

떨어진 입출력 장치를 가지는 프로그래머블 로직 컨트롤러를 위한 스케줄링 알고리즘

WCRT-Reducing Scheduling Algorithm for Programmable Logic Controllers with Remote I/Os

정 승권*, 권 옥현**

*서울대학교 전기공학부(Tel: +82-2-880-7314; Fax: +82-2-871-7010; E-mail: jsk@cisl.snu.ac.kr)

**서울대학교 전기공학부(Tel: +82-2-880-1800; Fax: +82-2-885-4459; E-mail: whkwon@cisl.snu.ac.kr)

Abstract In this paper, a scheduling algorithm is proposed for a programmable logic controller(PLC) with remote I/Os, assuming the multi-tasking facilities. Since sequence programs are executed on the application processor and I/O data are transmitted by the network processor concurrently, the proposed algorithm schedules the data transmission as well as the sequence program execution. The suggested algorithm guarantees the bounded WCRT(worst case response time), which is the one third of the WCRT in the absence of scheduling. Computer simulation shows that the algorithm can be easily applied to a real PLC without critical constraints on utilization of resources and inter-relation among tasks.

Keywords programmable logic controller, remote I/Os, response time, scheduling algorithm

1. 서론

PLC(Programmable Logic Controller)는 순차제어를 위해 개발된 것으로 FMS, 화학 공정, 교통 시스템 등에 널리 사용되고 있다[5, 6]. 제어 대상이 되는 플랜트가 점점 복잡해짐에 따라 PLC도 대형화되고 복잡해 지고 있는데, 많은 소용량 PLC가 네트워크로 연결되어 복잡한 플랜트의 일부분씩을 제어하는 것은 통신량이 큰 부담이 되므로 복잡한 플랜트의 제어에는 하나의 대용량 PLC가 이용된다. 대용량 PLC는 병렬 혹은 어레이 프로세서 형태의 컴퓨터 구조를 가지므로[5] 병렬 처리, 네트워크, 실시간 계산 등의 여러 컴퓨터 분야의 연구가 대용량 PLC에 적용될 수 있다. 현재까지의 PLC 연구중 많은 부분은 주로 PLC의 성능 향상에 초점이 맞추어져 있다. 따라서 PLC 전용 마이크로프로세서, 전용 연산을 도와주는 보조 연산 장치 등이 제안되었고[2, 3], 시스템 구조적으로는 병렬, 분산 시스템의 형태를 갖춘것이 많이 제안되었다[7, 9]. PLC는 센서 입력 리드, 순차 제어 프로그램 실행, 액츄에이터 출력값 변경의 과정을 반복하며 플랜트를 제어하고 이러한 일련의 과정이 수행되는 주기를 스캔 타임이라고 하므로 스캔 타임은 PLC에 대한 성능 지표가 된다. 속도에 대한 PLC의 성능 지표로서 최악 반응 시간(WCRT:Worst Case Response Time)이라는 것이 있는데 이것은 센서에 의해 입력되는 값이 얼마나 빨리 출력에 반영되는가를 나타내는 것이다. 그런데 현재 PLC의 주종을 이루는 소용량 PLC의 경우는 스캔 타임과 WCRT이 항상 같으므로 대개의 경우 스캔 타임이 속도에 대한 기존의 연구들에서 속도에 대한 PLC의 성능 지표는 스캔 타임으로 대표되었다.

대용량 PLC는 제어해야 할 입출력 접점수가 많고 입출력 디바이스들이 위치적으로 떨어져 있으므로 중앙 제어 모듈과 여러개의 리모트 입출력 모듈들로 구성된다. 중앙 제어 모듈과 입출력 모듈들은 네트워크로 연결되므로 중앙 제어 모듈은 다시 제어 프로그램 실행부(PEU:Program Execution Unit)와 데이터 전송부(DTU:Data Transmission Unit)로 나누어져 병렬적으로 제어 프로그램 실행과 입출력 데이터의 전송을 수행한다. 그런데 이러

한 구조는 스캔 타임을 줄일 수는 있으나 WCRT은 프로그램 실행과 데이터 전송을 한 유니트에서 실행하는 것에 비해 짧아지지 않는다. 왜냐하면 최악 경우 DTU에서 읽은 입력 데이터는 다음 스캔에서 PEU에 의해 사용되며 PEU에서 갱신한 출력 데이터는 다음 스캔에서 리모트 입출력 모듈로 DTU에 의해서 전송되기 때문이다. 따라서 WCRT은 스캔 타임의 세배가 되며 만일 소형 PLC에서와 같이 WCRT을 스캔 타임과 같이 줄이고자 한다면 PEU와 DTU의 병렬 수행이 불가능해진다.

그런데 PLC가 멀티-태스킹 기능을 제공할 경우, 즉 여러 시퀀스 제어 프로그램들이 각기 다른 주기로 독립적으로 실행될 수 있다고 하면 WCRT은 프로그램 실행과 데이터 전송의 스케줄에 따라 WCRT을 스캔 타임 이내로 줄이는 것이 가능하다. 동시에 PEU와 DTU의 사용율은 최대로 유지될 수 있는데 각기 다른 프로그램의 실행과 데이터 전송이 PEU와 DTU에서 파이프라인 처리될 수 있기 때문이다. 프로그램 실행과 데이터 전송을 WCRT을 줄이도록 스케줄하기 위해서는 각각에 대해 시작시간과 마감시간의 범위를 여러 상호 관계로부터 유도해야 한다. 상호 관계는 프로그램간의 직접 데이터 참조와 WCRT을 스캔 타임 이내로 제한하는 목적으로부터 발생하며 이를 만족시키는 스케줄 알고리즘은 기존 여러 연구 결과[9, 10]에 없는 것으로 새롭게 추가되어야 하는 것이다. 스케줄링 알고리즘에 의해서 각 프로그램 실행과 데이터 전송의 시작시간이 사전에 정해져도 이것을 실제 시스템에 구현하는 것은 어려운 문제인데, PEU, DTU라는 다른 두 유니트간에 시간의 동기화를 맞추는 일이 쉽지 않다. 이것은 네트워크 프로토콜이 무엇인가 그리고 그것을 어떻게 구현하는가에 따라 다르게 결정되어야 하는 문제이다.

본 논문에서는 리모트 입출력 장치를 가지는 PLC의 동작 방식을 분석하고 여기서 WCRT을 스캔 타임 이내로 줄이는 요구사항을 유도한다. 이를 기초로 프로그램 실행과 데이터 전송시간을 스케줄하는 스케줄링 알고리즘을 제안한다. 스케줄링 알고리즘의 성능을 컴퓨터 시뮬레이션으로 보이고 그 적용 가능성을 제시한다.

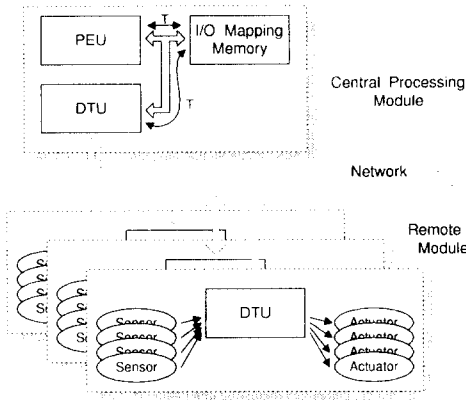


그림 1 리모트 입출력 장치를 가지는 PLC의 구조
Fig. 1 Architecture of a PLC with remote I/Os

2. PLC의 동작 특성

2.1 PLC 구조

PLC는 입력값 갱신, 제어 시퀀스 프로그램 실행, 출력값 변경이라는 세가지 기본 동작으로 플랜트를 제어하므로 구성도 센서, 액츄에이터, 프로그램 실행 유니트(PEU)로 이루어진다. 소용량 PLC의 경우는 프로그램 실행 유니트에 센서와 액츄에이터가 병렬 포트를 통해 바로 연결되어 있으며 대용량 PLC의 경우는 센서와 액츄에이터의 수가 매우 많고 또한 위치적으로 분산되어 있으므로 네트워크에 의해 프로그램 실행 유니트에 연결된다. 따라서 네트워크로 데이터를 송수신하기 위한 유니트(DTU)가 별도로 필요하며 대용량 PLC는 그림 1과 같은 구조를 가지게 된다. 그림 1에서와 같이 중앙 처리 모듈에는 집적 메모리가 있어서 모든 입력 데이터와 출력 데이터가 정해진 위치에 저장된다. 따라서 PEU는 프로그램 실행시 필요한 입력 데이터를 점점 메모리에서 가져오며 프로그램에 의해 바뀐 출력을 점점 메모리에 저장한다. DTU는 점점 메모리의 데이터를 실제로 경신하는 일, 즉 네트워크로 데이터를 송수신 하며 이때 데이터 경신 주기는 프로그램 실행 주기와 같게 된다. 리모트 입출력 모듈은 DTU와 센서, 액츄에이터로 이루어진다. DTU는 중앙 처리 모듈에서 요구하는 데이터를 전송하거나 전송되어 온 데이터를 받아들인다. DTU는 데이터를 메모리 버퍼에 가져다 놓고 네트워크와 인터페이스하는데 메모리 버퍼와 센서, 액츄에이터간에는 스캔 타임보다 극히 작은 주기로 맞추어진다.

2.2 멀티-태스킹

현재까지 대부분의 PLC는 하나의 시퀀스 프로그램이 하나의 스캔 타임을 가지고 실행되었다. 그런데 제어해야할 집짐수가 많아지면서 점점더 특성이 다를 경우가 많아지게 되어 제어 프로그램이 여러 부분으로 나누어져 서로 다른 스캔 타임으로 실행되는 것이 바람직하게 되었다[8]. 즉, 어느 부분은 50ms로 입력을 스캔하여야 상태 변화를 감지할 수 있고 어느 부분은 1s에 한번씩 입력을 받아들여도 된다고 할 때, 멀티-태스킹을 지원하지 않는다면 모두 50ms에 프로그램이 한번씩 수행되어야 하므로 필요 이상의 빠른 하드웨어 자원이 요구되기 때문이다. PLC가 멀티-태스킹을 지원할 경우 태스크들이 각각의 특성을 유지한 채 실행되기 위해서는 스케줄링이 필요하다. 일반적으로 스케줄링은 프로그램의 실행에 대해서만 생각하지만 대용량 PLC에서 WCRT이 성능 지표로 고려된다면 프로그램 실행과 함께 데이터 전송이 같이 스케줄되어야 한다. 프로그램에서 입력

데이터를 사용하는 명령어가 실행될 때 사용되는 데이터가 얼마나 오래전에 전송되어 온 것인가와 출력 데이터를 만드는 명령어가 실행되고 나서 얼마 후에 그 값이 액츄에이터에 나타나는가는 프로그램 실행과 데이터 전송의 스케줄에 달려 있기 때문이다. 물론 리모트 입출력 장치가 없어서 PEU와 DTU가 분리되지 않았다면 위의 두 값은 항상 일정하게 정해진 값으로 얻어질 수 있을 것이다. PLC의 경우 제어 프로그램과 입출력 장치의 구성이 결성되면 프로그램과 프로그램의 특성이 변하는 일이 거의 없으므로 사전(pre-run time) 스케줄링을 하는 것이 바람직하다.

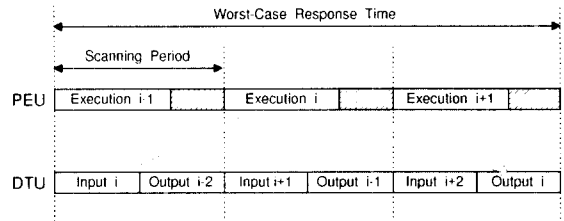


그림 2 리모트 입출력 장치를 가진 PLC의 데이터 흐름
Fig. 2 Data flow in a PLC with remote I/Os

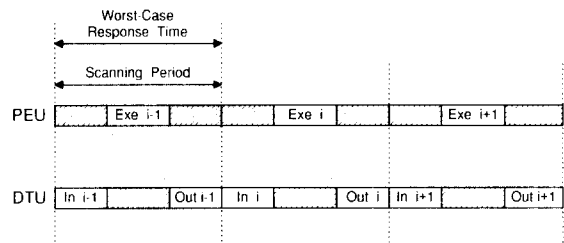


그림 3 동기가 맞추어진 경우의 데이터 흐름
Fig. 3 Data flow with task synchronization

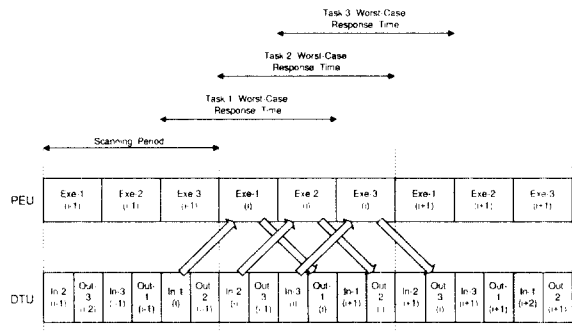


그림 4 멀티-태스킹 환경에서의 데이터 흐름
Fig. 4 Data flow with multi-tasking facilities

2.3 최악 반응 시간(WCRT)

PLC의 속도를 나타내는 지표로 스캔 타임과 WCRT을 들 수 있다. 스캔 타임은 입력과 프로그램 실행 그리고 출력이 얼마의 주기로 실행되는가를 나타내며 CPU 성능, 네트워크 전송율, 입출력 디바이스의 액세스 타임 등 하드웨어 성능에 좌우된다. WCRT은 입력이 출력에 반영되는 최대 시간을 의미하는데 이것은 스캔 타임과 같이 하드웨어 성능에 좌우될 뿐 아니라 스케줄링과 같은 소프트웨어적인 요소에도 좌우된다. 일반적인 하나의 CPU만이 사용되는 소용량 PLC에서는 스캔 타임과 WCRT은 그 크기가 같지만 그림 2와 같이 프로그램 실행과 데이터의 전송이 각기 다른 유니트에서 실행되는 대용량 PLC의 경우

WCRT은 스캔 타임의 세배가 된다. 만일 하나의 시퀀스 제어 프로그램이 실행될 경우 소용량 PLC에서와 같이 WCRT을 스캔 타임과 같이 줄이고자 한다면 그림 3에서와 같이 PEU와 DTU가 동시에 실행되지 못하므로 자원의 낭비가 생기게 된다. 그런데 PLC가 멀티-태스킹을 지원한다면 데이터 전송과 프로그램 실행의 스케줄 여하에 따라 그림 4와 같이 PEU와 DTU의 사용율을 최대로 유지한 채 WCRT을 스캔 타임내로 제한하는 것이 가능하다. WCRT을 스캔 타임 이내로 줄이기 위해서는 중앙 처리 모듈에서 PEU와 DTU간의 동기가 맞아야 하며 중앙 처리 모듈의 DTU와 리모트 모듈의 DTU간의 동기가 맞아야 한다. 또한 프로그램 실행과 데이터 전송이 사전에 스케줄되어야 한다.

3. 제안된 스케줄링 방법

WCRT이 스캔 타임이내가 되도록 스케줄링되기 위해서는 몇 가지 필요조건들이 만족되어야 한다. 첫째는 주기적인 태스크만이 존재하며 주기들간에는 하모닉한 관계가 성립하여야 한다는 것이다. 일반적으로 PLC는 입출력 장치들을 오버 샘플링하여 제어하므로 하모닉한 주기로 태스크들을 묶는 것은 충분히 가능하다. 둘째는 프로그램의 실행 시간과 데이터 전송시간이 사전에 특정값으로 예측될 수 있다는 것이다. 시퀀스 제어 프로그램이 정해지면 데이터의 크기가 정해지므로 필드버스와 같은 실시간 네트워크에서는 전송 시간이 예측될 수 있고 프로그램 실행 시간도 실제 최대값보다 어느 정도 큰 값을 예측될 수 있다. 셋째는 예측된 값에 의해 PEU와 DTU의 사용율이 1을 넘지 않아야 하며 각 태스크에 대해 PEU와 DTU의 사용율 합이 역시 1을 넘지 않아야 된다는 것이다. 식 (1), (2), (3)은 이러한 관계를 나타내며 여기서 태스크 i 의 실행 주기가 T^i , 실행 시간은 E^i , 입력 데이터 전송 시간은 N_{in}^i , 출력 데이터 전송 시간은 N_{out}^i 로 각각 표시되었다.

$$\sum_i \frac{E^i}{T^i} < 1 \quad (1)$$

$$\sum_i \frac{N_{in}^i + N_{out}^i}{T^i} < 1 \quad (2)$$

$$\frac{E^i + N_{in}^i + N_{out}^i}{T^i} < 1 \quad (3)$$

3.1 선행관계 조사

PLC에서 각 태스크들이 다른 태스크 출력 데이터를 참조할 경우는 프로그램 실행과 데이터 전송사이에 복잡한 관계가 생기므로 스케줄링시 특별한 고려를 해야한다. 이것이 필요한 이유는 태스크가 다른 태스크의 출력 데이터를 참조할때 프로그램에서는 데이터를 점점 데이터 메모리에서 가져오므로 실제로 액츄에이터에 반영되지 않은 값을 연산에 이용할 수 있기 때문이다. 출력 데이터 참조가 발생할때 WCRT은 참조되는 출력이 액츄에이터에 반영된 이후부터 그것을 참조하는 프로그램의 출력이 액츄에이터에 나타날 때까지의 최대 시간으로 정의한다.

그림 5에는 태스크간에 출력 참조가 발생할때 생길수 있는 문제를 예로 들었다. 태스크 3이 태스크 2의 출력을 참조할때

$$EX2 \Rightarrow EX3 \Rightarrow OUT3 \Rightarrow OUT2$$

과 같은 프로그램 실행과 데이터 전송 순서는 논리적 오류를 발생시킨다(여기서 $EX\#$ 은 태스크 #의 실행을 $OUT\#$ 은 태스크 #의 출력 데이터 전송을 의미). 그림 5에서 $Y3$ 을 펌프 구동 $Y4$ 를 밸브 오픈이라고 하면 시퀀스 프로그램에서는 $Y4$ 를 참조하여 $Y3$ 을 생성하므로 밸브가 열린 후에 펌프를 구동하라는 의미가 된다. 그러나 그림 5와 같이 스케줄된다면 태스크 3은 태스크 2에

서 밸브 오픈이라는 결과를 참조하여 실제로 밸브가 오픈되기 전에 펌프를 구동하므로 프로그램의 의미와는 다른 결과를 생성하게 된다. 이러한 논리적 오류이외에 WCRT을 증가시키는 경우도 있는데 그림 5에서와 같은 태스크간의 관계가 설정되어 있을때 태스크 1의 실행 주기가 태스크 2의 실행 주기보다 크다고 하면

$$EX2 \Rightarrow EX1 \Rightarrow OUT1 \Rightarrow OUT2$$

와 같은 실행 순서는 태스크 2의 WCRT을 증가시킨다. 만일 태스크 2의 실행 주기가 더 크다면 $EX1 \Rightarrow OUT1$ 이 태스크 2의 실행 주기 사이에 한번 이상 나타나므로 WCRT은 그림 5에서와 같이 증가하지 않는다.

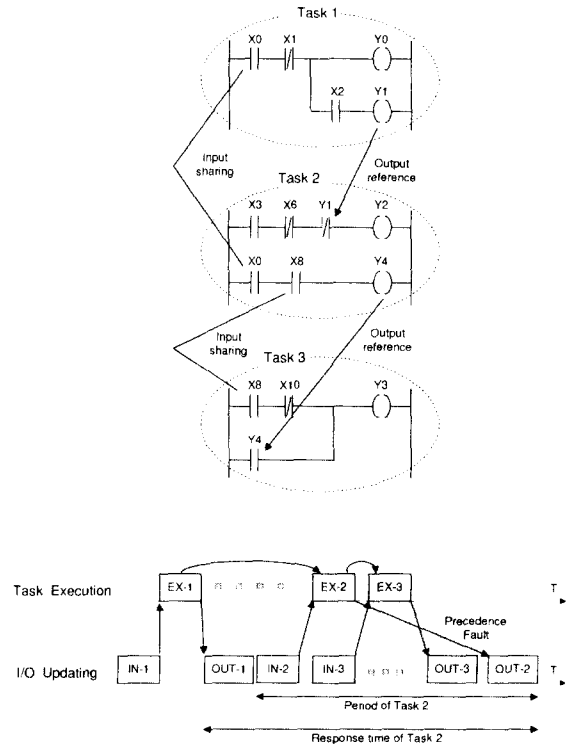


그림 5 태스크간의 참조 관계
Fig. 5 Inter-task relations

3.2 스케줄링 알고리즘

각 태스크의 WCRT이 스캔 타임 이내로 제한되고 3.1절에서와 같은 실행순서가 되지 않도록 프로그램 실행과 데이터 전송을 스케줄할 수 있는 알고리즘을 제안한다. 제안되는 알고리즘은 태스크별로 스캔 타임내에서 프로그램과 데이터 전송을 최소 스케줄링 시간 단위로 움직여가며 스케줄링 목적이 만족될때까지 스케줄하는 것이다. 입력 파라미터로는 태스크 주기, 태스크별 프로그램 실행 시간, 태스크별 입력 데이터 전송시간, 태스크별 출력 데이터 전송시간, 태스크간 출력 참조 관계 등이 되며 스케줄링 결과는 각 프로그램과 데이터 전송의 시작 시간이다. 시작 시간은 태스크별로 하나의 시간차로 표현되는데 주기의 정수배를 여기에 더해 모든 시간 구간에서 시작시간을 구할 수 있게 된다. 주어진 알고리즘에서 태스크 시작 시간차는 $\Phi_{ex}^k, \Phi_{in}^k, \Phi_{out}^k$ 로 표시되어 있다.

Main{

$R_{ex} = (-\infty, \infty); R_{in} = (-\infty, \infty);$

/* Set the residual time of PEU and DTU vacant */

if(Schedule(0)=OK) Assert(Schedule_OK);

```

else Assert(Schedule_NO);
/* Assert(#) : produce the scheduling result of # */
} /* Main */
Schedule(k){
  if(k==M) return OK; /* M is the number of task */
  for( $\Phi_{ex}^k=0; \Phi_{ex}^k < T^k; \Phi_{ex}^{k+} = T_{min}^k$ )
  for( $\Phi_{in}^k = \Phi_{ex}^k - T^k + E^k; \Phi_{in}^k < \Phi_{ex}^k; \Phi_{in}^{k+} = T_{min}^k$ )
  for( $\Phi_{out}^k = \Phi_{ex}^k + E^k; \Phi_{out}^k \leq \Phi_{in}^k + T^k - N_{out}^k; \Phi_{out}^{k+} = T_{min}^k$ ){
  /*  $\Phi_{ex}^k, \Phi_{in}^k, \Phi_{out}^k$  meets the constraint that sequence is
   $IN_k \Rightarrow EX_k \Rightarrow OUT_k$  and the response time is bounded
  such as  $\Phi_{out}^k + N_{out}^k - \Phi_{in}^k < T^k$  */
  DMex = [ $\Phi_{ex}^k + T^k \cdot I, \Phi_{ex}^k + T^k \cdot I + E^k$ ];
  DMio = [ $\Phi_{in}^k + T^k \cdot I, \Phi_{in}^k + T^k \cdot I + N_{in}^k$ ]  $\cup$ 
  [ $\Phi_{out}^k + T^k \cdot I, \Phi_{out}^k + T^k \cdot I + N_{out}^k$ ];
  /* I is an arbitrary integer */
  /* DMex is occupied time by task execution */
  /* DMio is occupied time by data transmission */
  if( $R_{ex} \supseteq DM_{ex}$  AND  $R_{io} \supseteq DM_{io}$  AND PRECHK(k)==OK){
  /* PRECHK() inspects precedence relation about
  output reference */
  Rex = Rex - DMex;
  Rio = Rio - DMio; /* update residual time */
  if(Schedule(k+1)!=OK){
  Rex = Rex + DMex;
  Rio = Rio + DMio; /* restore residual time */
  }
  } /* for for for */
} /* Schedule */

```

4. 성능 분석

제안된 알고리즘은 대용량 PLC 모델에 적용할 경우 식 (4)와 같은 WCRT R 을 얻을 수 있다. 여기서 T_{poll} 은 리모트 입출력 장치에서 전송되는 데이터와 센서, 액츄에이터 간의 값과 일치시키는 주기이다. 일반적으로

$$\forall i T_{poll} \ll T$$

이 성립하므로 제안된 알고리즘을 적용하지 않았을때의 WCRT R' 은 R 의 약 3배가 된다.

$$R_i = 2T_{poll} + T' \quad (4)$$

$$R'_i = 2T_{poll} + 3T' \quad (5)$$

제안된 스케줄링 알고리즘은 대부분의 경우에 대해 스케줄된 시간표를 생성해 낼 수 있다. 그림 6은 PEU와 DTU의 사용률이 각각 90%이상인 여러 경우에 대해 제안된 알고리즘이 스케줄된 결과를 생성할 수 있는 확률을 보여준다. 그림 6에서와 같이 태스크의 수가 어느 정도 이상 많아지면 스케줄링 필요 조건인 식 (3)이 대부분의 경우에 대해 만족하게 되며 리모트 입출력 장치의 수가 많아지면 식 (3)이 만족되는 범위하에서 대부분 스케줄된 결과를 얻을 수 있음을 보여준다.

5. 결론

본 논문에서는 멀티-태스킹을 지원하며 리모트 입출력 장치를 가지는 대용량 PLC를 위한 스케줄링 알고리즘을 제안하였다.

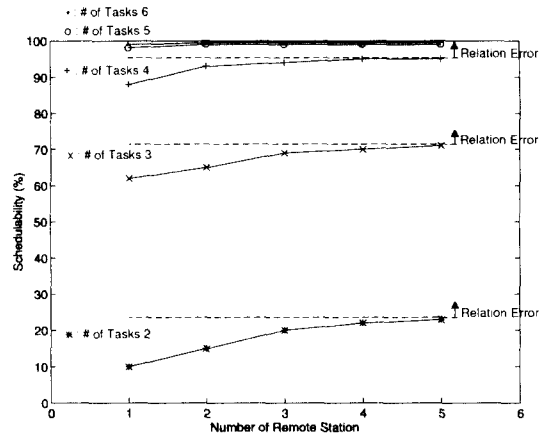


그림 6 스케줄 가능성 모의실험

Fig. 6 Simulation of Schedulability

제안된 알고리즘은 프로그램 실행과 데이터 전송을 스케줄하여 PLC의 속도에 대한 성능 지표인 WCRT을 세배 단축시키는 것이다. 또한 컴퓨터를 통한 모의 실험 결과 실제 PLC에 적용되어도 스케줄 목적에 맞는 시간표를 대부분 경우 생성할 수 있는 실용성 있는 것이다. 제안된 알고리즘은 대용량 PLC의 특정 구조에 대해 적용될 수 있는 것으로 다른 여러 구조에 대한 확장 은 추가 과제라 할 수 있다.

참고 문헌

- [1] C. D. Locke, "Software Architecture for Hard Real-Time Applications: Cyclic Executives vs. Fixed Priority Executives," *The Journal of Real-Time Systems*, vol. 4, pp. 37-53, 1992.
- [2] J. Kim, J. Park, and W. H. Kwon, "Architecture of a Ladder Solving Processor for Programmable Controllers," *Microprocessors and Microsystems*, vol. 16, no. 7, pp. 369-379, 1992.
- [3] G. Rho, K. Koo, N. Chang, J. Park, Y. Kim, and W. H. Kwon, "Implementation of a RISC Microprocessor for Programmable Logic Controllers," *Microprocessors and Microsystems*, vol. 19, no. 10, pp. 431-440, 1992.
- [4] J. Xu and D. L. Parnas, "Scheduling Processes with Release Times, Deadlines, Precedence, and Exclusion Relations," *IEEE Trans. on Software Eng.*, vol. 16, no. 3, pp. 360-369, 1990.
- [5] C. D. Simpson, *Programmable Logic Controllers*, Prentice Hall, 1994.
- [6] I. Warnock, *Programmable Controllers - Operation and Application*, Prentice Hall, 1988.
- [7] GE Fanuc Automation, *Series 90-30 Programmable Controller Installation Manual*, GE Fanuc Automation, GFK-0356D, 1992.
- [8] International Electrotechnical Commission, *Programmable Controllers - Part 3: Programming Languages*, IEC Publication 1131-3, 1992.
- [9] Siemens Automation Group Industrial Automation Systems, *SIMATIC S5 S5-135U, S5-155U and S5-155H Programmable Controllers Catalog ST 54.1*, Siemens Automation Group Industrial Automation Systems, 1994.