

여기서, $n = \log_2 m, i = 1, 2, \dots, n-1$ 이고,

$$Q_{\frac{m}{2}} = 2^{(2n+1)} \text{ 이다.}$$

2.2 블럭펄스함수(Block Pulse Fuction)

블럭펄스함수는 1970년대 중반부터 월쉬함수를 이용한 시스템 해석방법에 대한 복잡성을 줄이기 위해 이용되면서 블럭펄스함수와 블럭펄스함수 연산행렬이 갖는 단순성의 이점에 의해 제어공학분야에 현재까지 광범위하게 사용되고 있다. 이러한 블럭펄스함수의 UIO 설계분야로의 적용은 식(1),(2),(3)과 같이 미분연산행렬로 유도된 월쉬함수와는 다르게 미분연산을 위한 반복연산 알고리즘으로 유도되었다. 식(4)는 블럭펄스함수의 적분연산행렬의 역행렬을 이용하여 표현된 미분연산행렬이다.

$$P_{b_m}^{-1} = \frac{2m}{t_f} \begin{bmatrix} 1 & -2 & 2 & \dots & 2 \times (-1)^{m-1} \\ 0 & 1 & -2 & \dots & 2 \times (-1)^{m-2} \\ 0 & 0 & 1 & \dots & 2 \times (-1)^{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (4)$$

UIO 설계에 대한 실질적 적용을 위해 블럭펄스함수의 반복적 미분연산알고리즘은 다음과 같이 표현된다. 먼저, 임의의 실유계 연속함수 $f(t)$ 를 식(5)와 같이 m 개의 블럭펄스함수로 유한급수 전개하여 표현하고,

$$f(t) \doteq \sum_{i=1}^m f_i \phi_{b_i}(t) = f_m^T \Phi_{b_m}(t) = \Phi_{b_m}^T(t) f_m \quad (5)$$

$$\text{여기서, } f_m^T = [f_1, f_2, \dots, f_m]$$

$$\Phi_{b_m}(t) = [\phi_{b_1}(t), \phi_{b_2}(t), \dots, \phi_{b_m}(t)]^T$$

이에 대한 1계 미분을 취한 함수를 식(6)로 나타내면,

$$\frac{d}{dt} f(t) \doteq \sum_{i=1}^m \bar{f}_i \phi_{b_i}(t) \quad (6)$$

블럭펄스함수의 직교성과 배타성을 이용하여 유도된 미분연산을 위한 반복알고리즘은 식(7),(8)과 같이 표현된다.[3]

$$\bar{f}_1 = \frac{2m}{t_f} [f_1 - f(0)] \quad (7)$$

$$\bar{f}_{i+1} = \frac{2m}{t_f} [f_{i+1} - f_i] - \bar{f}_i$$

여기서, $i = 1, 2, \dots, m-1$

$$\bar{f}_i = \frac{2m}{t_f} \left[f_i + \left(2 \times (-1)^i \sum_{j=1}^{i-1} (-1)^j f_j \right) + (-1)^i f(0) \right] \quad (8)$$

여기서, $i = 1, 2, \dots, m-1$

2.3 하알함수(Haar Fuction)

잘 알려진 바와 같이 하알함수는 다음의 식(9)과 같이 정의되는 불연속의 완전 정규직교함수이다.

$$h_0(t) = 1$$

$$h_1(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2} \\ -1, & \frac{1}{2} < t \leq 1 \end{cases} \quad (9)$$

⋮

$$h_{(2^p+g)}(t) = \begin{cases} \sqrt{2^p}, & \frac{g}{2^p} \leq t < \frac{(g+1)}{2^p} \\ -\sqrt{2^p}, & \frac{(g+1)}{2^p} \leq t < \frac{(g+1)}{2^p} \\ 0, & \text{그 외 구간} \end{cases}$$

여기서, $p = 0, 1, \dots, \log_2 \frac{m}{2}$ 이고, $g = 0, 1, \dots, 2^p - 1$ 이다. 하알함수의 미분연산행렬의 유도는 블럭펄스함수를 이용하여 유도할 수 있는데, 식(10)에서 각각 β_i 는 i 번째 블럭펄스 함수, h_i 는 i 번째 하알함수의 계수 값을 나타낸다.

$$\begin{bmatrix} h_0(t) \\ h_1(t) \\ h_2(t) \\ h_3(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} \beta_0(t) \\ \beta_1(t) \\ \beta_2(t) \\ \beta_3(t) \end{bmatrix} \quad (10)$$

식(10)는 식(11),(12),(14),(15)를 통해

$$H_m(t) = \psi_{h_{k(m)}} \cdot B_m(t) \quad (11)$$

$$\int_0^1 \psi_{h_{k(m)}} \cdot B_m(t) dt = P_{h(m)} \cdot \psi_{h_{k(m)}} \cdot B_m(t) \quad (12)$$

$$\int_0^1 B_m(t) dt = \frac{1}{m} \psi_{h_{k(m)}}^T \cdot P_{h(m)} \cdot \psi_{h_{k(m)}} \cdot B_m(t) \quad (13)$$

$$\frac{1}{m} \psi_{h_{k(m)}}^T \cdot P_{h(m)} \cdot \Psi_{h(m)} = P_{\beta(m)} \quad (14)$$

$$\text{여기서, } \Psi_{h(m)} \cdot \Psi_{h(m)}^T = m \cdot I_m$$

$\Psi_{h(m)}^{-1} = \frac{1}{m} \Psi_{h(m)}^T$ 의 성질을 이용하면, 식(15)을 구할 수 있다.

$$P_{\beta(m)} = \Psi_{h(m)}^{-1} \cdot P_{h(m)} \cdot \Psi_{h(m)} \quad (15)$$

양변에 역행렬을 취하면, 식(16)으로 함수의 미분연산행렬을 간명하게 구할 수 있다.

$$P_{\beta(m)}^{-1} = \Psi_{h(m)}^{-1} \cdot P_{h(m)}^{-1} \cdot \Psi_{h(m)} \quad (16.1)$$

$$\begin{aligned} P_{h(m)}^{-1} &= \Psi_{h(m)} \cdot P_{\beta(m)}^{-1} \cdot \Psi_{h(m)}^{-1} \\ &= \frac{1}{m} \Psi_{h(m)} \cdot P_{\beta(m)}^{-1} \cdot \Psi_{h(m)}^T \end{aligned} \quad (16.2)$$

3. 적용예제

(예제1)

$$f(t) = t^2 \text{ 인 경우, } g(t) = 2t = \frac{d}{dt} f(t) \text{ 와}$$

$h(t) = \frac{1}{3} t^3 = \int_0^t f(x) dx$ 인 경우 블럭펄스함수 미분연산행렬과 적분연산행렬을 이용한 각각 $f(t), g(t), h(t)$ 함수의 근사화는 그림1, 그림2와 같이 표현된다.

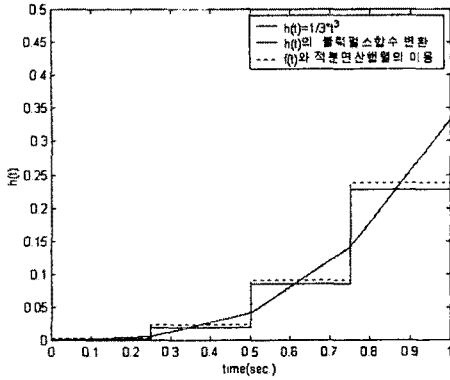


그림 1. $f(t)$, $h(t)$ 의 블럭펄스함수 전개

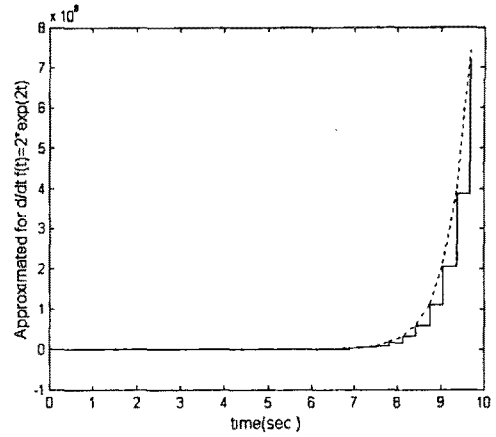


그림 4. $f_1(t) = \frac{d}{dt} f_2(t) = 2 \cdot e^{2t}$, $f_2(t) = e^{2t}$

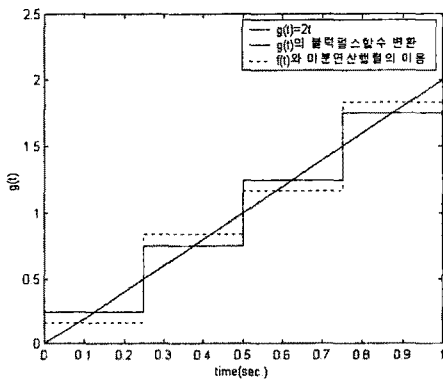


그림 2. $f(t)$, $g(t)$ 의 블럭펄스함수 전개

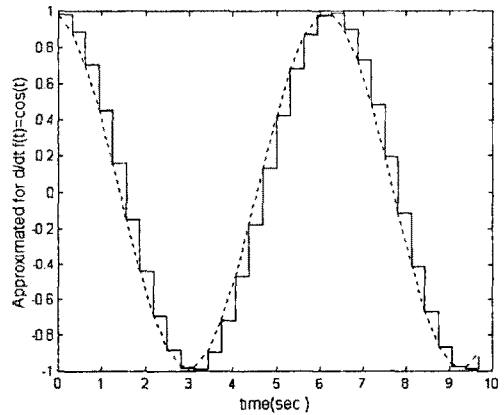


그림 5. $f_1(t) = \frac{d}{dt} f_2(t) = \cos(t)$, $f_2(t) = \sin(t)$

(예제2)

$\frac{1}{2} t^2$, e^{2t} , $\sin(t)$ 의 함수에 대해서 하일함수 미분연산 행렬을 이용하여 구해진 각각의 미분함수 t , $2e^{2t}$, $\cos(t)$ 와 비교해보자.

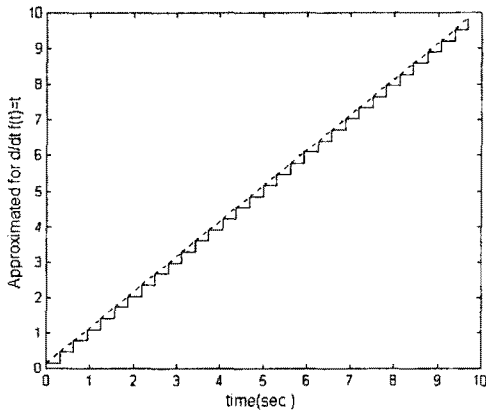


그림 3. $f_1(t) = \frac{d}{dt} f_2(t) = t$, $f_2(t) = \frac{1}{2} t^2$

4. 결 론

일반적으로 미지입력이 포함된 선형동적 시스템에 대해서 좌표변환방법을 이용하여 Luenberger 관측기 형태의 UIO를 설계할 경우, 피할 수 없이 미지입력 추정식에 출력의 미분 값이 포함되는 것이 사실이다. 이러한 경우 실제의 하드웨어 설계를 통한 미지입력 추정은 출력 값(센싱된 값)의 작은 노이즈 혹은 잡음에도 민감하게 반응하게 되어 정확한 값을 추정하기 어렵게 된다. 따라서, 본 연구에서 소개한 직교함수변환과 직교함수 미분연산행렬(식)을 이용하여, 설계된 UIO의 대수적인 변환을 통해 출력단의 잡음에 덜 민감한 대수적 미지입력 추정기를 구성할 수 있으며, 이는 향후 시스템 고장 진단을 통한 고장값 추정 등에 용이하게 활용될 수 있을 것으로 사료된다.

[참 고 문 헌]

- [1] J. H. Lee and Z. Bien. "Design of reduced-order observers for linear system with unknown inputs." Trans KIEE. vol 43, no 1, pp 71-75, 1994.
- [2] K. B. Datta and B. M. Mohan. "Orthogonal functions in system and control." World Scientific Publishing, 1995
- [3] P. Ahn, M. K. Kim, and D. S. Ahn. "A novel approach to unknown input observer design via block pulse function's differential operation." Proc. of IFAC World Congress. vol. D, pp 255-260, 1999.