

네트워크기반실시간제어시스템을위한지연보상기개발

Development of Delay Compensator for Network Based Real-time Control Systems

김승용, 김홍열, 김대원
(Seungyong Kim*, Hongryeol Kim**, Daewon Kim**)

Abstract - This paper proposes the development of delay compensator to minimize performance degradation caused by time delays in network-based real-time control systems. The delay compensator uses the time-stamp method as a direct delay measuring method to measure time delays generated between network nodes. The delay compensator predicts the network time delays of next period in the views point of time delays and minimizes performance degradation from network through considering predicted time delays. Control output considering network time delays is generated by the defuzzification of probable time delays of next period. The time delays considered in the delay compensator are modeled by using a timed Petri net model. The proposed delay prediction mechanism for the delay compensator is evaluated through some simulation tests by measuring deviation of the predicted delays from simulated delays.

Key Words : network based real-time control system, time delay, delay compensator, Petrinet

1. 서 론

네트워크 기반의 시스템은 다양한 이기종 시스템의 결합을 보장할 뿐만 아니라 이에 따르는 비용을 최소화 할 수 있기 때문에 최근 수많은 연구가 진행되었다. 또한 최근에는 네트워크 기반 시스템 이동 로봇과 같은 실시간 시스템에 적용하는 연구가 활발히 진행되고 있다[1][2]. 네트워크 시스템을 실시간 제어 시스템에 적용할 때 발생하는 가장 큰 문제점은 모든 시스템 구성 요소가 네트워크라는 하나의 매체를 공유하여 정보의 전달을 수행하기 때문에 시스템 지연시간이 상존하게 된다는 것이다. 특히 최근의 실시간 시스템은 실시간 운영체제를 사용하여 운영이 되기 때문에 이러한 실시간 운영체제에서의 지연시간과 네트워크로 인한 지연시간의 결합은 예측하지 못한 시스템의 제어 성능 저하의 원인이 되기도 한다.

네트워크 기반 시스템의 지연시간 발생에 따른 제어 성능을 보상해주기 위한 기존의 연구로 평균 지연시간을 고려한 연구[3]와 최악 지연시간을 고려한 연구[4]가 있다. 하지만 기존 연구가 오프라인(offline)에서의 네트워크 지연시간의 모델링(modeling)[4], 혹은 데이터 수집결과를 기반[3]으로 했기 때문에 온라인에서 발생하는 다양한 분포를 갖는 지연시간의 대표 값이 될 수 없다는 단점을 갖는다. 이러한 기존 연구의 단점 극복을 위해 온라인(online) 지연시간을 기반으로 수행된 연구로 과거의 지연시간만을 타임스탬프(time-stamp) 방식으로 측정하여 PID 제어 보상을 수행한 연구[5]와 타임스탬프 방식의 과거 지연시간 측정과 마코프 체인(Markov chain)을 이용한 미래의 지연시간 예측을 통해 LQG(Linear Quadratic Gaussian) 제어 보상을 수행한 연구[6]가 있다. 하

지만 이러한 연구 또한 네트워크의 제어기 노드(node)가 구동 노드로 제어정보를 전송할 때 제어에 고려되어야할 예측 지연시간이 고려되어 않거나[5], 이벤트 트리븐(event-driven) 시스템에서 무작위 지연시간 발생을 대상으로 하였기 때문에 [6] 운영체제 스케줄링(scheduling)과 네트워크 메시지 스케줄링이 주로 타임 트리븐(time-driven) 방식으로 설계되는 최근의 실시간 시스템에는 적합하지 않는다는 단점을 갖는다. 뿐만 아니라 아직 네트워크 시스템에서 실시간 운영체제 스케줄링까지 고려한 연구는 아직 수행된 바가 없다.

따라서 본 논문에서는 타임 트리븐 방식으로 설계된 네트워크 기반 실시간 제어 시스템을 위한 온라인 지연 보상기 개발을 수행한다. 본 논문의 지연 보상은 제어기로의 입력 전달 시 발생하는 지연시간의 측정을 위해서 타임스탬프 방식의 직접적인 지연 계측 방식을 사용하며, 제어기에서의 출력 전달 시 발생하는 지연시간의 예측을 위해 이전 출력 전달 지연시간을 기반으로 하여 다음 주기의 네트워크 지연시간을 확률적으로 예측한다. 확률적으로 예측된 네트워크 지연시간은 비퍼지(defuzzy)화를 통해 제어기에서의 출력 전달 시 발생이 예상되는 예측 네트워크 지연시간이 되어, 제어기로의 입력 전달 시 발생하는 측정 지연시간과 함께 지연 보상의 입력이 된다. 지연 보상기의 확률적인 지연시간 예측은 시스템의 시간 페트리넷(Petri net) 동적 모델을 이용해 수행되며, 페트리넷 동적 모델은 실시간 운영체제의 스케줄링을 고려하여 수행된다. 본 논문에서는 제안된 지연 보상의 성능을 좌우하는 예측 지연시간의 타당성을 입증하기 위하여 제안된 방식에 의한 예측 지연시간과 모의실험을 통한 실제 발생 지연시간과의 편차를 기존의 네트워크 지연을 고려한 연구에서 고려된 지연시간과 비교, 평가한다.

2. 본 론

2.1 네트워크 기반 실시간 제어 시스템

본 논문에서 대상으로 하는 네트워크 기반 실시간 제어 시스템은 그림1과 같다. 그림1에서 각 네트워크 노드에는 하나 이상의 다중 프로세스(process)가 운영되며, 운영되는 프로세스들은 실시간 운영체계에 의해 동작의 실시간성을 보장받게 된다. 네트워크 환경에서의 이러한 다중 프로세스 운영 환경은 최근의 네트워크 기반 로봇[2]과 같이 복잡한 분산 시스템의 일반적인 형태이다.

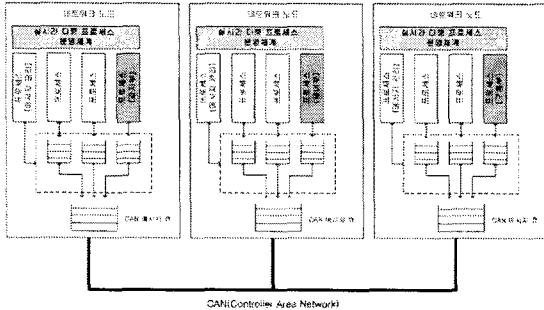


그림 1. 네트워크 기반 실시간 제어 시스템

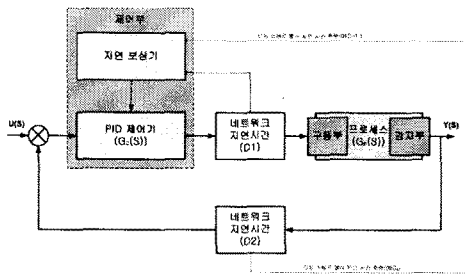


그림 2. 제어 폐루프

각 노드에 의해 공유되는 네트워크 상에서 운영되는 메시지 역시 실시간 시스템의 경우 운영 체계를 유사하게 스케줄링 방식을 적용하여 실시간성을 보장받게 된다. 예를 들어 대표적인 실시간 네트워크 프로토콜인 CAN(Controller Area Network)의 경우, 운영체계 스케줄링에서 사용되는 RMS(Rate Monotonic Scheduling)나, EDFs(Earliest Deadline First Scheduling)같이 운영체계를 위해 개발된 스케줄링 기법을 적용한 기존의 연구가 있다[7][8]. 본 논문에서는 각 노드의 운영체계 스케줄링 방식으로 RMS를 사용하고, 이러한 노드들이 RMS 방식으로 스케줄링되는 CAN을 통해 연결된 네트워크 제어시스템을 대상으로 하여 지연 보상기 개발을 수행한다. 그림 1에서 네트워크 메시지의 주기적인 전송을 위해 각 프로세스에서 발생된 메시지는 바로 메시지 큐(message queue)에 전달하지 않고 일종의 버퍼(buffer)를 사용하여 대기시키며, 대기 중인 전송 메시지는 짧은 주기를 갖는 메시지 관리 프로세스에 의해 CAN의 메시지 스케줄링 주기에 따라 버퍼에서 메시지 큐로 이동하게 된다. 본 논문에서는 이러한 메시지 관리 프로세스의 주기를 CAN의 메시지 최소 전송 주기에 비해 아주 작은 값으로 설정할 수 있음

을 감안하여, 버퍼에서 메시지 큐로의 이동에 필요한 지연시간은 발생하지 않는 것으로 가정한다.

그림 1의 시스템에서 전체 네트워크 시스템에 분산되어 있는 감지부, 제어부, 구동부는 그림 2와 같이 하나의 제어 폐루프를 구성한다. 그림 2에서는 일반적인 제어 시스템에 네트워크를 통한 지연시간을 고려하여 감지부에서 제어부로의 지연시간 D_2 와 제어부에서 구동부로의 지연시간 D_1 을 시스템 구성요소로 추가한다.

2.2 지연시간의 동적 모델

그림1과 같은 네트워크 기반 제어 시스템에서 프로세스와 네트워크 메시지를 타임 드리븐 방식으로 특정 주기를 기반으로 운영할 때에는 메시지 전송 및 수신을 수행하는 프로세스 수행 주기와 네트워크 메시지 전송 주기의 비동기성으로 인해 중단 프로세스 간 지연시간 D_{EZE} 는 식 (1)과 같이 표현된다[9]. 식 (1)에서 D_{TP} 는 메시지 전송 노드 실시간 운영체계 스케줄링에 의한 메시지 전송 프로세스의 지연시간, D_{TP2NAW} 는 메시지 전송 노드에서 전송 프로세스의 수행 주기와 메시지 전송 주기의 비동기성으로 인한 대기시간, D_N 은 메시지 스케줄링에 의한 메시지의 지연시간, D_{N2RPW} 는 메시지 수신 노드에서 수신 프로세스의 수행 주기와 메시지 수신 시기의 비동기성으로 인한 대기시간, 그리고 D_{RP} 는 메시지 수신 노드 실시간 운영체계 스케줄링에 의한 메시지 수신 프로세스의 지연시간을 의미한다.

$$D_{EZE} = D_{TP} + D_{TP2NAW} + D_N + D_{N2RPW} + D_{RP} \quad (1)$$

본 논문에서는 식 (1)의 지연시간 구성 요소와 실시간 시스템의 페트리넷 모델기법인 DDSPN(Discrete time Deterministic and Stochastic Petri Net)[10]을 이용하여 중단 프로세스 간 지연시간의 동적 모델링을 수행한다. DDSPN을 이용하여 수행된 중단 프로세스간 지연시간의 동적 모델은 그림 3과 같다. 그림 3에서 각각의 지연시간 구성 요소의 온라인에서의 값은 발생 확률을 갖는 유한한 개수의 트랜지션(transition)으로 표현된다. 예를 들어 메시지 스케줄링에 의한 메시지의 지연시간 D_N 은 $D_{N1}, D_{N2}, \dots, D_{Nk}$ 의 k개의 지연시간 값이 발생할 수 있으며 각각의 발생 확률은 $P_{N1}, P_{N2}, \dots, P_{Nk}$ 과 같다.

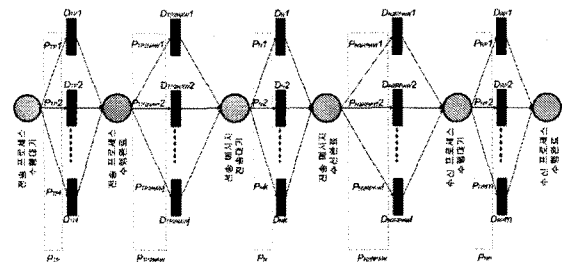


그림 3. DDSPN을 이용한 지연시간의 동적 모델

이렇게 각각의 지연시간 구성요소가 유한개 값의 확률적 분포 형태를 갖는 것은 프로세스의 수행과 네트워크 메시지의 전송이 주기성을 갖고 이루어질 때 RMS 방식으로 스케줄링

을 수행하는 경우 나타나게 된다. 주기적인 프로세스의 수행 혹은 메시지의 전송에 있어 RMS 방식으로 스케줄링을 하게 되면 스케줄링의 대상인 동일한 노드에서 운영되는 전체 프로세스의 주기 조합 혹은 전체 메시지의 주기 조합에 따라 최소공배주기를 갖고 유한한 개수의 지연시간이 반복되게 되며[11], 이로 인해 프로세스 주기와 네트워크 메시지 주기의 비동기성으로 인한 대기시간이 유한한 개수를 가지고 반복되게 되며, 결과적으로 전체 네트워크 시스템의 종단 간 지연시간 또한 유한한 개수를 가지고 반복하게 된다[9].

2.3 퍼지 추론 지연 보상기

본 논문에서 제안하는 퍼지 추론 지연 보상기는 그림4와 같다. 제안된 지연 보상기는 네트워크 전송 메시지에서 현재의 감지부에서 제어부로의 지연시간 D2(t)와 이전 제어 주기에서 제어부에서 구동부로의 지연시간 D1(t-1)을 추출한다. 지연 보상기는 D1(t-1)을 기반으로 하여 종단 간 지연시간의 페트리넷 동적 모델 및 각 지연시간 구성요소의 지연시간 페트리넷 모델을 이용하여 D1(t)를 확률적으로 예측하고 퍼지 추론 과정을 거쳐 최종적으로 D1(t) 값을 예측한다. 예측된 D1(t)는 D2(t)와 함께 시스템의 시변 요소로 추가되어 분석적 제어기 파라미터 설정 방식을 통해 PID 제어기 보상을 수행한다.

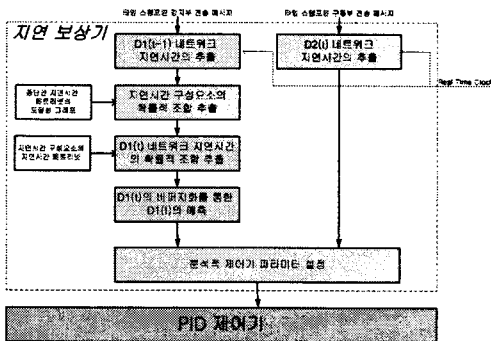


그림 4. 퍼지 추론 지연보상기의 구성

$$Tr (A_x D) = D1(t-1) \tag{2}$$

$$D = \begin{bmatrix} D_{TP1} & D_{TP2NAW1} & D_{N1} & D_{N2RPAW1} & D_{RP1} \\ D_{TP2} & D_{TP2NAW2} & D_{N2} & D_{N2RPAW2} & D_{RP2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D_{TPi} & \vdots & \vdots & \vdots & \vdots \\ 0 & D_{Nj} & \vdots & \vdots & \vdots \\ \vdots & 0 & D_{Nk} & \vdots & \vdots \\ \vdots & \vdots & 0 & D_{N2RPAWl} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & D_{RPm} \end{bmatrix} \quad (\text{when } m > l > k > j > i)$$

D1(t)의 예측을 위해 보상기는 먼저 종단간 지연시간 페트리넷의 도달성 그래프에서 식 (2)를 만족하는 n개의 (5×max(i,j,k,l,m)) 크기의 행렬 A_x에 대해 식(3)의 조건부 확률을 구한다. 식 (2)에서 행렬 A_x는 각 지연시간 구성요소를 열의 요소로 한 행렬로서 각 구성 요소의 최대 크기의 행을 갖고 각 열에서 구성 요소의 크기와 같은 다음 행부터 최대 구성 요소 크기까지의 행은 0인 (max(i,j,k,l,m)×5) 크기의 행렬이다. 도달성 그래프의 전체 종단 간 지연시간 공간

에서 D1(t-1)의 발생 가능 조건에서 D1(t-1)을 구성할 수 있는 조합인 n개의 행렬이 존재하는 확률인 식 (3)의 조건에서 D1(t)의 예측을 위해 각각의 지연시간 구성요소들의 지연시간 페트리넷을 그림 5와 같이 작성한다.

$$P(A_x | D_{E2E} = D1(t-1)) = \frac{P(A_x)}{P(D_{E2E} = D1(t-1))} \tag{3}$$

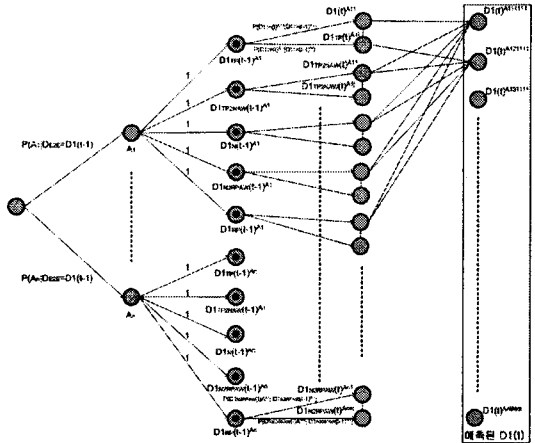


그림 5. 지연시간 페트리넷

그림 5에서 확률적으로 예측된 D1(t)는 식 (4)와 같이 대표적인 비퍼지화 방식인 COG(Center Of Gravity) 방식을 이용하여 최종적인 D1(t) 값의 예측에 사용된다.

$$D1(t) = \frac{\sum_{x=1}^{n+m} \{P(A_x | D_{E2E} = D1(t-1)) \times \sum D1(t)^{Ax} \times P(D1(t)^{Ax})\}}{\sum_{x=1}^{n+m} \{P(A_x | D_{E2E} = D1(t-1)) \times \sum P(D1(t)^{Ax})\}} \tag{4}$$

$$\sum D1(t)^{Ax} \times P(D1(t)^{Ax}) = D1(t)^{Ax11111} \times P(D1(t)^{Ax11111}) + \dots$$

$$D1(t)^{Axijklm} \times P(D1(t)^{Axijklm})$$

$$\sum P(D1(t)^{Ax}) = P(D1(t)^{Ax11111}) + \dots P(D1(t)^{Axijklm})$$

2.4 모의실험 및 결과

본 논문에서 제안한 지연 보상기의 성능 입증을 위해 지연 보상기의 성능을 좌우하는 예측 지연시간과 실제 지연시간의 편차를 모의실험 한다. 모의실험을 위해 그림 3과 같은 네트워크 실시간 시스템은 표1과 같은 시간 특성을 갖는 태스크와 메시지 집합으로 구성한다. 본 논문의 성능평가 대상이 되는 제어페루프의 제어부, 감지부, 구동부 프로세스와 네트워크 메시지는 표1의 음영으로 표시한다. 그림 6은 표1의 프로세스와 네트워크 메시지 집합을 모의실험했을 때 측정된 네트워크 지연시간과 기존의 연구결과에서의 지연시간 및 본 논문에서 제안된 메카니즘의 예측 지연시간과의 절대값 편차를 표현한다. 제안된 지연시간 예측값의 유용성을 평가하기 위해 각각 기존의 연구 결과에서 사용되었던, 최악 지연시간, 평균 지연시간, 그리고 과거 지연시간과 모의실험을 통해 측정된 지연시간과의 절대값 편차를 비교한다. 그림 6의 결과에서 평균적인 편차는 본 논문에서 제안한 예측 지연시간이 111ms로 가장 작고, 최악지연시간이 377ms로 가장 큰을 확인하였다. 최대 편차의 경우에도 제안한 예측

지연시간이 190ms로 가장 작고, 최악지연시간이 430ms로 가장 큼을 확인하여 제안된 메카니즘의 예측 지연시간 편차가 가장 작기 때문에 실제 발생하는 지연시간과 가장 유사함을 확인하였다.

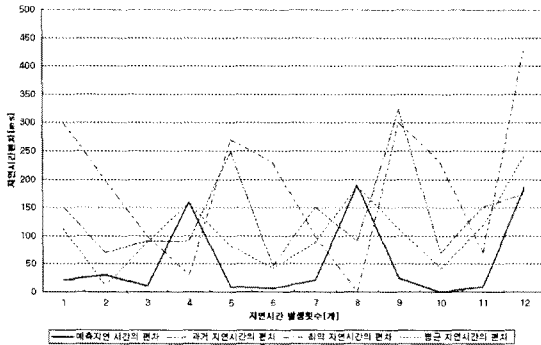


그림 6. 모의실험된 측정 지연시간과 각 지연시간의 편차

표 1. 프로세스 집합 및 네트워크 메시지 집합의 시간특성

위치	주기 [mS]	계산시간[mS]	
		메시지크기[Byte]	
제어부	100	20	
제어부	200	10	
제어부	300	20	
제어부	400	30	
구동부	100	30	
구동부	200	30	
구동부	300	30	
구동부	500	10	
감지부	100	30	
감지부	200	20	
감지부	400	10	
감지부	600	20	
네트워크 메시지	10	8	
네트워크 메시지	20	8	
네트워크 메시지	50	8	
네트워크 메시지	60	8	
네트워크 메시지	80	8	
네트워크 메시지	100	8	
네트워크 메시지	120	8	
네트워크 메시지	140	8	

3. 결론

본 논문에서는 타임 드리븐 방식으로 설계된 네트워크 기반 실시간 제어 시스템을 위한 온라인 지연 보상기 개발을 수행하였다. 본 논문의 지연 보상기는 제어기로의 입력 전달 시 발생하는 지연시간의 측정을 위해서 타임스탬프 방식의 직접적인 지연 계측 방식을 사용하며, 제어기에서의 출력 전달 시 발생하는 지연시간의 예측을 위해 이전 출력 전달 지연시간을 기반으로 하여 다음 주기의 네트워크 지연시간을 확률적으로 예측한다. 확률적으로 예측된 네트워크 지연시간은 비퍼지(defuzzy)화를 통해 제어기에서의 출력 전달 시 발생이 예상되는 예측 네트워크 지연시간이 되어, 제어기로의 입력

전달 시 발생하는 측정 지연시간과 함께 지연 보상기의 입력이 된다. 지연 보상기의 확률적인 지연시간 예측은 시스템의 시간 페트리넷(Petri net) 동적 모델을 이용해 수행되며, 페트리넷 동적 모델은 실시간 운영체계의 스케줄링을 고려하여 수행된다. 본 논문에서는 제안된 지연 보상기의 성능을 좌우하는 예측 지연시간의 타당성을 입증하기 위하여 제안된 방식에 의한 예측 지연시간과 모의실험을 통한 실제 발생 지연시간과의 편차를 기존의 네트워크 지연을 고려한 연구에서 고려된 지연시간과 비교하여 가장 우수함을 확인하였다.

향후 과제로 제안된 지연 보상기를 실제 실시간 네트워크 시스템에 적용하여 그 성능을 입증할 예정이며, 또한 예측 지연시간의 정확도의 향상을 위한 연구를 병행할 예정이다.

참고 문헌

- [1] Hongryeol Kim, Daewon Kim, HongSeok Kim, and Hogil Lee, "Toward the Personal Robot Software Framework", ICCAS, pp. 2307-2312, 2002
- [2] Sookyung Sohn, Hongryeol Kim, Daewon Kim, Hongseok Kim, and Hogil Lee, "Obstacle Avoidance Algorithm for a Network-based Autonomous Mobile Robot", ICCAS, pp. 831-833 2004
- [3] A. Mota, P. Fonseca, J. A. Fonseca, "System Modeling and Identification in CAN based Distributed Control Systems", DCCS2000: 16th IFAC Workshop on Distributed Computer Control Systems, 2000
- [4] 전종만, 김대원, "메시지 지연시간을 고려한 CAN 기반 피드백 제어 시스템의 응답특성 분석", 대한전기학회논문집, Vol. 51D, No5, pp. 190-196, 2002
- [5] 김홍열, 광권천, 김대원 "CAN 기반 피드백 시스템의 고속전철 여압시스템 적용" Journal of Control, Automation and System Engineering, Vol.9, No.11, pp. 953-968, 2003
- [6] Johan Nilsson, Bo Bernhardsson, "LQG Control Over a Markov Communication Network" proc. of the 36th CDC, pp. 4586-4591, Dec 1997
- [7] Dukjin Pae, Hongryeol Kim, Daewon Kim, Hongseok Kim, "Traffic Analysis of A Network-based Control System Using the CAN", 2002 Congress of IFAC, July 2002
- [8] 이병훈, 김홍열, 김대원, "CAN기반 실시간 시스템을 위한 확장된 EDS 알고리즘 개발", 대한전기학회논문집, Vol.51D, No.7, pp. 294-301, 2002
- [9] Hongryol Kim, Joomin Kim, Daewon Kim, "Development of Coordinated Scheduling Strategy with End-to-End Respose Time Analysis for the CAN-Based Distributed Control Systems", Proceeding of 2004 IEEE/RSJ International Conference on Intelligent Robot and System, pp. 2099-2104
- [10] Armin Zimmermann, Jörn Freiheit, and Günter Hommel, "Discrete Time Stochastic Petri Net for Modelling and Evaluation of Real-time Systems" Workshop on Parallel and Distributed Real-time Systems, San Fransisco, April 23-24, 2001
- [11] J.Lehoczky, L.Sha, Y.Ding, "The Rate Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behavior", IEEE Proc. on Real TimeSymposium, pp.166-171, 1989