論 文
57-9-23

동적 센서네트워크에서의 유동적 경계선 추종 제어

Dynamic Boundary Tracking Control in Active Sensor Network

張 世 龍* · 李 起 龍* · 宋 奉 燮** · 左 東 京† · 洪 錫 敎**
(Seyong Jang · Giroung Lee · Bongsob Song · Dongkyoung Chwa · Sukkyo Hong)

Abstract - In this paper, the motion coordination algorithm of mobile agents in active sensor network is proposed to track the dynamic boundary for environmental monitoring. While most of dynamic boundary tracking algorithms in the literature were studied under the assumption that the boundary and/or its evolving rate is known a priori, the proposed algorithm is assumed that the individual active agent can measure the state of environment locally without any information of the boundary. When the boundary is evolving dynamically, the formation of active agents is designed to achieve two objectives. One is to track boundary layer based on the measured information and a small deviation. The other is to maintain a uniform distance between adjacent agents. The algorithm structure based on a state diagram is proposed to achieve these two objectives. Finally, it will be shown in the simulations that all given agents converge to a desired boundary layer and maintain a formation along the boundary. (e.g., a circle, an ellipse, a triangle and a rectangle)

Key Words : Dynamic boundary tracking, Active sensor network

1. 서 론

최근 산불과 해양 기름 유출 사고 등의 환경 피해 사건으로 인해 환경을 감시하고 이를 추적하는 방법에 대한 관심은 급속하게 증가하게 되었다. 환경을 감시하는 몇 가지 방법 중 동적 센서 네트워크를 이용한 방법의 발전 동기는 다른 방법을 이용한 것보다 더욱 효과적인 정보를 제공할 수 있다는 점에서 찾을 수 있다. 예를 들어 인공위성을 이용한 단일 센서를 이용한 방법은 해상도가 낮고 공전주기로 인해 실시간 감시에 제약을 받고, 고정형 센서 네트워크는 많은 수의 센서가 필요하고 센서가 소실될 위험이 있다[1].

한편 동적 센서 네트워크는 효과적인 정보 획득을 위해 사용할 수 있지만 유동적으로 움직이는 센서들의 정보를 효과적으로 취합하고, 경계를 추종하기 위해 이동로봇의 효과적인 이동 방법(motion coordination strategy)이 반드시 필요하다[2]. 이로 인해 측정값으로부터 경계를 추정하거나, 목표로 하는 경계로 수렴하여 추종하는 이동로봇의 이동 방법에 대한 다양한 연구들이 대체로 두 그룹으로 나뉘어 진행되고 있다[2-5]. 첫째로 측정값의 정보를 사전에 알고 이를 이용하는 그룹이다. Marthaler와 Bertozzi는 측정값의 포텐셜 함수를 최소화하는 방법으로 정적 스칼라 필드에서의 레벨 커브(level curve)를 찾고 이를 추종하는 snake algorithm을 사용하였다[3]. [4]에서는 측정값의 경사 함수(gradient function)와 음함수(implicit function)를 이용하여 이를 수행하고 있다. 두 번째 그룹은 경계의 수학적 모델을 로봇의 이동 방법에 포함시키는 방법을 사용한다. [2]에서는 경계의 확장 비율이 주어진 상황에서 이를 로봇의 이동 방법에 수식화하여 사용하고 있고, [5]에서는 보간법에 기초로 한 곡률 전개 알고리즘(curvature deployment algorithm)을 사용하고 있다.

이러한 연구들은 대부분 측정값의 정보 혹은 경계의 확장 정보, 곡률, 경계의 초기 값 등이 초기 조건으로 주어져 있다고 가정하고 있다. 이러한 가정은 이동로봇과 더불어 경계를 재분석 할 수 있는 운영 서버가 존재하거나, 앞에서 말한 단일 센서와 연계할 때 가능한 것들이다. 이 경우 운영 서버는 경계를 재분석하기 위해 매 주기마다 모든 고정형 센서 네트워크에서 들어오는 통신 데이터를 받아서 경계를 재분석해야 한다. 이 때 센서의 수가 많을수록 운영 서버가 일정 주기마다 받는 통신 데이터는 많아지고, 운영 서버와 거리상 멀리 떨어져 있는 고정형 센서의 정보는 다른 센서를 거쳐 오기 때문에 그만큼 시간이 지연된다. 또한 이렇게 받은 통신 데이터를 이용하여 재분석한 경계 정보는 다시 이동로봇에 전송해야 한다. 즉 운영 서버는 매 주기마다 고정형 센서 네트워크에서 들어오는 많은 통신 데이터와 이를 토대로 경계를 재분석하기 위한 복잡한 계산식을 필요로 하고 있다. 따라서 초기 조건이나 서버, 고가의 단일 센서 등의 가정을 순환할 수 있는 알고리즘이 필요하다. 이 논문에서는 경계의 초기정보나 수학적 모델을 알아야 하는 기존 연구들의 가정을 완화하고 경계나 이동 전략 생성을 위한 관리자급의 운영 서버를 사용하지 않는 간단한 전략을 사용하며, 고정형 센서 네트워크 및 운영 서버를 이용하지 않고 각각의 이동형 센서(이동로봇)만을 이용하여 경계를 추종하도록 함으로

* 學生會員 : 亞洲大學 電子工學部 碩士課程
** 正 會 員 : 亞洲大學 電子工學部 敎授 · 工博
† 교신저자, 正會員 : 亞洲大 電子工學部 助敎授 · 工博
E-mail : dkchwa@ajou.ac.kr
接受日字 : 2007年 12月 24日
最終完了 : 2008年 8月 1日

써 상대적으로 복잡도가 완화되도록 유동적 경계선 추종을 위한 동적 센서 네트워크에서의 이동로봇의 이동 전략을 제시한다. 이동로봇은 경계에 대한 정보를 초기 조건으로 가지고 있지 않고 서버로부터 제공받지도 않기 때문에 상태 변수를 스스로 판단하는 분산형 시스템의 형태로 설계하였다. 게다가 이렇게 분산된 시스템이 협력적으로 경계로 수렴하는 목표를 달성하기 위하여 이동로봇이 각각이 가질 수 있는 상태 변수와 상태 변화 조건, 그리고 상태별 이동 전략을 제시하고 시뮬레이션을 통해 검증하였다

2. 문제 설정

동적 센서 네트워크에서의 유동적 경계선 추종을 위해 N개의 이동로봇으로 구성된 네트워크 그룹의 이동 방법은 다음과 같은 두 가지 목표를 달성하여야 한다.

1) 설정된 목표값으로 구성된 경계선에서 작은 크기의 벗어남을 허용하는 지역을 경계구역(boundary layer)으로 설정하고 이 안에서 모든 이동로봇이 위치한다.
2) 경계를 효율적으로 추정하기 위하여 이동로봇간의 분포를 균일하게 한다.

위의 목표가 달성되면 경계 위에 위치한 이동로봇들의 좌표를 바탕으로 경계선을 근사할 수 있다. 이 때 경계 구역 위에 있는 이동로봇의 좌표를 경계를 구성하는 꼭지점으로 생각하여 이를 연결하는 다각형(polygon)을 만들면 간단한 형식의 근사화 방법이 된다[6]. 더불어 제안하는 알고리즘을 성립시키고 기술하기 위해 다음과 같이 가정한다.

1) 경계의 모양은 산불 상황을 가정하여 원형 혹은 타원형으로 가정하고, 다른 볼록 집합 형태(삼각형, 사각형)에 대하여 고려한다.
2) 경계는 이동로봇이 충분히 반응할 수 있는 정도의 느린 변화의 확장률을 가진다.
3) 이동로봇의 제어를 위한 기구학 및 동역학은 고려하지 않는다.
4) 경계의 크기 변화에 따른 잉여로봇의 처리는 고려하지 않으므로 잉여 로봇은 발생하지 않는다.
5) 이동로봇은 초기 상태에서 각각의 ID를 가지고 있으며, ID의 순서는 흐트러지지 않는다.

참고로 위 가정에 대해 간략히 언급을 하면 다음과 같다. 가정 1)은 기존의 다른 연구에서도 도입한 가정이며, 가정 2)가 성립할 경우에만 유동성 경계선 추종이 가능하므로 이 가정은 타당하다. 가정 3)을 바탕으로 한 경계선 추종을 위한 알고리즘은 기존에 많이 연구되어 온 기구학 및 동역학을 고려한 이동로봇의 제어와 결합하여 확장할 수 있으며, 가정 4)는 일정한 개수의 이동로봇 경계 추종만을 다루고 있음을 의미한다. 가정 5)는 실제 구현상의 문제를 고려한 것으로 이 논문에서는 알고리즘 측면에 한정하여 다루도록 한다.

3. 유동적 경계선 추종 제어 알고리즘 구조

이 장에서는 먼저 상태 변수 판별에 중요한 역할을 하는 측정값의 상태 변수와 플래그(flag)에 대하여 기술한다. 또한 경계의 추정을 위한 균일 분포된 이동로봇에 대해 정의하고 이를 바탕으로 이동로봇의 상태 변수에 대해 정의한다. 마지막으로 상태 변수들의 변화를 기술하고 상태 변화의 조건과 상태 변수의 판별 과정에 대하여 기술한다. 이를 통해 이 논문에서 제안하는 이동로봇의 이동 전략의 구조를 파악할 수 있다.

3.1 상태 변수 정의

3.1.1 측정값의 상태 변수

측정값의 상태 변수는 플래그와 함께 이동로봇의 상태 변수를 결정하기 위한 한 요소이다. 그림 1과 같이 목표값(T_d)들의 모임을 나타내는 경계선을 $\partial\Omega$로 표기한다. 이 때 변위 δ를 갖는 지역, 즉 $T_d \pm \delta$의 지역을 '경계구역(Boundary Layer)'라고 가정하면 다음과 같은 측정값의 3가지 상태를 정의할 수 있다.

정의1. 측정값의 상태

- $\Omega = \{T \mid |T - T_d| < \delta,\ T_d > 0,\ \delta > 0\}$인 Ω에 대하여 측정값 $T_i \in \Omega$이면, i 번째 이동로봇의 측정값은 Within Boundary Layer(WBL) 상태 변수에 있다.
- $\Omega_{in} = \{T \mid T > T_d + \delta,\ T_d > 0,\ \delta > 0\}$인 Ω_{in}에 대하여 측정값 $T_i \in \Omega_{in}$이면, i 번째 이동로봇의 측정값은 Inner Boundary Layer(IBL) 상태 변수에 있다.
- $\Omega_{out} = \{T \mid T < T_d - \delta,\ T_d > 0,\ \delta > 0\}$인 Ω_{out}에 대하여 측정값 $T_i \in \Omega_{out}$이면, i 번째 이동로봇의 측정값은 Outer Boundary Layer(OBL) 상태 변수에 있다.

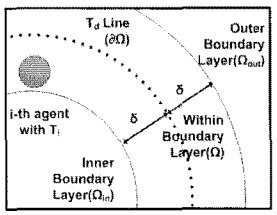


그림 1 측정값의 세 가지 상태 (WBL, IBL, OBL)
Fig. 1 The three state of a measurement (WBL, IBL, OBL)

3.1.2 플래그 설정

플래그는 이동로봇의 상태 변수를 판별하는 요소 중의 하나이다. 각각의 이동로봇은 매 샘플링 시간마다(sampling time) 자신의 플래그와 좌우 로봇의 플래그, 자신의 측정값을 기준으로 그 값을 결정한다. 그림 2에서 보는 것처럼 플래그는 0, 1, 2의 값을 가지며 각각의 플래그 변화는 다음과 같이 설정할 수 있다.

- 모든 이동로봇은 초기 조건으로 0의 플래그를 가진다. ($flag_{i,0}(0) = 0$)
- $flag_i(k-1) = 0$인 이동로봇의 측정값이 OBL 상태에 있고 $flag_{i\pm1}(k-1) = 0$이면 $flag_i(k) = 0$이다.
- $flag_i(k-1) = 0$인 이동로봇의 측정값이 OBL 상태에 있고 $flag_{i+1}(k-1) = 1$ 혹은 $flag_{i-1}(k-1) = 1$이면 $flag_i(k) = 2$이다. 이 때 i 번째 이동로봇은 $i+1$ 혹은 $i-1$ 번째 로봇의 adjacent라고 한다.
- $flag_i(k-1) = 0$인 이동로봇의 측정값이 WBL 상태에 있으면 $flag_i(k) = 1$이다.

- $flag_i(k-1) = 2$인 이동로봇의 측정값이 OBL 상태에 있으면 $flag_i(k) = 2$이다.
- $flag_i(k-1) = 2$인 이동로봇의 측정값이 WBL 상태에 있으면 $flag_i(k) = 1$이다.
- $flag_i(k-1) = 1$인 이동로봇은 그 이동로봇이 초기화 될 때까지 $flag_i(k) = 1$이다

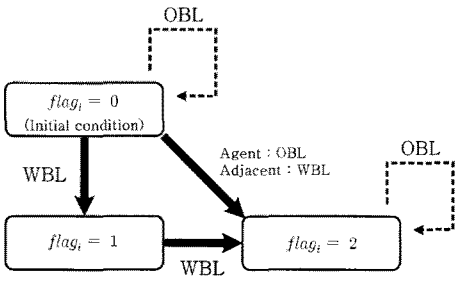


그림 2 플래그와 플래그 변화 조건

Fig. 2 Flags and their changing conditions

3.1.3 균일 분포된 (Uniformly Distributed) 이동로봇

각 이동로봇의 위치를 경계를 포함하는 다각형의 꼭지점으로 취급하여 경계의 모양과 위치를 추정해 낼 수 있다[2]. 이를 위해 다음과 같이 균일하게 분포된 이동로봇을 정의한다.

정의 2. 균일하게 분포된 이동로봇

모든 이동로봇이 WBL 상태 변수에 있을 때, i 번째 이동로봇은 $i+1$ 번째 이동로봇의 거리와 $i-1$ 번째 이동로봇의 거리를 계산하여 이 두 거리의 차이(ΔD_i)를 계산한다. 계산식은 다음과 같다.

$$\Delta D_i = \left| \|R_{i+1}(k) - R_i(k)\| - \|R_i(k) - R_{i-1}(k)\| \right|,$$

$$R_i(k) = [x(k)\ y(k)]^T$$

이 때 i 번째 이동로봇은 ΔD_i를 줄이기 위해 자신의 위치를 조정하게 된다. 즉 $i+1$ 번째 이동로봇과 $i-1$ 번째 이동로봇의 중간지점에 올 수 있도록 이동로봇의 속도를 조정하게 된다. 이러한 과정을 통해 $\max(|\Delta D_i|) \le \epsilon$ $(\epsilon > 0)$을 만족한다면 모든 이동로봇은 boundary layer에서 균일하게 분포되었다고 말할 수 있다.

3.1.4 이동로봇의 상태 변수

이동로봇의 초기 상태와 경계 추종, 로봇의 균일 분포 상태 등을 구별하기 위해 정의하고, 각 상태에 따라 이동로봇은 각기 다른 이동 전략을 사용함으로써 경계를 추종하는 목표를 달성하기 위해 위에서 정의한 측정값에 의한 3가지 상태를 바탕으로 각 이동로봇이 가지는 플래그($flag_i$)와 인접 로봇간의 거리의 차(ΔD_i)를 사용하여 다음과 같은 7개의 상태를 정의한다.

정의 3. 이동로봇의 상태 변수

- i 번째 이동로봇의 측정값이 OBL이고 $flag_i = 0$이며 $flag_{i\pm1} = 0$이면 이동로봇은 Free Formation(FF) 상태 변수에 있다.
- i 번째 이동로봇의 측정값이 OBL이고 $flag_i \neq 1$, $flag_{i+1} = 1$ 혹은 $flag_{i-1} = 1$이면 이동로봇은 Outer Boundary Layer Initially(OBLI) 상태 변수에 있다.
- i 번째 이동로봇의 측정값이 OBL이고 $flag_i = 1$이면 이동로봇은 Outer Boundary Layer for Climbing(OBLC) 상태 변수에 있다.
- i 번째 이동로봇의 측정값이 WBL이고 $|\Delta D_i| < \epsilon$이면 이동로봇은 Within Boundary Layer for Staying (WBLS) 상태 변수에 있다.
- i 번째 이동로봇의 측정값이 WBL이고 $|\Delta D_i| > \epsilon$이면 이동로봇은 Uniform Distribution(UD) 상태 변수에 있다.
- i 번째 이동로봇의 측정값이 IBL이면 이동로봇은 Inner Boundary Layer for Descent(IBLD) 상태 변수에 있다
- 모든 이동로봇이 균일하게 분포되어 있으면 모든 이동로봇은 Boundary Estimation(BE) 상태 변수에 있다.

3.2 이동 전략의 알고리즘 구조

3.2.1 알고리즘 상태도

이 논문에서 제시하는 각 이동로봇의 이동 전략은 그림 3과 같이 상태 변수 블록도(상태도) 구조로 나타낼 수 있다. 이동로봇들은 초기 조건으로 $flag_i = 0$과 FF 상태를 갖는다. 각각의 이동로봇들은 서로 다른 상태에 있을 수 있으며 조건을 만족할 때 다른 상태로 변화된다. 최종적으로 모든 이동로봇이 경계선으로 수렴하고, 균일하게 분포되어 경계선을 추종 및 추정하기까지 상태 변화는 계속되며 자신의 상태에 맞는 이동 전략을 선택하게 된다. 이 때 그림 3의 상태의 변화 조건 (1)~(10)은 다음과 같다

(1) WBL, $flag_i = 1$, $|\Delta D_i| > \epsilon$
(2) WBL, $flag_i = 1$, $|\Delta D_i| < \epsilon$
(3) $flag_i = 2$
(4) $|\Delta D_i| > \epsilon$
(5) $|\Delta D_i| < \epsilon$
(6) IBL
(7) WBL, $|\Delta D_i| > \epsilon$
(8) OBL
(9) WBL, $|\Delta D_i| < \epsilon$
(10) $\max(|\Delta D_i|) \le \epsilon$

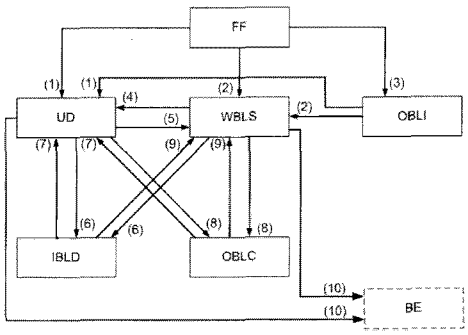


그림 3 이동 전략의 알고리즘 상태도

Fig. 3 The state diagram of the motion coordination algorithm structure

3.2.2 상태 변수 판별

그림 4는 그림 3에서 말한 상태 변화를 판별하기 위하여 각각의 이동로봇이 각 샘플링 시간에 진행하는 작업의 시작

과 끝을 순서도 형태로 나타낸 그림이다. 먼저 모든 이동로봇은 센서 네트워크를 통해 인접한 이동로봇의 위치와 플래그 정보를 획득한다. 다음으로 각 이동로봇은 측정값($T_i(k)$)과 통신을 통해 습득한 정보를 바탕으로 플래그를 결정하고 ($flag_i(k)$) 플래그와 측정값을 바탕으로 현재 상태를 결정하며, 현재 상태에 따라 이동 전략을 결정한다. 각각의 이동 전략은 이동로봇의 다음 위치를 산출해 낸다.

$R(k+1) = [x(k+1)\ \ y(k+1)]^T$

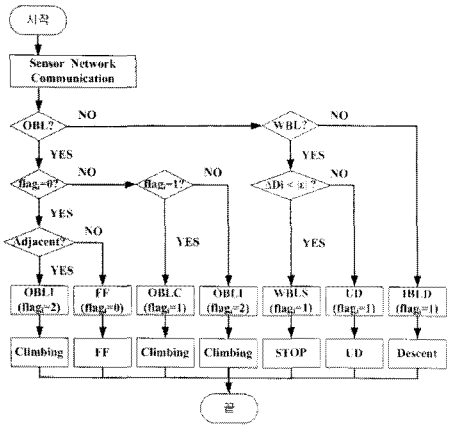


그림 4 상태 변수판별 순서도

Fig. 4 The flow chart of state decision

4. 알고리즘 상의 상태 변수별 이동전략

이 장에서는 각 상태 변수별 이동 전략에 관하여 기술한다. 각 이동로봇은 자신의 상태에 따라 이동 전략을 선택하며 서로 다른 전략을 사용할 수 있다. 이 논문에서 사용된 알고리즘의 배경은 N 개의 이동로봇이 대형을 유지하며 이동하다가 최초로 경계와 만나는 이동로봇으로부터 경계를 감싸안듯 이동하여, 모든 이동로봇이 경계로 수렴하는 것이다. 이것은 흡사 채찍으로 통나무를 휘감을 때 일어나는 현상과 비슷하다.

4.1. Free Formation(FF) 상태 변수

첫 번째는 FF 상태 변수이다. 모든 이동로봇이 초기에 경계로 이동할 때 0의 플래그를 가진 상태에서 이루어진다. FF 상태 변수에서는 간단한 'Leader Following Formation Control : $l-\phi$ controller' 기법을 사용한다[7]. 그림 5에서 보는 바와 같이 이 방법은 리더로 지정된 이동로봇과 거리(l) 및 각도(ϕ)를 유지하며 이동하여 전체 대형을 유지하는 방법이다. 초기 조건으로 리더의 ID(ID_l)가 부여되며, 리더는 관측하고자 하는 지역의 대략적인 자동 운행 경로를 가지고 있다고 가정한다. FF 상태 변수에서의 이동 전략을 수식으로 표현하면 다음과 같다.

$$R_i(k+1) = \begin{cases} G(k,\ R_i(k)) & \text{if } i = j \\ R_i(k) + l[\cos\phi_i\ \ \sin\phi_i]^T & \text{if } i \neq j \end{cases}$$

여기서 j는 초기 리더, $\phi_i = \begin{cases} \pi/2 & \text{if } i > j \\ -\pi/2 & \text{if } i < j \end{cases}$, l은 로봇간 거리, G는 초기 리더의 항법 함수이다. 이 때 ϕ와 l은 설계 변수로 이를 설정함에 따라 대형의 형태가 달라지며, 이 논문에서는 $\pm\pi/2$로 설정하여 수평 대형이 되도록 설정하였다.

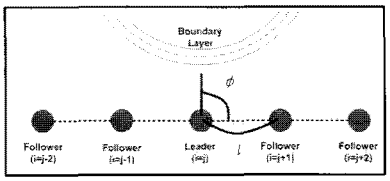


그림 5 리더 추종 대형제어 : $l-\phi$ 제어기

Fig. 5 Leader following formation control : $l-\phi$ controller

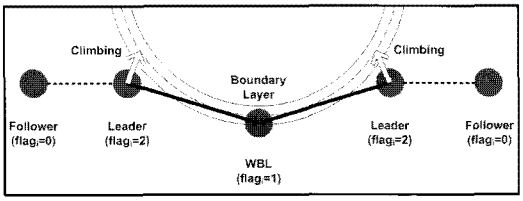


그림 6 FF 상태변수에서 WBLS 혹은 UD 상태변수로의 전이

Fig. 6 Transition from FF to WBLS or UD state

4.2 FF 상태변수에서 WBLS 혹은 UD 상태 변수로의 변화

FF 상태 변수에서 대형을 유지한 채 이동하던 이동로봇들 가운데 하나가 경계와 만나게 되면 각각의 이동로봇들은 서로 다른 상태에 놓이게 된다. 그림 6은 그림 5에서 j 번째 이동로봇이 경계와 만난 경우의 상태 변화를 나타낸 것이다. 3장에서 기술한 바대로 j 번째 이동로봇은 WBL이므로 플래그가 1로 세팅되고 WBLS 혹은 UD 상태 변수로 바뀐다. $j \pm 1$ 번째 이동로봇은 OBL이고, WBL의 이웃하고 있으므로 플래그가 2로 세팅되고, OBLI 상태 변수로 바뀐다. 이 때 $j \pm 1$ 번째 이동로봇은 $j \pm 2$ 번째 이동로봇의 FF 이동 전략을 위해 새로운 리더로 지목되고, 전체의 큰 대형은 2개의 작은 대형으로 나누어진다. WBLS와 UD 상태변수에서의 이동 전략은 다음 항에서 기술하며, OBLI에서의 이동 전략은 4.4에서 기술할 IBLD 및 OBLC 상태변수에서의 이동 전략을 따른다.

4.3 WBLS 및 UD 상태 변수

측정값이 WBL의 상태에 있는 이동로봇은 두 가지 이동 목적을 가진다. 첫째로, 경계선에 도달하였으므로 그 위치를 유지하는 것과 둘째로, 이웃 이동로봇간의 거리를 균일하게 만들기 위해서 더 멀리 있는 로봇의 방향으로 이동하는 것이다. 이 두 가지 목적을 만족시키기 위해 3.1.3에서 정의한 ΔD_i를 사용하여 다음과 같은 식을 작성하였다.

$$R_i(k+1) = R_i(k) + |\lambda_i| f(i,\ \lambda_i\ :\ k) V_R \text{ for } i = 1, 2, \cdots, N$$

여기서 $\lambda = \begin{cases} 1 & \text{if } \Delta D_i > \epsilon \\ -1 & \text{if } \Delta D_i < \epsilon \\ 0 & \text{if } |\Delta D_i| \leq \epsilon \end{cases}$,

$$f(i,\ \lambda_i) = \begin{bmatrix} \cos\left(\tan^{-1}\left(\dfrac{y_{i-\lambda_i} - y_i}{x_{i+\lambda_i} - x_i}\right) + \theta_i\right) \\ \sin\left(\tan^{-1}\left(\dfrac{y_{i+\lambda_i} - y_i}{x_{i+\lambda_i} - x_i}\right) + \theta_i\right) \end{bmatrix},$$

$\theta_i = -\lambda_i \dfrac{\pi}{2\delta}(T_i - (T_d - \delta))$이다.

$\Delta D_i > \epsilon$는 $\|R_{i+1}(k) - R_i(k)\| > \|R_i(k) - R_{i-1}(k)\| + \epsilon$와 같으므로 $i+1$ 번째 이동로봇은 $i-1$ 번째 이동로봇보다 i 번째 이동로봇에서 허용 범위(ϵ)이상으로 더 멀리 있다. 따라서 i 번째 이동로봇은 $i+1$ 번째 이동로봇을 향하여 이동하게 된다. 이러한 방법은 Yuan Cao와 Rafael Fierro에 의해 제안되었다[2]. 논문 [2]에서 그들은 이동로봇이 비전 센서를 이용하여 경계를 따라 주행하는 능력과, 경계를 따라 선 적분(line integral)에 의한 거리를 구할 수 있는 능력이 있기 때문에 더 멀리 있는 로봇과 자신과의 중간점을 향해 나아가는 알고리즘을 사용하고 있다. 그러나 이 논문에서는 경계의 정보를 획득하기 힘들기 때문에 선적분에 의한 거리가 아닌 유클리디안(Euclidean distance) 거리를 사용한다. 또한 이 논문에서 사용되는 이동로봇은 비전 센서를 사용하지 않는 일반적인 경우의 이동로봇이므로 이웃하는 이동로봇간의 거리를 좁히기 위해 경계선을 지나 Inner Boundary Layer를 통과하여 이동하는 경우도 발생한다(그림 7 참고). 산불과 같은 상황에 적용할 경우 이것은 매우 위험한 경우이기 때문에 $f(i, \lambda_i)$에 θ_i를 추가하여 이를 예방하였다. 그림 8은 θ_i의 효과를 나타내는 것이다. $|\Delta D_i| \le \epsilon$인 경우 WBLS, $\lambda_i = 0$이므로 i번째 이동로봇은 현재의 위치를 유지한다.

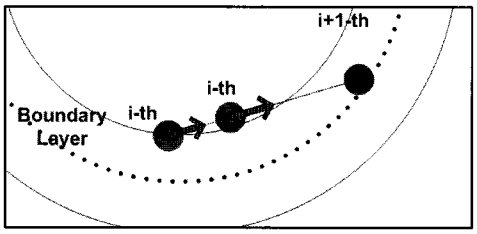


그림 7 θ_i를 고려하지 않은 경우의 이동로봇의 진행
Fig. 7 The moving direction of a agent without θ_i

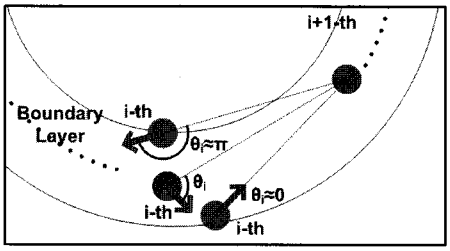


그림 8 θ_i를 고려한 경우의 이동로봇의 진행
Fig. 8 The moving direction of a agent with θ_i

4.4 IBLD 및 OBLC 상태 변수

경계의 확장으로 인해 이동로봇의 측정값은 WBL에서 IBL로 바뀌거나, 경계의 축소로 인해 WBL에서 OBL로 바뀔 수 있다. 또한 균일 분포를 유지하기위해 이동하는 중간에서도 이러한 측정값 변화는 일어날 수 있다. 게다가 앞에서 언급한 $j \pm 1$의 경우와 같이 FF 상태 변수에서 OBLI 상태 변수로 변화하는 경우도 있다. 측정값이 목표값을 초과하는 IBL과 목표값에 미치지 못하는 OBL에서의 이동 목적은 WBL로의 수렴으로 동일하다. 이 목적을 만족하기 위해 IBL(IBLD 상태 변수)에서는 하강하여 WBL로, OBL(OBLI와 OBLC 상태변수)에서는 등반하여 WBL로 수렴하는 전략을 사용한다.

등반과 하강의 방향은 기본적으로 경계의 중심 방향과 중심에서 먼 방향으로 말할 수 있다. 산불과 같은 경계의 중심 방향으로 갈수록 측정값이 커지는 경우 중심 방향이 등반이고, 중심에서 먼 방향이 하강 방향이다. 이 방향은 경계의 법선의 방향과 비슷한 방향이다. 경계위에 있는 두 이동로봇을 연결한 선을 경계의 접선으로 가정하면 다음 식과 같이 두 방향을 정의할 수 있다.

$$\vec{D}_{i-C} = \frac{1}{D_{i+\eta}} \begin{bmatrix} \cos\theta & -\eta\sin\theta \\ \eta\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_{i+\eta}(k) - x_i(k) \\ y_{i+\eta}(k) - y_i(k) \end{bmatrix}$$

$\Rightarrow$ 등반 (climbing)

$$\vec{D}_{i-D} = \frac{1}{D_{i+\eta}} \begin{bmatrix} \cos\theta & \eta\sin\theta \\ -\eta\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_{i+\eta}(k) - x_i(k) \\ y_{i+\eta}(k) - y_i(k) \end{bmatrix}$$

$\Rightarrow$ 하강 (descent)

여기서 $D_{i+\eta} = \|R_{i+\eta}(k) - R_i(k)\|$, η는 $flag_i = 1$ 일 때 $\eta = \begin{cases} -1 & \text{if } i = N \\ 1 & otherwise \end{cases}$, $flag_i = 2$ 일 때 $\eta = \begin{cases} -1 & \text{if } i = j+1 \\ 1 & \text{if } i = j-1 \end{cases}$, $\frac{1}{D_{i+\eta}} \begin{bmatrix} x_{i+\eta}(k) - x_i(k) \\ y_{i+\eta}(k) - y_i(k) \end{bmatrix}$은 법선방향으로의 단위 벡터이다. 한편 위 식에서 사용된 η는 플래그에 영향을 받게 된다. 이것은 경계의 축소나 균일 분포시에 발생하는 OBLC 상태(플래그 1)와 FF 상태 변수에서 변화 과정에서 발생하는 OBLI 상태(플래그 2)를 구분하기 위해서 필요하다. 플래그가 2인 OBLI 상태에서는 플래그가 1인 j 번째 이동로봇을 기준으로 법선 방향을 산출하기 때문이다.

위에서 정의한 두 방향을 이용하여 다음과 같은 다음 이동 목표위치 산출식을 작성하였다.

$$R_i(k+1) = R_i(k) + \gamma\vec{D}_{i-C}V_R + \mu\vec{D}_{i-D}V_R$$

여기서 $\gamma = \begin{cases} 0 & \text{if } IBL \\ 1 & \text{if } OBL \end{cases}$, $\mu = \begin{cases} 1 & \text{if } IBL \\ 0 & \text{if } OBL \end{cases}$, V_R은 이동로봇의 이동 속도이다.

IBLD 상태 변수에 있는 이동로봇은 $\gamma = 0$, $\mu = 1$이 되어, 하강 방향만이 적용되고, OBLC 혹은 OBLI 상태변수에 있는 이동로봇은 $\gamma = 1$, $\mu = 0$이 되어, 등반 방향만이 적용된다. 그림 9에서 등반($j-1$번째)과 하강($j+1$번째)의 두 방향을 알 수 있다.

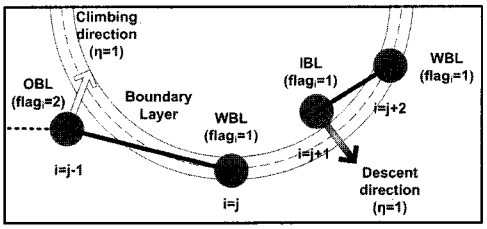


그림 9 등반, 하강의 두 방향
Fig. 9 Two directions : Climbing and Descent

4.5 Boundary Estimation(BE) 상태 변수

경계를 추정하는 방법은 다양하게 연구되고 있다. 일반적인 경우 경계 추정 오차를 최소화하는 비용 함수를 정의하여 사용하며, 특히 볼록 집합형태에서는 "best N-vertices

polytope"의 문제로 다루어진다[6]. 이렇게 결정된 경계 추정법에 따라 서버는 이동로봇을 배치하고, 이를 다시 이동로봇에 전송하여 이동에 사용하게 된다.

한편 이 논문에서 이동로봇은 분산형 시스템으로 서버로부터 정보를 받지 않고, 이웃하는 이동로봇과의 통신만을 이용하여 상태에 따른 이동 전략을 선택하여 이동한다. 이동로봇은 서버로 현재 위치와 상태만을 전송할 뿐이다. 따라서 경계를 추정하고 BE 상태를 판별하는 것은 관측용 서버에서 이루어진다. 관측용 서버는 경계를 추정하여 이동 전략에 적용, 이동로봇의 다음 위치를 이동로봇에 재전송하는 다른 연구에서의 집중형 서버가 아닌 이동로봇의 위치와 상태만을 파악하는 서버이다. 또한 경계를 추정하는 알고리즘은 이동로봇들의 좌표를 ID순으로 연결, 로봇들을 경계의 꼭지점으로 취급하고 이를 선으로 연결하여 만들어진 다각형을 경계로 사용하는 다각형(polygon) 방법을 사용한다. 이 때 경계의 추정을 효율적으로 하기 위하여 로봇간의 거리를 균일하게 분포시키는 알고리즘을 사용하였다.

5. 시뮬레이션 결과

제안한 알고리즘을 검증하기 위해서 MATLAB을 이용, 반지름 R(m)의 산불 상황에 대한 시뮬레이션을 실시하였다. EMBYR(Ecological Model for Burning the Yellowstone Region) 모델에 따르면 산불은 바람이 없는 균일연소체(fuel) 상황에서 원의 형태를 갖고, 바람이 불면 바람이 부는 방향으로 더 빨리 확장하는 타원의 형태를 갖는다[1]. 로봇들의 초기 위치는 불길(fire line)의 가운데에서 R+(m) 아래로 떨어진 곳에서 3(m) 간격으로 배치되어 있으며, 로봇의 이동 속도는 2(m/s), 샘플링 시간은 0.025초(sec)로 설정하였다. 설계변수는 $\phi = 90^\circ$, $\theta_i = 90^\circ$, $\epsilon = 0.1$(m)로 설정하였다. 측정값은 온도값으로 가정하였고, 온도는 열복사에 의해 불길(fire line)까지의 거리(D)에 제곱에 반비례 하는 것으로 가정하였다. 불길에서의 온도는 1000℃로, 불길에서 무한대의 거리에 있는 곳의 온도를 20℃로 가정하였다. 불길에서 약 6(m) 떨어진 곳의 온도를 50℃로 가정하고 $T_d = 50$, $\delta = 2$ 혹은 5℃로 설정하였다. 온도식은 $T_i = \frac{1000}{D^2} + 20$와 같다.

산불 상황을 가정하여 정적(혹은 시불변) 및 유동적(혹은 시변) 상황과 원형, 타원형의 경계를 조합 총 4가지 형태에 대하여 시뮬레이션을 실시하고 보다 다양한 응용 예를 위하여 블록 집합 형태의 대표적인 형태인 삼각형과 사각형의 경계에 대하여 시뮬레이션을 실시하였다. 각 경계의 형태에 따라 이동로봇의 수, 확장률을 달리해 시뮬레이션을 실시하고, 온도변화와 모든 이동로봇이 균일하게 분포될 때까지(BE상태)의 시간(수렴 시간)을 관찰하였다. 그림 10은 정적 원형 시뮬레이션의 경계 추종 과정을 순차적으로 보여주고 있다. 그림 10(a)는 초기의 FF 상태이고, 그림 10(b)는 FF 상태 변수에서 다른 상태 변수로 변화하는 것을, 그림 10(c)는 모든 이동로봇의 플래그가 1이 된 후 IBLD, OBLC 상태변수를 반복하고 있는 것을 보여 주고, 그림 10(d)는 BE 상태변수를 보여주고 있다.

시뮬레이션에 사용한 타원의 수식은 다음과 같다.

$$[x(t)\ \ y(t)] = p(t)\left[\cos\psi\ \ \frac{1}{2}\sin\psi\right] \in \Omega_{fire}$$

여기서 $p(0) = 10 = R$, $\dot{p}(t)$는 상수, Ω_{fire}는 불길(fire-line)로 타원의 경계선, $\psi \in [0,\ 2\pi)$이고, $R = 10$, $R += 10$이다.

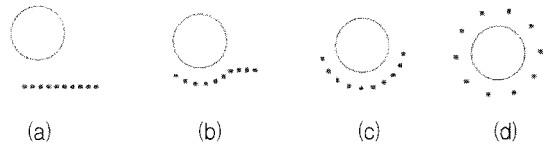


그림 10 경계 추종 과정 : 정적 원형

Fig. 10 Sequences of boundary tracking : static circle

정적 및 유동적 원형, 타원형 경계의 시뮬레이션 결과를 표 1~4에 정리하였다. 표 1과 표 2는 정적 원형 및 타원의 경계에 대하여 이동로봇의 수를 변화시키며 수렴 시간을 관측한 결과이다. 표 3과 표 4는 유동적 원형 및 타원의 경계에 대하여 이동로봇의 수 N을 10으로 고정하고, 경계의 확장률을 변화시키며 수렴 시간을 관측한 결과이다.

표 1은 정적 원형 경계의 결과를 보여주고 있다. 경계의 반지름이 10(m)로 작기 때문에 개체 수는 20대가 최고였으며 약 32초 후에 경계로 수렴하였다. 순차적으로 수렴하는 알고리즘의 특성상 많은 수의 이동로봇이 사용될 경우 수렴 시간이 길어졌다. δ의 크기와 수렴 시간은 직접적인 관계가 없었고, 경계지역을 이탈하지 않고 경계로 잘 수렴하는 결과를 보여 주었다.

표 1 제안된 알고리즘의 성능 : 정적 원

Table 1 Performance of the proposed method : static circles

N	R(m)		수렴 시간(sec)		최대 온도 이탈(℃)	
	$\delta = \pm 2$	$\delta = \pm 5$	$\delta = \pm 2$	$\delta = \pm 5$	$\delta = \pm 2$	$\delta = \pm 5$
4	10	10	18.95	19.23	1.996	4.988
8	10	10	20.95	21.23	1.996	4.988
12	10	10	23.10	23.33	1.996	4.988
16	10	10	26.35	26.55	1.996	4.988
20	10	10	32.38	32.25	2.072	5.075

표 2는 변화하지 않는 원형의 경계에 대하여 이동로봇의 개체수를 변화시키며, 시뮬레이션을 진행한 결과를 표로 나타낸 것이다. 원형과 마찬가지로 20대의 개체 수까지 N을 변화시켰으며 약 35초 후에 경계로 잘 수렴하는 결과를 보여 주고 있다.

표 2 제안된 알고리즘의 성능 : 정적 타원

Table 2 Performance of the proposed method : static ellipses

N	R(긴폭, 짧은폭) (m)		수렴 시간 (sec)		최대 온도 이탈 (℃)	
	$\delta = \pm 2$	$\delta = \pm 5$	$\delta = \pm 2$	$\delta = \pm 5$	$\delta = \pm 2$	$\delta = \pm 5$
4	10:5	10:5	19.38	19.70	1.936	4.956
8	10:5	10:5	19.85	20.12	1.936	4.956
12	10:5	10:5	22.38	22.52	1.936	4.956
16	10:5	10:5	26.78	26.83	1.936	4.956
20	10:5	10:5	35.03	37.73	2.232	5.198

표 3은 변화하는 원형의 경계에 대하여 이동로봇의 개체수 N을 10으로 고정하고, 반지름의 확장률을 변화시키며 시뮬레이션을 진행한 결과를 표로 나타낸 것이다. 반지름의 확

장률이 클수록 수렴시간이 길어졌고, 1(m/s) 이상의 확장률에서는 로봇의 이동속도가 확장률을 극복하지 못하여 정상적인 결과가 나오지 않았다. 0.8(m/s)에서는 $\delta \pm 2$의 조건에서 수렴하지 못하였는데 이는 θ_i의 변화량이 빠른 확장 속도로 인해 급격하게 변화하기 때문이다. 그러나 느리게 변화하는 경우나 경계 지역의 넓이를 크게 잡은 경우에는 경계로 잘 수렴하는 결과를 보여 주었다.

표 3 제안된 알고리즘의 성능 : 유동적 원

Table 3 Performance of the proposed method : dynamic circles

R의 확장률	R(시작:수렴후) (m)		수렴 시간 (sec)		최대 온도 이탈 (℃)	
	$\delta = \pm 2$	$\delta = \pm 5$	$\delta = \pm 2$	$\delta = \pm 5$	$\delta = \pm 2$	$\delta = \pm 5$
0.1	10:12.32	10:12.34	23.15	23.35	1.959	4.889
0.2	10:14.91	10:14.96	24.55	24.78	1.855	4.995
0.4	10:21.26	10:21.34	28.15	28.35	2.114	5.019
0.8	×	10:41.00	×	41.00	×	5.256

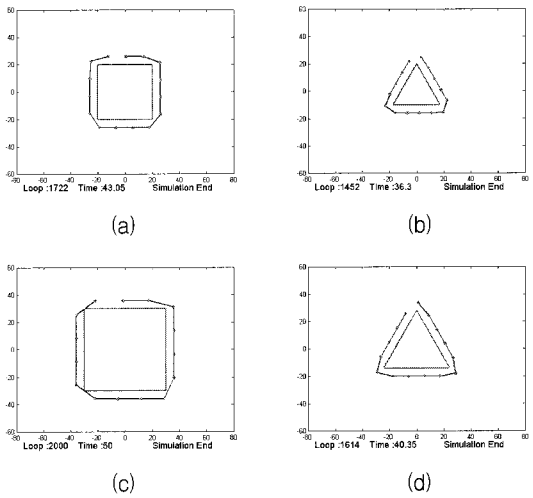
표 4는 변화하는 원형의 경계에 대하여 이동로봇의 개체수 N을 10으로 고정하고 반지름의 확장률을 변화시키며 시뮬레이션을 진행한 결과를 표로 나타낸 것이다. 반지름의 확장률이 클수록 수렴시간이 길어졌고 원형과는 달리 0.8(m/s) 이상의 확장률에서도 정상적인 결과를 보여주고 있다. 이것은 타원의 확장률을 긴 방향 반지름을 기준으로 하였기 때문에 짧은 방향의 확장률이 작아지는 결과에서 비롯된 결과로 분석된다.

표 4 제안된 알고리즘의 성능 : 유동적 타원

Table 4 Performance of the proposed method : dynamic ellipses

R(긴쪽)의 확장률	R(긴쪽-시작:수렴후) (m)		수렴 시간 (sec)		최대 온도 이탈 (℃)	
	$\delta = \pm 2$	$\delta = \pm 5$	$\delta = \pm 2$	$\delta = \pm 5$	$\delta = \pm 2$	$\delta = \pm 5$
0.1	10:12.16	10:12.19	21.60	21.19	1.938	4.990
0.2	10:14.52	10:14.57	22.58	22.85	1.848	4.924
0.4	10:19.95	10:20.03	24.88	25.08	2.021	4.996
0.8	10:35.62	10:35.78	32.03	32.23	2.114	5.131

그림 11은 원, 타원과 함께 볼록 집합 형태의 대표적인 형태인 삼각형과 사각형의 경계 추종 시뮬레이션 결과를 보여주고 있다. 응용 예로 사용한 산불 상황인 경우 원과 타원 형태 경계가 고려될 수 있으나 제안하는 방법의 보다 다양한 응용으로의 확장을 위해 실시한 실험으로 수렴 가능 여부를 확인한 결과를 도시하였다. 경계의 왜곡과 확장에 따른 로봇 이동능력을 고려하여 $N = 15$, $R = 20$(m), R+= 20 (m), R확장률 = 0.2(m/s)로 시뮬레이션을 실시한 결과이다. 유클리디안 거리와 다각형 추정법을 사용함에 따라 경계의 각 꼭지점에 해당하는 부분에서 경계의 왜곡이 발생하지만, 정적 경계 및 유동적 경계를 잘 추종하고 있다. 그림 11(a)의 정적 사각형의 경우 43.05초의 수렴 시간을 확인하였고, 그림 11(b)의 정적 삼각형은 36.3초, 그림 11(c)의 유동적 사각형은 50초, 그림 11(d)의 유동적 삼각형은 40.35초의 수렴 시간을 확인하였다.



(a) (b) (c) (d)

그림 11 경계 추종 성능 : (a) 정적 사각형, (b) 정적 삼각형, (c) 유동적 사각형, (d) 유동적 삼각형

Fig. 11 Boundary tracking : (a) static rectangle, (b) static triangle, (c) dynamic rectangle, (d) dynamic triangle

6. 결 론

이 논문에서는 동적 센서 네트워크에서의 유동적 경계선 추종 제어를 위한 분산형 이동로봇의 이동 전략을 제시하였다. 알고리즘은 크게 세 가지 부분으로 구성된다. 첫째는 경계로 접근할 때 사용하는 리더 추종 대형 제어, 둘째는 이웃한 이동로봇간의 유클리디안 거리를 균일하게 분포시키는 균일 분포 알고리즘, 셋째는 경계 구역을 추종하기 위해 등반과 하강을 반복하는 알고리즘이다. 제시한 알고리즘의 적용 가능성을 다양한 경계 형태와 정적, 동적 상황에 따라 시뮬레이션을 통해 확인하였다. 시뮬레이션 결과는 확장률, 경계의 모양, 이동로봇의 수의 변화에 대하여 수렴 시간과 온도 변화에 따라 비교 분석하였다. 결과적으로 제안한 알고리즘은 볼록 집합 형태의 대표적인 원과 타원, 삼각형, 사각형 형태의 경계에 대하여 잘 추종한다고 할 수 있다. 그러나 이 논문의 경계 추정에서 사용한 다각형 방법은 이동로봇의 수가 적을 때 경계의 왜곡이 심해질 수 있는 단점이 있다. 이 경우 [8]에서와 같이 타원 근사법(ellipsoidal approximation)과 같은 방법을 사용할 수도 있으며 이는 앞으로 해결해 나갈 문제점이다.

이 논문에서 제시한 알고리즘은 기존의 고정형 센서 네트워크 및 운영 서버를 이용할 경우 발생되는 복잡한 계산식이나 통신 지연을 유발하는 많은 통신 데이터를 필요로 하지 않는다. 또한 실시간으로 경계 지역으로 수렴하는 성능을 보여 주며 경계의 변화에 대하여서도 잘 적응하고 있다. 앞으로 좀 더 다양한 환경에 적용하기 위하여 다양한 형태의 경계에 대한 이동 전략 및 경계 추정 알고리즘 개선을 준비하고 있으며, 그 이후 좀 더 실제적인 응용을 위해 Non-convex 형태로의 경계 모양의 확장, 이동로봇의 동역학, 센서의 성능을 고려한 알고리즘의 개선과 로봇간 거리에 따른 이동로봇의 개체수 변화, 경계의 추정법 개선 등이 차후로 다루어질 것이다.

감사의 글

본 연구는 한국과학재단 특정기초연구 R01-2006-000-11373-0 지원으로 수행되었음.

참 고 문 헌

[1] D. W. Casbeer, R. W. Beard, T. W. McLain, and S. Li, "Forest fire monitoring with multiple small uavs," in Proc. American Control Conf., Portland, Oregon USA, pp. 3530-3535, June 8-10 2005.

[2] Yuan Cao and Rafael Fierro, ""Dynamic Boundary Tracking Using Dynamic Sensor Nets,"" in Proc. 44th IEEE Conf. on Decision and Control, Manchester Grand Hyatt Hotel, San Diego, CA, USA, December 13-15, pp.703-708, 2006.

[3] D. Marthaler and A. L. Bertozzi, "Tracking environmental level sets with autonomous vehicles," in Recent Developments in Cooperative Control and Optimization, S. Butenko, R. Murphey, and P. M. Pardalos, Eds. Kluwer Academic Publishers, 2003.

[4] L. Chaimowicz, N. Michael, and V. Kumar, "Controlling swarms of robots using interpolated implicit functions, " Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2498-2503, April 2005.

[5] S. Susca, S. Mart´ınez, and F. Bullo, "Monitoring environmental boundaries with a robotic sensor network, " in Proc. American Control Conf., Minneapolis, MN, pp. 2072-2077, June 14-16 2006.

[6] A. Ganguli, S. Susca, S. Mart´ınez, F. Bullo, and J. Cort´es, "On collective motion in sensor networks: Sample problems and distributed algorithms," in Proc. 44th IEEE Conf. on Decision and Control, and the European Control Conf., Seville, Spain, pp. 4239 - 244., December 12-15 2005.

[7] Hiromasa Takahashi, Hiroaki Nishi, "Autonomous Decentralized Control for Formation of Multiple Mobile Robots Considering Ability of Robot," in IEEE Transactions on Industrial Electronics., Vol. 51(6), pp.1272-1279, December. 2004.

[8] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, "Linear Matrix Inequalities in System and Control Theory", SIAM Studies in Applied Mathematics, 1994.

저 자 소 개

장 세 용 (張 世 龍)

1979년 5월 5일생. 2006년 아주대학교 전자공학과 졸업. 2007년 동 대학원 전자공학과 석사 졸업

Tel : +82-31-219-2489

E-mail : seyong@ajou.ac.kr

이 기 룡 (李 起 龍)

1979년 6월 9일생. 2006년 아주대학교 전자공학과 졸업. 2006년~현재 동 대학원 전자공학과 박사과정

Tel : +82-31-219-2489

E-mail : leegr007@ajou.ac.kr

송 봉 섭 (宋 奉 燮)

1996년 한양대 기계공학과 졸업. 1999년 U.C Berkeley 기계공학과 대학원 석사. 2002년 동 대학원 박사. 현재 아주대학교 기계공학부 부교수

Tel : +82-31-219-2339

E-mail : bsong@ajou.ac.kr

좌 동 경 (左 東 京)

1971년 12월 23일생. 1995년 서울대 제어계측공학과 졸업. 2001년 동 대학원 제어계측공학과 졸업(공박). 2005년~현재 아주대 전자공학부 조교수

Tel : +82-31-219-1815

E-mail : dkchwa@ajou.ac.kr

홍 석 교 (洪 錫 敎)

1948년 8월 23일생. 1971년 서울대 전기공학과 졸업. 1973년 동 대학원 석사. 1981년 동 대학원 박사. 1976년~현재 아주대학교 전자공학부 교수

Tel : +82-31-219-2478

E-mail : skhong@ajou.ac.kr