1/29/01

NASA/TM-2000-210619

Self-Tuning of Design Variables for Generalized Predictive Control

Chaung Lin
Institute for Computer Applications in Science and Engineering
Langley Research Center
Hampton, Virginia

Jer-Nan Juang
Langley Research Center
Hampton, Virginia

December 2000

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-2000-210619

Self-Tuning of Design Variables for Generalized Predictive Control

Chaung Lin
Institute for Computer Applications in Science and Engineering
Langley Research Center
Hampton, Virginia

Jer-Nan Juang
Langley Research Center
Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

December 2000

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Self-Tuning of Design Variables For Generalized Predictive Control[1]

Chaung Lin[2]

National Tsing Hua University
Hsinchu, Taiwan 300
Republic of China

Jer-Nan Juang[3]

NASA Langley Research Center
Hampton, Virginia 23681
United States of America

Abstract

Three techniques are introduced to determine the order and control weighting for the design of a generalized predictive controller. These techniques are based on the application of fuzzy logic, genetic algorithms, and simulated annealing to conduct an optimal search on specific performance indexes or objective functions. Fuzzy logic is found to be feasible for real-time and on-line implementation due to its smooth and quick convergence. On the other hand, genetic algorithms and simulated annealing are applicable for initial estimation of the model order and control weighting, and final fine-tuning within a small region of the solution space. Several numerical simulations for a multiple-input and multiple-output system are given to illustrate the techniques developed in this paper.

[1] This research was performed when the first author visited the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia, USA.

[2] Professor of Engineering and System Science, clin@ne.nthu.edu.tw

[3] Principal Scientist, Structural Dynamics Branch, MS 230, j.juang@larc.nasa.gov

Introduction

Adaptive predictive feedback control has been successfully used to suppress plate vibrations for acoustic noise reduction as presented in Ref. 1. The adaptive predictive controller consists of an on-line identification technique coupled with a control scheme. The basic design variables used in generalized predictive control (GPC) include autoregressive exogeneous (ARX) model order for on-line identification and control weighting for computing control force. They must be determined before on-line implementation. The values of these variables are typically obtained by experience or trial-and-error methods. Therefore, an experienced engineer is needed to perform the tuning process. Also, the optimal performance might not be achieved.

An on-line tuning method using fuzzy logic rules will be developed. Starting with a set of proper initial values for the design variables, GPC performance can be gradually improved. The required data are collected and then sent to the higher level. At the higher level, the fuzzy logic rules are used to adjust the design variables. The procedure is repeated until some stopping criterion is met. However, the stability of the GPC controller cannot be guaranteed.

To solve the aforementioned problem, an optimization problem is formed. An objective function is defined, which includes a term to reflect performance and a controller stability penalty term. A genetic algorithm or simulated annealing is adopted to perform optimal search for control weighting and ARX model order. Both methods need to evaluate the objective function over the solution space, which means that poor performance may be presented during the search process. Therefore, this method is suitable for the initial tuning and the fine-tuning of the control weighting. During the initial tuning, the objective function evaluation should use an identified system model instead of on-line experimental data. However, during the fine-tuning, the range of solution space is limited so that only the acceptable performance will occur. Therefore, on-line tuning is feasible in the latter case.

Generalized Predictive Control

Consider the predictive matrix equation as defined in Ref. (1)

$$y_s(k) = \mathrm{T}u_s(k) + \Theta v_p(k-p) \tag{1}$$

The vectors $y_s(k)$ and $u_s(k)$ consist of current and future output and input data, respectively, from the current time step *k* to a desired future time step *k+s-1*,

$$y_s(k) = \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+s-1) \end{bmatrix} \quad \mathit{and} \quad u_s(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+s-1) \end{bmatrix} \tag{2}$$

and $v_p(k-p)$ contains past input and output data from time step *k-p* to *k-1*,

$$v_p(k-p) = \begin{bmatrix} y(k-p) \\ \vdots \\ y(k-1) \\ u(k-p) \\ \vdots \\ u(k-1) \end{bmatrix} \tag{3}$$

The matrix T is a Toeplitz matrix which is formed from the pulse response sequence. The matrix Θ is a rectangular coefficient matrix. The integer *p* is the order of the identified model and *s* is the prediction horizon.

The goal of the predictive control is to compute the control force $u_s(k)$ that minimizes the cost function

$$J(k) = \frac{1}{2}\left\{[y(k) - \tilde{y}(k)]_s^T [y(k) - \tilde{y}(k] + u_s(k)^T \Lambda u_s(k)\right\} \tag{4}$$

where Λ is a positive-definite matrix to weight the force vector $u_s(k)$. Taking the derivative of the resulting equation with respect to $u_s(k)$ with the aid of Eq. (1) yields

$$u_s(k) = -[\mathrm{T}^T\mathrm{T} + \Lambda]^{-1}\mathrm{T}^T[\Theta v_p(k-p) - \tilde{y}_s(k)] \tag{5}$$

The first r values of the force vector $u_s(k)$ are applied to the r control inputs, the remainder is discarded, and a new control sequence is calculated the next time step. In order to carry out the above process, the desired outputs in the vector $\tilde{y}_s(k)$ must be given. In the regulation problem, the desired plant output is zero. Taking the first r rows with zero desired outputs results in

$$u(k) = \text{First } r \text{ rows of } \{-[\mathrm{T}^T\mathrm{T} + \Lambda]^{-1}\mathrm{T}^T\}\Theta v_p(k-p) \tag{6}$$

When Λ is a zero matrix, the closed-loop system will be unstable for non-minimum systems because the matrix T is rank deficient. The quantity Λ must be carefully tuned to make the system stable.

Application of Fuzzy Logic

In theory, the smaller the value of Λ, the better the performance of GPC, which is due to rather large control magnitude. However, there is a limitation of control magnitude and so the performance and control magnitude should be traded off, which is determined by Λ. For simplicity, assume that the weighting matrix Λ is an identity matrix multiplied by a scalar λ. This is a common assumption in practice in predictive control applications. The fuzzy logic rules are developed to tune the value of λ. To understand fuzzy logic, the reader is directed to Ref. 2 for further information.

Control Weighting Tuning

In this study, the antecedent part of fuzzy rules has two inputs, i.e., performance index (PI) and control index (CI), which are defined as follows.

$$\mathrm{PI} = \frac{\mathrm{RMS}\left[\mathrm{y}^{\mathrm{T}}(k)y(k)\right]}{\mathrm{RMS}\left[y_{op}^{T}(k)y_{op}(k)\right]} \tag{7}$$

$$\mathrm{CI} = \mathrm{Max}_{\mathrm{k}}\left|u(k)\right| / u_{\max} \tag{8}$$

where $y(k)$ is the system output with control, $y_{op}(k)$ is the system output without control, $u(k)$ is the control, u_{max} is the maximum control force allowed, and k is the time step.

The membership functions of linguist variables are shown in Fig.1 and the rule matrix is shown in Table 1. The output linguist variable (multiplication factor) has ten linguist labels, which are represented by 1 to 10 as shown in Fig.1. The number 7 means zero action. Note that the multiplication factor value is an exponent to a number that is two in this study. For example, if PI = 0.5 and CI = 0.8 computed from Eqs. (7) and (8), then PI is A and CI is A with degree 1 (see Fig. 1). From Table 1, only one fuzzy rule is activated and its label is 6. The corresponding value from the bottom figure in Fig. 1 results in -0.5. The final multiplication to the control weighting λ is 2^x=0.70711 where x= -0.5. The integer 2 is introduced to regulate the change of λ. Any other integer or number may be used for different applications. The recursive update law for the control weighting is

$$\lambda_{new} = 2^x \lambda_{old} \tag{9}$$

Now consider the case where PI = 0.4, then both fired degrees of G and A from the top figure of Fig. 1 are 0.5. In addition, if CI = 1, the degree of linguist label U from the middle figure of Fig. 1 is 1. According to Table 1, the activated rules are linguist labels 8 and 9. The defuzzified value becomes 0.75. Thus the multiplication to the control weighting is $2^{0.75}$. The negative value of x means the multiplier is less than one, i.e., decrease λ, and the positive value means greater than one, i.e., increase λ. The fuzzy rules are briefly described as below.

(1) If the controller performance is good and the control effort is close to the maximum value, then λ should be increased, i.e., using the linguist label 8~10.

(2) If the controller performance is not satisfied and the control magnitude has room to increase, then λ should be decreased, i.e., using linguist label 1~6.

(3) If both PI and CI are satisfied or there is no room to increase control magnitude, then λ should be kept constant, i.e., using the linguist label 7.

The universe of discourse defined for the fuzzy variables in this study may not be suitable for other systems. However, this problem can be overcome by multiplying the signals by a scalar before fuzzification so that the fuzzy rules can be applied correctly. For example, if the value of variable PI is 0.3, the degree of good is 1.0 according to Fig. 1. If the value 0.3 is excellent in a specific system, the signal PI should be multiplied by 1/3 to become 0.1 and PI should be excellent with degree close to 1 after fuzzification. On the other hand, if this value is only acceptable, the signal should be multiplied by 5/3 to yield 0.5 so that it will belong to acceptable with degree close to 1 after fuzzification. With these modifications, the defined fuzzy logic rules can be applied to many different systems.

The computational procedure for control weighting tuning can be summarized as follows

1. Perform an open loop experiment with a given input sequence *u(k)* where $k = 1,2,\ldots l$ to obtain the open-loop output response history $y_{op}(k)$ with $k = 1,2,\ldots l$. Note that the system may also be excited by unknown disturbances.
2. Set the control weighting to an initial value, $\lambda = 1$ for example.
3. Perform system identification to compute T and Θ shown in Eq. (1).
4. Determine the control law according to Eq. (6).
5. Close the system loop using the control force defined by Eq. (6).
6. Compute the performance index (PI) and the control index (CI) using Eqs. (7) and (8).
7. Update the control weighting using the fuzzy rules described earlier.
8. Repeat step 3 to 7 until the weighting λ converges.

ARX Model Order Tuning

The fuzzy rules may also be developed to tune the ARX model order. Usually, high order models produce better output prediction error. However, when the model order exceeds a certain number, the model only improves slightly at the expense of considerable computation time, which imposes limitations on the sampling rate. Therefore, it is necessary to adopt a proper model order to trade-off computation time.

In this study, three variables are considered to determine the action. These are normalized output prediction error (PE), change in PE (CPE), and normalized model order (ORD), which are defined as follows.

$$\mathrm{PE} = \frac{\mathrm{RMS}\left(\left\| y(k) - \hat{y}(k) \right\|\right)}{\mathrm{RMS}\left(\left\| y(k) \right\|\right)} \tag{10}$$

$$\mathrm{CPE} = sign\left(N(k) - N(k-1)\right) \times \left[\mathrm{PE}(k\text{-}1) - \mathrm{PE}(k)\right] \tag{11}$$

$$\mathrm{ORD} = N(k) \,/\, N_{\max} \tag{12}$$

where y(k) and $\hat{y}(k)$ are output and predicted output, respectively, N and $N_{\max}$ are model order and maximum model order, respectively, and k is the time step. Equation (11) includes the term *sign(N(k)-N(k-1))* for CPE to indicate the direction of change of model order. If CPE is positive, the change of model error will be positive; otherwise, the direction will be reversed.

In order to simplify the fuzzy rule structure, a hierarchical structure is adopted. The first level's input linguistic variables are PE and CPE, and the output linguist variable is an order multiplier, which is referred to as OUT1. The input variables for the second level are ORD and OUT1 and the output linguist variable is a multiplier that adjusts the result of the first level, which is called OUT2.

The membership functions for the first level fuzzy variables are shown in Fig.2 and rule matrix is shown in Table 2. The main ideas are as follows.

(1) If CPE is negative, then decrease the model order, i.e., OUT1 < 0.

(2) If CPE is positive, then increase the model order, i.e., OUT1 > 0.

(3) If CPE is close to zero, then maintain the same order, i.e., OUT1 = 0.

The membership functions and rule matrix for the second level fuzzy rules are shown in Fig. 3 and Table 3, respectively. The main purpose of these rules is to adjust the magnitude of the first level output. When the model order is close to the allowable value, which is determined by the computer, the number of multiplication will be decreased; otherwise maintain the same number of multiplication. The final multiplier of the order is OUT2*(1.+ OUT1).

Consider the simple example where PE = 0.35, i.e., PE is B with degree 1 from the top figure of Fig. 2, and $\mathrm{CPE} = 3\times10^{-3}$, i.e., CPE is PS with degree 1 from the middle figure of Fig 2. This means that the prediction error has been either reduced by increasing the model order or increased by reducing the model order. In either case, the model order may be increased. The linguist label of OUT1 is 4 according to Table 2 and the degree is 1. As a result, the value of OUT1 is 0.4 from the bottom figure of Fig. 2. For the second-level rules, OUT1 = 0.4 corresponds to the label 3 with degree 1 as shown in the top figure of Fig. 3. If ORD = 0.4, i.e., ORD is 3 with degree 1 from the middle figure of Fig. 3, then OUT2 is 4 according to Table 3, whose value is 0.8 after defuzzification from the bottom figure of Fig. 3. The final multiplier to the original order is $0.8\times(1+0.4)=1.12$, i.e., increasing the model order in the next step.

Consider another example where PE = 0.1, i.e., PE is G with degree 1, and $\mathrm{CPE} = -3\times10^{-3}$, i.e., CPE is NS with degree 1. There are two possible scenarios. Either the prediction error becomes worse by increasing the model size, or decreasing the model order has resulted in reducing the prediction error. Thus, the model order must be reduced. From Table 2, OUT1 is 2 with degree 1 whose value is 0 from the bottom figure of Fig. 2. For the second level, OUT1 = 0 corresponds to the linguistic label 1 with degree 1 for the second-level rules from Fig. 3. If ORD = 0.6, i.e., ORD is 4 with degree 1. Then the OUT2 is 5 from Table 3 whose value is 0.9 after defuzzification. The final multiplier becomes $0.9\times(1+0)=0.9$, i.e., decreasing the model size in the next step.

If CPE = 0 and PE is any value, then OUT1 is 2, with a value equal to 0. This implies that OUT1 is 1 for the second-level rules. If ORD is 1, 2 or 3, then OUT2 is 6, with a value close to 1. The final multiplier becomes $1\times(1+0)=1$, i.e., keeping the same model order.

Application of Genetic Algorithm (GA) and Simulated Annealing (SA)

The control weighting used in GPC must be determined before on-line operation. In using fuzzy logic as described above, the control weighting is initially set to a "proper" value such as unity. The proper value is case dependent. Two alternative methods, Genetic Algorithm (GA) and Simulated Annealing (SA), are examined in this section to search for an optimal parameter value based on a defined objective function. Both GA and SA are derivative-free optimization methods, which are suitable for this purpose.

In the regulator problem, the objective function is the summation of the output deviation from a set point over a finite-time duration and a controller stability penalty term. GA and SA methods need a large number of function evaluations, which may result in poor performance during the search. These function evaluations are not practical for on-line experiments. Therefore, an alternative way is to use an identified model instead of the real system to generate the function value. It was observed that identifying a high-order model in combination with a proper model reduction to capture the major dynamic characteristics would produce a proper model to represent the real system.

GA and SA are briefly described here but the reader should refer to Ref. 3 for further details.

Genetic Algorithm

GA encodes each point in the solution space into a binary bit string called a chromosome. Each point is associated with a fitness value, which is usually calculated from objective function. In a generation, GA keeps a set of points, which is called population. To generate the new generation, GA constructs a new population using genetic operators such as crossover and mutation. The members with higher fitness value have better chance to survive and to participate in mating operations. After several generations, the population contains members with better fitness values. The procedures are as follows.

Step 1. Initialization:

(a) Randomly generate the initial population over the solution space.

(b) Evaluate the fitness value of each individual.

Step 2. Elitism: Keep a certain number of best members to the next generation.

Step 3. Mating:

(a) Select two members from the population according to their fitness value.

(b) Apply crossover according to the crossover rate.

(c) Apply mutation according to the mutation rate.

(d) Repeat (a) to (c) until enough members are generated to form the next generation.

Step 4. Repeat steps 2 and 3 until a stopping criterion is met.

Simulated Annealing

Simulated annealing was derived from physical characteristics of spin glasses. The principle behind simulated annealing is similar to the phenomena when metals are cooled at a controlled rate in temperature. It is now commonly used to solve an optimization problem. The solution points are evaluated at the different temperature, which is decreasing during the search. At high temperatures, faraway points may be accepted. The new point may possess a worse function value than that of the previous point which is determined by the acceptance function

$$h(\Delta E, T) = \frac{1}{1 + \exp(\Delta E / (cT))} \tag{13}$$

The quantity c is a system-dependent constant, T is the temperature, and ΔE is the energy difference between two different states x_{new} and x:

$$\Delta E = f(x_{new}) - f(x) \tag{14}$$

where f is the system-dependent objection function. When the temperature decreases to a low temperature, SA evaluates the objective function at the local points. The procedure is as follows.

Step 1. Choose a start point x and set a high temperature T. Evaluate the objective function $f(x)$.

Step 2. Set the new point $x_{new} = x + \Delta x$, where Δx is randomly generated.

Step 3. Calculate the new value of the objective function $f(x_{new})$.

Step 4. (a) Calculate acceptance function $h(\Delta E, T)$

(b) Generate a random probability p.

(c) If $h(\Delta E, T) > p$, then set $x = x_{new}$ and $f(x) = f(x_{new})$.

Repeat (a) to (c) for a set number of times.

Step 5. Reduce the temperature T according to the annealing schedule.

Step 6. Repeat steps 2 to 5 until stopping criterion is met.

Numerical Example

A simplified representation of a mechanical system is used to illustrate various concepts developed in this paper. Several different cases will be discussed for a multi-input/multi-output system.

Consider a nine-degree-of-freedom spring-mass-damper system

$$M\ddot{w} + \Xi\dot{w} + Kw = u$$

where

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_8 \\ w_9 \end{bmatrix}; \qquad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_8 \\ u_9 \end{bmatrix}$$

$$M = \begin{bmatrix} m_1 & 0 & \cdots & 0 & 0 \\ 0 & m_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & m_8 & 0 \\ 0 & 0 & \cdots & 0 & m_9 \end{bmatrix}$$

$$\Xi = \begin{bmatrix} \varsigma_1 + \varsigma_2 & -\varsigma_2 & \cdots & 0 & 0 \\ -\varsigma_2 & \varsigma_2 + \varsigma_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \varsigma_8 + \varsigma_9 & -\varsigma_9 \\ 0 & 0 & \cdots & -\varsigma_9 & \varsigma_9 \end{bmatrix}$$

and

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & \cdots & 0 & 0 \\ -k_2 & k_2 + k_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & k_8 + k_9 & -k_9 \\ 0 & 0 & \cdots & -k_9 & k_9 \end{bmatrix}$$

The quantities m_i, ς_i and k_i for $i = 1, 2, \cdots, 9$ are the mass, damping coefficients, and stiffness, respectively. For this system, the order of the equivalent state-state representation is 18. The control force or disturbance force applied to each mass is denoted by $u_i, i = 1, 2, \cdots, 9$. The variables $w_i, i = 1, 2, \cdots, 9$ are the positions of the nine masses measured from their equilibrium positions. In the simulation, $m_1 = m_2 = \cdots = m_9 = 1\,\text{kg}$, $k_1 = k_2 = \cdots = k_9 = 1000\,\text{N/m}$, and $\varsigma_1 = \varsigma_2 = \cdots = \varsigma_9 = 1\,\text{N-sec/m}$. The system is sampled at 40 Hz. Measurements are assumed to be the accelerations of the nine masses, $y_i = d^2 w_i / dt^2$.

Let us consider two-control-input, single-disturbance-input and three-output case where the control inputs to the system are the forces on the third and ninth masses (i.e., u_3, u_9), the disturbance input is at the second mass (i.e., u_2), and the outputs are the accelerations of the fourth, fifth, and eighth masses (non-collocated actuator-sensor). Therefore, the smallest order of the ARX model is 18.

Assume that the disturbance forcing function is $u_2 = 2\cos(2\pi\omega t)$ with $\omega = 6$ Hz. Applying the disturbance signal u_2, and two independent random signals for u_3 and u_9 to the system would then generate an open-loop simulation. Both random signals are normally distributed. A total of 500 data points were collected for system identification and an initial controller design. To simulate the real system, let the three open-loop outputs be disturbed by some measurement noise so that the signal to noise ratio is 100. The noise is random normally distributed.

Application of Fuzzy Logic

With the open-loop data available, applying fuzzy logic for model tuning produced Table 4. All identified models were obtained by using the two control inputs and the three measured outputs without counting the disturbance. Here the disturbance was assumed unknown. It is known that proper system identification techniques would recover an accurate input/output map in the presence of periodic input disturbances. The quantities PE and CPE were calculated using 500 data points. The CPE value for the first iteration is empty because there is no prediction error available before the first iteration. Careful examination of Table 4 indicates that increasing the order size does indeed improve the prediction accuracy. However, the change in prediction error (CPE) shows less than 1% reduction after the model order 18. Therefore, the model order 18 appears to be a reasonable choice for GPC controller designs. The ARX model order 18 corresponds to the model size 54 (i.e., 54 times the number of outputs) for a state-space representation.

Application of fuzzy logic for tuning the control weighting λ produces Table 5. The initial weighting 1 was arbitrarily chosen. Table 5 shows that it takes about 12 steps for the control weighting to converge to a constant value. On the other hand, it takes only 9 steps for the performance index (PI) to converge. In other words, PI is not as sensitive as the control weighting for the updating process. The maximum control force was set to 15N. The control index (CI) in Table 5 implies that the feedback control force at any time step during the closed-loop control simulation never exceeded 80% of the maximum force allowed. Table 1 for the rules of tuning λ may be modified to adjust the maximum allowable CI to a desired value. Each closed-loop simulation used the control force shown in Eq. (6) with the parameter matrices determined by the input-output data and control weighting computed in the previous step, and input/output data. A total of 500 data points were collected for computing new PI and CI at each step. Note that the objective of the closed-loop simulation is to reduce the response excited by the unknown disturbance input.

In using fuzzy logic, the heuristic rules are established from experience and the weighting λ must be initialized. The resulting GPC controllers may not be stable even though the corresponding closed-loop system is stable. It is known that a stable controller will provide a more robust close-loop system. It is extremely difficult, if not impossible, to generate the heuristic rules to guarantee the stability of a GPC controller, because the explicit relationship between the control weighting and the GPC controller stability is unknown.

Applications of Genetic Algorithm and Simulated Annealing

For the same example, let the objective function for the genetic algorithm be

$$J(\lambda,n)=\frac{1}{\mathrm{PI}}=\frac{\mathrm{RMS}\left[y_{opt}^{T}(k)\,y_{op}(k)\right]}{\mathrm{RMS}\left[y^{T}(k)\,y(k)\right]}$$

The objective is to maximize $J(\lambda,n)$ subject to the constraint that the GPC controller must be stable. The variables (members) to be optimized are the control weighting λ and the ARX order n. For an unstable GPC controller, a negative value may be assigned for $J(\lambda,n)$ to artificially force the chosen variables out of the mating operations. The optimal variables after 200 iterations yielded a control weighting $\lambda=0.00169$, an ARX order $n=16$, and a performance index $\mathrm{PI}=27.1\%$. With each new set of selected variables, a closed-loop simulation was performed to obtain a set of 500 data points to compute the objective function.

The parameters used in this example for the genetic algorithm are the number of iterations 10, the population size 20, the mutation rate of 0.01, crossover rate of 1, number of bits used in variable encoding 16, range of weighting $10^{-3}\leq\lambda\leq 10$, and the range of the ARX order $5\leq n\leq 30$. Without any measurement noise and periodic disturbance, the ARX order 6 is enough to describe the system with 18 degrees of freedom in the state-space representation and 3 measurement outputs. With noise and disturbance present, it is known that the optimal order should be higher to accommodate the computational modes in addition to the system modes. Several simulations have been performed. Since the initial parameters are randomly chosen for each simulation, optimal results are somewhat different. However, they are not far apart from each other. In particular, the PIs are all within the range of 27% to 28%.

The objective function for the application of simulated annealing is

$$J(\lambda,n) = \mathrm{PI} = \frac{\mathrm{RMS}\left[y^T(k)y(k)\right]}{\mathrm{RMS}\left[y_{opt}^T(k)y_{opt}(k)\right]}$$

which is the same as that shown in Eq. (7) used for the application of fuzzy logic. The solution is constrained to yield stable controllers. . A large positive value may be assigned for $J(\lambda,n)$ to accommodate the constraint because SA is used to search for a minimum. The optimal results after 446 iterations yielded a control weighting $\lambda = 0.0014$, identified ARX order $n = 15$, and performance index $\mathrm{PI} = 26.6\%$. The results are similar to the ones obtained using the genetic algorithm. Both optimal GPC controllers computed from GA and SA are indeed stable.

Instead of the ARX order as an input variable, additional control weighting is used such that each control force has its own weighting. Assuming that the order is n=15, the optimal results after 236 iterations are $\lambda_1 = 0.00256, \lambda_2 = 0.00037$, and PI=25.9% where λ_1 and λ_2 are computed for the control forces u_3 and u_9 located at the third mass and the ninth mass, respectively. The weighting for u_9 is one order of magnitude less than the one for u_3. It means that the control force at the free end of the spring-mass-damper system works harder than the other force. The resulting performance is slightly better than other cases.

Concluding Remarks

Three techniques have been developed to determine the basic and most important design variables for a generalized predictive controller. The design variables include the order of the system/controller and the weighting of the control force. Three well-known techniques, fuzzy logic, genetic algorithms (GA), and simulated annealing (SA), are used to derive these techniques to initialize and fine-tune the design variables. The main contributions include the introduction of proper performance indexes and heuristic rules for fuzzy logic, and objective functions for GA and SA. It is believed that the optimization approaches have not yet been introduced in the field of predictive control.

Numerical examples show that fuzzy logic provides a smooth and fast convergence in the sense that the current update is always better than the previous ones in terms of performance. The variables used in the GA and SA algorithms have gone through a rough ride with a considerable number of iterations to converge to the final values. Similarly, the resulting objective function has also jumped around irregularly. For small values of weighting, poor performance (unstable closed-loop response) may occur. It clearly indicates that fuzzy logic is more suitable for real-time adaptive implementations than the GA and SA algorithms. In a very narrow region of the solution space, GA and SA may be feasible for use in real-time/on-line implementation. Fuzzy logic requires an expert to establish rule matrices and membership functions. In addition, it provides no guarantee on the stability of the feedback controllers. Assuming that an input/output map of the system is accurately identified from a set of input and output data and that an internal model of the unknown disturbance is properly estimated, GA and SA may be used to compute a set of "optimal" variables to serve as the initial values, and then fuzzy logic is applied to perform real-time/on-line adaptive control. One unique feature of the GA and SA is that they guarantee the stability of the controller.

References

1. Juang, Jer-Nan, and Eure, Kenneth W., "Predictive Feedback and Feedforward Control for Systems with Unknown Disturbances," NASA/TM-1998-208744, December 1998.
2. Tsoukalas, Lefteri H., and Uhrig, Robert E., Fuzzy and Neural Approaches in Engineering (Adaptive and Learning Systems for Signal Processing, Communications and Control Series, John Wiley & Sons; ISBN: 0471160032, 1997
3. Jang, Jyh-Shing Roger, Sun, Chuen-Tsai, and Mizutani, Eiji, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall; ISBN: 0132610663, 1996, pp. 175-186

Table 1 Rule Matrix for Tuning λ

				CI		
		E	G	A	B	U
	E	6	7	7	8	10
	G	5	6	7	7	9
PI	A	5	5	6	7	8
	B	4	5	5	6	7
	U	1	1	3	5	7

Table 2 Rule Matrix for Order Multiplier

				CPE		
		NM	NS	ZO	PS	PM
	E	2	2	2	3	3
	G	1	2	2	3	3
PE	A	1	1	2	3	4
	B	1	1	2	4	5
	U	1	1	2	4	6

Table 3 Rule Matrix for Adjusting Order Multiplier

				ORD			
		1	**2**	**3**	**4**	**5**	**6**
	1	6	6	6	5	4	3
	2	6	6	5	4	3	2
OUT1	**3**	6	6	4	4	3	1
	4	6	6	4	3	2	1
	5	6	5	3	2	1	1

Table 4: Fuzzy Logic Application for Model Order Tuning

	Model Order	Prediction Error (PE %)	Change in PE (CPE %)
1	6	15.08	?
2	9	9.93	5.14
3	12	8.22	1.71
4	15	5.05	3.17
5	18	3.72	1.33
6	21	2.86	0.86
7	25	2.67	0.19
8	29	2.50	0.17
9	33	2.34	0.17
10	37	2.22	0.12

Table 5: Fuzzy Logic Application for Control Weighting Tuning

	Control Weighting	Performance Index (PI %)	Control Index (CI %)
1	1	62.47	27.48
2	0.4061	48.02	32.89
3	0.2030	39.69	36.86
4	0.1015	33.22	40.64
5	0.0508	30.07	44.63
6	0.0254	27.99	51.84
7	0.0133	26.55	61.46
8	0.0078	25.26	68.84
9	0.0057	25.25	73.06
10	0.0045	24.99	77.28
11	0.0041	24.88	79.51
12	0.0040	24.86	80.06
13	0.0040	24.86	80.03
14	0.0040	24.86	80.00
15	0.0040	24.86	80.00

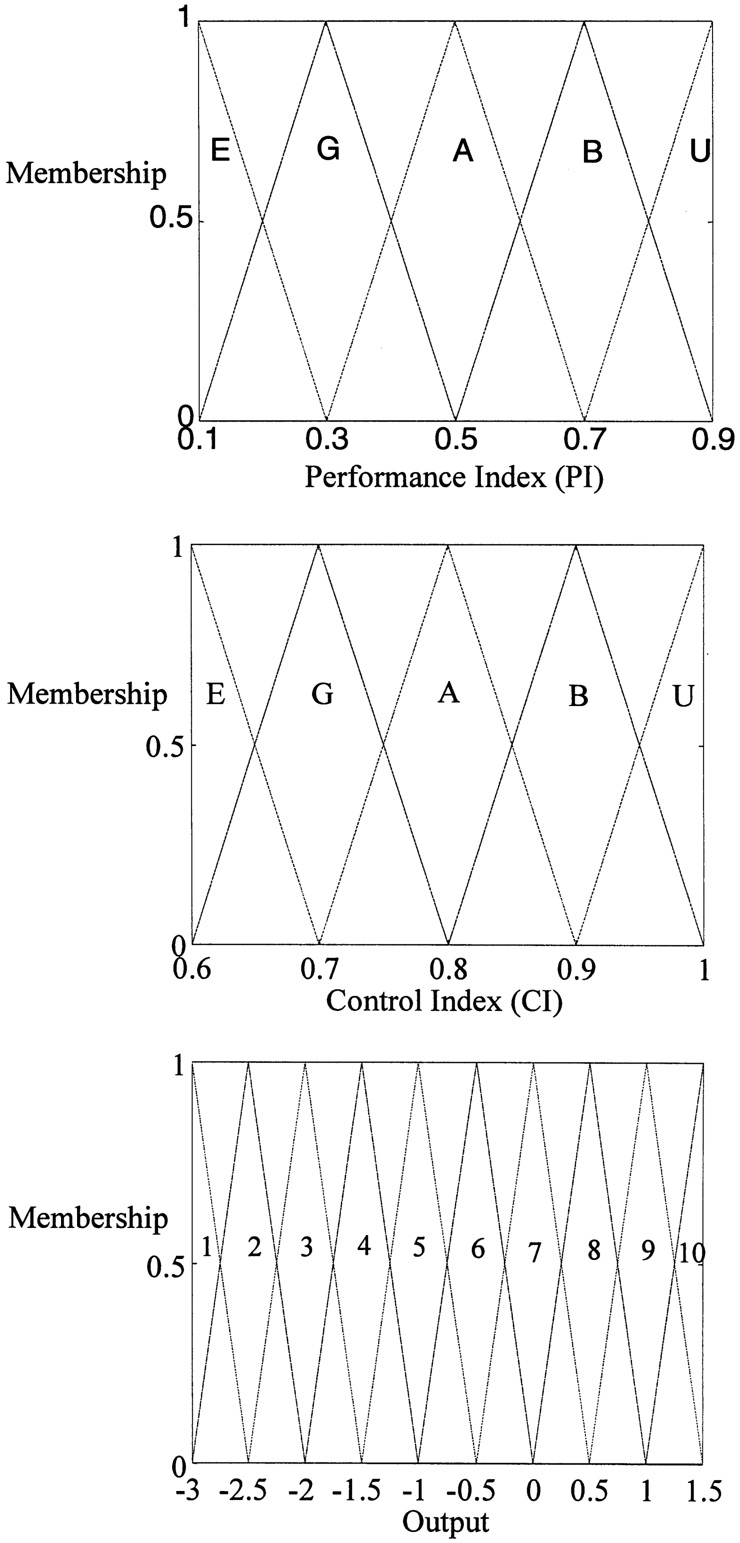

Fig. 1 Membership functions for Tuning λ

(E : Excellent G : Good A : Acceptable B : Bad U : Unacceptable)

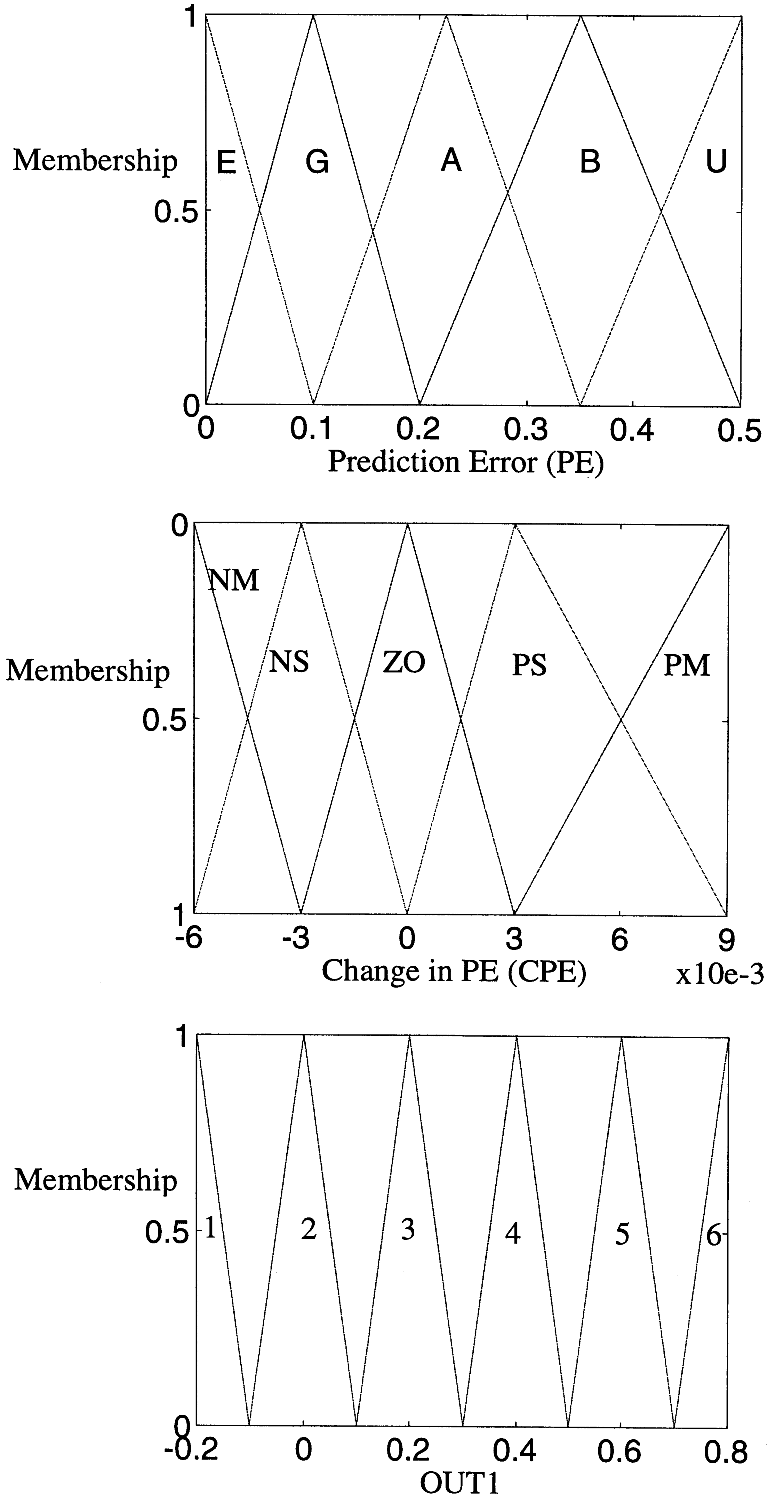

Fig. 2 Membership Functions for Order Multiplier
(E : Excellent G : Good A : Acceptable B : Bad U : Unacceptable
N : Negative P : Positive S : Small M : Medium ZO : Zero)

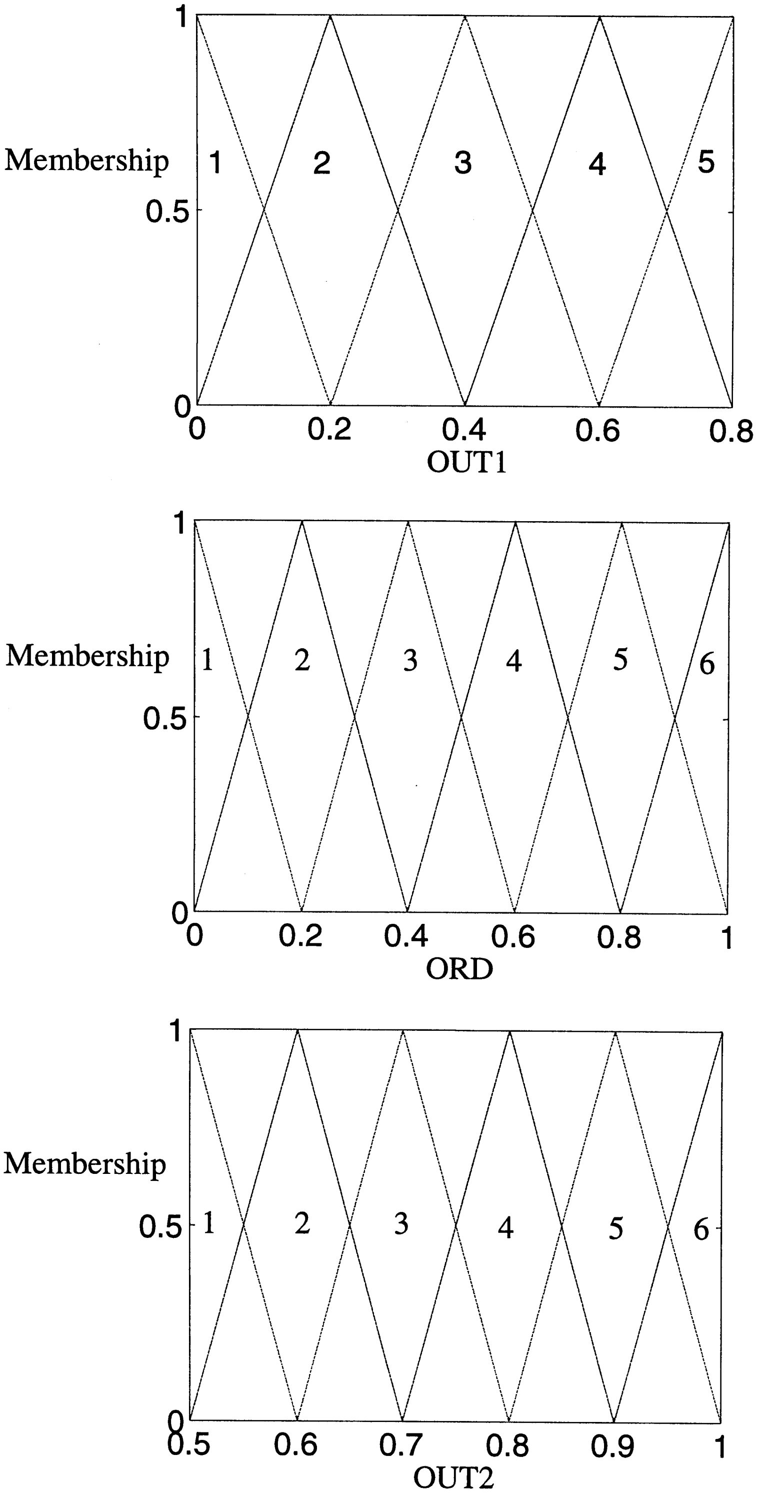

Fig. 3 Membership Functions for Adjusting Multiplier

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
December 2000

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Self-Tuning of Design Variables for Generalized Predictive Control

5. FUNDING NUMBERS
WU 632-10-14-04

6. AUTHOR(S)
Chaung Lin and Jer-Nan Juang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L-17969

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/TM-2000-210619

11. SUPPLEMENTARY NOTES
Lin: Institute for Computer Applications in Science and Engineering, Hampton, VA; Juang: NASA Langley Research Center, Hampton, VA

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 39 Distribution: Nonstandard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Three techniques are introduced to determine the order and control weighting for the design of a generalized predictive controller. These techniques are based on the application of fuzzy logic, genetic algorithms, and simulated annealing to conduct an optimal search on specific performance indexes or objective functions. Fuzzy logic is found to be feasible for real-time and on-line implementation due to its smooth and quick convergence. On the other hand, genetic algorithms and simulated annealing are applicable for initial estimation of the model order and control weighting, and final fine-tuning within a small region of the solution space. Several numerical simulations for a multiple-input and multiple-output system are given to illustrate the techniques developed in this paper.

14. SUBJECT TERMS
Predictive control; Fuzzy logic; Genetic algorithms; Simulated annealing

15. NUMBER OF PAGES
26

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL