

Online Identification and Control of an Electric Oven using Self-Tuning Regulator

Thanh Huyen Bui
Faculty of Electrical and Electronics Engineering
HCMC University of Technology
Ho Chi Minh City, Vietnam

ABSTRACT

The paper presents the process of the design of self-tuning regulator (STR) using pole assignment method in controlling electric oven. First, data of the oven including supplied power and temperature is acquired in order to find the model structure of the oven. On the basis 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 Real-Time Workshop are used to implement the controller in real time.

General Terms

Online Identification, Control.

Keywords

System Identification, pole assignment, STR, Real time.

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 simplify this 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 separating 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].

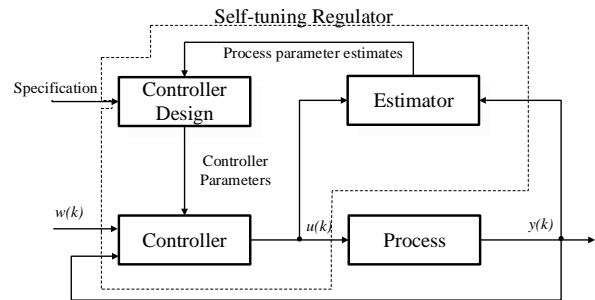


Fig. 1. Block Diagram of Self-Tuning Regulator

In this paper, methods to identify and design controller to regulate temperature of oven will be presented.

The rest of this paper is organized as follows: section 2 introduces online parameter identification, while section 3 describes design of self-tuning regulator. Section 4 presents methods to identify offline model of oven. Section 5 presents results implemented on real oven and the final section is conclusions.

2. ONLINE 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 changes in operation conditions, aging of equipment, etc. Thus, the offline parameter estimation is inefficient.

Online estimation schemes refer to those estimation schemes that provide frequent estimates of plant parameters by properly processing the plant I/O data online. 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 continuously so that $\hat{y}(\theta, k)$ approaches to $y(k)$ when k increases. (Under certain input conditions, \hat{y} is close to y implies that $\theta(k)$ is close to the unknown θ^*).

The online estimations procedure, therefore, involves three following 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 to θ^* when $k \rightarrow \infty$.

Remark: In adaptive control, where the convergence of $\theta(k)$ to θ^* 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 experience, 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 θ can be written as follows:

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

Where:

q^{-1} : back shift differential operator

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

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

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

$e(k)$: white noise.

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

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

Where:

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

and similar for the C, D and F polynomials, while

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

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 REGULATOR

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 analytically 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:

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

Where:

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

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

$$C(q^{-1}) = 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_{nc}q^{-nc}$$

with n_a, n_b, n_c 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 follows:

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

Where:

$$\theta_{k-1} = [\hat{a}_1, \hat{a}_2, \dots, \hat{a}_{na}, \hat{b}_1, \hat{b}_2, \dots, \hat{b}_{nb}, \hat{c}_1, \hat{c}_2, \dots, \hat{c}_{nc}]$$

$$\Phi_{k-1}^e = [-y_{k-1}, -y_{k-2}, \dots, -y_{k-na}, u_{k-d-1}, u_{k-d-2}, \dots, u_{k-d-nb}, e_{k-1}, e_{k-2}, \dots, e_{k-nc}]^T$$

with

θ_{k-1} : estimation parameter vector.

Φ_{k-1}^e : regression vector.

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

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

where Φ_{k-1} is given here by the pseudo-regressor:

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

The algorithm to identify θ is given in formulas 6 and 7.

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

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

θ is identified by minimizing the quadratic 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 θ_k is obtained by adding a weighted prediction error term $y_k - \theta_{k-1}^T \Phi_{k-1}$ to the previous estimate θ_k . 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 θ_{k-1} . The parameter φ is called the forgetting factor and is usually chosen in range $0.95 < \varphi < 1$. The choice of φ depends on how the properties of the system change. Smaller values of φ result in a faster forgetting, which can be used to cope with nonlinear and fast changing systems. Likewise values of φ close to 1 result in slower forgetting, and can be used for systems that change gradually.

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

3.2 Two-degree-of-freedom controller

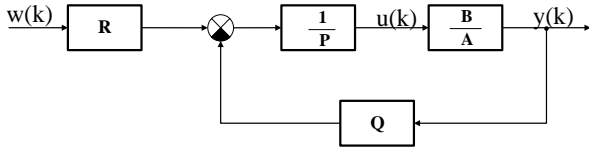


Fig. 2. Two-degree-of-freedom controller

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, 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:

$$P(q^{-1}) = p_0 + p_1q^{-1} + p_2q^{-2} + \dots + p_{np}q^{-np}$$

$$Q(q^{-1}) = q_0 + q_1q^{-1} + q_2q^{-2} + \dots + q_{nq}q^{-nq}$$

$$R(q^{-1}) = r_0 + r_1q^{-1} + r_2q^{-2} + \dots + r_{nr}q^{-nr}$$

n_p, n_q, n_r 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.

Eliminating $u(k)$, the following equations can be obtained for the closed loop system.

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

Thus, the closed loop characteristic polynomial is:

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

This equation is known as the Diophantine equation or the Bézout's identity 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, (assumed $e(k) = 0$):

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

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:

$$\deg(Q) \leq \deg(P), \deg(R) \leq \deg(P)$$

$$\deg(A_m) - \deg(B_m) \geq \deg(A) - \deg(B)$$

$$\deg(A_0) \geq 2\deg(A) - \deg(B^+) - \deg(A_m) - 1$$

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 divides B_m

$$\deg(A_m) - \deg(B_m) \geq \deg(A) - \deg(B)$$

$$\deg(A_0) \geq 2\deg(A) - \deg(A_m) - 1$$

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. OFFLINE 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.

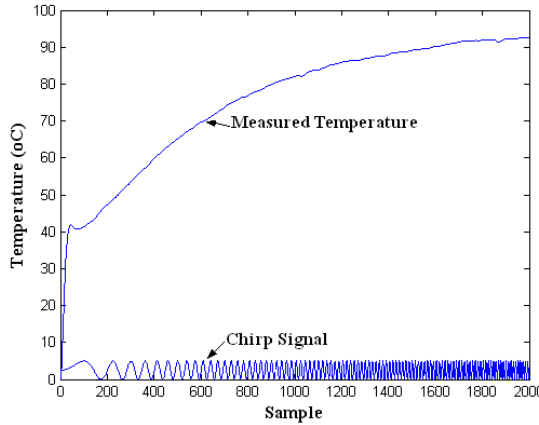


Fig. 3. Model Data

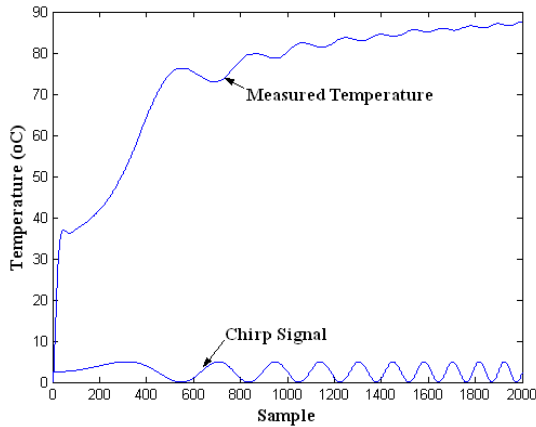


Fig. 4. Validation Data

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:

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

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.

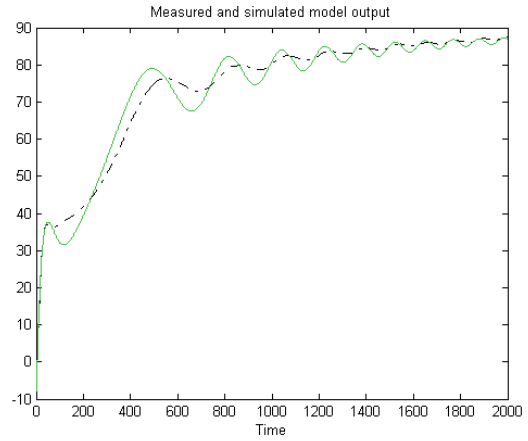


Fig. 5. Model Prediction. Notation: validation data (dotted), predicted data (solid)

5. IMPLEMENTED RESULTS

Set reference temperature be $80^{\circ}C$, obtained 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: $T_0 = 1s, \varphi = 1, \omega = 0.02rad/s, \xi = 0.85, A_0 = 1$.

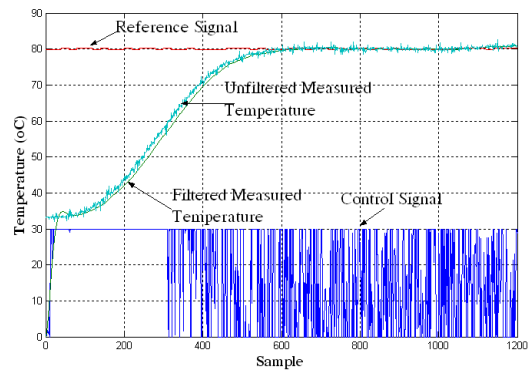


Fig. 6. Implemented result in the real time

The estimated parameters of the oven are given in Table 2 and Fig.7, Fig.8.

Table. 2. Values of parameters

Parameters	Predicted Values	Estimated Values
a_1	-2.9442	-2.9434
a_2	2.8899	2.8967
a_3	-0.9457	-0.9533
b_1	$3.8467e - 4$	$4.2184e - 5$
b_2	0	$4.9162e - 4$
b_3	0	$-1.2942e - 4$

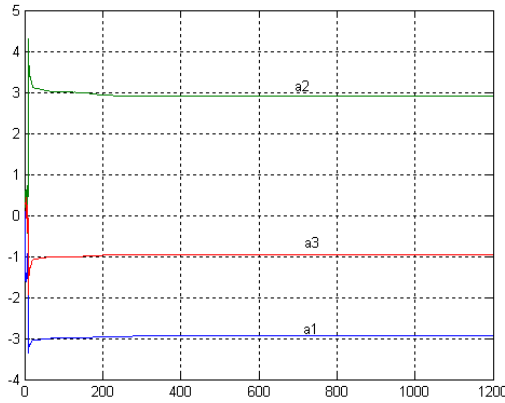


Fig. 7. Estimated parameters of a

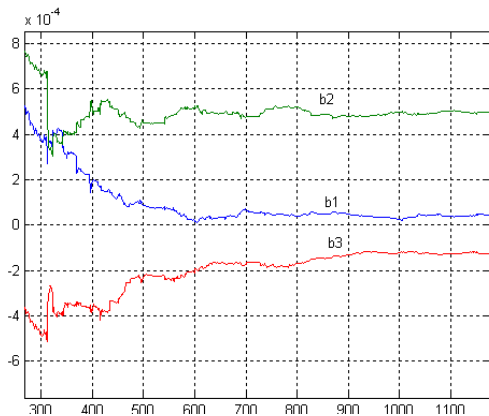


Fig. 8. Estimated parameters of b

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.

6. CONCLUSION

In this paper, the model structure of the oven plant is identified using offline acquired data. On the basis of the obtained model, the parameters of the oven will be online identified. Then, these parameters will be used to design controller using pole assignment method. Results are implemented on oven in real time have demonstrated that estimated parameters approximate predicted ones, performance of controlled process is good. The future work is to validate the effects of forgetting factor on online identification process as well as control performance. Besides, this method is also used to identify and control nonlinear objects.

7. REFERENCES

- [1] Bobál, V.; Böhm, J.; Fessl, J.; Macháček, J. “Digital self-tuning controller – Algorithms, Implementation and Applications”, 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.