

제어안정도를 고려한 이동로봇 행동단위의 가변 주기 제어

김홍열\*, 김주민\*, 김대원\*, 양광웅\*  
 명지대학교 정보제어공학과\*, 한국생산기술연구원\*

Rate Modulation Strategy for Behaviors of a Mobile Robot with Considering Control Stability

Hongryeol Kim, Joomin Kim, Daewon Kim, Kwangwoong, Yang  
 Department of Information Control Engineering Myongji University\*, KITECH\*

**Abstract** - In this paper, behaviors performing controls of a mobile robots are designed based on the concept of closed-loop controls. A control architecture, in which the behaviors coordinate concurrently with one others to achieve a task is also proposed. In the control architecture, task manager allocates valid resources and configures the weight of the behavior outputs to achieve the task. For practical implementation, a rate modulation strategy based on RMS(Rate Modulation Strategy) is proposed for efficient resource allocations. The rate modulation strategy is performed with considering control stability. The efficiency of the proposed rate modulation strategy is validated through simulation tests.

1. 서 론

지능적인 이동로봇 제어의 실현을 위해 최근에 행동(behavior) 기반의 제어 구조에 대한 연구 및 적용이 활발히 진행되고 있다[1][2][3]. 행동 기반의 제어 구조는 지능을 연구하는 생체학에서 비롯되었으며, 인공 지능 분야의 연구 성과로부터도 많은 영향을 받았다.

대부분의 기존 행동 기반 이동로봇에 대한 연구는 자극(stimulus)이 주어졌을 때 단순히 반응(response)을 행하는 행동 단위를 기준으로 이루어졌지만, 최근의 생체학의 연구 결과에 의하면 실제 대부분의 지능적인 행동 단위는 반응의 결과로서 자극의 변화 혹은 환경의 변화가 피드백(feedback)되는 형태로 모델링(modelling)된다[4].

기존 행동 기반의 이동로봇에 관한 연구가 갖는 또 다른 문제점은 제어 구조의 설계에 있어 추상적인 개념의 제안에만 그치고 있어 이러한 제어 구조의 실제 구현 시에 고려되어야 할 유용 자원의 제한에 대한 연구가 미비한 실정이다.

이러한 유용 자원의 제한의 극복에 관한 연구는 제어기의 실시간 운영체제 분야에서 활발히 진행되고 있는데, 이러한 연구들의 대부분이 예측 불가능한 부하의 발생시에 시스템의 최종적인 실패 방지만을 목적으로 하였고 제한된 자원의 효율적인 이용에 관한 연구는 이루어지지 않았다[5][6]. 최근에는 유용 자원 제한의 극복에서 고려되어야 할 연구로서 스케줄링(scheduling)의 변화가 제어 안정도에 미치는 영향에 관한 연구가 수행된 바 있다[7].

본 논문에서는 이동로봇의 제어를 수행하는 행동 단위를 폐루프(closed-loop) 개념을 이용하여 설계하고, 이러한 행동 단위 들의 동시적인 협조 수행을 통해 하나의 작업 단위인 태스크(task)를 수행하는 제어 구조를 제안한다. 또한 제안된 제어 구조에서 태스크의 성공적인 완료를 위해 유용 자원의 효과적인 분배 및 행동 단위 출

력의 가중치를 수행해주는 태스크 관리자의 개념을 제안한다. 본 논문에서 제안된 제어 구조의 구현을 위해 실시간 운영체제를 갖는 이동로봇 제어기 환경에서의 중앙처리장치 점유 스케줄링 전략으로서, RMS(Rate Monotonic Scheduling)를 기반으로 한 가변 주기 전략을 제안한다. 또한 이러한 가변 주기 전략을 수행하기 위해 RMS의 이용을 제한과 행동 단위의 제어 안정도를 판별하여 주기 가변 범위를 결정하는 방식을 제안하며, 모의실험을 통해 제안된 방식의 타당성 및 효율성을 입증한다.

2. 본 론

2.1 태스크 수행 제어 구조

태스크는 동시에 수행하는 행동 단위의 동작 협조에 의해 수행되는 작업 단위로서, 이동로봇의 입장에서 의미 있는 최소의 작업 단위를 의미한다.

하나의 태스크를 수행하는 제어 구조는 동시에 수행하며 폐루프 구조를 갖는 다수의 관측형 행동 단위와 주위 환경의 변화에 따라 이러한 행동 단위 출력 값의 가중치 조정 및 행동 단위의 수행 주기를 결정해주는 태스크 관리자로 이루어진다. 하나의 태스크를 수행하기 위한 제어 구조의 구성이 그림 1과 같다.

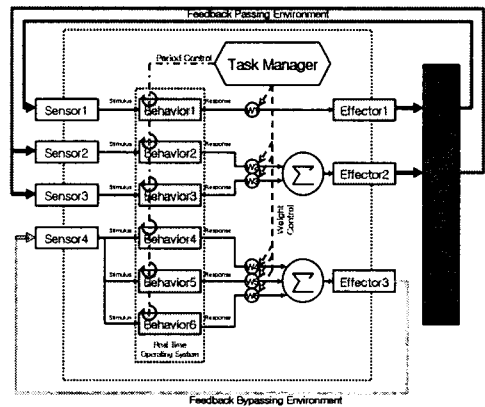


그림 1. 태스크 수행 제어 구조

그림 1에서 보는 바와 같이 하나의 태스크 수행 제어 구조는 다수 개의 센서(sensor)와 다수 개의 이펙터(effector), 그리고 센서로부터의 자극이 주어진 경우 이펙터로 반응을 발생하는 행동 단위로 구성 되어 있다. 다수의 행동 단위가 하나의 이펙터에 반응을 할 경우는 모터 스키마(motor schema) 이론[2]과 같이 벡터 합(vector summation)을 통해 각각의 반응이 통합된다. 이 때 각 반응의 스칼라(scalar)에 가중치( $W_n$ )를 곱하여 벡터 합이 이루어지며, 가중치는 주위 환경에 따라 태스크 관리자에 의해 설정된다.

각각의 행동 단위는 실시간 운영체제에서 하나의 쓰레드(thread)로서 주기적으로 수행되며, 수행 주기 역시 주위 환경의 변화에 따라 태스크 관리자에 의해 설정 된다.

그림1에서 보는 바와 같이 각각의 행동 단위는 반응의 결과로서 이펙터의 출력 값을 통한 환경의 변화 혹은 환경의 변화를 일으키지 않고 자극만의 변화를 피드백 받는다[4]. 대부분의 실제 피드백은 환경의 변화를 센서가 감지함으로써 이루어지며, 환경의 변화를 일으키지 않고 센서를 통한 자극만을 변화시키는 경우는 예를 들어 눈동자의 움직임을 통한 시야의 변화와 같은 경우에 해당한다.

## 2.2 가변 주기 제어

본 논문에서 제안한 태스크 수행 제어 구조는 실시간 운영체제 상에서 수행된다. 실시간 운영체제는 멀티 쓰레드(multi thread)-혹은 멀티 태스크(multi task) 운영 환경을 제공하며 그림 1의 행동 단위 각각은 쓰레드로 구현 된다.

멀티 쓰레드 환경에서 중앙처리장치 점유를 위한 스케줄링 전략은 크게 정적인 스케줄링 전략과 동적인 스케줄링 전략으로 나눌 수 있다. 일반적으로 동적인 스케줄링 전략은 스케줄링 연산에 필요한 중앙처리장치의 사용이 증가하여 효율이 안좋은 것으로 알려져 있다[8].

본 논문에서는 정적인 스케줄링의 대표적인 방식인 RMS[9] 기반의 가변 주기 제어 방식을 통해 태스크 관리자가 각각의 행동 단위의 수행 주기를 결정한다. RMS 방식은 최소의 주기를 갖는 쓰레드가 최고의 우선 순위를 갖고 수행되는 방식으로서 온라인(online) 상에서 주기를 가변한다면 수행되는 쓰레드의 우선 순위를 변경할 수 있다. 선점형 운영체제 환경에서 RMS 전략을 사용했을 경우에 중앙처리장치 이용율의 제한은 식(1)과 같다.

$$\sum_{i=1}^n \frac{C_i}{T_i} \leq n \left( 2^{\frac{1}{n}} - 1 \right) \quad (1)$$

식 (1)에서  $C_i$ 는 각 쓰레드의 연산 수행 시간을 의미하며,  $T_i$ 는 각 쓰레드의 수행 주기, 그리고  $n$ 은 쓰레드의 개수를 의미 한다. 식(1)을 만족하는 조건에서 특정한 쓰레드  $\tau_i$ 는 다음의 식(2)를 만족하는 경우 실시간성을 보장 받는다.

$$r_i = C_i + B_i + \sum_{V_{j \in hp(i)}} \left\lfloor \frac{r_j}{T_j} \right\rfloor C_j \leq T_i \quad (2)$$

식(2)에서  $r_i$ 는 쓰레드의 최악 응답 시간을 의미하며,  $B_i$ 는 자신보다  $\tau_i$ 보다 우선 순위가 낮은 쓰레드에 의해 발생하는 지연 시간을 의미하고  $hp(i)$ 는  $\tau_i$ 보다 우선 순위가 높은 쓰레드의 집합을 의미 한다.

스케줄링의 관점에서는 RMS 기반의 주기 가변 전략을 수행할 경우에 기본적으로 식(1)과 식(2)를 만족하는 어떠한  $T_i$ 도 허용할 수 있다.

하지만 피드백 제어 시스템의 관점에서는 그림1은 이산(discrete) 시스템인 제어가 연속(continuous) 시스템인 환경(environment)을 피드백 제어하는 것과 같으므로 이산 제어기의 샘플링(sampling) 시간에 해당하는 수행 주기의 선정은 제어의 안정도에 영향을 미치게 된다. 제어 안정도를 만족하는 수행 주기의 선정은 식(3)을 만족하여야 한다[10].

$$4 \leq \frac{T_r}{T_i} \leq 10 \quad (3)$$

식(3)에서  $T_r$ 은 시스템의 상승 시간(rise time)을 의미한다.

식(1), 식(2) 그리고 식(3)에서 보는 바와 같이 행동 단위의 구현 매체인 쓰레드의 주기  $T_i$ 는 식(1), 식(2) 그리고 식(3)을 공통적으로 만족하는 최소값이 존재하며, 식(3)을 만족하는 최대 값이 존재한다.

본 논문에서 제안하는 가변 주기 제어 알고리즘은 초기에 모든 쓰레드는 식(3)을 만족하는 최대의  $T_i$ 로 설정되어 태스크 관리자에 의해 주위 환경의 변화에 따라 식 (4)와 같이 스케일 인자(scale factor)  $\alpha_i$ 가 곱해져서 주기가 감소하는 방식이다.

$$T_i(t) = T(t-1) \times \alpha_i(t) \quad (T(0) = T_{max,i}, \alpha_i(t) \leq 1) \quad (4)$$

가변 주기 제어 알고리즘의 순차적인 수행은 다음과 같으며, 제어 알고리즘은 모든 쓰레드 주기의 최소 공배수에 해당하는 시간이 되면 태스크 관리자에 의해 수행 된다.

**STEP 1:** 각각의 쓰레드의 주기를 식(3)을 만족하는 최대 주기인  $T_{max,i}$ 로 설정 한다.

**STEP 2:** 태스크 관리자에 식(2)와 식(3)을 만족하는 범위에서 주위환경 정보에 의해 설정된 스케일 인자  $\alpha_i$ 의 집합  $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ 을 구한다.

**STEP 3:** 스케일 인자  $\alpha_i$ 의 집합  $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ 을 이용하여 주기  $T_i$ 의 집합  $\{T_1, T_2, \dots, T_n\}$ 을 구한다.

**STEP 4:** 정해진 주기  $T_i$ 의 집합  $\{T_1, T_2, \dots, T_n\}$ 이 식(1)을 만족하는지 여부를 확인 한다.

**STEP 5-1:** STEP 4가 만족하는 경우  $T_i$ 의 집합  $\{T_1, T_2, \dots, T_n\}$ 의 최소 공배수 만큼의 시간 후에 STEP 2를 반복 한다.

**STEP 5-2:** STEP 4가 만족하지 않는 경우  $T_n, T_{n-1}, \dots, T_1$ 의 순서대로 순환하면서 스케일 인자  $\alpha_i$ 를 증가시키면서 STEP2를 반복한다.

## 2.3 모의 실험

본 논문에서 제안한 주기 가변 기법의 타당성 및 효율성을 입증하기 위하여, 그림1의 제어 구조에서 주기가 변을 하는 경우와 그렇지 않은 경우에 대해 모의 실험을 통해 성능 비교를 수행한다.

모의 실험에서 태스크 관리자는 환경 정보를 가지함수인  $V_i(t)$ 로 사상하고 가지함수에 반비례하는  $\alpha_i(t)$ 를 결정한다. 본 논문에서 사용하는 각 행동 단위의 성능 비교의 지표는 식(5)와 같다.

$$\text{Performance\_Index}(t) = \sum \frac{V_i(t)}{T_i(t)} \quad (5)$$

식 (5)에서 보는 바와 같이 전체 시스템의 성능 향상을 위해서는 가치 함수 값이 가장 큰 행동 단위의 주기를 최소화할 때 최대의 시스템 성능을 기대할 수 있다.

표 1. 모의 실험을 위한 행동 단위의 시간 특성 값

행동 단위	$C_i$ [ms]	기본 RMS 설정 $T_i$ [ms]	$T_{max,i}$ [ms]
Behavior1	20	100	200
Behavior2	20	100	400
Behavior3	20	200	600
Behavior4	20	200	800
Behavior5	20	400	1000
Behavior6	20	400	2000

식 (5)의 가치함수 개념은 시스템의 최종 실패 방지를 위한 기존의 연구 결과에서 우선순위 선정의 방법으로써 연구된 바 있다[11].

모의 실험을 위해 그림 1의 행동 단위들의 시간 특성 값을 표 1과 같이 가정하였으며, 반응적인 행동 단위의 특성 상 기억 장치 등의 공유 자원 접근에서 발생하는 지연 시간  $B_i$ 는 0으로 가정하였다.

모의 실험에 사용되는 각 행동 단위에 있어서 시간의 흐름에 따른 가치함수의 변화는 그림 2와 같이 가정한다. 실제의 가치함수는 태스크 관리자가 환경 정보를 이용하여 실시간으로 생성한다.

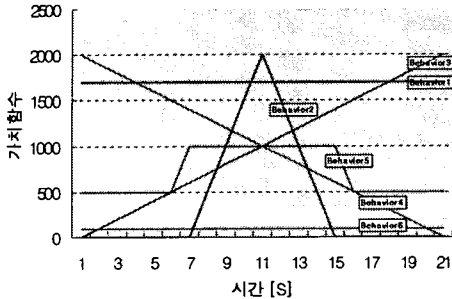


그림 2. 각 행동 단위의 가치 함수

그림 2와 같은 가치함수가 주어졌을 경우에 기존의 RMS 스케줄링 전략과 본 논문에서 제안하는 주기가변 전략의 성능 지표값의 추이는 그림 3과 같다. 그림 3은 그림 4와 같이 주기 가변이 이루어졌을 때의 결과이며, 그림 4의 주기 가변은 실시간적인 가치함수 값의 비교를 통해 단순히 비율적으로 스케일 인자인  $\alpha_i(t)$ 를 결정했을 때의 결과이다.

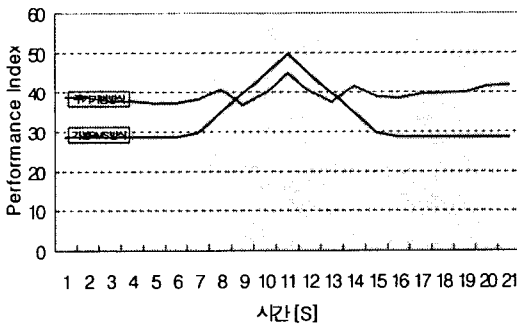


그림 3. 성능 지표 추이의 비교

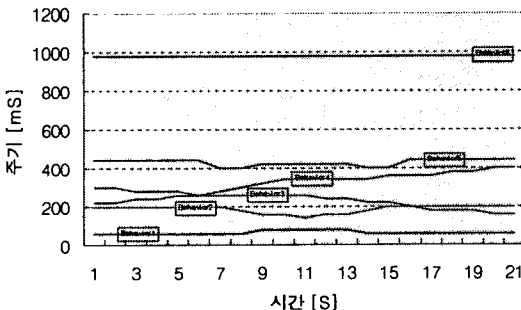


그림 4. 행동 단위의 주기 가변

그림 3의 결과서 보는 바와 같이 비교적 단순한 방법으로 스케일 인자를 결정했음에도 평균적으로 성능 지표 값이 향상되었음을 알 수 있으며, 또한 주위 환경의 변화에 따른 성능 지표의 편차가 기존의 RMS 방식에 비해 상대적으로 작음을 알 수 있다.

### 3. 결 론

본 논문에서는 이동로봇의 제어를 수행하는 행동 단위를 페루프 개념을 이용하여 설계하고, 이러한 행동 단위들의 동시적인 협조 수행을 통해 하나의 작업 단위인 태스크를 수행하는 제어 구조를 제안 하였다. 제안된 제어 구조는 최근의 생체학적 연구 결과를 토대로 하여 보다 지능적인 제어를 가능하게 할 것으로 예상된다.

제안된 제어 구조에서 태스크의 성공적인 완료를 위해 유용 자원의 효과적인 분배 및 행동 단위 출력의 가중치를 수행해주는 태스크 관리자의 개념을 제안하였다.

또한 본 논문에서는 제안된 제어 구조의 구현을 위해 실시간 운영체제를 갖는 이동로봇 제어기 환경에서의 중앙 처리 장치 점유 스케줄링 전략으로, RMS를 기반으로 한 가변 주기 전략을 제안하였으며, 이러한 가변 주기 전략을 수행하기 위해 RMS의 이용을 제한과 행동 단위의 제어 안정도를 판별하여 주기 가변 범위를 결정하는 방식을 제안하였다.

마지막으로 본 논문에서는 모의 실험을 수행하여 제안된 가변 주기방식의 타당성 및 효율성을 입증하였다.

향후 연구 과제로 주위환경 정보를 기반으로 하여 가용 자원의 이용율을 최대한으로 할 수 있는 주기 스케일 인자의 선택에 관한 연구가 수행될 예정이다.

### [참 고 문 헌]

- [1] R.A.Brooks, "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, Vol. RA2, No. 1, pp.14-23, 1986
- [2] R.C.Arkin, "Motor Schema Based Mobile Robot Navigation", Int. Journal of Robotics Research, Vol 8, pp.92-112, 1989
- [3] R.C.Arkin, "Behavior-based Robotics", MIT Press, 1998
- [4] H.A.Mallot, "Behavior-oriented approaches to cognition: theoretical perspectives", Theory in Biosciences 116, pp.196-220, 1997
- [5] G.Beccari, S.Caselli, M.Reggiani and F. Zanichelli, "Rate Modulation of Soft Real-Time Tasks in Autonomous Robot Control Systems", The 11th Euromicro Conference on Real-Time Systems (ECRTS99), pp.9-11, 1999.
- [6] T.W.Kuo, A.K.Mok, "Incremental Reconfiguration and Load Adjustment in Adaptive Real-Time Systems," IEEE Transactions on Computers, Volume 46, Number 12, pp.1313-1324, 1997
- [7] D.Seto, J.P.Lehoczky, L.Sha, K.G.Shin, "On Task Schedulability in Real-Time Control Systems", Proceedings of RTSS 96, pp.13-21, 1996.
- [8] H.Chetto, M.Chetto, "Some Results of the Earliest Deadline Scheduling Algorithm", IEEE Transaction on Software Engineering, Vol. 15 (10), pp.1261-1269, 1989
- [9] J.Lehoczky, L.Sha, Y.Ding, "The Rate Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behavior", IEEE Proc. on Real Time Symposium, pp.166-171, 1989
- [10] K.J.Astrom, B.Wittenmark, "Computer-Controlled Systems", Prentice Hall, 1997
- [11] G.Buttazzo, M.Spuri, F.Sensini, "Value Vs. Deadline Scheduling in Overload Conditions", IEEE Proc. on Real Time Systems Symposium, IEEE Computer Society Press., 1998