Systematic Design of Fuzzy Controllers Using Inductive Reasoning

François E. Cellier

Department of Electr. & Comp. Engr.
University of Arizona
Tucson, AZ 85721
U.S.A.

EMail: Cellier@ECE.Arizona.Edu

Francisco Mugica Alvarez

Departament d'ESAII
Universitat Politècnica de Catalunya
Diagonal 647, 2na. planta,
Barcelona 08028 Spain

EMail: Mujica@ESAII.UPC.ES

Abstract

In this paper, a new systematic design methodology for fuzzy controller design is presented. For any desired plant output, it is possible to find the optimal plant input that will produce a plant output that is as close as possible to the desired plant output. However, this constitutes an open-loop design. In this paper, a new methodology is introduced that allows to compute a signal that is close to the optimal plant input as a function of system inputs and plant outputs. To this end, an inductive reasoning model is created that estimates the optimal plant input from given system inputs and plant outputs. The inductive reasoning model can be interpreted and realized as a fuzzy controller. Thereby, a large portion of the controller is realized through feedback, and the previous open-loop design is converted to an equivalent and more robust closed-loop design.

Introduction

Fuzzy controllers have become quite popular over the past years, particularly in Japan. At least four reasons can be mentioned that make fuzzy controllers attractive:

1. *Price:* A fuzzy controller can be realized cheaply. Special chips have been designed that can be used to implement fuzzy controllers for a large variety of different industrial processes.
2. *Flexibility:* A fuzzy controller can be designed with very little knowledge of the plant it is supposed to control. Consequently, one and the same fuzzy controller can be used to control different types of processes. Only the classical PID-controller can compete with the fuzzy controller in flexibility.
3. *Robustness:* Contrary to the optimal state-feedback controller that is very sensitive to parameter variations, a fuzzy controller can deal much more reliably with a plant whose parameters are time-varying. While a human aircraft pilot is unable to compute an optimal flight path in his or her head by solving a matrix Riccati equation, he or she is able to control the aircraft successfully and reliably in situations where any one of today's autopilots would fail miserably. When an anomaly has occurred, the first thing that the human pilot will do is to switch off the autopilot. However, under normal circumstances, the autopilot can fly the aircraft more economically (consume less kerosene) than a human pilot could do. In some sense, optimality can be traded for robustness. The same holds true for fuzzy controllers. A fuzzy controller can never compete with an optimal controller in terms of efficiency, but it can be built to be considerably more robust than any optimal controller.
4. *Adaptability:* Since a fuzzy controller requires less knowledge of the environment it operates in, it is easier to make it adapt itself to a changing situation than any optimal controller.

Except for point #1 above, which is purely economically founded, all other points are closely related to each other. They all deal with questions of efficiency versus flexibility, of specialization versus generality.

If humanity is ever to outgrow its cradle by colonizing other planets, it will have to rely on an army of fairly autonomously operating robots that will be needed to prepare these other planets for human arrival. These robots are not manufacturing robots. It is not essential that they produce as much merchandise per time unit as possible. It is much more important that they are *robust*, i.e., can operate on their own without running into any sort of trouble, that they are *adaptive*, i.e., can reliably accommodate to a changing environment, and that they are *flexible*, i.e., can be used for various different tasks. Fuzzy control may be an answer to some of these demands.

Fuzzy controllers are essentially rule-based controllers whereby continuous variables are discretized (recoded) into classes. A recoded fuzzy variable preserves its quantitative information in a fuzzy membership function that it carries along with its class value. Operations performed on fuzzy variables are performed separately on their class values (using finite state automata techniques, so-called "rules") and on their fuzzy membership functions (using fuzzy logic).

The rules and fuzzy membership functions employed in a fuzzy controller are usually determined heuristically, i.e., they are manually coded on the basis of an intuitive understanding of the functioning of the underlying process to be controlled.

A systematic design of the rules and/or their accompanying fuzzy membership functions has been attempted in the past. For example, a type of *genetic algorithm* [5] has been successfully employed to optimize the behavior of a fuzzy controller used in an autonomous spacecraft rendezvous maneuver [6]. More recently, a neural network of the associative memory type was employed to initially train (off-line) and then adapt (on-line) the parameters of a fuzzy controller for an inverted pendulum [9].

This paper presents a new systematic design of fuzzy controllers that can be used to control any plant for which the inverse dynamics problem can be solved. The methodology employed in the design is centered around *fuzzy inductive reasoning* [1,10], a technique geared at the qualitative simulation of dynamical continuous-time processes [2].

The underlying *inverse dynamics problem* is a well-known control problem that has been studied extensively, particularly in the context of robot control. Given the desired path of the end-effector (the result of solving the path planning problem), find the optimal position of each joint (inverse kinematics problem), then find the optimal forces and torques in each joint (inverse dynamics problem) [4]. It is not the purpose of this paper to reiterate on inverse dynamics. Instead, an example will be used where the solution of the inverse dynamics problem is trivial.

This is a proof-of-concept paper. The application chosen to demonstrate the approach is simple and generic. It is simple enough to be described in full, yet complex enough to prove the practicality of the approach. More realistic applications, such as fuzzy control of a double inverted pendulum, and fuzzy control of a large robot arm, are currently under development. Upon completion, these applications will be reported elsewhere.

Fuzzy Recoding

Fuzzy recoding denotes the process of converting a crisp variable to a fuzzy variable. Figure 1 shows the fuzzy recoding of a variable called "systolic blood pressure."

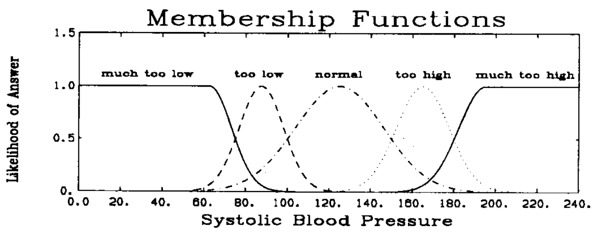

Figure 1. Fuzzy recoding.

It was decided to use bell-shaped fuzzy membership functions rather than the more commonly used triangular membership functions. This membership function can be easily calculated using the equation:

$$Memb_i = \exp(-k_i \cdot (x - \mu_i)^2) \qquad (1)$$

where x is the continuous variable that needs to be recoded, and k_i is determined such that the membership function $Memb_i$ degrades to a value of 0.5 at the neighboring landmarks. In the above example, a crisp systolic blood pressure of 135.0 is recoded into a fuzzy variable with a class value of 'normal', a fuzzy membership function of 0.895, and a side function of 'right.' Thus, a single crisp value is recoded into a fuzzy triple. Any systolic blood pressure with a crisp value between 100.0 and 150.0 will be recoded into a fuzzy variable with the class value 'normal.' The fuzzy membership function denotes the numerical value of the bell-shaped curve shown on Fig.1, always a value between 0.5 and 1.0, and the side function indicates whether the crisp value is to the left or to the right of the maximum of the fuzzy membership function. Obviously, the fuzzy variable contains the same information as the original crisp variable. The crisp value can be regenerated accurately from the fuzzy triple, i.e., without any loss of information.

Due to space limitations, details of how crisp variables are optimally recoded into fuzzy variables will not be given in this paper. These details are provided in [1,10].

Fuzzy Optimal Masks

A mask denotes relationship between different variables. For example, given the following raw data model consisting of five variables, namely the inputs u_1 and u_2 and the outputs y_1, y_2, and y_3 that are recorded at different values of time.

$$\begin{array}{l} time \\ 0.0 \\ \delta t \\ 2\cdot\delta t \\ 3\cdot\delta t \\ \vdots \\ (n_{rec}-1)\cdot\delta t \end{array} \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \end{array} \qquad (2)$$

Each column of the raw data model contains the class values of one fuzzy variable recorded at different values of time, and each row contains the recordings of the class values of all fuzzy variables at one point in time. The raw data matrix is accompanied by a fuzzy membership matrix and a side matrix of the same dimensions.

A mask denotes a relationship between fuzzy variables. For example, the mask

$$\begin{array}{l} t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & -2 & -3 & 0 & 0 \\ -4 & 0 & +1 & 0 & 0 \end{pmatrix} \end{array} \qquad (3)$$

denotes the following relationship pertaining to the five variable system:

$$y_1(t) = \tilde{f}(y_3(t-2\delta t), u_2(t-\delta t), y_1(t-\delta t), u_1(t)) \qquad (4)$$

Negative elements in the mask matrix denote inputs of the qualitative functional relationship. The example mask has four inputs. The sequence in which they are enumerated is immaterial. They are usually enumerated from left to right and top to bottom. A positive element in the mask matrix denotes the output. Thus, Eq.(3) is simply a matrix representation of Eq.(4). The mask must have the same number of columns as the raw data matrix. The number of rows of the mask matrix is called the *depth* of the mask. The mask can be used to flatten a dynamic relationship out into a static relationship. The mask can be shifted over the episodical behavior. Selected inputs and outputs can be read out from the raw data matrix and can be written on one row next to each other. Figure 2 illustrates this process.

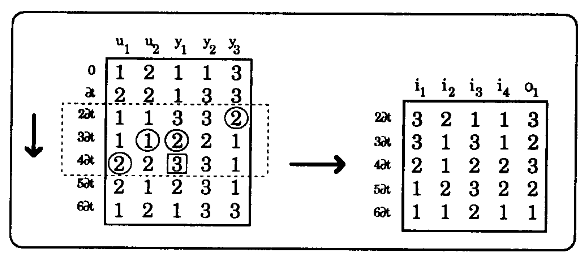

Figure 2. Flattening relationships through masking.

After the mask has been applied to the raw data, the formerly dynamic episodical behavior has become static, i.e., the relationship is now contained within a single row

$$o_1(t) = \tilde{f}(i_1(t), i_2(t), i_3(t), i_4(t)) \qquad (5)$$

The resulting matrix is called *input/output matrix.*

How is the mask selected? A *mask candidate matrix* is constructed in which negative elements denote potential inputs, and the single positive element denotes the true output of the mask.

A good mask candidate matrix for the previously mentioned five variable system might be

$$\begin{array}{c} t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & +1 & 0 & 0 \end{pmatrix} \end{array} \tag{6}$$

A mask candidate matrix is an ensemble of all acceptable masks. The optimal mask selection algorithm determines the best among all masks that are compatible with the mask candidate matrix. The mask of Eq.(3) is one such mask. The optimal mask is the one mask that maximizes the forecasting power of the inductive reasoning process, i.e., the mask that results in the most deterministic input/output matrix.

Due to space limitations, the details of how the optimal mask selection algorithm works are omitted from this paper. These details are also provided in [1,10].

Fuzzy Forecasting

Once the optimal mask has been determined, it can be applied to the given raw data matrix resulting in a particular input/output matrix. Since the input/output matrix contains functional relationships within single rows, the rows of the input/output matrix can now be sorted in alphanumerical order. The result of this operation is called the *behavior matrix* of this system. The behavior matrix is a finite state machine. For each combination of input values, it shows which output is most likely to be observed.

Forecasting is now a straightforward procedure. The mask is simply shifted further down beyond the end of the raw data matrix, future inputs are read out from the mask, and the behavior matrix is used to determine the future output, which can then be copied back into the raw data matrix. In fuzzy forecasting, it is essential that, together with the class value of the output, also a fuzzy membership value and a side value are forecast. Thus, fuzzy forecasting predicts an entire fuzzy triple from which a crisp variable can be regenerated whenever needed.

In fuzzy forecasting, the membership and side functions of the new input are compared with those of all previous recordings of the same input class value contained in the behavior matrix. The one input with the most similar membership and side functions is identified. For this purpose, a cheap approximation of the regenerated quantitative signal

$$d = 1 + side * (1 - Memb) \tag{7}$$

is computed for every input variable of the new input set, and the regenerated d_i values are stored in a vector. This reconstruction is then repeated for all previous recordings of the same input set. Finally, the $\mathcal{L}_2$ norms of the difference between the d vector of the new input and the d vectors of all previous recordings of the same input are computed, and the previous recording with the smallest $\mathcal{L}_2$ norm is identified. Its *output* and *side* values are then used as forecasts for the *output* and *side* values of the current state.

Forecasting of the new membership function is done a little differently. Here, the five previous recordings with the smallest $\mathcal{L}_2$ norms are used (if at least five such recordings are found in the behavior matrix), and a distance-weighted average of their fuzzy membership functions is computed and used as the forecast for the fuzzy membership function of the current state.

Absolute weights are computed as follows:

$$w_{abs_i} = \frac{d_{max} - d_i}{d_{max}} \tag{8}$$

The absolute weights are numbers between 0.0 and 1.0. Using the sum of the five absolute weights:

$$s_w = \sum_{\forall i} w_{abs_i} \tag{9}$$

it is possible to compute relative weights:

$$w_{rel_i} = \frac{w_{abs_i}}{s_w} \tag{10}$$

Also the relative weights are numbers between 0.0 and 1.0. However, there sum is always equal to 1.0. It is therefore possible to interpret the relative weights as percentages. Using this idea, the membership function of the new output can be computed as a weighted sum of the membership functions of the outputs of the previously observed five nearest neighbors:

$$Memb_{out_{new}} = \sum_{\forall i} w_{rel_i} \cdot Memb_{out_i} \tag{11}$$

More details on fuzzy forecasting are provided in [1].

An Example

Given a linear SISO plant with the transfer function:

$$G(s) = \frac{s^2 + 3 \cdot s + 7}{s^2 + 5 \cdot s + 10} \tag{12}$$

The plant was chosen as a proper but not strictly proper transfer function since, in this case, computation of the inverse dynamics is trivial. The goal is to design a fuzzy feedback controller around this plant such that the overall system behaves similarly to a linear system with the transfer function:

$$G_{tot}(s) = \frac{1}{s+1} \tag{13}$$

Obviously, this task can be accurately accomplished by the classical controller shown on Fig.3

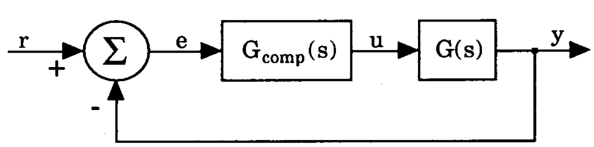

Figure 3. Classical controller design.

where

$$G_{comp}(s) = \frac{s^2 + 5 \cdot s + 10}{s \cdot (s^2 + 3 \cdot s + 7)} \tag{14}$$

In this paper, a fuzzy controller will be used instead. The control system with the fuzzy controller is shown on Fig.4

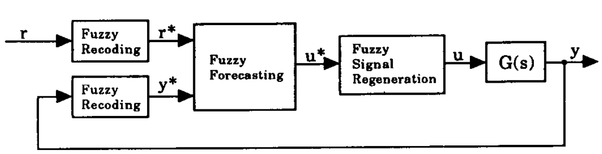

Figure 4. Fuzzy controller design.

The crisp system input, r, is converted into a fuzzy variable, r^*, by means of fuzzy recoding. Similarly, also the system output, y, is converted into a fuzzy variable, y^*. A fuzzy controller computes a fuzzy control input, u^*, by means of fuzzy forecasting. The fuzzy control signal is then converted back to the crisp control signal, u, by means of fuzzy signal regeneration.

The fuzzy controller is designed using the following procedure. In a first experiment, a simulation program is written that simulates the system shown on Fig.5

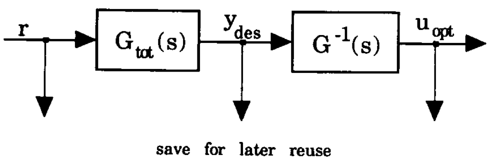

Figure 5. Data extraction.

A binary random input, r, is applied to a model of the desired total transfer function. Such an input excites the system optimally well at all frequencies. At the output of the closed-loop transfer function, $G_{tot}(s)$, the desired output signal, y_{des}, is measured. This signal is fed into the inverse transfer function, $G^{-1}(s)$, of the plant. As a result, the optimal control input, u_{opt}, is found.

All three variables, r, y_{des}, and u_{opt} are stored in the *measurement matrix* from which the *raw data matrix* is obtained by means of (off-line) fuzzy recoding.

The raw data matrix can then be used to determine an optimal mask. Since the optimal mask should approximately cover the slowest time constant of the closed-loop system (1.0 seconds), a mask depth of 3 would suggest the use of a communication interval of 0.5 seconds, i.e., the measurement matrix (and the raw data matrix) should contain entries (rows) that are 0.5 seconds apart.

Unfortunately, fuzzy inductive forecasting will predict only one value of u per sampling interval. Thus, the overall control system of Fig.4 will react like a sampled-data control system with a sampling rate of 0.5 seconds. From a control system perspective, the variables should be sampled considerably faster, namely once every 0.05 seconds.

Therefore, it was decided to choose the following mask candidate matrix:

$$\begin{array}{l} {}_t\backslash^x \\ t-20\delta t \\ t-19\delta t \\ \vdots \\ t-11\delta t \\ t-10\delta t \\ t-9\delta t \\ \vdots \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} r & y & u \end{array} \\ \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ -1 & -1 & -1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ -1 & -1 & +1 \end{pmatrix} \end{array} \tag{15}$$

of depth 21. As mandated by control theory, the sampling interval δt is chosen to be 0.05 seconds. Yet, as dictated by the inductive reasoning technique, the control input, u, at time t will depend on past values of r, y, and u at times $t-0.5$ and $t-1.0$.

The optimal mask found with this mask candidate matrix is:

$$\begin{array}{l} {}_t\backslash^x \\ t-20\delta t \\ t-19\delta t \\ \vdots \\ t-11\delta t \\ t-10\delta t \\ t-9\delta t \\ \vdots \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} r & y & u \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & -1 & +1 \end{pmatrix} \end{array} \tag{16}$$

In other words:

$$u(t) = \tilde{f}(y(t)) \tag{17}$$

Unfortunately, this "optimal" mask won't work. Due to the direct coupling between the plant input, u, and the plant output, y, the optimal mask suggests that knowledge of the current value of the plant output, y, is sufficient to predict the optimal value of the plant input, u. In an open-loop situation, this is correct. If $y(t)$ is given, $u(t)$ can be estimated accurately with this optimal mask. However, this is a chicken-and-egg problem. If $y(t)$ is given, $u(t)$ can be computed, and once $u(t)$ is known, $y(t)$ can be computed also. There exists an algebraic loop between these two variables.

The fact that the plant was chosen as a proper but not strictly proper transfer function made the solution of the inverse dynamics problem easy, but, at the same time, made the fuzzy control problem considerably more difficult. The optimal mask algorithm optimizes the mask for open-loop. If the plant has low pass characteristics, the optimal mask will also work in a closed-loop setting. However, in the given example, some of the trivial masks (such as the above "optimal" mask) exhibit poor tracking behavior, others show stability problems.

In our case, it was necessary to search through the mask history, i.e., through the set of suboptimal masks. It was found that the second best mask of complexity four (containing four non-zero elements) exhibits both good tracking behavior and good stability behavior. The mask is as follows:

$$\begin{array}{l} {}_t\backslash^x \\ t-20\delta t \\ t-19\delta t \\ \vdots \\ t-11\delta t \\ t-10\delta t \\ t-9\delta t \\ \vdots \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} r & y & u \end{array} \\ \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ -2 & -3 & +1 \end{pmatrix} \end{array} \tag{18}$$

The first 1800 rows (90 seconds) of the raw data matrix were used as past history data to compute the optimal mask. Fuzzy forecasting was used to predict new fuzzy triples of u for the last 200 rows (10 seconds) of the raw data matrix. From the predicted fuzzy triples, crisp values were then regenerated.

Figure 6 compares the true "measured" values of u obtained from the original simulation (solid line) with the forecast and regenerated values obtained from fuzzy inductive reasoning (dashed line) in open loop, i.e., the "measured" time trajectories $r(t)$ and $y(t)$ were optimally recoded into the fuzzy signals $r^*(t)$ and $y^*(t)$. Fuzzy forecasting was then used to estimate the fuzzy signal $u^*(t)$. Fuzzy signal regeneration was used to reconstruct the crisp signal $u(t)$, which was then compared with the previously "measured" trajectory $u_{opt}(t)$.

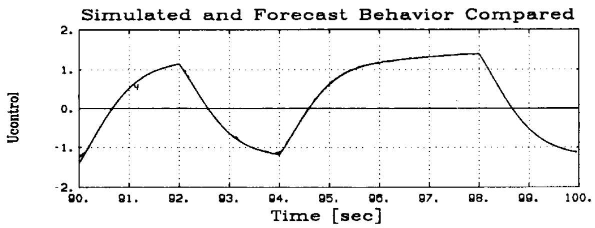

Figure 6. Simulated and forecast control input compared.

Figure 7 shows the configuration used in this experiment.

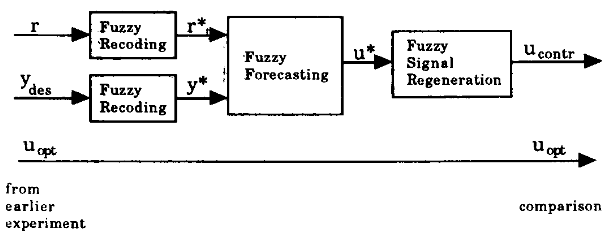

Figure 7. Qualitative simulation experiment in open–loop.

The results are encouraging. There is hardly any difference between the optimal trajectory, u_{opt}, and the output of the fuzzy controller, u, in open–loop. Quite obviously, the optimal mask contains sufficient information to be used as a valid replacement of the true inverse dynamics. Notice that the fuzzy inductive reasoning model was constructed solely on the basis of measurement data.

In the next experiment, the fuzzy controller was inserted into the overall system as previously shown on Fig.4. The crisp control input, r, is converted to a qualitative triple, r^*, using fuzzy recoding. Also the crisp plant output, y, is converted to a qualitative triple, y^*. From these two qualitative signals, a qualitative triple of the plant input u^*, is computed by means of fuzzy forecasting. This qualitative signal is then converted back to a crisp signal, u using fuzzy signal regeneration. The plant itself is described by means of a differential equation model.

Forecasting was restricted to the last 200 sampling intervals, i.e., to the time span from 90.0 seconds to 100.0 seconds. Figure 8 compares the desired plant output, $y_{des}(t)$, from the purely quantitative simulation (solid line) with the output, y, of the model containing the fuzzy controller (dashed line).

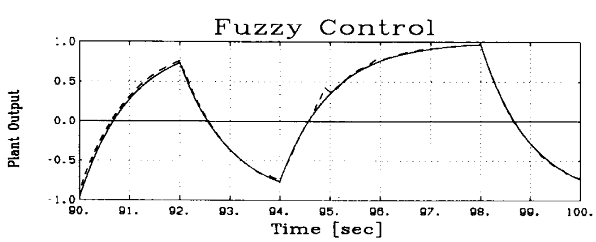

Figure 8. Validation of fuzzy controller.

As can be seen, the plant with the fuzzy controller behaves indeed almost exactly like $1/(s+1)$ as desired. The new design approach worked beautifully although the direct input/output coupling in the plant made the design task considerably more difficult. It has been shown that fuzzy inductive reasoning can indeed be used to systematically design fuzzy controllers for systems with multiple controller inputs. If the plant contains multiple plant inputs (controller outputs), each controller output is computed separately by a different optimal mask.

Summary and Conclusions

The example demonstrates the validity of the chosen approach. Control systems containing a fuzzy controller designed using inductive reasoning are similar in effect to sampled–data control systems. *Fuzzy recoding* takes the place of analog–to–digital converters, and *fuzzy signal regeneration* takes the place of digital–to–analog converters. However, this is where the similarity ends. Sampled–data systems operate on a fairly accurate representation of the digital signals. Typical converters are 12–bit converters, corresponding to discretized signals with 4096 discrete levels. In contrast, the fuzzy inductive reasoning model employed in the above example recoded all three variables into fuzzy variables with the three classes 'small,' 'medium,' and 'large.' The quantitative information is retained in the fuzzy membership functions that accompany the qualitative signals. Due to the small number of discrete states, the resulting finite state machine is extremely simple. Fuzzy membership forecasting has been shown to be very effective in inferring quantitative information about the system under investigation in qualitative terms.

Due to the space limitations inherent in a publication in conference proceedings it was not possible to provide, in this paper, any details of the programs used for simulation. Fuzzy inductive reasoning is accomplished using SAPS–II [3], a software that evolved from the General System Problem Solving (GSPS) framework [7,8,13]. SAPS–II is implemented as a (FORTRAN–coded) function library of CTRL–C [12]. A subset of the SAPS–II modules, namely the recoding, forecasting, and regeneration modules have also been made available as an application library of ACSL [11], which is the software used in the mixed quantitative and qualitative simulation runs. More details will be provided in an enhanced version of this paper that is currently being prepared for submission to a journal.

References

[1] Cellier, F.E., (1991). *Continuous System Modeling*, Springer–Verlag, New York.

[2] Cellier, F.E., A. Nebot, F. Mugica, and A. de Albornoz, (1992). "Combined Qualitative/Quantitative Simulation Models of Continuous–Time Processes Using Fuzzy Inductive Reasoning Techniques," *Proceedings SICICA'92, IFAC Symposium on Intelligent Components and Instruments for Control Applications*, Málaga, Spain, May 22–24.

[3] Cellier, F.E., and D.W. Yandell, (1987). "SAPS–II: A New Implementation of the Systems Approach Problem Solver," *International J. of General Systems*, **13**(4), pp. 307–322.

[4] Fu, K.S., R.C. Gonzalez, and C.S.G. Lee, (1987). *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw–Hill, St. Louis.

[5] Goldberg, D.E., (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison–Wesley, Reading, Mass.

[6] Karr, C.L., L.M. Freeman, and D.L. Meredith, (1989). "Improved Fuzzy Process Control of Spacecraft Autonomous Rendezvous Using a Genetic Algorithm," *Proceedings SPIE Intelligent Control and Adaptive Systems Conference*, Philadelphia, Penn., pp. 274–288.

[7] Klir, G.J., (1985). *Architecture of Systems Problem Solving*, Plenum Press, New York.

[8] Klir, G.J. (1989). "Inductive Systems Modelling: An Overview," *Modelling and Simulation Methodology: Knowledge Systems' Paradigms* (M.S. Elzas, T.I. Ören, and B.P. Zeigler, Eds.), Elsevier Science Publishers B.V. (North–Holland), Amsterdam, The Netherlands, pp. 55–75.

[9] Kosko, B. (1992). *Neural Networks and Fuzzy Systems — A Dynamical Systems Approach to Machine Intelligence*, Prentice–Hall, Englewood Cliffs, N.J.

[10] Li, D., and F.E. Cellier, (1990). "Fuzzy Measures in Inductive Reasoning," *Proceedings WSC'90, Winter Simulation Conference*, New Orleans, La., December 10–13, pp. 527–538.

[11] Mitchell & Gauthier Assoc., *ACSL: Advanced Continuous Simulation Language — User Guide and Reference Manual*, Concord, Mass.

[12] Systems Control Technology, Inc., *CTRL–C, A Language for the Computer–Aided Design of Multivariable Control Systems, User's Guide*, Palo Alto, Calif.

[13] Uyttenhove, H.J., (1979). *SAPS — System Approach Problem Solver*, Ph.D. Dissertation, SUNY Binghampton, N.Y.