

# 정상상태 特性을 개선한 디지털 制御機 設計

## (A Digital Controller Design To Improve Steady-State Characteristics)

金 榮 吉\*, 朴 政 用\*, 李 相 培\*

(Young Kil Kim, Mignon Park and Sang Bae Lee)

### 要 約

基準入力は 정상상태 오차에 영향을 주는 要因 中의 하나다. 本 論文은 基準入력에 의해 야기되는 정상상태 오차를 제거할 수 있는 새로운 디지털 制御機 設計 방법을 제시한다. 먼저 基準入力 함수 형태에 따라, 基準入力 項이 제거된 새로운 制御 시스템 式이 유도되고, 最適制御를 利用하여 유도된 시스템의 出力을 最小로 하는 制御法則이 구해진다. 상태공간 방법에 의해, 제시된 制御 알고리즘은 不安定 시스템을 포함해서 시불변 선형 시스템에 적용될 수 있다.

### Abstract

The reference input is one of causes having an effect upon the steady-state error. This paper describes a design method of a digital controller to remove the steady-state error caused by the reference input. According to the types of the reference input, new system equations to remove the reference input term from controlled system equations are derived first. And, using the optimal control theory the control law is obtained to minimize the output of the new system. Based on the state-space approach, the proposed control algorithm can be applied to time-invariant linear systems including the unstable systems.

### I. 序 論

정상상태 오차는 임의의 형태의 入力이 制御 시스템에 인가되었을 때 制御의 정확도를 나타내는 尺度가 된다. 실제 制御 시스템에서, 여러가지 理由로 시스템의 정상상태 出力이 基準入력과 정확히 一致하기란 매우 힘들다. 따라서 정상상태 오차는 制御 시스템에서 거의 피할 수 없는 것이며, 制御시스템 設計 目的 中의 하나가 이 오차를 最小로 유지하거나 어떤 허용치

이내로 만드는 것이다.

一般的으로 정상상태 오차를 야기시키는 요인으로는 비선형 미찰력, 능동소자의 비선형 입출력 특성, 양자화 오차등 비선형 요소와 基準入力, 雜音 등에 의한 선형요소들이 있다. 여기서 비선형 요소들에 대한 制御 시스템 오차의 一般의인 解를 얻기란 힘들다.

本 論文은 선형요소, 특히 基準入력에 따른 制御 시스템의 정상상태 오차를 개선하려는 것이다.

종래 디지털 制御 시스템에서 정상상태 특성을 개선하기 위해 여러가지 방법이 사용되었다. PID 制御機에서는 습의 作用을 制御 루프속에 도입하였고, 最適 制御 방법에서는 수행지수(performance Index)에 前後時間에 대한 制御入力の 差를 포함시켰으며, 自己

\*正會員, 延世大學校 電子工學科

(Dept. of Electron. Eng., Yonsei Univ.)

接受日字: 1985年 2月 7日



이때 행렬  $\Phi$  및  $F$ 는 식(6)으로 부터 주어진다. 또한 식(5)로 부터 오차  $e(k)$ 는 다음과 같이 쓸 수 있다.

$$e(k) = x_1(k) + x_2(k) \\ = [1 \quad 1 \quad 0 \quad \dots \quad 0] X(k) \quad (8)$$

식(8)은 식(1b) 및 (2b)와 유사하게 생각하여 식(7)로 주어지는 시스템의 出力으로 볼 수 있다. 이때 제어하고자 하는 시스템의 특성, 즉 식(1)의 행렬  $A$  및 벡터  $B, C$  값에 따라 식(8)이 달라질 수 있다. 이에 대한 자세한 것은 다음 節에서 설명한다.

이제 매 샘플링 순간마다 식(8)의  $e(k)$ 를 最小로 하는 적당한 制御人力  $u(k)$ 를 구할 수 있다면 식(1)로 주어지는 시스템의 基準人力에 따른 정상상태 오차는 없어질 것이다. 더구나 정상상태 오차는 시스템 동작 중에 변수 ( $A, B, C$ )들이 일정한 한계 이내로 변한다 해도 상관없이 제거될 것이다(완전히 다른 시스템일 경우는 식(5)의 상위 2개의 벡터 ( $x_1, x_2$ )를 다르게 정의함으로써 식(8)을 변경해야 한다).

다음, 基準人力이 램프함수일 때는 식(2)와 (3)의 양변에 2차 차(second difference)를 취하면,

$$\Delta^2 L_0(k+1) = A_0 \Delta^2 L_0(k) + B_0 \Delta^2 u(k) \quad (9a)$$

$$\Delta y(k) = C_0^T \Delta^2 L_0(k) \quad (9b)$$

$$\Delta^2 e(k) = -\Delta^2 y(k) \quad (9c)$$

식(10)과 같이 새로운 상태벡터를 정의하여 상태공간 방정식으로 나타내면 식(11)과 같다.

$$\begin{aligned} x_1(k) &\equiv e(k-1) - e(k-2) \\ x_2(k) &\equiv e(k-1) \\ x_3(k) &\equiv \Delta^2 e(k) = e(k) - 2e(k-1) + e(k-2) \\ &= -\Delta^2 l_1(k) \\ x_4(k) &\equiv \Delta^2 l_2(k) \\ &\vdots \\ x_{n-2}(k) &= \Delta^2 l_n(k) \\ x_{n-3}(k) &\equiv u(k-2) \\ x_{n-4}(k) &\equiv u(k-1) \end{aligned} \quad (10)$$

다시 쓰면

$$X(k+1) = \Phi X(k) + F u(k) \quad (11)$$

여기서,

$$\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & a_1 & -1 & 0 & 0 & \dots & -b_1 & 2b_1 \\ 0 & 0 & -a_2 & 0 & 0 & 0 & \dots & b_2 & -2b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -a_n & 0 & 0 & 0 & \dots & b_n & -2b_n \\ 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 \end{bmatrix}$$

$$F = [0 \quad 0 \quad -b_1 \quad b_2 \quad \dots \quad b_n \quad 0 \quad 1]^T \quad (12b)$$

또한 식(10)으로 부터 오차  $e(k)$ 는

$$e(k) = x_1(k) + x_2(k) + x_3(k) \\ = [1 \quad 1 \quad 1 \quad 0 \quad \dots \quad 0] X(k) \quad (13)$$

식(13)은 식(11)로 주어지는 시스템의 출력으로 볼 수 있으며, 제어 시스템의 특성에 따라 다르게 될 수 있다.

같은로 基準人力이 2차함수 이상 일 경우는 上記 두 경우와 유사하게 생각할 수 있으므로 여기서는 생략한다.

이제 식(8)과 (13)으로 표현되는 시스템 오차  $e(k)$ 를 매 샘플링 순간마다 最小값으로 만드는 制御人力  $u(k)$ 는 最適制御 방법을 사용하여 쉽게 구할 수 있다. 일반적으로 最適制御는 다음과 같은 수행지수(performance Index)가 最小가 되는 制御人力  $u(k)$ ,  $k=0, 1, \dots, N$ 를 얻는 것이다.

$$J = \frac{1}{2} \sum_{k=0}^N | X^T(k) Q X(k) + g u^2(k) | \\ = \frac{1}{2} \sum_{k=0}^N | e^2(k) + g u^2(k) | \quad (14)$$

이 때 주목할 것은 하중행렬(weighting matrix)  $Q$ 가 식(8)과 (13)으로 부터 식(15)와 (16)으로 각각 주어진다.

$$Q = \begin{bmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (15)$$

$$Q = \begin{bmatrix} 1 & 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (16)$$

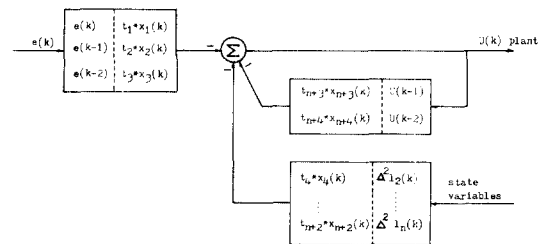


그림 1. 디지털 제어기 구조  
Fig. 1. The Structure of Digital Controller.

式(7)은 혹은 式(8)의 제한조건에 대한 式(4)의 수렴 지수를 最小로 하는 일반적인 最適制御 해는

$$u(k) = -T(k)X(k) \tag{17}$$

여기서 轉환벡터  $T(k)$ 는

$$T(k) = (g + \Gamma^T S(k+1) \Gamma)^{-1} \Gamma^T S(k+1) \phi \tag{18a}$$

$$S(k) = \phi^T \{ S(k+1) - S(k+1) \Gamma (g + \Gamma^T S(k+1) \Gamma)^{-1} \Gamma^T S(k+1) \} \phi + Q \tag{18b}$$

$$S(N) = Q$$

결과적으로 기준입력에 따른 정상상태 오차를 개선시킬 수 있는 디지털 제어기의 구조는 그림 1과 같다.

### III. 하중인자(Q, g)의 선택

前節의 最適制御 문제에서 시스템을 安定하게 制御할 수 있는 원하는 制御 入力  $u(k)$ 를 얻기 위해서는 하중인자(Q, g)를 적당히 선택하여야 한다.

一般的으로 상태벡터 하중행렬Q는 대칭이고 음이 아닌 행렬이어야 하며, 制御 入力 하중값 g는 양의 수이어야 한다. 이들은 제어하고자 하는 시스템의 특성과 원하는 과도응답에 따라 다르게 선택되어진다. 원하는 정확한 制御 특성을 얻기 위한 하중인자 선택은 일반적으로 약간의 시행오차를 거쳐 결정된다.

本論文에서의 상태벡터 하중행렬Q는 一般的인 最適制御에서의 하중행렬 선택 방법과는 전혀 다르게 선택된다. 일반적인 最適制御에서 하중행렬들은 위에서 설명한 一般的인 선택방법에 근거하여 상태벡터 자체와는 무관하게 선택되지만 여기서의 하중행렬Q는 式(8)과 (13)으로 부터 알 수 있듯이 주어진 시스템으로부터 정상상태 오차를 제거하기 위해 새로이 정의된 상태벡터  $X(k)$ 에 따라 결정된다. 따라서 제어하고자 하는 시스템의 특성에 따라 하중행렬Q를 변경하려면 상태벡터  $X(k)$ 를 다르게 정의해야 한다. 여기서 시스템 특성에 따른 하중행렬Q를 선택하는 一般的인 방법은 아직 발견하지 못하였지만, 간단한 2차 시스템인 경우 비교적 쉽게 여러가지 Q값을 얻을 수가 있다. 여러가지 Q값에 대한 컴퓨터 시뮬레이션 결과 두 極이 모두 음인 경우는 式(10)으로 정의되는 상태벡터  $X(k)$ 에 의한 Q값(式(16)), 두 極이 모두 양인 시스템인 경우는 式(19)로 정의되는 상태벡터  $X(k)$ 에 의한 Q값(式(20)), 그리고 두 極이 양과 음에 각각 하나씩 있는 시스템일 때는 式(21)로 정의되는 상태벡터  $X(k)$ 에 의한 Q값(式(22))이 각각 적당함을 보였다.

$$\begin{aligned} x_1(k) &\equiv e(k-2) \\ x_2(k) &\equiv e(k-1) \\ x_3(k) &\text{ 이하는 式(10)과 同--} \end{aligned} \tag{19}$$

이때,

$$Q = \begin{bmatrix} 1 & -2 & -1 & 0 & \dots & 0 \\ -2 & 4 & 2 & 0 & \dots & 0 \\ -1 & 2 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{20}$$

$$x_1(k) \equiv e(k-2)$$

$$x_2(k) \equiv e(k-1) - e(k-2)$$

$$x_3(k) \text{ 이하는 式(10)과 同--} \tag{21}$$

이때,

$$Q = \begin{bmatrix} 1 & 2 & 1 & 0 & \dots & 0 \\ 2 & 4 & 2 & 0 & \dots & 0 \\ 1 & 2 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{22}$$

또한, 制御 入力에 대한 하중값 g는 컴퓨터 시뮬레이션 결과 0.1 정도나 10<sup>5</sup> 정도가 적당함을 보였다.

### IV. 不安定한 시스템에 대한 應用

前 述된 정상상태 오차를 제거하기 위한 制御機 設計 방법은 安定한 시스템에 대한 것이다. 그러나 이 방법은 不安定한 시스템에도 확장시킬 수 있다.

이제 式(1)의 制御 시스템이 不安定하다고 하자. 우선 이 시스템의 極들을 원하는 안정한 위치로 배치하기 위한 制御 入力  $U_p(k)$ 은

$$\begin{aligned} U_p(k) &= -K(k)L(k) \\ &= -[K_1(k), K_2(k), \dots] \{l_1(k), l_2(k), \dots\}^T \end{aligned} \tag{23}$$

여기서 상태벡터 轉환량  $K(k)$ 는 Z영역 페루프 시스템 특성방정식(24)의 根들이 원하는 위치에 있도록 정한다.

$$\det |ZI - A + BK| = 0 \tag{24}$$

그러나 주어진 시스템의 極을 변경시키기 위한 制御 入力  $U_p(K)$ 는 만으로는 基準 入力에 따른 정상상태 오차는 없앨 수 없다. 制御 시스템의 極을 원하는 위치로 배치하면서 基準 入力에 따른 정상상태 오차를 제거하기 위한 制御 入力は 각각의 制御 入力を 疊하여 줌으로써 가능하다. 즉,

$$U_T(k) = U_p(k) + U(k) \tag{25}$$

여기서  $U_p(k)$ 는 極배치를 위한 것이며,  $U(k)$ 는 정상상태 오차를 제거하기 위한 것이다. 이 때  $U(k)$ 는 페루프 시스템에 대한 값으로 off-line으로 계산되어야 한다.

V. 컴퓨터 시뮬레이션

제시된 디지털 制御機의 유효성을 증명하기 위해, 3 가지 2 차 시스템을 선정하여 컴퓨터 시뮬레이션을 실시한다. 편의상 基準入力이 램프인 경우에 유도된 制御機에 램프 및 계단입력을 함께 인가한다.

制御利得  $T(k)$ 는 式(18)로 부터 상태벡터  $X$ 와 무관하게 계산됨을 알 수 있다. 시간길이  $N=50$ 으로 선택하여 컴퓨터에 의해 계산된 制御利得  $T = [t_1, t_2, \dots, t_50]$ 을 그림 2-1와 그림2-2에 나타낸다. 그림에서  $N=20$  이상이면 모두 일정한 값으로 수렴함을 알 수 있다. 따라서 예제들에 대해 실시한 컴퓨터 시뮬레이션은  $T(k), K=0, \dots, N$  값들을 일일이 저장하여 사용하지 않고,  $N$ 을 충분히 크게하여 얻은 수렴값을 대로  $T$ 값으로 사용한다. 실제 마이크로프로세서를 이용하여 실시간으로 最適制御할 경우 이러한 방법이 유용하다.

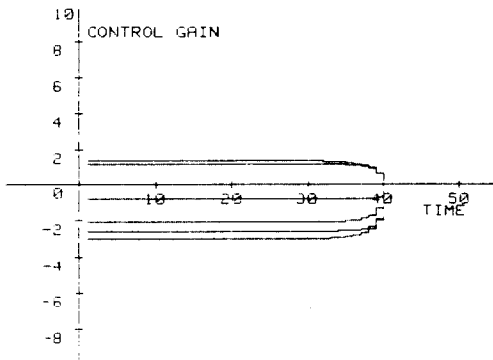


그림2-1. 시간에 대한 제어이득 ( $Q = 式(20) g=0.3$ )  
Fig. 2-1. Control gains vs. time ( $Q = Eq. 20) g=0.3$ ).

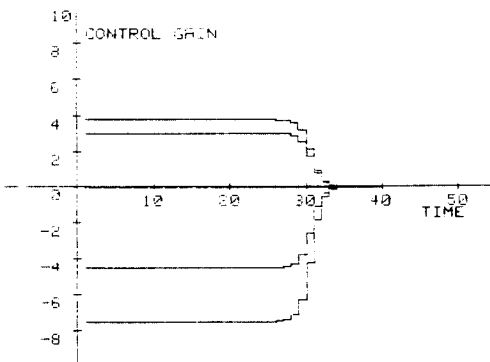


그림2-2. 시간에 대한 제어이득 ( $Q = 式(22) g=10^5$ )  
Fig. 2-2. Control gains vs. time ( $Q = Eq(22) g=10^5$ ).

例 1)

$$A = \begin{bmatrix} -1.5 & 1.0 \\ -0.54 & 0.0 \end{bmatrix} \quad B = \begin{bmatrix} 1.0 \\ -0.4 \end{bmatrix} \quad C = [1.0 \quad 0.0]$$

이것은 두 극이 모두 음인 安定한 시스템이다. 하중 행렬  $Q$ 는 式(16), 하중값  $g=0.08$ 을 사용한 시뮬레이션 결과를 그림 3에 나타낸다.

例 2)

$$A = \begin{bmatrix} -0.9 & 1.0 \\ 0.1 & 0.0 \end{bmatrix} \quad B = \begin{bmatrix} 1.0 \\ -0.4 \end{bmatrix} \\ C = [1.0 \quad 0.0]$$

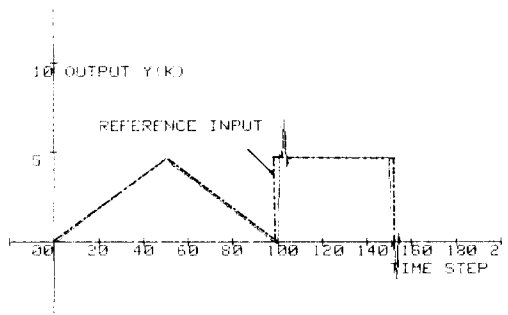


그림 3. 例 1의 램프와 계단 기준 입력에 대한 출력  
Fig. 3. The output of example 1 to ramp and step reference input.

이것은 두 극이 양과 음인 安定한 시스템이다. 하중 행렬  $Q$ 는 式(20), 하중값  $g=0.3$ 을 사용한다. 결과는 그림 4에 나타낸다.

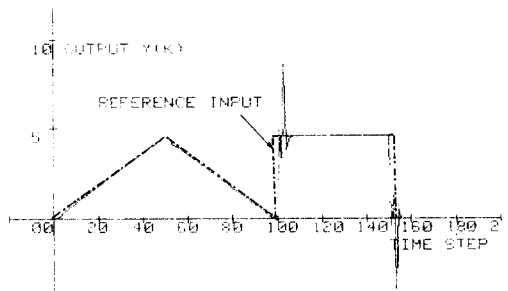


그림 4. 例 2의 램프와 계단기준입력에 대한 출력  
Fig. 4. The output of example 2 to ramp and step reference input.

例 3)

$$A = \begin{bmatrix} 1.0 & 1.0 \\ -2.0 & 0.0 \end{bmatrix} \quad B = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} \quad C = [1.0 \quad 0.0]$$

이것은 不安定한 시스템으로 페루프의 극을 0.5와 0.6이 되도록 한다. 하중행렬  $Q$ 는 式(22), 하중값  $g=10^4$ 을 사용한다. 결과는 그림 5에 나타낸다.

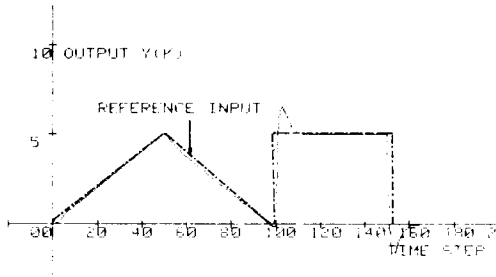


그림 5. 例 3의 램프와 계단기준입력에 대한 출력  
Fig. 5. The output of example 3 to ramp and step reference input.

## VI. 結 論

정상상태 特性을 개선한 디지털 제어 알고리즘을 유도한다. 주어진 制御 시스템으로 부터 基準人力 項이 제거된 새로운 시스템을 정의하고, 이 시스템의 출력을 最小로 하는 制御法則을 最適制御 방법을 利用하여 구한다. 不安定 시스템을 포함하여 3 가지 선택된 시스템에 대해 실시한 컴퓨터 시뮬레이션은 좋은 정상상태 특성을 보인다.

이 알고리즘은 상태공간 방법으로 선형 시불변 시스템에 적용될 수 있으며, 制御되는 시스템 변수들이 動作 中에 변한다. 해도 무관하기 때문에 실제 실현될 경우 매우 좋은 결과가 예상된다.

앞으로 多入出力(MIMO) 시스템에 대한 연구가 바람직하다.

## 參 考 文 獻

- [1] M.S. Calovic and N.M. Cuk, "proportional integral derivative realization of optimal linear quadratic regulator." *proc. IEE*, no. 11, pp. 1441-1443, 1974.
- [2] M. Tomizuka and D.E. Rosenthal. "On the optimal digital state vector feedback controller with integral and preview actions." *Trans ASME*, vol. 101, pp. 172-178, June 1979.
- [3] Yih T, Tsay, and Leang S. Shieh. "state-space approach for self-tuning feedback control pole assignment." *proc. IEE* no. 3, pp. 93-101, 1981.
- [4] Gene. F. Frankline and J. David Powell. *Digital Control of dynamic systems*. Addison-Wesley publishing company Inc. 1980.
- [5] Benjamin C. Kuo. *Automatic control systems. Fourth edition*, PRENTICE-HALL. 1982.