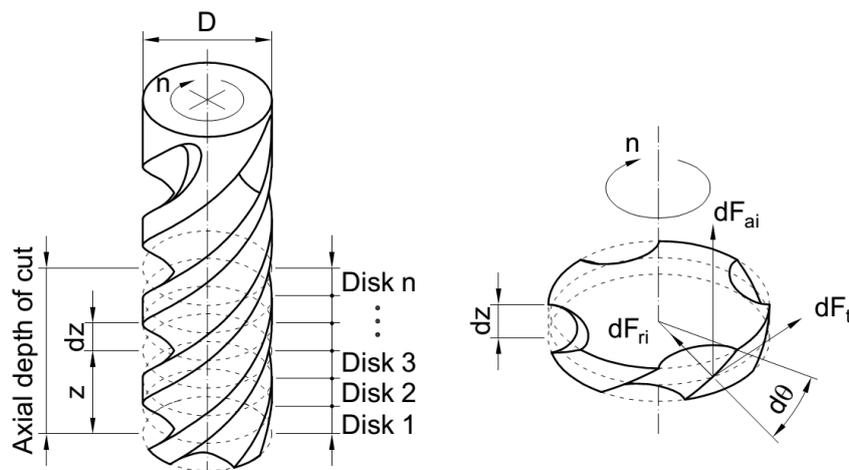


Figure 2. Representation of the cutting force model of the cutting edge i of a disk dz in milling.

$$\begin{cases} dF_{ti} = K_{tc}a_i(\theta_i, \kappa)db + K_{te}dS \\ dF_{ri} = K_{rc}a_i(\theta_i, \kappa)db + K_{re}dS \\ dF_{ai} = K_{ac}a_i(\theta_i, \kappa)dB + K_{ae}dS \end{cases} \quad (25)$$

Of which:

K_{tc}, K_{rc}, K_{ac} are tangential, radial and axial cutting force coefficients in milling.

K_{te}, K_{re}, K_{ae} are tangential, radial and axial edge force coefficients in milling.

dS is thickness of an element of the i th cutting edge.

db is length of cut in the direction of the cutting velocity vector.

$a_i(\theta_i, \kappa)$ is chip thickness that cut by the segment of the i th cutting edge (26).

$$a(\theta_i, \kappa) = S_z \sin \theta_i \sin \kappa \quad (26)$$

The force components according to (21)–(23) can be conveniently represented on the x, y, z axes of the tool frame $O_{EXEY_1EZ_1}$. Summing the cutting force system with respect to the origin of the tool frame, we obtain the torques of the cutting forces. These torques are determined based on the position of the acting point of resultant force of the cutting force system with respect to the tool frame $O_{EXEY_1EZ_1}$. The contact area and the state of interaction between the tool and the workpiece are difficult to determine accurately, therefore, the determination of the resultant force value of the cutting force system and their positions, as well as their torques, are complex and less accurate. This study does not aim to find out an accurate calculation of the cutting forces but rather to find out the way to eliminate its effects.

The next section represents the application of a dynamic model that is derived above for the Inverse Dynamics + PD controller.

4. Inverse Dynamics + PD Controller

The control of the 8 DOF robot–platform system is the coordinating movement control of the 6 DOF robot with the tool and the 2 DOF platform with the workpiece so as to ensure the tool tracking on the given form-shaping tool path in the task space. Movement of the robot system is carried out thanks to the joint driving system. The control effects are applied to the joints, so the joints are directly the control objects; the joint positions always follow to the desired positions so that the position deviations (error) are less than a specified value consistent with the required accuracy.

The Inverse Dynamics + PD controller is the common controller, popular for industrial robots, based on the rule of inverse dynamics control combined with the outer loop linear PD control.

Using the differential equations of motion of robot system (8), choosing the control rule as follows (27).

$$[U] = [U]_{(8 \times 1)} = [M(q)]_{(8 \times 8)}u_{(8 \times 1)} + [C(q, \dot{q})\dot{q}]_{(8 \times 1)} + [G(q)]_{(8 \times 1)} + [Q(q)]_{(8 \times 1)} \quad (27)$$

u is the chosen control signal that has the PD structure (28).

$$U = \ddot{q}_{d(8 \times 1)} + K_{v(8 \times 8)}\dot{e}_{(8 \times 1)} + K_{p(8 \times 8)}e_{(8 \times 1)} \quad (28)$$

In which e, \dot{e}, \ddot{e} are the vectors that represent deviations of the joint positions, velocities and accelerations at an instant machining time (29).

$$\begin{cases} e = e_{(8 \times 1)} = q_d - q = [e_1, e_2, \dots, e_8]^T \\ \dot{e} = \dot{e}_{(8 \times 1)} = \dot{q}_d - \dot{q} = [\dot{e}_1, \dot{e}_2, \dots, \dot{e}_8]^T \\ \ddot{e} = \ddot{e}_{(8 \times 1)} = \ddot{q}_d - \ddot{q} = [\ddot{e}_1, \ddot{e}_2, \dots, \ddot{e}_8]^T \end{cases} \quad (29)$$

$q_d, \dot{q}_d, \ddot{q}_d$ are the vectors of the desired joint positions, velocities, and accelerations in accordance with the requirements of machining engineering, which are the set trajectories (7) calculated by the kinematic problem.

q, \dot{q}, \ddot{q} are the vectors of the actual joint positions, velocities, and accelerations.

$K_P(8 \times 8), K_V(8 \times 8)$ are the diagonal matrix of the proportional gain coefficients, the derivative of each separate joint corresponding to the controller, respectively, (30) and (31).

$$K_P = \text{diag}\{k_{p1}, k_{p2}, \dots, k_{p8}\}; k_{pi} > 0 \tag{30}$$

$$K_V = \text{diag}\{k_{v1}, k_{v2}, \dots, k_{v8}\}; k_{vi} > 0 \tag{31}$$

Substituting (27) and (28) into (8) we find out the erroneous differential equation of the control system (32),

$$[\ddot{e}]_{(8 \times 1)} + [K_V]_{(8 \times 8)}[\dot{e}]_{(8 \times 1)} + [K_P]_{(8 \times 8)}[e]_{(8 \times 1)} = 0 \tag{32}$$

or write in a separate form of independent equations for the joints (33).

$$\ddot{e}_i + K_{V_i}\dot{e}_i + K_{P_i}e_i = 0; i = 1, \dots, 8 \tag{33}$$

The Inverse Dynamics + PD controller model is demonstrated in Figure 3. The simulation results are shown in Section 6.

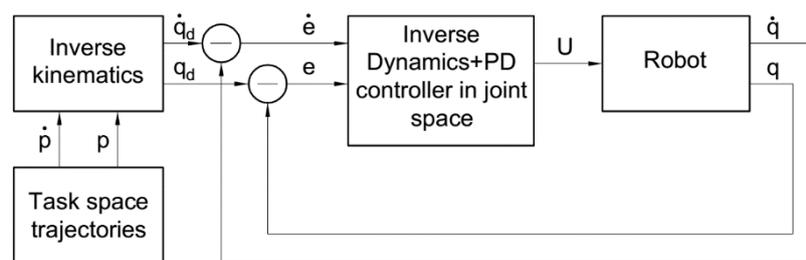


Figure 3. Scheme of the inverse dynamics method combined with the law of derivative proportional control (Inverse Dynamics + PD) controller in joint space

5. The Controller Based on Fuzzy Logic

The literature on fuzzy control is very diversified, however for easy understanding this article briefly introduces fuzzy control. Some knowledge that is relevant and used in the paper is presented in Appendix A. Based on the contents presented in the Appendix A, it is easy to find out relevant issues when looking up references. Subsequent contents of the article with references (if any) in the Appendix A may help to clarify the fuzzy control base that is briefly presented below.

This section presents the fuzzy controller design for a milling machining robot. The article uses the Mamdani fuzzy controller and selects the rule of Max-Min composition, the method of central defuzzification and triangular membership functions.

The following shows the steps to design a fuzzy controller and propose fuzzy rules for the robot controller (See also Appendix B).

5.1. Select the Input, Output Signals and Physical Value Domains

5.1.1. Select Input and Output Signals

Inputs are joint position errors $e(t)$ and joint velocity errors $\dot{e}(t)$ over time of the robot system. Outputs are the adjustment amounts of the joint driving forces/torques $u(t)$ at each machining instant to ensure that the errors $e(t), \dot{e}(t)$ are as small as the requirement of machining accuracy.

u is the vector of the adjustment amounts of joint forces/torques of the robot and platform, which is calculated according to Formula (34),

$$u = \tau - \tau^d = [u_1, u_2, \dots, u_8]^T \tag{34}$$

where τ^d is the vector of desired driving forces/torques at the joints in accordance with theoretical computations and requirements of machining engineering, then the joint position error $e(t) \rightarrow 0$ and the joint velocity error $\dot{e}(t) \rightarrow 0$.

τ is the vector of the actual forces/torques at the joints in accordance with the requirements of machining engineering so that the joint position errors $e(t)$ and the joint velocity errors $\dot{e}(t)$ approach the zero asymptotic point.

u_1, u_2, \dots, u_8 are the adjustment amounts of forces/torques at each joint 1, 2, ..., 6, B and D of the robot system, respectively.

At each machining instant corresponding to a pair of input signal values, which are the joint position error $e(t)$ and the joint velocity error $\dot{e}(t)$, the controller calculates the adjustment amount of the joint forces/torques $u(t)$, and gives a signal $u(t)$ to adjust the errors $e(t)$, $\dot{e}(t)$ approaching and converging at the zero asymptotic point:

5.1.2. Determine the Physical Value Domains for Control Signals

Depending on the parameters of the machining process, kinematics, dynamics and the required force/torque for the robot to perform machining motion according to the machining requirements, it will estimate the physical value domain for position errors, velocity errors, and adjustment amount of the force/torque of the i th joint ($i = 1, \dots, 8$) of the fuzzy controller which is shown in (35).

$$\begin{cases} e_i = [e_{imin}, e_{imax}]^T (\text{rad}, \text{m}) \\ \dot{e}_i = [\dot{e}_{imin}, \dot{e}_{imax}]^T (\text{rad.s}^{-1}, \text{m.s}^{-1}) ; \quad i = 1, \dots, 8 \\ u_i = [u_{imin}, u_{imax}]^T (\text{Nm}, \text{N}) \end{cases} \tag{35}$$

5.2. Fuzzification of Input and Output Signal Datas

5.2.1. Build Linguistic Variables

Language variables are key elements of fuzzy control rule. Each linguistic variable consists of a linguistic value domain and physical value domain. The linguistic value domain is a set of linguistic values. The linguistic value is set up by a fuzzy set.

The number of selected linguistic values must be appropriate, not complicated and fully represent the characteristics of the control system.

Fuzzification of input and output signal data includes the name of the errors of position and velocity, and the amount of force/torque adjustment as y_i ($y_i = e_i, \dot{e}_i, u_i; i = 1, \dots, 8$), $y_{imin} \leq y_i \leq y_{imax}$. The physical value domain X of $y_i \in [y_{imin}, y_{imax}]$ is divided into five subdomains X_j ($j = 1, \dots, 5$), the subdomains overlap and the subdomain X_j is set up by fuzzy set F_j . The fuzzy set F_j is represented by linguistic values as in Table 2.

Table 2. Description of physical domains, fuzzy set and linguistic values of the inputs and outputs.

Serial No.	Physical Domain X_j	Fuzzy Set F_j	Name of Linguistic Value	Notation of Linguistic Value
1	X_1	F_1	Big Negative	NB
2	X_2	F_2	Small Negative	NS
3	X_3	F_3	Zero	Z
4	X_4	F_4	Small Positive	PS
5	X_5	F_5	Big Positive	PB

5.2.2. Set up Membership Function of the Errors of Position, Velocity and the Amount of Joint Force/Torque Adjustments

It is necessary to build a function of linear form (right trapezoid, triangle, left trapezoid) of the position error e_i , the velocity error \dot{e}_i , and the amount of force/torque adjustment of joint i shown in Figure 4.

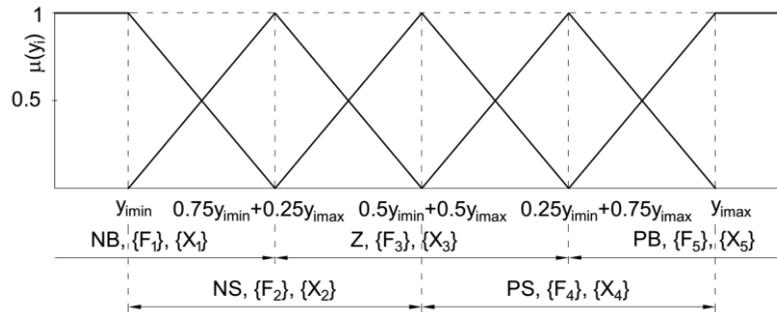


Figure 4. The membership function of position error, velocity error and adjustment amount of driving torque of joint i : y_i ($y_i = e_i, \dot{e}_i, u_i; i = 1, \dots, 8$).

The membership function of y_i ($y_i = e_i, \dot{e}_i, u_i; i = 1, \dots, 8$) in the NB fuzzy set (Figure 4) is calculated according to (36).

$$\mu(y_i) = \begin{cases} 1; & y_i \leq y_{imin} \\ \frac{y_i - 0.75y_{imin} - 0.25y_{imax}}{0.25y_{imin} - 0.25y_{imax}}; & y_{imin} < y_i \leq 0.75y_{imin} + 0.25y_{imax} \end{cases} \quad (36)$$

The membership function of y_i ($y_i = e_i, \dot{e}_i, u_i; i = 1, \dots, 8$) in the NS fuzzy set (Figure 4) is calculated according to (37).

$$\mu(e_i) = \begin{cases} \frac{y_i - y_{imin}}{-0.25y_{imin} + 0.25y_{imax}}; & y_{imin} \leq y_i \leq 0.75y_{imin} + 0.25y_{imax} \\ \frac{y_i - 0.5y_{imin} - 0.5y_{imax}}{0.25y_{imin} - 0.25y_{imax}}; & 0.75y_{imin} + 0.25y_{imax} \leq y_i \leq 0.5y_{imin} + 0.5y_{imax} \end{cases} \quad (37)$$

The membership function of y_i ($y_i = e_i, \dot{e}_i, u_i; i = 1, \dots, 8$) in the Z fuzzy set (Figure 4) is calculated according to (38).

$$\mu(y_i) = \begin{cases} \frac{y_i - 0.75y_{imin} - 0.25y_{imax}}{-0.25y_{imin} + 0.25y_{imax}}; & 0.75y_{imin} + 0.25y_{imax} \leq y_i \leq 0.5y_{imin} + 0.5y_{imax} \\ \frac{y_i - 0.25y_{imin} - 0.75y_{imax}}{0.25y_{imin} - 0.25y_{imax}}; & 0.5y_{imin} + 0.5y_{imax} \leq y_i \leq 0.25y_{imin} + 0.75y_{imax} \end{cases} \quad (38)$$

The membership function of y_i ($y_i = e_i, \dot{e}_i, u_i; i = 1, \dots, 8$) in the PS fuzzy set (Figure 4) is calculated according to (39).

$$\mu(y_i) = \begin{cases} \frac{y_i - 0.5y_{imin} - 0.5y_{imax}}{-0.25y_{imin} + 0.25y_{imax}}; & 0.5y_{imin} + 0.5y_{imax} \leq y_i \leq 0.25y_{imin} + 0.75y_{imax} \\ \frac{y_i - y_{imax}}{0.25y_{imin} - 0.25y_{imax}}; & 0.25y_{imin} + 0.75y_{imax} \leq y_i \leq y_{imax} \end{cases} \quad (39)$$

The membership function of y_i ($y_i = e_i, \dot{e}_i, u_i; i = 1, \dots, 8$) in the PB fuzzy set (Figure 4) is calculated according to (40).

$$\mu(y_i) = \begin{cases} \frac{y_i - 0.25y_{imin} - 0.75y_{imax}}{-0.25y_{imin} + 0.25y_{imax}}; & 0.25y_{imin} + 0.75y_{imax} \leq y_i < y_{imax} \\ 1; & y_i \geq y_{imax} \end{cases} \quad (40)$$

From the above, we realize that input and output vector signals are position errors $e(t)$, velocity errors $\dot{e}(t)$, and amount of torque adjustment $u(t)$ of joints of the robot system within the real (explicit) physical value domain identifying them as different, but with the same linguistic value name.

5.3. Set up the Control Laws

Based on the input variables that are joint position errors $e(t)$ and velocity errors $\dot{e}(t)$, we develop a fuzzy rule table to calculate the adjustment amount of joint force/torque $u(t)$. At the machining time, the input variables are signals of position errors $e(t)$, velocity errors $\dot{e}(t)$ of joints within the physical domain $\{X_1, \dots, X_5\}$ of a fuzzy set, represented by a linguistic value in the domain $\{NB, NS, Z, PS, PB\}$ for each variable. The controller will generate a signal to control the amount of force/torque adjustment $u(t)$ within the physical domain $\{X_1, \dots, X_5\}$, represented by a linguistic value in the domain $\{NB, NS, Z, PS, PB\}$ and this $u(t)$ signal corrects the input values of coordinate errors e_i , velocity errors \dot{e}_i of joints approaching the physical value domain described by the linguistic value Z .

So, from the two input variables $e(t)$, $\dot{e}(t)$, along with 5 linguistic values of each variable, a rule system will be set up inclusive of $5^2 = 25$ control rules for the combination of variables $e(t)$, $\dot{e}(t)$, as Table 3.

Table 3. Fuzzy rules.

Amount $u(t)$	Adjustment	Velocity Error $\dot{e}(t)$				
		NB	NS	Z	PS	PB
Position error $e(t)$	NB	PB	PB	PB	PS	Z
	NS	PB	PS	PS	Z	NS
	Z	PB	PS	Z	NS	NB
	PS	PS	Z	NS	NS	NB
	PB	Z	NS	NB	NB	NB

The formula to set the fuzzy compositional rule system for the fuzzy controller (41).

$$R : \bigcup_{i=1}^n R_i \tag{41}$$

$$R_i : [(e(t) \cup \dot{e}(t)) \rightarrow u(t)], i = 1, \dots, 25$$

From Table 3 and Formula (41), there is the fuzzy compositional rule system (42) including fuzzy conditional statements:

$$\left\{ \begin{array}{l} \text{If } e(t) \text{ is NB and } \dot{e}(t) \text{ is NB then } u(t) \text{ is PB or} \\ \text{If } e(t) \text{ is NB and } \dot{e}(t) \text{ is NS then } u(t) \text{ is PB or} \\ \vdots \\ \text{If } e(t) \text{ is PB and } \dot{e}(t) \text{ is PB then } u(t) \text{ is NB} \end{array} \right. \tag{42}$$

5.4. Set up the Compositional Rule of Inference for the Fuzzy Controller

At each machining time, the input control signals of position errors $e(t)$, velocity errors $\dot{e}(t)$ are always within two physical value domains, set up by two fuzzy sets, represented by two linguistic values of the domain $\{NB, NS, Z, PS, PB\}$. Therefore, it is necessary to determine the Compositional Rule of Inference. We find in turn the membership function value of output signals which is the amount of force/torque adjustment $u(t)$ of each Compositional Rule R_k ($k = 1, \dots, 25$) and the rule system for the input control signals of coordinate errors $e(t)$, velocity errors $\dot{e}(t)$ of the controller. Select the method to execute the Compositional Rule of Inference according to Mamdani composition MAX-MIN Rule (43) and (44).

$$F'_u = \bigcup_{i=1}^{25} F'_{u_i}; \mu_{F'_u}(y) = \bigvee_{i=1}^{25} \mu_{F'_{u_i}}; F'_i = (F'_e, F'_{\dot{e}}) \circ R_i; (de = \dot{e}) \tag{43}$$

$$\mu_{F'_{u_i}}(y) = \alpha_i \wedge \mu_{F_{u_i}}(y) ; \alpha_i = \min \left\{ \max_y (\mu_{F'_e}(y) \wedge \mu_{F_{ei}}(y)), \max_y (\mu_{F'_{de}}(y) \wedge \mu_{F_{dei}}(y)) \right\} \quad (44)$$

Use Rule (43) and (44), in turn, to determine the membership function value for each Compositional Rule and total Compositional Rule system. The output result of each Compositional Rule is the membership function value corresponding to the fuzzy set F' (note that each membership function value corresponds to a fuzzy set F'_i and the membership function value of the whole system corresponds to the general fuzzy set F'), the fuzzy set F' is in 5 fuzzy sets represented by 5 linguistic values of the domain {NB, NS, Z, PS, PB}.

5.5. Defuzzification

The defuzzification is to determine the explicit (crisp) value y' of the output (the amount of force/torque adjustment $u(t)$ of joint) from the membership function value of the fuzzy set F' . Use the center of area method (COA) for defuzzification to the fuzzy controller of the robot system when surface shaping machining. This method is easy to apply, involving all output fuzzy sets of the Compositional Rule of Inference. The results of this method are highly accurate (45).

$$y' = \frac{\int y \mu_{F'}(y) dy}{\int \mu_{F'}(y) dy} \quad (45)$$

where y' is the actual output value taken in accordance with the abscissa of the domain central point enclosed by membership function constituting (line) with the horizontal axis and S' is the domain defining the fuzzy set F' . Figure 5 shows the principal structure of the fuzzy controller.

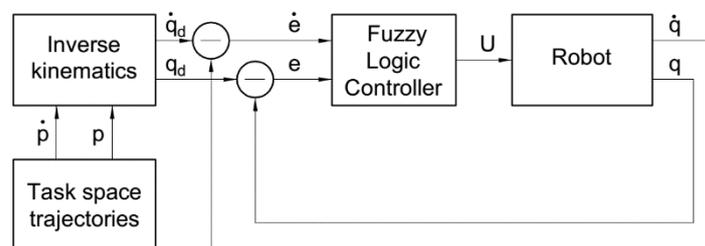


Figure 5. Scheme of fuzzy logic control for the robot–platform system in machining.

Applying the fuzzy logic as presented, this paper proposes to set up a fuzzy controller for a robot in a milling process based on the incomplete dynamic model, which means ignorance of the cutting force determination. The control block of the robot controller has a double structure consisting of an outer loop that is a fuzzy control block to determine the adjustment value based on the erroneous information of joint positions and velocities; and the inner loop is the inverse dynamics calculation block from the robot dynamic model, in which there is no cutting force component. Figure 6 shows a SIMULINK model representing the fuzzy controller for the robot system. Figure 7 shows a double-loop structure of the control block in the fuzzy controller, enabling the calculation of driving force.

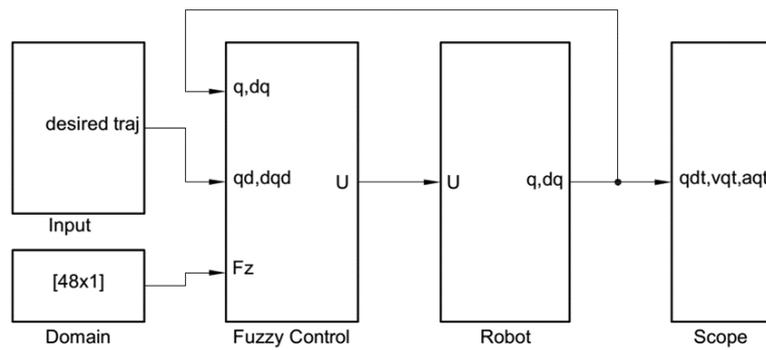


Figure 6. General block diagram of fuzzy control.

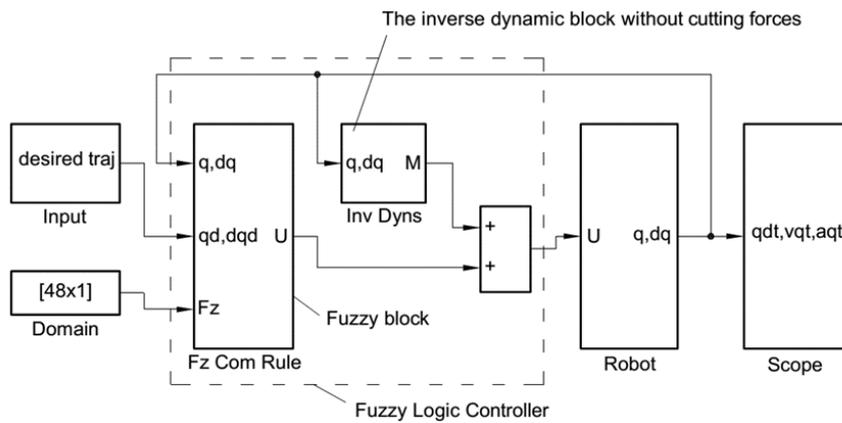


Figure 7. Block diagram of dual-loop fuzzy control.

6. Simulation Results

In order to verify the quality of the designed controller, this section presents the application of a robot carrying out the down milling process of a mold prototype of a hydraulic pump’s central housing, which is made of Ti6Al4V titanium alloy material. The machining trajectory is the path L_i , which represents the milling path of the cutter head moving from outside to inside to complete the machining surface (Figure 8). The cutter is an end mill cutter that has four cutting edges/flutes, with a 20 mm diameter (Figure 9).

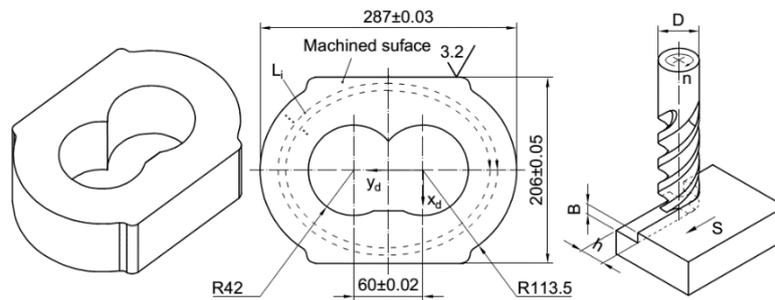


Figure 8. The workpiece and the tool path.

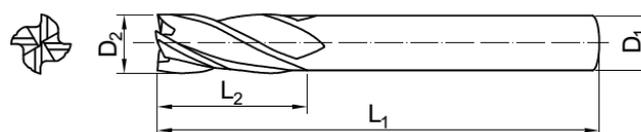


Figure 9. The end mill cutter.

The parameters of geometric, kinematic and dynamic of the robot, platform and workpiece are determined and created as the input parameter file.

The cutter parameters and the milling process parameters are shown in Table 4, where Z is the number of cutting edges (flutes) of the cutter, v_c is cutting speed, v_r is the relative velocity between the tool and the workpiece, S_z is the feed rate on a cutting edge, B is milling width ($B = 2.2$ mm) and h is milling depth ($h = 15$ mm).

Table 4. The cutter parameters and the milling process parameters.

Cutter's Material	D_1 (mm)	D_2 (mm)	L_1 (mm)	L_2 (mm)	Z (tooth)	v_c (m/min)	v_r (mm/s)	S_z (mm/tooth)	h_0 (mm)	Cooling Liquor
Carbide	20	20	125	50	4	31.3	3.3	0.1	2.2	Emunxi

Assume that with the two methods of calculation and selections, the cutting force coefficients in (25) are shown in Table 5 and the results of cutting force calculation according to the empirical formula are shown in Figure 10. In particular, Figure 10 (a)÷(d), respectively, represent cutting forces in the x , y , z directions and the torque in z direction of the tool frame. The small torque components in x and y directions should be ignored.

Table 5. Cutting force coefficients in milling.

Case	K_{tc} (N/mm ²)	K_{rc} (N/mm ²)	K_{ac} (N/mm ²)	K_{te} (N/mm)	K_{re} (N/mm)	K_{ae} (N/mm)
1	1825	770	735	29.7	55.7	1.8
2	1963	646	778	29.7	55.7	1.8

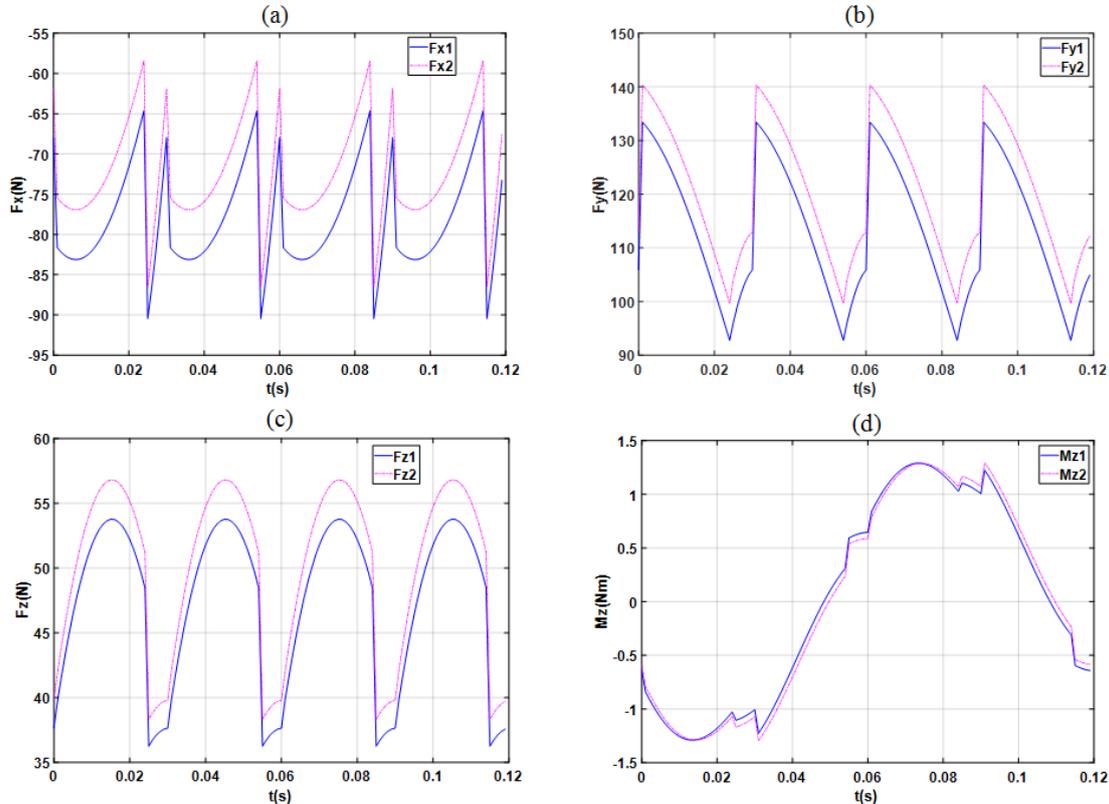


Figure 10. The cutting forces corresponding to the two cases of calculation and cutting force coefficient selection. (a) F_x -(N);(b)- F_y (N); (c) F_z -(N);(d)- M_z (Nm).

6.1. Simulation of Inverse Dynamics + PD Controller

The Inverse Dynamics + PD controller is designed as shown in Section 4. Here, the coefficient matrices of proportional gain K_P and derivative K_V are selected in accordance with (46) and (47).

$$K_P = \text{diag}\{22500, 22500, 22500, 22500, 16900, 16900, 16900, 16900\} \tag{46}$$

$$K_V = \text{diag}\{300, 300, 300, 300, 260, 260, 260, 260\} \tag{47}$$

In order to illustrate this, suppose that in the output block (robot block) of the Inverse Dynamics + PD controller in Figure 3, the cutting forces are precisely determined in accordance with the cutting force coefficients in case 1 of Table 5. The simulations are carried out for two cases where the cutting force in the input control block is calculated according to the corresponding coefficients in the two cases of Table 5, respectively. For more convenience, the simulation results that correspond to the above two cases are called PD1 and PD2.

The simulation results are shown in Figure 11. In particular, Figure 11a represents the position error of first joint in comparison with the desired trajectory. Similarly, Figure 11b–h represent the position errors of the second, third,..., eighth joints respectively. The graphs that represent the errors of joint positions show the different errors corresponding to each joint. The errors of the joints that are near the end effector are greater than that of the far ones. Summarizing the joint position errors, we get the errors of the operational point of the cutting point of the tool with respect to the toolpath. Figure 12 shows the error of trajectory of the cutting point of the tool in comparison with the set trajectory as the toolpath. We realize that the trajectory error of PD1 is smaller than that of PD2 because PD1 is corresponding to the assumption of calculating cutting forces precisely.

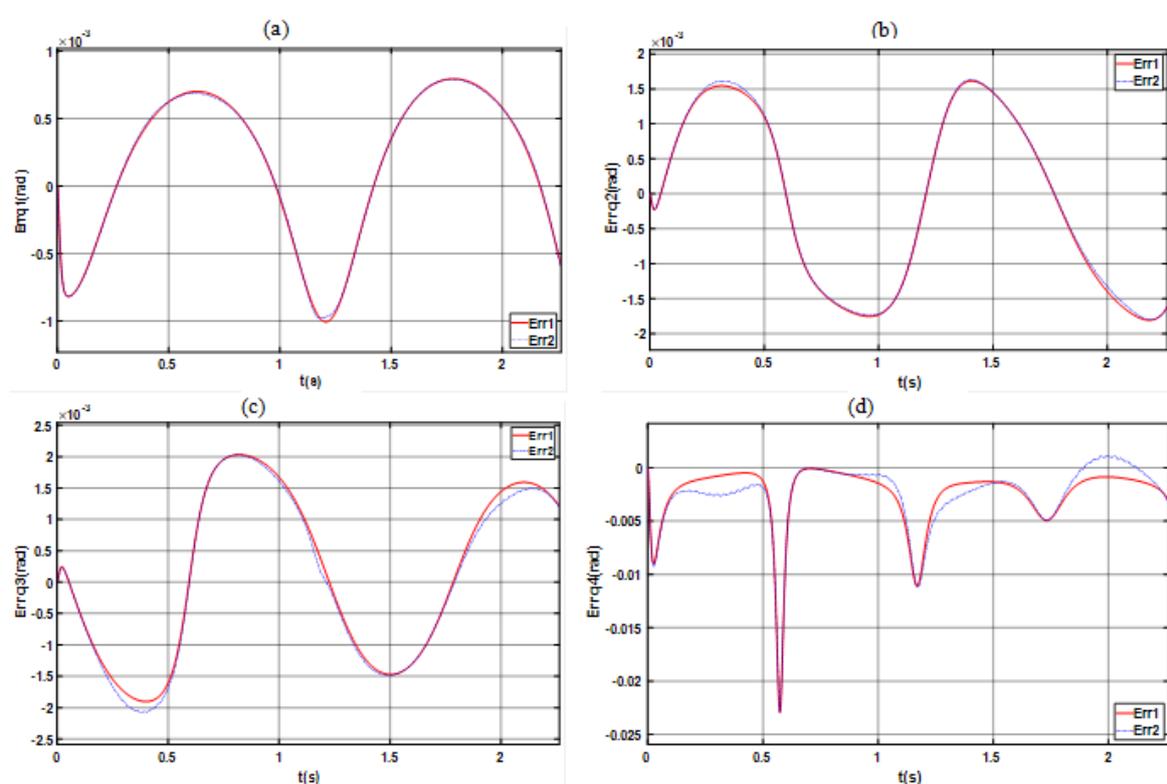


Figure 11. Cont.

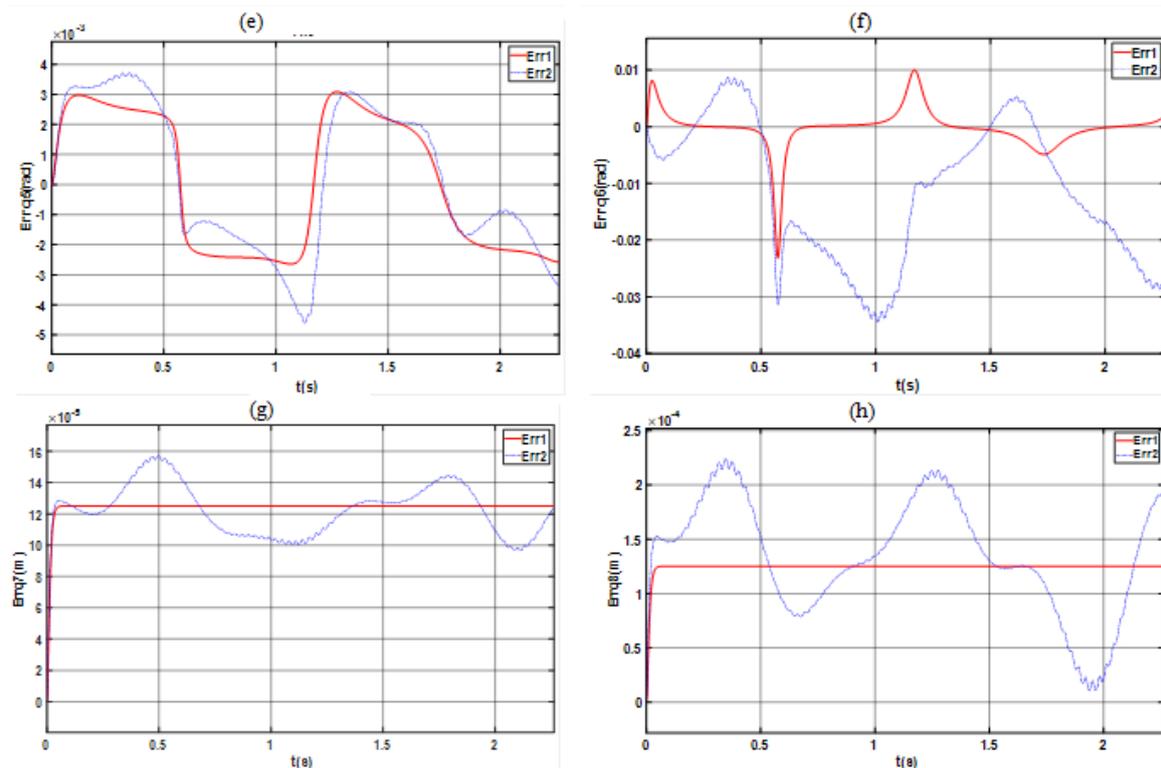


Figure 11. The errors of the joint positions of the PD1, PD2 controllers.

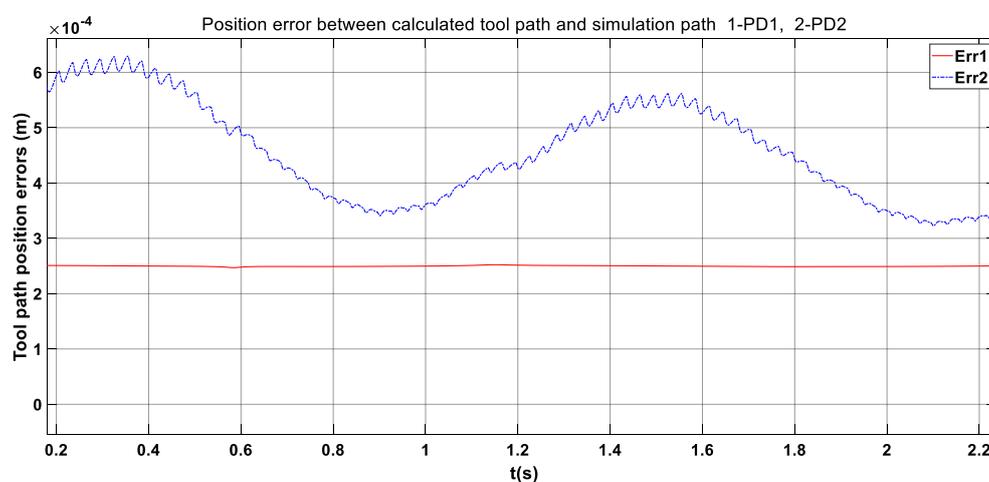


Figure 12. The errors of the cutting trajectory with respect to the set one are the toolpaths of the PD1, PD2 controllers.

Thus, with the analysis and the choice of different cutting force coefficients, the control results will be different and affect the machining precision of the robot significantly.

6.2. Simulation of the Fuzzy Controller

Depending on the given data of the kinematic, dynamic, etc., parameters, the physical value domain of input and output terms of the fuzzy controller are estimated as shown in Table 6.

Table 6. The physical value domain of input and output quantities of the fuzzy controller.

Link	e_i	\dot{e}_i	u_i
1	[-0.001,0.001] (rad)	[-0.5,0.5] (rad/s)	[-18,000,18,000] (Nm)
2	[-0.001,0.001] (rad)	[-2.4,2.4] (rad/s)	[-44,000,44,000] (Nm)
3	[-0.006,0.006] (rad)	[-2.5,2.5] (rad/s)	[-42,000,42,000] (Nm)
4	[-0.009,0.009] (rad)	[-12.0,12.0] (rad/s)	[-18,000,18,000] (Nm)
5	[-0.008,0.008] (rad)	[-7.0,7.0] (rad/s)	[-9800,9800] (Nm)
6	[-0.008,0.008] (rad)	[-10.0,10.0] (rad/s)	[-5200,5200] (Nm)
7	[-0.001,0.001] (m)	[-0.2,0.2] (m/s)	[-4700,4700] (N)
8	[-0.001,0.001] (m)	[-0.5,0.5] (m/s)	[-3100,3100] (N)

Figure 13 is the graph representing the comparison of joint position errors of the robot between the fuzzy controller and the two cases of the Inverse Dynamics + PD controller.

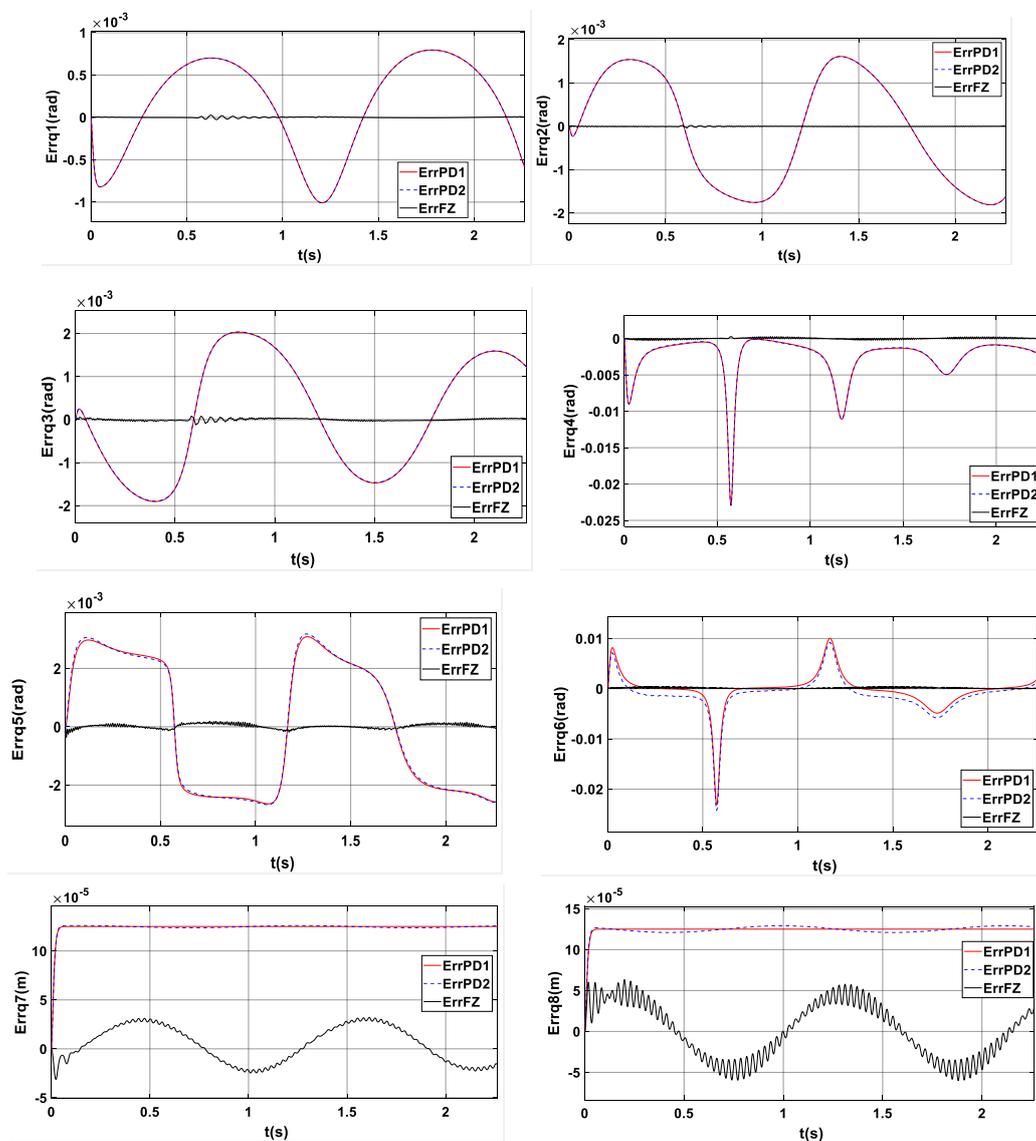


Figure 13. The joint position errors of the PD1, PD2, fuzzy controllers.

Figure 14 shows the trajectory error of the cutting point of the tool in the operational frame $O_d x_d y_d z_d$, with the lines representing the error of the fuzzy controller, and the two cases of Inverse Dynamics + PD controller.

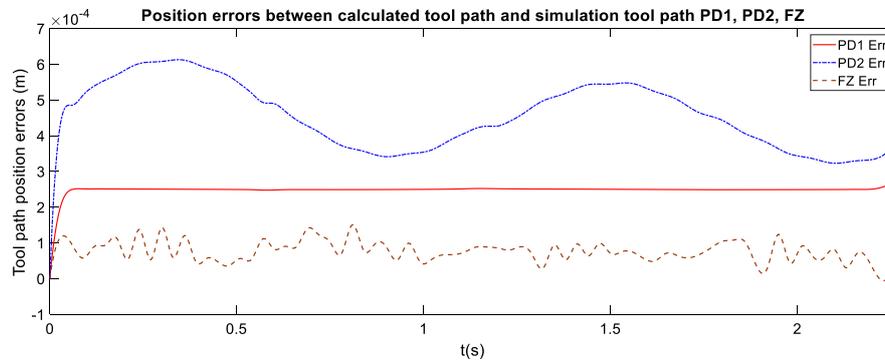


Figure 14. The errors of the cutting trajectory with respect to the set one are the toolpaths of the PD1, PD2, FZ controllers.

One can see that the joint position errors of the fuzzy controller are smaller than that of the PD1 and PD2 controllers.

Figure 15a shows the trajectory of the cutting point of the tool in the operational frame. Due to small errors, the desired and actual trajectories of the fuzzy controller and Inverse Dynamics + PD controller are approximately coinciding.

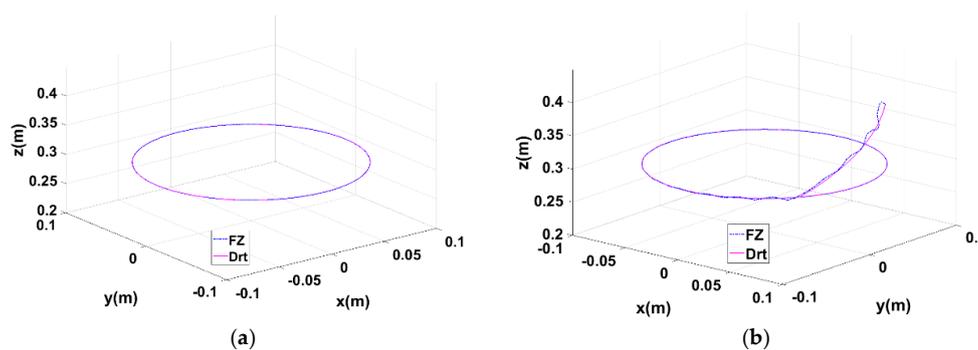


Figure 15. The trajectory of the cutting point with respect to the operational frame. (a)- when there is no initial deviation; (b)- there is initial deviation.

To verify the adjustment capability of the fuzzy controller, the simulation of the fuzzy controller is performed by the initial value of the trajectory as a deviation of the desired values.

Figure 15b shows the asymptotical and convergent process of the trajectory of the cutting point from the initial deviation to coincide with the toolpath (desired trajectory) in the milling process.

Figure 16 shows the simulation result of the fuzzy controller with initial trajectory error, time history of the joint positions, with the splines that represent the desired, actual trajectories and errors.

From the results, it can be seen that the fuzzy controller is able to control the system to the required state when there are perturbations.

In general, the simulated results show that with the assumption of ideal computed cutting forces, the Inverse Dynamics + PD explicit controller and fuzzy controller give accurate control results.

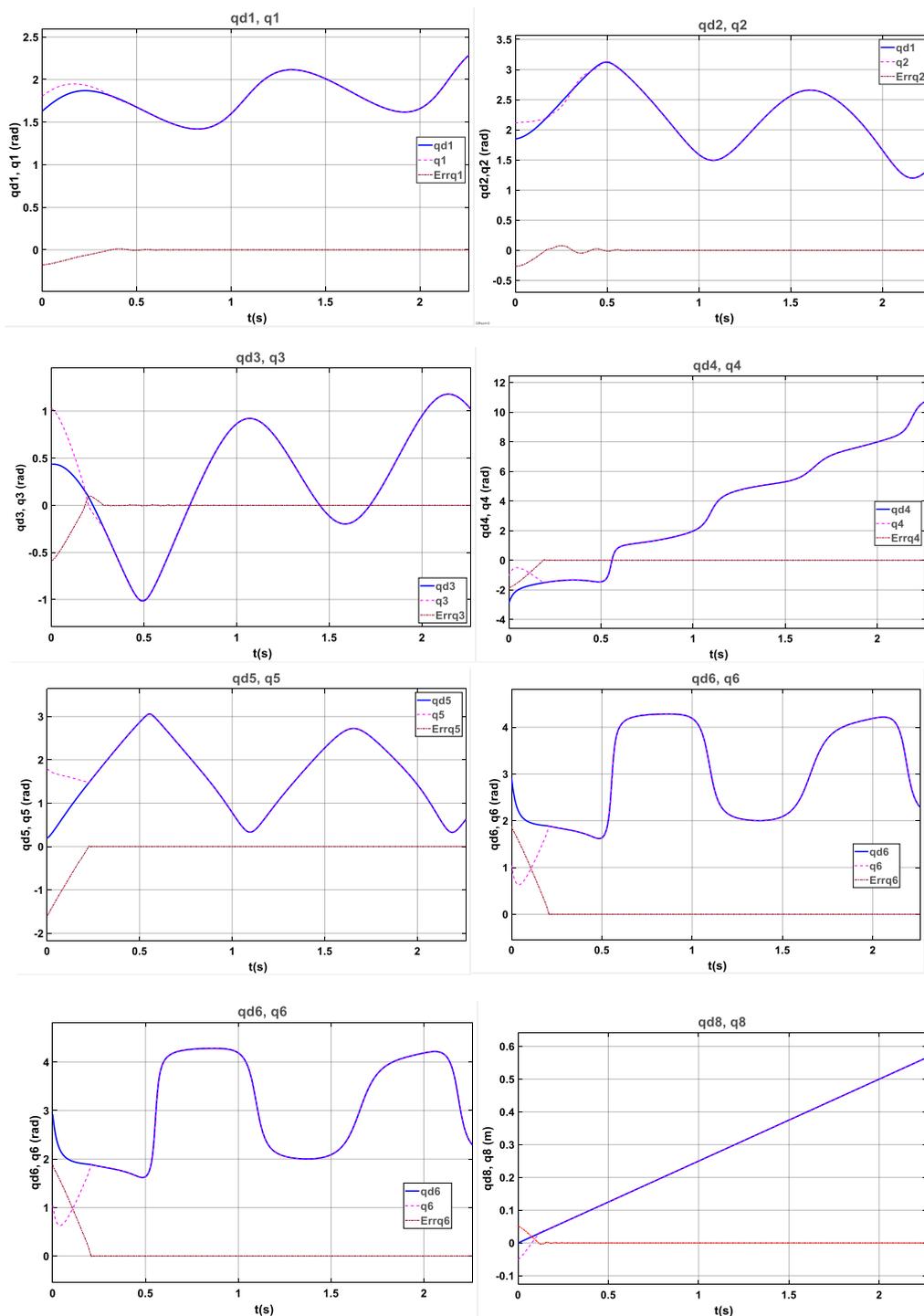


Figure 16. The simulation result of the fuzzy controller with initial trajectory error.

6.3. Stability Analysis

As mentioned, the designed fuzzy controller is a Mamdani type fuzzy controller, in which the development of fuzzy rule is based on the rules of the PD controller. In general, such a controller has good stability except perturbations.

The input signals of the controller include the joint position errors $e(t)$ and the joint velocity errors $\dot{e}(t)$, and the output signal of the controller is the adjustment amounts of the joint driving forces/torques $u(t)$ at each machining instant. A good fuzzy control rule system including the above-mentioned rules

will ensure that the calculation of the output value is consistent with the change of input signal and there is less possibility of overshoot. The numerical simulation results show this fact.

On the other hand, the input signal that can greatly change the output quantity is the physical value domain of driving force. Numerical experiments show that when the physical value domain of the driving force is selected at a smaller level than the value to be reached, the control simulation results are not good, with large errors, which are understandable. When the domain of physical values is expanded, the errors decrease, and when the physical value domain is large enough to cover all the necessary values of the driving force, the control accuracy will be good, as shown by the simulation results. The next question will be whether increasing the input signal by expanding the domain of physical values without control, i.e., when the value domain is selected as too large, the value of the output driving force may greatly increase? The fuzzy Inference and Defuzzification show that the output physical value of a variable depends on the input physical value domain and the membership function value of such variable. As the physical value domain increases, the output value increases but non-linearly. Numerical experiments in the direction of gradually expanding the domain of input physical values so that the physical value of the output $u(t)$ increases gradually to the appropriate value, we realize: The error of $e(t)$, $\dot{e}(t)$ reduction; the membership value of linguistic variables corresponding to the large value of the output $u(t)$ decreases; membership values of linguistic variables with small values increase. In the fuzzy set [NB, NS, Z, PS, PB], the membership value of $u(t)$ corresponding to Z is fast progressing to 1, other values decrease and even the NB, PB values almost go to 0 in the first steps. Based on this and based on the fuzzy inference formula, it is easy to see that the growth rate of the physical value of the output $u(t)$ decreases and the slope of the graph decreases. The slope of the graph gradually reaches 0, the graph is horizontal, then $u(t)$ does not increase. Above that limit, if the physical value domain of the input $u(t)$ continues to rise too high, the physical output value $u(t)$ does not continue to increase, but the controller error will increase again.

Figure 17 shows the membership function values corresponding to [NB, NS, Z, PS, PB] of the fuzzy set $u(t)$, with the physical value domain of $u(t)$ selected as in Table 6. The membership function value corresponding to NB, PB is almost zero, with Z close to 1. The membership function values corresponding to NS and PS of the physical value $u(t)$ are opposite the sign and adjust the physical output value $u(t)$.

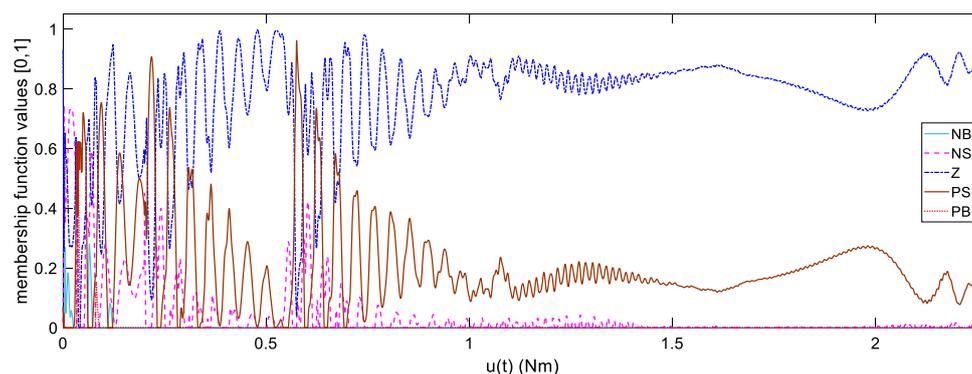


Figure 17. Membership function value of the fuzzy set $u(t)$ corresponding to the physical value domain of $u(t)$ is selected as in Table 6.

An increase of the physical value domain of $u(t)$ by 3 times obtains the results shown in Figure 18. Membership function values corresponding to NS, PS are about 3 times smaller than the values shown in Figure 17.

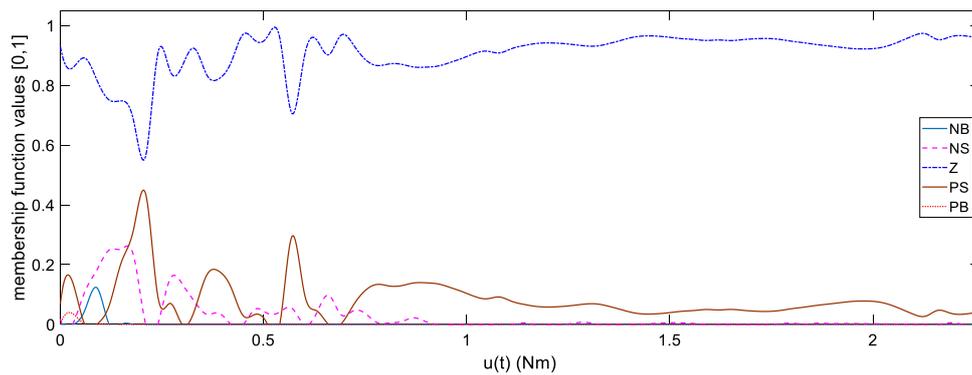


Figure 18. Membership function value of the fuzzy set $u(t)$ corresponding to the physical value domain of $u(t)$ is chosen to be 3 times larger than the value in Table 6.

Therefore, thanks to the integration of expert knowledge, the fuzzy rules are chosen, then there will be a reasonable upper bound value for the physical value domain of control output signal. At that time, it is possible to arbitrarily select the input physical value domain to get good control results without fear of instability because there is no excessive increase of driving force. Numerical experiments are carried out when control accuracy is almost the best. In this case, when expanding the physical value domain of the driving force by 5 times, the result shows that there are some early points of the control process, when the system is in the start-up phase, that the driving force increases by about (35%÷45%). However, with the entire control process on the operating trajectory, the driving force has no significant change and control accuracy is guaranteed. Thus, with the experiment mentioned above, it is feasible to select a value domain smaller than the upper bound limit.

In addition, through the mentioned numerical simulation results, the controller can return the system to trajectory when there are errors by disturbance.

Compare the fuzzy controller (FZ) with a fuzzy controller integrated with PID (FZ + PID) shown in Figures 19 and 20.

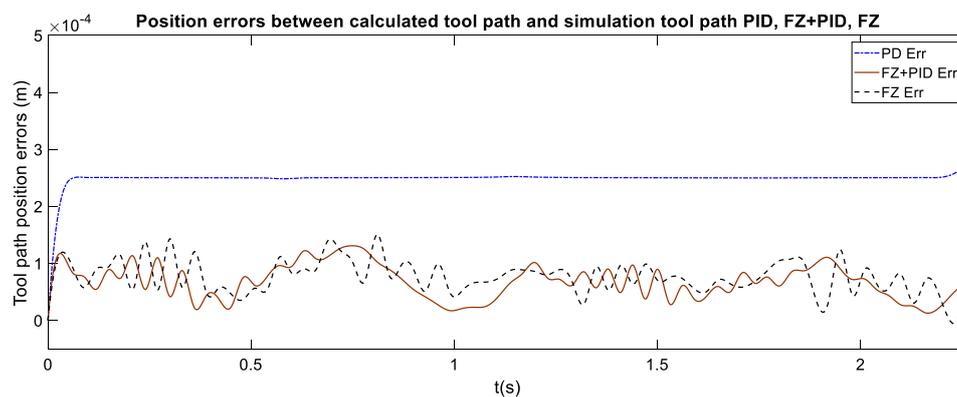


Figure 19. Tool path position errors between desired value and simulation result in normal cutting force mode.

The result shows that the accuracies of Fuzzy and Fuzzy + PID controllers are almost the same. On the other hand, with noise increasing the external load (cutting force) within 50%, the accuracy is still guaranteed and the errors are negligible. Figures 19 and 20 also show the simulation results of the Inverse Dynamics + PD controller (no perturbation, Figure 19), and Inverse Dynamics + PID (with perturbation, Figure 20). The adjustment capability of the Inverse Dynamics + PID controller is found to be slower.

Thus, among many methods of ensuring the stability and accuracy of the fuzzy controller, it is possible to say:

- the fuzzy controller has an integrated PID control unit,
- the fuzzy controller without integrated PID unit can still be stable and ensure the accuracy, by choosing a good fuzzy rule and physical value domain of appropriate input and output variables.

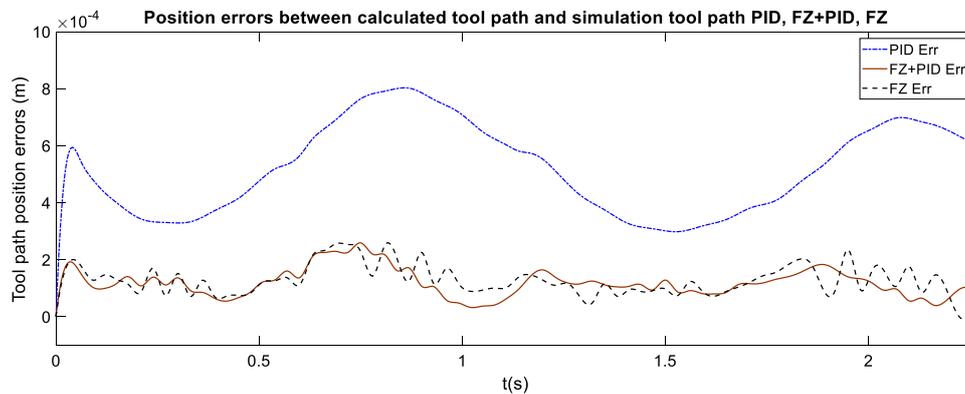


Figure 20. Tool path position errors between desired value and simulation result in cutting force mode increases by 50% due to noise.

7. Conclusions

The results show that the controller based on fuzzy logic can meet the requirements of accuracy and reliability. Furthermore, by neglecting the calculations of the cutting forces, it is more convenient to deal with only linear algebraic calculations in fuzzy control rules, which in turn help to greatly reduce the time and computation load and increase the efficiency of the robot control system.

From the simulation results, the fuzzy controller may be even more accurate than the explicit controller, although this is only relative because the presented controllers have not been optimized. The parameters of the above controllers have been selected “quite randomly” in simulation cases.

Theoretically, the explicit controller is accurate when the dynamic model is accurate and contains no perturbations. However, in practice, it is difficult to determine precisely all the system parameters. This leads to errors in the calculation of the control signals.

Meanwhile, the fuzzy controller computes the control law based on the input and output signal errors and the fuzzy rule. Thus, the calculation results can be flexibly adjusted based on the input and output signal errors. This explains the possibility that the fuzzy controller may be more accurate than the explicit controller. In addition, the last simulation result with initial deviation of the trajectory shows that the fuzzy controller is able to control the system to the required state when there are perturbations.

The method of developing fuzzy rules and determining the physical value domain implemented in the article gives good simulation results of the fuzzy controller, which can be applied for designing the fuzzy controller for milling robots.

Difficulties in cutting force calculation and accurately determining the dynamic quantities for complex structural robot systems are always a challenge. Besides, one cannot compensate for all the perturbations such as noise and vibrations occurring due to interactions when the robot is machining or from the environment. Therefore, it is not possible to design an ideal explicit controller. Finding methods to eliminate the aforementioned factors when applying robots in machining is indispensable and compulsory. The controller based on fuzzy logic has been validated to be feasible and reliable via simulation results. It can be developed in practise to overcome the aforementioned uncertainty.

Author Contributions: Methodology, conceptualization and formal analysis, K.B.P.; investigation, K.B.P., H.T.H. and S.V.H.; survey the backgrounds, K.B.P., H.T.H. and S.V.H.; design and collect the preprocessing data, K.B.P., H.T.H.; visualization, writing–original draft, K.B.P., H.T.H.; writing–review & editing, K.B.P.; software, K.B.P.;

check the draft, H.T.H. and S.V.H.; discussion and estimation the efficiency of this methodology, K.B.P., H.T.H. and S.V.H.; layout editing, H.T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Appendix A. Summary Fuzzy Logic

Unlike the crisp control that works based on a crisp logic, i.e., binary logic, the fuzzy controller works based on fuzzy logic, implementing many-valued or infinite-valued logic. The basis of crisp logic is that the crisp set has rigid boundaries while the fuzzy logic depends on fuzzy set with soft boundaries. The fuzzy set enables us to represent the real world easily and closely to human thinking. Crisp logic and fuzzy logic are mathematical languages, crisp sets and fuzzy sets include objects, elements, etc., and they are the basis for the mathematical operations. In the theories of fuzzy set, fuzzy logic develops a pretty large mathematical basis for performing mathematical operations on fuzzy sets. Accordingly, firstly, the fuzzy set elements are assigned a linguistic variable in which the linguistic values represented by the element’s membership function on the fuzzy set. Based on the universe of discourse, by quantitative or qualitative inferences, the linguistic value of a linguistic variable is determined. In fuzzy control, this process is called fuzzification process. In order for fuzzy inference to be performed with linguistic variables on fuzzy sets, the tools are developed. Firstly, characteristic parameters of fuzzy set; membership function to determine membership grade of a variable of an element in a fuzzy set. The Inferences: fuzzy relations; fuzzy rule base; Compositional Rule of Inferences . . . In fuzzy control, these tools are used to perform two main functions: implication and composition of fuzzy rules to come to desired results. This process plays an important role in the fuzzy controller (Appendix B and Figure A1). The physical system only receives and performs commands by physical values, therefore, defuzzification is the process of converting linguistic values into physical (crisp) values based on fuzzy inferences. Accordingly, the fuzzy controller operates in a process with three above-mentioned modules (Appendix B).

Fuzzy logic is a language, a mathematical language, like any other language we know. The elements on which fuzzy logic depend is called fuzzy set and it is also alternatively called the fuzzy algebra. Fuzzy logic is essentially a mixture of various mathematical languages to define another new language as fuzzy logic. Let X be a non-empty set (called the universal set or the universe of discourse or simply domain), which is a set that has all possible members of a particular domain.

A Fuzzy Set F is determined in universal set X in which each element is an ordered pair $(x, \mu_F(x))$, where: $\mu_F: X \rightarrow [0, 1]$. Logical mapping μ_F is considered as membership function of fuzzy set F .

Characteristic parameters of fuzzy set:

The support (defined domain) of fuzzy set F (defined on the set X) is the crisp subset and defined as

$$\text{Supp}(F) = \{x \in X : \mu_F(x) > 0\} \tag{A1}$$

The core (trusted domain) of a fuzzy set F (defined on the set X) is the crisp subset and defined as

$$\text{Core}(F) = \{x \in X : \mu_F(x) = 1\} \tag{A2}$$

The height (h) of fuzzy set F (defined on the set X) is the maximum membership function value attained by any point:

$$h(F) = \sup_{x \in X} \mu_F(x) \tag{A3}$$

An α -cut of a fuzzy set F (defined on the set X) is a crisp set which its each membership function value in F are greater than or equal to α .

$$\alpha F = \{x \in X : \mu_F(x) \geq \alpha\} \tag{A4}$$

A strong α -cut of a fuzzy set F (defined on the set X) is a crisp set which its each membership function value in F are strictly greater than α .

$$\alpha^+F = \{x \in X : \mu_F(x) > \alpha\} \tag{A5}$$

It is easy to see that:

$$\text{Core}(F) = {}^1F; \text{Supp}(F) = {}^{0+}F \tag{A6}$$

The crossover point of a fuzzy set F (defined on the set X) is a point x in X such that

$$\text{Crossover}(F) = \{x \in X : \mu_F(x) = 0.5\} \tag{A7}$$

Typical Membership Function: There are many methods of establishing membership functions to determine the membership grade of an x element on a fuzzy set F , such as: triangular membership function, trapezoidal form, Gaussian form, Sign form, Sigmoidial form, Campanulate form. In fuzzy logic control, the triangular and trapezoidal membership functions are often used.

Operations on fuzzy sets: There are many operations on fuzzy sets, but in this paper we introduce three basic operations: unions, intersection of two fuzzy sets and fuzzy complementation. Consider two fuzzy sets A, B on the universe of discourse X . For a given element x of the X , these operations are introduced. Based on the definition of fuzzy union and intersection, there are many formulas that can be applied to calculate the membership function of the union or intersection of two fuzzy sets, such as the formula based on maximum rule, Lukasiewicz rule, direct sum formula, etc. Here, the formulas based on maximum and minimum rules are introduced:

Unions (max rule):

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \vee \mu_B(x); \forall x \in X \tag{A8}$$

Intersection (min rule):

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \wedge \mu_B(x); \forall x \in X \tag{A9}$$

Complement:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x); \forall x \in X \tag{A10}$$

Linguistic variable: A linguistic variable is a variable that represents a physical quantity by words or sentences in natural or artificial language. A linguistic value is a natural language term which is derived using quantitative or qualitative reasoning. A fuzzy linguistic variable x will take values that are linguistic values. Thus, a variable can both be represented by physical values and linguistic values. For example, velocity variables can be represented by two value domains. The physical value domain:

$$V = \{x \in R : x \geq 0\} \tag{A11}$$

The linguistic value domain:

$${}^L V = \{\text{veryslow}(VL), \text{slow}(L), \text{normal}(M), \text{fast}(F), \text{veryfast}(VF)\} \tag{A12}$$

Here, the universal set is the velocity set V , the fuzzy set ${}^L V$ is defined on set V . With a physical value $x \in V$, the membership function can be determined, corresponding to subsets of ${}^L V$, and represented by vector μ as follow:

$$x \rightarrow \mu = [\mu_{VL}(x), \mu_L(x), \mu_M(x), \mu_F(x), \mu_{VF}(x)]^T \tag{A13}$$

The above mapping is called the fuzzification process, in which the physical value of a variable is transformed into linguistic value.

Fuzzy relations: A fuzzy relation R on Cartesian product of universal sets $U_1 \times U_2 \times \dots \times U_n$ is any fuzzy subset of $U_1 \times U_2 \times \dots \times U_n$. Thus, a fuzzy relation R is defined by a membership function

$$\mu_R(x_1, x_2, \dots, x_n) : U_1 \times U_2 \times \dots \times U_n \rightarrow [0, 1] \text{ or } \mu_R(x_1, x_2, \dots, x_n) \in [0, 1] \quad (A14)$$

If the Cartesian product is formed by just two sets $U_1 \times U_2$, the relation is called a binary fuzzy relation on $U_1 \times U_2$.

A relation R between fuzzy sets A and B denotes composition of A and B

$$R = \{(x, y), \mu_R(x, y) : \mu_R(x, y) = \min(\mu_A(x), \mu_B(y)) \text{ or } \mu_R(x, y) = \mu_A(x) \bullet \mu_B(y)\} \quad (A15)$$

Composition of fuzzy relations R and S: Let R, S, T be fuzzy relations on the Cartesian spaces $X \times Y, Y \times Z$, and $X \times Z$, respectively, $x \in X, y \in Y, z \in Z$.

Union:

$$\mu_{R \cup S}(x, y) = \max\{\mu_R(x, y), \mu_S(x, y)\} = \mu_R(x, y) \vee \mu_S(x, y) \quad (A16)$$

Intersection:

$$\mu_{R \cap S}(x, y) = \min\{\mu_R(x, y), \mu_S(x, y)\} = \mu_R(x, y) \wedge \mu_S(x, y) \quad (A17)$$

Complement:

$$\mu_{\bar{R}}(x, y) = 1 - \mu_R(x, y) \quad (A18)$$

Max–Min composition:

$$T = R \circ S = \left\{ (x, y), \max_y \{\min(\mu_R(x, y), \mu_S(y, z))\} : x \in X, y \in Y, z \in Z \right\} \quad (A19)$$

Max–Product composition:

$$T = R \circ S = \left\{ (x, y), \max_y (\mu_R(x, y) \bullet \mu_S(y, z)) : x \in X, y \in Y, z \in Z \right\} \quad (A20)$$

Max–Average composition:

$$T = R \circ S = \left\{ (x, y), \max_y (\mu_R(x, y) + \mu_S(y, z)) : x \in X, y \in Y, z \in Z \right\} \quad (A21)$$

Fuzzy rules and implications: The fuzzy implication (also known as fuzzy rule, fuzzy If-Then rule, or fuzzy conditional statement) is defined in form: “If x is A then y is B”, where A and B are two linguistic variables defined by fuzzy sets A and B on the universal sets X and Y, respectively. There are two well-known fuzzy implication rules, widely used in fuzzy system and fuzzy control:

Min operation rule (Mamdani implication rule): The fuzzy membership function of fuzzy implication

$$\mu_{A \rightarrow B}(x, y) = \min\{\mu_A(x), \mu_B(y)\} = \mu_A(x) \wedge \mu_B(y) \quad (A22)$$

Product operation rule (Larsen):

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \bullet \mu_B(y) \quad (A23)$$

Fuzzy compositional rules: In this paper, the method Mamdani is introduced. The method Mamdani uses a minimum operator as a fuzzy implication operator, and max-min operator for the composition. Suppose fuzzy rule is Single-Input/Single-Output (SISO) Rule Base as R: if x is A then z is C, infer result C' for another rule if x is A' then z is C', where A, C, A' are known. When input data is a singleton such as $A' = x_0$,

$$C' = A' \circ R; \mu_{C'}(x) = \min\{\mu_A(x_0), \mu_C(z)\} = \alpha \wedge \mu_C(z); \text{ where } \alpha = \mu_A(x_0) \tag{A24}$$

When input data is fuzzy set A' ,

$$C' = A' \circ R; \mu_{C'}(x) = \max\{\min(\alpha, \mu_C(z))\} = \max(\alpha \wedge \mu_C(z)); \text{ where } \alpha = \min(\mu_{A'}(x), \mu_A(x)) \tag{A25}$$

The α is called the “matching degree”, “satisfaction degree”, or “firing strength”.

Multi-Input/Single-Output (MISO) Rule Base:

R_i : if x is A_i and y is B_i then z is C_i , $i = 1, 2, \dots, n$. Where $A_i, B_i, C_i, i = 1, 2, \dots, n$ and A', B' are known, the output C' is obtained as follow:

$$C' = \bigcup_{i=1}^n C'_i; \mu_{C'}(z) = \bigvee_{i=1}^n \mu_{C'_i}; \mu_{C'_i}(z) = \alpha_i \wedge \mu_{C_i}(z) \tag{A26}$$

When input data are singletons such as $x = x_0$ and $y = y_0$,

$$C'_i = A' \circ R_i; \alpha_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0) \tag{A27}$$

When input data are fuzzy sets, A' and B' ,

$$C'_i = (A', B') \circ R_i; \alpha_i = \min\left\{\max_x(\mu_{A'}(x) \wedge \mu_{A_i}(x)), \max_y(\mu_{B'}(y) \wedge \mu_{B_i}(y))\right\} \tag{A28}$$

Defuzzification: In practical applications, the control commands work by crisp value. Therefore it is necessary to defuzzify the result of the fuzzy inference. Defuzzification is the process where the mapping is done to convert the fuzzy results into crisp values. There are many methods for defuzzification technique. This paper introduces two well-known and widely used methods in fuzzy system and fuzzy control.

Center of area method (COA):

$$y' = \frac{\int_S y \mu_{C'}(y) dy}{\int_S \mu_{C'}(y) dy}; \text{ or } y' = \frac{\sum_{j=1}^n y_j \mu_{C'}(y_j)}{\sum_{j=1}^n \mu_{C'}(y_j)} \tag{A29}$$

where, the first formula for the case y is continuous; S is the Support (defined domain) of fuzzy set C' ; n is the number of quantization levels of the output; C is a fuzzy set defined on the output dimension (y).

Mean of maximum method (MOM):

$$y' = \sum_{j=1}^k \frac{y_j}{k} \tag{A30}$$

where, y_j is the control action whose membership functions reach the maximum; k is the number of such control actions.

Appendix B. Fuzzy Logic Controller

In general, a fuzzy controller has three main components (Figure A1):

1. The interface consists of a fuzzification module and some additional ones for solving problems such as generating trajectories of motion, differentiation, integration ... Fuzzification is the method of translating a crisp quantity into a fuzzy quantity.

2. The compositional device consists of two main modules that are fuzzy rules and the inference mechanism. Two approaches for construction of fuzzy logic control are Mamdani and Takagi—Sugeno’s approaches. This paper uses the Mamdani approach in developing fuzzy rules as well as fuzzy inference. In order to develop fuzzy rule systems, the essential nature and key characteristics of the object being controlled are analyzed based on expert knowledge and experience. The effectiveness of these based on the fuzzy approach are the most important factors, i.e., deciding whether fuzzy rules are good or not, to ensure good quality control.
3. Defuzzification is the operation process that inverses the process of fuzzification, where the fuzzy results are converted into crisp values for the controller to perform physical operations.

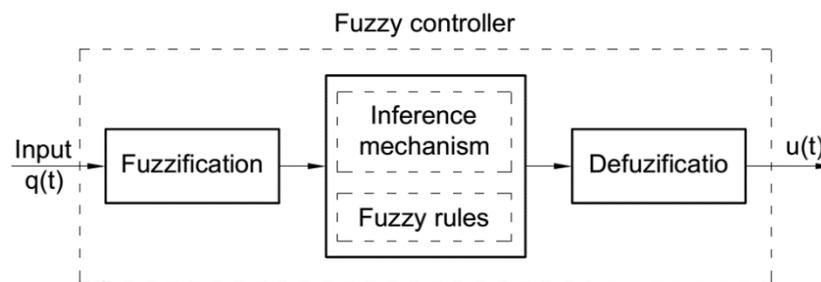


Figure A1. A general fuzzy controller.

References

1. Appleton, E.; Williams, D.J. *Industrial Robot Applications*; Springer Science & Business Media: Berlin, Germany, 2012.
2. Ji, W.; Wang, L. Industrial robotic Machining: A review. *Int. J. Adv. Manuf. Technol.* **2019**, *103*, 1239–1255. [CrossRef]
3. Pandremenos, J.; Doukas, C.; Stavropoulos, P.; Chryssolouris, G. Machining with robots: A critical review. In Proceedings of the DET 2011, Athens, Greece, 28–30 September 2011; pp. 1–8.
4. Petko, M.; Gac, K.; Góra, G.; Karpel, G.; Ochoński, J.; Kobus, K. CNC system of the 5-axis hybrid robot for milling. *Mechatronics* **2016**, *37*, 89–99. [CrossRef]
5. Cen, L.; Melkote, S.N. CCT-based mode coupling chatter avoidance in robotic milling. *J. Manuf. Process.* **2017**, *29*, 50–61. [CrossRef]
6. COMET, E.U. FP7-Project: Plug-and-Produce Components and Methods for Adaptive Control of Industrial Robots Enabling Cost Effective, High Precision Manufacturing in Factories of the Future. Available online: <http://www.cometproject.eu> (accessed on 1 September 2010).
7. Petko, M.; Karpel, G.; Gac, K.; Góra, G.; Kobus, K.; Ochoński, J. Trajectory tracking controller of the hybrid robot for milling. *Mechatronics* **2016**, *37*, 100–111. [CrossRef]
8. Gołda, G.; Kampa, A. Modelling of cutting force and robot load during machining. In Advanced Materials Research. *Trans Tech Publ.* **2014**, *1036*, 715–720.
9. Cen, L.; Melkote, S.N. Effect of robot dynamics on the machining forces in robotic milling. *Procedia Manuf.* **2017**, *10*, 486–496. [CrossRef]
10. Perrusquía, A.; Yu, W.; Soria, A. Position/force control of robot manipulators using reinforcement learning. *Ind. Robot Int. J. Robot. Res. Appl.* **2019**, *46*, 267–280. [CrossRef]
11. Lacerda, H.B.; Lima, V.T. Evaluation of cutting forces and prediction of chatter vibrations in milling. *J. Braz. Soc. Mech. Sci. Eng.* **2004**, *26*, 74–81. [CrossRef]
12. Leonesio, M.; Villagrossi, E.; Beschi, M.; Marini, A.; Bianchi, G.; Pedrocchi, N.; Isaev, A. Vibration analysis of robotic milling tasks. *Procedia Cirp* **2018**, *67*, 262–267. [CrossRef]
13. Wang, G.; Dong, H.; Guo, Y.; Ke, Y. Dynamic cutting force modeling and experimental study of industrial robotic boring. *Int. J. Adv. Manuf. Technol.* **2016**, *86*, 179–190. [CrossRef]
14. Lazoglu, I. A new identification method of specific cutting coefficients for ball end milling. *Procedia Cirp* **2014**, *14*, 182–187.

15. Ghorbani, H.; Moetakef-Imani, B. Specific cutting force and cutting condition interaction modeling for round insert face milling operation. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 1705–1715. [[CrossRef](#)]
16. Nan, C.; Liu, D. Analytical Calculation of Cutting Forces in Ball-End Milling with Inclination Angle. *J. Manuf. Mater. Process.* **2018**, *2*, 35. [[CrossRef](#)]
17. Kaymakci, M.; Kilic, Z.M.; Altintas, Y. Unified cutting force model for turning, boring, drilling and milling operations. *Int. J. Mach. Tools Manuf.* **2012**, *54*, 34–45. [[CrossRef](#)]
18. Davoudinejad, A.; Tosello, G.; Parenti, P.; Annoni, M. 3D finite element simulation of micro end-milling by considering the effect of tool run-out. *Micromachines* **2017**, *8*, 187. [[CrossRef](#)]
19. Huo, D.; Chen, W.; Teng, X.; Lin, C.; Yang, K. Modeling the influence of tool deflection on cutting force and surface generation in micro-milling. *Micromachine* **2017**, *8*, 188. [[CrossRef](#)]
20. Luo, M.; Chong, Z.; Liu, D. Cutting forces measurement for milling process by using working tables with integrated PVDF thin-film sensors. *Sensors* **2018**, *18*, 4031. [[CrossRef](#)]
21. Tuysuz, O.; Altintas, Y.; Feng, H.Y. Prediction of cutting forces in three and five-axis ball-end milling with tool indentation effect. *Int. J. Mach. Tools Manuf.* **2013**, *66*, 66–81. [[CrossRef](#)]
22. Wang, M.; Gao, L.; Zheng, Y. An examination of the fundamental mechanics of cutting force coefficients. *Int. J. Mach. Tools Manuf.* **2014**, *78*, 1–7. [[CrossRef](#)]
23. Díaz-Tena, E.; Ugalde, U.; De Lacalle, L.L.; De la Iglesia, A.; Calleja, A.; Campa, F.J. Propagation of assembly errors in multitasking machines by the homogenous matrix method. *Int. J. Adv. Manuf. Technol.* **2013**, *68*, 149–164.
24. Aydın, M.; Köklü, U. Identification and modeling of cutting forces in ball-end milling based on two different finite element models with Arbitrary Lagrangian Eulerian technique. *Int. J. Adv. Manuf. Technol.* **2017**, *92*, 1465–1480. [[CrossRef](#)]
25. Artetxe, E.; Urbikain, G.; Lamikiz, A.; López-de-Lacalle, L.N.; González, R.; Rodal, P. A mechanistic cutting force model for new barrel end mills. *Procedia Eng.* **2015**, *132*, 553–560. [[CrossRef](#)]
26. Lamikiz, A.; De Lacalle, L.L.; Sánchez, J.A.; Salgado, M.A. Cutting force integration at the CAM stage in the high-speed milling of complex surfaces. *Int. J. Comput. Integr. Manuf.* **2005**, *18*, 586–600. [[CrossRef](#)]
27. Gonzalo, O.; Jauregi, H.; Uriarte, L.G.; de Lacalle, L.L. Prediction of specific force coefficients from a FEM cutting model. *Int. J. Adv. Manuf. Technol.* **2009**, *43*, 348. [[CrossRef](#)]
28. Lamikiz, A.D.; De Lacalle, L.L.; Sanchez, J.A.; Salgado, M.A. Cutting force estimation in sculptured surface milling. *Int. J. Mach. Tools Manuf.* **2004**, *44*, 1511–1526. [[CrossRef](#)]
29. Lin, X.; Wu, G.; Zhang, Y.; Cui, T.; Zhang, B.; Sun, P. The identification of the cutting force coefficients for ball-end finish milling. *Int. J. Adv. Manuf. Technol.* **2019**, *102*, 4121–4135. [[CrossRef](#)]
30. Moges, T.M.; Desai, K.A.; Rao, P.V.M. Modeling of cutting force, tool deflection, and surface error in micro-milling operation. *Int. J. Adv. Manuf. Technol.* **2018**, *98*, 2865–2881. [[CrossRef](#)]
31. Peng, B.; Bergs, T.; Schraknepper, D.; Klocke, F.; Döbbeler, B. A hybrid approach using machine learning to predict the cutting forces under consideration of the tool wear. *Procedia Cirp* **2019**, *82*, 302–307. [[CrossRef](#)]
32. Zhang, X.; Yu, T.; Wang, W. Cutting forces modeling for micro flat end milling by considering tool run-out and bottom edge cutting effect. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2019**, *233*, 470–485. [[CrossRef](#)]
33. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [[CrossRef](#)]
34. Mamdani, E.H. Twenty years of fuzzy control: Experiences gained and lessons learnt. In Proceedings of the 1993 Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, USA, 28 March–1 April 1993; pp. 339–344.
35. Kumar, V.; Nakra, B.C.; Mittal, A.P. A review on classical and fuzzy PID controllers. *Int. J. Intell. Control Syst.* **2011**, *16*, 170–181.
36. Hampel, R.; Wagenknecht, M.; Chaker, N. (Eds.) *Fuzzy Control: Theory and Practice*; Springer Science & Business Media: Berlin, Germany, 2013.
37. Boutalis, Y.; Christodoulou, M.A.; Theodoridis, D.; Kottas, T. *System identification and adaptive control. Theory and Applications of the Neurofuzzy and Fuzzy Cognitive Network Models*; Springer International Publishing: Cham, Switzerland, 2014; 313p.
38. Matía, F.; Marichal, G.N.; Jiménez, E. (Eds.) *Fuzzy Modeling and Control: Theory and Applications*; Atlantis Press: Paris, France, 2014; 288p.
39. Hooda, D.S.; Raich, V. *Fuzzy Logic Models and Fuzzy Control: An Introduction*; Alpha Science International: Oxford, UK, 2017.

40. Sharma, K.D.; Chatterjee, A.; Rakshit, A. Intelligent Adaptive Fuzzy Control. In *Intelligent Control*; Springer: Singapore, 2018; pp. 3–21.
41. De Silva, C.W. *Intelligent Control: Fuzzy Logic Applications*; CRC press: Boca Raton, FL, USA, 2018.
42. Lee, K.H. *First Course on Fuzzy Theory and Applications*; Springer Science & Business Media: Berlin, Germany, 2004.
43. Barros, L.C.D.; Bassanezi, R.C.; Lodwick, W.A. *A First Course in Fuzzy Logic, Fuzzy Dynamical Systems, and Biomathematics: Theory and Applications*; Springer-Verlag: Berlin/Heidelberg, Germany, 2017; 299p.
44. Liu, H.; Wang, T.; Wang, D. Constant cutting force control for CNC machining using dynamic characteristic-based fuzzy controller. *Shock Vib.* **2015**, *2015*. [[CrossRef](#)]
45. Kovač, P.; Rodić, D.; Gostimirović, M.; Savković, B.; Ješić, D. Application of adaptive neuro fuzzy systems for grinding process modeling. In Proceedings of the 6th International Conference on Information Society and Technology ICIST, Society for Information Systems and Computer Networks Pub., Belgrade, Serbia, 28 February–2 March 2016; 2016; pp. 247–250.
46. Huang, S.J.; Shy, C.Y. Fuzzy logic for constant force control of end milling. *IEEE Trans. Ind. Electron.* **1999**, *46*, 169–176. [[CrossRef](#)]
47. Zhao, Y.; Collins, E.G. Fuzzy PI control design for an industrial weigh belt feeder. *IEEE Trans. Fuzzy Syst.* **2003**, *11*, 311–319. [[CrossRef](#)]
48. Huang, L.; Ge, S.S.; Lee, T.H. Fuzzy unidirectional force control of constrained robotic manipulators. *Fuzzy Sets Syst.* **2003**, *134*, 135–146.
49. Kizir, S.; BİNGÜL, Z. Fuzzy impedance and force control of a Stewart platform. *Turk. J. Electr. Eng. Comput. Sci.* **2014**, *22*, 924–939. [[CrossRef](#)]
50. Chen, C.H.; Wang, C.C.; Wang, Y.T.; Wang, P.T. Fuzzy logic controller design for intelligent robots. *Math. Probl. Eng.* **2017**, 1–12. [[CrossRef](#)]
51. Stoian, V.; Ivanescu, M. Robot Control by Fuzzy Logic. In *Frontiers in Robotics, Automation and Control*; IntechOpen Pub.: Rijeka, Croatia, 2008; pp. 111–132.
52. Vashisth, H.; Woo, P.Y. Application of fuzzy logic to robotic control. In Proceedings of the 1996 IEEE IECON. 22nd International Conference on Industrial Electronics, Control, and Instrumentation, Taipei, Taiwan, 9 August 1996; Volume 3, pp. 1867–1872.
53. Chaudhary, H.; Panwar, V.; Sukavanum, N.; Prasad, R. Fuzzy PD+ I based hybrid force/position control of an industrial robot manipulator. *Ifac Proc. Vol.* **2014**, *47*, 429–436. [[CrossRef](#)]
54. de Silva, C.W. Applications of fuzzy logic in the control of robotic manipulators. *Fuzzy Sets Syst.* **1995**, *70*, 223–234. [[CrossRef](#)]
55. Mendes, N.; Neto, P.; Pires, J.N.; Loureiro, A. An optimal fuzzy-PI force/motion controller to increase industrial robot autonomy. *Int. J. Adv. Manuf. Technol.* **2013**, *68*, 435–441.
56. Mendes, N.; Neto, P. Indirect adaptive fuzzy control for industrial robots: A solution for contact applications. *Expert Syst. Appl.* **2015**, *42*, 8929–8935. [[CrossRef](#)]
57. Khoi, P.B.; Van Toan, N. Hedge-Algebras-Based Controller for Mechanisms of Relative Manipulation. *Int. J. Precis. Eng. Manuf.* **2018**, *19*, 377–385. [[CrossRef](#)]
58. Toan, N.V.; Khoi, P.B. Fuzzy-based-admittance controller for safe natural human–robot interaction. *Adv. Robot.* **2019**, 1–9. [[CrossRef](#)]
59. Wakileh, B.A.M.; Gill, K.F. Use of fuzzy logic in robotics. *Comput. Ind.* **1988**, *10*, 35–46. [[CrossRef](#)]
60. Khoi, P.B.; Van Toan, N. Optimizing Neuro-Fuzzy System for Robot Control. *J. Sci. Technol.* **2014**, *52*, 685–699.
61. Lin, C.H.; Wang, S.H.; Lin, C.J. Interval Type-2 Neural Fuzzy Controller-Based Navigation of Cooperative Load-Carrying Mobile Robots in Unknown Environments. *Sensors* **2018**, *18*, 4181. [[CrossRef](#)]
62. Van Toan, N.; Khoi, P.B. A control solution for closed-form mechanisms of relative manipulation based on fuzzy approach. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1–11. [[CrossRef](#)]
63. Hsu, F.Y.; Fu, L.C. A new adaptive fuzzy hybrid force/position control for intelligent robot deburring. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 10–15 May 1999; Volume 3, pp. 2476–2481.
64. Hendzel, Z.; Burghardt, A.; Gierlak, P.; Szuster, M. Conventional and fuzzy force control in robotised machining. *Solid State Phenom. Trans Tech Publ.* **2014**, *210*, 178–185. [[CrossRef](#)]
65. Cuka, B.; Kim, D.W. Fuzzy logic based tool condition monitoring for end-milling. *Robot. Comput.-Integr. Manuf.* **2017**, *47*, 22–36. [[CrossRef](#)]

66. Hsu, F.Y.; Fu, L.C. Intelligent robot deburring using adaptive fuzzy hybrid position/force control. *IEEE Trans. Robot. Autom.* **2000**, *16*, 325–335. [[CrossRef](#)]
67. Khoi, P.B.; Van Toan, N. Application of fuzzy logic for controlling mechanisms of relative manipulation robot (MRM robot). *J. Sci. Technol.* **2016**, *54*, 385–401. [[CrossRef](#)]
68. Khoi, P.B. Calculation and simulation the program motion of mechanisms of relative manipulation (MRM). *J. Sci. Technol.* **2009**, *3*, 19–28.
69. Khoi, P.B. Kinematic problems of programming control of mechanism of relative manipulation. In Proceedings of the National Conference on Mechanics, Hanoi, Vietnam, 8–9 April 2009; pp. 317–323.
70. Khoi, P.B.; Hai, H.T. Investigation of kinematics and motion planning for mechanical machining robots. *Proc. Natl. Conf. Eng. Mech. Vietnam* **2015**, *2*, 407–418.
71. Hai, H.T. Kinematic Modelling of a Robot in Form-shaping Milling Complex Surfaces. *Eur. J. Eng. Res. Sci.* **2019**, *4*, 26–31. [[CrossRef](#)]
72. Khoi, P.B.; Huy, L.Q. (Eds.) Kinematic modeling of of the process of grinding turbine blades by using robots. In Proceedings of the 10th National conference on Mechanics, Hanoi, Vietnam, 8–9 December 2017; Volume 1, pp. 803–812.
73. Afonin, V.L.; Khoi, P.B. *Method for Calculating Action and Constrained Reactions Forces in Mechanisms of Relative Manipulation while Executing Programming Motion*; IMASH, RAS: Moscow, Russia, 1997.
74. Khoi, P.B. Applying Principle of Compatibility for Analyzing Forces of Mechanism of Relative Manipulation Robot. PhD Thesis, Mechanical Engineering Research Institute of the Russian Academy of Sciences (IMASH RAS), Moscow, Russia, 4 December 1997.
75. Khoi, P.B. Dynamical investigation of relation manipulation mechanisms in mechanical processing. In Proceedings of the National Conference on Mechanics, Hanoi, Vietnam, 8–9 April 2004; pp. 181–190.
76. Khoi, P.B.; Hai, H.T. Robot dynamics in mechanical processing. *Proc. Natl. Conf. Eng. Mech. Vietnam* **2015**, *2*, 419–427.
77. Khoi, P.B.; Hai, H.T. Force analysis of a robot in machining process. *Proc. Natl. Conf. Mach. Mech. Vietnam* **2015**, *1*, 346–359.
78. Hai, H.T. Effect of cutting forces on the form-shaping motion in robotic milling. *Am. J. Eng. Res.* **2019**, *8*, 176–185.
79. Hai, H.T. Inverse dynamic analysis of miling machining robot: Application in calibration of cutting force. *Vietnam J. Sci. Technol.* **2019**, *57*, 773–787.
80. Khoi, P.B.; Hai, H.T.; Sinh, H.V. Control of inverse dynamics of robot in milling. *Proc. Natl. Conf. Eng. Mech. Vietnam* **2017**, *1*, 352–361.
81. Abele, E.; Bauer, J.; Pischon, M.; Stryk, O.V.; Friedmann, M.; Hemker, T. Prediction of the tool displacement for robot milling applications using coupled models of an industrial robot and removal simulation. In Proceedings of the CIRP 2nd Inter Conference on Process Machine Interactions, Vancouver, BC, Canada, 10–11 June 2010.

