J Intell Robot Syst (2009) 54:733–754
DOI 10.1007/s10846-008-9286-6

Vision-Based Robotic Motion Control for Non-autonomous Environment

Shiuh-Jer Huang · Shian-Shin Wu

Received: 29 October 2007 / Accepted: 12 September 2008 / Published online: 14 October 2008

Abstract A visual servo control system with SOPC structure is implemented on a retrofitted Mitsubishi Movemaster RV-M2 robotic system. The hardware circuit has the functions of quadrature encoder decoding, limit switch detecting, pulse width modulation (PWM) generating and CMOS image signal capturing. The software embedded in Nios II micro processor has the functions of using UART to communicate with PC, robotic inverse kinematics calculation, robotic motion control schemes, digital image processing and gobang game AI algorithms. The digital hardware circuits are designed by using Verilog language, and programs in Nios II micro processor are coded with C language. An Altera Statrix II EP2S60F672C5Es FPGA chip is adopted as the main CPU of the development board. A CMOS color image sensor with 356 × 292 pixels resolution is selected to catch the environment time-varying change for robotic vision-based servo control. The system performance is evaluated by experimental tests. A gobang game is planned to reveal the visual servo robotic motion control objective in non-autonomous environment. Here, a model-free intelligent self-organizing fuzzy control strategy is employed to design the robotic joint controller. A vision based trajectory planning algorithm is designed to calculate the desired angular positions or trajectory on-line of each robotic joint. The experimental results show that this visual servo control robot has reliable control actions.

Keywords Visual servo · Robotic system · Self-organizing fuzzy control · FPGA chip

S.-J. Huang (✉) · S.-S. Wu
Department of Mechanical Engineering,
National Taiwan University of Science and Technology,
No. 43, Keelung Road, Sec. 4, Taipei 106, Taiwan
e-mail: sjhuang@mail.ntust.edu.tw

S.-J. Huang
Department of Vehicle Engineering,
National Taipei University of Technology,
No. 1, Sec. 3, Chung-Hsiao East Road, Taipei 106, Taiwan

1 Introduction

Robotic manipulator has important application role in industrial automation. The specifications of robotic structure, performance, motion speed and accuracy are fully dependent on the application types. In addition, various sensing and measuring devices are integrated into the robotic system to construct new automatic systems for various new application fields. Hence, how to design appropriate robotic mechanism structure, motion planning and control, and integrate the complementary devices and technology into a flexile automatic system are still the hot robotic research topics. Sensors and measuring schemes are the necessary technologies in robotic automation systems to constitute different functions. Vision system is the key component for the robotic automation operation in non-autonomous environment.

Since robotic manipulators are multi-input and multi-output systems, their motion planning and control have complicated operating features. Traditional single CPU controller is difficult to achieve this computation and communication works with appropriate efficiency. Due to the development of semi-conductor and digital circuit design technology, the new system-on-programmable-chip (SOPC) provides fertilize functions for the servo control, image processing and network communication purposes to operate on a single chip by software and hardware implementation. Originally, most of the FPGA chips are used in communication and signal processing. Currently, they have been employed in motor control [1], the PID control of robotic arm [2], and mini robot football game control [3]. It provides the functions of motion control, measuring signal integration and network communication. Here, a SOPC system chip is employed to construct a vision-based robotic control system in non-autonomous environment.

Different hardware control structures will influence the selection of control algorithms. If the robotic system dynamic model is well known, the traditional model-based computed torque method has an excellent control performance [4]. However, the accurate dynamic model for a multi-axis manipulator is difficult to establish due to its nonlinear and dynamics coupling behaviors. Then adaptive control [5, 6] was proposed to improve this problem by introducing the system identification technique. It is time-consuming work and it is not suitable for joint distributed controller structure. Hence the model-free intelligent control scheme is adopted in the robotic motion control field [7, 8]. However, the design of a traditional fuzzy controller depends fully on an expert or the experience of an operator to establish the fuzzy rules bank. Generally, this knowledge is difficult to obtain. A time consuming adjusting processes is required to achieve the specified control performance. Hence, a self-organizing fuzzy controller with learning ability was proposed in [9], new establishing processes of the fuzzy rules bank had been found for reducing the trial-and-error effort. It simplifies the designing processes and facilitates the implementation of a fuzzy controller. Later, the modified learning methods for self-organizing fuzzy controller (SOFC) were proposed in [10, 11] for further simplification. The output error and error change were used to correct the control inputs instead of the performance table. It was employed to manipulate the robotic motion control [12]. Later, a self-tuning fuzzy controller was proposed by using a specified performance index based upon certain system responses to self-adjust fuzzy parameters and modify fuzzy rules bank [13]. A self learning mechanism combined with analytic fuzzy rules was used as a self regulating fuzzy controller to improve the control performance [14]. However, it needs a complicated learning mechanism or a specific performance decision table which is designed by trial-and-error process. Its application still has certain difficulty.

Actually, human control action is based on a few rules in the brain through continuous regulating to generate complicated control actions instead of depending on many rules. This concept is employed in this paper to develop a SOFC. For each sampling instant, the output response error and error change stimulate two fuzzy subsets of their corresponding universe of discourse E and CE. Then at most four fuzzy rules in the fuzzy rules table were fired in spite of the whole rules bank. Hence only four rules are modified in each sampling step. This approach can reduce significantly the burden of data base and computing time for increasing the sampling frequency. Since, this approach has learning ability to establish and regulate the fuzzy rules bank continuously, its control implementation can be started with zero initial fuzzy rules. The fuzzy rules were adjusted on-line based on a simple equation instead of a performance decision table.

The development of image processing technology and machine vision application are hot research topics in recent years, too. Machine vision has been widely used in industry for inspection, measurement and recognition purposes [15–17] to substitute the labor and reduce the manual error and production cost. How to integrate the machine vision into the robotic system for improving the robotic manipulator working ability in widespread industrial applications is a popular research investigation in recent years [18–20]. Here, a machine vision is integrated into an old Mitsubishi Movemaster RV-M2 manipulator for establishing an automatic gobang game system. The machine vision is introduced to search the pieces positions of both players on a chessboard and find a competitive strategy, then guide the robot to bring the chess piece to the appropriate location. This system can be employed in a flexible non-autonomous environment for executing random assembly or pick-and-place and collision avoidance operations.

2 System Structure

The vision servo robotic control structure with Altera Nios II embedded development kit is shown in Fig. 1. The Nios II development board is designed to send digital signals to the Lab made DC servo motor drivers with LMD18200 IC for actuating each joint motor of the robotic system; and detect each joint motor angular position and the workspace image data for constituting a multi inputs non-autonomous closed loop control system. This development board has an embedded Altera Stratix system-on-a-programmable-chip (SOPC). User can define the micro-processor specification under a graphic interface, integrate the digital logic circuits coded with Verilog VHDL or AHDL hardware languages to constitute a FPGA control system. Here, Verilog HDL (Hardware Description Language) is selected to code the hardware circuits of this vision servo robotic control system. The main servo control system can be divided into FPGA internal hardware implementation and Nios II micro-processor software programs implementation two parts. The main functions of FPGA hardware circuits are motor optical encoder decoding, limit switch detecting, pulse width modulation (PWM) generating and CMOS image signal

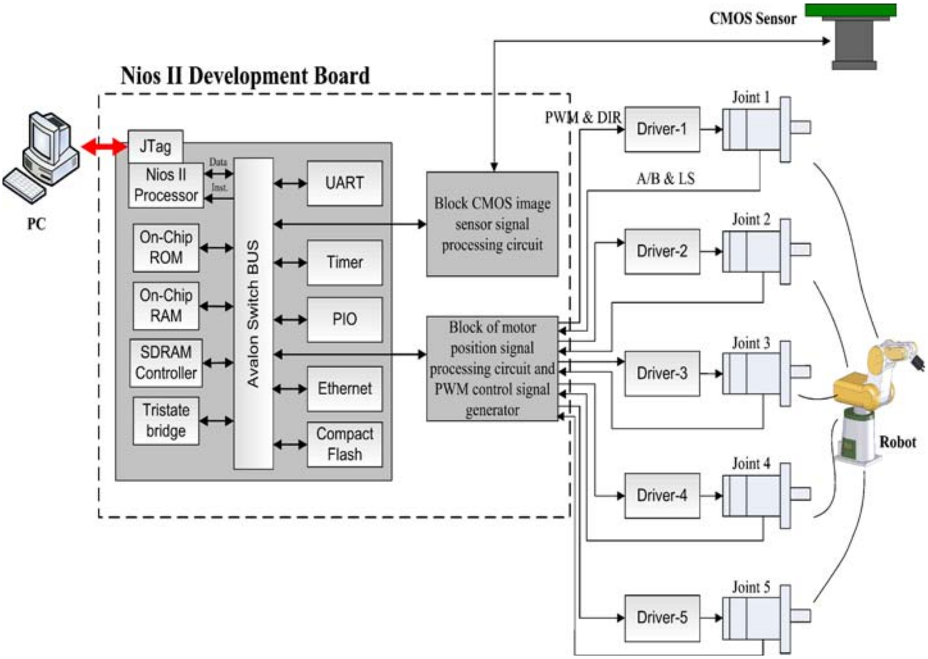
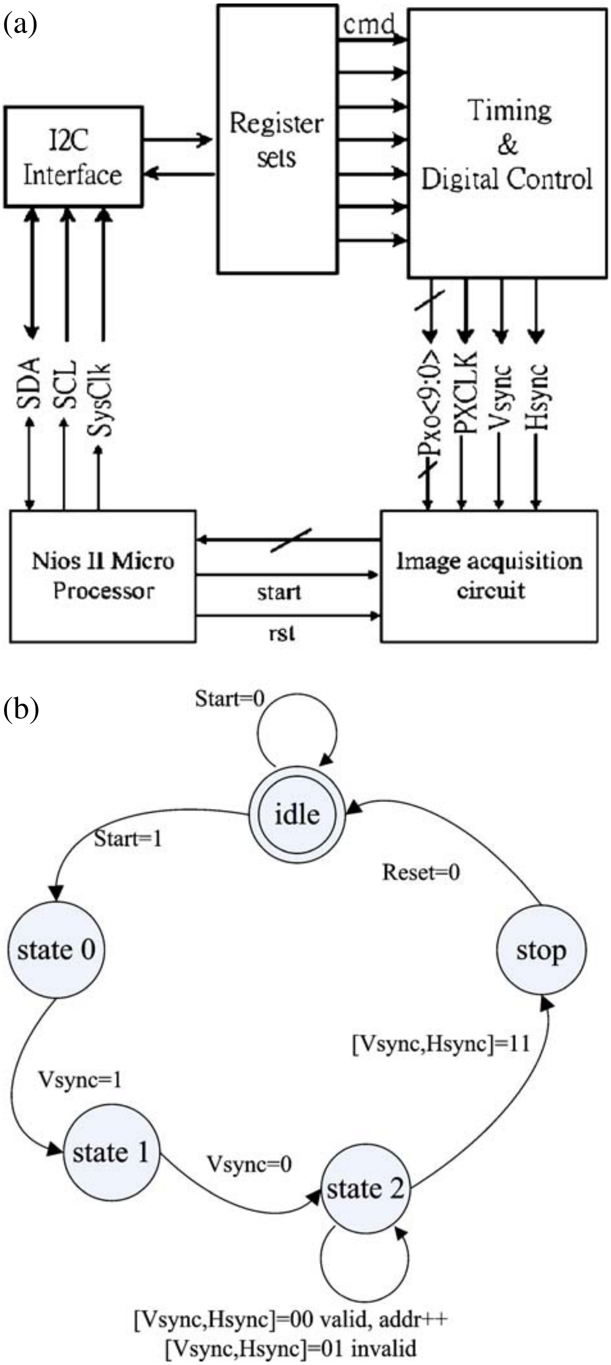


Fig. 1 SOPC Robotic control system structure

capturing. The functions of the Nios II micro-processor software programs are the communication with PC by using Universal Asynchronous Receiver Transmitter (UART), robotic inverse kinematics calculation, robotic motion trajectory planning, robotic motion control schemes, digital image processing and gobang game AI algorithms. A four-bit delay filter with four serial D type flip-flop is designed to suppress the high frequency noise of feedback signals. A 13 bits control signal is used to regulate the duty circle of the servo motor PWM signal. A CMOS color image sensor with 356 × 292 pixels resolution is used to extract the environment time-varying change for robotic vision-based motion control. The presetting process of this image sensor is to set its internal buffer through the I2C dual direction serial communication interface. It can specify the DAC, color gain, global gain and output frequency [21]. In this study, 15 frames can be detected per second. An image extracting hardware circuit is designed and integrated with Nios II micro-processor and I2C dual direction serial communication interface as shown in Fig. 2a. A finite states flow chart is employed to describe the hardware circuit operation procedures as Fig. 2b. When the initial idle state receives a start signal, it will transfer to state 0 for waiting the next frame beginning signal $Vsync\uparrow$. Then it will enter the state 1 for waiting effective pixel ($Vsync\downarrow$). State 2 is designed to distinguish the effective pixel ($[Vsync, Hsync] = 00$) and useless pixel ($[Vsync, Hsync] = 01$). Only effective pixel signal is extracted by this hardware circuit. When both the horizontal and vertical synchronous signals are high levels, the system transfer into stop state and finally reverts to idle state.

Fig. 2 **a** Image extracting hardware circuit; **b** finite states operation flow chart



The robotic system is an old Mitsubishi Movemaster RV-M2 manipulator with a retrofitted FPGA control structure to substitute original commercial controller. The motors encoder resolutions are 740, 970, 740, 630, and 460 pulses per degree for joint 1 to joint 5, respectively.

3 Trajectory Planning and Inverse Kinematics

In order to achieve the manipulator positioning and trajectory tracking control in workspace, the kinematics and inverse kinematics and trajectory planning should be derived and planned. Generally, the end-effector working position or motion path in Cartesian space are converted into control variables in joint coordinates for controlling purpose by using the inverse kinematics and Denavit–Hartenberg (D–H) transformation matrix. Although some efficient analysis methods had been proposed [22, 23], they are time consuming and complicated mathematical operations. Firstly, the orientation and position transformation matrix of end-effector with respect to the inertial coordinate frame can be derived from the multiplication of each joint (D–H) transformation matrix as

$$^{\mathrm{ref}}T_{\mathrm{tool}} =^{0} A_1 \times^{1} A_2 \times^{2} A_3 \times^{3} A_4 \times^{4} A_5 = \begin{bmatrix} u_x & v_x & w_x & p_x \\ u_y & v_y & w_y & p_y \\ u_z & v_z & w_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $^{i-1}A_i$ is a D–H homogeneous transformation matrix of each joint for $i = 1, 2, 3, 4, 5$.

$$^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Those robot link parameters θ_i, α_i, a_i and d_i are listed in Table 1. $[\mu \;\; v\, w]$ is a rotational matrix and p is a position vector of the manipulator end-effector with respect to the base coordinate system.

Traditionally, the procedure of solving this inverse kinematics equation is to firstly solve θ_1 from the transformation matrix $[T_5]$ by multiplying it with $^{0}A_1^{-1}$ as

$$\begin{bmatrix} u_xC_1 + u_yS_1 & v_xC_1 + v_yS_1 & w_xC_1 + w_yS_1 & p_xC_1 + p_yS_1 - a_1 \\ -u_z & -v_z & -w_z & d_1 - p_z \\ -u_xS_1 + u_yC_1 & -v_xS_1 + v_yC_1 & -w_xS_1 + w_yC_1 & p_yC_1 - P_xS_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{234}C_5 & -C_{234}S_5 & -S_{234} & f_1 \\ S_{234}C_5 & -S_{234}S_5 & C_{234} & f_2 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Table 1 D–H parameters of Mitsubishi RVM2 robot

	a_i	α_i	d_i	θ_i
1	a_1 (120 mm)	α_1 ($-\pi/2$)	d_1 (400 mm)	θ_1
2	a_2 (250 mm)	0	0	θ_2
3	a_3 (200 mm)	0	0	θ_3
4	0	α_4 ($-\pi/2$)	0	θ_4
5	0	0	d_5 (165 mm)	θ_5

where $C_i \equiv \cos\theta_i$, $S_i \equiv \sin\theta_i$, $C_{ij} \equiv \cos(\theta_i + \theta_j)$, $S_{ij} \equiv \sin(\theta_i + \theta_j)$, $C_{ijk} \equiv \cos(\theta_I + \theta_j + \theta_k)$, $S_{ijk} \equiv \sin(\theta_I + \theta_j + \theta_k)$, and

$$\begin{aligned} f_1 &= a_2C_2 + a_3C_{23} - d_5S_{234} \\ f_2 &= d_5C_{234} + a_2S_2 + a_3S_{23} \end{aligned} \tag{4}$$

By comparing the matrices terms on both sides of Eq. 3, we can obtain

$$p_yC_1 - p_xS_1 = 0 \quad \text{and} \quad u_xS_1 - u_yC_1 = S_5 \tag{5}$$

Then, θ_1 and θ_5 can be solved as

$$\theta_1 = \tan^{-1}\left[\frac{p_x}{p_y}\right] \tag{6}$$

$$\theta_5 = \sin^{-1}\left(u_xS_1 - u_yC_1\right) \tag{7}$$

And

$$\theta_{234} = \tan^{-1}\left[\frac{-u_z}{u_xC_1 + u_yS_1}\right] \tag{8}$$

After some triangular mathematical operations, θ_2 and θ_3 can be solved as

$$\theta_3 = \cos^{-1}\left[\frac{b_1^2 + b_2^2 - a_2^2 - a_3^2}{2a_2a_3}\right] \tag{9}$$

$$\theta_2 = \tan^{-1}\left[\frac{-b_1a_3S_3 + b_2\left(a_2 + a_3C_3\right)}{b_1\left(a_2 + a_3C_3\right) + b_2a_3S_3}\right] \tag{10}$$

Where

$$\begin{aligned} b_1 &= p_xC_1 + p_yS_1 - a_1 + d_5S_{234} \\ b_2 &= -p_z + d_1 - d_5C_{234} \end{aligned} \tag{11}$$

Then

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \tag{12}$$

Since the chessboard and the operation range of this vision based robotic system are limited on horizontal plane, the end-effector orientation is specified as orthogonal and point down to the X–Y horizontal plane. Then the Denavit–Hartenberg

(D–H) transformation matrix of end-effector with respect to the reference inertia coordinate is specified as

$$^{ref}T_{tool} = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \cdot {}^4A_5 \equiv \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & -1 & 0 & y \\ 0 & 0 & -1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

With this end-effector orientation limitation for specific application, the trigonometric functions calculation for the inverse kinematics can be reduced from 17 times to seven comparing with that of traditional inverse kinematics. The computer time on the Nios II SOPC can be reduced from 4.5 to 2.5 ms for increasing the system closed loop sampling frequency. Based on these kinematics and inverse kinematics equations, the joint and Cartesian space trajectory planning can be carried out and the robotic motion control performance can be evaluated.

For the multi-input and multi-output system, the motion trajectory of each joint corresponding to the point to point (PTP) motion in Cartesian space need be planned to move and stop at the same time for the simultaneous motion purpose. The trapezoid speed curve with a constant acceleration and deceleration time T_a, a constant acceleration value a_m and the total motion time T, is the most popular trajectory planning for the PTP motion. There relationships are

$$a_m T_a = v_m \quad \text{and} \quad T = \frac{X}{v_m} + T_a \tag{13}$$

The acceleration, acceleration time and the total motion time are three parameters for this trajectory planning. Since the motors of this five degrees of freedom (DOF) commercial robot had been appropriate designed, they can normally operate in its workspace. The axis with largest angular moving range is chosen to calculate the total motion time T based on the specified maximum acceleration, a_m, and maximum velocity, v_m. Then the acceleration time and the maximum velocity of the other axes can be derived based on their moving angles, total motion time and the maximum acceleration information.

$$T_{ai} = \frac{T}{2} - \sqrt{\frac{T^2}{4} - \frac{X_i}{a_m}} \quad \text{and} \quad v_m = a_m T_{ai} \tag{14}$$

Those trajectory planning parameters of this five DOF robot are summarized in Table 2.

Table 2 The parameters of MIMO trapezoid speed curve trajectory planning

	Moving angle	T_a	T	v	a
Largest moving angle axis	$X \geq \frac{v_m^2}{a_m}$	$\frac{v_m}{a_m}$	$T = \frac{X}{v_m} + T_a$	v_m	a_m
Largest moving angle axis	$X < \frac{v_m^2}{a_m}$	$\sqrt{\frac{X}{a_m}}$	$2T_a$	$\sqrt{Xa_m}$	a_m
Other axes	X_i	$\frac{T}{2} - \sqrt{\frac{T^2}{4} - \frac{X_i}{a_m}}$	T	$a_m T_{ai}$	a_m

4 Self-Organizing Fuzzy Learning

Since the control input of a fuzzy logic control system is calculated from fuzzy inference, it does not need any system mathematical model. This model free feature eliminates the difficult modeling process when designing controllers for complicated dynamic systems. Fuzzy logic control has been successfully employed in many industrial applications. However, there have been no definite procedures for designing an optimal fuzzy controller. The parameters of the membership functions and fuzzy rules need to be selected by an expert or based on experience. Usually these parameters are designed by a trial-and-error process to obtain an appropriate performance, which is often a time-consuming process. Therefore, the intelligent fuzzy controller with learning ability was proposed to facilitate implementation.

The main difference between a SOFC and a traditional fuzzy controller is the properties of their data base and fuzzy rules. The data base and fuzzy rules of a traditional fuzzy controller are fixed after the design step. However, the data base and fuzzy rules of a SOFC are accumulated or modified continuously based upon a learning strategy during the control processes to improve the system output precision. The SOFC was first proposed in [9] and its learning structure was modified in [10, 11]. The learning scheme was based on a performance decision table. However, the design of a performance decision table is as difficult as the design of a fuzzy rule table. Then the output error and error change were employed directly to modify the linguistic fuzzy rules table [24]. The fuzzy rules table of this SOFC can be started with zero initial fuzzy rules.

The self-organizing part is introduced into a traditional fuzzy controller to constitute a self-organizing fuzzy controller as shown in Fig. 3. The self-organizing part consists of three steps: performance measure, model estimation and rule modification. Usually, two physical features, for example system output error and error change, are measured as performance index to establish a performance decision table which

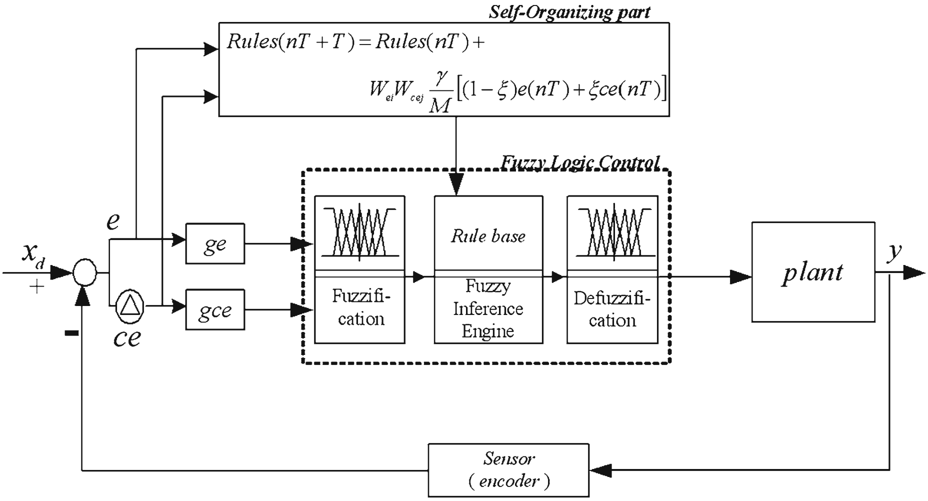


Fig. 3 Self-organizing fuzzy controller block diagram

is similar to establishing a fuzzy rules table. The model estimation is used to find the relationship between the system output performance and the control input. Then the performance measure is employed to calculate the correction value of each fuzzy rule based on the estimation model. However, the appropriate performance decision table is difficult to establish for each control system. Here a real-time linguistic SOFC control strategy is proposed by using two parameters to take care of the function of the performance measure instead of the performance decision table.

For the rules modification, the dimension of the rules table is limited to that of the original fuzzy rules table, the correction value of each fuzzy rule is introduced into the original fuzzy rule as a new control rule. This approach can improve the data base expansion defect of the Procky [9] scheme and increase the computing speed. Generally, the system dynamic response feature can be represented as an auto-regression and moving average (ARMA) model:

$$\begin{aligned}
\theta\,(nT) &= A\left(q^{-1}\right)\theta\,(nT-T) + Mu\,(nT-mT) + B\left(q^{-1}\right)u\,(nT-mT-T)\\
A\left(q^{-1}\right) &= a_0 + a_1 q^{-1} + \cdots + a_{r-1}q^{-(r-1)}\\
B\left(q^{-1}\right) &= b_0 + b_1 q^{-1} + \cdots + b_{s-m-1}q^{-(s-m-1)}
\end{aligned} \tag{15}$$

where mT is the system time delay. M is the direct system forward gain for this position control system. The values of r, s and m depend on the dynamic characteristics of the control system. Due to system nonlinearity and uncertainty, they are difficult to estimate for this five DOF robotic system. Fortunately, fuzzy control has a model free feature. It does not require definite mathematical model and system parameters. Hence, they are not employed in the following controller design. If the system is excited with a different control input $u'(nT-mT)$ at time step $nT-mT$, there will be a new output value at time step nT. The input difference, Δu, of the servo motor control voltage will cause a system joint output deviation $\Delta\theta$. If the deviations $\Delta\theta$ and $\Delta\dot{\theta}$ are small, then the relationships between control input and corresponding output deviations can be described

$$\Delta\theta = M \times \Delta u \quad \text{and} \quad \Delta\dot{\theta} = \frac{M}{T}\Delta u \tag{16}$$

If a system at time step nT has angular output error $\Delta\theta$ and error rate $\Delta\dot{\theta}$, the theoretical corresponding control input correction values are Δu_e and Δu_{ce}, respectively.

$$\Delta u_e \equiv \frac{\Delta\theta}{M} \quad \text{and} \quad \Delta u_{ce} \equiv \frac{T\Delta\dot{\theta}}{M} \tag{17}$$

Since the system has only one control input u, the control input correction must be an appropriate combination of the above two terms. In general, the following form can be chosen:

$$\Delta u = (1-\xi)\,\Delta u_e + \xi\,\Delta u_{ce} \tag{18}$$

ξ is a design parameter representing the weighting distribution between Δu_e and Δu_{ce}. If there is a large difference between the system angular output $\theta(nT)$ and desired value θ_d, an appropriate design choice is to select a value $\theta'(nT)$ between $\theta(nT)$ and θ_d. Then the system joint output θ will approach the desired value θ_d gradually with a weighting parameter γ.

$$\theta'(nT) = (1-\gamma)\,\theta(nT) + \gamma\theta_d \quad , \quad 0 < \gamma < 1 \tag{19}$$

Then the system output and output change deviations become

$$\Delta\theta(nT) = \gamma\left[\theta_d - \theta(nT)\right] \equiv \gamma e(nT) \tag{20}$$

$$\Delta\overset{\bullet}{\theta}(nT) = \gamma\dot{e}(nT) \equiv \frac{\gamma}{T}ce(nT) \tag{21}$$

From Eqs. 16 and 20, the correction value of the control input can be represented as

$$\Delta u = \frac{\gamma}{M}\left[(1-\xi)\,e(nT) + \xi ce(nT)\right] \tag{22}$$

The output error, $e(nT)$, and error change, ce(nT), are divided into five fuzzy subsets from -1 to $+1$ with interval 0.5. For each control step, the fuzzy input variables, i.e. the system output error and error change will stimulate two fuzzy subsets of the E and CE universe of discourse, respectively. Since the control input u is derived from the fuzzy rules inference, the rules modification will influence four fuzzy rules for each control step. The correction value of each fuzzy rule is proportional to its exciting strength w_{ij}. The excitation strength is designed as a triangular membership function and calculated with a linear interpolation algorithm. Then the new control input of the ith rule is

$$\begin{aligned} u_i(nT+T) &= u_i(nT) + \Delta u_i \\ &= u_i(nT) + w_{ei}w_{cei}\frac{\gamma}{M}\left[(1-\xi)\,e(nT) + \xi ce(nT)\right] \end{aligned} \tag{23}$$

The term $\frac{\gamma}{M}$ in the above equation can be considered as the correction weighting. Here, M is chosen as 1 in order to eliminate the identification procedure and reduce computing time during implementation. The correction weighting is regulated by the parameter γ only. A larger value of γ will introduce a large correction of fuzzy rules and system output oscillation. This parameter only influences the transient response but not the steady state performance. According to experimental experience, this parameter can be selected as a large value (for example 0.9), and it can be adjusted to a smaller value (for example 0.3) after the learning procedure has converged. It is not crucial for this control strategy. Generally, a γ value of between 0.3 and 0.9 can achieve stable convergent SOFC systems. The appropriate value for the desired dynamic performance can be found from experimental tests.

5 Self-Organizing Fuzzy Controller

The general form of a self-organizing fuzzy control rule can be expressed as

$$R_i \ : \ \text{If } \Delta\theta \text{ is } A \text{ and } \Delta\overset{\bullet}{\theta} \text{ is } B, \ \text{then } U \text{ is } C \tag{24}$$

where R_i is the ith rule, $\Delta\theta$ and $\Delta\overset{\bullet}{\theta}$ are the states of the system output to be controlled and U is the control input voltage. A, B and C are the corresponding fuzzy subsets of the input and output universe of discourse.

The output importance of each rule is dependent on the membership functions of the linguistic input and output variables. The membership functions of the angular error and velocity error and the control voltage can be divided with unequal span in conjunction with the system characteristics. Since the SOFC has learning capability, it does not need to find the appropriate shape of membership functions. The equal span triangular membership function is employed in this paper. The range of the fuzzy variables also can be adjusted according to the system variation by using the gain parameters ge, gce and gu. The membership function used for the fuzzification is a triangular type. The function can be expressed as

$$\mu\left(x\right) = \frac{1}{W}\left(-\left|x-a\right| + W\right) \tag{25}$$

where W is the distribution span of the membership function, x is the input variables and a is the parameter corresponding to value 1 of the membership function.

Since the output response error and error change stimulate two fuzzy subsets of the universe of disclose E and CE, respectively in each sampling interval, four fuzzy control rules in fuzzy rules table are fired only in spite of the whole rules bank. A linear interpolation fuzzy operation scheme is employed to interference and defuzzy the fuzzy variable from these four fuzzy control rules for obtaining the control input voltage for each joint servo motor in each control step. The equation can be described as

$$\begin{aligned}
U_1 &= \underline{U}_{i,j} + \left(\underline{U}_{i+1,j} - \underline{U}_{i,j}\right)\mu_{a_{i+1}}\left(e\right) \\
U_2 &= \underline{U}_{i,j+1} + \left(\underline{U}_{i+1,j+1} - \underline{U}_{i,J+1}\right)\mu_{a_{i+1}}\left(e\right) \\
U &= U_1 + \left(U_2 - U_1\right)\mu_{b_{j+1}}\left(ce\right)
\end{aligned} \tag{26}$$

where $\mu_{a_{ij}}\left(e\right)$ is the membership function value of the fuzzy set linguistic variable, $U_{i,j}$ is the control value of each exciting fuzzy control rule and U is the resulting fuzzy control voltage value of the ith control step.

During the experimental implementation, the fuzzy rule learning scheme shown in Eq. 23 has latent stability problem. It should be investigated and improved for practical learning application. Theoretically, if the fuzzy rules bank has been converged to an ideal situation through self-learning process; it should be suitable for these environmental conditions without further learning. However, the fuzzy update rule is minutely modified except that both e and ce are equal to zero simultaneously. Then the fuzzy control rule u_i may grow and diverge gradually. Narendra and Annaswamy [25] had proposed an e-modification adaptive rule to modify the parameters update rule for improving the adaptive control robustness. Here, this e-modification concept is introduced into the fuzzy rule adapting scheme of a SOFC to improve the control system stability and robustness. Then, the fuzzy rules updating law of this new SOFC, Eq. 23, can be written as:

$$\begin{aligned} u_i(nT+T) &= u_i(nT) + \Delta u_i \\ &= u_i(nT) + w_{ei} w_{\mathrm{ce}i} \frac{\gamma}{M}\big[(1-\xi)\, e(nT) + \xi \mathrm{ce}(nT)\big] \\ &\quad - k\left|\big[(1-\xi)\, e(nT) + \xi \mathrm{ce}(nT)\big]\right| u_i(nT) \end{aligned} \tag{27}$$

Where the last term of Eq. 27 is the additional e-modification component and k is the e-modification parameter. In addition, the high frequency small oscillation of e and ce will reflect in the fuzzy update rule minute modification. That will cause the fuzzy controller to suppress the small oscillation and induce the system output chattering. Chen and Khalil [26] had proposed a dead-zone scheme to solve the small parameters variation problem of nonlinear systems neural network adaptive control by ignoring the small error within a pre-specified tolerance without parameters update. This filtering strategy is adopted in the fuzzy rule learning mechanism of this new SOFC to smooth the control law and maintain the fuzzy rule learning ability. Then the overall fuzzy rule updating equation is represented as

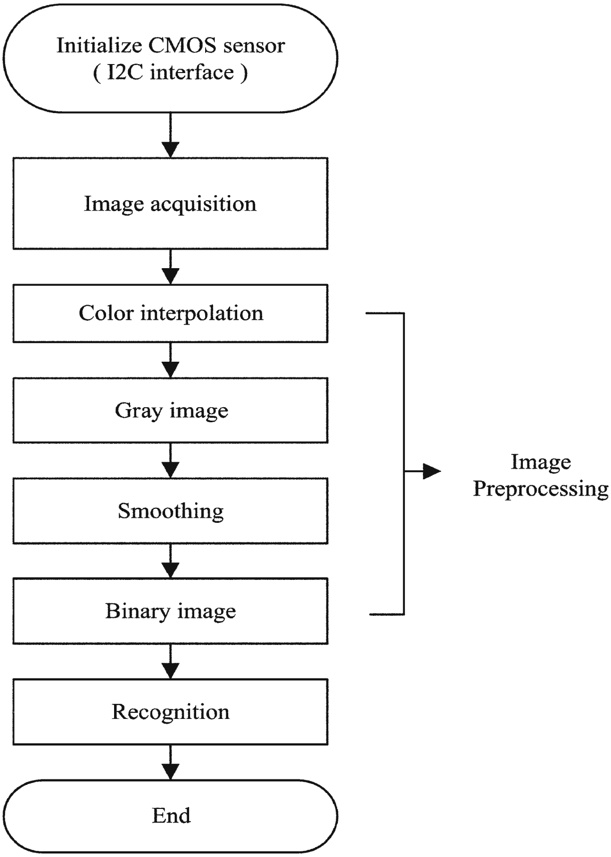
$$\begin{aligned} &if\ e \le d \Rightarrow \Delta u_i = 0 \\ &if\ e > d \Rightarrow \Delta u_i = w_{ei} w_{\mathrm{ce}i} \frac{\gamma}{M}\big[(1-\xi)\, e(nT) + \xi \mathrm{ce}(nT)\big] \\ &\qquad\qquad - k\left|\big[(1-\xi)\, e(nT) + \xi \mathrm{ce}(nT)\big]\right| u_i(nT), \end{aligned} \tag{28}$$

where the positive constants k and d are the e-modification parameter and the specified dead-zone tolerance, respectively. Hence, the proposed new SOFC control strategy has a stable and robust fuzzy rules learning algorithm.

6 Digital Image Processing

The image raw data extracted from CMOS sensor must go through certain processes to obtain useful information. The process flow chart is shown in Fig. 4. The image preprocessing includes Color interpolation to recover RGB colors, image gray operation, smoothing for eliminating noise and binary operations. For the commercial image products, color filter array (CFA) is installed ahead the camera of CMOS sensor to obtain the RGB three different color signals. Each pixel only need one sensor and provides one of the RGB color. Hence, un-sampling two colors should be calculated by using the interpolation operation. The green position on each row is alternate align and the neighboring two rows are interlaced. The red color is located on the odd row and the blue color is located on the even row. Since, the rows number of the red and blue colors are half of the green color, the sampling data can reduce two third. Since green color highly correlates with the brightness, this kind of arrangement has high resolution to the brightness. Here, the bilinear interpolation calculation is employed to recover the color signals.

Fig. 4 Image processing flow chart

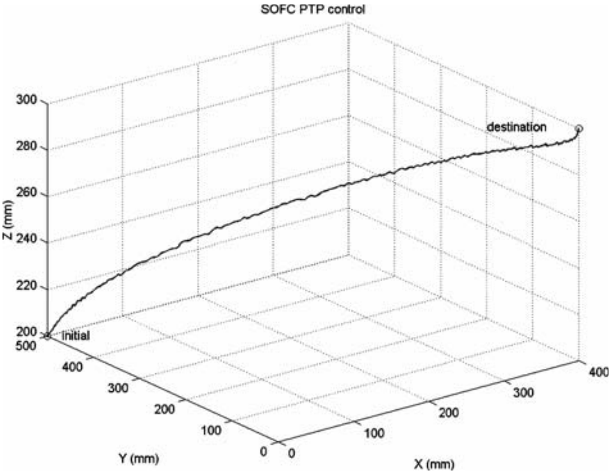


YCbCr color space is widely applied on the digital image color space representation. Human eye has good acuminous with respect to brightness Y and it is described as image gray signal. The brightness signal is depicted as Y and the aberration signals are presented as Cb and Cr. Where Cb is the color difference between blue and reference color (orange–blue axis), and Cr is the color difference between red and reference color (purple–red axis). They can be employed to assist the object identification process. The image pre-processing has transformed the 8 bits image raw data of the pixel into 24 bits RBG color space data. In order to reduce the un-necessary calculation of the Nios II micro-processor, the RGB image data is converted into 8 bits gray signal by the following formula [27]

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.167 & -0.332 & 0.500 \\ 0.500 & -0.419 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{29}$$

Usually, the image raw data extracted from the image sensor has various noise spot or fringe. Image smoothing operation can be employed to eliminate this kind of disturbance. Here, median filter is used to cancel the image noise by selecting the

Fig. 5 PTP motion trajectory in Cartesian space by using SOFC



median luminance value of the pixels gray scale around a target pixel as the gray scale of this target pixel. Then, a histogram equalization can be used to enhance certain subjects by selecting an appropriate threshold value for the gray histograms

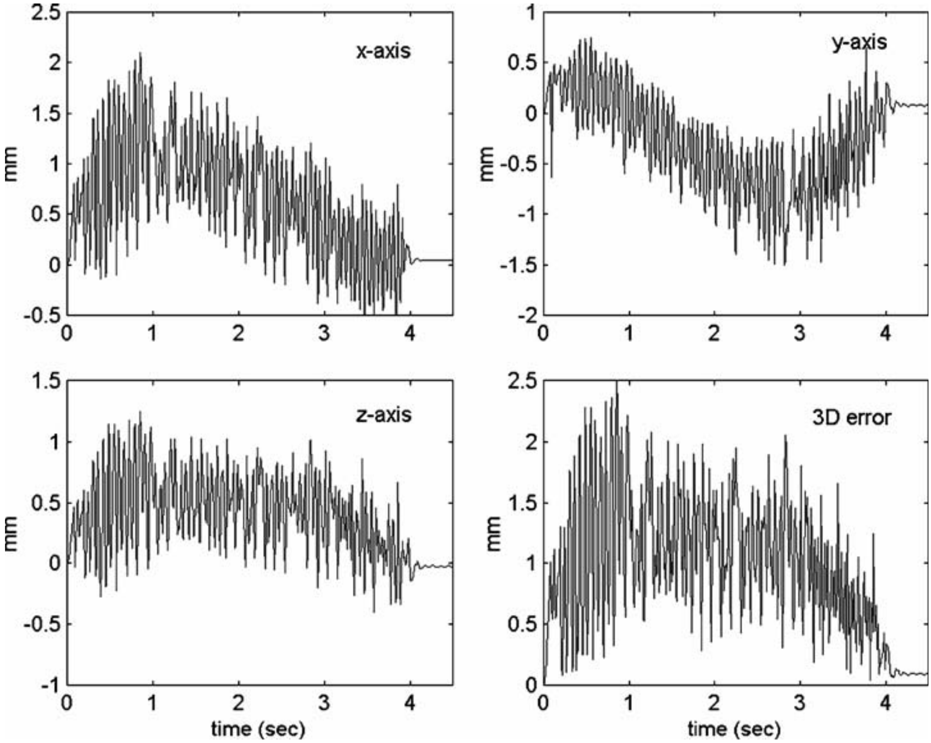


Fig. 6 Position tracking error in each axis and 3D path error

of the image feature to distinguish the object from the background. This threshold value is chosen to convert the analog image brightness signal into binary signal of each pixel. The selection of this threshold value is based on a frequency distribution histogram to distinguish the image and background luma distribution. This image processing scheme is employed to identify the pieces locations on a chessboard of both chess players. Then the FPGA controller will determine the optimal next chess position based on gobang game AI algorithm and competitive strategy, and send to the robotic controller for trajectory planning and control. The robotic motion controller guides the robot to pick up a chess piece and move to the specified location for placing it.

7 Experimental Results

For this SOPC vision servo robotic system, the machine vision is used to provide the end-effector moving target for robot PTP motion or trajectory tracking purpose.

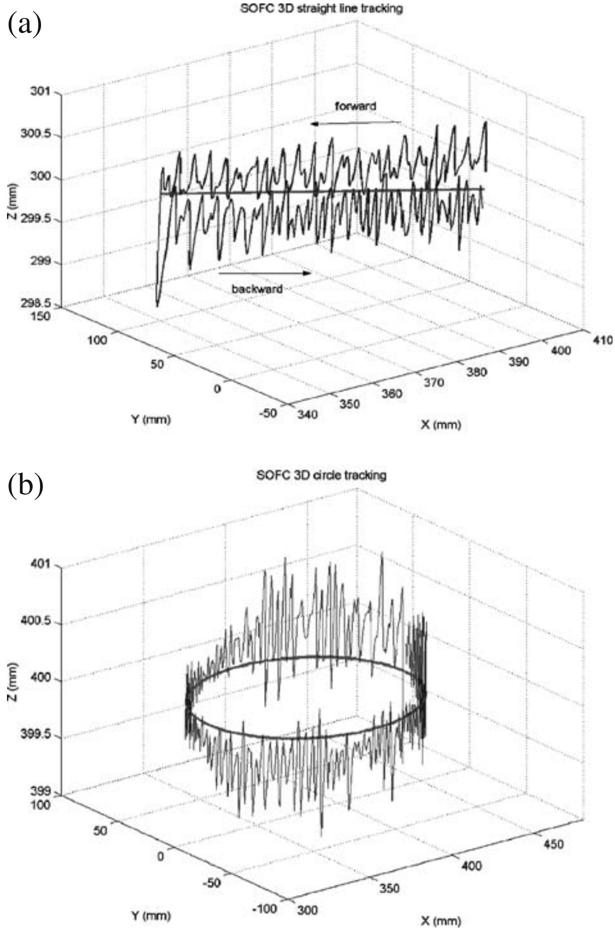


Fig. 7 Position responses of **a** straight line and **b** circle path in Cartesian space

The image is extracted and processed in the SOPC chip and then sent to a PC for evaluating the correctness of the image signal through a UART chip. In order to avoid the collision in the non-autonomous environment, the trajectory planning is required for the robotic motion control, and an appropriate controller is designed to monitor the end-effector motion trajectory. Here, the self-organizing fuzzy controller was employed to control this Mitsubishi RV-M2 5 DOF robotic system described in Section 2. In order to investigate the control performance, the following experiments were performed. The sampling frequency of these experiments was set as 200 Hz. Since the angular output error E and error change CE were divided into 5 fuzzy subsets from −1 to +1 with interval 0.5, the gain parameters ge and gce were selected to regulate the fuzzy input variables into that range. A parameter gu was used to adjust the control input voltage. The adjustable parameters for this SOFC were specified as ge = 200, gce = 100 and gu = 25. The choice of these parameters is not sensitive to controller implementation. If these parameters are varied within 50% and 200% of the original specified values, the control system performance is not changed significantly. The parameter M in the rule modification Eq. 23 was set as 1. The value of the learning rate parameter γ was selected as 0.5. The weighting parameter ξ was chosen as 0.5. The fuzzy control rules of this SOFC algorithm were established by three times of experimental running before executing the following

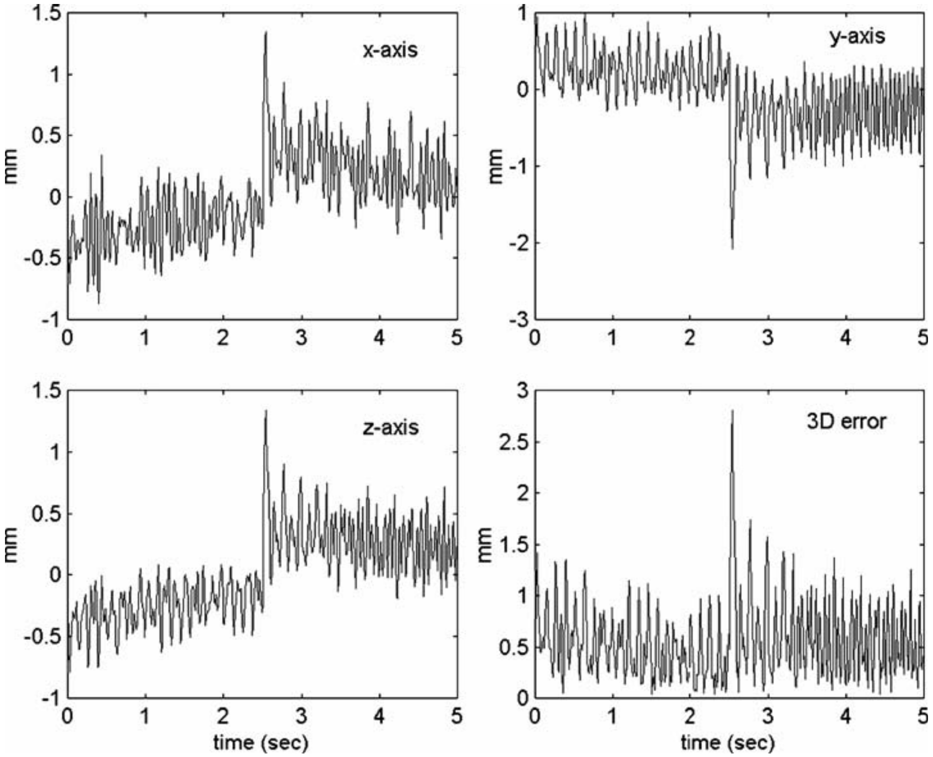


Fig. 8 Position responses of X, Y and Z and contouring error of straight line path

experiments. The e-modification parameter k is selected as 0.05 and the dead-zone size d is set as 0.001, respectively.

Case (A): In order to evaluate the control performance of robotic spot welding and assembly implementations… etc, by using SOFC controller, a PTP motion trajectory of the end-effector was planned first. The planning trajectory for the robotic end-effector is a trapezoid speed curve with a constant acceleration and deceleration for each joint and it is moving from $(0, 500, 200)$ mm to $(400, 0, 300)$ mm in Cartesian space with 4 s total traveling time. The maximum angular acceleration of each joint is limited to $30°/s^2$. The motion trajectory in Cartesian space and the position error in each coordinate axis are shown in Figs. 5 and 6, respectively. The maximum angular tracking error of each joint is less than 0.2°. The overall position trajectory tracking error is less than 2 mm. The destination steady state position error is 0.097 mm.

In addition to the PTP control, one straight line from $(400, 0, 300)$mm to $(350, 100, 300)$ mm and one circle with radius 60 mm and center $(410, 0, 400)$mm in X–Y plane of Cartesian space are specified for trajectory tracking control purpose. The position responses in Cartesian space are shown in Fig. 7a, b, respectively. The maximum angular tracking error of each joint is less than 0.2°. The contouring tracking error

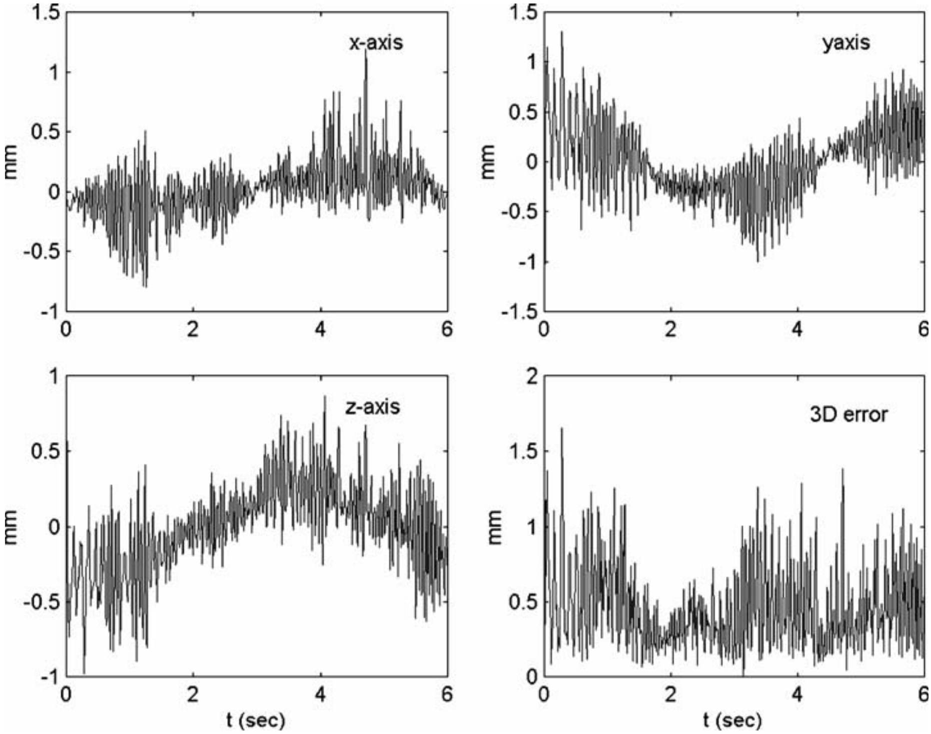


Fig. 9 Position responses of X, Y and Z and contouring error of circle path tracking

in Cartesian space is always less than 1.5 mm and the position tracking error in each coordinate axis is less than 1 mm. The position responses of X, Y and Z axes, and the contouring errors of the straight line and circle tracking are shown in Figs. 8 and 9, respectively.

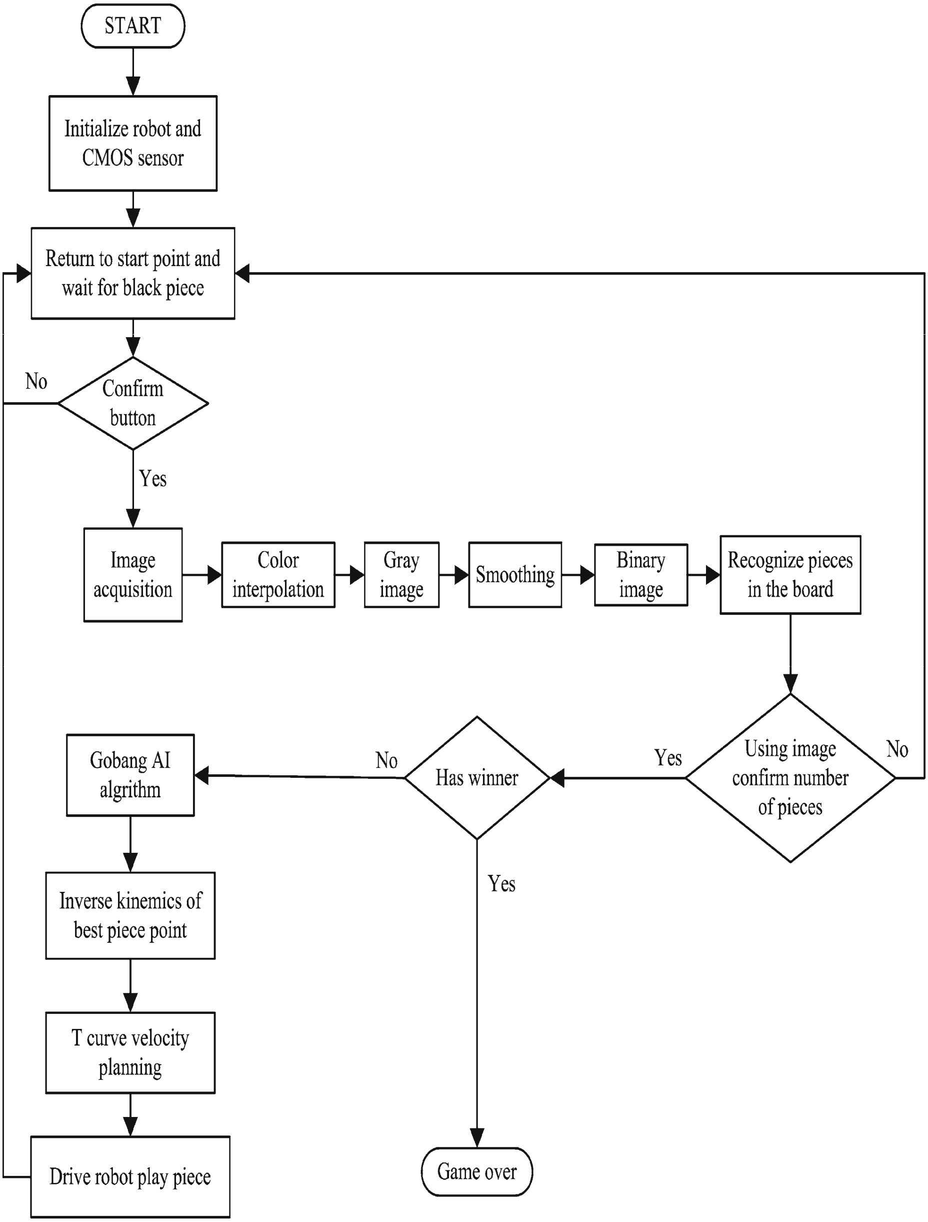

START
Initialize robot and CMOS sensor
Return to start point and wait for black piece
Confirm button
No
Yes
Image acquisition
Color interpolation
Gray image
Smoothing
Binary image
Recognize pieces in the board
Using image confirm number of pieces
No
Yes
Has winner
No
Yes
Gobang AI algrithm
Inverse kinemics of best piece point
T curve velocity planning
Drive robot play piece
Game over

Fig. 10 The flow chart of this non-autonomous robotic gobang game application

Case (B): In order to evaluate the dynamic response of this vision servo robotic motion control system, a gobang game is playing within a 7 × 9 chessboard specified in X–Y plane. Each block has 38 × 34 pixels. If a black chess piece is located in the block, the black pixels number is between 380 and 550. If a white chess piece is located in the block, the black pixels number is between 80 and 150. If no chess piece is located in the block, the black pixels number is between 0 and 10. This information can be used to distinguish the chess pieces locations in chessboard. The

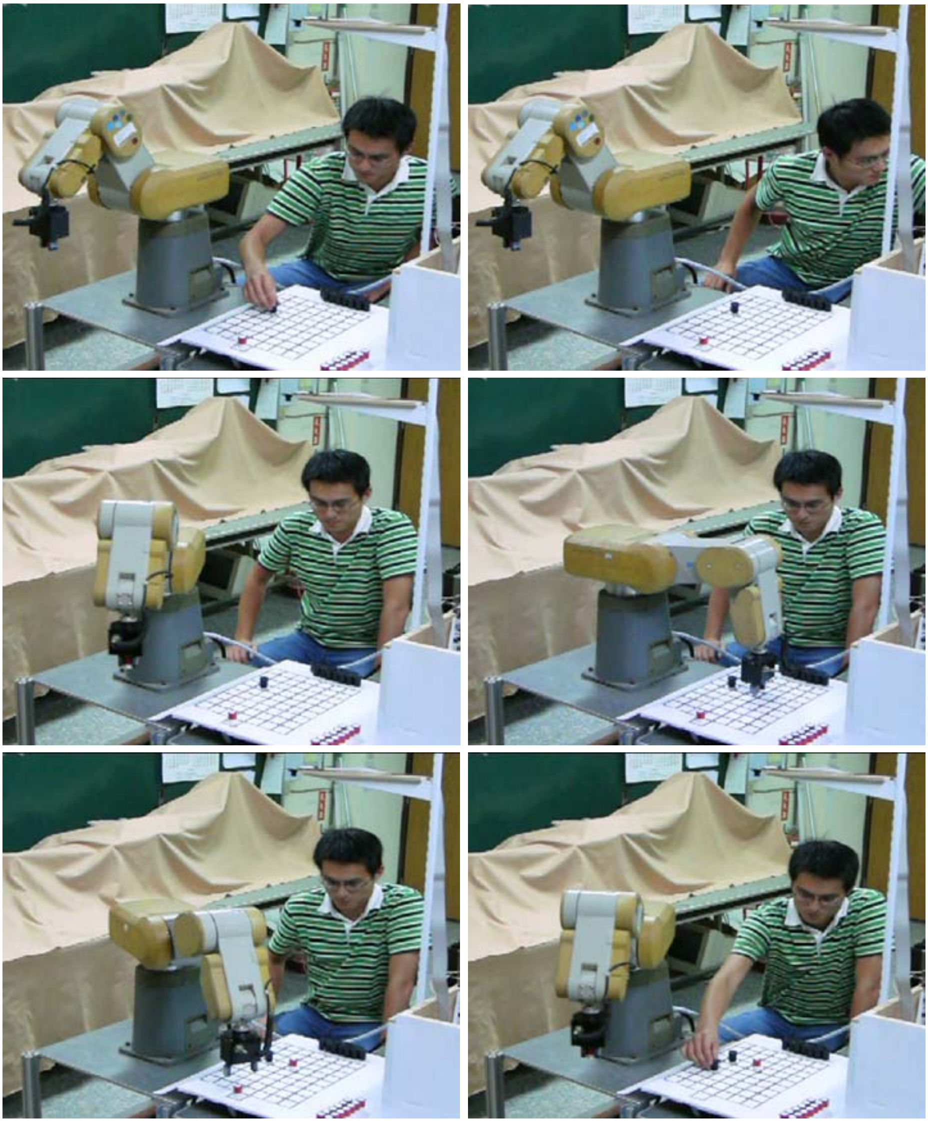


Fig. 11 Robotic gobang game executing pictures

flow chart of this non-autonomous human-robot gobang game is plotted in Fig. 10. Six executing pictures are shown in Fig. 11 with competitor playing turn, end-effector moves to the calculate chess piece position, robot put down the piece, end-effector grasp a piece, robot moving back to the origin and the competitor play again. This vision based robotic control system can run successfully in this gobang game.

8 Conclusions

A visual servo SOPC control structure is implemented on a retrofitted Mitsubishi 5 DOF robot for motion control and non-autonomous gobang game. The self-organizing fuzzy controller was coded inside the FPGA chip for each joint motion control of this robot. This control strategy establishes the appropriate fuzzy rules bank by continuous learning instead of by trial-and-error process to simplify the implementation difficulty of a fuzzy controller. The fuzzy rules table can be initialized as zero. The feature of the proposed SOFC structure is that four rules are modified only by learning for each sampling interval instead of all rules bank. It can reduce the computing time and data base. The experimental results show that this intelligent control system can effectively monitor the robotic end-effector to track various trajectories planned in Cartesian space. The vision guide robotic system can play the gobang game with competitor in the non-autonomous environment. This SOPC control structure can be employed in a flexible non-autonomous environment for executing random assembly or pick-and-place and collision avoidance operations.

References

1. Lin, F.J., Wang, D.H., Huang, P.K.: FPGA-based fuzzy sliding-mode control; for a linear induction motor drive. Proc. IEEE Int. Conf. Electrical Power Appl. **152**(5), 1137–11148 (2005) Sept
2. Kung, Y.S., Shu, G.S.: Development of a FPGA-based motion control IC for robot arm. Proc. IEEE Int. Conf. Ind. Technol., pp. 1397–1402, Hong Kong, 14–17 December 2005
3. Okura, M., Murase, K.: Artificial evolution of FPGA that control a miniature mobile robot Khepera. In: Proceedings of the Autonomous Mini-robots for Research and Edulainmente (AniiRE2003), pp. 103–111 (2003)
4. Tzafestas, S., Dristsas, L.: Combined computed torque and model reference adaptive control of robot system. J. Franklin Inst. **327**(2), 273–294 (1990)
5. Lee, H.-H., Chlick, F.E.: Design of a adaptive control law for robotic manipulator. J. Robot. Syst. **11**(4), 241–255 (1994) June
6. Dessaint, L.-A., Sand, M., Hebert, B.: An adaptive controller for a direct-drive SCARA Robot. IEEE Trans. Ind. Electron. **39**(2), 105–111 (1995) April
7. Huang, S.-J., Lian, R.-J.: A hybrid fuzzy logic and neural network algorithm for robot motion control. IEEE Trans. Ind. Electron. **44**(3), 408–417 (1997)
8. Huang, S.-J., Lee, J.-S.: A stable self-organizing fuzzy controller for robotic motion control. IEEE Trans. Ind. Electron. **47**(2), 421–428 (2000)
9. Procky, T.J., Mamdani, E.H.: A linguistic self-organizing process controller. Automatica **15**, 15–30 (1979)
10. Zhang, B.S., Edmunds, J.M.: Self-organizing fuzzy logic controller. IEE Proceedings-D **139**(5), 460–464 (1992)
11. Shao, S.: Fuzzy self-organizing controller and its application for dynamic processes. Fuzzy Sets Syst. **26**, 151–164 (1988)
12. Wakileh, B.A.M., Gill, K.F.: Robot control using self-organizing fuzzy logic. Comput. Ind. **15**(3), 175–186 (1990)

13. Maeda, M., Murakami, S.: A self-tuning fuzzy controller. Fuzzy Sets Syst. **51**, 29–40 (1992)
14. Wu, Z.Q., Wang, P.Z., Teh, H.H.: A rule self-regulating fuzzy controller. Fuzzy Sets Syst. **47**, 13–21 (1992)
15. Bauchspiess, A., Absi, A., Sadek, C., Dobrzanski, L.A.: Predictive sensor guided robotic manipulators in automated welding cells. J. Mater. Process. Technol. **109**(1–2), 13–19 (2001) 1 February
16. Xiong, Y., Quek, F.: Machine vision for 3D mechanical part recognition in intelligent manufacturing environments. In: Proceedings of the Third International Workshop on Robot Motion and Control, RoMoCo '02, pp. 441–446 (2002)
17. Lin, C.-S., Lue, L.-W.: An image system for fast positioning and accuracy inspection of ball grid array boards. Microelectron. Reliab. **41**(1), 119–128 (2001) January
18. Xiao, N.-F., Nahavandi, S.: Multi-agent model for robotic assembly system. In: Proceedings of the 5th Biannual World Automation Congress, vol. 14, pp. 495–500 (2002)
19. Blasco, J., Aleixos, N., Roger, J.M., Rabatel, G., Molto, E.: Automation and emerging technologies: robotic weed control using machine vision. Biosyst. Eng. **83**(2), 149–157 (2002) October
20. Son, C.: Optimal control planning strategies with fuzzy entropy and sensor fusion for robotic part assembly tasks. Int. J. Mach. Tools Manuf. **42**(12), 1335–1344 (2002) September
21. PAS106BCB283 datasheet, Version 2, Pixart Image Inc. http://www.pixart.com.tw (2002)
22. Wang, L.T., Chen, C.C.: A combined optimization method for solving the inverse kinematics problem of mechanical manipulator. IEEE Trans. Robot. Autom. **7**(4), 489–499 (1991)
23. Kazerounian, K.: On the numerical inverse kinematics of robotic manipulator. AMSEJ of Mechanisms, Transmissions and Automation in Design, vol. 109, pp. 8–13 (1987) March
24. Yang, C.Z.: Design of real-time linguistic self-organizing fuzzy controller. Master thesis, Department of Mechanical Engineering, National Taiwan University (1992)
25. Narendra, K.S., Annaswamy, A.M.: A new adaptive law for robust adaptation without persistent excitation. IEEE Trans. Automat. Contr. **AC-32**(2), 134–145 (1987)
26. Chen, F.C., Khalil, H.K.: Adaptive control of nonlinear systems using neural networks a dead zone approach. In: Proceedings of the 1991 American Control Conference, pp. 667–672 (1991)
27. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing using MATLAB. Pearson, Prentice Hall, Upper Saddle River, NJ (2004)