# APPLICATION OF SELF-TUNING CONTROLLER USING POLE ASSIGNMENT METHOD IN CONTROLLING ELECTRIC OVEN

Nguyen Duc Hoang and Bui Thanh Huyen

Automatic Control Department, Electrical and Electronic Faculty
Ho Chi Minh City University of Technology

**ABSTRACT**

The paper presents the process of the design of self-tuning controller using pole assignment method in controlling electric oven. At first, acquiring data of the oven (supplied power and temperature) in order to find the model structure of the oven. On the basic of the obtained model, the parameters of the oven will be online identified. Afterward, these parameters will be used to design controller using pole assignment method. Thus, when the parameters of the oven change, the control parameters will change correspondingly so that the final performance is as desired.

MatLab, Simulink and Realtime Workshop are used to implement the controller in realtime.
**Keyword:** system identification, pole assignment, STR.

## 1. INTRODUCTION

The majority of processes met in industrial practice have stochastic character, *viz*. the output at time *t* cannot be exactly determined from I/O data at time *t-1*. Traditional controllers with fixed parameters are often unsuited to such processes because their parameters change. Parameter changes are caused by changes in the manufacturing process, in the nature of the input materials, fuel, machinery use (wear) *etc.* Fixed controllers cannot deal with this. One possible alternative for improving the quality of control for such processes is the use of adaptive control systems. [1].

In an adaptive system it is assumed that the regulator parameters are adjusted all the time. This implies that the regulator parameters follow changes in the process. However, it is difficult to analyze the convergence and stability properties of such system. To symplify the problem it can be assumed that the process has constant but unknown parameters. When the process is known, the design procedure specifies a set of desired controller parameters. The adaptive controller should converge to these parameter values even when the process is unknown. A regulator with this property is called *self-tuning*, since it automatically tunes the controller to the desired performance. The *self-tuning regulator* (STR) is based on the idea of seperating the estimation of unknown parameters from the design of the controller. [2]. The basic idea is illustrated in Fig.1.

The unknown parameters are estimated on-line, using a recursive estimation method. The estimated parameters are treated as if they are true; i.e.,the uncertainties of the estimates are not considered. This is called *certainty equivalence principle.* Many different estimation schemes can be used, such as stochastic approximation, least squares, extended and generalized least squares, instrumental variable, and maximum likelihood. The design method is chosen depending on the specifications of the closed loop system. [2].

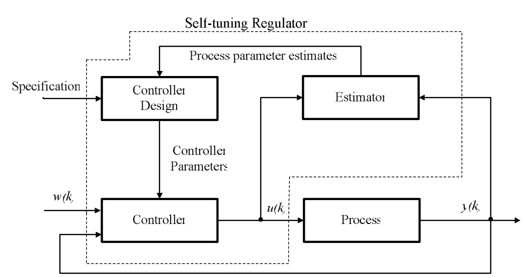


Fig.1 Block Diagram of Self-tuning Regulator

## 2. ON-LINE PARAMETER ESTIMATION

### 2.1 Introduction

In many applications, plant (model) structure may be known, but its parameters may be unknown and time-varying due to change in operation conditions, aging of equipment, etc. Thus, the off-line parameter estimation is inefficient.

On-line estimation schemes refer to those estimation schemes that provide frequent estimates of plant parameters by properly processing the plant I/O data on-line. The essential idea behind is the comparison of the observed system response $y(k)$ with the output of a parameterized model $\hat{y}(\theta,k)$ , whose structure is the same as that of plant model. Then, $\theta(k)$ is adjusted continuous so that $\hat{y}(\theta,k)$ approaches $y(k)$ as $k$ increases.(Under certain input conditions, $\hat{y}$ being close to $y$ implies that $\theta(k)$ is close to the unknown $\theta^*$.)

The on-line estimations procedure, therefore, involves 3 steps:

- Select an appropriate plant parameterization.
- Select an adaptive law for generating or updating $\theta(k)$.
- Design the plant input so that $\theta(k)$ approaches $\theta^*$ as $k \to \infty$.

**Remark:** In adaptive control, where the convergence of $\theta(k)$ to $\theta^*$ is usually not one of the objectives, the first two steps are the most important ones.

### 2.2 Model Structures

When systems cannot be modeled with use of physical principles. The reason for this may be lack of information about the systems function. Another case would be when the physical relationships are too complex to unreal. The remedy for this is to use standard models, which by experence, are able to handle many cases in dynamic systems. The most common class of such standard models is linear system.

A general time discrete model parameterized by $\theta$ can be written as

$$y(k) = \mathrm{G}(q^{-1},\theta)u(k) + \mathrm{H}(q^{-1},\theta)e(k) \qquad (1)$$

where

$q^{-1}$ : back shift differential operator

$q^{-1}y(k) = y(k-1)$

$\mathrm{G}(q^{-1},\theta)$ : system model

$\mathrm{H}(q^{-1},\theta)$ : noise model

$e(k)$ : white noise.

In black-box modeling with prediction error methods the following general model structure is often used:

$$\mathrm{A}(q^{-1})y(k) = \frac{\mathrm{B}(q^{-1})}{\mathrm{F}(q^{-1})}u(k) + \frac{\mathrm{C}(q^{-1})}{\mathrm{D}(q^{-1})}e(k) \qquad (2)$$

where

$$\mathrm{A}(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \ldots + a_{na}q^{-na}$$

and analoguos for the $C$, $D$ and $F$ polynomials, while

$$\mathrm{B}(q^{-1}) = b_1q^{-1} + b_2q^{-2} + \ldots + b_{nb}q^{-nb}$$

Table 1 shows common model structures that are special cases of (2) [3].

Table 1 Common model structures

| Polynomial used | Name of the model structure |
|---|---|
| B | FIR (Finite Impulse Response) |
| A,B | ARX |
| A,B,C | ARMAX |
| B,F | OE(Output Error) |
| B,C,D,F | BJ(Box-Jenkins) |

## 3. SELF-TUNING CONTROLLER

### 3.1 Recursive Extended Least Squares (RELS(PR)) (A Prior Prediction Errors)

The least-square method is commonly used in system identification. Its principle is that the unknown parameters of a mathematical model should be chosen by minimizing the sum of the square of the difference between the actually observed and the analyically predicted output values with possible weighting that measure the degree of precision. The least-squares criterion is quadratic, so an analytic solution to the least-squares problem exists as long as the measured variable is linear in the unknown parameters.

It is assumed that the process is described by the single-input, single output (SISO) system

$$\mathrm{A}(q^{-1})y(k) = q^{-d}\mathrm{B}(q^{-1})u(k) + \mathrm{C}(q^{-1})e(k) \qquad (3)$$

where

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + ... + a_{na} q^{-na}$$
$$B(q^{-1}) = b_1 q^{-1} + b_2 q^{-2} + ... + b_{nb} q^{-nb}$$
$$C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2} + ... + c_{nc} q^{-nc}$$

with *na, nb, nc* are the order of the polynomials A, B, C respectively and *d* is time delay .

This is an ARMAX model.

The model is linear in the parameters and can be written in the vector form as

$$y(k) = \theta_{k-1}^T \Phi_{k-1}^e + e(k) \qquad (4)$$

where

$$\theta_{k-1} = \left[\hat{a}_1, \hat{a}_2, ..., \hat{a}_{na}, \hat{b}_1, \hat{b}_2, ..., \hat{b}_{nb}, \hat{c}_1, \hat{c}_2, ..., \hat{c}_{nc}\right]^T$$
$$\Phi_{k-1}^e = \left[-y_{k-1}, -y_{k-2}, ..., -y_{k-na}, u_{k-d-1}, u_{k-d-2}, ..., u_{k-d-nb}, e_{k-1}, e_{k-2}, ..., e_{k-nc}\right]^T$$

with

$\theta_{k-1}$ : estimation parameter vector.

$\Phi_{k-1}^e$ : regression vector.

If $\Phi_{k-1}^e$ were available, the RLS(Recursive Least Squares) algorithm could be used to recursively estimate $\theta$. In reality, all of $\Phi_{k-1}^e$ is known except for its last *nc* components. In RELS(PR) such components are replaced by using the *a prior* prediction errors.

$$\varepsilon_k = y_k - \theta_{k-1}^T \Phi_{k-1} \qquad (5)$$

where $\Phi_{k-1}$ is given here by the pseudo-regressor

$$\Phi_{k-1} = \left[-y_{k-1}, -y_{k-2}, ..., -y_{k-na}, u_{k-d-1}, u_{k-d-2}, ..., u_{k-d-nb}, \varepsilon_{k-1}, \varepsilon_{k-2}, ..., \varepsilon_{k-nc}\right]^T$$

The algorithm to identify $\theta$ is given in Table 2.

Table 2 Fomulations to estimate parameter

$$\theta_k = \theta_{k-1} + \frac{C_{k-1}\Phi_{k-1}}{\varphi + \Phi_{k-1}^T C_{k-1} \Phi_{k-1}} (y_k - \theta_{k-1}^T \Phi_{k-1}) \qquad (6)$$

$$C_k = \frac{1}{\varphi}\left[C_{k-1} - \frac{C_{k-1}\Phi_{k-1}\Phi_{k-1}^T C_{k-1}}{\varphi + \Phi_{k-1}^T C_{k-1}\Phi_{k-1}}\right] \qquad (7)$$

$\theta$ is identified by minimizing the quaratic index $J_k = \frac{1}{2}\sum_{i=1}^{k} \varphi^{k-i} (y_i - \theta_k^T \Phi_{i-1})^2$ .

The RELS algorithm above can be interpreted intuitively. The estimate $\theta_k$ is obtained by adding a weighted prediction error term $y_k - \theta_{k-1}^T \Phi_{k-1}$ to the previous estimate $\theta_{k-1}$ . The term $\theta_{k-1}^T \Phi_{k-1}$ can be viewed as the value of *y* at time *k* predicted by the model (4) with the previous estimates $\theta_{k-1}$. The parameter $\varphi$ is called the forgetting factor and is usually chosen in range $0.95 < \varphi < 1$. The choice of $\varphi$ depends on how the properties of the system change. Smaller values of $\varphi$ result in a faster forgetting, which can be used to cope with nonlinear and fast changing systems. Likewise values of $\varphi$ close to 1 result in slower forgetting, and can be used for systems that change gradually.

The symmetric convariance matrix $C_k$ is defined by $C_k = \left(\sum_{i=1}^{k} \Phi_{i-1}\Phi_{i-1}^T\right)^{-1}$ with the initial condition positive $C(0) = C_0$ definite. By this definition, it is easy to see that $C_k \simeq \left(C_0^{-1} + \sum_{i=1}^{k} \Phi_{i-1}\Phi_{i-1}^T\right)^{-1}$ Notice that $C_k$ can be made arbitrarily close to $\left(\sum_{i=1}^{k} \Phi_{i-1}\Phi_{i-1}^T\right)^{-1}$ by choosing $C_0$ sufficiently large [2].

### 3.2 Two-degree-of-freedom controller

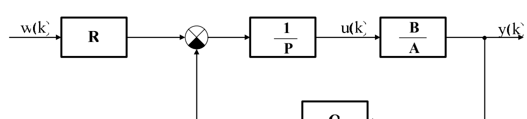


Fig.2 Two-degree-of-freedom controller

The process model is described as in (3)

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k) + C(q^{-1})e(k)$$

Assume that the polynomials $A(q^{-1})$ and $B(q^{-1})$ are co-prime, i.e. they do not have any common factors. Furthermore, $A(q^{-1})$ is monic. That is, that the coefficient of the highest power is unity.

A general linear controller can be described by

$$P(q^{-1})u(k) = R(q^{-1})w(k) - Q(q^{-1})y(k) \quad (8)$$

where

$$P(q^{-1}) = p_0 + p_1q^{-1} + p_2q^{-2} + ... + p_{np}q^{-np}$$
$$Q(q^{-1}) = q_0 + q_1q^{-1} + q_2q^{-2} + ... + q_{nq}q^{-nq}$$
$$R(q^{-1}) = r_0 + r_1q^{-1} + r_2q^{-2} + ... + r_{nr}q^{-nr}$$

*np*, *nq*, *nr* are the order of the polynomials P, Q, R respectively.

This controller consists of a feedforward with the transfer operator $\frac{R(q^{-1})}{P(q^{-1})}$ and a feedback with the transfer operator $\frac{Q(q^{-1})}{P(q^{-1})}$. It thus has two degrees of freedom. A block diagram of the closed-loop system is illustrated in the Fig.2.

Eliminate $u(k)$, we can obtain the following equations for the closed loop system.

$$y(k) = \frac{BR}{AP+BQ}w(k) + \frac{CP}{AP+BQ}e(k) \quad (9)$$

The closed loop characteristic polynomial is thus

$$A_c = AP+BQ \quad (10)$$

This equation is known as the Diophantine equation or the Benzout identify and it plays a central role in many aspects of modern control theory.

The desired closed loop response is:

$$y_m(k) = \frac{B_m}{A_m}w(k) \quad (11)$$

Then, we have ( assume $e(k) = 0$ ):

$$\frac{BR}{AP+BQ} = \frac{B_m}{A_m}$$

Generally deg(AP + BQ) > deg($A_m$). It means that BR and AP + BQ have a common factor $A_0$. As it is desirable to cancel only stable, thus decompose B as $B = B^+B^-$

where

$B^+$ : contains stable zeros that can be cancelled

$B^-$ : contains unstable zeros that should not be cancelled

Then

$$\frac{BR}{AP+BQ} = \frac{B_m}{A_m} \Leftrightarrow \frac{B^+B^-R}{AP+B^+B^-Q} = \frac{A_0B_m}{A_0A_m}$$

It follows that $P = P_1B^+$

Thus

$$AP_1 + B^-Q = A_0A_m \quad (12)$$

Causality of the controller imposes

$$\deg(Q) \le \deg(P)$$
$$\deg(R) \le \deg(P)$$
$$\deg(A_m)-\deg(B_m) \ge \deg(A)-\deg(B)$$
$$\deg(A_0) \ge 2\deg(A)-\deg(B^+)-\deg(A_m)-1$$

To simplify, here only represent pole placement design with no zeros cancelled algorithm.

### 3.3 Algorithm

Data: Polynomials A and B.
Specifications: Polynomials $A_m$, $B_m$, and $A_0$.
Compatibility conditions:

$$B \text{ divides } B_m$$
$$\deg(A_m)-\deg(B_m) \ge \deg(A)-\deg(B)$$
$$\deg(A_0) \ge 2\deg(A) - \deg(A_m) - 1$$

Step 1. Solve $AP + BQ = A_0A_m$

Step 2. Form $R = \frac{B_m}{B}A_0$

Step 3. The control law is

$$P(q^{-1})u(k) = R(q^{-1})w(k) - Q(q^{-1})y(k)$$

## 4. OFF-LINE IDENTIFICATION

### 4.1 Data Collection

The purpose of the off-line identification is to find a model structure of the oven. Two data sets are collected from the system, model and validation data. The model data is used for model estimation. The validation data is only used for comparison with predicted data from an estimated model. The model and validation data are collected in the open-loop system. To be able to determine an accurate model of a system, it is necessary to excite as much information as possible from the process. The solution is to use an input signal with a vast frequency content. In the work to be presented here, a *chirp signal,* a sine wave with increasing frequency, is used as input signal.

Using a chirp signal with frequencies between 0 and 0.2 Hz and amplitude 5 V to collect model data. Besides another chirp signal with frequencies between 0 and 0.01 Hz and amplitude 5 V to collect validation data. See the plots below.

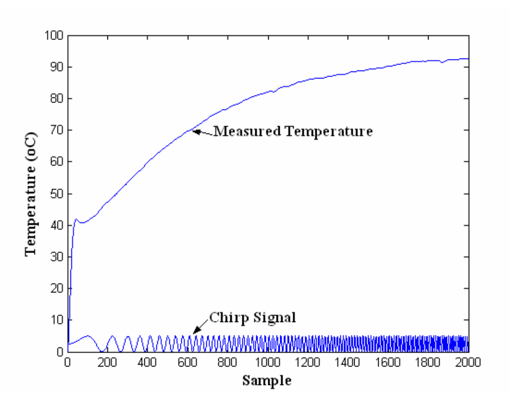


Fig.3. Model Data

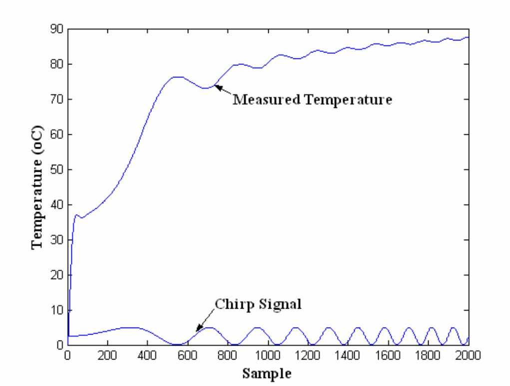


Fig.4. Validation Data

### 4.2 System Identification

The System Identification Toolbox for Matlab is used to find the model structure of the oven. ARMAX model is used in this section. The experimental results argue that ARMAX model *amx3131* give the best result, i.e. model has 3 poles and 2 zeros.

The fit or the multiple correlation coefficient is defined as

$$fit\% = (1 - \frac{\sum (y(k) - \hat{y}(k))^2}{\sum y^2(k)}).100 \quad (13)$$

where $y(k)$ is the measured value and $\hat{y}(k)$ is the predicted value.

Using data in the Fig.4 as validation data gives $fit = 81.67\%$ and Fig.5 compares validation data and prediction data.

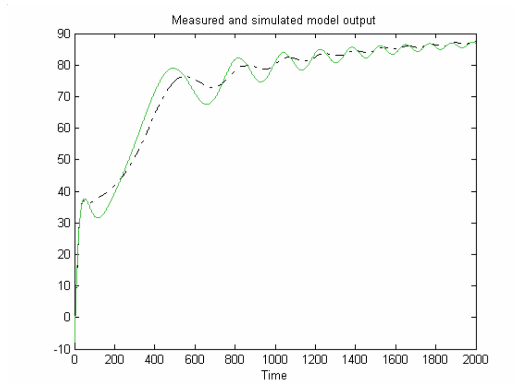


Fig.5. Model Prediction. Notation: validation data (dotted), predicted data (solid)

### 5. IMPLEMENTED RESULTS

Set reference temperature be 80°C, obtain result as Fig.6. Here, reference temperature is added a PRBS(Pseudo Random Binary Signal) having amplitude ± *0.15* and frequency *0.05rad/s* to increase information in input. Besides the other parameters are chosen as : $T_0 = 1s$, $\varphi = 1$, $\omega = 0.02 rad/s$, $\zeta = 0.85$, $A_o = 1$.

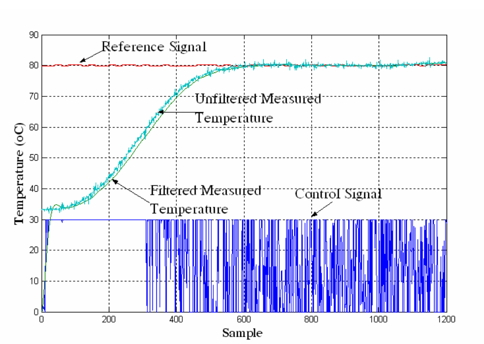


Fig.6. Implemented result in the realtime

The estimated parameters of the oven are given in Table 3 and Fig.7a, Fig.7b.

Table 3 Values of parameters

| Parameters | Predicted Values | Estimated Values |
|---|---|---|
| a1 | -2.9442 | -2.9434 |
| a2 | 2.8899 | 2.8967 |
| a3 | -0.9457 | -0.9533 |
| b1 | 3.8467e-4 | 4.2184e-5 |
| b2 | 0 | 4.9162e-4 |
| b3 | 0 | -1.2942e-4 |

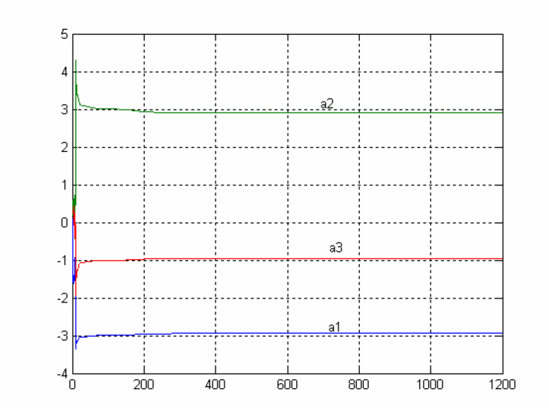


Fig.7a. Estimated parameters of a

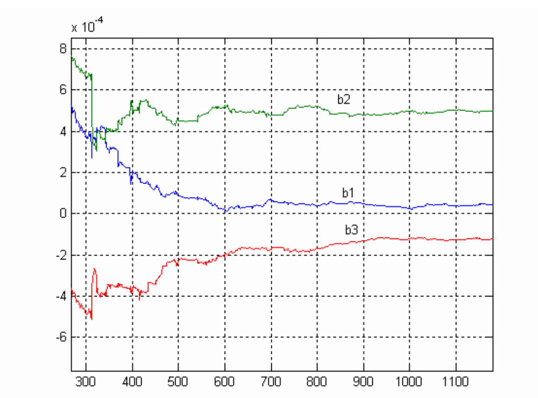


Fig.7b. Estimated parameters of b

**Remarks**:

- Estimated values approximate predicted values and converge fast.
- Performance of controlled process is good, non-overshoot, fast steady state and small steady state error.

## REFERENCES


1. Bobál, V.; Böhm, J.; Fessl, J.; Machácek, J., Digial self-tuning controller – Algorithms, Implementation and Aplications, Springer, 2005, chapter 2.
2. K.J. Åström & B. Wittenmark, Adaptive Control, Addison-Wesley, 1989.
3. Ulf Pettersson, Identification and Adaptive Control of a Coordinate Measuring Machine, Linköping University Electronic Press, 2004, chapter 4,5.
4. Bobál, V.; Böhm, J., Self-tuning controllers simulink library, Tomas Bata University in Zlin, Czech Republic, 2001.