15

# Design and Implementation of Fuzzy Control for Industrial Robot

Muhammad Suzuri Hitam

## 1. Introduction

The dynamic equations of motion for a mechanical manipulator are highly non-linear and complex. It is therefore, very difficult to implement real-time control based on a detailed dynamic model of a robot, if not impossible (Luh et al., 1980; Lee et al., 1982). The control problem becomes more difficult if adaptive control is necessary to accommodate changing operational conditions. Such a requirement frequently exits in the manufacturing environment; therefore, an alternative design approach would be attractive to the industrial practitioner. A better solution to the complex control problem might result if human intelligence and judgement replaces the design approach of finding an approximation to the true process model. A practical alternative would be the use of fuzzy logic. It has been reported that fuzzy logic controllers performed better, or at least as good as, a conventional controller and can be employed where conventional control techniques are inappropriate (Li et al., 1989; Sugeno, 1985; Ying et al., 1990). In contrast to adaptive control, fuzzy logic algorithms do not require a detailed mathematical description of the process to be controlled and therefore the implementation of fuzzy logic should, theoretically, be less demanding computationally. Fuzzy logic algorithms can be designed for environments where the available source information is not accurate, subjective and of uncertain quality. Furthermore, these algorithms provide a direct means of translating qualitative and imprecise linguistic statements on control procedures into precise computer statements. In this chapter, a proposed fuzzy logic design to control an actual industrial robot arm is outlined. The description of fuzzy logic controller is described in Section 2. It includes the methodology for the design of a fuzzy logic controller for use in robotic application. Section 3 presents the robot control system architecture. In Section 4, the relevant issues that arise relating to the design techniques employed are discussed in detailed. These issues include choise of sampling time, fuzzy rules design strategy, and controller tuning strategy. To evaluate the effectiveness of the proposed design strategy, studies are made to

Source: Industrial-Robotics-Theory-Modelling-Control, ISBN 3-86611-285-8, pp. 964, ARS/plV, Germany, December 2006, Edited by: Sam Cubero

investigate which design strategy leads to the best control performance under various robot conditions. Section 5 concludes this chapter.

## 2. Description of Fuzzy Logic Controller Architecture

The basic structure of the fuzzy logic controller (FLC) most commonly found in the literature is presented in Fig. 1 (Lee, 1990a). The basic configuration of a fuzzy system is composed of a fuzzification interface, a knowledge base, a fuzzy inference machine and a defuzzification interface as illustrated in the upper section of Fig. 1. The measured values of the crisp input variables are mapped into the corresponding linguistic values or the fuzzy set universe of discourse at the fuzzification interface. The knowledge base comprises both the fuzzy data and fuzzy control rules. The fuzzy data base contains all the necessary definitions used in defining the fuzzy sets and linguistic control rules whereas, the fuzzy control rule base includes the necessary control goals and control policy, as defined by an experts, in the form of a set of linguistic rules. The fuzzy inference engine emulates human-decision making skills by employing fuzzy concepts and inferring fuzzy control actions from the rules of inference associated with fuzzy logic. In contrast to the fuzzification stage, the defuzzification interface converts the values of the fuzzy output variables into the corresponding universe of discourse, which yields a non-fuzzy control action from the inferred fuzzy control action.

In general, for a regulation control task, the fuzzy logic controller maps the significant and observable variables to the manipulated variable(s) through the chosen fuzzy relationships. The feedback from the process output is normally returned a crisp input into the fuzzification interface. The crisp or non-fuzzy input disturbance, illustrated in Fig. 1, would normally include both error and change in error, and these are mapped to their fuzzy counterparts at the fuzzification stage. These latter variables are the inputs to the compositional rules of inference from which the fuzzy manipulated variable is obtained. At the output from the defuzzification process, a crisp manipulated variable is available for input to the process. In conclusion, it can be stated that to design a fuzzy logic controller, six essential stages must be completed:

1. Input and output variables to be used must be identified.
2. Design the fuzzification process to receive the chosen input variables.
3. Establish the data and rule bases.
4. Select the compositional rule of inference for decision making.
5. Decide which defuzzification process is to be employed.
6. Develop the computational units to access the data and rule bases.

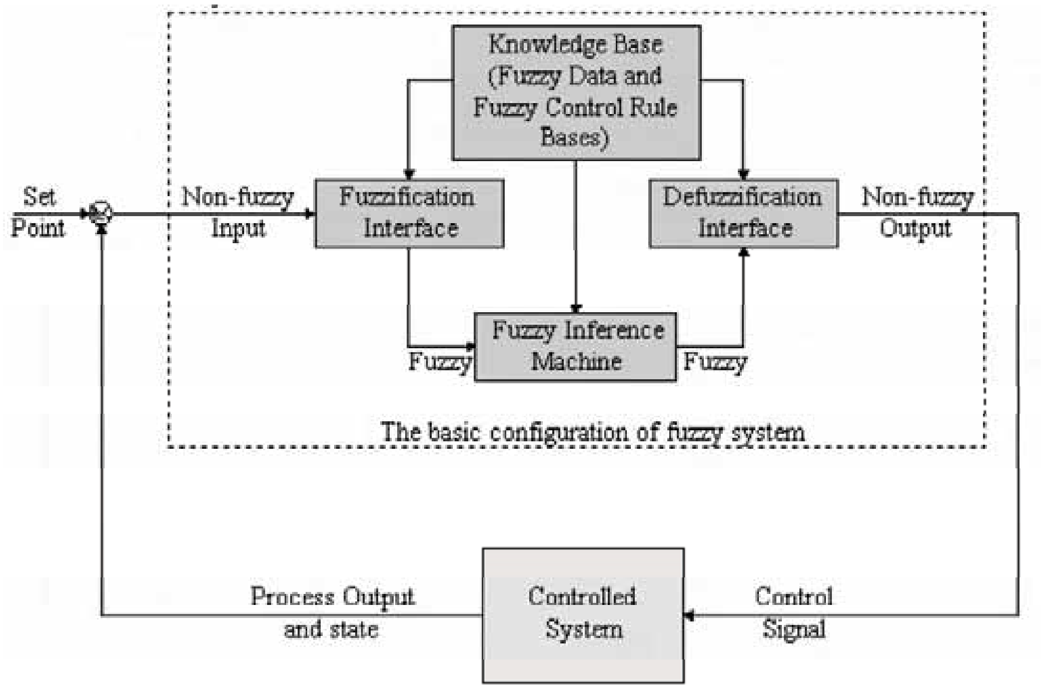


Figure 1. The general form of the fuzzy logic control architecture

### 2.1 Input and Output Variables

In any fuzzy logic control system, the observed input must be fuzzified before it is introduced to the control algorithm. The most commonly used antecedents at this fuzzification stage are the state variables, error and rate of change in error. For the case of positioning a joint within a robot arm, the first variable is the difference (error) between the desired and the current joint position. The value of the second state variable is the numerical difference between two successive values of error (change in error). These two state variables give a good indication of the instantaneous performance of the system and both variables are quantifiable by fuzzy sets. In this project, error (E) and change in error (CE) are defined as the input fuzzy sets and the controlled action (CU) as the output fuzzy set. The evaluation of the error and the change in error at sample interval, k, is calculated as follows :

$$\text{Error}(k) = \text{Demand}(k) - \text{Actual position}(k) \tag{1}$$

$$\text{Change in error}(k) = \text{Error}(k) - \text{Error}(k-1) \tag{2}$$

### 2.2 Method of Representing Fuzzy Sets

According to Lee (1990a), there are two methods for defining a fuzzy set; numerical and functional, depending on whether the universe of discourse is discrete or continuous. In the case of a discrete universe, a numerical definition is employed where the value of the membership function is represented by a vector; the order of the vector dependent on the degree of discretisation. The user has to specifically define the grade of membership of each cardinal in the fuzzy sets. For a continuous universe of discourse, a functional definition can be utilised to define the membership function of a fuzzy set. The triangle, trapezoidal and the bell shaped functions are the popular types found in many engineering applications. In this Chapter, this latter form of representation is adopted. The evaluation of the membership function is evaluated on-line during process operation. A combination of bisected trapezoidal, trapezoidal and triangular shaped fuzzy set templates are used to represent the input and output variables; template shapes that are readily evaluated and require the minimum of computer memory storage. At present, researchers are still looking for the best guidance to determine the best shape for a fuzzy set to provide an optimum solution to a specific control problem. In general, the use of simple shapes could provide satisfactory performance. The geometry of these templates can be defined by the base width and the side slope when mapped to the universe of discourse.

### *2.2.1 Mapping Fuzzy Sets to the Universe of Discourse*

In any application, it is essential for a practitioner to identify the most appropriate parameters prior to the mapping of the fuzzy sets to the chosen universe of discourse; the determination of the size of both the measurement and control spaces; the choice of the discretisation levels for both the measurement and control spaces, the definition of the basic fuzzy sets within these discretised spaces and finally the sample interval to be used. The size of both the measurement and control spaces can be directly determined by estimating the probable operating range of the controlled system. However, the choice of the discretisation levels in both the measurement and control spaces, and the fuzzy set definitions can only be defined subjectively and are normally based on the experience and judgement of the design engineer. From a practical point of view, the number of quantisation levels should be large enough to provide an adequate resolution of the control rules without demanding excessive computer memory storage. Generally 5 to 15 level of discretisations are found to be adequate. It should be emphasised that the choice of these parameters has a significant influence on the quality of the control action that can be achieved in any application (Lee, 1990a). The use of higher resolution in the discretisation levels will result in an increase in the number of control rules and thereby make the formulation of these control rules more difficult. It should also be emphasised that the fuzzy sets selected should always completely cover the whole of the intended working range to ensure that proper control action can be inferred for every state of the process. The union of the support sets on which the primary fuzzy sets are defined should cover the associated universe of discourse in relation to some value, ε. This property is referred to as the "ε-completeness" by Lee (1990a). To ensure a dominant rule always exists, the recommendation is that the value of ε at the crossover point of two overlapping fuzzy sets is 0.5. At this value of ε, two dominant rules will be fired. To define the input fuzzy sets, error (E) and change in error (CE), the following procedure is adopted. In the case of the former fuzzy sets, the maximum range of error for a particular joint actuator is calculated. For example, a robot waist joint with a counter resolution of 0.025 degree per count, and a maximum allowable rotation of 300.0 degree would result in a maximum positional error of 12000 counts. A typical schematic representation for the error fuzzy set universe of discourse would be as illustrated in Fig. 2. The linguistic terms used to describe the fuzzy sets in Fig. 2 are:

{ ***NB, NM, NS, ZE, PS, PM, PB*** }

where N is negative, P is positive, B is big, M is medium, S is small and ZE is zero; a notation that is used throughout this chapter. Combinations of these letters are adopted to represent the fuzzy variables chosen, for example PositiveBig, PositiveMedium and PositiveSmall. As a result, 7 discretisation levels are initially defined for each input and output domain. The size and shape of the fuzzy sets displayed in Fig. 2 are chosen subjectively and tuned during process operation to obtain the most appropriate response. The proposed tuning methodology of these fuzzy sets is detailed later in Figure 4.2.

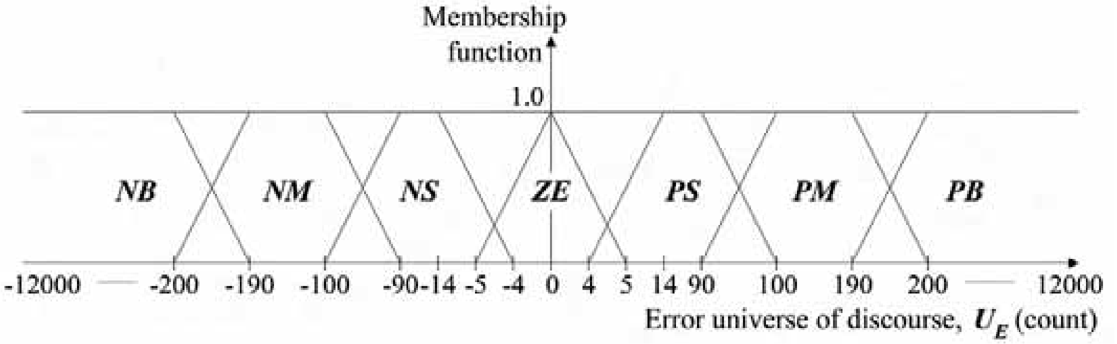


Figure 2. Universe of discourse: error fuzzy sets

To determine the domain size for the change in error variable in this project, an open loop test was conducted. In this test, a whole range of voltage (from the minimum to the maximum) was applied to each of the robot joint actuator and the respective change in angular motion error was recorded every sample interval. From this information, the fuzzy sets illustrated in Fig. 3 for the change in error were initially estimated. Although the open loop response of the system will be different from the close loop response, it will give a good initial guide to the size of the domain appropriate for use with the fuzzy logic controller.

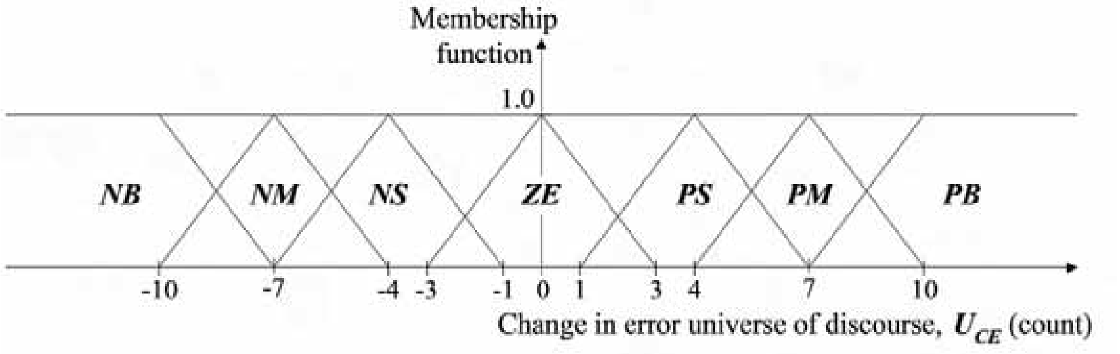


Figure 3. Change in error fuzzy sets domain of discourse

It should be noted that the choice of sampling interval is very important because it will affect the maximum change in error value recorded. It was found that the use of a very high sampling rate caused the recorded maximum change in angular motion error to be close to zero and this made it impossible to define the location of each fuzzy set in the domain of discourse. For example, a sampling period of 0.001 seconds will result in a maximum change in waist positional error of 2 counts; a value found experimentally. In a similar manner, the control variable output fuzzy sets were selected. However, in this particular case, the dimentionality of the space is determined by the resolution of the available D/A converters. The D/A converters adopted are of an 8-bit type which yield 256 resolution levels as indicated on the horizontal axis in Fig. 4(a). Again, the universe of discourse was partitioned into 7 fuzzy set zones as depicted in Fig. 4(b).

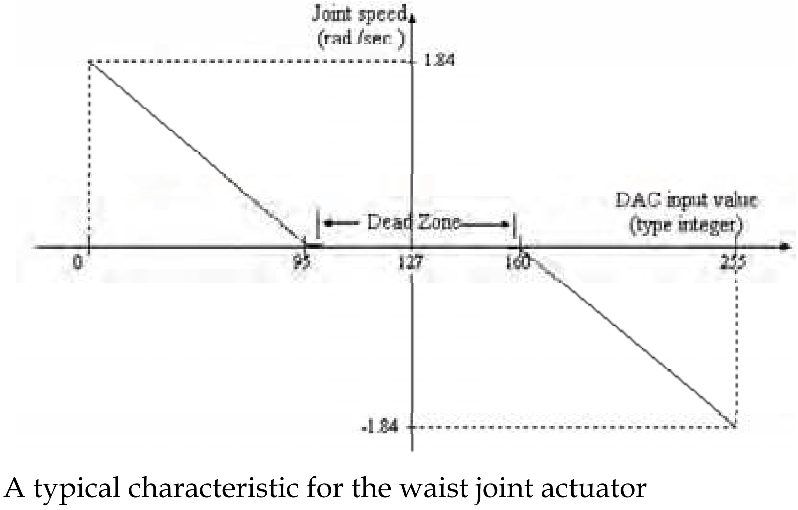


Figure 4(a). A typical characteristic for the waist joint actuator

It should be noted that the fuzzy set labelled Zero is defined across the dead zone of the dc-servo motor in order to compensate for the static characteristics of the motor in this region. The initial sizes and distribution of the fuzzy sets are tuned during operation to improve the closed loop performance of the system.

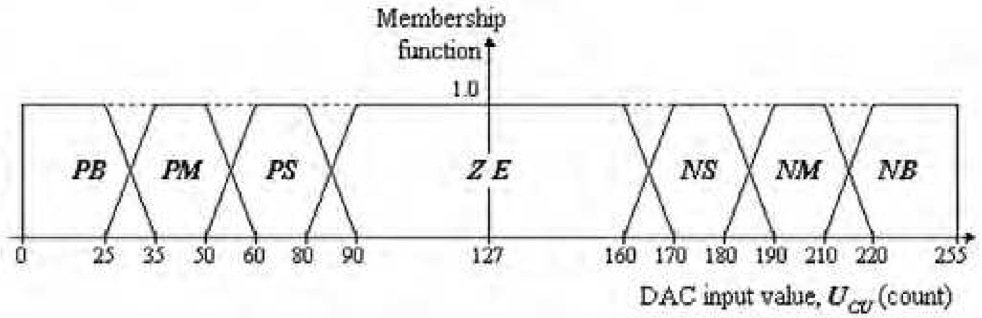


Figure 4(b). Control action domain of discourse

#### 2.2.1.1 Transforming a Crisp Input to a Fuzzy Variable

Consider the trapezoidal representation of an error fuzzy set as illustrated in Fig. 5. Let an input error at sample interval $k$ be $e(k) \in U_E$ and the corresponding membership grade of the fuzzy set $E_i \subset U_E$ be defined by the template [a, b, c, d]. Therefore, its membership ership function, $\mu_{E_i}$ can be directly evaluated using the expression:

$$\left.\begin{array}{l} \textbf{IF } e(k) \notin (a, d) \textbf{ THEN } \mu_{E_i}(e(k)) = 0.0 \\ \textbf{IF } e(k) \in [a, b) \textbf{ THEN } \mu_{E_i}(e(k)) = (e(k) - a)\, slope.ab \\ \textbf{IF } e(k) \in [b, c] \textbf{ THEN } \mu_{E_i}(e(k)) = 1.0 \\ \textbf{IF } e(k) \in (c, d] \textbf{ THEN } \mu_{E_i}(e(k)) = (d - e(k))\, slope.cd \end{array}\right\} \quad (3)$$

where the gradients *slope.ab* and *slope.cd* are calculated from the expressions;

$$slope.ab = \frac{1.0}{b - a} \quad (4)$$

$$slope.ab = \frac{1.0}{d - c}. \quad (5)$$

In a similar manner the properties of a triangular or bisected trapezoidal fuzzy set template can be defined.

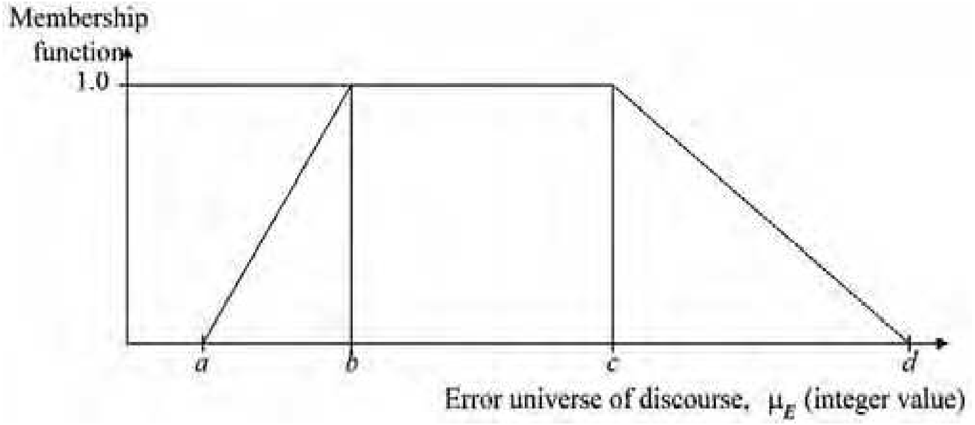
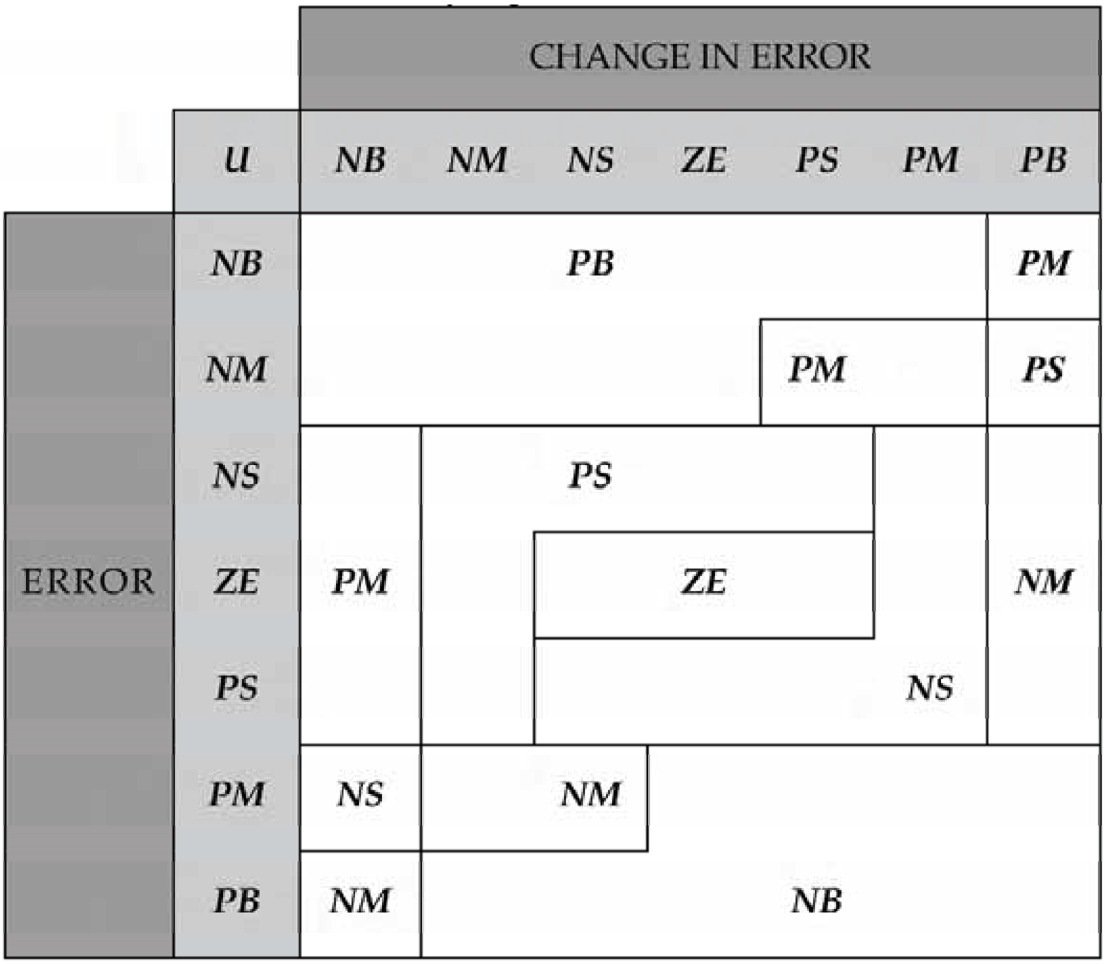


Figure 5. Trapezoidal representation of an error fuzzy set.

## 2.3 Defining the Fuzzy Rule Base

The fuzzy rule base employed in FLC contains fuzzy conditional statements which are currently chosen by the practitioner from a detailed knowledge of the operational characteristics of the process to be controlled. The fuzzy rule base can be derived by adopting a combination of four practical approaches which are mutually exclusive, but are the most likely to provide an effective rule base. These can be summarised as follows (Lee, 1990a): 1. Expert experience and control engineering knowledge. In nature, most human decision making are based on linguistic rather than numerical descriptions. From this point of view, fuzzy control rules provide a natural framework for the characterisation of human behaviour and decision making by the adoption of fuzzy conditional statements and the use of an inference mechanism. 2. Operational experience. The process performance that can be achieved by a human operator when controlling a complex process is remarkable because his reactions are mostly instinctive. An operator through the use of conscious or subconscious conditional statements derives an effective control strategy. These rules can be deduced from observations of the actions of the human controller in terms of the input and output operating data. 3. Fuzzy model of the process. The linguistic description of the dynamic characteristics of a controlled process may be viewed as a fuzzy model of a process. Based on this fuzzy model, a set of fuzzy control rules can be generated to attain an optimal performance from a dynamic system. 4. Learning. Emulation of human learning ability can be carried out through the automatic generation and modification of the fuzzy control rules from experience gained. The rule base strategy adopted in this work is developed from operational and engineering knowledge. The initial control rule base adopted is displayed in the look-up table, Table 1. This table should be read as:

$$\textbf{IF}\ \ (\textit{error}\ \text{is}\ \textbf{\textit{NegativeMedium}})\ \ \textbf{AND}\ \ (\textit{change in error}\ \text{is}\ \textbf{\textit{Zero}})$$
$$\textbf{THEN}\ \ (\textit{control action}\ \text{is}\ \textbf{\textit{PositiveBig}}) \tag{6}$$



| | | CHANGE IN ERROR | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | *u* | *NB* | *NM* | *NS* | *ZE* | *PS* | *PM* | *PB* |
| ERROR | *NB* | *PB* | *PB* | *PB* | *PB* | *PB* | *PB* | *PM* |
| | *NM* | *PB* | *PB* | *PB* | *PB* | *PM* | *PM* | *PS* |
| | *NS* | *PM* | *PS* | *PS* | *PS* | *PS* | *NS* | *NM* |
| | *ZE* | *PM* | *PS* | *ZE* | *ZE* | *ZE* | *NS* | *NM* |
| | *PS* | *PM* | *PS* | *NS* | *NS* | *NS* | *NS* | *NM* |
| | *PM* | *NS* | *NM* | *NM* | *NB* | *NB* | *NB* | *NB* |
| | *PB* | *NM* | *NB* | *NB* | *NB* | *NB* | *NB* | *NB* |

Table 1. Initial rules selected for fuzzy logic controller

## 2.4 Fuzzy Inference Mechanism

One virtue of a fuzzy system is its inference mechanisms which is analogous to the human decision making process. The inference mechanism employs the fuzzy control rules to infer the fuzzy sets on the universe of possible control action. The mechanism acts as a rule processor and carries out the tasks of manoeuvring the primary fuzzy sets and their attendant operations, evaluating the fuzzy conditional statements and searching for appropriate rules to form the output action. As mention earlier, the input and output variables of error, change in error and control action, $\boldsymbol{U_E}$ , $\boldsymbol{U_{CE}}$ and $\boldsymbol{U_{CU}}$. respectively, are all chosen to be discrete and finite, and are in the form of;

$$E \subset U_E \ , \ CE \subset U_{CE} \ \text{ and } CU \subset U_{CU} \tag{7}$$

where  indicates a fuzzy subset. As a result of selecting 7 discretisation levels for each fuzzy input and output variable, i.e. ***PB***, ***PM***, ***PS***, etc., 49 fuzzy control rules result. These control rules are expressed in the form of fuzzy conditional statements;

$$\begin{aligned}&\textbf{IF } (\textit{error is } \boldsymbol{E}) \ \textbf{ AND } (\textit{change in error is } \boldsymbol{CE}) \\ &\textbf{THEN } (\textit{control action is } \boldsymbol{CU})\end{aligned} \tag{8}$$

At sample interval *k*, the *j*th fuzzy control rule, equation (8), can be expressed as;

$$\textbf{IF } (e(k) \text{ is } \boldsymbol{E_j}) \textbf{ AND } (ce(k) \text{ is } \boldsymbol{CE_j}) \textbf{ THEN } (cu(k) \text{ is } \boldsymbol{CU_j}) \tag{9}$$

where *e*(*k*), *ce*(*k*) and *cu*(*k*) denote the error, change in error and manipulated control variable respectively. The *j*th fuzzy subsets $\boldsymbol{E_j}$ , $\boldsymbol{CE_j}$ and $\boldsymbol{CU_j}$ are defined as;

$$\boldsymbol{E_j} = \{ (e \ , \mu_E(e)) \}, \boldsymbol{CE_j} = \{ (ce \ , \mu_{CE}(ce)) \}, \boldsymbol{CU_j} = \{ (cu \ , \mu_{CU}(cu)) \} \tag{10}$$

Alternatively, Equation (9) can be evaluated through the use of the compositional rule of inference. If the minimum operator is utilised, the resulting membership function can be expressed as;

$$\begin{aligned}\mu_{\Re_j}(e(k), ce(k), cu(k)) &= \left[\mu_{E_j}(e(k)) \textbf{ AND } \mu_{CE_j}(ce(k))\right] \Rightarrow \mu_{CU_j}(cu(k)) \\ &= \boldsymbol{min}\left[\mu_{E_j}(e(k)) \ , \ \mu_{CE_j}(ce(k)) \ , \ \mu_{CU_j}(cu(k))\right]\end{aligned} \tag{11}$$

where the symbol □ indicates the fuzzy implication function and $\Re_j = E_j \times CE_j \times CU_j$ denotes the fuzzy relation matrix on

$$\Re = \bigcup_{j=1}^{49} \Re_j = \underset{j=1}{\overset{49}{\boldsymbol{max}}} \Re_j \tag{12}$$

In term of the membership functions, this can be expressed as;

$$\mu_{\Re_j}(e, ce, cu) = \underset{j=1}{\overset{49}{\boldsymbol{max}}}\left\{ \boldsymbol{min}\left[\mu_{E_j}(e), \mu_{CE_j}(ce) \ , \mu_{CU_j}(cu)\right] \right\} \tag{13}$$

To use the result of Equation (13), a defuzzification process is necessary to produce a crisp output for the control action value.

### 2.5 Choosing Appropriate Defuzzification Method

Several approaches (Lee, 1990b) have been proposed to map the fuzzy control action to a crisp value for input to the process. Basically, all have the same aim that is, how best to represent the distribution of an inferred fuzzy control action as a crisp value. The defuzzification strategies most frequently found in the literature are the maximum method and centre of area method:

1. The maximum method. Generally, the maximum method relies on finding the domain value, $z_0$, that maximises the membership grade which can be represented by;

$$z_o = \max_{z \in W} \mu_{C_T'}(z) \tag{14}$$

In the case when there is more than one maximum membership grade in ***W***, the value of $z_0$ is determined by averaging all local maxima in ***W***. This approach known as mean of maximum method (MOM) is expressed as;

$$z_o = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{15}$$

where $z_i = \max_{z \in w} \mu_{c_T}(z)$ $z_i = \max_{z \in w} \mu_{c_T}(z)$ and $n$ is the number of times the membership function reaches the maximum support value.

2. The center of area method (COA). The center of area method sometimes called the centroid method produces the center of gravity of the possibility distribution of a control action. This technique finds the balance point in the output domain of the universe of discourse. In the case when a discrete universe of discourse with m quantisation levels in the output, the COA method produces;

$$z_o = \frac{\sum_{i=1}^{m} \mu_{C_T'}(z_i)\, z_i}{\sum_{i=1}^{m} \mu_{C_T'}(z_i)} \tag{16}$$

where *zi* is the *i*th domain value with membership grade of µ (*zi* ).

## 3. Experimental Setup

The robot control system is composed of the host computer, the transputer network, and the interface system to a small industrial robot. The schematic representation of the control structure is presented in Fig. 6.

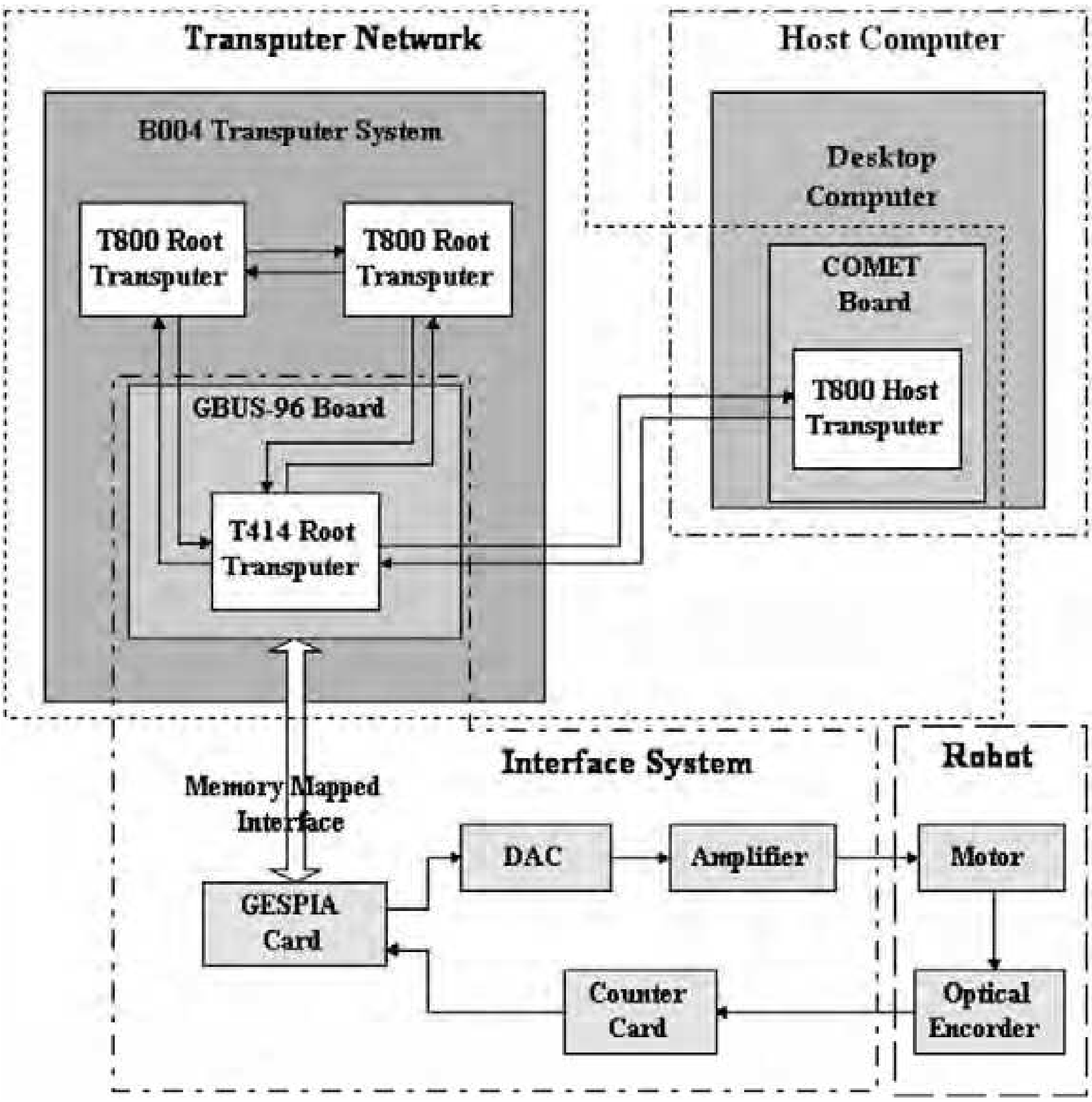


Figure 6. Schematic representation of robot control architecture.

The controller structure is hierarchically arranged. At the top level of the system hierarchy is a desktop computer which has a supervisory role for supporting the transputer network and providing the necessary user interface and disc storage facilities. The Transputer Development System acts as an operating system with Occam II as the programming language. At the lower level are the INMOS transputers; in this application one T800 host transputer is resident on the COMET board and mounted in one of the expansion slots of the desktop computer with the remaining three transputers resident in a SENSION B004 system. The robot is the RM-501 Mitsubishi Move Master II, with the proprietary control unit removed to allow direct access to the joint actuators, optical encoders and joint boundary detection switches. The host transputer also provides an interface facilities to the user, for example, input and output operation from the keyboard to the screen. The three transputers resident in the SENSION B004 system are a T414 transputer which is resident on the GBUS-96 board and provides a memory mapped interface to the robot through a Peripheral Interface Adapter (PIA) Card. The remaining two T800 root transputers are used to execute the controller code to the robot. The PIA card allows a parallel input and output interface to the robot joint actuators and conforms to the interface protocol implemented on the GBUS-96 which is known as a GESPIA Card. The actual hardware arrangement together with the interfacing employed is shown in  Fig. 7, with the Mitsubishi RM-501 robot shown in Fig. 8.

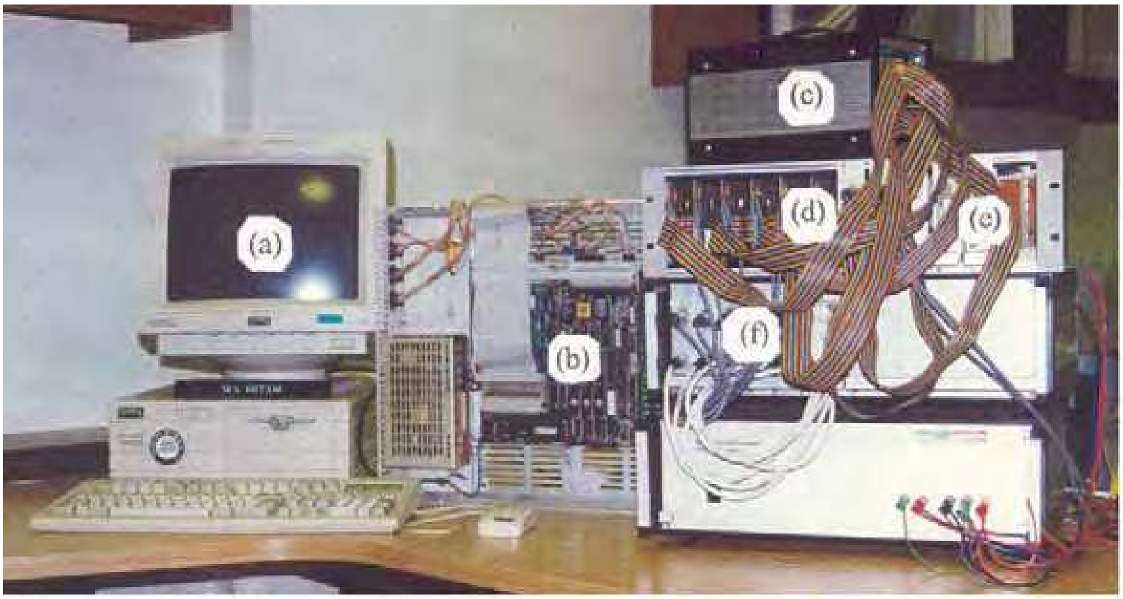


Figure 7.  System hardware and interfacing. (a) host computer, (b) B004 transputer system, (c) GESPIA card, (d) DAC cards, (e) counter cards and (f) power amplifier.

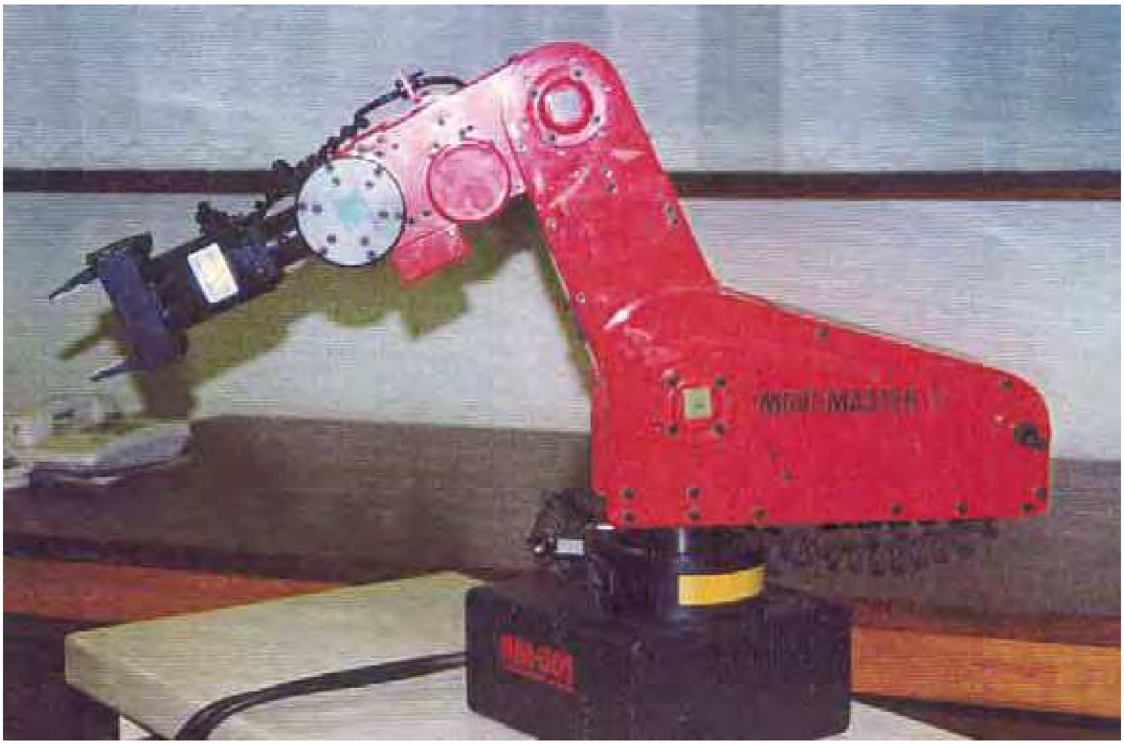


Figure 8. The Mitsubishi RM-501 Move Master II Industrial Robot.

### 3.1 The Mitsubishi RM-501 Move Master II Robot

This industrial robot is a five degree of freedom robot with a vertical multi-joint configuration. The robot actuators are all direct current servo motors, but of different powers. At the end of each joint, a sensor is provided to limit the angular movement. The length of link and its associated maximum angular motion is listed in Table 2. Fig. 9(a) and 9(b) illustrate the details of the robot dimensions1 and it's working envelop. The maximum permissible handling weight capacity is 1.2 kg including the weight of the end effector. Table 2 The Mitsubishi RM-501 Move Master II geometry.

| **Join** | **Link Length (mm)** | **Maximum Rotation (Degree)** |
|---|---|---|
| Waist | 250 | 300 |
| Shoulder | 220 | 130 |
| Elbow | 160 | 90 |
| Wrist roll | 65 | +-90 |
| Wrist pitch | 65 | +-180 |

Table 2. The Mitsubishi RM-501 Move Master II geometry

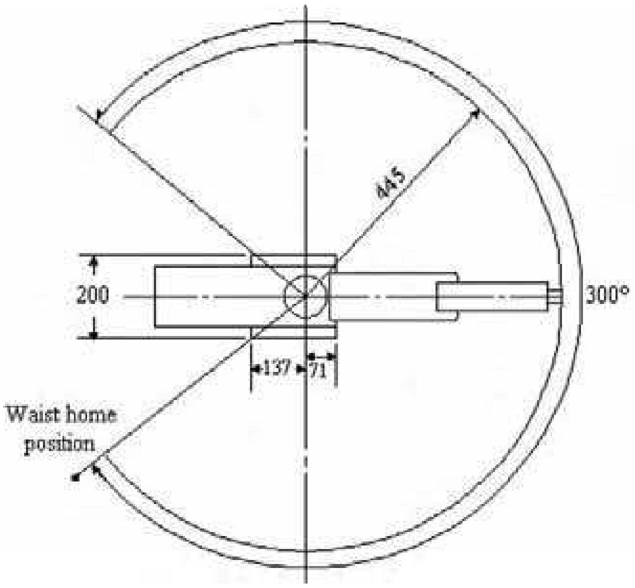


Figure 9(a). Range of movement of waist joint and robot dimensions (all dimensions are measured in millimeter).

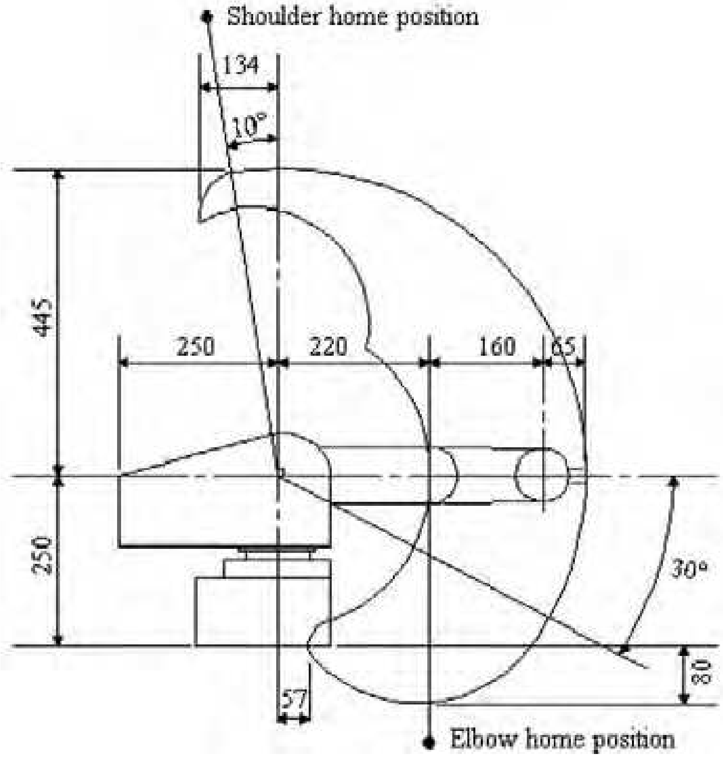


Figure 9(b). Robot dimension and range of movement when hand is not attached.

## 4. Experimental Studies

A program code for the development of a FLC was written in the Occam language and executed in a transputer environment. This approach would enable the evaluation of the robustness of the controller design proposed and applied to the first three joint of a RM-501 Mitsubishi industrial robot. A T800 transputer is assigned to position each joint of the robot independently. To determine the effect on controller performance of changing different controller parameters, one joint only is actuated and the other two are locked. In the first experiment the impact on overall robot performance of changes in sample interval was assessed. This was followed by an investigation into how best to tune a controller algorithm and whether guide-lines can be identified for future use. The problem is to overcome the effect of changing robot arm configuration together with a varying payload condition.

### 4.1 The Choice of Sampling Time

Inputs (error and change in error) to the fuzzy logic control algorithm that have zero membership grades will cause the membership grades of the output fuzzy sets to be zero. For each sample period, the on-line evaluation of the algorithm with 49 control rules has been found by experiment to be 0.4 milliseconds or less. Hence, to shorten the run time, only inputs with non-zero membership grades are evaluated. For times of this magnitude, real-time control is possible for the three major joint controllers proposed. It has been cited in the literature that it is appropriate to use a 0.016 seconds sampling period (60 Hertz) because of its general availability and because the mechanical resonant frequency of most manipulators is around 5 to 10 Hz (Fu et al., 1987). Experiments have been carried out to determine how much improvement can be achieved by shorten the sampling period from 0.02 seconds to 0.01 seconds. In the first experiment, the waist joint is subjected to a 60.0 degree (1.047 radian or 2400 counter count) step disturbance with all other joints in a temporary state of rest. The results shown in Fig. 10 suggest that very little improvement in transient behaviour will be achieved by employing the shorter sampling period. The only benefit gained is a reduction in the time to reach the steady state of 0.4 seconds. In a second test, the waist joint is commanded to start from its zero position and to reach a position of 5 degree (0.0087 radian or 20 counter count) in 2 seconds; it remains at this position for an interval of 1 second after which it is required to return to its home position in 2 seconds as showed in Fig. 11. Again the benefit is only very marginal and of no significance for most industrial applications. Despite these results, it was decided that the higher of the two sampling rates would generally ensure better transient behaviour, hence the 0.01 seconds sampling period is used throughout this project.

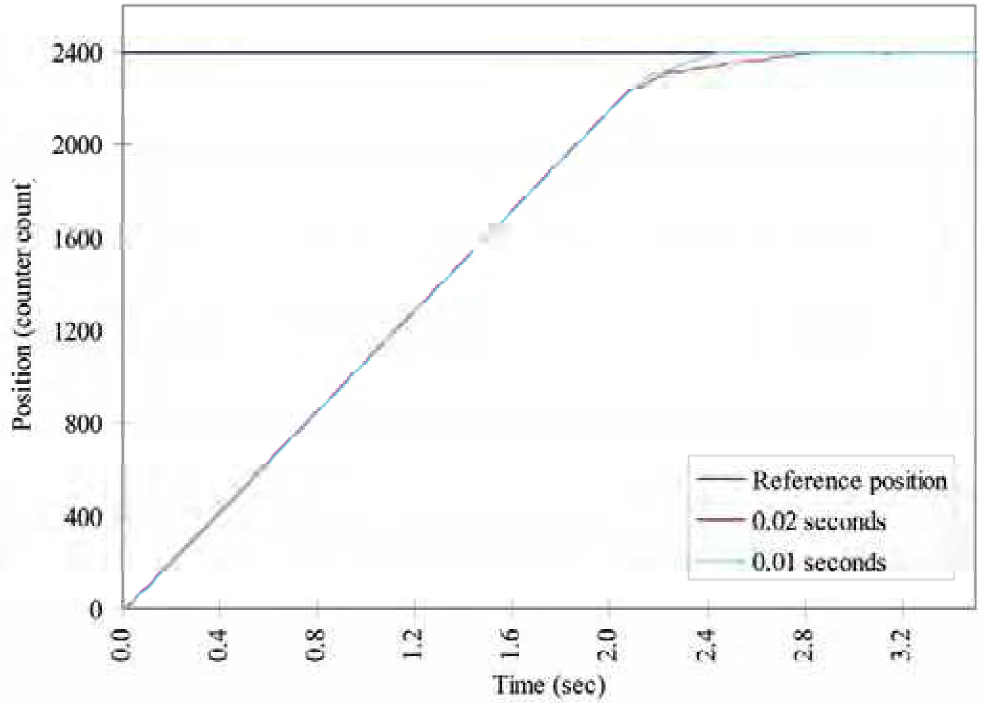


Figure 10. Waist response to a step input for different sampling periods.

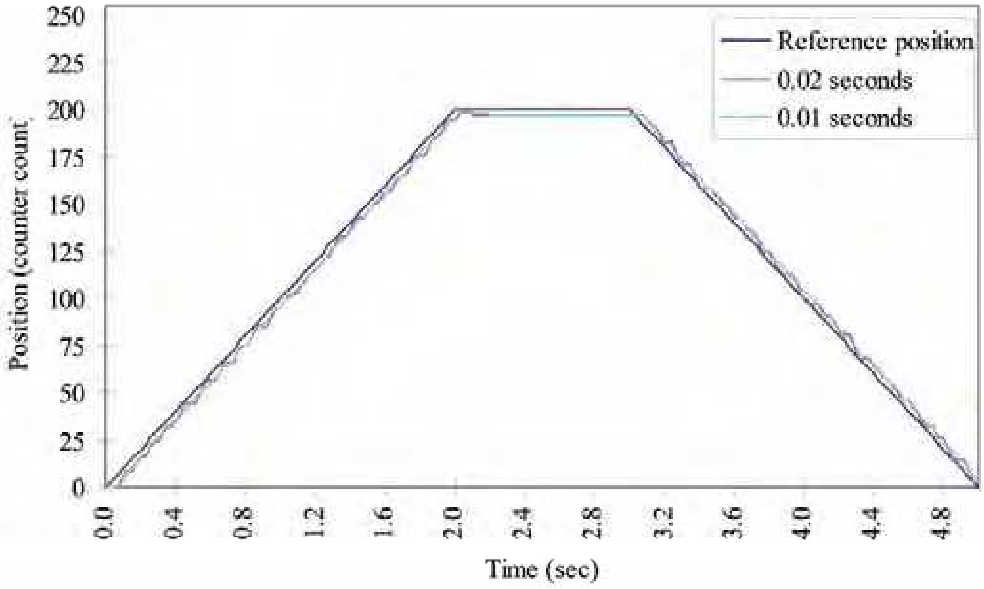


Figure 11. Waist trajectory tracking at different sampling periods.