

대규모 시스템을 위한 준최적 저감도성 비집중형 제어기에 관한 연구

(A Study on Decentralized Suboptimal Controller for Large Scale Systems with Low Sensitivity Property)

吳 尚 錄*, 徐 一 弘**, 卞 增 男*

(Sang-Rok Oh, Il Hong Suh and Zeungnam Bien)

要 約

본 논문에서는 대규모 시스템을 위한 준최적 저감도성 비집중형 제어기를 설계하기 위하여 준최적 이득 행렬을 구하기 위한 필요조건과 알고리즘을 유도하였다. 또한 감도를 고려하지 않은 종래의 비집중 제어기와의 비교를 위하여 수치적 예를 들어 보였다.

Abstract

A local suboptimal controller with low sensitivity property is studied for a case of large scale systems. More specifically, necessary conditions for the suboptimal gain matrices are obtained by a parameter optimization technique, from which a design method in the form of an algorithm is suggested. To show the validity of the result and, in particular, to appreciate the effect of adding the sensitivity function, a numerical example is included.

I. 序 論

제어 시스템을 분석 또는 설계할 때에, 보통 중앙으로 시스템 정보가 집중되며, 제어기는 집중된 많은 양의 정보를 처리할 수 있다고 가정한다. 그러나 만일 부시스템들이 지역적으로 널리 분포되어 있는 대규모 시스템인 경우, 모든 상태 정보를 한 곳에 모으는 것은 경제적으로나 신뢰성에 있어서 비실제적이게 된다. 이러한 어려움들을 해결하기 위해 대규모 시스템에 있어서 비집중 제어분야에서 많은 연구가 진행되어 왔

으며, 특히 Singh 및 Hassan^[1, 2]은 비집중 준최적 제어기의 설계 방법을 제안하였다. 그러나 기존의 이러한 제어방법들은 시스템 파라미터들을 정확히 알고 있고 또 시간이 지남에 따라 변하지 않는다는 가정을 전제로 한 것으로, 실제 대규모 시스템에 있어서 모델링의 부정확성이나, 시스템 파라미터들이 외란에 의해 원래의 명목치(nominal value)로부터 약간씩 벗어나게 되는 경우, 파라미터 외란이 없다는 가정하에 설계된 Singh 및 Hassan^[1, 2]의 준최적 제어기로서 원하는 시스템의 동작특성을 얻을 수가 없다. 이러한 문제를 해결하기 위해 Kreindler와 O' Reilly^[3, 4]는 제어기를 설계하는데 있어서 감도 최소화 방법들을 제안하였는 바, 이들은 중앙 집중 제어기 형식으로 설계되어 있어 제어기는 전체 시스템의 상태 정보를 모두 사용하므로 대규모 시스템에 직접 응용하기가 어렵다. 본 논문에서는, 대규모 시스템을 위한 저감도 비집중

*正會員, 韓國科學技術院. 電氣 및 電子工學科
(Dept. of Electrical Science, KAIST)

**正會員, 大宇重工業技術研究所
(Daewoo Heavy Industries, Ltd.)

중 준최적 제어를 설계하는데 있어서 각 제어기는 상태변수 뿐 아니라 상태 변수 및 제어변수의 궤적감도변수 (trajectory sensitivity variable) 까지 포함하여 비집중 형태로 설계하며 이렇게 하여 설정된 문제에서 준최적 이득행렬 을 얻기 위한 필요조건 및 알고리즘 (algorithm) 을 제안하기로 한다.

II. 문제 설정

시스템의 다이내믹스 (dynamics) 와 초기 조건이 다음과 같이 주어지는 대규모 시스템을 생각하기로 하자.

$$x_i = A_i x_i + B_i u_i + \sum_{j=1}^N A_{ij} x_j, \quad x_i(0) = x_{i0}, \quad (1)$$

$$y_i = C_i x_i, \quad \text{for } i=1, 2, \dots, N$$

여기서 $x_i \in R^{n_i}$, $u_i \in R^{m_i}$ 그리고 $y_i \in R^{m_i}$ 이다. 이를 벡터상태 방정식 형태로 나타내면 다음과 같이 쓸 수 있다.

$$\dot{x} = Ax + Bu, \quad x(0) = x_0, \quad y = Cx. \quad (2)$$

이제 (1)식 또는 (2)식으로 기술되는 대규모 시스템에 대한 비집중 제어기 (decentralized controller) 를 다음과 같이 설정하자.

$$u_i = K_i(x_i - x_{si}) + G_i z_i, \quad i=1, 2, \dots, N \quad (3)$$

$$\dot{z}_i = y_i - y_{si}, \quad z_i(0) = 0, \quad i=1, 2, \dots, N \quad (4)$$

여기서 z_i 는 부가적인 m_i -차원 상태 벡터이고 x_{si} , y_{si} 등은 정상 상태 (steady state)에서의 명목치 (nominal value)로 정의한다. 또한 (3)식에서 K_i 는 비례 제어기의 이득행렬이며, G_i 는 계단기준입력 u_{si} 가 가해졌을 때 출력 y_i 의 정상 상태 오차가 없도록 하기 위한 적분 제어기의 이득행렬이다. 이 제어기의 블록도를 그림 1에 보였다. 편의상 $\Omega(s)$ 와 $\Omega_1(s)$ 를

$$\Omega(s) = \{K \in R^{m \times n} \mid K = \text{block diag. } (K_1, \dots, K_N) : K_i \in R^{m_i \times n_i}\}, \quad (5)$$

$$\Omega_1(s) = \{G \in R^{m \times m} \mid G = \text{block diag. } (G_1, \dots, G_N) : G_i \in R^{m_i \times m_i}\} \quad (6)$$

라 정의하기로 한다.

그러면 일반적으로 고려되는 준최적화 제어문제는 다음과 같다.^[5]

식(1) 또는 식(2)로 기술되는 대규모 시스템에 대하여 식(3) 및 식(4)의 형태로 주어지는 제어기로서

$$J_1(x, u) = \int_0^\infty \{ (x - x_s)^T Q (x - x_s) + (u - u_s)^T R (u - u_s) \} dt \quad (7)$$

을 최소화하는 이득행렬 $K \in \Omega(s)$ 와 $G \in \Omega_1(s)$ 을 구하라. 단 $Q \geq 0$ 이며 $R > 0$ 임.

여기서 준최적화 제어 문제라 함은, 전형적인 최적 제어문제와는 달리 제어기의 형태를 임의로 선택하지 않고 (3)식의 형태로 제한하면서 (7)식을 최소화하는 K, G 를 선택하여야 하기 때문이다. 이러한 문제를 풀어 구한 K, G 는 준최적 이득행렬이 된다. 그러나 전장에서 언급했듯이 만일 (1)식으로 표현되는 시스템 파라미터가 외란 (disturbance)에 의해서나 혹은 모델링의 부정확성으로 말미암아 원래의 명목치로부터 약간 변할 경우 위와 같이 설정한 기존의 준최적화 제어문제를 풀어 구한 이득행렬 K 와 G 는 준최적 (suboptimal)이 아닐 수 있으며, 심한 경우에는 시스템을 불안정화 시키기도 한다. 이에 대해서는 마지막 절의 수치예제에서 실제 예를 들어 보이기로 한다. 따라서 이러한 어려움을 해결하기 위해 파라미터의 변화에 대해 제어 특성이 별로 바뀌지 않는 저감도 제어기의 설계 방법을 다음과 같이 생각하였다. 우선 (2)식의 A, B 행렬에 있어서, 이들 각각의 원소가, 명목치에 대한 교란 (perturbation)의 정도 (amount)를 나타내는 양으로서 스칼라 파라미터 α 의 함수라고 하자. 그리고 이때 궤적감도변수 (trajectory sensitivity variable) $v_i(t)$ 와 제어감도변수 (control sensitivity variable) $w_i(t)$ 를 다음과 같이 정의하자.

$$v_i(t) \triangleq \frac{\partial x_i(t)}{\partial \alpha}, \quad v_i(0) = 0, \quad i=1, \dots, N \quad (8)$$

$$w_i(t) \triangleq \frac{\partial u_i(t)}{\partial \alpha}, \quad w_i(0) = 0, \quad i=1, \dots, N$$

또는 벡터형태로 쓰면 다음과 같이 쓸 수 있다.

$$v \triangleq [v_1^T \ v_2^T \ \dots \ v_N^T]^T, \quad (9)$$

$$w \triangleq [w_1^T \ w_2^T \ \dots \ w_N^T]^T.$$

그러면 (2), (3), (8), (9)식으로 부터 다음 식을 얻을 수가 있다.

$$\dot{v} = Av + Bw + Tx + Mu, \quad v(0) = 0 \quad (10)$$

$$w = Kv + G \frac{\partial z}{\partial \alpha}$$

여기서 $T = \frac{\partial A}{\partial \alpha} \Big|_{\alpha=\alpha_0}$, $M = \frac{\partial B}{\partial \alpha} \Big|_{\alpha=\alpha_0}$ 이다.

또한 α_0 는 α 의 명목치 (nominal value)이다. T 와 M 은 각각의 원소가 α 의 함수인 A, B 행렬을 알고 있으므로 계산에 의해 그 값을 알 수 있다. 저감도 제어기를

설계하기 위하여 다음과 같은 quadratic cost functional을 생각할 수 있다.

$$J_2(v, w) = \int_0^{\infty} (v^T S v + w^T R_1 w) dt \quad (11)$$

여기서 $S \geq 0$, $R_1 > 0$ 이다.

그러면 앞에서 제시한 기존의 준최적화 제어 문제에 이들 감도변수를 고려하여 저감도 비집중 준최적화 제어 문제를 설정할 수 있다. 그런데 (7), (11)식에서 보듯이 최소화하고자 하는 목적함수가 2개이므로 vector valued optimization technique (또는 multiobjective optimization technique)^[8]을 이용하여 풀 수 있는데 여기서는 그중 한 방법인 Pareto 최적화 방법을 이용하여, 두 목적함수 J_1 과 J_2 의 weighting을 똑같이 1로 하여 새로운 목적함수 J 를 $J = J_1 + J_2$ 로 정의한 후 (2)~(4), (7), (10), (11)식으로 부터 다음과 같이 저감도 비집중 준최적화 제어 문제를 설정할 수 있다.

문제 1.

$$\begin{aligned} \dot{x} &= Ax + Bu, \quad x(0) = x_0, \\ \dot{v} &= Av + Bw + Tx + Mu, \quad v(0) = 0, \\ y &= Cx, \\ \dot{z} &= y - y_s, \quad z(0) = 0, \\ u &= K(x - x_s) + Gz \\ w &= Kv + G \frac{\partial z}{\partial \alpha} \end{aligned} \quad (12)$$

로 주어지는 제약 조건하에서 비용함수(cost functional)

$$\begin{aligned} J(\cdot) &= \int_0^{\infty} \{ (x - x_s)^T Q (x - x_s) + \\ &+ (u - u_s)^T R (u - u_s) \\ &+ v^T S v + w^T R_1 w \} dt \end{aligned} \quad (13)$$

를 최소화하는 $K \in \Omega(s)$ 와 $G \in \Omega_1(s)$ 를 구하라.

Kreindler^[3]는 상기한 문제 1과 유사한 저감도 제어기 설계 문제를 연구하였다. 그러나 그가 제시한 제어기는 전체 시스템의 모든 상태 정보를 이용하도록 구성된 바, 즉 제어입력 u 를

$$u = Kv + \hat{K}v \quad (14)$$

로 하여, 이를 비집중 제어기 형태로 고치면

$$u_i = K_i x_i + \hat{K}_i v_i \quad (15)$$

로 쓸 수 있다. 이와 같이 비집중 제어기를 설계하면 (1), (8)식으로부터 다음의 (16)식을 얻을 수 있다.

$$\begin{aligned} \dot{x}_i &= [A_{ii} + B_i K_i] x_i + \sum_{j=1}^n A_{ij} v_j + B_i \hat{K}_i v_i \\ \dot{v}_i &= \frac{\partial}{\partial \alpha} [A_{ii} + B_i u_i + \sum_{j=1}^n A_{ij} x_j] \\ &+ A_{ii} v_i + B_i w_i + \sum_{j=1}^n A_{ij} v_j \\ &= \frac{\partial}{\partial \alpha} [A_{ii} + B_i K_i] x_i + \frac{\partial}{\partial \alpha} B_i \hat{K}_i v_i \\ &+ \frac{\partial}{\partial \alpha} \left[\sum_{j=1}^n A_{ij} \right] x_i + A_{ii} v_i + B_i K_i v_i \\ &+ B_i \hat{K}_i \frac{\partial^2 x_i}{\partial \alpha^2} + \sum_{j=1}^n A_{ij} v_j \end{aligned} \quad (16)$$

그런데 (16)식에서 보면 (15)식으로 주어진 제어기는 비집중화된 형태인데도 불구하고 사실상 다른 부시스템의 상태 정보를 모두 이용해야 \dot{v}_i 를 얻을 수 있으며 또한 미리 알려지지 않은 상태정보인 $x_i(t)$ 의 2차 미분 정보를 이용하므로 (15)식으로 주어진 제어기는 (3)식으로 주어진 제어기보다 대규모 시스템을 제어하기 위해서는 비현실적이라는 것을 알 수 있다. 따라서 본 논문에서는 (3)식으로 표현되는 제어기를 구성하도록 제안하고, 이 경우 제어기 설계 방법이 어떻게 바뀌는가 연구하였다.

다음 장에서는 앞에서 설정된 문제 1에서의 준최적 이득행렬 K 와 G 를 구하기 위한 필요조건들을 유도하고 반복 계산 알고리즘을 제시하기로 한다.

III. 준최적 이득행렬을 구하기 위한 필요조건들의 유도

문제 1에 기술된 조건하에서 우선 정상상태의 각 상태 변수들의 값들은 정상 상태에서 출력 오차가 없다면 $\dot{x} = 0$ 이고 $y = y_s$ 이다. 또한.

$$\text{rank} \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} = n + m \quad (17)$$

$$\text{또는 rank } (CA^{-1}B) = m$$

$$\text{또는 } \det(CA^{-1}B) \neq 0 \quad (A \text{는 non-singular})$$

이라는 가정하에서 (12)식으로부터 u_s 와 x_s 가 유일하게 다음과 같이 정해진다.^[5]

$$u_s = -(CA^{-1}B)^{-1} y_s, \quad (18)$$

$$x_s = A^{-1}B(CA^{-1}B)^{-1} y_s.$$

여기서 새로운 상태변수 $\hat{x}, \hat{u}, \hat{y}, \hat{v}$ 및 \hat{w} 를 다음과 같이 정의하기로 하자.

$$\dot{\hat{x}} = \begin{bmatrix} z - z_s \\ x - x_s \end{bmatrix}, \hat{u} = u - u_s, \hat{y} = y - y_s, \hat{v} = \frac{\partial \hat{x}}{\partial \alpha}, \hat{w} = \frac{\partial u}{\partial \alpha} \quad (19)$$

(19)식의 새로운 상태 변수들을 이용하여 (12)식을 바꿔 쓰면

$$\begin{aligned} \dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}\hat{u}, \hat{x}_s(0) = \hat{x}_s \\ \dot{\hat{v}} &= \hat{A}\hat{v} + \hat{T}\hat{x} + \hat{M}\hat{u} + \hat{B}\hat{w} \\ \hat{y} &= \hat{C}\hat{x} \end{aligned} \quad (20)$$

로 쓸 수 있고 여기서

$$\begin{aligned} \hat{A} &= \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix}, \hat{B} = \begin{bmatrix} 0 \\ B \end{bmatrix}, \hat{C} = [0 \quad C], \\ \hat{T} &= \begin{bmatrix} 0 & 0 \\ 0 & T \end{bmatrix}, \hat{M} = \begin{bmatrix} 0 \\ M \end{bmatrix}, \hat{x}_s = \begin{bmatrix} -z_s \\ x_s \quad -x_s \end{bmatrix} \end{aligned}$$

이다. 또한 이때 제어 입력은 \hat{u} 은

$$\begin{aligned} \hat{u} &= (K\hat{C}_m + G\hat{D})\hat{x}, \\ \hat{C}_m &= [0 \quad I], \hat{D} = [I \quad 0]. \end{aligned} \quad (21)$$

로 쓸 수 있고 (13)식의 비용함수(cost functional) J를 새로운 상태 변수로 바꿔 쓰면

$$\begin{aligned} J &= \int_0^\infty (\hat{x}^T \hat{Q} \hat{x} + \hat{u}^T \hat{R} \hat{u} + \hat{v}^T \hat{S} \hat{v} + \hat{w}^T \hat{R}_1 \hat{w}) dt \\ \hat{Q} &= \begin{bmatrix} 0 & 0 \\ 0 & Q \end{bmatrix}, \hat{S} = \begin{bmatrix} 0 & 0 \\ 0 & S \end{bmatrix} \end{aligned} \quad (22)$$

로 쓸 수 있다. 여기서 다시 \bar{x} 와 \bar{u} 를 다음과 같이 정의하면 (20)~(22)식을 기존의 준최적화 문제에서 처럼 보다 간단한 식의 형태로 바꿀 수 있다. 즉,

$$\bar{x} = \begin{bmatrix} \hat{x} \\ \hat{v} \end{bmatrix}, \bar{u} = \begin{bmatrix} \hat{u} \\ \hat{w} \end{bmatrix} \quad (23)$$

로 \bar{x} 와 \bar{u} 를 정의하면 (20)~(22) 식은

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}\bar{u}, \bar{x}(0) = [\hat{x}_s^T \quad 0]^T, \quad (24)$$

$$\bar{u} = \bar{K}\bar{x}, \quad (25)$$

$$J = \int_0^\infty (\bar{x}^T \bar{Q} \bar{x} + \bar{u}^T \bar{R} \bar{u}) dt \quad (26)$$

$$\bar{A} = \begin{bmatrix} \hat{A} & 0 \\ \hat{T} & \hat{A} \end{bmatrix}, \bar{B} = \begin{bmatrix} \hat{B} & 0 \\ \hat{M} & \hat{B} \end{bmatrix}, \bar{Q} = \begin{bmatrix} \hat{Q} & 0 \\ 0 & \hat{S} \end{bmatrix}$$

$$\bar{R} = \begin{bmatrix} R & 0 \\ 0 & R_1 \end{bmatrix}, \bar{K} = \begin{bmatrix} K\hat{C}_m + G\hat{D} & 0 \\ 0 & K\hat{C}_m + G\hat{D} \end{bmatrix}$$

와 같이 간단하게 쓸 수 있고, 문제 1은 결국 (24), (25)식으로 주어지는 제약조건하에서 (26)식을 최소화하는 \bar{K} 를 구하는 문제와 같게 된다. 이러한 문제는 파라미터 최적화 방법(parameter optimization technique)^(4, 6)을 이용하여 다음과 같이 풀 수 있다.

(24), (26)식에 \bar{u} 를 대입하면

$$\dot{\bar{x}} = (\bar{A} + \bar{B}\bar{K})\bar{x}, \bar{x}(0) = [\hat{x}_s^T \quad 0]^T, \quad (27)$$

$$J = \int_0^\infty \bar{x}^T (\bar{Q} + \bar{K}^T \bar{R} \bar{K}) \bar{x} dt, \quad (28)$$

로 쓸 수 있고 (27)식의 $\bar{A} + \bar{B}\bar{K}$ 가 asymptotically stable matrix이라는 가정하에 (28)식의 J는 다음의 (29)식으로 쓸 수 있다.

$$J = \text{trace} [(\bar{Q} + \bar{K}^T \bar{R} \bar{K}) F]. \quad (29)$$

단, (29)식의 F는

$$F \triangleq \int_0^\infty \bar{x} \bar{x}^T dt$$

로 정의되는 행렬로써, 다음의 Lyapunov 방정식의 positive definite 한 해의 하나이다.

$$(\bar{A} + \bar{B}\bar{K})F + F(\bar{A} + \bar{B}\bar{K})^T + V_0 = 0, \quad (30)$$

$$V_0 = \begin{bmatrix} \hat{x}_s \hat{x}_s^T & 0 \\ 0 & 0 \end{bmatrix}$$

따라서 (28)식을 최소화하는 \bar{K} 는, (30)식의 해 중에서 (29)식을 최소화하는 \bar{K} 와 같으며 이러한 \bar{K} 를 구하는 것은 잘 알려진 Lagrange multiplier 방법을 이용하여 구할 수 있다.

Lagrangian Φ 를 (12), (29), (30)식으로 부터

$$\Phi = \text{trace} (\bar{Q} + \bar{K}^T \bar{R} \bar{K}) F$$

$$+ \text{trace} \{L [(\bar{A} + \bar{B}\bar{K})F + F(\bar{A} + \bar{B}\bar{K})^T + V_0]\}$$

$$+ P^T (u_s - Gz_s) \quad (31)$$

로 정의한다. (31)식에서 L은 $2(n+m) \times 2(n+m)$ symmetric Lagrange multiplier matrix이고 P는 Lagrange vector이다. 다음 이 Lagrangian Φ 를 K, G, F, L 및 P에 대하여 각각 편미분을 하고 그들을 각각 0으로 놓으면 최적 이득행렬 K, G를 얻기 위한 필요조건을 유도할 수 있다. 그 결과를 다음의 정리 1

에 요약하였다.

정리 1 ($\bar{A} + \bar{B}\bar{K}$)가 asymptotically stable matrix 일때 \bar{K} 가 문제 1의 최적 이득 행렬이기 위한 필요 조건들은 다음과 같다.

$$\frac{\partial \Phi}{\partial L} = (\bar{A} + \bar{B}\bar{K})F + F(\bar{A} + \bar{B}\bar{K})^T + V_0 = 0, \quad (32)$$

$$\begin{aligned} \frac{\partial \Phi}{\partial F} &= (\bar{A} + \bar{B}\bar{K})^T L + L(\bar{A} + \bar{B}\bar{K}) + \bar{Q} \\ &+ \bar{K}^T \bar{R} \bar{K} = 0, \end{aligned} \quad (33)$$

$$\frac{\partial \Phi}{\partial P} = u_s - Gz_s = 0 \quad (34)$$

$$\begin{aligned} \frac{\partial \Phi}{\partial K} &= 2RK \hat{C}_m F_{11} \hat{C}_m^T + 2R_1 K \hat{C}_m F_{22} \hat{C}_m^T \\ &+ 2RG \hat{D} F_{11} \hat{C}_m^T + 2R_1 G \hat{D} F_{22} \hat{C}_m^T \\ &+ 2\hat{B}^T (L_{11} F_{11} + L_{12} F_{21} + L_{21} F_{12} + L_{22} F_{22}) \\ &\hat{C}_m^T + 2\hat{M}^T (L_{21} F_{11} + L_{22} F_{21}) \hat{C}_m^T = 0, \end{aligned} \quad (35)$$

$$\begin{aligned} \frac{\partial \Phi}{\partial G} &= 2RK \hat{C}_m F_{11} \hat{D}^T + 2R_1 K \hat{C}_m F_{22} \hat{D}^T + 2RG \\ &\hat{D} F_{11} \hat{D}^T + 2R_1 G \hat{D} F_{22} \hat{D}^T + 2\hat{B}^T (L_{11} F_{11} \\ &+ L_{12} F_{21} + L_{21} F_{12} + L_{22} F_{22}) \hat{D}^T \\ &+ 2\hat{M}^T (L_{21} F_{11} + L_{22} F_{21}) \hat{D}^T - Pz_s^T = 0, \end{aligned} \quad (36)$$

여기서

$$F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix}.$$

위의 정리 1에서, (32), (33) 식으로 부터 F와 L 행렬을 얻을 수 있는데 이때 F와 L은 모두 $2(n+m) \times 2(n+m)$ 차원의 행렬들이므로 계산시에 매우 큰 계산능력을 필요로 하게된다. 이러한 어려움을 해결하기 위해 다음과 같은 계산 알고리즘을 제시한다. $\bar{A} + \bar{B}\bar{K}$ 행렬을 보면

$$\bar{A} + \bar{B}\bar{K} = \begin{bmatrix} \hat{A} + \hat{B}(\hat{K}_m + \hat{G}\hat{D}) & 0 \\ \hat{T} + \hat{M}(\hat{K}_m + \hat{G}\hat{D}) & \hat{A} + \hat{B}(\hat{K}_m + \hat{G}\hat{D}) \end{bmatrix}$$

$$\Delta = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}$$

이고 (37) 식에서 $\Lambda_{11} = \Lambda_{22}$, $\Lambda_{12} = 0$ 이므로 이를 이용하여 (32), (33) 식을 풀어 쓰면

$$\Lambda_{11} F_{11} + F_{11} \Lambda_{11}^T = -\hat{\alpha}_0 \hat{\alpha}_0^T, \quad (38)$$

$$\Lambda_{11} F_{21} + F_{21} \Lambda_{11}^T = -F_{11} \Lambda_{21}, \quad (39)$$

$$\Lambda_{11} F_{22} + F_{22} \Lambda_{11}^T = -(\Lambda_{21} F_{12} + F_{21} \Lambda_{21}^T), \quad (40)$$

$$\begin{aligned} \Lambda_{11}^T L_{22} + L_{22} \Lambda_{11} &= -[\hat{S} + (\hat{K}_m + \hat{G}\hat{D})^T \\ &R_1 (\hat{K}_m + \hat{G}\hat{D})], \end{aligned} \quad (41)$$

$$\Lambda_{11}^T L_{21} + L_{21} \Lambda_{11} = -L_{22} \Lambda_{21}, \quad (42)$$

$$\begin{aligned} \Lambda_{11}^T L_{11} + L_{11} \Lambda_{11} &= -[\hat{Q} + (\hat{K}_m + \hat{G}\hat{D}) R (\hat{K}_m \\ &+ \hat{G}\hat{D}) + L_{12} \Lambda_{21} + \Lambda_{21}^T L_{21}], \end{aligned} \quad (43)$$

로 쓸 수 있다. 따라서 F행렬을 계산할 때에 (38)식에서 우선 F_{11} 을 구하고 그 값을 이용하여 (39)식에서 F_{21} 을 구하고 또 $F_{12} = F_{21}^T$ 이므로 그 값을 이용하여 (40)식에서 F_{22} 를 구하면 F행렬 전체를 구할 수 있다. L행렬도 마찬가지로 (41)~(43)식에서 차례로 구할 수 있다. 이와같이 하여 F, L행렬들을 계산할 때에는 $(n+m) \times (n+m)$ 차원의 행렬들을 계산하면 되므로 그 계산량이 훨씬 감소하게 된다. 이때에 (39), (40) 또는 (42), (43)식은 그 형태는 Lyapunov 식과 같지만 우변의 행렬 값이 일반적으로 negative semidefiniteness를 만족하지 않으므로 따라서 구한 F_{21} , F_{22} 또는 L_{21} , L_{11} 행렬들은 positive definite 해가 아닐 수도 있다. 그러나 원래의 (32), (33)식은 Lyapunov 식이므로 $(\bar{A} + \bar{B}\bar{K})$ 행렬이 asymptotically stable matrix이면 F와 L행렬은 항상 positive definite하다. 따라서 앞의 계산 알고리즘을 이용하여 구한 F_{21} , F_{22} 또는 L_{21} , L_{11} 들은 각각 positive definite 하지 않을 수도 있지만 이들로 이루어진 F 및 L행렬은 항상 positive definite하다.

앞의 정리 1로부터 준최적 이득행렬 \bar{K} 를 계산하는 반복 계산 알고리즘을 다음에 제시하기로 한다. 우선 feasible direction matrix H_K 와 H_G 를 각각

$$H_K = \text{block diag}(H_{K_1}, \dots, H_{K_N}) \in R^{m \times n}$$

$$H_G = \text{block diag}(H_{G_1}, \dots, H_{G_N}) \in R^{m \times m}$$

로 정의하면 이들 H_k 와 H_c 는 각각 $\partial\Phi/\partial K$ 와 $\partial\Phi/\partial G$ 를 계산하여 블록 대각선 요소(block diagonal entry)들을 제외한 나머지 부분을 0으로 놓은 행렬로 되므로 $H_k \in \Omega(s)$ 이고 $H_c \in \Omega_1(s)$ 이다. 또한 $\Sigma_1(s)$ 를

$$\Sigma_1(s) = \{K \mid K \in \Omega(s), G \in \Omega_1(s), \bar{A} + \bar{B}K$$

is asymptotically stable}

로 정의하고 K 와 G 의 초기 행렬 값을 $\bar{K} \in \Sigma_1(s)$ 가 되도록 주어졌다고 가정한다. 그러면 H_k 와 H_c 를 이용하여 반복계산 알고리즘을 다음과 같이 제시한다.

[알고리즘]

Step 1°. 반복지수 $\ell = 0$ 로 하고, $K^{(\ell)}, G^{(\ell)}$ 을 추측한다.

Step 2°, (38)~(43)으로 부터 $F^{(\ell)}, L^{(\ell)}$ 을 계산하고 (35), (36)으로 부터 $\partial\Phi/\partial K, \partial\Phi/\partial G$ 를 계산한다.

Step 3°. $H_k^{(\ell)}, H_c^{(\ell)}$ 을 계산한다.

Step 4°. 수렴도 조사 : 양의 상수 ϵ_1, ϵ_2 에 대해 $\|H_k^{(\ell)}\| < \epsilon_1, \|H_c^{(\ell)}\| < \epsilon_2$ 를 만족하면 stop하고 만족하지 않으면 step 5°로 간다.

Step 5°. 다음 반복 계산을 위해

$$K^{(\ell+1)} = K^{(\ell)} - a H_k^{(\ell)}, G^{(\ell+1)} = G^{(\ell)} - a H_c^{(\ell)}$$

로 새로운 K 와 G 를 계산하고 $\ell = \ell + 1$ 로 하여 step 2°로 간다.

위의 반복 계산 알고리즘 step 5°에서 a 는 step size로써

$$J(K - a H_k, G - a H_c) < J(K, G)$$

를 만족하는 적당한 양의 상수로 한다. 또한 매 반복 계산시의 $\hat{A} + \hat{B}(K \hat{C}_m + G\hat{D})$ 의 점근 안정성(asymptotical stability)은 Geromel과 Bernussou^[2]가 제시한 바와 같이 초기 이득 행렬 K 와 G 를 $\hat{A} + \hat{B}(K \hat{C}_m + G\hat{D})$ 가 안정 행렬이 되도록 선택하면 항상 보장된다.

다음 장에서는 앞의 결과를 이용하여 간단한 예제를 풀어 보기로 한다.

※ 그림 2, 3에서 실선은 감도를 고려한 경우, 점선은 감도를 고려하지 않은 경우의 응답이다.

VI. 수치 예제

다음과 같은 2개의 선형 부시스템이 상호 연결된 대규모 시스템을 생각하자.

$$(S1) \dot{x}_1 = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2+0.1\alpha & -3+0.1\alpha \end{bmatrix} x_1 +$$

$$+ \begin{bmatrix} 0 & 0 \\ 0.5+0.1\alpha & 1+0.1\alpha \end{bmatrix} x_2 + \begin{bmatrix} 0 \\ 1+0.1\alpha \end{bmatrix} u_1$$

$$y_1 = [1 \quad 0] x_1,$$

$$(S2) \dot{x}_2 = \begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -3+0.1\alpha & -4+0.1\alpha \end{bmatrix} x_2$$

$$+ \begin{bmatrix} 0 & 0 \\ 1+0.1\alpha & 0.5+0.1\alpha \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ 1+0.1\alpha \end{bmatrix} u_2$$

$$y_2 = [1 \quad 0] x_2$$

$$x_1^T = [1 \quad 1 \quad 1 \quad 1], y_1^T = [0.5 \quad 0.5]$$

$$J(\cdot) = \sum_{i=1}^2 \int_0^\infty \{ (x_i - x_{si})^T Q_i (x_i - x_{si}) + (u_i - u_{si})^T \cdot R_i (u_i - u_{si}) + v_i^T S_i v_i + w_i^T R_{ii} w_i \} dt$$

여기서 가중행렬 Q_i, R_i, S_i, R_{ii} 는 전부 항등 행렬로 선택했다. 또한 α 의 명목치는 0으로하고 α 가 변하는 구역 I_α 는 $I_\alpha = [0, 10]$ 으로 한다. $\alpha = 0$ 일 때의 최적 이득행렬과 비용함수 J 의 값, 또 α 가 I_α 내에서 변할 때의 비용함수 값 등을 표 1에 보였고, 표 1의 최적 이득행렬로 설계된 시스템에서 α 가 변할 때의 출력응답을 감도를 고려하지 않은 경우와 비교하여 그림 2, 그림 3에 보였다. 표 1에서 보듯이 감도를 고려한 경우의 비용함수 값이 그렇지 않은 경우의 값에 비해 훨씬 작고 또한 $\alpha = 10$ 인 경우는 감도를 고려한 경우 시스템은 여전히 안정하나 감도를 고려하지 않은 경우 시스템은 불안정 해졌다는 것을 알 수 있다. 또 그림 2, 그림 3에서 감도를 고려한 경우의 출력응답이 더 빠른 정착시간을 보인다는 것을 알 수 있다.

V. 結 論

본 논문에서는 대규모 시스템을 위한 저감도 비집중 준최적 서어보(servo) 제어를 제시함과 동시에, 제어기의 이득행렬을 구하기 위한 필요조건을 유도하였고, 반복계산 알고리즘을 제시하였다. 제시된 제어기가 시스템의 외란에 의해 혼란되었을 때도 비용함수 값, 정착시간, overshoot 등에 대해 시스템을 잘 제어한다는 것을 수치적 예를 들어 보였다. 본 방법은 저감도 비집중 상태 레환 안정화 제어기 설계에 직접 응용할 수 있다.^[7]

參 考 文 獻

[1] M.G. Singh and M. Hassan, "A robust decentralized controller for intercon-

nected dynamical systems”, *Proc. IEEE*, vol. 125, 1978.

[2] J. C. Geromel and J. Bernussou, “An algorithm for optimal decentralized regulation of linear quadratic interconnected systems”, *Automatica*, vol. 15, no. 4, 1979.

[3] E. Kreindler, “On minimization of trajectory sensitivity”, *Int. J. Control*, vol. 8, no. 1, 1968.

[4] J. O’Reilly, “Low sensitivity feedback controllers for linear systems with incomplete state information”, *Int. J. Control*, vol. 29, no. 6, 1979.

[5] S. Fukata, A. Mohri and M. Takata, “On the determination of the optimal feedback gains for multivariable linear systems incorporating integral action”, *Int. J. Control*, vol. 31, no. 6, 1980.

[6] W.S. Levine and M. Athans, “On the determination of the optimal constant output feedback gains for linear multivariable systems”, *IEEE Trans. AC*, vol. AC-15, no. 1, 1970.

[7] S.R. Oh, *A Study on Low Sensitivity Local Suboptimal Controllers*. KAIST,

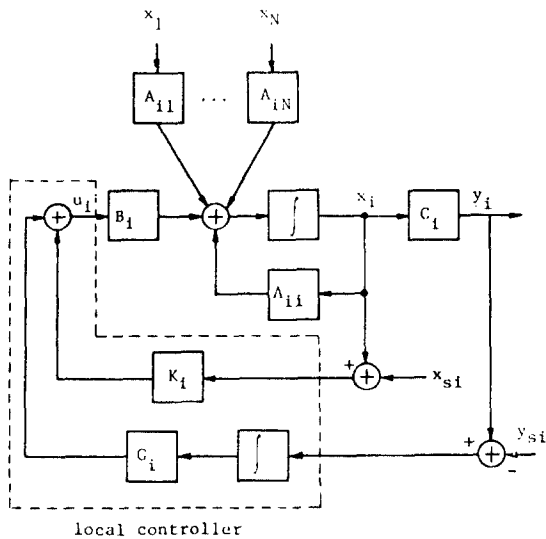


그림 1. 비집중 제어기로 구성된 i-번째 부시스템의 블럭도

Fig. 1. Block diagram of i-th subsystem with local controller.

MS. Thesis, 1982.

[8] M.E. Salukvadze, *Vector-Valued Optimization Problems in Control Theory*. Academic Press, New York, 1979.

표 1. 감도를 고려한 경우와 고려하지 않은 경우의 비교

Table 1. Comparison of proposed suboptimal controller and conventional suboptimal controller.

	감도를 고려한 경우	감도를 고려하지 않은 경우
최적이득 행렬 K^*	$\begin{bmatrix} 0.296 & -4.662 & 0 & 0 \\ 0 & 0 & -0.090 & -3.695 \end{bmatrix}$	$\begin{bmatrix} 0.520 & -4.656 & 0 & 0 \\ 0 & 0 & -0.696 & -3.62 \end{bmatrix}$
최적이득 행렬 G^*	$\begin{bmatrix} -2.031 & 0 \\ 0 & -2.435 \end{bmatrix}$	$\begin{bmatrix} -2.263 & 0 \\ 0 & -2.789 \end{bmatrix}$
$\alpha = 0$ 일때 J^*	6.104	6.641
$\alpha = 1$ 일때 J^*	6.350	7.631
$\alpha = 5$ 일때 J^*	12.252	21.251
$\alpha = 10$ 일때 J^*	30.697	시스템이 불안정

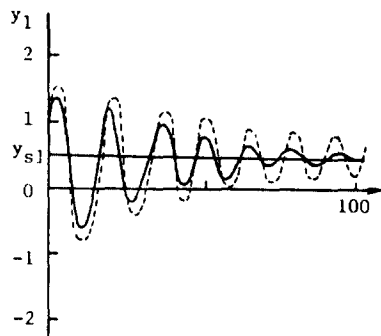


그림 2. $\alpha = 5$ 일 때의 y_1 궤적
Fig. 2. y_1 trajectory when $\alpha = 5$.

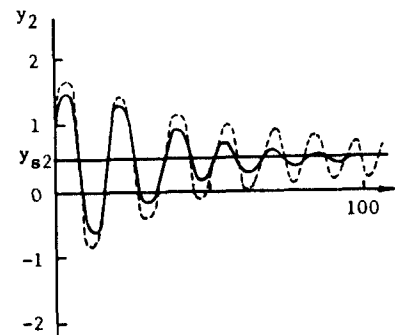


그림 3. $\alpha = 5$ 일 때의 y_2 의 궤적
Fig. 3. y_2 trajectory when $\alpha = 5$. ***