Iterative Learning Control Based on Inverse Process Model

Kwang Soon Lee

Dept. of Chem. Eng., Sogang University

ABSTRACT

An iterative learning control scheme is newly designed in the frequency domain. Purposing for batch process control, a generic form of feedback-assisted first-order learning scheme is considered first, and the inverse model-based learning algorithm is derived through convergence analysis in the frequency domain. To enhance the robustness of the proposed scheme, a filtered version is also presented.

Performance of the proposed scheme is evaluated through numerical simulations.

INTRODUCTION

The iterative learning is rather a new control concept which has been developed as a teaching mechanism for robot manipulators. With only limited knowledge of the process, the learning algorithm calculates an updated control signal from the past operation records such that the tracking error is reduced as operation undergoes iteration.

The concept of the learning control had been presented by many researchers with many different forms, but was elaborated as a formal theory by Arimoto and his associates (1984). After the contribution of Arimoto et al.(1984), many different learning control methods have been proposed (Togal and Yamano, 1985; Gu and Loh, 1987; Bondi et al., 1988; Kawamura et al., 1988; Bien and Huh, 1989) including the first-order, the higher-order, the PID-type, and the other learning algorithms. The usual approach of the existing works is to presume learning control scheme first and then to find the convergence conditions associated with the process models described in the time domain. In addition, it has been generally assumed that both the process output (position) and its derivative (velocity) are measurable, which is true in mechanical systems but not true in chemical processes. As a consequence, the proposed learning algorithms have only restricted types and the resulting convergence conditions are hard to test in chemical process control, where the transfer function models are extensively used.

In the present work, intending utilization for control of chemical batch processes where various disturbances are introduced in an unexpected manner, the existing open loop learning algorithm is modified first by combining feedback control for disturbance rejection. Starting from a generic formulation of the feedback-assisted first-order learning control scheme, an inverse model-based learning algorithm is proposed through convergence analysis in the frequency domain. This basic learning control scheme is then further modified to a filtered version to enhance the robustness. Finally, performance of the proposed scheme is evaluated through numerical simulations.

PROCESS DESCRIPTION

The batch process usually deals with multiple grades of multiple products based on recipes. For a certain grade of a certain product, however, the same recipe is invoked at the respective batch, so the operation is repetitive.

Suppose that a batch process for a given production recipe is described by a SISO transfer function model as follows:

$$y = Pu + d + w \qquad (1)$$

Variables in eq. (1) may be functions of s or z according to the domain under consideration. In this model, u and y denote the maipulated and process variables, respectively; both d and w stand for unknown disturbances, but d denotes a repetitive one which occurs with the same pattern in each batch whereas w is a random perturbation including unrested initial states. An example of repetitive disturbances is the heat of reaction in a batch reactor. Although the heat of reaction may have different profiles from batch to batch, the difference is usually slim and a good part of the profile is reproduced. Let the subscript k denote the k-th batch. Then d satisfies

$$d_k = d_{k-1} = d \qquad (2)$$

OBJECTIVE OF THE LEARNING CONTROL

Now we assume that only an approximate description of the process is available. Incorporating a further assumption that w is absent, the objective of the learning control is to progressively achieve a perfect tracking as the batch number increases while compensating the effects of the repetitive disturbance completely. In mathematical terms, the objective can be written as

$$\lim_{k\to\infty} \| y_k - r \| = 0 \qquad (3)$$

for an appropriate norm definition.

BASIC SCHEME OF THE FEEDBACK-ASSISTED ITERATIVE LEARNING CONTROL

Perfect control is virtually impossible in real-time control of continuous processes when the control signal is calculated based on the real time measurements of the process variables. If the exact process model is available and all the disurbances are measurable, perfect control can be achieved at least theoretically. Even in this case, however, the compensators may appear as differentiators and/or noncausal ones which are physically unrealizable.

In control of batch processes, somewhat different but favorable operational situation is afforded in view of the controller design, in that the same operation is repeated in each bach and there is a sufficient interval between batches. This enables us to assess the control performance of the previous batch and to determine an updated control signal which may reduce the control error in the next batch. For example, filtering the output error in the previous batch with an inverse process model will yield the input error signal. Biasing the previous control input by the amount of the input error signal will give a corrected control input. Realizability of the inverse process model in view of causality and/or differentiability does not matter because the calculation is made for the previous batch records.

To generalize the above idea, we consider the following compensator in which all the available variables in the previous batch are combined linearly plus a output feedback loop for real-time rejection of random disturbances.

$$u_k = Fu_{k-1} + Er - H_1 y_{k-1} - H_0 y_k \tag{4}$$

In the above equation, F, E, H_0 and H_1 are transfer functions. Except H_0, they need not be proper and/or causal because their operations are taken on preacquisited or prespecified signals. Multiplying P on both sides of eq. (3) and then replacing Pu with $y-d-w$ gives the following recursion equation.

$$[1+PH_0]y_k = [F-PH_1]y_{k-1} + PEr + [1-F]d + [w_k - Fw_{k-1}] \tag{5}$$

Assume that w is zero and $\{y_k\}$ converges. Then the limit y_∞ is

$$y_\infty = \frac{1}{1-F+P[H_0+H_1]}[PEr + [1-F]d] \tag{6}$$

If we choose

$$F = 1 \quad \text{and} \quad E = H_0 + H_1 \tag{7}$$

the effect of d is self-eliminated and y_∞ is equal to r. The basic formular of the feedback assisted iterative learning control (FBALC) scheme through which the proposed control objective can be achieved is therefore

$$u_k = u_{k-1} - H_1 y_{k-1} - H_0 y_k + [H_0 + H_1]r \tag{8}$$

Note that the choice in eq. (7) does not require any information about the process. The first two terms in the righthand side of eq. (8) compose the learning block whereas the third term represents the feedback control block. The filtered reference trajectory may be splitted over these two blocks with any proportion primarily according to the type of the feedback controller chosen. If the feedback controller is of error-feedback type, then $H_1 r$ is assigned to the learning block. If a two-degree-of- freedom controller of the type

$$u = Cr - H_0 y \tag{9}$$

is used in the feedback loop, then $[H_0 + H_1 - C]r$ is apportioned to the learning block. In Fig. 1, a schematic diagram of the FBALC is given. With this control scheme, eq. (5) is rearranged to

$$y_k = \frac{1-PH_1}{1+PH_0} y_{k-1} + \frac{P[H_0+H_1]}{1+PH_0} r + \frac{1}{1+PH_0}[w_k - w_{k-1}] \tag{10}$$

CONVERGENCE

When $w = 0$, eq. (10) can be recast to a recusion equation for the control error.

$$e_k = \frac{1-PH_1}{1+PH_0} e_{k-1} \quad \text{where } e = r - y \tag{11}$$

To derive the convergence condition for the above equation, we introduce the $\| \cdot \|_2$ norm which is defined by

$$\| e \|_2^2 = \frac{1}{2\Pi}\int_{-\infty}^{\infty} | e(j\omega) |^2 d\omega = \int_0^{\infty} | e(t) |^2 dt \tag{12}$$

Taking the norm on both sides of eq. (12) gives the following inequality.

$$\| e_k \|_2 = \left\| \frac{1-PH_1}{1+PH_0} e_{k-1} \right\|_2 \le \left\| \frac{1-PH_1}{1+PH_0} \right\|_2 \| e_{k-1} \|_2 \tag{13}$$

Here, $\| \frac{1-PH_1}{1+PH_0} \|_2$ is the induced operator norm defined by

$$\left\| \frac{1-PH_1}{1+PH_0} \right\|_2 = \sup_{\omega} \left| \frac{1-P(j\omega)H_1(j\omega)}{1+P(j\omega)H_0(j\omega)} \right| \tag{14}$$

From eq. (13), we can see eq. (11) converges to its stationary point, say $e = 0$, if e_1 is bounded and

$$\sup_{\omega} \left| \frac{1-P(j\omega)H_1(j\omega)}{1+P(j\omega)H_0(j\omega)} \right| < 1 \quad \text{or equivalently}$$

$$| 1-P(j\omega)H_1(j\omega) | < | 1+P(j\omega)H_0(j\omega) | \text{ for all } \omega \in (-\infty, \infty) \tag{15}$$

In the first run of the iterative learning, only the feedback control will be carried out. Therefore, the boundedness of e_1 is satisfied if the feedback loop is input-output stable. From this reasoning, we have the following theorem for the convergence of the basic scheme of the FBALC.

<Theorem 1> Consider the FBALC algorithm, eq.(8) applied to the batch process, eq.(1). Assume there is no random disturbance w. If $1+P(s)H_0(s)=0$ has only stable roots and the inequality (15) holds, then

$$\lim_{k\to\infty} \| r - y_k \|_2 = 0$$

When theorem 2 holds, e_k is bounded as follows.

$$\| e_k \|_2 \le \left\| \frac{1-PH_1}{1+PH_0} \right\|_2^{k-1} \| e_1 \|_2 \tag{16}$$

FBALC BASED ON AN INVERSE PROCESS MODEL

In the preceding section, we have shown that the operator norm determines the rate of convergence of the FBALC. If the norm is zero, perfect tracking is achieved in only one iteration. If it is close to 1, very slow convergence will result. This tells that the most preferred choice of H_1 is P^{-1}. Since the exact process model is practically unavailable, we propose a FBALC based on an inverse process model as follows.

$$u_k = u_{k-1} - \overline{P}^{-1} y_{k-1} - H_0 y_k + [H_0 + \overline{P}^{-1}]r \tag{17}$$

where $\overline{P}$ is a nonminal process model. The convergence condition for this scheme is described by the following inequality.

$$\left| \frac{\overline{P}(j\omega) - P(j\omega)}{\overline{P}(j\omega)} \right| < | 1 + P(j\omega)H_0(j\omega) | \tag{18}$$

Eq. (18) indicates that the FBALC converges if the relative model uncertainty is bounded by the absolute value of the return difference of the feedback loop.

Usually, the feedback controller, H_0, is designed such that it contains an integral mode for rejection of drifting disturbances. With this kind of controller, typical plots of the loop gain and the resulting allowadble error bound appear as shown Fig. 2. From this figure, we can see that a large model error is permitted in the low frequency range whereas only a limited model error, less than 100 %, is allowed in the high frequency range. Especially, at the crossover frequency, the allowed model error is reduced to $|1-1/GM|$, where GM is the gain margin. If GM=2 is chosen, the allowed model error is limited to only 50 % at the crossover frequency. Usually, the models tend to describe well the steady state and low frequency behavior of processes but become inaccurate at high frequencies. In this respect, the allowable model error bound for the FBALC seems to be somewhat stringent in the medium to high frequency range.

Graphical Interpretation of the Convergence Condition

Graphically, eq. (15) means that $P(j\omega)/\overline{P}(j\omega)$ should remain inside the circle with the center at (1,0) and radius $|1+P(j\omega)H_0(j\omega)|$ for each w. One thing we have to notice is the convergence region approaches the unit circle centered at (1,0) as ω increases to infinity. Since the circle in the limit is placed only in the first and fourth quadrants touching the origin, $P(j\omega)/\overline{P}(j\omega)$ should at least have a finite magnitude and produce a net angle contribution between -90^0 to 90^0 at the infinite frequency.

$$\lim_{\omega\to\infty} |P(j\omega)/\overline{P}(j\omega)| < \infty \quad \text{(magnitude condition)} \qquad (19)$$

$$\lim_{\omega\to\infty} |\arg P(j\omega)/\overline{P}(j\omega)| < 90^o \quad \text{(phase condition)} \qquad (20)$$

Although these conditions reflect only a part of the necessity for convergence, they play an important role especially in ruling out the cases where the convergence is violated.

Now we consider some important process models and investigate what specific problems arise when the FBALC is applied to each process.

[Case 1] First-order Process

Assume that $P=K/(\tau s+1)$ and $\overline{P}=\hat{K}/(\hat{\tau}s+1)$. Then $P/\overline{P}=(K/\hat{K})(\hat{\tau}s+1/\tau s+1)$. Here, $(K/\hat{K})(\hat{\tau}s+1/\tau s+1)$ is a linear fractional transformation and maps $s=j\omega$, $\omega\in[0,\infty)$, into a half circle as shown in Fig. 3. The half circle starts at $K/\hat{K}$ and terminates at $K\hat{\tau}/\hat{K}\tau$. As far as the $\hat{K}$ and $\hat{\tau}$ have correct signs, $P(j\omega)/\overline{P}(j\omega)$ never crosses the imaginary axis and runs into the left half plane. Thus, both the phase and the magnitude conditions are satisfied, which implies that convergence is likely.

To have a faster convergence, the plot should preferably terminate at a point close to (1,0). From this consideration, we can draw the following modeling guidelines for this case: Overestimation or underestimation of both K and τ is safer than overestimation of K with underestimation of τ or vice versa. The most unfavorable choice is overestimation of τ with underestimation of K.

[Case 2] When the order of the process transfer function is not exactly known.

First, consider the case where the pole excess of $P(s)$ is larger than that of $\overline{P}(s)$. In this case, $p(j\omega)/\overline{P}(j\omega)$ approaches the orgin with the angle of -90 multiplied by pole excess difference between $P(s)$ and $\overline{P}(s)$ as ω goes to infinity. Therefore, if the pole excess differs by larger than 1, the phase condition is violated and the FBALC inevitably diverges. On the other hand, when the difference in pole excess is 1, the phase condition is satisfied and the convergence is still likely.

Next we consider that the pole excess of $P(s)$ is smaller than that of $\hat{P}(s)$. In this case, the magnitude condition (19) is always violated.

We can recognize that exact knowledge of the process order is very important in FBALC implementation. Underestimation of the process pole excess may be allowed but the difference should not be more than 1.

[Case 3] Nonminimum Phase Processes

Nonminimum phase processes contain right half plane (RHP) zeros or time delay or both.

First, consider a process with a pure time delay of which the value is not exactly known. In this case, $\arg P(j\omega)/\hat{P}(j\omega)$ increases without bound with ω, which says that the phase condition (20) is not satisfied. In Fig. 4, we show the loci $P(j\omega)/\hat{P}(j\omega)$ for various model parmeters when $P(s)=Ke^{-ds}/(\tau s+1)$ and $\overline{P}=\hat{K}e^{-\hat{d}s}/(\hat{\tau}s+1)$. We can clearly see the convergence condition can never be satisfied as long as there is an error in time-delay estimation. The degree of violation, however, can be minimized by properly choosing model parameters.

Next, we consider the process with RHP zeros. In order for the learning block not to be unstable, $P(s)$ should not have unstable zeros. Note that a RHP zero gives a phase lag from 0 to -90 with increasing magnitude ratio as ω increases. Attempt to compensating the magnitude change in $P(j\omega)/\overline{P}(j\omega)$ by placing LHP zeros in $P(j\omega)$ will double the phase lag, thus violates the phase condition (20). Likewise, attempt to compensating the phase lag of the RHP zeros by placing LHP poles in $P(j\omega)$ only results in amplification of the diverging magnitude. This violates the magnitude condition (21). Simply neglecting the RHP zeros also causes similar problems.

As far as the basic FBALC algorithm is concerned, it seems to be very hard to satisfy the convergence condition. Exact knowledge of the time delay as well as the process order, which is impossible in practice, is a minimum requirement. In general, the process signals are majorly composed of low frequency components. Therefore, even when the convergence condition is infringed at high frequencies, we can still anticipate tendency of convergence for the first few iterations. Symptoms of divergence will emerge thereafter.

Actually, the learning algorithm need not be iterated continuously. The learning control signal obtained through a few number of iteration before the learning starts to show symptoms of divergence will still be helpful in improving the control performance.

FILTERED FBALC FOR IMPROVED ROBUSTNESS

The basic FBALC derived in the preceding sections has an ability to accomplish the asymptotically perfect control with an imperfect process model. However, the mathematical requirement of the model seems to be somewhat stringent. For example, the order of the process should be estimated as precisely as possible: the time-delay should be known exactly. Even in this case, there seems to be no appropriate way to deal with the processes with unstable zeros.

In this section, we modify the basic FBALC to relax the restrictive modeling requirements while suffering a loss of the perfectness of the original control objective to a degree. For this purpose, we consider the following filtered algorithm.

$$u_k = Fu_{k-1} - F\overline{P}^{-1}y_{k-1} - H_0y_k + [H_0 + F\overline{P}^{-1}]r \qquad (21)$$

where F is a low-pass filter. Multiplying P on both sides of eq. (21) and substituting Pu with $y-d-w$ gives

$$[1+PH_0]y_k = F[1-P\overline{P}^{-1}]y_{k-1} + P[H_0+F\overline{P}^{-1}]r + [1-F]d + [w_k - Fw_{k-1}] \qquad (22)$$

Through the similar analysis as in the basic FBALC, we can obtain the following theorem for the convergence of the filtered algorithm.

<Theorem 2> Consider the filtered FBALC, eq.(21) applied to the batch process, eq.(1). Assume that there is no random disturbance ω. In this case, if $1+P(s)H_o(s)=0$ has only stable roots and

$$\sup_{\omega} |F(j\omega)| \left| \frac{P(j\omega)\overline{P}^{-1}(j\omega)-1}{P(j\omega)H_0(j\omega)+1} \right| < 1 \qquad (23)$$

then the FBALC loop is stable and $\{ y_k(s) \}$ converges to the limit

$$y_\infty(s) = \frac{P[H_0 + F\,\overline{P}^{-1}]\; r(s) + [1-F]\; d(s)}{[1-F] + P[H_0 + F\,\overline{P}^{-1}]} \qquad (24)$$

Convergence Region

The condition (24) can be rewritten as

$$\left| 1 - \frac{P(j\omega)}{\overline{P}(j\omega)} \right| < \frac{|\, 1 + P(j\omega)H_0(j\omega) \,|}{|\, F(j\omega) \,|} \quad \text{for all } \omega \qquad (25)$$

When $F(s)$ is a low pass filter with the unit d.c. gain, the convergence region especillay at high frequencies will be greatly enlarged while rendering the low frequency converence region, which already has a sufficient margin, almost intact. Therefore, with the aid of this filter, the robustness of the FBALC is greatly improved, and estimation errors in the process order as well as time delay can now be allowed to some extent. In addition, the process with RHP zeros may also become an object of the FBALC.

From the view point of robustness, it will be more advantageous to have a filter which has a small magnitude. However, such a filter leads the ultimate output profile more distorted. A trade-off is, therefore, needed between robustness and performance when we choose a filter.

Ultimate Tracking Performance

As is given in theorem 2, the asymptotic perfectness of the basic FBALC is inevitably distorted when the filter is introduced. Of course, degree of the distortion depends on the detailed shape of the filter $F(s)$. When $F(s) = 1$, the algorithm coincides with the original basic FBALC. On the other hand, when F(s) = 0, the learning block ceases to work and the FBALC becomes the feedback-only controller. In this case, the limit profile appears to be

$$y_\infty(s) = \frac{PH_0}{1+PH_0} r(s) + \frac{1}{1+PH_0} d(s) \qquad (26)$$

which itself is the closed-loop response of a feedback loop. Considering these two extremes, we can expect that the performance of the filtered FBALC lies between the perfect tracking and the one by the feedback-only control.

Design Guide of the Learning Block Filter

As was discussed above, $F(s)$ should be a low pass filter with the unit d.c. gain. The filter need not be of any specific form. Its design, however, requires a trade-off between robustness and performance as is in the feedback controller design. We arbitrarily suggest the following form for the filter

$$F(s) = \frac{1}{(\lambda s + 1)^n} \qquad (27)$$

which has two adjustable parameters. Although no rigid rule is there in determining the order n, n should preferrably be equal to or greater than the pole excess of $\overline{P}(s)$ in order for the compensator $F(s)\overline{P}(s)$ not to be a differentiator. The parameter λ determines the band-width of the filter. At the corner frequency, $\omega = 1/\lambda$, the filter gain reduces to $1/\sqrt{2}^n$. Since the convergence region reaches minimum at around the crossover frequency of the loop gain, $P(s)H_0(s)$, λ should preferably be determined such that $1/\lambda$ is smaller than the crossover frequency so as to sufficiently enlarge the restricted convergence region around the crossover frequency.

NUMERICAL ILLUSTRATIONS

In this section, two numerical examples are given to demonstrate the behavior of the FBALC. In both examples, we assume the following second-order process subject to a sinusoidal-type external disturbance.

$$y(s) = [1/(s+1)(3s+1)]\; u(s) \;+\; 0.75/(s^2 + 0.75^2) \qquad (28)$$

The following error-feedback PI controller is assumed in the feedbak loop.

$$H_0(s) = 5(1 + 1/2s) \qquad (29)$$

The nominal models are taken differently in each example.

[Example 1] In this example, the following model is used in the learning block design.

$$\overline{P}(s) = 1.2/(s+1)(1.5s+1) \qquad (30)$$

It is assumed that the order of the process is known but the steady state gain and also the dominating time constant are falsely estimated.

In Figs. 5 and 6, performance of the basic FBALC is summerized. The loci, $1 - P(j\omega)/\overline{P}(j\omega)$ and $1 + P(j\omega)H_0(j\omega)$, are shown together in Fig. 5 to examine the convergence property. We see that the convergence condition is nicely met although the dominant time constant is estimated 100% smaller than the true value. Result of the learning control is shown in Fig. 6. Starting from the PI-only control in the first run, the output almost converges to the reference trajectory in three iterations while rejecting the external disturbance. The control input, however, shows somewhat large hunting. It is majorly caused by the two-times numerical differentiation in the learning block.

Fig. 7 is the learning results when the modifying filter with $n = 2$ and $\lambda = = 0.35$ is introduced to moderate the excessive fluctuation in the control input. The converged output profile is a little bit distorted, but the hunting of the control input could be significantly moderated.

[Example 2] In this example, the process transfer function is modelled as

$$\overline{P}(s) = 1.2e^{-0.7s}/2.5s + 1 \qquad (31)$$

It is assumed that the short time constant is approximated by a time delay and the dominant time constant as well as the steady state gain are estimated with some error.

Fig. 8 shows the convergence behavior of the basic FBALC. As might be expected, the convergence condition can not be met in this case, which is again verified through numerical simulation shown in Fig. 9. As is partly shown in Fig. 9, however, tracking performance of the FBALC appears to be better than that of the PI-only control up to fifth iterations. After then, the output starts to oscillate showing a tendency of divergence.

To remedy the divergence, we introduced a modifying fiter with $n = 2$ and $\lambda = 0.82$. Figs. 10 and 11 show the behaviors of the filtered FBALC. We can see that the convergence condition is now satisfied but the output converges to a somewhat distorted profile. The converged profile, however, is still better than the one the PI-only control can give. The control input is significantly moderated in this example, too, compared to that obtained in the the basic FBALC.

Through the above two examples, we see that the modifying filter resolves the potential divergence problem while relieving the excesive fluctuation in the control input. Distortion of the output profile, however, becomes serious in some cases. In example 2, the output response up to the third run looks better than the converged profile by the filtered FBALC. In this case, it will be better to take the learning control signal at the third run by the basic FBALC and to use it for the subsequent runs.

CONCLUSIONS

An inverse model-based feedback-assisted learning control has been proposed together with a filtered version, purposing for batch process control. Some of the important consequences obtained through analyzing the proposed algorithm and through numerical simulations are as follows:

1. Convergence rate of the learning is maximized when the inverse process transfer function is used in the learning block.
2. Bound of the allowable modeling error varies with frequency. Generally, the error bound becomes most stringent at around the crossover frequency of the feedback loop.
3. In order for the basic algorithm to be convergent, estimation error in process order may be allowed only when the estimated value is smaller than the true value by one; the time delay should be known exactly.
4. The restricted convergence region in the basic learning scheme can be enlarged by low-pass filtering of the learning block but resulting in some distortion of the ultimate output profile. By properly choosing the filter while trading-off between convergence and tracking performance, we can significantly improve the overall performance of the learning.
5. Though the proposed learning control scheme has been developed for linear systems, it also has a potential to be adapted for control of a large class of nonlinear systems. It is owing to the intrinsic property of the learning algorithm, which can self-compensate disturbances with the same pattern.

REFERENCE

1. Arimoto, S., S. Kwwamura, and F. Miyazaki, "Bettering Operations of robotics by learning", J. Robotic System, 1, 123-140 (1984).
2. Bien, Z, and K. M. Huh, "Higher-Order iterative learning control algorithm", IEE Proc. 136, 105-112 (1989).
3. Bondi, P., G. Casalino and L. Gambardella, "On the iterative learning control theory for robotic manipulators, IEE J. of Robotics and Autom., 4, 14-22 (1988).
4. Elicabe, G. E. and G. R. Meira, "Estimation and control in polymerization reactors; a review", Polymer Eng. Sci., 28, 121-135 (1988).
5. Gu, Y. and N. Loh, "Learning control in robotic systems" , Proc. IEEE Int. Sym. on Intelligent Control, Philadelphia, 360-364 (1987).

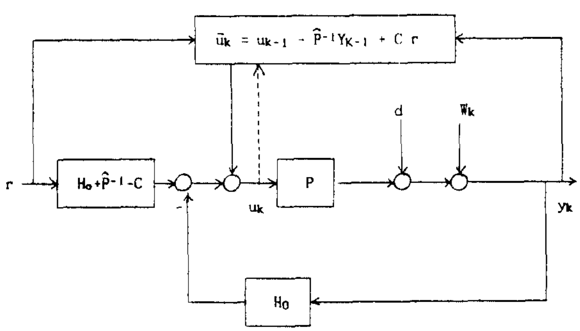

Fig 1. Schematic diagram of the FBALC

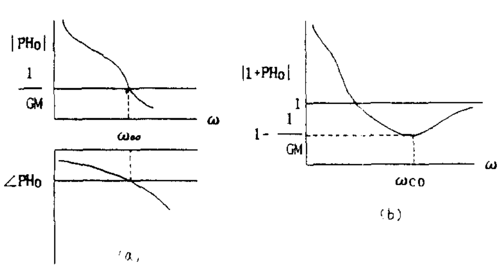

Fig 2. Bode plot of typical loop gain (a) and allowable model error bound for convergence (b)

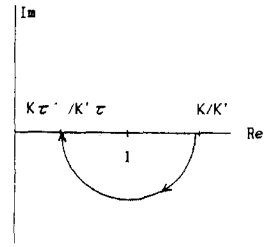

Fig 3. Nyquist plot of first-order process

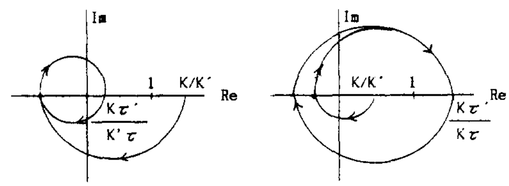

Fig 4. Nyquist plot of nonminimun phase process

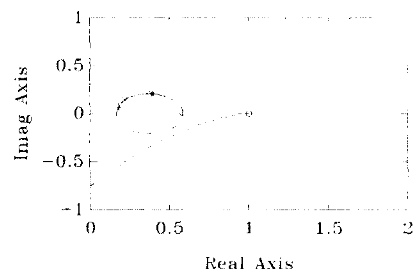

Fig 5. Nyquist plot of Example 1. without filter

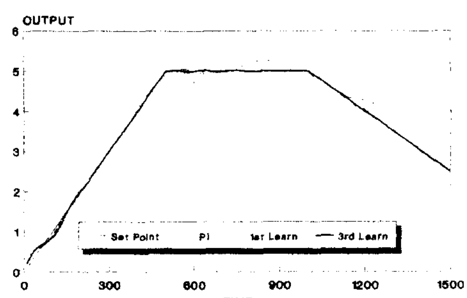

Fig 6. Result of Example 1. without filter

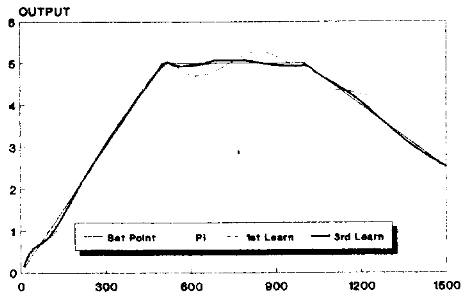

Fig 7. Result of Example 1. with filter

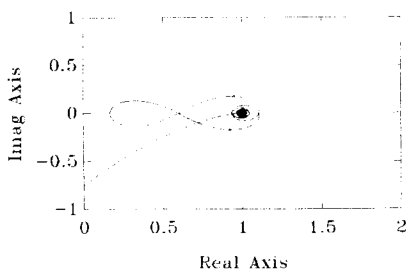

Fig 8. Nyquist plot of Example 2. without filter

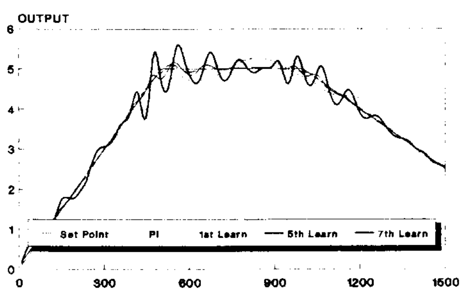

Fig 9. Result of Example 2. without filter

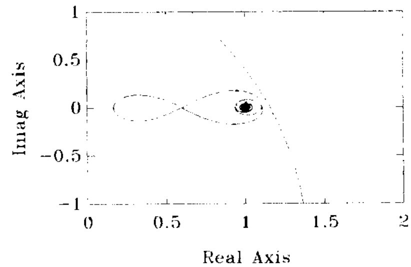

Fig 10. Nyquist plot of Example 2. with filter

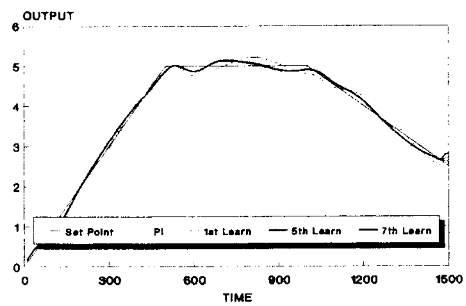

Fig 11. Result of Example 2. with filter