

Continued Fraction Expansion을 이용한 Dead Time 근사의 새로운 접근

조원휘 · 이지태[†]

엔깁스
445-160 경기도 화성시 반송동 93-10
*경북대학교 화학공학과
702-701 대구시 북구 산격동 1370
(2012년 5월 14일 접수, 2012년 8월 7일 채택)

New Approach Using the Continued Fraction Expansion for the Dead Time Approximation

Wonhui Cho and Jietae Lee[†]

enGibbs, 93-10 Bansong-dong, Hwasung, Gyeonggi 445-160, Korea
**Department of Chemical Engineering, Kyungpook National University, 1370 Sangyuk-dong, Buk-gu, Daegu 702-701, Korea*
(Received 14 May 2012; accepted 7 August 2012)

요 약

Dead time은 공정의 동특성을 기술할 때 매우 자주 나타나는 것으로 공정의 동특성 모사 혹은 제어 시스템 분석에 많은 어려움을 준다. 이 어려움을 줄이기 위해 무한 차원의 dead time을 유한 차원의 전달함수로의 근사가 필요한데, 여기에는 Pade 근사가 자주 사용된다. Dead time의 정밀한 근사를 위해서는 고차의 Pade 근사가 필요한데, 고차의 Pade 근사식은 외우기 쉽지 않고 수치적으로 안정적이지 못하다. 이 Pade 근사와 같은 전달함수를 주지만 수치적으로 우수한 continued fraction 전개를 이용하는 방법을 제안하고자 한다. 제안하는 방법은 수치적으로 우수할 뿐만 아니라 매우 체계적이어서 쉽게 기억할 수 있어 공정제어 강의와 계산에 편리하게 이용할 수 있을 것이다.

Abstract – Dead times appear often in describing process dynamics and raise some difficulties in simulating process dynamics or analyzing process control systems. To relieve these difficulties, it is needed to approximate the infinite dimensional dead time by the finite dimensional transfer function and, for this, the Pade approximation method is often used. For the accurate approximation of the dead time, high order Pade approximation is needed and the high order Pade approximation is not easy to memorize and is not stable numerically. We propose a method based on the continued fraction expansion that provides the same transfer functions. The method is excellent numerically as well as systematic to be memorized easily. It can be used conveniently for the process control lecture and computations.

Key words: Dead Time, Pade Approximation, Continued Fraction, Numerically Stable State Equation

1. 서 론

Dead time은 공정의 동특성을 기술하는데 매우 자주 나타난다 [10]. 파이프를 통한 유체 이송에 따른 실질적으로 존재하는 dead time도 있으며, 고차의 동특성을 근사하는 과정에서 나타나기도 한다. 이 dead time은 제어에 매우 중요하기 때문에 이를 정확히 얻기 위한 실험적 방법이 여러 제안되어 있으며 [11,12], 또한 공정의 동특성의 모사 혹은 제어시스템 설계 및 분석에 어려움을 주기 때문에 많은 이론적 연구가 있어 왔다 [9]. 이런 분석의 어려움을 줄이기 위해 무한 차원의 dead time을 유한 차원의 전달함수로 근사하여 사용하기도 한다. 이 근사에는 수치적 방법과 해석적 방법으로 나눌 수 있

다 [9]. 수치적 방법으로 주어진 주파수 영역에서 위상각 오차를 최소화하는 방법이 제안되어 ultimate period 정보를 해석적으로 찾는 데 이용되었다 [7]. 해석적 방법으로는 작은 시간상수의 일차함수 여럿이 연결되어 있는 것으로 하는 방법과 Pade 근사에 의한 방법이 소개되어 있다 [3,5,10]. 앞의 방법은 근사 수렴 속도가 느려 뒤의 Pade 근사가 주로 사용되고 있다. Pade 근사는 dead time과 이것의 근사 전달함수 둘의 Taylor 급수 전개가 주어진 차수까지 같게 되도록 하는 근사 방법이다. Taylor 급수에서 낮은 차수의 항들이 같으므로 낮은 주파수에서 잘 맞는 근사가 얻어진다. 높은 주파수 영역에서는 근사 정도가 떨어지는데 계단응답에서 급격한 변화가 있는 시간지연 근처의 응답에서 많은 차이를 보이는 것에서 이를 알 수 있다. 정밀한 근사를 위해서는 높은 차수의 Pade 근사가 필요한데 이를 위한 식이 이미 개발되어 있다 [5]. 그러나 이것의 유도는 고차의 선형방정식을 풀어야 하기 때문에 손으로 계산하기에는 어려움이 있어 공정

[†]To whom correspondence should be addressed.
E-mail: jtleee@knu.ac.kr

[‡]이 논문은 KAIST 박선원 교수님의 정년을 기념하여 투고되었습니다.

제어 강의 등에 사용하기 부적합하며, 또한 이 근사식은 다항식의 분수 형태로 수치적으로 안정적이지 못하다.

여기서는 Pade 근사와 같은 결과를 주지만 유도 과정이 쉽고, 변수 값들이 체계적이어서 외우기 쉬우며, 수치적으로 안정한 continued fraction 전개 방법을 이용할 것을 제안한다. Continued fraction 전개 방법은 1700년대 Euler 및 Lambert 등에 의해 체계화된 전개 방법으로 원주율 등이 무리수인 것의 증명에 사용되기도 하였다[2]. 동적 시스템 분석 등에 다양한 쓰임새가 있어 한동안 많은 연구가 있었으나[4,8] 현재 널리 이용되고 있지는 못하다. Exponential 함수에 관한 continued fraction 전개는 이미 잘 알려져 있으며, 이는 바로 dead time의 근사이지만 이를 이용한 예는 아직 없는 것으로 조사되고 있다. 이 continued fraction 전개 방법을 소개하고, 공정시스템에의 몇몇 응용을 밝히고자 한다.

2. Dead time의 기존의 해석적 근사 방법들

단일 입력 $U(s)$ 와 출력 $Y(s)$ 를 갖는 선형시스템 $Y(s) = G(s)U(s)$ 에서 다음의 전달함수로 표현되는 dead time의 근사를 다룬다.

$$G(s) = e^{-s} \quad (1)$$

이 전달함수는 변수 s 의 scaling을 통하여 일반적인 dead time 전달 함수 $G(s) = e^{-ks}$ 에 관한 식으로 전환될 수 있다.

식 (1)은 e 의 정의에 따라 다음과 같이 근사할 수 있다[10].

$$G(s) = \lim_{k \rightarrow \infty} \left(\frac{1}{1 + \frac{s}{k}} \right)^k \approx \left(\frac{1}{1 + \frac{s}{n}} \right)^n \quad (2)$$

이 근사는 같은 시간상수의 일차시스템이 여럿 연결된 것으로 간단하지만 근사 수렴이 늦어 제한된 용도모만 사용되고 있다.

수렴이 빠른 근사를 위하여 Pade 근사가 널리 사용되고 있다. 이는 Taylor 전개에 근거를 두고 있다.

$$G(s) = e^{-s} = 1 - s + \frac{1}{2!}s^2 - \frac{1}{3!}s^3 + \frac{1}{5!}s^4 - \dots \quad (3)$$

이것과 Taylor 전개가 주어진 변수 개수 만큼의 항까지 같게 되는 다음 형태의 전달함수를 구한다.

$$P_{[m,n]}(s) = \frac{b_0 + b_1s + b_2s^2 + \dots + b_ms^m}{1 + a_1s + a_2s^2 + \dots + a_ns^n} \quad (4)$$

이 근사를 Pade 근사라 부른다[8]. 참고로 $P_{[1,2]}(s)$ 를 구하는 과정을 살펴본다.

$$P_{[1,2]}(s) = \frac{b_0 + b_1s}{1 + a_1s + a_2s^2} \approx 1 - s + \frac{1}{2}s^2 - \frac{1}{6}s^3 + \dots \quad (5)$$

정리하면 다음을 얻을 수 있다.

$$\begin{aligned} b_0 + b_1s &\approx (1 + a_1s + a_2s^2) \left(1 - s + \frac{1}{2}s^2 - \frac{1}{6}s^3 + \dots \right) \\ &= 1 + (-1 + a_1)s + \left(\frac{1}{2} - a_1 + a_2 \right) s^2 + \left(-\frac{1}{6} + \frac{1}{2}a_1 - a_2 \right) s^3 + \dots \end{aligned} \quad (6)$$

따라서

$$\begin{aligned} b_0 &= 1 \\ b_1 &= -1 + a_1 \\ \frac{1}{2} - a_1 + a_2 &= 0 \\ -\frac{1}{6} + \frac{1}{2}a_1 - a_2 &= 0 \end{aligned} \quad (7)$$

이 식들에서 아래 두 식의 2원 연립 방정식을 풀어 a_1 과 a_2 를 구할 수 있고, 이를 위 식에 대입하여 b_0 와 b_1 을 구할 수 있다. 이런 방식으로 Taylor 전개가 가능한 함수를 식 (4)의 다항식의 분수 형태의 함수로 근사할 수 있다.

중요한 exponential 함수의 경우 해석적 식이 이미 알려져 있다[5,9].

$$P_{[m,n]}(s) = \frac{\sum_{k=0}^m \frac{(m+n-k)!m!}{(m+n)!(m-k)!k!} (-s)^k}{\sum_{k=0}^n \frac{(m+n-k)!n!}{(m+n)!(n-k)!k!} (-s)^k} \approx e^{-s} \quad (8)$$

몇몇을 기술하면 Table 1과 같다. 여기서 m 이 n 보다 하나 적거나 같은 것들만 나타내었는데, 이것에는 몇몇 이유가 있다. n 이 전체 시스템의 차수를 결정하고 시스템의 모사 및 분석에서 어려움이 이것에 주로 좌우되기 때문에 여기에 따른 값이 주어진다. 주어진 n 에서 최대한 정밀한 근사를 주는 물리적으로 가능한 최대 크기인 $m=n$ 이

Table 1. Pade approximations of the exponential function, e^{-s}

Orders	Approximations
$P_{[0,1]}(s)$	$\frac{1}{1+s}$
$P_{[1,1]}(s)$	$\frac{2-s}{2+s}$
$P_{[1,2]}(s)$	$\frac{6-2s}{6+4s+s^2}$
$P_{[2,2]}(s)$	$\frac{12-6s+s^2}{6+4s+s^2}$
$P_{[2,3]}(s)$	$\frac{60-24s+3s^2}{60+36s+9s^2+s^3}$
$P_{[3,3]}(s)$	$\frac{120-60s+12s^2-s^3}{120+60s+12s^2+s^3}$
$P_{[3,4]}(s)$	$\frac{840-360s+60s^2-4s^3}{840+480s+120s^2+16s^3+s^4}$
$P_{[4,4]}(s)$	$\frac{1680-840s+180s^2-20s^3+s^4}{1680+840s+180s^2+20s^3+s^4}$
$P_{[4,5]}(s)$	$\frac{15120-6720s+1260s^2-120s^3+5s^4}{15120+8400s+2100s^2+300s^3+25s^4+s^5}$
$P_{[5,5]}(s)$	$\frac{30240-15120s+3360s^2+420s^3+30s^4-s^5}{30240+15120s+3360s^2+420s^3+30s^4+s^5}$
$P_{[5,6]}(s)$	$\frac{332640-151200s+30240s^2-3360s^3+210s^4-6s^5}{332640+181440s+45360s^2+6720s^3+630s^4+36s^5+s^6}$
$P_{[6,6]}(s)$	$\frac{665280-332640s+75600s^2-10080s^3+840s^4-42s^5+s^6}{665280+332640s+75600s^2+10080s^3+840s^4+42s^5+s^6}$

유리하다. 그러나 $m=n-1$ 인 경우가 모사기에서 다소 우수한 결과를 주기도 하기 때문에 이 경우도 간혹 사용되기도 한다.

3. Continued Fraction 전개에 의한 근사

Continued fraction 전개는 여러 모양의 것이 제안되어 있으나, 여기서는 Caer second form[4]으로 알려진 다음과 같은 식을 고려한다.

$$G(s) = \frac{h_0}{h_1 + \frac{s}{h_2 + \frac{s}{h_3 + \frac{s}{h_4 + \dots}}}} \tag{9}$$

이것을 간략하게 다음과 같이 쓰기도 한다.

$$G(s) = \frac{h_0}{h_1 + \frac{s}{h_2 + \frac{s}{h_3 + \frac{s}{h_4 + \dots}}}} \tag{10}$$

먼저 다음의 연속시간 시스템을 이산시간 제어를 위한 이산시간 시스템으로 전환하는 과정에서 나타나는 zero-order hold의 전달함수를 고려한다[6].

$$Q(s) \equiv \frac{1-e^{-s}}{s} = 1 - \frac{1}{2!}s + \frac{1}{3!}s^2 - \frac{1}{4!}s^3 + \dots \tag{11}$$

이 전달함수는 이것의 Taylor 전개로부터 다음과 같은 나누기를 하여 식 (9)와 같은 continued fraction 전개를 얻을 수 있다.

$$Q(s) = 1 - \frac{1}{2!}s + \frac{1}{3!}s^2 - \dots$$

$$= \frac{1}{1 - \frac{1}{2!}s + \frac{1}{3!}s^2 - \dots}} = \frac{1}{1 + \frac{\frac{1}{2}s - \frac{1}{3!}s^2 + \dots}{1 - \frac{1}{2}s + \frac{1}{3!}s^2 - \dots}}} = \frac{1}{1 + \frac{s}{2 - s - \frac{1}{3}s^2 + \dots}}} \tag{12}$$

몇몇 항들을 구하면 계수 사이의 규칙이 얻어지며, 다음을 얻을 수 있다.

$$Q(s) = \frac{1}{1 + \frac{s}{2 - \frac{s}{3 + \frac{s}{2 - \frac{s}{5 + \dots}}}}} \tag{13}$$

식 (12)의 과정은 다소 많은 계산을 필요로 한다. 이를 체계적으로 하는 Routh 안정성 계산 과정과 매우 흡사한 방법이 제안되어 있다 [4](부록 A 참조).

이제 dead time은 다음과 같이 얻어진다[1].

$$G(s) = e^{-s} = 1 - sQ(s) = 1 - s \frac{1}{1 + \frac{s}{2 - \frac{s}{3 + \frac{s}{2 - \frac{s}{5 + \dots}}}}} \tag{14}$$

또한 다음과 같은 전개도 가능하다.

$$G(s) = e^{-s} = \frac{1}{e^s} = \frac{1}{1 + sQ(-s)} = \frac{1}{1 + \frac{s}{1 - \frac{s}{2 + \frac{s}{3 - \frac{s}{2 + \frac{s}{5 - \dots}}}}} \tag{15}$$

여기서는 식 (14)를 주로 이용하는데, 두 번째 항부터 continued fraction 전개를 하여 근사 시스템의 안정성에 약간의 의심이 있다. 다음의 정리가 이를 해결한다.

예비정리 1

$Q_{[n-1,m]}(s)$ 은 식 (13)을 절사하여 다항식의 분수 형태로 정리하였을 때 분자 및 분모 다항식의 차수가 각각 $n-1$ 과 n 이 되는 것을 나타낸다. 그러면 $Q_{[n-1,m]}(s)$ 의 분모 다항식은 $P_{[n,m]}(s)$ 의 분모 다항식과 같다.

증명

Continued fraction 전개의 특성에 따라[2,8]

$$Q(s) - Q_{[n-1,m]}(s) = O(s^{2n+1})$$

여기서 $O(s^{2n+1})$ 는 s 에 관한 차수가 $2n+1$ 보다 크거나 같은 항들로 이루어졌다는 의미이다. 식 (14)에 이를 대입하면

$$e^{-s} - (1 - sQ_{[n-1,m]}(s)) = (1 - sQ(s)) - (1 - sQ_{[n-1,m]}(s)) = Q(s^{2n+2})$$

가 얻어진다. 이것은

$$P_{[n,m]}(s) = (1 - sQ_{[n-1,m]}(s))$$

인 것을 나타내며, 예비정리 1의 결과를 얻을 수 있다.

이 예비정리는 식 (13)의 $Q(s)$ 를 절사하여 dead time의 근사식을 만들더라도 Pade 근사와 같은 시스템 안정성이 보장되는 것을 보여 주고 있다.

Continued fraction 전개는 다양한 응용이 가능하다. 하나는 이 전개를 절사하면 바로 Pade 근사가 되는 것이 밝혀져 있다. 예를 들면, 식 (15)에서 두 개의 s 까지만 유지하는 절사를 하면 다음의 Pade 근사가 얻어진다.

$$e^{-s} = \frac{1}{1 + \frac{s}{1 - \frac{s}{2}}} = \frac{1 - \frac{s}{2}}{1 + \frac{s}{2}} \tag{16}$$

항수를 달리하면 $P_{[n-1,m]}(s)$ 와 $P_{[n,m]}(s)$ 가 얻어진다. 식 (14)와 (15)는 매우 체계적이어서 쉽게 외울 수 있으며, Pade 근사식을 유도하는데 사용할 수 있다. 그러나 continued fraction 전개의 장점은 이 Pade 근사를 체계적으로 하는 데도 있지만 다음의 dead time 시스템을 근사하는 상태방정식에 더 큰 의미가 있다.

4. 상태방정식 형태의 dead time 근사 시스템

Continued fraction 전개 형태로 표현된 전달함수는 이를 Pade 근사 형태로 바꾸지 않고도 상태방정식 형태로 변환하는 방법이 구비되어 있다(부록 B 참조). 이를 먼저 식 (13)에 적용한다. $Q_{[n-1,m]}(s)$ 를 구현하는 행렬($\tilde{A}, \tilde{b}, \tilde{c}$)는 다음과 같이 된다.

$Q_{[n-1,m]}(s)$:

$$\dot{x} = \tilde{A}x + \tilde{b}u$$

$$y = \tilde{c}x$$

$$\tilde{A} = -2 \begin{pmatrix} 1 & -1 & 1 & -1 & \dots & (-1)^{n+1} \\ 1 & 2 & -2 & 2 & \dots & 2(-1)^n \\ 1 & 2 & 3 & -3 & \dots & 3(-1)^{n+1} \\ 1 & 2 & 3 & 4 & \dots & 4(-1)^n \\ \dots & & & & & \\ 1 & 2 & 3 & 4 & \dots & n \end{pmatrix}, \tilde{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

$$\tilde{c} = 2(1 \ -1 \ 1 \ -1 \ \dots \ 1) \tag{17}$$

그러면 $G(s)$ 를 근사하는 상태방정식은 $G(s) = 1-sQ(s)$ 이므로

$$\dot{x} = \tilde{A}x + \tilde{b}u$$

$$y = u - \hat{y} = u - \tilde{c}\hat{x} = u - \tilde{c}\tilde{A}\hat{x} - \tilde{c}\tilde{b}u = -\tilde{c}\tilde{A}\hat{x} + (1 - \tilde{c}\tilde{b})u \tag{18}$$

가 된다.

정리하면, dead time 시스템 $Y(s) = e^{-s}U(s)$ 를 근사하는 n차 시스템으로

$CF_{[n,m]}(s)$:

$$\dot{x} = Ax + bu$$

$$y = cx + du$$

$$A = \tilde{A} = -2 \begin{pmatrix} 1 & -1 & 1 & -1 & \dots & (-1)^{n+1} \\ 1 & 2 & -2 & 2 & \dots & 2(-1)^n \\ 1 & 2 & 3 & -3 & \dots & 3(-1)^{n+1} \\ 1 & 2 & 3 & 4 & \dots & 4(-1)^n \\ \dots & & & & & \\ 1 & 2 & 3 & 4 & \dots & n \end{pmatrix}, b = \tilde{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

$$c = -\tilde{c}\tilde{A} = \begin{cases} (4 \ -4 \ 24 \ \dots \ 2n(n+1)) & n = \text{odd} \\ (0 \ -12 \ 12 \ -40 \ \dots \ -2n(n+1)) & n = \text{even} \end{cases}$$

$$d = 1 + \tilde{c}\tilde{b} = \begin{cases} -1 & n = \text{odd} \\ 1 & n = \text{even} \end{cases} \tag{19}$$

가 얻어진다.

식 (15)에 부록 B의 방법을 적용하여 바로 $P_{[n-1,m]}(s)$ 와 같은 시스템이 되는 상태방정식 시스템을 얻을 수 있다.

$CF_{[n-1,m]}(s)$:

$$\dot{x} = Ax + bu$$

$$y = cx$$

$$A = - \begin{pmatrix} 1 & -3 & 5 & -7 & \dots & (2n-1)(-1)^{n+1} \\ 1 & 3 & -5 & 7 & \dots & (2n-1)(-1)^n \\ 1 & 3 & 5 & -7 & \dots & (2n-1)(-1)^{n+1} \\ 1 & 3 & 5 & 7 & \dots & (2n-1)(-1)^n \\ \dots & & & & & \\ 1 & 3 & 5 & 7 & \dots & (2n-1) \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

$$c = (1 \ -3 \ 5 \ -7 \ \dots \ (-1)^{n+1}(2n-1)) \tag{20}$$

정리 1

식 (19)의 시스템 $CF_{[n,m]}(s)$ 은 dead time의 Pade 근사식 $P_{[n,m]}(s)$ 을 구현한다.

증명

식 (17)은 식 (13)을 근사하는 시스템이며, 따라서 식 (19)는 식 (15)를 근사하는 것이 된다. 예비정리 1에 따라 이 근사는 $P_{[n,m]}(s)$ 와 같다.

5. 검토 및 모사

5-1. Dead time 근사 상태방정식의 응용

식 (19)와 (20)은 dead time을 근사하는 시스템으로 다양한 용도로 편리하게 사용할 수 있다. Laplace Transform을 하면 Pade 근사식을 얻을 수 있다. 즉 식 (19)로부터 다음과 같이 $P_{[n,m]}(s)$ 를 얻을 수 있다.

$$P_{[n,m]}(s) = c(sI - A)^{-1}b + d \tag{21}$$

이 계산은 MATLAB에서 '[num,den]=ss2tf(A,b,c,d)'로 할 수 있다. 또한 Taylor 전개 계수

$$\begin{aligned} G(s) &\approx c(sI - A)^{-1}b + d = d - cA^{-1}(I - sA^{-1})^{-1}b \\ &= d - cA^{-1}(I + A^{-1}s + A^{-2}s^2 + A^{-3}s^3 + \dots)b \\ &= d - cA^{-1}b - cA^{-2}bs - cA^{-3}bs^2 - cA^{-4}bs^3 - \dots \end{aligned} \tag{22}$$

와 Markov 계수

$$\begin{aligned} G(s) &\approx c(sI - A)^{-1}b + d = d + cs^{-1}(I - As^{-1})^{-1}b \\ &= d + c(I + As^{-1} + A^2s^{-2} + A^3s^{-3} + \dots)bs^{-1} \\ &= d + cb\frac{1}{s} + cAb\frac{1}{s^2} + cA^2b\frac{1}{s^3} + cA^3b\frac{1}{s^4} + \dots \end{aligned} \tag{23}$$

를 쉽게 행렬 연산을 통하여 얻을 수 있다. 이 과정에서 행렬 A의 연산을 여러 차례 하기 때문에 수치적 안정성이 중요하다. 아래 비교에서 보이겠지만 식 (19) 및 (20)의 행렬 A는 수치적으로 매우 안정한 행렬이다.

식 (19)의 상태방정식은 시스템의 pole과 zero를 계산하는데도 유용하게 사용된다. 비례제어 시스템의 root locus 계산에 사용된 예를 보인다. 순수 dead time 시스템($G(s)=e^{-s}$)에 비례제어기(제어기 이득= K_c)가 설치되었을 때, 폐루프 시스템의 pole은 다음 식의

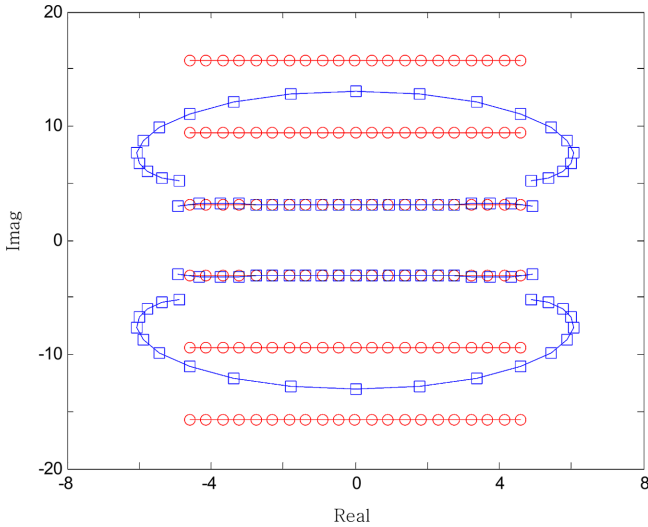


Fig. 1. Root loci of $1+K_c e^{-s}$ for the approximation orders of $n=4$ (Square: \square) and 80 (Circle: \circ), where $K_c \in [0.01, 100]$ The loci with $n=80$ are identical to the exact loci.

근이다.

$$1 + K_c e^{-s} = 0 \tag{24}$$

이 식은 $s = \ln K_c + j(2k+1)\pi$, $k = 1, 2, \dots$ 가 근이 된다는 것을 쉽게 알 수 있다. 수치적으로 이를 구하려면 dead time 부분을 근사하여야 한다. 여기에는 Pade 근사와 다항식의 근을 구하는 프로그램이 이용할 수 있고, 또한 식 (19)에 비례제어기가 설치된 다음의 페루프 시스템 행렬의 고유치를 찾아도 된다.

$$A_{cl} = A - K_c bc \tag{25}$$

다항식의 근을 구하는 문제의 수치적 불안정을 고려하면 식 (25)의 페루프 시스템 행렬의 고유치를 구하는 방식이 추천된다. Fig. 1에 근사 차수가 4와 80인 경우의 근의 궤적을 보였다. 근이 $s = \ln K_c + j(2k+1)\pi$ 형태이므로 K_c 가 바뀔 때 따른 근은 실수축에 평행한 선이 얻어지는데, 그림에서는 근사차수 $n=80$ 과 같다. 근사 차수 $n=4$ 에서는 실제 값과는 (Fig. 1에서 $n=80$ 의 그래프) 상당한 차이가 있음을 알 수 있다. 손으로 계산할 때는 $n=2$ 와 같은 낮은 차수의 근사가 일반적이지만 Fig. 1은 이 계산이 오차가 많은 것임을 보여주며, 실제 계산에는 다소 높은 차수의 근사가 필요함을 보여주고 있다. 이때 수치적으로 안정한 본 continued fraction에 근거하는 방법이 추천된다.

5-2. 시스템 행렬의 condition number 비교

Pade 근사는 분수 형태의 다항식을 기본으로 한다. 다항식은 수치적으로 안정적이지 못하다. 예를들면, $P_{[6,6]}(s) = \frac{665280 - 332640s + 75600s^2 - 10080s^3 + 840s^4 - 42s^5 + s^6}{665280 + 332640s + 75600s^2 + 10080s^3 + 840s^4 + 42s^5 + s^6}$ 으로 여기에는 매우 큰 계수 665280이 있는가 하면 계수가 작은 1도 있어 작은 수치적 오차에도 결과는 큰 오차를 가질 수 있다. 이 시스템의 모사를 위하여 보통의 canonical 형태의 상태방정식으로 변환하면 시스템 행렬 A는 다음과 같이 된다.

$$A = \begin{pmatrix} -42 & -840 & -10080 & -75600 & -332640 & -665280 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{26}$$

이 행렬의 condition number는 $\text{cond}(A) = 8.4 \times 10^5$ 가 된다. MATLAB은 dead time의 Pade 근사를 위한 명령어가 있는데, 이것으로 $P_{[6,6]}(s)$ 를 얻고(명령어: '[num,den] = pade(1,6)'), 이로부터 상태방정식을 구하면(명령어: '[A,b,c,d] = tf2ss(num,den)') 위의 식 (26)의 시스템 행렬이 얻어진다. 이렇게 하지 않고 바로 상태방정식이 얻어지도록 하면(명령어: '[A,b,c,d] = pade(1,6)'), 시스템 행렬 A는 다음과 같이 된다.

$$A = \begin{pmatrix} -42 & -26.25 & -19.6875 & -9.2285 & -5.0757 & -2.5378 \\ 32 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \end{pmatrix} \tag{27}$$

이 행렬의 condition number는 $\text{cond}(A) = 76$ 가 된다. 반면에 제안된 식 (19)에서의 시스템 행렬 A는

$$A = -2 \begin{pmatrix} 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 2 & -2 & 2 & -2 & 2 \\ 1 & 2 & 3 & -3 & 3 & -3 \\ 1 & 2 & 3 & 4 & -4 & 4 \\ 1 & 2 & 3 & 4 & 5 & -5 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix} \tag{28}$$

으로 이 행렬의 condition number는 $\text{cond}(A) = 8.8$ 이다. Condition number는 역행렬을 구하는데 있어 어려움의 정도를 나타내는 것으로 1에 가까울수록 어려움이 없는 것을 나타낸다. 이 condition number가 크면 역행렬 등의 계산에서 수치적 오차가 크게 나타나는 것을 의미한다.

5-3. 계단응답 비교

계단응답 비교를 수행하였다. Fig. 2는 순수한 dead time만 있는 개루프 시스템 $Y(s) = e^{-s}U(s)$ 에 단위 계단 입력이 들어갈 때의 출력 응답이다. Pade 근사 시스템, Pade 근사의 각 계수에 10^{-6} 의 상대 오차가 첨가된 시스템, 그리고 제안된 continued fraction 근사 시스템의 출력 응답을 비교하였다. 계산은 모두 MATLAB으로 하였다. 근사 차수 $n=36$ 에서 10^{-6} 의 상대 오차를 갖는 Pade 근사 시스템은 발산하는 결과를 보였다. 10^{-6} 의 상대 오차를 갖는 Pade 근사 시스템을 비교한 것은 최근 embedded CPU에서 single precision으로 계산이 구현될 때 나타날 수 있는 오차이기 때문이다. Pade 근사 계수만 10^{-6} 의 상대 오차를 가지고 나머지 계산은 정밀하게 이루어졌다. Single precision으로 모든 계산이 이루어지는 embedded CPU에서는 $n=36$ 보다 더 낮은 차수에서 수치적 문제를 일으킬 것으로 보인다. Pade 근사 시스템과 제안된 continued fraction 근사 시스템이 $n=76$ 까지는 같은 결과를 주었다. $n=77$ 에서 Pade 근사 시스템은 약간의 오차를 보이다가 $n=78$ 에서 발산하는 것을 볼 수 있었다. 모든 계산

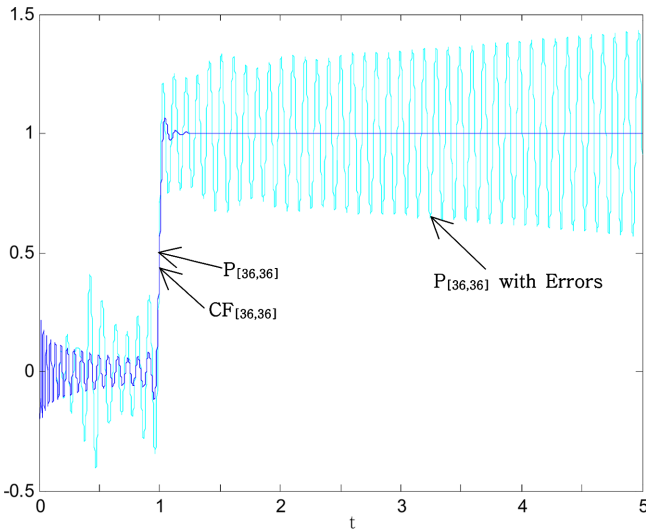


Fig. 2. Unit step responses of approximate systems for the pure dead time system, $G(s) = e^{-s}$.

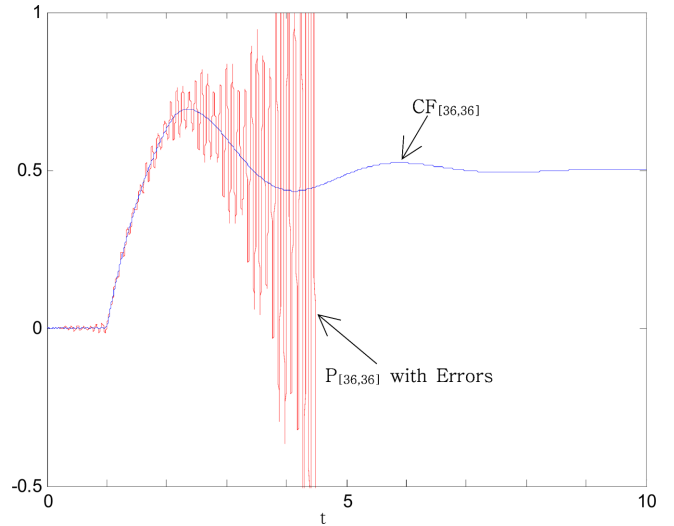


Fig. 4. Unit step responses of P control systems for the first order plus time delay process ($G(s)=e^{-s}/(s+1)$, P controller gain=1, dead time approximation order (n)=36).

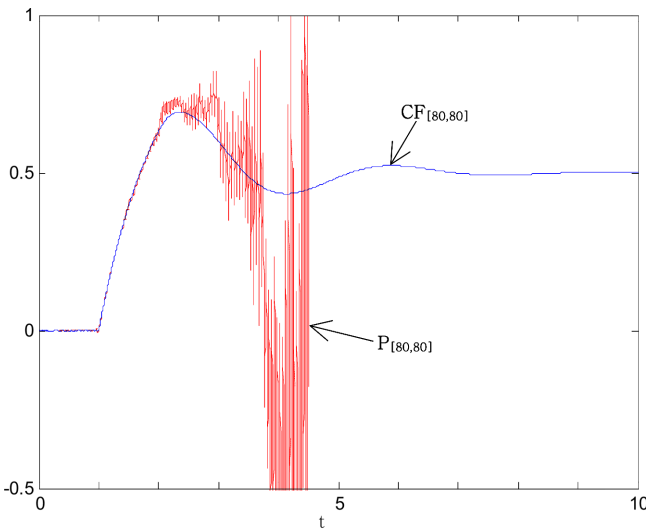


Fig. 3. Unit step responses of P control systems for the first order plus time delay process ($G(s)=e^{-s}/(s+1)$, P controller gain=1).

$$e^{-s} = \frac{1}{1+s \frac{1}{1-\frac{s}{2+\frac{s}{3-\frac{s}{2+\frac{s}{5-\dots}}}}}}$$

전개를 절사하면 Pade 근사가 얻어진다. 이 전달함수는 다음과 같이 상태방정식 시스템 (A,b,c,d)로 변환된다.

$$A = \tilde{A} = -2 \begin{pmatrix} 1 & -1 & 1 & -1 & \dots & (-1)^{n+1} \\ 1 & 2 & -2 & 2 & \dots & 2(-1)^n \\ 1 & 2 & 3 & -3 & \dots & 3(-1)^{n+1} \\ 1 & 2 & 3 & 4 & \dots & 4(-1)^n \\ \dots & & & & & \\ 1 & 2 & 3 & 4 & \dots & n \end{pmatrix}, b = \tilde{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

$$c = -\tilde{c}\tilde{A} = \begin{cases} (4 \ -4 \ 24 \ \dots \ 2n(n+1)) & n = \text{odd} \\ (0 \ -12 \ 12 \ -40 \ \dots \ -2n(n+1)) & n = \text{even} \end{cases}$$

$$d = 1 + \tilde{c}\tilde{b} = \begin{cases} -1 & n = \text{odd} \\ 1 & n = \text{even} \end{cases}$$

혹은

$$A = - \begin{pmatrix} 1 & -3 & 5 & -7 & \dots & (2n-1)(-1)^{n+1} \\ 1 & 3 & -5 & 7 & \dots & (2n-1)(-1)^n \\ 1 & 3 & 5 & -7 & \dots & (2n-1)(-1)^{n+1} \\ 1 & 3 & 5 & 7 & \dots & (2n-1)(-1)^n \\ \dots & & & & & \\ 1 & 3 & 5 & 7 & \dots & (2n-1) \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

$$c = (1 \ -3 \ 5 \ -7 \ \dots \ (-1)^{n+1}(2n-1))$$

이 방법은 (1) 근사식이 체계적이어서 쉽게 기억할 수 있으며, (2)

이 double precision 보다 더 정밀한 계산으로 이루어지기 때문에 매우 큰 근사 차수 까지 수치적 차이를 보이지 않는 것을 볼 수 있다.

Fig. 3과 4는 일차시간지연 시스템에 비례제어기가 설치되었을 때의 계단 설정치 응답을 보였다. 페루프 시스템의 전달함수는 $Y(s) = \frac{e^{-s}}{s+1+e^{-s}} R(s)$ 이다. 여기서 R(s)는 설정치 변수의 Laplace Transform이다. Fig. 3은 개루프 응답에서와 마찬가지로 $n=36$ 에서 10^{-6} 의 상대 오차를 갖는 Pade 근사 시스템이 발산하는 것을 보여준다. Fig. 4는 $n=80$ 에서 Pade 근사 시스템이 발산하는 결과를 보여주고 있다.

6. 결 론

공정의 동특성 기술에 자주 나타나는 dead time을 근사하는 방법을 제안하였다. 자주 사용되는 Pade 근사와 같은 결과를 주지만 여러 장점을 가지고 있는 다음의 continued fraction 전개에 근거하는 방법을 이용하였다.

dead time 시스템을 근사하는 상태방정식이 쉽게 얻어지고, (3) 얻어진 상태방정식은 수치적으로 매우 안정적이라는 장점들을 가진다. 이들을 모사를 통하여 확인하였다.

감 사

본 연구는 교육과학기술부 한국연구재단(중견(핵심)연구자지원사업, 2011-0013841)의 연구 지원으로 수행되었으며 이에 감사드립니다.

참고문헌

1. Abramowitz, M. and Stegun, I. A., *Handbook of Mathematical Functions*, Dover, New York(1965).
2. Brezinski, C., *History of Continued Fractions and PadApproximants*, Springer-Verlag, New York(1980).
3. Byrnes, C. I. and Lindquist, A., "Stability and Instability of Partial Realizations," *Systems and Control Letters*, **2**, 99-105(1982).
4. Chen, C. F. and Shieh, L. S., "Continued Fraction Inversion by the Routh's Algorithm," *IEEE Trans. Circuit Theory*, **16**(2), 197-202(1969).
5. Golub, G. H. and van Loan, C. F., *Matrix Computations*, Johns Hopkins University Press, Baltimore(1989).
6. Kuo, B. C., *Digital Control Systems*, 2nd ed., Saunders College Publishing, Orlando(1992).
7. Lee, J., Park, H. and Sung, S. W., "Analytic Expressions of Ultimate Gains and Ultimate Periods with Phase-Optimal Approximations of Time Delays," *Canadian J. Chemical Engineering*, **83**, 990-995(2005).
8. Lorentzen, L., "Pade Approximation and Continued Fractions," *Applied Numerical Mathematics*, **60**, 1364-1370(2010).
9. Richard, J. P., "Time-delay Systems: an Overview of Some Recent Advances and open Problems," *Automatica*, **39**, 1667-1694(2003).
10. Seborg, D. E., Edgar, T. F., Mellichamp, D. A. and Doyle, III, F. J., *Process Dynamics and Control*, 2nd ed., Wiley, New Jersey (2010).
11. Ju, S., Kim, S. J., Byeon, J., Chund, D., Sung, S. W. and Lee, J., "A Study on the First Order Plus Time Delay Model Identification from Noisy Step Responses," *Korean Chem. Eng. Res.(HWA-HAK KONGHAK)*, **46**, 949-957(2008).
12. Byeon, J., Kim, J. S., Sung, S. W., Ryoo, W. and Lee, J., "Third Quadrant Nyquist Point for the Relay Feedbak Autotuning of PI Controllers," *Korean J. Chem. Eng.*, **28**, 342-347(2011).

부 록

A. Routh Array에 의한 Continued Fraction 전개 계수 계산[4]:

다음의 다항식 전달함수를 고려한다.

$$G(s) = \frac{A_{21} + A_{22}s + \dots + A_{2m}s^m}{A_{11} + A_{12}s + \dots + A_{1n}s^n} \tag{A1}$$

이 전달함수의 계수로 이루어진 다음의 Routh Array를 만든다.

$$\begin{matrix} A_{11} & A_{12} & A_{13} & \dots \\ A_{21} & A_{22} & A_{23} & \dots \\ A_{31} & A_{32} & A_{33} & \dots \\ A_{41} & A_{42} & \dots & \dots \end{matrix} \tag{A2}$$

여기서

$$A_{ik} = A_{(i-2)(k+1)} - \frac{A_{(i-2)1}A_{(i-1)(k+1)}}{A_{(i-1)1}}, i = 3, 4, \dots, k = 1, 2, \dots \tag{A3}$$

이제 G(s)를 근사하는 continued fraction 전개는 다음과 같이 된다.

$$G(s) \approx \frac{1}{h_1 + \frac{s}{h_2 + \frac{s}{h_3 + \frac{s}{h_4 + \dots}}}} \tag{A4}$$

$$h_j = \frac{A_{j1}}{A_{(j+1)1}}, j = 1, 2, \dots$$

여기서 m=n-1이면 근사가 아닌 같은 전달함수가 얻어진다. G(s)의 Taylor 전개식에도 A₁₁ = 1, A_{1k} = 0, k = 2, 3, ...로 두고 이 방법을 적용하여 식 (A4)와 같은 continued fraction 전개를 얻을 수 있다. 이 continued fraction 전개를 절사하고 다시 식 (A1) 형태로 정리하면 m=n-1 혹은 m=n이 되는 Pade 근사가 된다[4].

B. Continued Fraction의 Realization[4]:

식 (A4)에서 h_{2n}에서 절사한 것은 다음과 같은 상태방정식 시스템의 전달함수와 같게 된다.

$$\begin{aligned} \dot{x} &= Ax + bu \\ y &= cx \end{aligned}$$

$$A = \begin{pmatrix} h_2h_1 & h_4h_1 & h_6h_1 & \dots & h_{2n}h_1 \\ h_2h_1 & h_4(h_1+h_3) & h_6(h_1+h_3) & \dots & h_{2n}(h_1+h_3) \\ h_2h_1 & h_4(h_1+h_3) & h_6(h_1+h_3+h_5) & \dots & h_{2n}(h_1+h_3+h_5) \\ \dots & \dots & \dots & \dots & \dots \\ h_2h_1 & h_4(h_1+h_3) & h_6(h_1+h_3+h_5) & \dots & h_{2n}(h_1+h_3+\dots+h_{2n-1}) \end{pmatrix}$$

$$b = (1 \ 1 \ 1 \ \dots \ 1)^T$$

$$c = (h_2 \ h_4 \ h_6 \ \dots \ h_{2n}) \tag{A5}$$

즉

$$G(s) = c(sI - A)^{-1}b \tag{A6}$$

를 얻을 수 있다. 식 (A5)는 식 (A4)가 될 수 있는 feedback/feedforward 제어시스템을 고안하여 유도할 수 있다[4].