

Multi-agent system의 Consensus에 대해 알아보기

심형보 (서울대학교 전기정보공학부)

각각의 동적 시스템이 네트워크로 연결되어 서로의 상태변수를 주고 받으면 개별 시스템이 할 수 없었던 새로운 일을 처리할 수 있게 된다. 본 편에서는 다개체 시스템(multi-agent system)에 관한 연구 결과 중 가장 기초적인 consensus 문제에 대해 알아보고, 이로부터 파생된 간단한 분산최적화 문제를 풀어보자.

1. 그래프 이론 기초부터 시작하자

기술적인 이야기로 서론을 시작하는 것이 조금 이상하지만 우선 그래프 이론으로부터 시작하고자 한다. 그림 1과 같이 표현된 대상을 그래프(graph)라 하는데, 여기서 원으로 표현된 것을 꼭지점(vertex) 혹은 노드(node), 그리고 화살표로 표현된 것을 변(edge)이라고 한다. 각각의 변은 가중치(weight)를 가질 수 있으며, 그림 1에서 화살표 옆에 표기된 수는 각 변의 가중치이다. 그래프를 표현하는 방법 중 그림을 그리는 것은 쉬운 방법처럼 보이나, 노드나 변이 매우 많아지면 딱히 그림이 쉽다고 말하기도 어려울 것이다. 그래서 그래프를 그림이 아닌 수식으로 표현하기 위해 그래프 G 를 $G = (V, E, A)$ 라고 표현하곤 하는데, 여기서 $V = \{1, 2, 3, 4\}$ 는 vertex의 집합이고 $E = \{(2, 1), (3, 1), (1, 3), (4, 3)\}$ 는 변의 집합이다. (그림 1과 비교하여 화살표의 방향을 어떻게 표기했나 살펴보자.¹) 이제 각 변의 가중치를 A 를 통해 표현하고 싶은데 $\{(2, 1: 3), (3, 1: 2), (1, 3: 1), (4, 3: 1)\}$ 과 같이 할 수도 있겠으나, 이런 경우 편리하게 사용할 수 있는 방법이 바로 행렬 표현법이다. 그래서 A 를 행렬

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

로 정의하는데, 잘 보면 행렬 A 의 (i, j) 원소 a_{ij} 의 값이 바로 변 (i, j) 의 가중치임을 알 수 있다. 이 행렬 A 를 인접행렬(adjacency matrix)이라고 부른다. 그러므로, 일반적으로 N 개의 노드를 가지는 그래프 G 에 있어, $V = \{1, \dots, N\}$ 이며 크기가 $N \times N$ 인 인접행렬 A 의 대각원소 a_{ii} 는 모두 0이며, 그 외의 원소 $a_{ij}(i \neq j)$ 는 0보다 같거나 크다는 것을 알 수 있다.

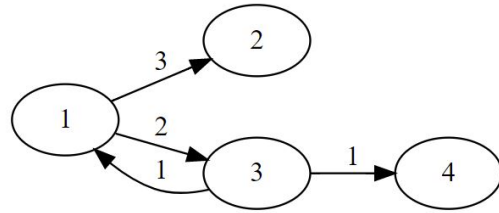


Fig. 1. 화살표는 변(edge)을 의미하고 방향이 있기 때문에 이러한 그래프를 directed graph, 줄여서 digraph라고들 한다.

그런데 인접행렬보다 더 자주 쓰이는 행렬은 (graph) Laplacian이라 부르는 행렬인데

$$L := D - A$$

로 정의되며, 여기서 D 는 대각행렬이며 i 번째 대각원소는 $\sum_{j=1}^N a_{ij}$ 로 정의된다. 즉, 인접행렬의 i 번째 행의 모든 원소의 합이다. 결국 L 의 (i, j) 번째 원소 l_{ij} 은 $i \neq j$ 인 경우 $l_{ij} = -a_{ij}$ 이고, $l_{ii} = \sum_{j=1}^N a_{ij}$ 이다. 따라서, 어떤 그래프에 대해 인접행렬과 Laplacian 행렬이 가진 정보는 동일하고, 그래프가 주어지면 Laplacian 행렬을 얻을 수 있다.

Laplacian 행렬은 앞으로 중요한 역할을 할 것이기 때문에 몇 가지 성질을 살펴보자. 우선 Laplacian 행렬은 정의에 따라 언제나 고유치(eigenvalue) 0을 가지며, 이 고유치 0에 해당하는 고유벡터 중 하나는 적어도 $1_N := [1, 1, 1, \dots, 1]^T \in \mathbb{R}^N$ 을 갖게 된다. (1_N 이란 기호는 모든 원소가 1인 크기 N 짜리 열벡터를 말한다.) 왜냐하면 L 의 각 행을 모두 더하면 항상 0이 될 수 밖에 없기 때문에 $L1_N = 0 \cdot 1_N$ 이 성립하기 때문이다. 한편, Laplacian 행렬의 0 아닌 고유치는 모두 실수부가 양이라는 사실이 알려져 있다. 왜 그럴까? 이를 보이기 위해 Gershgorin Theorem을 소개한다.

Gershgorin Circle Theorem: 주어진 임의의 행렬 $A \in \mathbb{R}^{n \times n}$ 가 있다고 하자. 복소평면 상에서 행렬 A 의 i 번째 대

¹화살표의 방향이 의미하는 내용과 E 에서 원소 표기 순서 등은 여러 논문에서 통일되어 있지 않으므로 매번 주의해야 한다.

각원소 a_{ii} 를 중심으로 하고 반지름이 $\sum_{j=1, j \neq i}^n |a_{ij}|$ 인 원반(disk)을 D_i 라고 정의하면, 행렬 A 의 고유치 λ 는 모두 원반 D_i 의 합집합에 속한다, 즉

$$\lambda(A) \in \cup_{i \in \{1, \dots, n\}} D_i$$

를 만족한다. □

증명: 이 정리의 증명은 어렵지 않아 함께 살펴보자. 행렬 A 의 어떤 고유치 λ 와 고유벡터 x 는 $Ax = \lambda x$ 를 만족한다. 이 때, 고유벡터 x 의 j 번째 (복소)원소 x_j 의 크기 (magnitude)가 가장 큰 원소를 x_{j^*} 라 하자, 즉,

$$|x_{j^*}| = \max_{j \in \{1, \dots, n\}} |x_j|.$$

이때 고유벡터는 영벡터가 아니므로 $|x_{j^*}| > 0$ 일 수 밖에 없다. 이제 열벡터 $Ax = \lambda x$ 에서 j^* 번째 원소만 따로 떼어 적어보면 $\sum_{j=1}^n a_{j^*j} x_j = \lambda x_{j^*}$ 이므로,

$$\lambda - a_{j^*j^*} = \sum_{j=1, j \neq j^*}^n a_{j^*j} \frac{x_j}{x_{j^*}}$$

가 유도되고, 이로부터 양변의 크기를 구하면

$$|\lambda - a_{j^*j^*}| \leq \sum_{j=1, j \neq j^*}^n |a_{j^*j}| \frac{|x_j|}{|x_{j^*}|} \leq \sum_{j=1, j \neq j^*}^n |a_{j^*j}|$$

가 된다. 주어진 λ 에 대해 항상 어떤 j^* 가 존재하므로 증명이 완성된다. □

이제 Gershgorin 정리를 Laplacian 행렬 L 에 적용하면, L 의 대각원소의 정의로부터 L 에 대한 원반 D_i 는 모두 원점을 지나고 복소평면의 우반평면에 위치하므로, 원점을 제외한 모든 고유치는 양의 실수부를 가질 수 밖에 없다.

한편, 그림 1에 나오는 모든 변이 양방향이고 가중치가 모두 1이라면, 양방향 화살표 대신 그림 2와 같이 간단히 표시하게 된다.

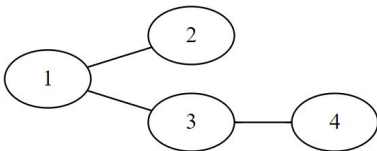


Fig. 2. 이 그래프의 직선은 양방향 화살표를 의미하는데, 변(edge)의 방향성이 없기 때문에 undirected graph라 불린다. 가중치는 양방향성이 동일하며, 이 그림처럼 생각할 경우 가중치가 모두 1이라 생각하면 된다.

Undirected graph를 나타내는 인접행렬과 Laplacian 행렬은 모두 대칭행렬이란 특징이 있고, 따라서 Laplacian 행렬 L 의 고유치는 모두 실수이고, 0보다 크거나 같다는 점을 기억하자.

한편, 그림 2와 그림 3의 차이는 여러가지가 있겠지만, 우선 그림 2의 그래프는 모든 노드가 다 연결되어 있는데 반해 그림 3의 그래프는 연결되어 있지 않다. 그렇다면 주어진 그래프가 연결되어 있는지 아닌지를 알기 위해서는 항상 이렇게 그림을 그려 판단해야 하는가?

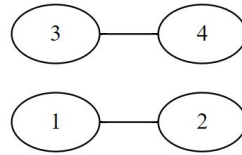


Fig. 3. 이 그래프의 모든 노드는 연결되어 있지 않고 두 개의 그룹으로 나누어져 있다.

대칭인 Laplacian 행렬의 고유치는 모두 실수이므로 순서대로 배열하여 $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$ 이라고 약속하자. 앞선 논의에서 Laplacian 행렬은 항상 0을 고유치로 갖기 때문에 λ_1 은 언제나 0일 수 밖에 없다. 즉,

$$0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N \tag{1}$$

이 성립한다. 이 때, 그래프의 연결 여부는 그림을 그리지 않아도 Laplacian 행렬의 second smallest eigenvalue² λ_2 가 0인지 아닌지만 확인하여도 알 수 있다. 이 신기한 결과를 정리한 뒤 증명해 보자.

정리: Undirected graph G 가 연결된 그래프일 필요충분 조건은

$$\lambda_2(L) > 0$$

이다. □

정리의 증명: 주어진 그래프의 Laplacian 행렬 L 뒤에 임의의 벡터 $x \in \mathbb{R}^N$ 를 곱한 Lx 의 i 번째 원소를 살펴보면, $a_{ii} = 0$ 이므로

$$(Lx)_i = l_{ii}x_i + \sum_{j=1, j \neq i}^N l_{ij}x_j = \sum_{j=1}^N a_{ij}x_i - \sum_{j=1}^N a_{ij}x_j \tag{2}$$

²The smallest eigenvalue는 당연히 λ_1 이며 그 값은 0이다. 한편 λ_2 를 해당 그래프의 algebraic connectivity라고 부른다. 그 이유는 해당 그래프에서 변의 갯수가 늘어나거나 가중치가 커져 그래프의 연결성이 강화될 수록 λ_2 가 커지는 경향이 있기 때문이다 [1].

이 성립함을 알 수 있다. 따라서

$$\begin{aligned}
 x^T Lx &= \sum_{i=1}^N x_i(Lx)_i = \sum_{i=1}^N \sum_{j=1}^N a_{ij}x_i(x_i - x_j) \\
 &= \left(\frac{1}{2} + \frac{1}{2}\right) \sum_{i,j=1}^N a_{ij}x_i^2 - \sum_{i,j=1}^N a_{ij}x_i x_j \\
 &= \frac{1}{2} \sum_{i,j=1}^N a_{ij}x_i^2 + \frac{1}{2} \sum_{i,j=1}^N a_{ij}x_j^2 - \sum_{i,j=1}^N a_{ij}x_i x_j \\
 &= \frac{1}{2} \sum_{i,j=1}^N a_{ij}(x_i - x_j)^2 = \frac{1}{2} \sum_{(i,j) \in E} a_{ij}(x_i - x_j)^2
 \end{aligned} \tag{3}$$

임을 알 수 있는데, 여기서 네번째 등식은 $a_{ij} = a_{ji}$ 임을 사용했고, 마지막 등식은 $(i, j) \notin E$ 인 경우 $a_{ij} = 0$ 임을 사용했다.

이 때 $X := \{x : x^T Lx = 0\}$ 이란 집합은 L 의 고유치 0에 해당하는 고유벡터가 생성하는 공간 $\{x : Lx = 0 \cdot x\}$ 과 동일함에 주목하자.³ 한편, 선형대수학에 따르면 대칭행렬의 고유치는 algebraic multiplicity와 geometric multiplicity⁴가 동일함이 알려져 있다. 그러므로, 벡터공간 X 의 차수(dimension)가 1이라면 λ_2 는 양수여야만 하고, X 의 차수가 2 이상이라면 λ_2 는 0이어야만 한다.

만일 해당 그래프가 연결되어 있다면 식 (3)의 마지막 항 $\sum_{(i,j) \in E} a_{ij}(x_i - x_j)^2 = 0$ 에 등장하는 변은 모든 노드를 연결할 것이기 때문에, 모든 x_i 가 같은 값을 가져야만 한다. 이는 L 의 고유치 0에 해당하는 고유벡터 x 는 0 아닌 상수 α 에 대해 $x = \alpha 1_N$ 를 만족한다는 뜻이다. 이 경우 이 고유벡터가 생성하는 벡터공간의 차수는 1이며, 따라서 $\lambda_2 > 0$ 이다. 반대로 $\lambda_2 > 0$ 이라면, L 의 고유치 0에 해당하는 고유벡터가 생성하는 공간의 차수는 1일 수밖에 없고, 이 경우에는 $\sum_{(i,j) \in E} a_{ij}(x_i - x_j)^2 = 0$ 을 만족하는 x 는 $\alpha 1_N$ 형태일 수 밖에 없다는 뜻인데, 이는 변들의 집합 E 가 모든 노드를 연결한다는 뜻이다. 그렇지 않다면 일부 x_i 들은 α 값을 가지고 다른 x_i 들은 $\beta (\neq \alpha)$ 를 가지면서 $x^T Lx = 0$ 을 만들 수 있기 때문이다 (예: $(x_1 - x_2)^2 + (x_3 - x_4)^2 = 0$). □

2. MULTI-AGENT SYSTEM은 무엇인가?

이제 각 노드가 동역학을 가진다고 생각해 보자. 예를 들어 노드 i 는

$$\dot{x}_i = u_i, \quad x_i(0) = \xi_i, \quad i \in \{1, \dots, N\} =: \mathcal{N} \tag{4}$$

³Laplacian 행렬 L 은 그 정의에 따라 언제나 positive semi-definite이므로 $L = N^T N$ 이라 쓸 수 있는 행렬 N 이 항상 존재하고, 따라서 $x^T Lx = \|Nx\|^2 = 0 \Leftrightarrow Nx = 0$, 즉 $Lx = 0$ 을 의미한다. 선형대수학 시간에 배우는 내용이다.

⁴고유치의 algebraic multiplicity는 고유치가 중첩된 횟수이고, geometric multiplicity는 해당하는 고유벡터가 생성하는 공간의 차수이다. 이와 관련된 선형대수 및 선형시스템에 대한 기초 지식은 [5]를 통해서도 얻을 수 있다.

와 같이 초깃값 ξ_i 를 가진 적분기라고 생각해 보자. 여기서 u_i 는 개별 노드의 입력인데, 개별 노드는 자신과 연결된 다른 노드 $j(j \neq i)$ 의 정보 x_j 를 가져와서 자신의 정보 x_i 와 차이를 구한 뒤, 정보를 가져온 변의 가중치를 곱해 모두 더하는

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i) \tag{5}$$

로 주어진다고 생각해 보자. 여기서, \mathcal{N}_i 는 노드 i 와 연결된 노드의 번호를 모아놓은 집합인데, 그림 2의 경우에 $\mathcal{N}_1 = \{2, 3\}$, $\mathcal{N}_4 = \{3\}$ 이다.

이와 같이 개별 동역학을 가지는 노드들이 그림 4처럼 서로 네트워크로 연결된 시스템을 multi-agent system이라 한다. 또한, 식 (5)와 같이 개별 노드의 입력이 다른 노드 정보와 자신의 정보의 차이 값에 대한 선형식으로 구성되는 경우 “diffusively coupled” multi-agent system이라 한다.

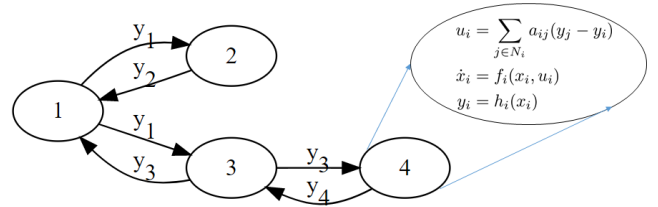


Fig. 4. 개별 노드가 동역학을 가지고 변을 따라 정보를 주고 받는 다개체 시스템의 예

3. MULTI-AGENT SYSTEM의 CONSENSUS는 무엇인가?

이제 식 (4)와 식 (5)로 연결된 전체 시스템의 동작을 살펴보고자 한다. 이를 위해 모든 시스템을 표현해 보려 하는데, $\dot{x}_i = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i)$ 를 모든 $i \in \mathcal{N}$ 에 대해 간단히 표현할 수 있는 방법이 있다. 식 (5)에서 $\sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i) = \sum_{j=1}^N a_{ij}(x_j - x_i)$ 이란 사실을 확인한다면, 식 (2)를 참고하여 전체 시스템을

$$\dot{x} = -Lx, \quad x(0) = \xi \tag{6}$$

로 쓸 수 있다는 것을 알게 된다.

시스템 (6)의 해는 어떻게 될까? 선형대수학이나 선형 시스템 수업을 들었다면, $x(t) = \exp(-Lt)x(0)$ 이 된다는 사실을 알고 있을 것이다. 또한 행렬 L 이 대칭행렬이란 사실로부터 L 의 고유치 $\lambda_i (i = 1, \dots, N)$ 에 대한 고유벡터 v_i 는 모두 실수벡터이고 서로 수직함을 알고 있다. 이

중 λ_1 과 v_1 은 이미 알고 있는 바와 같이 $\lambda_1 = 0, v_1 = 1_N$ 이다. 그러므로 $V := [v_1, v_2, \dots, v_N] \in \mathbb{R}^{N \times N}$ 이라 할때, 고유치 λ_i 를 대각원소로 갖는 대각행렬 $\Lambda \in \mathbb{R}^{N \times N}$ 에 대해 $LV = V\Lambda$ 가 성립함을 알 수 있다. 따라서 $\exp(-Lt) = V \exp(-\Lambda t) V^{-1}$ 이므로,

$$x(t) = v_1 e^{-\lambda_1 t} (V^{-1} x(0))_1 + v_2 e^{-\lambda_2 t} (V^{-1} x(0))_2 + \dots + v_N e^{-\lambda_N t} (V^{-1} x(0))_N$$

가 됨을 알 수 있다. 만약, 해당 그래프가 연결되어 있어 $\lambda_2 > 0$ 이라면, $2 \leq i \leq N$ 인 모든 λ_i 는 양수이므로,

$$\lim_{t \rightarrow \infty} x(t) = (V^{-1} x(0))_1 1_N$$

임을 알 수 있다. 그런데, $V^{-1}V = I$ 이어야 하므로, V^{-1} 의 첫번째 행 $(V^{-1})_1$ 은 $(1/N)1_N^T$ 여야만 한다. 그렇다면,

$$\lim_{t \rightarrow \infty} x(t) = \frac{1}{N} 1_N^T x(0) 1_N$$

이고 이것의 의미는, 모든 i 에 대해

$$\lim_{t \rightarrow \infty} x_i(t) = \frac{1}{N} \sum_{i=1}^N x_i(0) = \text{average of } \{x_i(0) : i \in \mathcal{N}\}$$

가 된다. 즉, 개별 노드들의 상태변수는 모두 전체 노드들의 초깃값의 ‘평균’으로 수렴하는 것이다!

개별 노드의 상태변수를 $x_i(t)$ 라 할때, 시간이 지날수록 모든 $x_i(t)$ 가 같은 값으로 수렴할때, 우리는 모든 노드 상태변수가 ‘consensus를 이루었다’라고 말한다. 위의 예는 consensus가 이루어질 뿐 아니라, 그들의 값이 전체 노드의 초깃값의 평균으로 수렴하는 예이다.

방금 살펴본 이 성질을 이용하면 매우 많은 노드들과 복잡한 변으로 이루어진 그래프에서 개별 노드가 가진 어떤 값의 평균을 중앙집중연산장치(centralized computation unit)없이도 평균값을 “분산적으로(distributedly)” 구할 수 있음에 주목하자. 중앙집중적으로 평균을 구하려면 모든 노드의 정보를 한 곳으로 모아 처리해야 하는 반면, 위에서 제시된 분산 알고리즘은 개별 노드가 자기 자신 주변에 놓인 노드들과만 지역적 통신(local communication)을 통해 전체 네트워크의 평균을 구할 수 있음을 보여준다. 이러한 성질은 많은 수의 드론의 편대비행(formation flight)에 쓰일 수 있고, 어떤 최적화 문제를 풀어야 하는 경우에 여러 노드의 네트워크 상에서 개별 노드가 자기 자신에게만 주어진 문제를 풀면서 주변의 노드와 상호 교류하기만 해도 전체 최적화 문제를 풀게 되는 경우(distributed optimization)에 사용될 수 있다. 다음 절에서 분산최적화에 대한 간단한 예제를 살펴보는

것으로 본 편을 마무리하고자 한다.

4. 분산최적화 맛보기

Strictly convex인 quadratic 비용함수 $J(x)$ 가 주어지고 이를 최소화하는 값 $x^* \in \mathbb{R}$ 를 찾고자 하는 최적화 문제

$$\min_x J(x) \tag{7}$$

가 있다고 하자. 이를 점근적으로 찾기 위한 gradient descent 방법은

$$\dot{x} = -\frac{\partial J}{\partial x}(x) \tag{8}$$

라는 동역학을 임의의 초깃값으로부터 푸는 것인데, 그렇게 하면 $x(t)$ 가 x^* 로 수렴한다는 사실이 알려져 있다.

그런데 만약 비용함수 J 가 여러 개의 quadratic 함수 J_i 의 합으로 구성되어 $J(x) = \sum_{i=1}^N J_i(x)$ 로 주어지는 반면, 개별 함수 J_i 는 어떤 네트워크에 참여하는 i 번째 사람만 알고 있는데 이 개별함수 J_i 의 정보를 다른 사람과 공유하고 싶어하지 않는다고 하자. 앞선 (8)의 방법은 어떤 중앙집중연산장치가 J_i 의 정보를 모아 풀 수 있는 방법이기 때문에 (8)은 더 이상 유효한 방법이 아니다. 과연 J_i 정보를 다른 이에게 알리지 않고도 전체 비용함수 J 를 최소화하는 x 를 구할 수 있을 것인가?

한가지 아이디어는 (7)로 주어진 최적화 문제를

$$\begin{aligned} \min_{x_1, x_2, \dots, x_N} & J_1(x_1) + J_2(x_2) + \dots + J_N(x_N) \\ \text{subject to} & x_1 = x_2 = \dots = x_N \end{aligned}$$

로 바꾸어 생각하는 것이다. 이 경우에도 동일한 해 x^* 를 찾을 수 있다는 점을 인지한다면, 각각의 개인(이제부터 각 개인을 노드라 하자)이

$$\begin{aligned} \min_{x_i} & J_i(x_i) \\ \text{subject to} & x_i = x_j, j \in \mathcal{N}_i \end{aligned}$$

라는 부속문제를 풀고, 이들 노드의 네트워크를 나타내는 그래프가 undirected이고 연결되어 있다면, 결국 같은 최적해를 얻게 될 것이다. 이때 노드 i 는 자신과 연결된 주변 노드 하고만 정보 x_i 를 주고 받으며 J_i 에 대한 정보를 공유할 필요가 없다. 이 아이디어를 구체화하여, 개별 노드는 임의의 초깃값 $x_i(0)$ 을 갖는

$$\dot{x}_i = -\frac{\partial J_i}{\partial x_i}(x_i) + k \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i) \tag{9}$$

라는 동역학을 푼다고 하자. 여기서 k 는 diffusive coupling항에 인가하는 이득으로 충분히 큰 양의 값으로 정

할 것이다. 이 식은 개별 비용함수 J_i 를 최소화하기 위한 부분과 x_i 들이 consensus를 이루도록 하는 두 개의 항으로 이루어졌음을 음미해 보자.

이제 (9)로 주어진 알고리즘을 해석하기 위해

$$f(x) = \begin{bmatrix} -\frac{\partial J_1}{\partial x_1}(x_1) \\ \vdots \\ -\frac{\partial J_N}{\partial x_N}(x_N) \end{bmatrix}$$

라고 하면, 전체 시스템을

$$\dot{x} = f(x) - kLx$$

와 같이 쓸 수 있다. 그러나 이 자체로는 이 시스템의 거동이 쉽게 해석되지 않으므로 좌표변환을 통해 좀더 쉽게 이해할 수 있도록 해 보자. 이를 위해 전술한 행렬 V 를 살펴보면

$$V = \begin{bmatrix} \mathbf{1}_N & R \end{bmatrix}$$

와 같이 쓸 수 있는데 $R = [v_2, \dots, v_N] \in \mathbb{R}^{N \times (N-1)}$ 이다. 또한, V^{-1} 는

$$V^{-1} = \begin{bmatrix} (1/N)\mathbf{1}_N^T \\ Q \end{bmatrix}$$

로 쓸 수 있고, Q 는 $(N-1) \times N$ 인 어떤 행렬이다. 이제 $z = V^{-1}x$ 인 좌표변환을 생각하면,

$$z = \begin{bmatrix} z_1 \\ z_r \end{bmatrix} = \begin{bmatrix} (1/N)\sum_{i=1}^N x_i \\ Qx \end{bmatrix} = V^{-1}x \quad (10)$$

가 되고 (단, $z_r \in \mathbb{R}^{N-1}$),

$$x = Vz = \mathbf{1}_N z_1 + Rz_r, \quad \text{즉, } x_i = z_1 + (R)_{i,r} z_r, \quad i \in \mathcal{N}$$

이라 할 수 있다. 한편, 그래프가 연결되어 있다고 가정하면

$$V^{-1}LV = \begin{bmatrix} 0 & 0 \\ 0 & \Lambda_{sub} \end{bmatrix}$$

인 행렬 $\Lambda_{sub} = \text{diag}(\lambda_2, \dots, \lambda_N) \in \mathbb{R}^{(N-1) \times (N-1)}$ 가 positive definite임을 알 수 있다. 이제 식 (10)을 통해 좌표변환을 하면 ($\mathbf{1}_N^T L = 0$ 이므로)

$$\dot{z}_1 = -\frac{1}{N} \sum_{i=1}^N \frac{\partial J_i}{\partial x_i}(z_1 + (R)_{i,r} z_r) \quad (11a)$$

$$\dot{z}_r = Qf(\mathbf{1}_N z_1 + Rz_r) - k\Lambda_{sub} z_r \quad (11b)$$

을 얻는다. 여기서 J_i 는 quadratic 함수이므로 $J_i(x_i) =$

$(1/2)q_i x_i^2 + p_i x_i + r_i$ 와 같이 주어질 것이고, 이로부터 식 (11)은 적절한 행렬 $f_{11} \in \mathbb{R}^{1 \times 1}$, $f_{12} \in \mathbb{R}^{1 \times (N-1)}$, $f_{21} \in \mathbb{R}^{(N-1) \times 1}$, $f_{22} \in \mathbb{R}^{(N-1) \times (N-1)}$, $g_1 \in \mathbb{R}^1$, $g_2 \in \mathbb{R}^{N-1}$ 을 이용하여

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_r \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} - k\Lambda_{sub} \end{bmatrix} \begin{bmatrix} z_1 \\ z_r \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \quad (12)$$

와 같이 쓸 수 있다. 이상의 유도로부터 $f_{11} = -(1/N)(q_1 + \dots + q_N) = -(1/N)(\partial^2 J)/(\partial x^2)$ 임을 알 수 있는데 J 가 strictly convex한 quadratic 함수이기에 $f_{11} < 0$ 가 성립한다. 따라서 충분히 큰 $k > 0$ 에 대해 행렬

$$\begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} - k\Lambda_{sub} \end{bmatrix}$$

은 Hurwitz하게 되고⁵ 시스템 (11)은 안정한 선형 시스템이다.

이제 z_r -동역학만 따로 살펴보면

$$\dot{z}_r = (f_{22} - k\Lambda_{sub})z_r + (f_{21}z_1 + g_2)$$

인데 전체 시스템이 안정하여 $z_1(t)$ 는 유계이므로, 두 번째 괄호의 항도 유계임을 알 수 있다. 그러므로 k 를 충분히 키운다면, $(f_{22} - k\Lambda_{sub})$ 은 안정한 행렬이 될 뿐만 아니라, $\|z_r(t)\|$ 도 시간이 지남에 따라 임의로 작게 만들 수 있음을 알 수 있다 [7, 4.8절 Ultimate Boundedness 참조]. 이제 $\|z_r(t)\|$ 이 충분히 작게 되면 식 (11a)에 의해 $z_1(t)$ 의 행동은

$$\dot{z}_1 = -\frac{1}{N} \sum_{i=1}^N \frac{\partial J_i}{\partial x_i}(z_1) = -\frac{1}{N} \frac{\partial J}{\partial x}(z_1)$$

과 유사하게 되어, $(1/N)$ 의 비례상수를 제외한다면 $z_1(t)$ 는 (8)의 해 $x(t)$ 와 유사하게 행동하여 x^* 에 근사적으로 수렴함을 알 수 있다. 이로써 개별 노드들은 x^* 의 정확한 값을 얻지는 못했지만 충분히 큰 k 에 대해 충분히 유효한 x^* 를 근사적으로 얻을 수 있다. (이러한 논의를 좀더 엄밀하게 분석하는 방법은 특이섭동이론(singular perturbation theory)을 사용하는 것이다. 이 이론에 대한 소개는 [6, 7]을 참조할 수 있다.)

5. 맺는 말

식 (4), (5)로 주어진 다개체 시스템과 식 (9)로 주어진 시스템의 가장 큰 차이는 식 (4), (5)의 경우에는 자체 동역학이 동일한 homogeneous 다개체 시스템인 반면, 식

⁵각자 증명해 보자! 한가지 방법은 Lyapunov 함수 $V(z) = z_1^2 + z_r^T z_r$ 를 g_1 과 g_2 가 없는 시스템 (12)에 따라 미분을 구하고 Young's inequality를 사용하여 cross term을 $|z_1|^2$ 과 $\|z_r\|^2$ 의 항으로 분해한 뒤, 큰 k 값으로 양의 $\|z_r\|^2$ 항의 계수를 음수로 만드는 것이다.

(9)의 경우는 각자의 자체 동역학이 서로 다른 heterogeneous 다개체 시스템이라는 것이다. 특별히 (9)에서 나타난 heterogeneity는 서로 다른 비용함수 J_i 때문에 등장한 피할 수 없는 heterogeneity이며, 사실 이러한 ‘다름’으로 인해 전체 문제 (8)을 개별 문제 (9)로 쪼개어 푸는 것 같은 효과가 나타났음을 상기해 보자. 이로써 각각의 동적 시스템이 네트워크로 연결되어 서로의 상태변수를 주고 받으면 개별 시스템이 할 수 없었던 새로운 일을 처리할 수 있음을 살펴본 셈이다.

참고

다개체 시스템 분야를 앞장서 연구하시고 필자를 이 분야로 이끌어 주신 서울대학교 서진현 명예교수님께 감사의 마음을 가지고 본 편을 작성하였다. 전반부 내용은 Bullo 교수의 책 [1]과 논문 [2]의 내용을, 후반부 내용은 논문 [3, 4]의 내용 중 일부를 간략화한 것이다. 본 편의 \LaTeX 조판본은 제어이론연구회 홈페이지에서 구할 수 있다. 한편, 그래프 그림을 그리는 도구로 Graphviz가 유용하다는 점과, <http://www.webgraphviz.com>에서 온라인으로 그림을 그릴 수 있다는 것을 알리고자 한다.

REFERENCES

- [1] F. Bullo, *Lectures on Network Systems*, Kindle Direct Publishing, available at <http://motion.me.ucsb.edu/book-1ns>, 2019.

- [2] J.H. Seo, H. Shim, and J. Back, “Consensus of high-order linear systems using dynamic output feedback compensator: Low gain approach,” *Automatica*, vol. 45, no. 11, pp. 2656–2664, 2009
- [3] J. Kim, J. Yang, H. Shim, J.S. Kim, and J.H. Seo, “Robustness of synchronization of heterogeneous agents by strong coupling and a large number of agents,” *IEEE Trans. on Automatic Control*, vol. 61, no. 10, pp. 3096–3102, 2016
- [4] J.G. Lee and H. Shim, “A tool for analysis and synthesis of heterogeneous multi-agent systems under rank-deficient coupling,” a draft archived at <https://arxiv.org/abs/1804.00638>, 2018
- [5] 심형보, 선형대수와 선형시스템 기초 (동영상 강의), <http://lecture.cds1.kr>
- [6] 심형보, Singular Perturbation (동영상 강의), <http://lecture.cds1.kr>
- [7] H. Khalil, *Nonlinear Systems*, 3rd edition, Prentice Hall, 2002.



심형보 교수는 2000년 서울대학교 전기공학부에서 박사학위를 받고 미국 산타 바바라 소재 캘리포니아주립대학에서 박사후 과정을 수료 후 현재 서울대학교에서 교수로 재직 중이다. *Automatica*, *IEEE Transactions on Automatic Control* 등 저널의 associate editor를 맡은 바 있다.