

# 주기 및 비주기 실시간 태스크 집합에 대한 수락제어 기법

박정근<sup>o</sup> 유민수<sup>\*\*</sup> 홍성수<sup>\*</sup>

<sup>\*</sup>서울대학교 전기·컴퓨터공학부  
{jpark<sup>o</sup>, sshong}@redwood.snu.ac.kr

<sup>\*\*</sup>한양대학교 정보통신대학  
msryu@hanyang.ac.kr

## Deterministic Admission Control for a Mixed Set of Periodic and Aperiodic Tasks

Jungkeun Park<sup>o</sup>, Minsoo Ryu<sup>\*\*</sup>, Seongsoo Hong<sup>\*</sup>

<sup>\*</sup>School of Electrical Engineering & Computer Science  
Seoul National University

<sup>\*\*</sup>College of Information & Communications  
Hanyang University

### 요 약

본 논문에서는 최악 수행시간이 알려진 주기 및 비주기 실시간 태스크 집합에 대한 수락제어 기법을 제안한다. 제안된 수락제어 기법은 주기 및 비주기 태스크가 혼합된 태스크 집합에 대해 이용률 요구(utilization demands)를 계산하여 태스크의 수락여부를 결정한다. 이용률 요구는 혼합 태스크 집합의 모든 태스크가 종료시한 내에 수행을 마치기 위한 프로세서의 이용률로 정의된다. 따라서 제안된 수락제어 기법은 수락된 모든 태스크의 종료시한을 보장한다. 제안된 이용률 요구 분석에 의한 수락제어 기법은 복잡한 스케줄링 가능성 분석이 필요하지 않아서 실시간 수락제어를 가능하게 한다.

### 1. 서 론

향후 내장형 시스템의 CPU와 네트워크 중심적 응용에 있어서 QoS 보장은 매우 중요한 문제이다. 예를 들어 멀티미디어를 지원하는 단말기에서는 제한된 자원으로 실시간 데이터를 전송하고 처리해야 한다. 그러나 현재 알려진 경성 실시간 보장 기술들 [1,2,3]은 대부분 주기적인 태스크 모델에 중점을 두어 다양한 QoS 응용에 직접 적용되기 어렵다. 반면 많은 네트워크 상의 QoS 보장을 요구하는 응용들에는 주기적 태스크와 비주기적 태스크가 동시에 존재한다. 예를 들어 멀티미디어 응용은 주기적 태스크에 의해 데이터 전송과 처리가 수행되고 정적인 데이터 처리는 비주기적 태스크에 의해 수행된다.

본 논문에서는 실시간 태스크의 수락제어(admission control)를 통해 종료시한을 보장할 수 있게 한다. 본 논문에서는 태스크 도착 시에 태스크의 최악 수행시간을 알 수 있는 경우만을 고려한다. 이 경우 수락제어 기법은 수락된 모든 태스크의 종료시한을 보장할 수 있어야 한다.

본 논문에서는 이용률 요구(utilization demands)에 기반하여 주기 및 비주기 혼합 태스크 집합에 대한 새로운 수락제어 기법을 제안한다. 이용률 요구는 혼합 태스크 집합의 모든 태스크가 종료시한 이내에 끝내기 위한 프로세서의 이용률로 정의된다. 본 논문에서는 이용률 요구를 EDF 스케줄링에 적용한다. 이용률 요구 분석은 복잡한 스케줄링 분석이 필요하지 않아서 실시간 수락제어를 가능하게 한다.

### 2. 모델과 가정

도착 순서로 정렬된 비주기 태스크의 집합을  $Q = \{\tau_1, \tau_2, \dots, \tau_i, \dots\}$ 로 나타낸다.  $Q(t) \subset Q$ 는 시간  $t$ 까지 수락되었지만 아직 종료하지 않은 태스크의 집합이다. 비주기 태

스크  $\tau_i \in Q$ 의 도착시간은  $A_i$ , 도착시간으로부터 상대 종료시한은  $d_i$ 로 표시하며 절대 종료시한  $D_i = A_i + d_i$ 이다. 각 비주기 태스크의 최악 수행시간은  $e_i$ 이다.

주기 태스크는  $\bar{\tau}_i$ 로 표시한다. 주기 태스크는 주기는  $\bar{T}_i$ 이고 주기 태스크의 각 인스턴스  $\bar{\tau}_{i,j}$ 의 상대 종료시한은 주기와 같고  $\bar{d}_i$ 로 표시한다. 각 주기 태스크의 최악 수행시간은  $\bar{e}_i$ 이다.

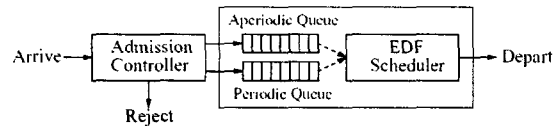


그림 1: 시스템 구조

본 논문에서는 그림 1과 같이 수락제어기와 태스크 큐 그리고 태스크 스케줄러로 구성된 단순한 시스템 구조를 고려한다. 또한 그림에서와 같이 본 논문에서는 EDF 스케줄만을 고려한다.

### 3. 수락제어 기법

#### 3.1 비주기 태스크에 대한 이용률 요구

독립적 비주기 태스크의 집합  $Q(t) = \{\tau_1, \tau_2, \dots, \tau_i, \dots\}$ 가 EDF에 의해 스케줄링 된다고 가정한다.  $\tau_i \in Q(t)$ 에 대해  $\tau_i$ 가 종료하기까지 남은 잉여 수행시간(residual execution time)을  $e_{i,t}$ 로 정의하고 절대 종료시한과 현재 시간의 차이를 리드 시간(lead time) [4]  $d_{i,t} = D_i - t$ 로 정의한다. 그림 2는  $\tau_k$ 만

수행될 때의 위의 값을 나타낸다.  $\tau_k$ 가 수행 가능하기 위해서는  $e_{k,i} \leq d_{k,i}$ 가 성립해야 한다.

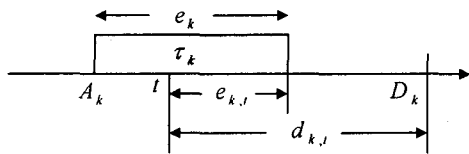


그림 2: 임의 수행시간과 리드 시간

임의 수행시간과 리드 시간에 의해  $\tau_i \in Q(t)$ 에 대한 이용률 요구  $U_{Q(t)}(\tau_i)$ 는  $\tau_i$ 가 종료시킨 이전에 마치기 위해 필요한 프로세서 시간을 리드 시간으로 나눈 값으로 정의된다. 일반적인 경우에  $\tau_i$ 는 높은 우선순위 태스크가 수행을 마친 후에 수행될 수 있으므로  $U_{Q(t)}(\tau_i)$ 는 분자에는  $\tau_i$ 보다 높은 우선순위 태스크의 임의시간을 포함한다.  $Q(t, ed(t')) \subset Q(t)$ 를 절대 종료시한이  $t'$ 보다 빠른 태스크들로 정의할 때  $\tau_i$ 에 대한 이용률 요구는 다음과 같다.

$$U_{Q(t)}(\tau_i) = \frac{\sum_{\tau_j \in Q(t, ed(D_i))} e_{j,i}}{d_{i,i}} \quad (1)$$

다음 정리는 비주기 태스크 집합에 대한 스케줄링 가능성의 필요충분 조건을 나타낸다.

정리 1: 비주기 태스크 집합  $Q(t)$ 는 임의의  $\tau_i \in Q(t)$ 에 대해

$$U_{Q(t)}(\tau_i) \leq 1 \quad (2)$$

가 만족될 때 필요충분 조건으로 EDF에 의해 스케줄 가능하다.

그러나  $\tau_i \in Q(t)$ 에 대한 스케줄링 가능성은  $\tau_i \in Q$ 에 대한 스케줄링 가능성과 다르다. 왜냐하면  $Q(t)$ 에는  $t$  이후에 수락될 태스크들이 포함되어 있지 않기 때문이다. 따라서 부등식 (2)에 의한 스케줄링 가능성은 다음 새로운 태스크가 도착할 때까지 유효하다. 따라서 매 비주기 태스크의 도착시점마다 부등식 (2)에 의한 스케줄링 가능성 조사가 수행되어야 한다.

### 3.2 주기 태스크에 대한 이용률 요구

주기 태스크 집합  $P = \{\tau_1, \tau_2, \dots, \tau_N\}$ 에 대해 각 주기 태스크의 인스턴스는 비주기 태스크로 볼 수 있다. 즉 어떤 주기 태스크의 인스턴스  $\tau_{i,j}$ 는 도착시간이  $A_k = \overline{A}_i + (j-1)\overline{T}_i$ 이고 상대 종료시한이  $d_k = \overline{T}_i$ 인 비주기 태스크  $\tau_k$ 로 볼 수 있다. 위와같이 주기 태스크 집합  $P$ 에 대해 가상 비주기 태스크 집합 (pseudo-a-periodic set)  $Q_p$ 를 연관시킬 수 있다. 이때  $P$ 는  $Q_p$ 가 스케줄 가능할 때만 필요충분 조건으로 스케줄 가능하다. 따라서 주기 태스크 집합에 대해서도 비주기 태스크에 대한 스케줄링 가능성 검사와 유사한 방법으로 스케줄링 가능성 검사를 할 수 있다.

그러나  $Q_p$ 는  $t$  이후의 태스크 도착에 대한 정보를 알 수 있고  $t$  이후에 도착한 태스크에 대해서도 종료시한이 보장되어야

한다는 점에서 일반 비주기 태스크 집합  $Q$ 와 다르다. 이를 위해 본 논문에서는 이용률 요구를 특정 시간 구간에 대한 값으로 확장한다.  $Q(t, t') \subset Q$ 를  $Q(t)$ 와 시간  $[t, t']$ 에서 수락될 태스크들을 포함하는 집합으로 정의한다. 이 정의에 의해  $Q(t, \infty)$ 는  $t$ 에 종료되지 않은 태스크들과  $t$  이후에 수락될 모든 태스크의 집합을 나타낸다.  $Q((t, t'), ed(D_i)) \subset Q(t, t')$ 을 절대 종료시한이  $D_i$ 보다 빠른 태스크 집합이라 할 때  $\tau_i \in Q(t, t')$ 에 대한 이용률 요구는 다음과 같이 확장 정의한다.

$$U_{Q(t, t')}(\tau_i) = \frac{\sum_{\tau_j \in Q((t, t'), ed(D_i))} e_{j,i}}{D_i - t} \quad (3)$$

위의 정의에 의해 더 일반적인 스케줄링 가능성 조건을 유도할 수 있다.

정리 2: 비주기 태스크 집합  $Q(t, t')$ 는 임의의  $\tau_i \in Q(t, t')$ 에 대해

$$U_{Q(t, t')}(\tau_i) \leq 1 \quad (4)$$

가 만족될 때 충분 조건으로 EDF에 의해 스케줄 가능하다.

위의 정리는 일반 비주기 태스크 집합  $Q$ 뿐만 아니라 가상 비주기 태스크 집합  $Q_p$ 에도 적용 가능하다. 비주기 태스크 집합에 대해서는  $Q(t, t') = Q(t)$ 로 놓으면 되고 가상 비주기 태스크 집합에 대해서는 임의의  $\tau_i \in Q(t, \infty)$ 에 대해  $U_{Q_p(t, \infty)}(\tau_i) \leq 1$ 의 조건을 만족시키면 모든 주기 태스크가 스케줄 가능하다. 일반적으로  $U_{Q_p(t, \infty)}(\tau_i) \leq U_p = \sum e_i / \overline{T}_i$ 를 만족한다.

### 3.3 혼합 태스크 집합에 대한 스케줄 가능성 조건

집합  $S$ 를 일반 비주기 태스크 집합  $Q$ 와 주기 태스크 집합  $P$ 에 대한 가상 비주기 태스크 집합  $Q_p$ 의 합집합이라 하면  $S = Q \cup Q_p$ 이다. 정리 2에 의해 임의의  $\tau_i \in S(t, \infty)$ 에 대해  $U_{S(t, \infty)}(\tau_i) \leq 1$ 의 조건을 만족시키면  $S(t, \infty)$ 는 스케줄링 가능하다. 그러나 이는 무한개의 태스크를 검사해야 하므로 구현할 수 없다.

본 논문에서는 busy period를 이용하여  $U_{S(t, \infty)}(\tau_i)$ 를 비주기 태스크  $Q(t)$ 에 의한 부분과 주기 태스크  $Q_p(t, \infty)$ 에 의한 부분으로 나눈다. 그리고 비주기 태스크 집합은 부등식 (4)에 의해, 주기 태스크 집합은  $U_p$ 에 의해 스케줄링 가능성 검사를 하여 혼합 태스크 집합의 스케줄링 가능성을 검사할 수 있다.

태스크 집합  $S$ 에 대해 idle구간 없이 수행되는 시간 구간들을 레벨 0 busy period라 한다. 종료시한이  $\tau_i \in S$ 의 종료시한  $D_i$ 보다 빠른 태스크 집합  $S((t, \infty), ed(D_i))$ 에 대해 마찬가지로 레벨  $\tau_i$  busy period를 정의할 수 있다.

태스크  $\tau_i$ 에 대해  $A_i$ 가 포함된 레벨  $\tau_i$  busy period를 잡고 그 시작 시간을  $b_i$ 라 하면 위의 busy period에 대한 이용률 요구는 다음과 같이 계산된다.

$$U_{S(b_i, D_i)}(\tau_i) \stackrel{def}{=} \frac{\sum_{\tau_j \in S((b_i, D_i), ed(D_i))} e_j}{D_i - b_i} \quad (5)$$

위의 식에서는 임의시간  $e_{j,i}$  대신에  $e_j$  가 사용되었다. 이는  $S((b_i, D_i), ed(D_i))$  의 어떤 태스크도 레벨  $\tau_i$  busy period의 시작  $b_i$  이전에 시작할 수 없기 때문이다. 다음 정리는 busy period에 대해서만 스케줄링 검사를 하면 된다는 것을 보인다.

정리 3: 혼합 태스크 집합  $S = Q \cup Q_p$  가 주어질 때  $S(t, \infty)$  는 임의의  $\tau_i \in S(t, \infty)$  에 대해

$$U_{S(b_i, D_i)}(\tau_i) \leq 1 \quad (6)$$

가 만족될 때 충분 조건으로 EDF에 의해 스케줄 가능하다.

본 논문에서는  $U_{S(b_i, D_i)}(\tau_i)$  를 주기 태스크에 의한 부분과 비주기 태스크에 의한 부분으로 나눈다. 여기서  $\tau_i$  는  $S(t, \infty)$  에서 선택된 임의의 태스크이고 모든 태스크에 대해 스케줄링 가능성이 보장되어야 한다. 임의의 시각  $t$  에서 임의의  $\tau_i \in S(t, \infty)$  에 대해

$$U_{S(b_i, D_i)}(\tau_i) = \frac{\sum_{\tau_j \in S((b_i, D_i), ed(D_i))} e_j}{D_i - b_i} \quad (7)$$

$$= \frac{\sum_{\tau_j \in Q_p((b_i, D_i), ed(D_i))} e_j}{D_i - b_i} + \frac{\sum_{\tau_j \in Q((b_i, D_i), ed(D_i))} e_j}{D_i - b_i} \quad (8)$$

$$= U_{pc} + U_{ac} \quad (9)$$

이 성립한다. 여기서  $U_{pc}$  는 주기 태스크에 의한 이용률 요구이고  $U_{ac}$  는 비주기 태스크에 의한 이용률 요구이다.

우선 주기 태스크에 의한 이용률 요구는  $U_p = \sum \bar{e}_i / \bar{T}_i$  에 의해 표현될 수 있다는 것을 보일 수 있다.  $Q_p((b_i, D_i), ed(D_i))$  중 절대 종료시한이 가장 긴 태스크를  $\tau_p$  라 하면  $D_p \leq D_i$  이다. 따라서

$$U_{pc} \leq \frac{\sum_{\tau_j \in Q_p((b_i, D_i), ed(D_i))} e_j}{D_i - b_i} \quad (10)$$

이다. 이때 분모는 구간  $[b_i, D_p]$  에서 주기 태스크에 의한 총 간섭시간 이므로 다음 관계가 성립한다.

$$U_{pc} \leq \frac{\sum_{\tau_j \in P} [(D_p - b_i) / \bar{T}_j] \bar{e}_j}{D_p - b_i} \leq \sum_{\tau_j \in P} \frac{\bar{e}_j}{\bar{T}_j} = U_p \quad (11)$$

비주기 태스크에 대해서도 유사하게  $Q((b_i, D_i), ed(D_i))$  중 절대 종료시한이 가장 긴 태스크를  $\tau_q$  라 하면

$$U_{ac} \leq \frac{\sum_{\tau_j \in Q((b_i, D_i), ed(D_i))} e_j}{D_i - b_i} = U_{Q(b_i, D_i)}(\tau_q) \quad (12)$$

이다. 위의 식에서  $b_i$  는  $S(t, \infty)$  상의  $\tau_i$  의 선택에 따라 달라질

수 있다.  $U_{ac}$  에서  $b_i$  의 영향을 없애기 위해 본 논문에서는 레벨 0 busy period를 고려한다.  $b_i$  를 포함하는 레벨 0 busy period의 시작 시간을  $B(t)$  라 하면  $b_i$  는 구간  $[B(t), t]$  에 수록된 태스크의 도착시간 중 하나가 된다. 따라서  $U_{ac}$  의 상한값은  $\tau_j \in S(B(t), t)$  인 모든 태스크에 대해서  $U_{Q(A_j, D_j)}(\tau_j)$  를 계산하여 얻을 수 있다. 또한 식 (12)에 의해  $[A_j, D_j]$  이전에 도착한 태스크들을 제외시킬 수 있다.  $U_{pc}$  와  $U_{ac}$  에 대한 상한값을 기반으로 다음 정리를 얻을 수 있다.

정리 4: 혼합 태스크 집합  $S = Q \cup Q_p$  가 주어질 때  $S(t, \infty)$  는 임의의  $\tau_j \in Q(t)$  에 대해 다음 조건이 만족될 때 EDF에 의해 스케줄 가능하다.

$$U_{ac}^{max} + U_p \leq 1 \quad (13)$$

$$U_{ac}^{max} = \max\{U_{Q(A_j, D_j) - Q(A_j)}(\tau_j) \mid \tau_j \in Q((B(t), t))\}$$

정리 4에 의해 주기 및 비주기 태스크가 혼합된 태스크 집합에 대해 스케줄링 가능성을 쉽게 분석할 수 있다. 구현 상으로는 현재 비주기 태스크 집합  $Q(t)$  와 주기 태스크 집합  $P$  를 위한 자료 구조만을 유지하면 되고 현재  $S(t)$  의 크기를  $n$  이라 할 때 제안된 스케줄링 가능성 분석은  $O(n^2)$  의 복잡도를 가진다.

#### 4. 결론

본 논문에서는 이용률 요구를 이용한 수락제어 방법을 제안하였다. 제안된 수락제어 방법은 주기 및 비주기 태스크가 혼합된 태스크 집합에 대해 이용률 요구를 계산하여 스케줄링 가능성을 조사할 수 있게 한다. 또한 제안된 수락제어 방법은 매우 적은 오버헤드로 구현될 수 있어 시스템 수행 중에 새로운 태스크의 수락여부를 효과적으로 계산할 수 있다.

본 논문에서는 태스크의 수행시간을 미리 결정할 수 있는 경우에 대해서만 살펴보았다. 그러나 본 논문에서 제시된 수락제어 방법은 태스크의 수행시간이 확률적 분포를 가지는 경우에 대해서도 쉽게 확장 가능하다.

#### 참고문헌

- [1] C. Liu and J. Layland. Scheduling algorithm for multiprogramming in a hard real-time environment. Journal of the ACM, 20(1):46-61, 1973.
- [2] N. Audsley, A. Burns, M. Richardson, and A. Wellings. Hard real-time scheduling: The deadline-monotonic approach. In Proceeding of IEEE Workshop on Real-Time Operating Systems and Software, pp. 133-137, 1991.
- [3] J. P. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. In Proceedings of IEEE Real-Time Systems Symposium, pp. 166-171, 1989.
- [4] J. P. Lehoczky. Real-time queueing theory. In Proceeding of IEEE Real-Time Systems Symposium, pp. 186-195, 1996.