

다중샘플링 다중작업을 수행하는 실시간제어시스템의 시계수제 한성 유도

Derivation of the Timing Constraints for Multi-Sampled Multitasks in a Real-Time Control System

이 대 현, 김 학 배
(Dae Hyun Lee, and Hag Bae Kim)

Abstract : A real-time control system, composed of the controlled processor and the controller computer(s), may have a variety of task types, some of which have tight timing-constraints in generating the correct control input. The maximum period of those task failures tolerable by the system is called the hard deadline, which depends on not only fault characteristics but also task characteristics. In the paper, we extend a method deriving the hard deadline in LTI system executing single task. An algorithm to combine the deadlines of all the elementary tasks in the same operation-mode is proposed to derive the hard deadline of the entire system. For the end, we modify the state equation for the task to capture the effects of task failures (delays in producing correct values) and inter-correlation. We also classify the type of executing the tasks according to operation modes associated with relative importance of correlated levels among tasks, into series, parallel, and cascade modes. Some examples are presented to demonstrate the effectiveness of the proposed methods.

Keywords : timing constraint/deadline, multi-sampling, series/parallel/cascade operation

I. 서론

근래에 속도와 성능면에서 눈부신 기술발전을 거듭하는 디지털컴퓨터는 항공기, 핵발전소 및 인공위성 등의 실시간제어시스템 뿐만 아니라 고신뢰도·고성능을 요구하는 실시간생산시스템에서 제어 및 감시기능을 위해서 널리 활용되고 있다. 일반적으로 아날로그 신호로 동작하는 시스템/플랜트의 안정성 유지와 성능향상을 위해서 디지털컴퓨터 제어기는 적절한 주변장치를 이용하는데, 이러한 장치의 활용으로 인해서 궤환루프(feedback loop)에서 필연적으로 작지 않은 시간적 지연현상이 발생하게 되지만, 일반적으로 시스템에 고장이 발생하지 않고 정상적인 동작을 수행할 때 이는 상대적으로 작으므로 시스템의 안정성 및 성능에 미세한 영향을 미치게 된다. 그러나, 이와는 다르게 내적(전체작업시간, 주기, 제어알고리즘, 자원제한성 등) 또는 외적(시스템이 운영되는 환경에서 발생하는 EMI, HIRF, NEMP 등)인 요인으로 인해서 발생하는 제어기 고장은 상당한 시간지연과 함께 시스템의 안정성에 심각한 영향을 미치게 된다. 따라서, 시스템의 안정성 유지를 위해서 궤환루프상의 제어기는 다양한 고장검출 및 고장회복알고리즘을 사용함으로써 이러한 고장을 극복하여야 한다[1]. 하지만, 고장회복을 통한 제어기의 올바른 동작을 위해서 궤환루프에서는 작지 않은 제어기의 계산지연시간(computation time delay)이 발생하게 되고, 이것이 예상외로 길어질 경우에 시간제한속성을 갖는 실시간제어시스템(이러한 특성을

갖는 대부분의 시스템이 safety/life-critical systems이다)에 인명 및 재산상의 돌이킬 수 없는 치명적인 결과를 낳을 수 있게된다[2]. 이와 같이 실시간제어시스템은 시스템 동작이 제어기의 논리적인 출력값(logical values)에만 의존하지 않고, 시간적인 제한성(timing constraints)을 만족시켜야만 시스템이 안정성을 유지하면서 올바르게 동작하는 시스템으로 정의된다.

지금까지 많은 연구자들은 원시/모의 실험 및 단순한 이론적 모델링을 이용한 접근법이 대부분이었다[2, 3]. 실제로 시스템에 대한 간단화한 모형을 제작함으로써 실험을 통한 계산지연시간의 효과를 분석하였거나[3], 시뮬레이션을 통해서 시스템을 안정하게 유지하기 위한 시간제한정보를 얻으려고 시도하였다[2]. 물론, 종료시한의 특성을 파악하여 시스템의 적절한 설계를 고려하기도 하였지만, 제어시스템의 설계를 위해서 필수적인 종료시한 값을 제어시스템의 유동적 특성을 반영하여 정확하게 유도한 연구는 구체적으로 지금까지 이루어지지 않았다. 실시간제어시스템에서 종료시한은 수행작업의 적절한 예정계획뿐만 아니라 성능 및 신뢰도 분석을 위해서도 필수적인 요소이므로 시스템의 동적특성을 나타내는 파라미터를 이용한 종료시한 분석은 필수적이다. 특히, 오늘날 많은 제어시스템들이 다양한 동적특성을 갖는 여러 작업을 동시에 또는 순차적으로 수행하도록 요구되고 있고, 이들간의 상호연관성 및 선행수행조건이 더욱더 복잡한 형태를 갖기 때문에, 이를 고려하여 시스템의 동적특성에 기반한 종료시한을 유도하는 것은 더욱 어려워지게 되었다.

이를 보완하기 위해, 본 논문에서는 [4][5]에서 제안된 단일작업을 수행하는 선형시불변시스템에 대한 종료시한

접수일자 : 1998. 5. 13., 수정완료 : 1998. 11. 12.

이대현, 김학배 : 연세대학교 기계전자공학부 전기전공

※ 본 논문은 과학기술재단의 1997년도 핵심연구지원사업에 의해 지원되었습니다.

유도알고리즘을 확장하여, 다중샘플링 다중작업을 수행하는 실시간제어시스템에 대해 각 개별 작업 종료시한을 이용하여 전체시스템 종료시한을 유도하기 위한 통합알고리즘을 제시한다. 다중작업을 수행하는 시스템의 동적특성에 기반한 상태방정식(state-space equation)을 이용하여 지연효과를 갖는 수정누적상태방정식(modified augmented state-space equation)을 유도한다. 이를 이용하여 [5]에서 제안된 순차적회귀기법을 활용하여 각각의 작업에 대한 종료시한을 유도하고 각 작업들이 수행되는 작업방식(operation modes)에 따라서 순차적회귀기법으로 얻어진 각각의 개별 작업종료시한들을 조합함으로써 전체 시스템의 종료시한을 유도할 것이다.

II. 작업수행방식 분류

생산시스템을 비롯하여 디지털 컴퓨터로 제어되는 대부분의 시스템/플랜트는 여러 종류의 작업을 동시에(Concurrent) 또는 연속적(sequential)으로 수행한다. 이러한 경우, 전체시스템의 종료시한에 영향을 미치는 세 번째 요소인 작업수행방식을 고려한 종료시한 유도 알고리즘이 중요하게 고려될 수 있다. 이 때, 개개의 작업수행에 의해 변하는 시스템의 상태변수가 하나이상의 작업수행과 관련되어 있는 경우에 작업이 상호의존적이라고 정의하고, 해당작업에 의한 상태변화는 상호연관성을 갖는 하나이상의 작업에 대한 관측상태정보에 의해서 추정될 수 있다.

각각의 개별작업, t_1, t_2, \dots, t_n 에 대한 종료시한 d_1, d_2, \dots, d_n 은 앞서 제시한 단일작업을 수행하는 선형시불변 시스템에 대한 종료시한 유도 알고리즘을 이용함으로써 계산하고 이러한 개별 종료시한을 각 작업수행방식에 따라서 본 논문에서 제안할 방식에 의해 조합함으로써 최종적으로 전체시스템의 종료시한을 유도한다. 이를 위해 작업수행방식을 다음과 같은 특징을 갖는 세 가지 형태로 분류한다.

- 병렬수행(parallel mode): 중요도가 낮지만, 반드시 종료시한 이내에 수행되어야 하는 작업 하나를 포함하면서 다수의 작업을 동시에 수행하는 방식으로서 적어도 하나의 작업이 자신의 종료시한을 만족하는 경우에 전체 시스템은 안정도를 유지하게 된다. 앞서 구한 각 작업의 종료시한 d_1, d_2, \dots, d_n 을 이용하여 전체시스템의 종료시한을 $\max(d_1, d_2, \dots, d_n)$ 의 형태로 얻을 수 있고, 이러한 작업들의 연관관계는 논리적으로 OR수행으로 간주한다. H/W면에서는 그 비용면에서 효과가 작고, 반면 S/W의 측면에서 프로그램수행의 신뢰도 및 검증을 위해서 병렬연결수행을 고려하기도 한다.

- 일련수행(series mode): 모든 작업은 동시에 수행되어지며 각각의 작업수행결과가 시스템을 안정하게 유지하기 위해서 모두 중요한 경우에 일련수행방식으로 다룬다. 하나의 작업이라도 일시적인 고장에 의해서 자신의 종료시한내에 수행되지 못하는 경우에 전체시스템은 안정성을 잃게 되며 그 영향이 치명적인 경우에는 시스템 전체가 다운되는 현상이 발생한다. 다중작업 t_1, t_2, \dots, t_n

의 각 종료시한이 d_1, d_2, \dots, d_n 일 때 전체시스템의 종료시한은 $\min(d_1, d_2, \dots, d_n)$ 으로 얻을 수 있으며, 논리적으로 AND수행으로 간주할 수 있다.

- 순차수행(cascade mode): 전체적으로 직전의 작업 결과가 다음작업의 입력으로 연결되는 구조로 한 작업의 실패는 그 이후의 작업수행을 진행시키지 못함으로써 이후의 모든 작업이 지연효과를 갖게된다. 대부분의 생산시스템이 이러한 수행작업방식을 갖고 있으며, 이 경우에는 개개의 작업 종료시한도 만족시켜야 할뿐만 아니라 전체시스템에서 요구되는 종료시한도 존재할 것이며 이 또한 만족시켜야 한다. 전체 시스템의 종료시한은 $\sum_{i=1}^n d_i$ 와 같은 단순합의 형태로 얻을 수 있으며, 개별 작업공정의 종료시한이 더 중요하며 전체 시스템 종료시한은 소프트(soft) 종료시한시스템에서 성능향상을 위해서 사용된다.

따라서, 각각의 종료시한 d_1, d_2, \dots, d_n 을 얻기 위해서는 (1)에서 각 제어기 고장으로 인해 작업수행의 영향을 받는 상태변수 x_1, x_2, \dots, x_n 중에서 하나의 작업수행 지연현상을 그 작업의 영향을 받는 상태에 대한 지연현상으로 해석함으로써, 이러한 지연현상이 다른 작업과의 연관성에 기반하여 시스템의 안정도에 미치는 영향을 분석해야 한다. 만약 모든 작업이 상호연관성을 갖지 않는 단순한 경우에는, 각 작업에 대한 수정상태방정식은 각각의 동적특성을 표현하는 독립된 누적상태벡터로써[5] (1)과 같이 각 작업특성을 나타내는 파라미터로써 유도될 수 있다.

$$\begin{aligned}
 x_1((m+1)N_1) &= A_1^{N_1} x_1(mN_1) + \sum_{j=N_1-i_1}^{N_1-1} A_1^j B_1 u_1(mN_1) \\
 &\quad + \sum_{j=0}^{N_1-i_1-1} A_1^j B_1 u_1(mN_1 + N_1 - j - 1) \\
 &\quad \vdots \\
 x_n((m+1)N_n) &= A_n^{N_n} x_n(mN_n) + \sum_{j=N_n-i_n}^{N_n-1} A_n^j B_n u_n(mN_n) \\
 &\quad + \sum_{j=0}^{N_n-i_n-1} A_n^j B_n u_n(mN_n + N_n - j - 1)
 \end{aligned} \tag{1}$$

III. 수정상태방정식 유도

시스템에서 수행하는 작업의 수가 증가할수록 누적통합수정상태방정식을 모든 경우에 대해서 유도하는 것이 계산적으로 불가능하므로(infeasible), 우선 가장 상호연관성이 깊은 두 작업간의 연관성만을 고려하고 다른 작업들은 모두 상호 비의존적·독립적으로 수행한다고 가정함으로써 단순화된 다중작업시스템에 대한 누적통합수정상태방정식을 유도한다.

3.1 동일주기에서 상대적 상관관계가 가장 큰 두 개의 작업에 대한 선형시불변시스템의 수정상태방정식 유도

다중작업을 수행하는 선형시불변시스템에서 모든 작업이 상호연관성을 갖고 있을 경우에 완전한 수정누적상태방정식을 유도하는 것은 계산량의 증가로 인해 현실적으로 불가능하다. 그러므로, 상대적으로 가장 큰 상호연관

성을 갖는 두 작업만을 서로 의존적 특성을 갖는 것으로 간주하고, 나머지 작업들은 서로 독립적으로 수행되는 것으로 간주한다. 두 작업 t_1, t_2 가 동일한 샘플링 주기를 갖고 수행되면, 각 작업에 영향을 받는 상태변수 x_1, x_2 는 시스템의 동적특성 파라미터 $A_{11}, A_{12}, A_{21}, A_{22}$, 및 B_1, B_2 를 이용하여 (2)와 같이 수정된 간략한 형태로 상태방정식을 얻는다.

$$\begin{bmatrix} x_1(mN+1) \\ x_2(mN+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(mN) \\ x_2(mN) \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1(mN) \\ u_2(mN) \end{bmatrix} \quad (2)$$

(2)에서 제어기의 고장으로 첫 번째 작업 t_1 이 올바른게 수행되지 않는 경우에 시간지연효과를 갖는 수정누적 상태방정식은 (3)과 같이 유도된다.

$$\begin{aligned} \begin{bmatrix} x_1(mN+i) \\ x_2(mN+i) \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^i \begin{bmatrix} x_1(mN) \\ x_2(mN) \end{bmatrix} \\ &+ \sum_{j=0}^{i-1} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^j \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1(mN+i-1-j) \\ u_2(mN+i-1-j) \end{bmatrix} \\ &\vdots \\ \begin{bmatrix} x_1(mN+N) \\ x_2(mN+N) \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^N \begin{bmatrix} x_1(mN) \\ x_2(mN) \end{bmatrix} \\ &+ \sum_{j=N-i}^{N-1} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^j \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1(mN+N-1-j) \\ u_2(mN+N-1-j) \end{bmatrix} \\ &+ \sum_{j=0}^{N-i-1} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^j \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1(mN+N-1-j) \\ u_2(mN+N-1-j) \end{bmatrix} \end{aligned} \quad (3)$$

(3)에서 표현상 간략화를 위해 $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^k$ 의 계산결과를 $\begin{bmatrix} a_{11}(k) & a_{12}(k) \\ a_{21}(k) & a_{22}(k) \end{bmatrix}$ 로 정의하면, 작업 t_1 에서 고장이 발생했을 경우에 각 작업에 대한 수정상태방정식은 (4)과 (5)로 얻을 수 있다.

(4)와 (5)에서 상호의존성을 갖는 두 작업은 자체작업의 고장뿐만 아니라 다른 작업의 고장으로 인한 시간지연에 의해서도 영향을 받게된다. (4)와 (5)를 통합하는 누적상태방정식을 표현하면 (6)과 (7)의

$$\begin{aligned} x_1(mN+N) &= a_{11}(N)x_1(mN) + a_{12}(N)x_2(mN) \\ &+ \sum_{j=N-i}^{N-1} [a_{11}(j)B_1u_1(mN) + a_{12}(j)B_2u_2(mN) \\ &+ N-1-j] + \sum_{j=0}^{N-i-1} [a_{11}(j)B_1u_1(mN+N-1-j) \\ &+ a_{12}(j)B_2u_2(mN+N-1-j)] \\ x_2(mN+N) &= a_{21}(N)x_1(mN) + a_{22}(N)x_2(mN) \\ &+ \sum_{j=N-i}^{N-1} [a_{21}(j)B_1u_1(mN) + a_{22}(j)B_2u_2(mN) \\ &+ N-1-j] + \sum_{j=0}^{N-i-1} [a_{21}(j)B_1u_1(mN+N-1-j) \\ &+ a_{22}(j)B_2u_2(mN+N-1-j)] \end{aligned} \quad (4)$$

식을 얻을 수 있고, 각각의 계수행렬 $a_{11}, a_{12}, a_{21}, a_{22}$ 와 $\beta_{11}, \beta_{12}, \beta_{21}, \beta_{22}$ 은 (8)의 형태이다.

$$\begin{aligned} X_1(m+1) &= a_{11}X_1(m) + a_{12}X_2(m) \\ &+ \beta_{11}X_1(m+1) + \beta_{12}X_2(m+1) \end{aligned} \quad (6)$$

$$\begin{aligned} X_2(m+1) &= a_{21}X_1(m) + a_{22}X_2(m) \\ &+ \beta_{21}X_1(m+1) + \beta_{22}X_2(m+1) \end{aligned} \quad (7)$$

$$\alpha_{mn} = \begin{bmatrix} 0 & \dots & 0 & a_{mn} - B_1F_{1n} - B_2F_{2n} \\ 0 & \dots & 0 & a_{mn}(2) - a_{m1}B_1F_{1n} - B_1F_{1n} - a_{m2}B_2F_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & a_{mn} - \sum_{j=0}^{i-1} a_{m1}(j)B_1F_{1n} - a_{m2}(j-1)B_2F_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_{mn} - \sum_{j=N-i}^{N-1} a_{m1}(j)B_1F_{1n} - a_{m2}(N-1)B_2F_{2n} \end{bmatrix}$$

$$\beta_{mn} = \begin{bmatrix} 0 & 0 & \dots & \dots & \dots & 0 \\ -B_2F_{2n} & 0 & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ r_1 & \dots & r_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ r_3 & \dots & r_4 & \dots & r_5 & 0 \end{bmatrix} \quad (8)$$

여기서, $r_1 = -a_{m2}(i-1)B_2F_{2n}$, $r_2 = -B_2F_{2n} - B_1F_{1n}$, $r_3 = -a_{m2}(N-2)B_2F_{2n} - a_{m1}(N-1-i-\delta(i))$, $r_4 = -a_{m2}(N-1-i)B_2F_{2n}$, $r_5 = -B_2F_{2n} - B_1F_{1n}$ 이다. 이 때, 수정누적상태방정식은 (9)로 표현된다.

$$\overline{X}(m+1) = \overline{\Gamma}\overline{X}(m) + \overline{\Omega}\overline{X}(m+1) \quad (9)$$

이에 대한 특성방정식은 (10)과 같다.

$$\det[(I - \overline{\Omega})z^N - \overline{\Gamma}] = 0 \quad (10)$$

만일, (10)에서 고장이 매주기 이진분포에 의해 발생된다고 가정하면, q_0, q_1, \dots, q_N 은 0, T_s, \dots, NT_s 의 지연시간의 확률로 표현된다. 그림 1은 한 예로서, $N=3$ 일 때, 고장발생확률을 나타내고 있다.

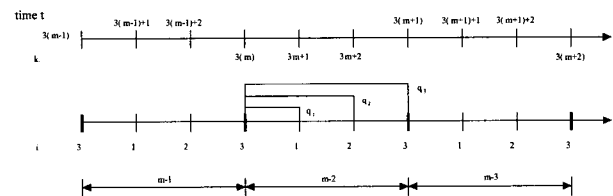


그림 1. 컴퓨터 고장의 stationary 발생을 보여주기 위해 새로 정의되었진 벡터의 시간 지표.

Fig. 1. Time index for newly-defined augmented vectors representing the stationary occurrence of computer failures(delays) when $N=3$.

이 지연시간 분포는 외부환경특성에 대한 분석(즉, field data) 및 통계기법(fitting data and estimation)으로부터 실제적으로 얻어질 수 있다. 그러므로, 고장확률변수 q_i 를 (10)에 적용하면 (11)과 같이 변형된다.

$$\det[(I - \sum_{i=0}^N q_i \overline{\Omega})z^N - \sum_{i=0}^N q_i \overline{\Gamma}] = 0 \quad (11)$$

(11)을 이용하여 t_1 작업의 고장에 의한 시스템 고유치의 궤적변화를 회귀론적(recursive)기법을 적용하여 검사할 수 있다.

3.2 다른주기에서 상대적 상관관계가 가장 큰 두 개의 작업에 대한 선형시불변시스템의 수정상태방정식 유도

앞서 제시한 방법과 동일한 방법으로 두 작업이 서로 다른 주기에서 수행되는(다중샘플링) 경우에 두 작업을 수행하는 시스템의 상태방정식을 유사한 방법으로 유도

한다. 두 작업 t_1, t_2 가 다른 주기 T_1, T_2 를 갖고 주기수 N_1, N_2 를 갖는다. T_1 과 T_2 는 서로 일정한 정수배를 가질 때, $T_1 = \alpha T_2$ (α 는 정수)의 관계를 설정할 수 있다. 이 때 주기수 $N_2 = \alpha N_1 = N$ (이때 N 은 큰 주기수 값을 선택한다)의 관계를 가질 때, 다중샘플링 다중작업 시스템의 상태방정식을 유도한다. t_1 의 주기가 t_2 보다 더 길기 때문에, $(\alpha-1)T_2$ 의 시간까지 t_1 은 제어입력신호를 발생시키지 않게 된다. t_2 의 고장이 발생했을 때 고장 지연주기수가 α 보다 큰 일반화된 수정상태방정식을 유도하기 위한 경우로 즉, $p\alpha < i < (p+1)\alpha$ (p 는 정수)일 때, 새로운 파라미터 i' 를 이용하여 $i = p\alpha + i'$ 로 표현하면 (12)를 유도할 수 있다.

$$\begin{aligned} \begin{bmatrix} x_1(mN+N) \\ x_2(mN+N) \end{bmatrix} &= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \right)^{\alpha-1} \begin{bmatrix} x_1(mN) \\ x_2(mN) \end{bmatrix} \\ &+ \sum_{j=p}^{N_1-1} \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \right)^j \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \\ &\begin{bmatrix} u_1(mN+(p-j)\alpha-1) \\ u_2(mN) \end{bmatrix} + \sum_{k=p}^{N_1-1} \sum_{j=0}^{k-1} \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \right)^j \\ &\begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \right)^k \\ &\begin{bmatrix} 0 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1(mN+(p-k)\alpha-2-j) \\ u_2(mN) \end{bmatrix} \\ &+ \sum_{j=\alpha-r-1}^{\alpha-2} \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \right)^{N_1-p} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \\ &\begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1(mN+(p+1-k)\alpha-2-j) \\ u_2(mN) \end{bmatrix} \\ &+ \sum_{j=0}^{\alpha-2} \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \right)^{N_1-p} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \\ &\begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1(mN+(p+1-k)\alpha-2-j) \\ u_2(mN+(p+1-k)\alpha-2-j) \end{bmatrix} \\ &+ \sum_{j=0}^{N_1-p-1} \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \right)^j \\ &\begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1(mN+(N_1-k)\alpha-1) \\ u_2(mN+(N_1-k)\alpha-1) \end{bmatrix} \\ &+ \sum_{k=0}^{N_1-k-1} \sum_{j=0}^{k-1} \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \right)^k \\ &\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1(mN+(N_1-k)\alpha-2-j) \\ u_2(mN+(N_1-k)\alpha-2-j) \end{bmatrix} \quad (12) \end{aligned}$$

단일주기에서 수행되는 두 작업에 대한 수정누적상태 방정식 및 특성방정식과 유사하게 (12)의 경우에도 제어 기고장주기동안 제어입력의 갱신이 지연되는 현상을 상태방정식에 공식화하는 과정을 적용하여 시간지연효과를 갖는 특성방정식을 유도함으로써 시스템의 안정도 분석에 이용하며, 이러한 시스템의 안정도를 깨뜨리는 최소의 정수 N 을 해당작업의 종료시한으로 정의하여 이 값을 수치적으로 유도할 수 있다.

IV. 다중작업 실시간시스템의 전체시스템 종료시한 유도 알고리즘 개발

이 장에서는 각 작업수행방식을 고려하여 효율적으로 전체시스템종료시한을 유도하기 위한 알고리즘을 단계별로 제시한다. 제시되는 알고리즘은 계산량을 줄이고 효

율적인 종료시한유도를 위해서 상당히 유용할 것이다. 작업 t_1, \dots, t_n 이 각각 주기 T_1, \dots, T_n 을 갖고, $T_1/\alpha_1 = T_2/\alpha_2 = \dots = T_n/\alpha_n$ 의 관계를 가지면, 주기에 의해 추출되는 표본의 최대개수가 N_1, \dots, N_n 일 때 $\alpha_1 N_1 = \alpha_2 N_2 = \dots = \alpha_n N_n$ (α_i 는 정수)의 관계식을 만족하고, 다중작업을 수행하는 시스템종료시한은 다음과 같은 과정으로 유도할 수 있다. 개별 작업의 종료시한은 d_1, \dots, d_n 이고 전체시스템의 종료시한은 d_{plant} 이다.

단계 1: 특성방정식을 유도한다.

단계 2: 작업수행모드(task operation mode)에 따라서 일련/병렬/순차수행방식으로 분류한다.

단계 3: 첫 번째 작업에 대해서 순차적회귀기법을 이용해서 종료시한 d_1 을 유도한다.

$$d_{plant} = d_1 = i_1 T_1$$

단계 4: 단계3에서 얻은 첫 번째 작업에 대한 가상최대지연시간요소인 i_1 를 이용하여 다음작업의 최대지연시간요소인 i_2 을 작업수행방식에 따라서 다음과 같이 구한다.

- 일련수행방식: $i_{k+1} = \left\lfloor \frac{\alpha_k}{\alpha_{k+1}} \right\rfloor i_k$
- 병렬수행방식: $i_{k+1} = \left\lceil \frac{\alpha_k}{\alpha_{k+1}} \right\rceil i_k$
- 순차수행방식: $i_{k+1} = \left\lceil \frac{\alpha_k}{\alpha_{k+1}} \right\rceil i_k$

단계 5: i_{k+1} 동안 제어입력신호가 갱신되지 않을 때 시스템의 안정도를 조사한다.

단계 6: d_{plant} 를 작업수행방식에 따라서 갱신한다.

• 일련수행방식: 시스템의 고유치가 1보다 작으면 $d_{plant} = d_k$ 가 되고 단계 7로 진행하고, 그렇지 않으면 i_{k+1} 을 감소시키고 단계 5로 복귀한다.

• 병렬수행방식: 시스템의 고유치가 1보다 크면 $d_{plant} = d_k$ 가 되고 단계 7로 진행하고, 그렇지 않으면 i_{k+1} 을 증가시키고 단계 5로 복귀한다.

• 순차수행방식: 시스템의 고유치가 1보다 크면 $d_{plant} = d_k$ 가 되고 단계 7로 진행하고, 그렇지 않으면 i_{k+1} 을 증가시키고 단계 5로 복귀한다.

단계 7: k 를 N 과 같을 때까지 하나씩 증가시킨다. k 가 N 보다 작으면 단계 4로 복귀하고, 그렇지 않으면 단계 8로 진행한다.

단계 8: 작업수행방식에 따라서 다음과 같이 전체 시스템의 종료시한을 유도할 수 있다.

- 일련수행방식: $d_{plant} = d_k = i_k T_k$
- 병렬수행방식: $d_{plant} = d_k = i_k T_k$
- 순차수행방식: $d_k = (i_{k+1} - i_k) T_k$

그림 2는 다중작업수행시스템 중 일련(series)수행방식의 종료시한 유도하기 위한 알고리즘을 간단히 도식화하였다. 그림 2에서 n 개의 작업이 있을 때 각 작업 $t_1, t_2, t_3, \dots, t_n$ 에 대한 실제적인 종료시한이 $d_1, d_2, d_3, \dots, d_n$ 이라고 정의된다. 작업 t_2 와 t_3 가 서로 상호연관성을 갖

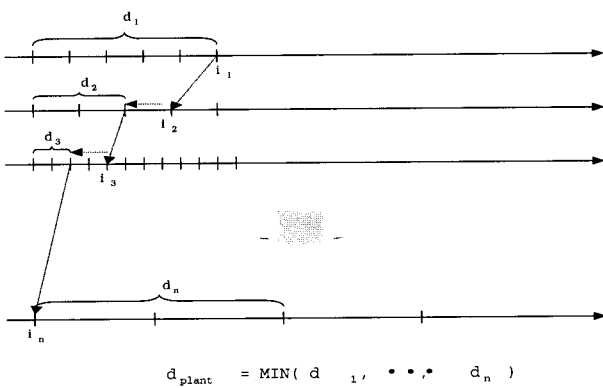


그림 2. 일련수행방식인 다중작업을 수행하는 전체 시스템의 종료시한 유도.
 Fig. 2. Derivation of the deadline for the entire system performing multitasks as series operation mode.

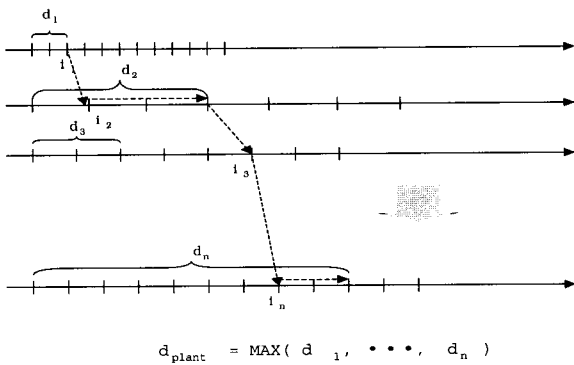


그림 3. 병렬수행방식인 다중작업을 수행하는 시스템의 종료시한 유도방법.
 Fig. 3. Derivation of the deadline for the entire system performing multitasks as parallel operation mode.

고 있으며 나머지 작업들은 모두 독립적으로 수행된다고 가정한다. 이 때, t_2 와 t_3 를 포함하는 수정누적상태방정식과 특성방정식을 유도하고, 나머지 작업들에 대한 각각의 개별특성방정식을 유도한다. 작업 t_1 에 대해서 순차적회귀기법을 이용하여 시스템을 안정화시키는 최대 지연주기수인 i_1 을 구하고, 이를 이용하여 작업 t_1 의 작업 종료시한 $d_1 = i_1 T_1$ 을 얻는다. 제시된 알고리즘상에서 i_1 을 이용하여 두 번째 작업에 대해 i_2 를 구하고, 작업 t_2 가 i_2 동안 시간지연효과를 가질 때 t_2 에 대한 특성방정식을 이용하여 시스템의 안정도를 검사한다. 일련수행방식에서 시스템의 전체종료시한은 각 작업 종료시한의 최소값으로 유도되므로, t_2 가 i_2 동안 지연되었을 때 시스템이 안정하게 유지되면 전체종료시한은 d_1 이 되고, 반대의 경우에는 전체종료시한은 d_2 로 변경하게 된다. 이런 과정을 반복적으로 수행함으로써 최소 종료시한 값을 구함으로써 효율적으로 전체종료시한을 유도할 수 있다. 그림 3은 모든 작업을 병렬로 수행하는 시스템의 경우에 종료시한을 유도하는 방법을 나타내고 있다.

경우에 따라서는 단일 시스템내에서 일련수행, 병렬수행, 그리고 순차수행의 방식으로 작업들이 혼합되어(hybrid) 있을 수 있다. 선행조건을 갖는 순차적수행방식에서는 작업간의 연관성이 없다고 가정하고, 순차적수행 작업들을 모듈화[6]한 후 연속적연결형태로 구성한다. 이 때, 일련수행 및 병렬수행을 하는 작업들은 하나의 모듈에 포함되어 동시에 수행될 것이다. 이렇게 되는 경우에 각 모듈종료시한 값을 구하고, 순차수행방식에서 전체종료시한은 각 모듈에서 얻은 종료시한 값들의 합으로 얻는다.

V. 수치예제

예제 1: 두 개의 작업 t_1 과 t_2 가 같은 샘플링 주기에 의해 상호연관성을 갖고며 일련수행(series)의 모드로 수행되는 선형시불변 시스템을 고려한다. 시스템의 동적특성은 다음과 같은 상태방정식에 의해 지배되는 두 작업의 상태레직에 의해 표현된다.

$$\begin{aligned} x_1(k+1) &= 1.2 x_1(k) + 0.3 x_2(k) + 1.1 u_1(k) \\ x_2(k+1) &= 0.1 x_1(k) + 1.4 x_2(k) + 1.4 u_2(k) \end{aligned}$$

제어목표는 $Q=2$ 이며, $R=5$ 를 갖는 이차성능지수의 최소화에 있다면, 최적의 제한제어입력을 LQR기법을 사용하여 아래와 같이 구할 수 있다.

$$K = \begin{bmatrix} 0.6081 & 0.2095 \\ 0.1129 & 0.7028 \end{bmatrix}$$

제어기조장이 있는 경우 worst case(고장의 지속시간이 항상 NT_s 인 경우)를 가정하여 극점위치를 표 1에서 나타내었다. 최초로 극점이 단위원을 벗어나는 경우인 $D=3T_s$ 가 이 시스템의 종료시한임을 쉽게 판별할 수 있다.

표 1. $P=q_n=1$ 인 경우의 pole의 위치와 N 과의 관계.

Table 1. The relationship between the pole positions and N when $P=q_n=1$.

N		2	3	4	5	6	7
when t_1 is failed	$ \lambda_{max} $ for t_1	0.6156	0.5157	1.0156	1.1538	2.3388	3.0666
	$ \lambda_{max} $ for t_2	0.3126	0.1535	0.1347	0.1458	0.0889	0.0759
when t_2 is failed	$ \lambda_{max} $ for t_1	0.0456	1.0048	1.8426	2.4836	3.7704	4.8606
	$ \lambda_{max} $ for t_2	0.5297	0.3545	0.3001	0.5548	0.2455	0.2287

극점의 위치는 최대지연뿐만 아니라, 지연 확률분포함수에도 영향을 받는데 $a_1 = 1 - P_1$, $a_2 = P_1(1 - P_2)$, $a_3 = P_1P_2$.일 때 $P_1 = 0.9$ 이고 P_2 가 0.1에서 0.9로 변화하는 경우의 시스템의 고유치는 아래의 표 2에 제시되어 있다.

표 2. $N=3$ 일 때 pole와 시간지연 확률분포와의 관계.

Table 2. The relationship between the pdf of failure durations and the eigenvalues when $N=3$.

P_2	0.1	0.2	0.3	0.4
$ \lambda_{max} $	0.4815	0.5641	0.6842	0.7334
0.5	0.6	0.7	0.8	0.9
0.8194	0.9061	0.9634	0.9911	1.0026

VI. 결론

본 논문에서는 시스템의 동적특성 및 외부환경변수에 의한 확률적 고장발생요인을 고려하여 순차적회귀기법을 통한 시스템의 종료시한을 관련파라미터로써 수치적으로 유도하는 알고리즘을 확장하여 다중샘플링

다중작업을 수행하는 시스템에 대한 종료시한을 유도하는 효율적이면서도 확장된 알고리즘을 제안하였다. 제시된 전체시스템 종료시한 유도 알고리즘은 효과적으로 적용가능하며, 실시간 제어기 구성 및 종료시한을 기반으로 RM 및 EDF와 같은 작업예정계획을 수립하는데, 효율적으로 활용할 수 있다.



이 대 현

1974년 7월 21일생. 1997년 연세대 전기공학과 졸업. 현재 동대학원 기계전자공학부 전기전공 석사과정. 관심분야는 실시간제어, 종료시한 유도, 실시간생산시스템, 신뢰도 평가.

참고문헌

[1] K. Shin and Y. Lee, "Error detection process - model, design, and its impact on computer performance," *IEEE Trans. on Computers*, vol. C-33, no. 6, pp. 529-539, June, 1984.

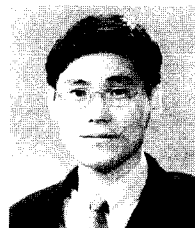
[2] K. Shin and X. Cui, "Effects of computing time delay on real-time control systems," *Proc. of 1988 American Control Conf.*, pp. 1071-1076, June, 1988.

[3] S. McConnel, D. Siewiorek, and M. Tsao, "The measurement and analysis of transient errors in digital computer systems," *Digest of Papers, FTCS-9*, pp. 67-70, June, 1979.

[4] H. Kim and K. Shin, "On the maximum feedback delay in a linear/nonlinear control system with input disturbances caused by controller-computer failures," *IEEE Trans. on Control Systems Technology*, vol. 2, no. 2, pp. 110-122, June, 1994.

[5] K. Shin and H. Kim, "Derivation and application of hard deadlines for real-time control systems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1403-1413, November, 1992.

[6] D. Peng and K. Shin, "Modeling of concurrent task execution in a distributed system for real-time control," *IEEE Trans. on Computers*, vol. c-36, no. 2, pp. 500-516, April, 1987.



김 학 배

1965년 10월 10일생. 1988년 서울대 전자공학과 졸업. 1990년 미국 미시간대 대학원 전기공학과(E ECS)졸업(석사). 1994년 8월 동 대학원 졸업(공학박사). 1994년 9월-1996년 8월 미국 National Research Council (NRC) Rearch Associate at NASA Langley Research Center. 1996년 9월-현재 연세대학교 기계전자공학부 전기전공 조교수. 관심분야는 실시간제어, 자동화공학, 고장허용기법, 신뢰도 평가.