

연결식 대형시스템을 위한 분산 동적 표면 제어

Decentralized Dynamic Surface Control
for Large-Scale Interconnected Systems송 봉 섭*
(Bongsob Song)

Abstract : An analysis methodology of Decentralized Dynamic Surface Control (DDSC) for the large-scale interconnected nonlinear systems is presented in this paper. While the centralized DSC approach proposed in [14] has a difficulty to check the quadratic stability for the large-scale systems numerically due to dramatic increases of the order of overall augmented error dynamics, DDSC is relatively easy to check the quadratic stability since lower order error dynamics of individual subsystems are used. Then, a systematic procedure for designing DDSC will be developed. Furthermore, after a quadratic function containing a reachable set is defined, it will be calculated numerically to indicate the performance of DDSC in the framework of convex optimization. Finally an illustrative example will be given for showing the advantages of DDSC compared with other decentralized nonlinear control techniques.

Keywords : decentralized nonlinear control, decentralized dynamic surface control, interconnected systems

I. 서론

전기, 전자, 그리고 컴퓨터 분야의 눈부신 발전에 힘입어 지능형 교통시스템(ITS), 반도체 및 디스플레이 생산라인, 원자력 발전소 등과 같은 대형(large-scale)시스템을 위한 제어시스템이 복잡해지고 대형화되고 있다. 이러한 대형시스템은 수학적으로 수많은 부분시스템(subsystem)이 연결된 연결식(interconnected) 시스템으로 모델링이 되고 있으며 이러한 연결식 시스템에 대한 연구가 최근 들어 더욱 활발해지고 있다 [1,2]. 일반적으로 시스템이 복잡해질수록 이를 담당하고 있는 중앙식(centralized) 제어기는 통신을 통하여 많은 정보를 교환할 수 있어야 하고 높은 계산능력이 필요하다. 이로 인하여 고성능의 제어기 하드웨어가 요구되고, 특히 제어기의 부분고장 시 시스템 전체가 영향을 받을 수 밖에 없다. 이러한 중앙식 제어방법의 한계점을 보완하기 위한 방법으로 1970년대부터 중앙식 제어기가 아닌 다수의 로컬제어기를 이용하는 분산식(decentralized) 제어방법이 연구되어 왔다 ([1,2]의 참조문헌 참조). 좀 더 구체적으로 서술하면, 선형시스템들 사이에 비선형 연결관계(interactions)를 가지고 있는 시스템에 대해서 분산식 적응제어가 개발되어왔고 ([3-5] 참조), 더 나아가 모든 상태 변수(state)를 측정할 수 없는 시스템에 대한 분산식 적응제어에 대해서도 많은 연구 결과를 문헌에서 찾아볼 수가 있다([6-8] 참조). 다만 지난 30여 년간 분산제어라는 연구 영역에 커다란 학문적 발전이 있었음에도 불구하고 비선형 시스템에 대한 분산식 제어방법에 대한 연구는 상대적으로 문헌에서 찾아보기가 힘들며 1990년대에 이르러서야 제안되기 시작했다. 예를 들어서, 연결식 비선형 시스템을 위한 분산식 적응제어방식을 [9,10]과 같은 문헌에서 찾아 볼 수 있다. 하지만, 이러한 대부분의 연구는 백스텝핑(backstepping)이라는 비선형제어기법을 이용한 것으로서

제어기 설계 시 제어 로직 자체가 복잡해 실제 활용 및 응용 시 많은 제한이 따를 수 있다[11].

본 논문에서는 좀 더 효율적으로 연결식 대형시스템을 제어하기 위하여 비선형제어기법 중 하나인 동적 표면 제어기법을 연결식 비선형 시스템으로 응용하기 위한 분산 동적 표면 제어(Decentralized Dynamic Surface Control - DDSC) 방법에 대해서 살펴보고자 한다. 동적 표면 제어 방법은 차량의 엔진제어를 위해서 처음 소개된 방법으로서 그 이후에 차량 서스펜션, 브레이크 등 차량 부품 제어를 비롯하여 무인자동차의 종방향 제어기 설계에 관련된 기법으로 많이 응용되어 왔다[12,13]. 또한, 이론적으로는 안정성에 대한 연구 및 제어기의 설계방법을 문헌에서 찾아 볼 수 있다[11,14].

본 논문에서는 중앙식 동적 표면 제어방법을 대형시스템에 적용하기 위하여 분산 동적 표면 제어방법으로 발전시키고자 한다. 수학적 기법을 이용하여 분산 동적 표면 제어의 설계 및 성능 예측에 대한 수학적 유도과정을 설명하겠지만 궁극적으로는 실제 응용 및 활용적 측면에서 다른 분산식 제어방법과 비교 시 훨씬 설계가 간단하고 계산량이 적음을 보이고자 한다. 본 논문은 크게 세 부분으로 구성되어 있다. 우선은 우리가 다루려고 하는 연결식 비선형 시스템을 II장에서 정확히 명시하고 이를 위한 제어기 설계방법을 III장에서 설명하고자 한다. 두 번째는 시스템의 안정성 및 강인성을 보장하기 위한 제어기 상수를 설계하는 방법론에 대해서 살펴보고자 한다. 마지막으로 IV장에서는 예제를 통하여 제안된 제어기법의 검증 및 문헌에 소개되어 있는 다른 제어방법과 비교하고자 한다.

II. 문제의 정의

우리가 다루려고 하는 연결식 비선형 시스템은 N개의 부분시스템으로 연결되어 있으며 i번째 비선형 시스템은 다음과 같이 표현된다.

$$\dot{\mathbf{x}}_i = U_i \mathbf{x}_i + B_i u_i + \mathbf{f}_i(\mathbf{x}_i) + \sum_{j=1, j \neq i}^N \mathbf{g}_{ij}(\mathbf{x}_j) \quad (1)$$

* 책임저자(Corresponding Author)

논문접수 : 2005. 10. 20., 채택확정 : 2006. 1. 6.

송봉섭 : 아주대학교 기계공학부(bsong@ajou.ac.kr)

※ 이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구입(KRF-2005-003-D00013).

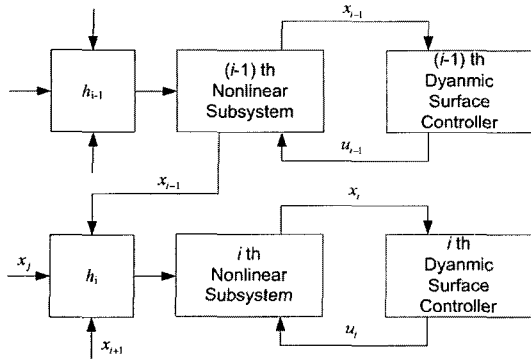


그림 1. DSC를 포함한 연결식 시스템.
Fig. 1. Schematic of interconnected nonlinear systems with DDSC.

$$= U_i x_i + B_i u_i + f_i(x_i) + h_i$$

여기서 $i = 1, \dots, N$, $x_i \in \mathbf{R}^{n_i}$, $x_j \in \mathbf{R}^{n_j}$ 는 상태변수, $u_i \in \mathbf{R}$ 는 제어입력이며, 위의 각각의 비선형 시스템은 strict-feedback 비선형 시스템이라고 불리는데 [15] 이는 U_i 행렬과 B_i 가 다음과 같이 정의된다.

$$U_i = \text{diag}([1, \dots, 1], 1) \in \mathbf{R}^{n_i \times n_i}, B_i = [0 \dots 0 \ 1]^T$$

여기서 $\text{diag}(x, i)$ 는 i 번째 super-diagonal에 x 벡터로 형성되어 있는 정방행렬을 뜻한다. 여기서, $\text{diag}([1, \dots, 1], 1)$ 는 행렬의 첫 번째 super-diagonal에 $[1, \dots, 1]$ 의 벡터로 이루어진 $n_i \times n_i$ 정방행렬을 의미한다. 또, 비선형 함수 항의 k 번째인 f_{ik} 가 오직 x_{i1}, \dots, x_{ik} 의 함수로 이루어져 있으며 부분시스템과의 연결식을 g_{ij} 로 나타내었다. 마지막으로 이러한 시스템을 위한 추가적인 가정들은 다음과 같이 요약될 수 있다.

- i) 비선형 함수 $f_{ik} : \mathbf{R}^k \rightarrow \mathbf{R}$ 은 C^∞ 함수이며 Lipschitz이다.
- ii) 비선형 함수 $g_{ij} : \mathbf{R}^{n_j} \rightarrow \mathbf{R}^{n_i}$ 는 주어지지 않았지만 다음과 같이 제한되어 있다.

$$\mathbf{h}_i^T \mathbf{h}_i \leq c$$

여기서 c 가 1인 경우, $\mathbf{h}_i = \sum_{j=1, j \neq i}^N g_{ij}(x_j)$ 를 단위피크(unit-peak) 함수라고 한다[16].

그림 1에서 보는 바와 같이 각각의 부분시스템에 분산식 제어기를 설계하는 것이다. 이 때, \mathbf{h}_i 에 대한 정보는 알 수 없다는 가정하에서 설계가 되어야 한다.

III. 분산 동적 표면 제어 (DDSC)

연결식 비선형 시스템에 대해서 분산 동적 표면 제어 방법을 통하여 좀 더 체계적으로 쉽게 제어기 설계가 가능함을 보이려고 한다. 본 장의 구성은 연결식 비선형 시스템에 대해서 분산 동적 표면 제어방법을 적용하는 방법과 확장된 오차방정식(augmented error dynamics)을 수학적으로 유도하고 이를

바탕으로 reachable sets ([16] 참조)을 볼록 함수 최적화(convex optimization)을 이용하여 수치적으로 예측해보고자 한다.

1. DDSC의 초기설계

(1)을 스칼라형태로 다시 써 보면 다음과 같이 수학적으로 기술할 수 있다.

$$\begin{cases} \dot{x}_{ik} = x_{i(k+1)} + f_{ik}(x_{i1}, \dots, x_{ik}) + h_{ik} & \text{for } 1 \leq k \leq n_i - 1 \\ \dot{x}_{in_i} = u_i + f_{in_i}(x_{i1}, \dots, x_{in_i}) + h_{in_i} \end{cases} \quad (2)$$

여기서 h_{ik} 는 h_i 의 k 번째 함수이며 연결된 다른 부분시스템의 상태변수에 영향을 미치는 함수이다. 즉,

$$h_{ik} = h_{ik}(\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_N)$$

우선 우리가 추종하는 값을 $x_{i1,d}$ 라고 했을 경우 첫 번째 오차표면은 다음과 같이 정의한 후 미분을 하면 다음과 같다.

$$\begin{aligned} S_{ik} &= x_{ik} - x_{ik,d} \\ \dot{S}_{ik} &= \dot{x}_{ik} - \dot{x}_{ik,d} = x_{i(k+1)} + f_{ik} + h_{ik} - \dot{x}_{ik,d} \end{aligned}$$

다음으로 $x_{i(k+1)}$ 을 제어기의 입력으로 가상적으로 생각하면 원하는 $x_{i(k+1)}$ 는 다음과 같이 계산될 수 있다[11].

$$\bar{x}_{i(k+1)} := -f_{ik} - h_{ik} + \dot{x}_{ik,d} - \lambda_{ik} S_{ik}$$

하지만, 부분시스템 사이의 상태변수 정보를 알 수 없는 경우이므로 가상적 입력 값을 다음과 같이 정의한다.

$$\bar{x}_{i(k+1)} := -f_{ik} + \dot{x}_{ik,d} - \lambda_{ik} S_{ik} \quad (3)$$

다음 단계로는 제어기의 복잡한 계산을 피하기 위해서 저주파 필터를 사용한다. 즉, 다음 오차표면을 정의하기 위해 (3)의 가상적 입력 값을 이용하여 다음과 같이 원하는 입력 값을 계산한다.

$$\tau_{i(k+1)} \dot{\bar{x}}_{i(k+1),d} + \bar{x}_{i(k+1),d} = \bar{x}_{i(k+1)} \quad (4)$$

여기서, $x_{i(k+1),d}(0) = \bar{x}_{i(k+1)}(0)$. 최종적으로 위의 과정을 $k = n_i - 1$ 까지 반복을 하면 최종적으로 i 번째 부분시스템의 원하는 입력 값을 구할 수 있다.

$$u_i = -f_{in_i} + \dot{x}_{in_i,d} - \lambda_{in_i} S_{in_i} \quad (5)$$

그림 2는 지금까지의 DDSC의 초기설계 순서를 도식적으로 설명하고 있다. 하지만, 원하는 성능을 만족시키기 위해서는 제어상수가 적절히 선택이 되어야 제어기 설계가 완료된다고 할 수 있다. 이를 위해 좀 더 해석적 방법이 필요하게 되었고 다음 절에서 살펴보고자 한다.

2. 오차방정식

좀 더 체계적인 해석을 하기 위해서 앞 절에서 제안한 제어기를 사용했을 경우, 그에 해당하는 오차방정식을 유도해 보고자 한다. 우선 (2)에 (3)과 (4)에서 정의된 $\bar{x}_{i(k+1)}$,

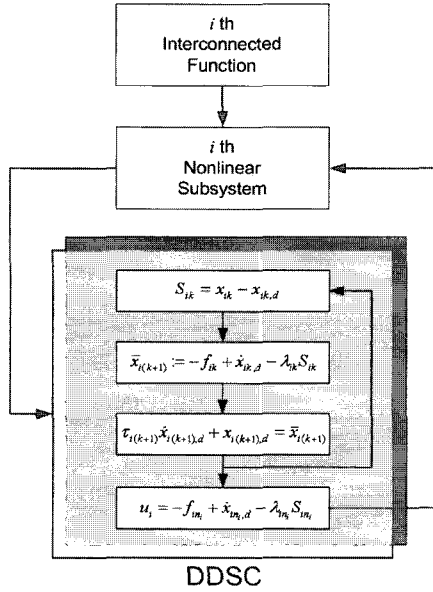


그림 2. i 번째 부분시스템을 위한 DDSC의 설계순서도.
Fig. 2. Design procedure of DDSC for the i -th subsystem.

$x_{i(k+1),d}$ 를 더하고 빼면 다음과 같이 오차로 이루어진 방정식을 유도할 수 있다.

$$\begin{aligned} \dot{x}_{ik} &= [x_{i(k+1)} - x_{i(k+1),d}] + [x_{i(k+1),d} - \bar{x}_{i(k+1)}] \\ &\quad + \bar{x}_{i(k+1)} + f_{ik}(x_{i1}, \dots, x_{ik}) + h_{ik} \\ &= S_{i(k+1)} + \xi_{i(k+1)} + \dot{x}_{i(k+1),d} - \lambda_{ik} S_{ik} + h_{ik} \\ \Rightarrow \begin{cases} \dot{S}_{ik} &= S_{i(k+1)} + \xi_{i(k+1)} - \lambda_{ik} S_{ik} + h_{ik} \\ \dot{S}_{in_i} &= -\lambda_{in_i} S_{in_i} + h_{in_i} \end{cases} \end{aligned} \quad (6)$$

여기서 필터오차 $\xi_{i(k+1)} := x_{i(k+1),d} - \bar{x}_{i(k+1)}$. 저주파 필터의 추가로 전체 시스템의 오차방정식을 구하기 위해서는 필터시스템의 오차방정식도 고려가 되어야 한다. 따라서, (3)과 (4)를 이용하여 필터오차에 대한 미분방정식은 다음과 같이 유도된다.

$$\begin{aligned} \dot{\xi}_{i(k+1)} &= \dot{x}_{i(k+1),d} - \dot{\bar{x}}_{i(k+1)} \\ &= -\frac{\xi_{i(k+1)}}{\tau_{i(k+1)}} + \dot{f}_{ik} - \ddot{x}_{i(k+1),d} + \lambda_{ik} \dot{S}_{ik} \\ \Rightarrow \begin{cases} -\lambda_{i1} \dot{S}_{i1} + \dot{\xi}_{i2} &= -\frac{\xi_{i2}}{\tau_{i2}} + \dot{f}_{i1} - \ddot{x}_{i1,d} \\ -\lambda_{ik} \dot{S}_{ik} + \dot{\xi}_{i(k+1)} - \frac{\dot{\xi}_{ik}}{\tau_{ik}} &= -\frac{\xi_{i(k+1)}}{\tau_{i(k+1)}} + \dot{f}_{ik} \end{cases} \end{aligned} \quad (7)$$

(6)와 (7)을 통합하여 아래와 같이 확장된 오차방정식 (augmented error dynamics)를 구할 수 있다.

$$\begin{bmatrix} I_n & 0 \\ -T_{S_i} & T_{\xi_i} \end{bmatrix} \begin{bmatrix} \dot{S}_i \\ \dot{\xi}_i \end{bmatrix} = \begin{bmatrix} A_{i1} & I_{n-1} \\ 0 & A_{i2} \end{bmatrix} \begin{bmatrix} S_i \\ \xi_i \end{bmatrix} + \begin{bmatrix} 0 \\ I_{n-1} \end{bmatrix} p_i + \begin{bmatrix} 0 & I \\ B_{r2} & 0 \end{bmatrix} w_i \quad (8)$$

여기서 벡터들은 다음과 같이 정의되며,

$$\begin{aligned} S_i &= [S_{i1}, \dots, S_{in_i}]^T \in R^n \\ \xi_i &= [\xi_{i2}, \dots, \xi_{in_i}]^T \in R^{n-1} \\ p_i &= [f_{i1}, \dots, f_{in_i-1}]^T \in R^{n-1} \\ w_i &= [\ddot{x}_{i1,d} \ h_i] \in R^{n+1} \end{aligned}$$

행렬 내에 부분행렬은 다음과 같이 정의된다.

$$\begin{aligned} A_{i1} &= U_i - \text{diag}(\lambda_{i1}, \dots, \lambda_{in_i}) \\ A_{i2} &= -\text{diag}(1/\tau_{i2}, \dots, 1/\tau_{in_i}) \\ T_{S_i} &= [\text{diag}(\lambda_{i1}, \dots, \lambda_{i(n-1)}) \ 0] \in R^{(n-1) \times n_i} \\ T_{\xi_i} &= I_{n-1} + \text{diag}\left(\left[-\frac{1}{\tau_{i2}}, \dots, -\frac{1}{\tau_{i(n-1)}}\right], -1\right) \end{aligned}$$

(8)의 왼편에 있는 2×2 블록행렬(block matrix)은 역행렬이 항상 존재하므로 다음과 같이 표현될 수 있다[14].

$$\dot{z}_i = A_i z_i + B_{pi} p_i + B_{wi} w_i \quad (9)$$

여기서 $z_i = \begin{bmatrix} S_i \\ \xi_i \end{bmatrix}$ 이고 주어진 행렬은 다음과 같다.

$$\begin{aligned} A_i &= \begin{bmatrix} A_{i1} & I \\ T_{\xi_i}^{-1} T_{S_i} A_{i1} & T_{\xi_i}^{-1} (T_{S_i} I - A_{i2}) \end{bmatrix}, \\ B_{pi} &= \begin{bmatrix} 0 \\ T_{\xi_i}^{-1} \end{bmatrix}, \quad B_{wi} = \begin{bmatrix} I \\ T_{\xi_i}^{-1} T_{S_i} \end{bmatrix} \end{aligned}$$

여기서 주목해야 할 점이 두 가지가 있다. 첫 번째, (9)는 [14]에서 유도된 중앙식 DSC의 오차방정식과 비슷해 보이지만 p_i 와 w_i 의 정의가 다름을 알 수 있다. 두 번째는 중앙식 DSC를 이용한다면 최종적인 오차방정식은 다음 아래와 같이 유도가 될 것이다.

$$\begin{bmatrix} \dot{z}_1 \\ \vdots \\ \dot{z}_N \end{bmatrix} = \begin{bmatrix} A_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_N \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} + \begin{bmatrix} B_{p1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_{pN} \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix} + \begin{bmatrix} B_{w1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_{wN} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}$$

하지만, 여러 부분시스템으로 연결이 되어 있다면 전체 오차방정식의 차수가 연결되어 있는 부분시스템의 개수와 비례하므로 고차원의 연립미분방정식이 얻어지고 안정성이나 성능을 해석하기에 많은 어려움이 예상된다. 따라서, 다음 절에서 이 문제점을 해결하기 위한 방안이 논의된다.

3. Reachable Set의 예측

본 절에서는 좀 더 쉽게 DDSC를 이용한 연결식 비선형

시스템의 성능을 예측하기 위한 방법이 논의된다. 이를 위해서 우선 다음과 같은 두 가지 정의를 이용하고자 한다.

정의 1: DDSC 제어기의 상수 값이 다음과 같이 주어진 시스템을 생각하자.

$$\Lambda_i = \{\lambda_{i1}, \dots, \lambda_{in_i}\}, \tau_i = \{\tau_{i1}, \dots, \tau_{in_i}\}$$

(10)와 같이 표현되는 집합 R_{wp} 는 (9)과 같은 오차방정식의 reachable set이라고 한다.

$$R_{wp} = \left\{ z_i(T) \mid \begin{array}{l} (9) \text{를 만족하는 } z_i, p_i, w_i, \\ z_i(0) = 0, T \geq 0 \end{array} \right\} \quad (10)$$

정의 2: 다음과 같은 이차함수(quadratic function) $V(z_i)$ 가 존재한다면,

$$\begin{aligned} V(z_i) &= z_i^T P z_i \text{ with } P > 0, \text{ and} \\ \frac{dV(z_i)}{dt} &\leq 0 \text{ for all } z_i, p_i, w_i \text{ satisfying} \\ (9), w_i^T w_i &\leq 1 \text{ and } V(z_i) \geq 1 \end{aligned}$$

다음과 같은 타원함수는

$$\varepsilon = \{z_i \mid z_i^T P z_i \leq 1\} \quad (11)$$

researchable set R_{wp} 를 포함한다[16].

정의 2는 [16]의 6장에 소개가 되어 있으며 이 정의를 이용함으로써 앞에서 언급했듯이 오차방정식 해석 시 전체 시스템을 불연속(piece-wise)인 부분시스템으로 해석함으로써 부분시스템의 최고차수 단위로 이차안정성(quadratic stability) 해석이 가능하고 더 나아가 reachable set을 포함하는 타원함수를 수치적으로 계산을 할 수 있다. 이 방법은 다음과 같이 요약될 수 있다.

정리 1: 다음과 같은 조건을 만족하는 DDSC 제어기의 상수 값 Λ_i, τ_i 이 존재한다면,

- i) $|p_i| \leq C_{zi} z_i + D_{wi} w_i$ 를 만족하는 C_{zi} 와 D_{wi} 가 존재하고
- ii) 다음과 같은 행렬부등식을 만족하는 P, Σ, α 가 존재한다면,

$$\begin{bmatrix} A_i^T P + P A_i + \alpha P + C_{zi}^T \Sigma C_{zi} & P B_{pi} & P B_{wi} \\ B_{pi}^T P & -\Sigma & 0 \\ B_{wi}^T P & 0 & -\alpha I \end{bmatrix} \leq 0, \quad (12)$$

타원함수 $\varepsilon = \{z_i \mid z_i^T P z_i \leq 1\}$ 는 (9)에 주어진 오차방정식의 reachable set을 포함한다.

증명: 위 정리에 대한 자세한 증명은 [14,16]에서 사용된 증명방법을 이용하여 유사하게 할 수 있으므로 생략한다. ■

여기서 정리 1의 결과를 사용시 주의해야 할 점을 언급하고자 한다. 정리 1의 결과를 이용하여 자동적으로 제어기의 목적을 만족시키는 제어기 상수를 수치적으로 계산할 수는 없다. 다시 말해서, 제어기 상수를 우선 정해야 그 후 해당 제어기 상수에 대한 성능을 reachable set를 포함하는 타원형 함수를 통하여 예측할 수 있다. 즉, 주어진 제어기 상수에 대

해서 이차안정성에 대한 해석을 가능하게 하는 결과이다.

위의 결과는 제어시스템의 해석뿐만 아니라 제어기를 설계하는데도 응용할 수가 있다. (9)에서 보는 바와 같이 오차방정식은 선형시스템에 p_i 와 w_i 가 외란(disturbance)으로 작용하는 시스템으로 가정할 수 있다. 다음 설계방법은 아래와 같이 요약될 수 있다.

- i) (9)에 있는 A_i 행렬은 오차방정식에 중요한 영향을 미치는 행렬(system matrix of error dynamics)로써 A_i 행렬에 원하는 고유치(eigenvalue)와 고유벡터(eigenvector)를 부여할 수 있는 제어기 상수를 고르면 된다.
- ii) 다음은 p_i 와 w_i 에 의해서 얼마나 오차가 더 발생하는가를 예측하는 것이다. 이 예측은 위에서 유도한 결과를 이용하여 이차함수인 타원함수를 이용하여 계산한다. 이를 위해서는 먼저 p_i 와 w_i 의 구속조건이 필요하다. p_i 의 구속조건은 II장의 조건(i)을 이용하여 p_i 의 linear upper bound인 정리 1의 C_{zi} 와 D_{wi} 를 구한다. w_i 의 구속조건은 II장의 조건(ii)을 이용하여 구할 수 있다.
- iii) 마지막으로 정리 1의 결과를 이용하여 (11)과 같은 타원함수를 수치적으로 구할 수 있다. 이 타원함수의 크기는 비선형 함수와 연결 함수가 존재 시 DDSC가 얼마나 강인하게 성능을 유지할 수 있는지를 나타내는 지표라 생각하는 것이 직관적인 해석이라고 할 수 있다.

IV. 예제

본 절에서는 [5,6]에서 소개된 연결식 도립진자(coupled inverted double pendulum)을 이용하여 지금까지 위에서 제안된 DDSC의 설계방법을 예제를 이용하여 설명하고 다른 제어기 법과 결과를 비교함으로써 검증을 하고자 한다.

1. 모델링

그림 2에서 보는 바와 같이 두 개의 연결식 도립진자에 대한 모델링은 뉴턴의 제 2법칙을 이용으로 하여 4차 미분방정식으로 다음과 같이 유도될 수 있다[5,6]

$$\ddot{\theta}_1 = \frac{g}{l} \sin \theta_1 + u_1 + \frac{ka^2}{m_1 l^2} (\sin \theta_2 \cos \theta_2 - \sin \theta_1 \cos \theta_1)$$

$$\ddot{\theta}_2 = \frac{g}{l} \sin \theta_2 + u_2 + \frac{ka^2}{m_2 l^2} (\sin \theta_1 \cos \theta_1 - \sin \theta_2 \cos \theta_2)$$

위 식은 (1)에서 보이는 형태로 다음과 같이 쓸 수 있다.

$$\begin{aligned} \dot{x}_i &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i + \begin{bmatrix} 0 \\ \frac{g}{l} \sin x_{i1} - \frac{ka^2}{m_i l^2} \sin x_{i1} \cos x_{i1} \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ \frac{ka^2}{m_i l^2} \sin x_{j1} \cos x_{j1} \end{bmatrix}_{j \neq i} \\ &= U_i x_i + f_i + h_i \end{aligned}$$

여기서 $x_i = [\theta_i \quad \dot{\theta}_i]^T$ 이고 다음과 같이 II장의 가정 (ii)를 만족한다.

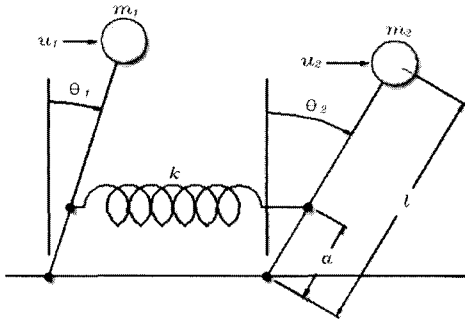


그림 3. 연결식 도립진자의 구성도.
Fig. 3. Schematic of interconnected inverted double pendulum.

$$h_i^T h_i = \left(\frac{ka^2}{m_i l^2}\right)^2 \sin^2 x_{j1} \cos^2 x_{j1} \leq \left(\frac{ka^2}{2m_i l^2}\right)^2 \equiv c \quad (13)$$

여기서 c 는 상수이다.

2. 오차방정식 유도

위에서 제시된 연결식 도립진자 문제의 제어목적은 $x_{i1} \rightarrow 0$ 이므로 $\dot{x}_{i1,d} = \ddot{x}_{i1,d} = 0$ 이다. 따라서, 첫 번째 오차표면은 $S_{i1} = x_{i1}$ 이며 III장 1절의 방법을 따라 다음과 같이 설계될 수 있다.

$$\begin{aligned} \dot{S}_{i1} &= \dot{x}_{i1} = x_{i2} \Rightarrow \bar{x}_{i2} = -\lambda_{i1} S_{i1} \\ \tau_{i2} \dot{x}_{i2,d} + x_{i2,d} &= \bar{x}_{i2} \\ \dot{S}_{i2} &= \dot{x}_{i2} - \dot{x}_{i2,d} \\ &= u_i + \frac{g}{l} \sin x_{i1} - \frac{ka^2}{m_i l^2} \sin x_{i1} \cos x_{i1} - \dot{x}_{i2,d} + \frac{ka^2}{m_i l^2} \sin x_{j1} \cos x_{j1} \\ \Rightarrow u_i &= -\frac{g}{l} \sin x_{j1} + \frac{ka^2}{m_i l^2} \sin x_{i1} \cos x_{j1} + \dot{x}_{i2,d} - \lambda_{i2} S_{i2} \end{aligned}$$

따라서 확장된 오차방정식은 다음과 같이 유도될 수 있다.

$$\begin{bmatrix} \dot{S}_{i1} \\ \dot{S}_{i2} \\ \dot{\xi}_{i2} \end{bmatrix} = \begin{bmatrix} -\lambda_{i1} & 1 & 1 \\ 0 & -\lambda_{i2} & 0 \\ -\lambda_{i1}^2 & \lambda_{i1} & \lambda_{i1} - 1/\tau_{i2} \end{bmatrix} \begin{bmatrix} S_{i1} \\ S_{i2} \\ \xi_{i2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} p_i + \begin{bmatrix} 0 \\ \frac{ka^2}{2m_i l^2} \\ 0 \end{bmatrix} w_i \quad (14)$$

여기서

$$\begin{aligned} p_i &= \frac{d}{dt} \left(\frac{g}{l} \sin x_{i1} - \frac{ka^2}{m_i l^2} \sin x_{i1} \cos x_{i1} \right) \\ w_i &= 2 \sin x_{j1} \cos x_{j1} = \sin 2x_{j1} \end{aligned}$$

따라서 $w_i^T w_i = \sin^2 2x_{j1} \leq 1$ 이므로 (13)에서의 상수 c 는 1로 가정할 수 있다.

시뮬레이션을 위하여 시스템 변수를 다음과 같이 가정해 보자.

$$a = \frac{g}{l} = 1, \quad b_i = \frac{ka^2}{m_i l^2} = 0.25$$

이 때 p_i 의 linear upper bound는 다음과 같이 표현될 수 있다.

$$\begin{aligned} |p_i| &= \left| \dot{x}_{i1} \left(\frac{g}{l} \cos x_{i1} - \frac{ka^2}{m_i l^2} (\cos^2 x_{i1} - \sin^2 x_{i1}) \right) \right| \\ &\leq \frac{5}{4} [1 - \lambda_{i1}] z_i = C_{zi} z_i \end{aligned}$$

3. Reachable set의 예측

초기조건이 $\theta_1(0) = 0.5, \theta_2(0) = -0.4, \dot{\theta}_i(0) = 0$ 으로 주어지고 DDSC 제어상수가 다음과 같이 주어졌을 때,

$$\{\lambda_{i1}, \lambda_{i2}, \tau_{i2}\} = \{1, 4, 0.02\}$$

그림 4는 θ_i 의 시간에 따른 변화를 보여주고 있다. 더욱이 [5]의 결과와의 비교를 나타내기 위해서 [5]에서 고려한 시스템 상수의 불확실성도 고려되었다. 그림에서 보는 바와 같이 두 제어기법 모두 두 개의 연결식 도립진자를 비슷한 성능을 가지고 안정화 시킴을 알 수 있다.

다음은 초기조건이 $\theta_1(0) = 1.0, \theta_2(0) = -0.8, \dot{\theta}_i(0) = 0$ 으로 주어지고 DDSC 제어상수가 다음과 같이 주어졌을 때,

$$\{\lambda_{i1}, \lambda_{i2}, \tau_{i2}\} = \{2, 8, 0.04\}$$

그림 5는 두 제어 기법을 이용했을 경우 θ_i 의 시간에 대한 변화를 보여주고 있다. DDSC의 방법이 [5]에서 보여준 결과보다 더 빨리 수렴함을 볼 수가 있다. 하지만 이런 비교보다는 그림 4와 그림 5에서 DDSC의 변화차이를 비교해보고 왜 이러한 결과가 나왔는지를 이해하는 것이 좀 더 의미가 있다고 생각한다. 다시 말해서, (14)으로 주어진 확장된 오차방정식에서 A_i 의 고유치를 살펴보면,

$$\begin{aligned} \text{when } \{\lambda_{i1}, \lambda_{i2}, \tau_{i2}\} &= \{1, 4, 0.02\}, \\ \text{eig}(A_i) &= \{-1.0208, -48.9792, -4.0\}, \\ \text{when } \{\lambda_{i1}, \lambda_{i2}, \tau_{i2}\} &= \{2, 8, 0.04\}, \\ \text{eig}(A_i) &= \{-2.1922, -22.8078, -8.0\}. \end{aligned}$$

모든 고유치가 s -평면에서 음수 실수 축에 위치하고 있음을 확인했다. 또한, 두 가지 다른 제어상수에 대한 고유치의 최대치를 비교해 보면 첫 번째 경우 고유치가 좀 더 $j\omega$ 축에 가까이 있음을 알 수 있다. 이러한 이유로 첫 번째의 경우 (그림 4)가 두 번째의 경우 (그림 5)보다 느린 반응을 보이는 주요인이 된다. 즉, A_i 행렬에 적당한 고유치를 할당함으로써 우리가 원하는 반응을 얻어낼 수 있다. 따라서, 이러한 사실을 이용하여 다른 비선형 제어방법에서 제시하는 설계방법에 비해 훨씬 해석이 쉽고 설계과정도 짧은 시간에 이루어질 수 있다.

마지막으로, $\{\lambda_{i1}, \lambda_{i2}, \tau_{i2}\} = \{2, 8, 0.04\}$ 으로 제어상수가 정해졌을 경우 reachable set을 수치적으로 구해서 제어기의 성능을 예측하고자 한다. 정리 1의 결과를 MATLAB기반 하에서 LMITOOL [15] 이라는 convex optimization 소프트웨어를 이용하여 (12)에서 보여준 행렬부등식을 풀면 reachable set의

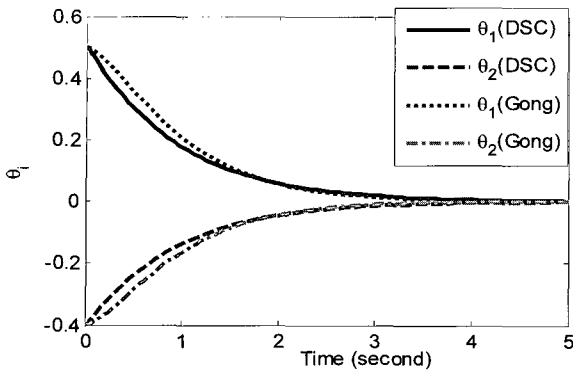


그림 4. $b_i = 0.25[1 + \sin(50t)]$ 와 같은 시스템 상수의 불확실성이 존재 시 두 제어기법에 대한 θ_i 의 변화 비교.
 Fig. 4. Time responses of θ_i for two controllers when the parametric uncertainty is given as $b_i=0.25 [1 + \sin(50t)]$.

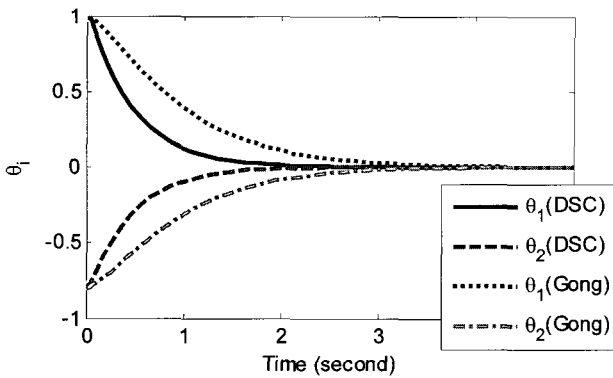


그림 5. 초기조건이 $\{\theta_1(0), \theta_2(0)\} = \{1.0, -0.8\}$ 인 경우 두 제어기법에 대한 θ_i 의 시간상의 변화 비교.
 Fig. 5. Time responses of θ_i for two controllers when the initial condition is given as $\{\theta_1(0), \theta_2(0)\} = \{1.0, -0.8\}$.

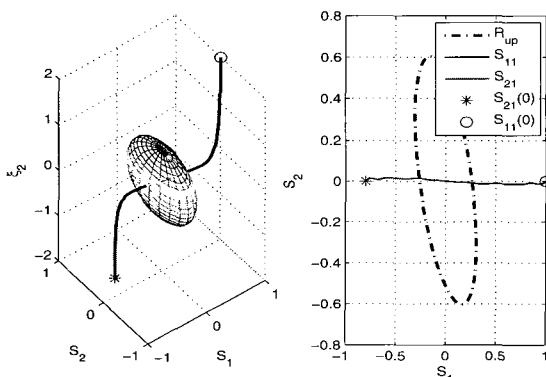


그림 6. Reachable set의 타원함수를 이용한 예측 및 오차의 궤적.
 Fig. 6. Estimation of an ellipsoidal function containing reachable sets and error trajectories.

타원함수를 이용한 bound는 다음과 같이 구해진다.

$$\mathcal{E} = \{z_i \mid z_i^T P z_i \leq 1\}$$

여기서

$$P = \begin{bmatrix} 14.61 & 3.91 & -2.13 \\ 3.91 & 3.79 & -1.35 \\ -2.13 & -1.35 & 2.73 \end{bmatrix}$$

그림 6은 위에서 구한 타원함수를 도식적으로 표현을 했으며 오차의 궤적이 일정시간 후에 이 타원으로 들어옴을 확인할 수 있다.

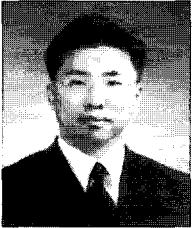
V. 결론

본 논문에서는 연결식 비선형 시스템에 대한 분산 동적 표면 제어의 설계 및 성능 예측에 대한 체계적 방법을 제안하였다. 다수의 부분시스템으로 이루어진 시스템에 대한 중앙식 동적 표면 제어방법의 한계점을 극복하고자 더 효율적인 해석 및 설계 방법을 개발하였다. 궁극적으로는 이미 문헌에서 소개되었던 다른 분산식 제어방법과 비교하여 응용 및 활용적 측면에서 훨씬 설계가 간단하고, 실시간 제어시스템에 적용하기 위한 첫 번째 선제 조건이라 할 수 있는 적은 계산량의 조건을 만족함을 보였다.

참고문헌

- [1] N. Sandell, P. Varaiya, M. Athans, and M. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 108-128, 1978.
- [2] D. D. Siljak, *Decentralized control of complex systems*, San Diego: Academic, 1991.
- [3] P. A. Ioannou, "Decentralized adaptive control of interconnected systems," *IEEE Transactions on Automatic Control*, vol. 31, no. 4, pp. 291-298, 1986.
- [4] D. T. Gavel and D. D. Siljak, "Decentralized adaptive control: structural conditions for stability," *IEEE Transactions on Automatic Control*, vol. 34, no. 4, pp. 413-426, 1989.
- [5] Z. Gong, C. Wen, and D. P. Mital, "Decentralized robust controller design for a class of interconnected uncertain systems: with unknown bound of uncertainty," *IEEE Transactions on Automatic Control*, vol. 41, no. 6, pp. 850-854, 1996.
- [6] D. Z. Zheng, "Decentralized output feedback stabilization of a class of nonlinear interconnected systems," *IEEE Transactions on Automatic Control*, vol. 34, no. 12, pp. 1297-1300, 1989.
- [7] S. Jain and F. Khorrami, "Decentralized adaptive output feedback design for large-scale nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, 1997.
- [8] Z. P. Jiang, "Decentralized and adaptive nonlinear tracking of large-scale systems via output feedback," *IEEE Trans. on Automatic Control*, vol. 45, no. 11, pp. 2122-2128, 2000.
- [9] S. Jain and F. Khorrami, "Decentralized adaptive control for a class of large-scale interconnected nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 42, no. 2, pp. 136-154, 1997.
- [10] C. Wen and Y. C. Soh, "Decentralized adaptive control using integrator backstepping," *Automatica*, vol. 33, no. 9, pp. 1719-1724, 1997.
- [11] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, "Dynamic surface control for a class of nonlinear systems"

- IEEE Trans. on Automatic Control*, vol. 45, p. 1893-1899, 2000.
- [12] J. Green and J. K. Hedrick, "Nonlinear speed control of automotive engines," in *Proceedings of American Control Conference*, pp. 2891-2897, 1990.
- [13] B. Song, J. K. Hedrick, "Design and experimental implementation of longitudinal control for automated transit buses," in *Proceedings of American Control Conference*, pp. 2751-2756, Boston, MA, 2004.
- [14] B. Song, J. K. Hedrick, and A. Howell, "Robust stabilization and ultimate boundedness of dynamic surface control system via convex optimization," *International Journal of Control*, vol. 75, no. 12, pp. 870-881, 2002.
- [15] M. Kricic, I. Kanellakopoulos, P. Kokotovic, *Nonlinear and adaptive control design*, Wiley, 1995.
- [16] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, SIAM, 1994.
- [17] L. El Ghaoui, J. Commeau, F. Delebeque, and R. Nikoukhah, "LMITool 2.2," available in <http://robotics.eecs.berkeley.edu/~elghaoui/#software>



송 봉 섭

1997년 한양대학교 정밀기계공학과 졸업. 1999년 U.C. Berkeley 기계공학과 석사 및 2002년 동 대학원 박사. 2002년~2003년 U.C. Berkeley California PATH Assistant Research Engineer. 2004년~현재 아주대학교 기계공학부 조교수. 관심분

야는 비선형 제어, 분산 제어, 센서네트워크, 최적화, 지능형 자동차, 무인 시스템.