Computer-aided learning of controller design: focus on fuzzy logic control

C. H. Lo, Y. K. Wong and A. B. Rad
Department of Electrical Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong
E-mail: eeykwong@polyu.edu.hk

Abstract A computer-aided controller design package is developed in this paper. The package provides a simulated environment for simulating the action of a controller under different parameter settings in order to achieve optimal system performance. The designed controller is then applied to a process plant for on-line control.

Keywords fuzzy control; process control; self-tuning control

Since fuzzy logic controllers (FLC) can cover a wider operating range, they are more robust than PID controllers.[1] Even without the derivation of a mathematical model, the FLC is suitable for controlling processes that are non-linear and ill-defined. However, the FLC has many parameters and its control performance depends on the tuning of these parameters. The tunable parameters are:

1 the scaling factors of the input and output variables;
2 the membership functions of the variables; and
3 the set of fuzzy control rules.

The tuning process of these parameters is non-trivial and can be time consuming since it is often done manually on a trial-and-error basis. To resolve these difficulties, a self-tuning fuzzy logic controller (STFC) is suggested to tune the input scaling factors of error and change of error (K_e and K_{de}) automatically. Depending on the system output response (such as overshoot, reaching time and amplitude of oscillation), the STFC can then adjust the amount of increase or decrease to the input scaling factors according to a set of fuzzy 'meta-rules' that is defined in advance. The process is repeated until a desired response is obtained.

A computer-aided controller design package is developed for simulating different controllers under various controller parameter settings. The package includes three kinds of controller: PID, FLC and STFC. A transfer function up to the *nth* order can be used as a system model to verify the action of the controller. Controllers with different settings can be compared easily in the package. The package is focused on fuzzy logic control because it will give students more chance to learn different kinds of controller besides the conventional PID controller. The package allows students to tune the input and output scaling factors and the membership functions of the variables since these two parameters are easier for them to handle. Basic fuzzy logic control theory is also provided in the package as a short tutorial. Moreover, different defuzzification methods, such as centre of area, mean of maximum, and various inference methods such as max-min, max-product, are also implemented.

After simulation, the designed controller is used to control the PT326 Process Trainer on-line. The PT326 Process Trainer is a heater plant with a centrifugal blower, a tube, and a heater grid. Heated air is blown by the centrifugal blower, through the length of tubing to the atmosphere. A sensor at the end of the tube measures the temperature of the heated air and then compares it with the set temperature. Control action is generated to bring the temperature of the heated air to the desired temperature. To supplement the traditional teaching methodology, a computer-aided instruction package is developed and installed in a P2 computer as a tool for controller design. The package developed is also being incorporated into laboratory exercises. Efforts have been made to integrate lecture material, laboratory practice and computer package into a coherent teaching program.

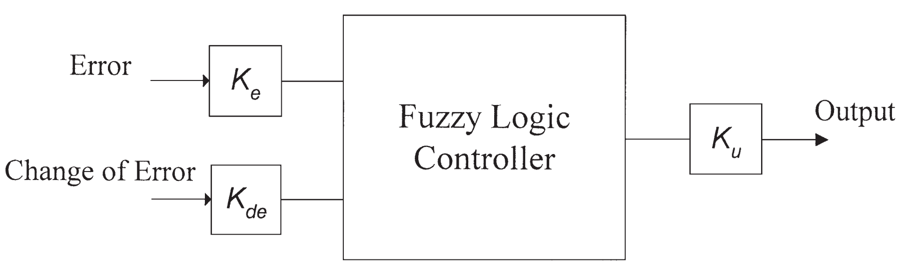
The rest of the paper is organized as follows: the STFC algorithm will be presented in the next section, followed by a simulation study, experimental results and a conclusion.

Self-tuning fuzzy logic controller (STFC)

In this section, the self-tuning fuzzy logic control algorithm is discussed. The fuzzy controller to be tuned has two control inputs: the error (e) and the change of error (Δe). The control action generated by the controller is the change, Δu, in the manipulated variable u. The self-tuning method tunes the input scaling factors K_e and K_{de} of the fuzzy logic controller (Fig. 1).

The input scaling factors are chosen to be the tuning parameters because they have the greatest influence on the basic sensitivity of the controller with respect to optimal choice of the operating areas of the input signal. The tuning of the two scaling factors, K_e and K_{de}, for the two control inputs, e and Δe, respectively, is done automatically by a set of fuzzy 'meta-rules'.

It should be noted that this tuning technique does not tune the rule base of the fuzzy logic controller. Assuming a working set of rules, the controller is tuned for



where; K_e, K_{de}: input scaling factors; K_u: output scaling factor.

Fig. 1 *Input and output scaling factors.*

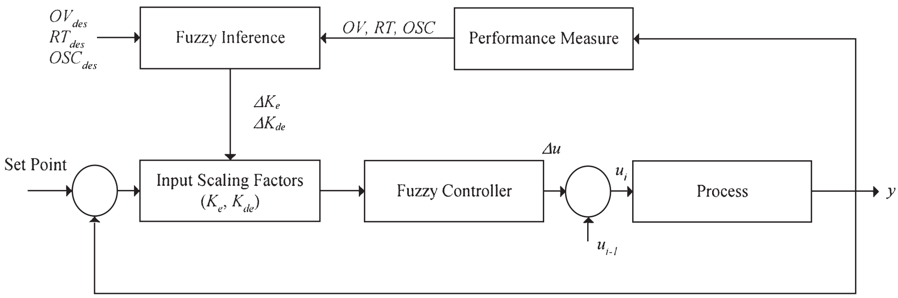


Fig. 2 *Schematic diagram of the tuning procedure.*

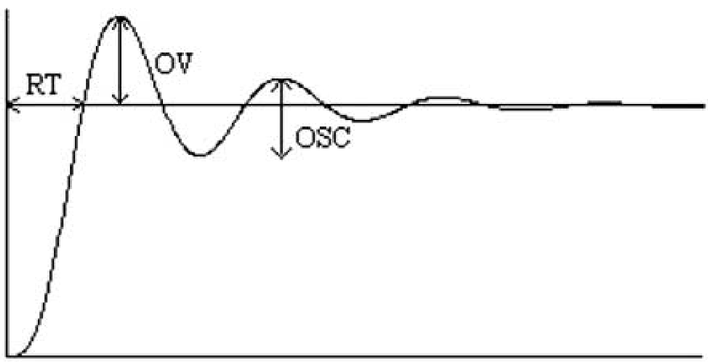


Fig. 3 *Step response – OV, RT and OSC.*

a desired response.[2] The algorithm consists of a performance measure, rule-base and an inference step as shown in Fig. 2.

Performance measure

When the control ends, the learning function of the STFC adjusts the scaling factors by evaluating the control result.[3,4] It evaluates the output response, such as overshoot (*OV*), reaching time (*RT*) and amplitude of oscillation (*OSC*). Fig. 3 shows the *OV*, *RT* and *OSC* of a typical step response.

The performance measures are obtained with respect to specified desirable values of overshoot percentage ($OV_{desired}$), reaching time ($RT_{desired}$) and percentage amplitude of oscillation ($OSC_{desired}$). Thus the actual performance measures used are:

$$\left.\begin{aligned} \Delta OV &= OV - OV_{desired} \\ \Delta RT &= RT - RT_{desired} \\ \Delta OSC &= OSC - OSC_{desired} \end{aligned}\right\} \qquad (1)$$

where *OV*, *RT* and *OSC* are the actual values obtained. The performance measure is actually carried out after the response is finished, at e.g., 20 s.

Fuzzy inference step

Within the inference step, the precise set values of ΔOV, ΔRT and ΔOSC are fuzzified into their corresponding membership functions, μ_{OV}, μ_{RT}, μ_{OSC}. Each membership function of the antecedent part is of the liner type, while the consequent part is of the singleton type as shown in Fig. 4. The fuzzy abbreviations *NL*, *NS*, *ZE*, *PS*, *PL*, *P* and *N* stand for *negative large*, *negative small*, *zero, positive small*, *positive large*, *positive* and *negative*, respectively. The rules used for tuning are of the form:

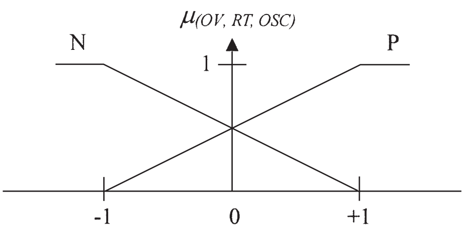
(a) **IF** *performance measure* is X_1, **THEN** ΔK_e is Y_1
(b) **IF** *performance measure* is X_2, **THEN** ΔK_{de} is Y_2

where *performance measure* is one of the three performance measures (ΔOV, ΔRT, ΔOSC). X_1 and X_2 are fuzzy sets describing these performance measures and Y_1 and Y_2 are fuzzy sets describing the amount of correction to the scaling factor. The complete set of meta-rules (scaling factor adjustment rules) used is shown in Fig. 4 (*b: Consequent*) and Table 1. This set of rules is called 'meta-rules' as they are meta-level heuristic rules.

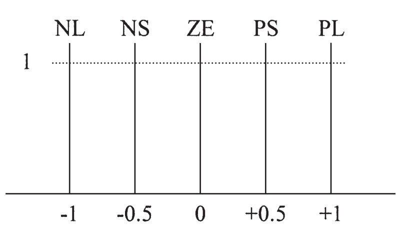
The interpretation of the meta-rules is as follows. By changing the scaling factor of each input to the controller, we effectively change the weight given to that input by the controller. For example, if the overshoot or amplitude of oscillation is larger than desired, i.e. ΔOV and ΔOSC are positive, then we need to decrease the scaling factor of ΔK_e and increase the ΔK_{de} so as to increase the effect of the proportional

TABLE 1 *Adjusting input scaling factors by using meta-rules*

	Scaling factors Adjustment	
Performance measure	ΔK_e	ΔK_{de}
ΔOV *Negative*	PL	NS
ΔOV *Positive*	NL	PS
ΔRT *Negative*	NL	PL
ΔRT *Positive*	PL	NL
ΔOSC *Positive*	NL	PL



(a) Antecedent



(b) Consequent

Fig. 4 *Membership function of the meta-rules.*

term on the controller. Similarly, if the system response is slower than desired, i.e. ΔRT is positive. The increase of the integral term of the controller is needed and hence the error scale is increased in this case. The opposite could be similarly interpreted.

However, the rules can conflict. For example, if ΔOV and ΔRT are both positive, then the rules say that ΔK_e should be both *NL* and *PL*. The fuzzy inference step resolves such conflicts by considering the relative strength of each conflicting rule by looking at the membership function of the corresponding rule. As a result, the fuzzy inference step is very important and necessary.

Tuning procedure

The rules change the interval of universe of discourse for error and change of error, and also each ΔK_e and ΔK_{de} implies an increase or decrease in the amount of K_e and K_{de} which are used in the next iteration. By applying the control results to the meta-rules and doing simplified fuzzy reasoning, ΔK_e and ΔK_{de} are then calculated. The input scaling factors of the fuzzy controller are adjusted by the following equations:

$$K_e(i+1) = K_e(i) + \Delta K_e \tag{2}$$

$$K_{de}(i+1) = K_{de}(i) + \Delta K_{de} \tag{3}$$

where i, $i+1$ are the present and future values, respectively. In this way, the scaling factors are changed at each ith interval. When the desired response is obtained, the adjustment of scaling factors is finished.

For the tuning to start, one needs to obtain an underdamped step response. ΔOV, ΔRT and ΔOSC can thus be calculated. By checking the performance measures with the meta-rules and doing simplified fuzzy reasoning, the scaling factors can then be adjusted by eqns (2) and (3). The iteration is repeated until the performance measures are within a user-specified or pre-determined tolerance. So, one may note that the STFC is actually an off-line control algorithm.

Simulation study

The simulation program is based on C/C++ computer language using LabWindows/CVI., a software development system for C programmers. It contains an interactive environment for developing programs as well as libraries of functions for creating data acquisition and instrument control applications. The main feature of the software is the simple procedure used to generate the graphical user interface compared with other C software like Turbo C++ and Borland C++. Thus an interactive and user-friendly interface (Fig. 5) can be easily developed.

Two plant models of different order are used to study the performance of the self-tuning fuzzy logic controller (both with $OV_{des} = 10\%$):

$$G_p(s) = \frac{1}{(s+1)^3} e^{-0.5s} \qquad \text{Model (1)}$$

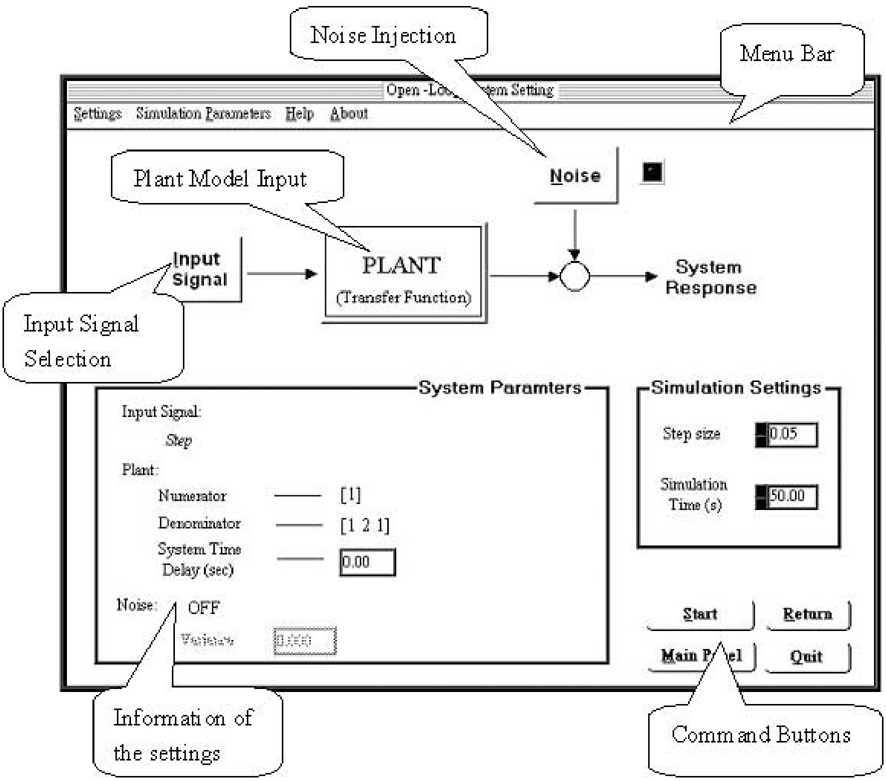


Fig. 5 *Graphical user interface of the program.*

$$G_p(s) = \frac{1.15}{0.52s^2 + 3s + 1} e^{-0.2s} \qquad \text{Model (2)}$$

Fig. 6 shows the control result of plant model (1) with initial values of $K_e = 1$; $K_{de} = 1$; $K_u = 1$. After 4 tunings, $K_e = 0.594$; $K_{de} = 1.406$; $K_u = 1$. It is obvious that in order to reduce the overshoot within the OV_{des} (10%) range, the value of K_e should be decreased, while K_{de} should be increased. Certainly, the response will be a bit slower than before. The observation is consistent with the meta-rules (Table 1). Fig. 7 shows the control result of plant model (2) with initial values of $K_e = 1$; $K_{de} = 1$; $K_u = 1$. After 2 tuning steps, $K_e = 0.832$; $K_{de} = 1.168$; $K_u = 1$. As before, to reduce the overshoot within the OV_{des} (10%) range, the value of K_e should be decreased, while K_{de} is increased (again consistent with the meta-rules). We can observe that the tuning of K_e and K_{de} is done by the computer automatically.

Since tuning the scaling factors of the FLC is a trial-and-error process, time-consuming and non-trivial, the STFC speeds up the tuning and tunes it automatically. Hence, much time is saved. The automatic nature of tuning the scaling factors makes the STFC more robust than the FLC.

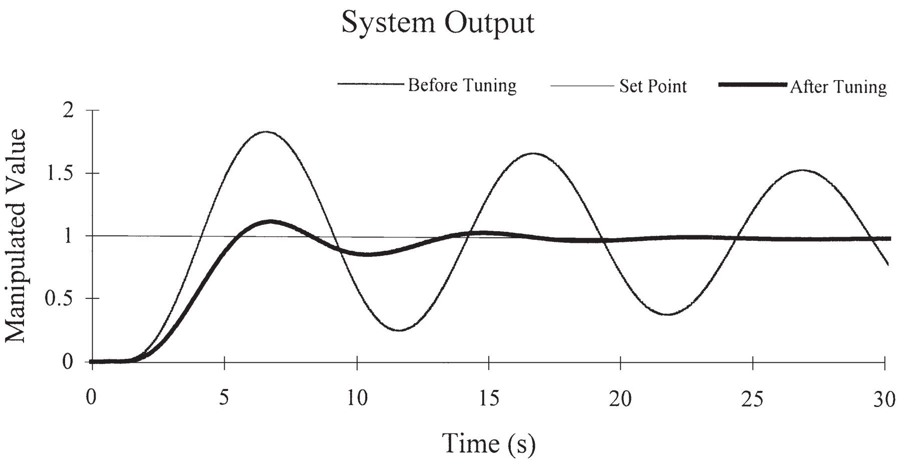


Fig. 6 *System response of model (1) before and after tuning by STFC (OV_{des} = 10%).*

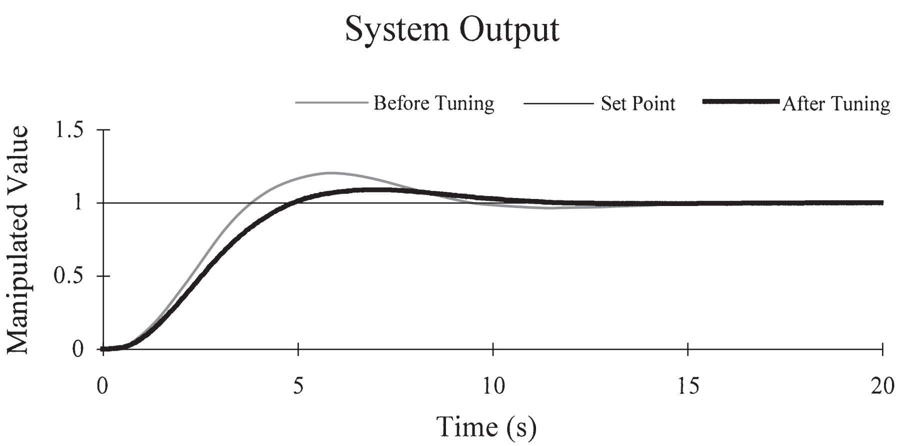


Fig. 7 *System response of model (2) before and after tuning by STFC (OV_{des} = 10%).*

Experimental

In this section, the equipment for conducting the experiment and the result will be presented. The PT326 Process Trainer is used to conduct the experiment. It is a self-contained process and control equipment. In this plant, air drawn from the atmosphere by a centrifugal blower is driven past a heater grid and through a length of tubing to the atmosphere again. The process consists of heating the air flowing in the tube to the desired temperature level and the purpose of the control equipment is to measure the air temperature, compare it with a value set by the user and generate a control signal which determines the amount of electrical power supplied to a

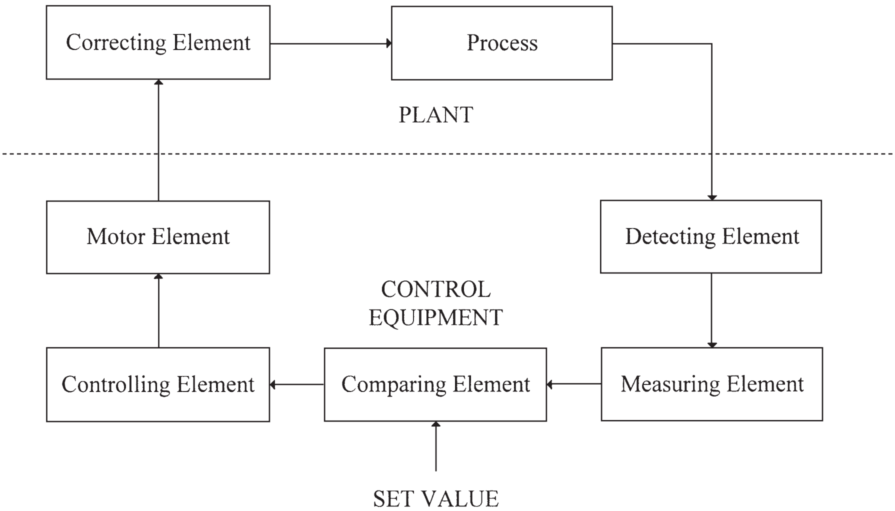


Fig. 8 *Basic components of PT326 when connected in closed-loop.*

correcting element, in this case a heater mounted adjacent to the blower. The basic components of the PT326 are shown in Fig. 8.

Least-squares approximation is employed in the signal conditioning section between the set temperature (PT326) and the magnitude of the voltage. Because the behaviour between the two variables is non-linear, it can be approximated as a parabolic equation. The equation listed below:

$$P(x) = b_0 + b_1 x + b_2 x^2 \tag{4}$$

$$\left.\begin{aligned} b_0 n + b_1 \sum x_j + b_2 \sum x_j^2 &= \sum y_j \\ b_0 \sum x_j + b_1 \sum x_j^2 + b_2 \sum x_j^3 &= \sum x_j \cdot y_j \\ b_0 \sum x_j^2 + b_1 \sum x_j^3 + b_2 \sum x_j^4 &= \sum x_j^2 \cdot y_j \end{aligned}\right\} \tag{5}$$

In eqn (4), b_0, b_1 and b_2 are cofficients, $P(x)$ is the voltage (V) and x is the temperature (°C); in eqn (5), x_j is the measured temperature data and y_j is the measured temperature in voltage.

After solving eqn (5), the output equation, eqn (4), for the conversion of temperature (x) to voltage amplitude ($P(x)$) is

$$P(x) = 0.00166x^2 + 0.04188x - 1.185 \tag{6}$$

Similar to the conversion of temperature to voltage, the equation for the conversion of voltage to temperature is

$$P(x) = 17.673 + 8.4154x - 0.4101x^2 \tag{7}$$

where $P(x)$ is the temperature and x is the voltage amplitude.

The control aim of the process is to track for the desired temperature after the user has changed the set temperature. The STFC control response is shown in Fig. 9 and the scaling factors before and after tuning is listed in Table 2 (Set temperature = 30°C and OV_{des} = 5%).

From the result, the STFC works well to bring the system to the desired set value (5% overshoot). The large undershoot shows that the algorithm is off-line in nature.

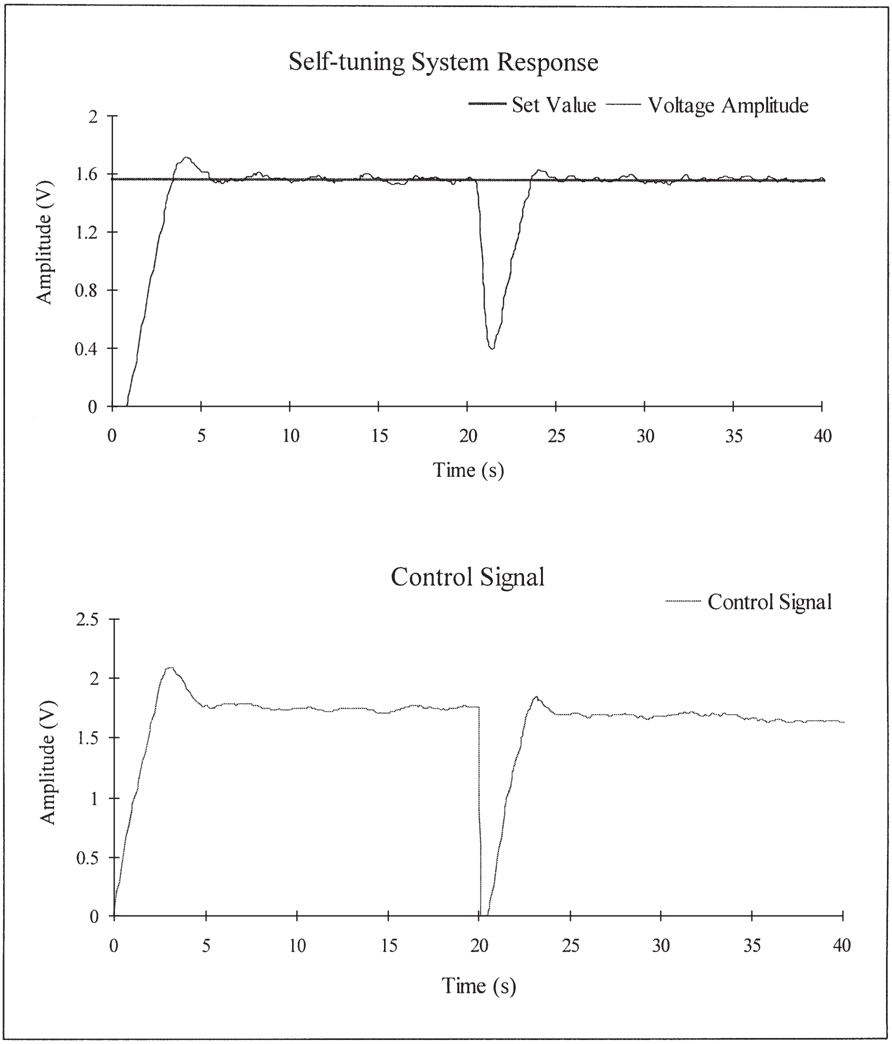


Fig. 9 *Experimental result of STFC with set temperature = 30°C. After 1 iteration (20 s each), the overshoot is within 5%. The initial value of input scaling factors:* $K_e = 2.2$, $K_{de} = 0.5$; *the final value:* $K_e = 1.904$, $K_{de} = 0.596$; *with* K_u *kept at 1.00. Sampling time = 0.1 s.*

TABLE 2 *Scaling factors before and after tuning*

Scaling factors	Before tuning	After tuning
K_e	2.2	1.904
K_{de}	0.5	0.596
K_u	1	1

However, the STFC algorithm is worth using because of the automatic nature of tuning the scaling factors. Tuning time is much reduced and the response is quite reliable and acceptable. Also, the scaling factors can be used directly to the standard fuzzy logic controller (FLC) since they both follow the same hierarchy; any transformation of data is not necessary. The control signal of the STFC is much gentler than in the PID case. Hence, the operating temperature can be extended to 45°C, instead of 20 ~ 40°C as in the case of the PID controller.

To supplement the traditional teaching methodology, the package provides an opportunity for students to experience the behaviour of controller actions when applied to a process plant. With the teaching aid, the students can investigate for themselves the control methods taught in class and observe the resulting performance achievable. In this manner, the students are provided with direct 'hands-on' experience of operating on-line control in a simulated environment. A tutorial on fuzzy logic control is included in the package (Fig. 10), so students can carry out self-paced learning. The package developed is also being incorporated into the laboratory exercises. Efforts have been made to integrate the lecture material, the laboratory practice and the computer package into a coherent teaching program.

During the first laboratory session, the students conduct an experiment to identify the parameters required to establish the process model of a system. They will then use the package to design a suitable controller to achieve optimal system performance. The students can carry out self-paced investigations with the package outside class contact hours. In doing so, they will be able to consolidate the knowledge they learnt in class. It is hoped that through self-motivation by carrying out independent study using the package, they can gain further insight into the subject matter.

In the second laboratory session, the students make use of an analogue controller for the physical implementation of the designed controller. An experiment is conducted to verify the computer simulation results and to provide an opportunity for acquiring practical experience. By the end of the second laboratory session, the students should have gained sufficient confidence in applying their knowledge and possess adequate practical experience. To reinforce what the students have gathered so far, they will then be encouraged to play with the various 'knobs' of the controller to actually carry out on-line control so as to obtain satisfactory system behavior.

The package has been evaluated by informal contacts and through observations on the students' performance in class, tests and examinations. Through informal contact with the students, most are in favour of having more computer-based teaching aids in future. Some students felt that computer-based teaching aids are more

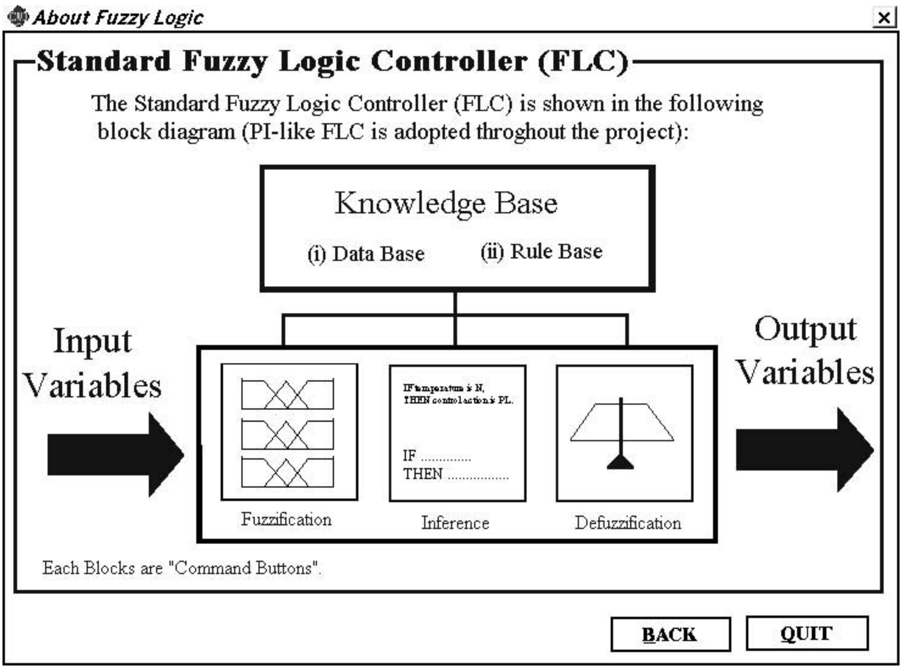


Fig. 10 *Tutorial panel on fuzzy logic control. Students can click on the block to show more details on the topic.*

useful in the subjects which required graphical simulation, especially topics involving tedious calculation such as engineering subjects. However, computer-aided instruction is not quite so applicable to teaching social subjects, such as 'management' or 'engineers in society' where discussion may be more useful to students. Moreover, students reported that through the package they acquired a deeper understanding and recollection of the topics relating to the package, especially the relationship between analytical design and practical applications. Some students suggested that questions and answers should be incorporated into the package so as to enable them to have more opportunity for self-paced learning in future. Finally, after using the computer-aided instruction package, students are more active, responsive and show more initiative when participating in the learning process, than previously. They tend to ask more questions and are more self-motivated to solve problems in the course.

Innovative methods

In the lecturing process, the experience and conditions which make a student receptive to learning should be considered. The content selected should have wide industrial applications (the society-centred point of view). Easy daily examples should be included as much as possible. If rules are learned, without an attempt to promote understanding, they may be wrongly applied in new situations. The curriculum should be concerned with the business of knowing, not just with knowledge. We should aim at increasing the power of learning for whatever we teach. A body of knowledge should be optimally structured so that it can be most easily learned. Structure means 'building blocks' of knowledge fused together to give effective learning. The sequencing and best methods of presenting that body of knowledge should be considered. Reinforcement mechanisms are necessary to ensure continued interest, such as rewards, incentives, feedback; and frequent evaluation of student performance and of the system used is desirable.

Conclusions

The fuzzy logic controller can be used in many cases where the plant is high-order and non-linear. It can process linguistic terms and vague data.[5,1,6] But the tuning of the FLC is time consuming and non-trivial. This problem is solved by using the self-tuning fuzzy logic controller. The STFC tunes the input scaling factors automatically. Hence, much time is saved in getting the controller to function properly. One major drawback of the STFC is that it is off-line, repeated learning. Generally speaking, the STFC is worth using because of its robustness, automatic nature and time saving.

This package provides an opportunity for students to experience the behaviour of controller actions when applied to a process plant. Under a Windows environment, the package possesses the essential attributes of being interactive, user-friendly and graphically oriented. The open-ended environment provides an opportunity for students to carry out self-paced study to gain further insight into the subject matter. As an educational aid, this CVI package can stimulate and promote students' motivation.

As modern technologies change rapidly, engineers and technologists are experiencing tremendous pressure in order not to be left behind. Therefore, adult education and distance learning play an important role in helping them to keep up with recent technological developments. Meanwhile, CVI provides a very effective means for on-the-job training and technology updating for adult learners; in particular it is user-friendly. As educational psychology principles are significant in speeding up the learning process of students, any CVI package should incorporate these principles as far as possible.

Acknowledgement

The authors gratefully acknowledge financial support from the Hong Kong Polytechnic University through grant G-V872.

References

1 R. R. Yager and D. P. Filev, 'Introduction to fuzzy logic control', in *Essentials of Fuzzy Modeling and Control* (John Wiley, New York, 1994), pp. 109–153.

2 D. W. Rathakrishnan and J. Yen, 'Performance evaluation of a self-tuning fuzzy controller', *Working Notes of the First International Workshop on Industrial Applications of Fuzzy Control and Intelligent Systems*. College Station, TX, 21–22 November 1991, pp. 389–397.
3 M. Maeda, and S. Murakami, 'A self-tuning fuzzy controller', *Fuzzy Sets and Systems*, **51** (1992), pp. 29–40.
4 M. Maeda, T. Sato, and S. Murakami, 'Design of the self-tuning fuzzy controller', in *Proc. Int. Conf. On Fuzzy Logic and Neural Networks*, Iizuka, Japan, 1990, pp. 393–396.
5 J. Yen, R. Langari and L. A. Zadeh, 'Introduction to Fuzzy Logic Control', in *Industrial Applications of Fuzzy Logic and Intelligent Systems* (IEE Press, New York, 1995), pp. 3–39.
6 T. J. Ross, 'Fuzzy Control Systems', in *Fuzzy Logic With Engineering Applications* (McGraw Hill, New York, 1995), pp. 469–525.