

## 고속 월쉬 변환을 이용한 쌍일차계의 해석에 관한 연구

(A Study on the Analysis of Bilinear Systems via Fast Walsh Transform)

김태훈\* · 심재선

(Tai-Hoon Kim · Jae-Sun Shim)

### 요 약

일반적으로 시스템의 해석에 직교 함수를 이용하는 경우에는 크로네커 곱(Kronecker product)에 의하여 고차 행렬에 대한 역변환이 필요하게 되며, 이로 인하여 많은 연산 시간이 필요하게 된다. 본 연구에서는 이 문제점을 해결하고자 고속 월쉬 변환을 이용하는 방법을 제시하였고, 이렇게 함으로써 크로네커 곱에 의한 다루기 힘든 고차 행렬이나 그에 따르는 행렬들의 계산을 필요없게 함으로써 연산의 부담을 줄일 수 있게 된다.

본 연구에서는 쌍일차계의 해석을 위한 직교 함수의 유한 급수 전개 방법과 고속 월쉬 변환 방법을 비교하여 봄으로써 본 연구에서 제안한 방법의 우수성을 표현하였으며, 시뮬레이션을 통하여 고속 월쉬 변환에 의한 쌍일차계 상태 해석 결과를 표시하였다.

### Abstract

Generally when the orthogonal functions are used in system analysis, the time consuming processes of high order matrix inversion for finding the Kronecker products and the truncation errors are occurred. In this paper, a method for the system analysis of bilinear systems via fast walsh transform is devised. This method requires neither the inversion of large matrices nor the cumbersome procedures for finding Kronecker products. Thus, both the computing time and required memory size can be significantly reduced.

Key Words : Bilinear system, Fast Walsh Transform, system analysis

### 1. 서 론

일반적으로 시스템의 해석에 직교 함수 유한 급수 전개를 이용하는 경우에 있어서, 크로네커 곱(Kronecker product)에 의해 도입되는 고차 행렬의 역변

환이 필요하게 됨으로써 많은 시간이 필요하게 됨과 동시에 오차에 의해 정확성을 기할 수 없다는 문제점을 가지게 된다[1-3].

본 연구에서는 이 문제점을 해결하고자 고속 월쉬 변환을 이용하는 방법을 제시하였고, 이렇게 함으로써 크로네커 곱에 의하여 생성되는 다루기 힘든 고차 행렬이나 그에 따르는 행렬들의 계산을 필요없게 됨과 동시에 연산 시간과 용량이 줄어들게 된다.

본 연구에서는 먼저 쌍일차계의 해석에 사용될 수 있는 직교 함수 유한 급수 전개에 의한 접근 방법과

\* 주저자 : 한국정보보호진흥원 선임연구원

Tel : 405-5323, Fax : 405-5319

E-mail : taihoon@netsgo.com

접수일자 : 2001년 11월 2일

1차심사 : 2001년 11월 12일

심사완료 : 2002년 1월 17일

고속 월쉬 변환을 이용한 쌍일차계의 해석에 관한 연구

본 연구에서 제안하는 고속 월쉬 변환에 의한 방법을 수식을 통하여 비교 고찰하였으며, 이러한 비교를 통하여 직교 함수의 유한 급수 전개 방법에 나타나는 고차 행렬의 역변환 등의 불편한 점을 극복할 수 있음을 보이고, 시뮬레이션을 통하여 쌍일차계의 상태를 해석하였다.

## 2. 직교 함수 유한 급수 전개

적분 가능한 임의의 함수  $f(t)$ 를 정의 구간  $[0, 1)$ 에서 직교 함수로 유한 급수 전개하면 다음과 같다 [1-3].

$$f(t) = \sum_{n=0}^{m-1} f_n \phi(n, t) \quad (2.1)$$

$$= \mathbf{F}^T \boldsymbol{\Phi}(t) = \boldsymbol{\Phi}^T(t) \mathbf{F}$$

단,  $\mathbf{F} = [f_0, f_1, f_2, \dots, f_{m-1}]^T$ ,

$$\boldsymbol{\Phi}(t) = [\phi(0, t) \phi(1, t) \dots \phi(m-1, t)]^T$$

여기서 T는 전치를 의미하고, m은 직교 함수의 전개 항수이며,  $f_n$ 은 직교 함수의 n번째 계수로 다 음의 적분 자승 오차를 최소화하도록 결정한다.

$$\varepsilon = \int_0^1 \left[ f(t) - \sum_{n=0}^{m-1} f_n \phi(n, t) \right]^2 dt \quad (2.2)$$

월쉬, 블록 펄스, 하알 함수 등은 직교 함수로서 시구간  $t \in [0, 1)$ 에서 정의되며, 각 직교 함수의 계수  $f_n$ 은 다음의 식 (2.3)과 같이 결정된다.

$$f_n = \int_0^1 \phi(n, t) f(t) dt \quad (2.3)$$

단,  $n = 0, 1, 2, \dots, m-1$

임의의 함수  $f(t)$ 와  $g(t)$ 의 곱으로 표현된 경우 를 살펴 보면,  $g(t)$ 는 식 (2.4)와 같으므로

$$g(t) = \mathbf{G}^T \boldsymbol{\Phi}(t) = \boldsymbol{\Phi}^T(t) \mathbf{G} \quad (2.4)$$

다음의 식 (2.5)와 같이 정리될 수 있다.

$$f(t)g(t) = \mathbf{F}^T \boldsymbol{\Phi}(t) \boldsymbol{\Phi}^T(t) \mathbf{G} \quad (2.5)$$

$$= \mathbf{F}^T \boldsymbol{\Psi}(t) \mathbf{G}$$

$$= \mathbf{F}^T \mathbf{A}_G \mathbf{G}$$

단,  $\boldsymbol{\Psi}(t) = \boldsymbol{\Phi}(t) \boldsymbol{\Phi}^T(t)$ 이며,  $\mathbf{A}_G$ 는 다음의 관 계로부터 결정된다[4].

$$\phi_{ij} = \phi_i(t) \phi_j(t) = \phi_{ib \cdot jb}(t) \quad (2.6)$$

또한 직교 함수의 적분은 역시 직교 함수로 표현 할 수 있으며, 그 관계는 다음과 같다.

$$\int_0^1 \boldsymbol{\Phi}(\tau) d\tau = \mathbf{P} \boldsymbol{\Phi}(t) \quad (2.7)$$

여기에서  $\mathbf{P}$ 를 적분 연산 행렬이라고 한다. 이 적 분 연산 행렬은 미분 방정식으로 표시된 시스템을 적분 방정식으로 변환하고 직교 함수를 도입할 때 적분 연산자로 사용된다[1-3].

## 3. 고속 월쉬 변환

고속 월쉬 변환은 이산 월쉬 변환을 보다 신속하 게 처리하기 위해 사용된다. 월쉬 함수  $m$ 항 전개를 이용하는 경우 모든 월쉬 함수는  $m$ 개의 세부 구간 에서 +1 혹은 -1 의 값만으로 구성되므로 각 월쉬 계수들은 원 함수의 이산값들의 가·감산에 의하여 간단히 계산할 수 있다. 고속 월쉬 변환 알고리즘은 Welch과 Guinn에 의해 제시되었으며 고속 푸우리 에 변환(fast Fourier transform)의 주파수축을 알고 리즘(decimation-in frequency Algorithm)을 기초로 하여 개발되었다[5].

고속 월쉬 변환 알고리즘은 그림 3.1과 같으며 그 림에서 점선은 감산을 나타내고 실선은 가산을 나타 낸다.

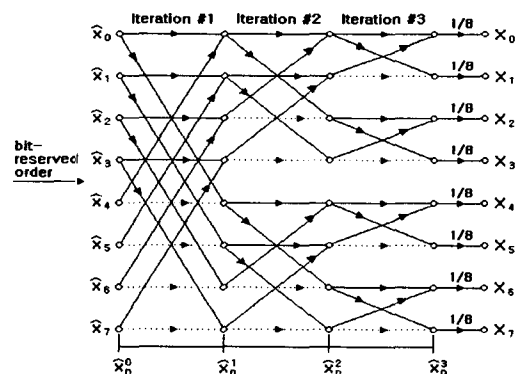


그림 3.1. 고속 월쉬 변환 신호흐름선도  
Fig. 3.1. The flow diagram for a fast Walsh transform

그림에서와 같이 고속 윌쉬 변환은 다음의 과정을 따른다.

[1] 윌쉬함수  $m$ 항 전개를 이용하는 경우 고속 윌쉬 변환은  $\log_2 m$ 개의 단으로 나눈다.

[2] 최초의 단( $p=0$ )에  $\bar{x}_i$ 를 비트 자리 바꿈(bit-reverse)하여 증가순서대로 재배열하고  $p$ 번째 단과  $p-1$ 번째 단과의 관계는 다음과 같다.

$$x_{(h)}^p = x_{(h)}^{p-1} + x_{(j+h)}^{p-1} \quad (3.1)$$

$$x_{(j+h)}^p = x_{(h)}^{p-1} - x_{(j+h)}^{p-1} \quad (3.2)$$

단,  $j = m/2^p$ ,  $h = l + k \times 2^v$ ,  $v = \log_2(j+1)$

$k = 0, 1, \dots, 2^{p-1} - 1$ ,  $l = 0, 1, \dots, j-1$

[3] 고속 윌쉬 변환과 역변환은 똑같은 과정에 의해 이루어지며, 고속 윌쉬 변환의 경우에는 식(3.1), (3.2)의 계산결과에  $1/m$ 을 곱하면 된다.

윌쉬 함수  $m$ 항 전개를 이용하는 경우 고속 윌쉬 변환은  $p (= \log_2 m)$ 개의 단계가 필요하고 각 단계에서  $m$ 번의 가·감산이 필요하므로 총  $m \log_2 m$  번의 가·감산을 통하여 이루어진다.

### 4. 쌍일차계의 해석

#### 4.1 직교 함수 유한 급수 전개에 의한 쌍일차계의 해석

다음과 같이 표현되는 쌍일차계에서

$$\dot{x}(t) = Ax(t) + Bu(t) + \sum_{i=0}^{r-1} N_i x(t) u_i(t) \quad (4.1)$$

단,  $x(0) = x_0$ ,  $x \in R^n$ ,  $u \in R^r$ ,

$N_i (i = 1, 2, \dots, r) \in R^{n \times n}$ ,

$A \in R^{n \times n}$ ,  $B \in R^{n \times r}$

식 (4.1)의 양변에 적분을 취하면 다음과 같다.

$$x(t) - x_0 = A \int_0^t x(\tau) d\tau + B \int_0^t u(\tau) d\tau + \sum_{i=0}^{r-1} N_i \int_0^t x(\tau) u_i(\tau) d\tau \quad (4.2)$$

$x(t)$ ,  $u(t)$ , 그리고  $u_i(t)$ 는 다음과 같이 직교 함수로 전개된다.

$$x(t) = X^T \Phi(t), \quad u(t) = U^T \Phi(t),$$

$$u_i(t) = U_i^T \Phi(t),$$

$$x(t) u_i(t) = X^T \Lambda_{u_i} \Phi(t) \quad (4.3)$$

단,

$$X^T = \begin{bmatrix} X_{10} & X_{11} & \dots & X_{1m-1} \\ X_{20} & X_{21} & \dots & X_{2m-1} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n0} & X_{n1} & \dots & X_{nm-1} \end{bmatrix} = \begin{bmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_n^T \end{bmatrix}$$

$$U^T = \begin{bmatrix} U_{10} & U_{11} & \dots & U_{1m-1} \\ U_{20} & U_{21} & \dots & U_{2m-1} \\ \vdots & \vdots & \ddots & \vdots \\ U_{r0} & U_{r1} & \dots & U_{rm-1} \end{bmatrix} = \begin{bmatrix} U_1^T \\ U_2^T \\ \vdots \\ U_r^T \end{bmatrix}$$

식 (4.2)에 식 (4.3)과 적분 연산 행렬을 도입하면 다음과 같이 식이 변형된다.

$$X^T \Phi(t) - X_0^T \Phi(t) = AX^T P \Phi(t) + BU^T P \Phi(t) + \sum_{i=0}^{r-1} NX^T \Lambda_{u_i} P \Phi(t) \quad (4.4)$$

단,  $X_0^T = [x_0, 0, 0, \dots, 0]$

식 (4.4)의 양변에 있는  $\Phi(t)$ 를 소거하면

$$X^T - X_0^T = AX^T P + BU^T P + \sum_{i=0}^{r-1} NX^T \Lambda_{u_i} P \quad (4.5)$$

과 같이 되고,  $X^T$ 는  $n \times m$ 차 행렬이므로  $X^T$ 를 구하기 위해서는 크로네커 곱이 필요하게 되고,

고속 월쉬 변환을 이용한 쌍일차계의 해석에 관한 연구

이를 이용하면 다음과 같이 된다[6].

$$\begin{aligned} \mathbf{x}_C - \mathbf{x}_{0C} &= \mathbf{P}^T \otimes \mathbf{A} \mathbf{x}_C + \mathbf{P}^T \otimes \mathbf{B} \mathbf{u}_C \\ &+ \sum_{i=0}^{r-1} (\mathbf{A}_{u_i} \mathbf{x}_C)^T \otimes \mathbf{N}_i \mathbf{x}_C \end{aligned} \quad (4.6)$$

단,

$$\mathbf{x}_C^T = [ \mathbf{X}_1 \ \mathbf{X}_2 \ \dots \ \mathbf{X}_n ] \quad (4.7a)$$

$$\mathbf{u}_C^T = [ \mathbf{U}_1 \ \mathbf{U}_2 \ \vdots \ \mathbf{U}_r ] \quad (4.7b)$$

식 (4.7)에 있는  $\mathbf{x}_C$ 와  $\mathbf{u}_C$ 에서의  $\mathbf{X}_i$ 와  $\mathbf{U}_i$ 는 식 (4.2)를 보면 알 수 있듯이 각각  $i$ 번째 상태 변수  $x_i(t)$ 와 제어 변수  $u_i(t)$ 에 대한 계수 벡터이다[7]. 또한  $\otimes$ 는 크로네커 곱을 의미하는 것으로, 각각  $n \times n$ 차 행렬과  $m \times m$ 차 행렬인  $\mathbf{A} = [a_{ij}]$ 와  $\mathbf{B} = [b_{ij}]$ 가 있다고 할 경우  $\mathbf{A} \otimes \mathbf{B}$ 는

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} b_{11} \mathbf{A} & b_{12} \mathbf{A} & \dots & b_{1n} \mathbf{A} \\ b_{21} \mathbf{A} & b_{22} \mathbf{A} & \dots & b_{2n} \mathbf{A} \\ \vdots & \vdots & \ddots & \vdots \\ b_{r1} \mathbf{A} & b_{r2} \mathbf{A} & \dots & b_{rn} \mathbf{A} \end{bmatrix} \quad (4.8)$$

과 같으며,  $\mathbf{A} \otimes \mathbf{B}$ 의 차원은  $nm \times nm$ 이 된다. 식 (4.7)에서  $\mathbf{x}_C$ 는  $\mathbf{x}_i (i = 1, 2, \dots, n)$ 가  $m$ 차 열벡터이므로  $nm$ 차 열벡터임을 알 수 있다.

식 (4.6)을  $\mathbf{x}_C$ 에 대하여 정리하면

$$\begin{aligned} \mathbf{x}_C &= \left[ \mathbf{I} - \mathbf{P}^T \otimes \mathbf{A} + \sum_{i=0}^{r-1} (\mathbf{A}_{u_i} \mathbf{P})^T \otimes \mathbf{N}_i \right]^{-1} \\ &\times (\mathbf{x}_{0C} + \mathbf{P}^T \otimes \mathbf{B} \mathbf{u}_C) \end{aligned} \quad (4.9)$$

와 같으며, 식 (4.7) 및 식 (4.2)의 관계로부터 상태 벡터인  $\mathbf{x}(t)$ 를 구할 수 있다.

하지만 이와 같은 방법을 사용하는 것은 크로네커 곱과 곱행렬에 의한 고차 행렬의 역변환이 필요하게 되고, 이로 인하여 데이터 용량이 커지게 되고 연산 시간이 증가하게 되는 단점이 있다.

예를 들어 전개 항수가 256인 경우라면,  $\mathbf{P}$ 와  $\mathbf{A}$

는 각각 크기가  $256 \times 256$ 이므로  $\mathbf{P}^T \otimes \mathbf{A}$ 의 크기는  $65536 \times 65536$ 가 될 것이고, 식 (4.9)에서는 이에 대한 역행렬 계산까지 필요하게 된다.

하지만 본 연구에서 제안하는 다음 방법을 사용하면, 식 (4.9)에서와 같은 고차 행렬의 계산을 피할 수 있으며, 또한 이에 대한 역행렬 계산도 피할 수 있게 된다.

## 4.2 고속 월쉬 변환에 의한 쌍일차계의 해석

식 (4.1)과 같이 표현되는 쌍일차계를 적분하면 식 (4.3)과 같이 되는데, 여기에 Picard의 반복 연산법을 적용하기 위하여 최초 해를  $\mathbf{x}^0(t) = \mathbf{x}_0$ 로 가정하고 식 (4.3)에 대입하면

$$\begin{aligned} \mathbf{x}^1(t) &= \mathbf{x}_0 + \mathbf{A} \int_0^t \mathbf{x}^0(\tau) d\tau + \mathbf{B} \int_0^t \mathbf{u}(\tau) d\tau \\ &+ \sum_{i=0}^{r-1} \mathbf{N}_i \int_0^t \mathbf{x}^0(\tau) u_i(\tau) d\tau \end{aligned} \quad (4.10)$$

과 같고, 반복 적용할 때의  $k$ 번째의 결과는 다음과 같다.

$$\begin{aligned} \mathbf{x}^k(t) &= \mathbf{x}_0 + \mathbf{A} \int_0^t \mathbf{x}^{k-1}(\tau) d\tau + \mathbf{B} \int_0^t \mathbf{u}(\tau) d\tau \\ &+ \sum_{i=0}^{r-1} \mathbf{N}_i \int_0^t \mathbf{x}^{k-1}(\tau) u_i(\tau) d\tau \end{aligned} \quad (4.11)$$

단,  $k = 1, 2, 3, \dots$

주어진 식(4.1)의 상태방정식이 Lipschitz 조건을 만족한다면 Picard에 의한 식(4.11)의 반복 연산에 의해 다음의 식(4.12)를 만족하는 식(4.1)의 미분방정식의 해가 존재한다[7-8].

$$e(t) = \lim_{k \rightarrow \infty} \| \mathbf{x}^k(t) - \mathbf{x}^{k-1}(t) \| \rightarrow 0 \quad (4.12)$$

$\mathbf{x}^{k-1}(t)$ 를 알고 있다고 할 때, 구간  $[0, T)$ 에서 월쉬 함수를 적용하여 식(4.11)의 해를 구하는 문제를 고려한다.

위 식(4.11)의  $\mathbf{x}^k(t)$ 와 초기조건 및  $\mathbf{u}(t)$ 를 각각

다음과 같이 월쉬 함수로 나타낸다.

$$x^k(t) = \sum_{i=0}^{m-1} X_i^k Wal(i, t) = X^k Wal(t) \quad (4.13)$$

$$x_0 = \sum_{i=0}^{m-1} X_{0,i} Wal(i, t) = X_0 Wal(t) \quad (4.14)$$

$$u(t) = \sum_{i=0}^{m-1} U_i Wal(i, t) = U Wal(t) \quad (4.15)$$

단,  $Wal(t)$ 는  $m$ 개의 월쉬 함수로 구성되는  $m$ 차 월쉬 벡터를 나타낸다.

$X^k = [X_0^k, X_1^k, \dots, X_{m-1}^k]$ 이며, 상태 벡터  $x^k(t)$ 의 월쉬 계수( $n \times m$ )행렬로 결정해야 할 값이다. 또한  $X_i^k$ 는 상태벡터  $x^k(t)$ 의  $i$  번째 월쉬 계수로서  $n$ 차 벡터이다. 초기 조건에 대한 월쉬 계수 행렬  $X_0 = [X_{00}, 0, \dots, 0]$ 의 첫 번째 계수만  $X_{00} = x_0$ 의 값을 취하며, 나머지는 모두 0의 값을 취한다.  $U_i$ 는  $u(t)$ 에 대한  $i$  번째 월쉬 계수로  $r$ 차 벡터이고,  $U = [U_0, U_1, \dots, U_{m-1}]$ 로  $u(t)$ 에 대한 월쉬 계수( $r \times m$ )행렬이다.

식(4.13)-(4.15)의 관계를 식(4.11)에 대입한 후 적분 연산 행렬을 도입하면 다음과 같다.

$$\begin{aligned} X^k Wal(t) &= X_0 Wal(t) \\ &+ AX^{k-1}P Wal(t) + BUP Wal(t) \\ &+ \sum_{i=0}^{m-1} N_i X^{k-1} \Lambda_{u_i} P Wal(t) \end{aligned} \quad (4.16)$$

단,  $P$ 는 시퀀스 배열 월쉬 함수 적분 연산 행렬

$$P = \begin{bmatrix} P_{m/2} & -\frac{1}{2m} R_{m/2} \\ \frac{1}{2m} R_{m/2} & 0_{m/2} \end{bmatrix} \quad (4.17)$$

식 (4.17)에서  $m$ 은 월쉬 함수 전개 항수이고, 식에서  $P$ 와  $R$ 은 정방 행렬이며, 첨자는 행렬의 크기를 나타낸다. 또한  $R$ 은 역대각 단위 행렬이다.

식 (4.16)에서 양변의 월쉬 벡터를 소거하면 다음의 식 (4.18)과 같게 되고,

$$\begin{aligned} X^k &= X_0 + AX^{k-1}P + BUP \\ &+ \sum_{i=0}^{m-1} N_i X^{k-1} \Lambda_{u_i} P \end{aligned} \quad (4.18)$$

이러한 대수 방정식으로부터  $x^k(t)$ 의 월쉬 계수 행렬인  $X^k$ 를 구할 수 있다. 그런데 적분연산행렬  $P$ 는 많은 0요소를 포함하고 있다. 월쉬 함수  $m (= 2^p)$ 항 전개를 이용하는 경우  $m^2$ 개의 요소 중  $2m-1$ 개의 요소만이  $\pm 2^{-(n+1)}$  ( $n=0, 1, \dots, p$ )의 값을 취하므로, 이러한 특성을 이용하면 식(4.16)의  $X^k$ 를 다음과 같이 구할 수 있다.

(1)  $i = 0$  인 경우

$$\begin{aligned} X_i^k &= X_{00} + A \sum_{j=0}^k \frac{1}{2^{j+1}} X_{2^j-1}^{k-1} \\ &+ B \sum_{j=0}^k \frac{1}{2^{j+1}} U_{2^j-1} \\ &+ \sum_{l=0}^{i-1} N_l \left[ \sum_{j=0}^k \frac{1}{2^{j+1}} X_{2^j-1}^{k-1} u_{l(2^j-1)} \right] \end{aligned} \quad (4.19)$$

(2)  $i = 2^{q-1}, 2^{q-1}+1, \dots, 2^q-1$

(단,  $q = 1, 2, \dots, p-1$ )인 경우

$$\begin{aligned} X_i^k &= -\frac{1}{2^{q+1}} [AX_{2^q-1-i}^{k-1} + BU_{2^q-1-i}] \\ &+ \sum_{j=q+1}^k \frac{1}{2^{j+1}} [AX_{2^j-1-i}^{k-1} + BU_{2^j-1-i}] \\ &+ \sum_{l=0}^{i-1} N_l \left[ -\frac{1}{2^{q+1}} X_{2^q-1-i}^{k-1} u_{l(2^q-1-i)} \right] \\ &+ \sum_{l=0}^{i-1} N_l \left[ \sum_{j=q+1}^k \frac{1}{2^{j+1}} X_{2^j-1-i}^{k-1} u_{l(2^j-1-i)} \right] \end{aligned} \quad (4.20)$$

(3)  $i = 2^{q-1}, 2^{q-1}+1, \dots, 2^q-1$  (단,  $q = p$ )인 경우

$$\begin{aligned} X_i^k &= -\frac{1}{2^{q+1}} [AX_{2^q-1-i}^{k-1} + BU_{2^q-1-i}] \\ &- \sum_{l=0}^{i-1} N_l \left[ \frac{1}{2^{q+1}} [X_{2^q-1-i}^{k-1} u_{l(2^q-1-i)}] \right] \end{aligned} \quad (4.21)$$

식(4.19)-(4.21)에서 결정된  $X^k$ 를 고속 월쉬 역변환하면  $x^k(t)$ 를 구할 수 있는데, 이전 단계에서 구한  $x^{k-1}(t)$ 를 비교하여 식(4.12)의 조건을 만족하는  $x^k(t)$ 가 식(4.1)의 해가 된다.

식 (4.9)에서 구한 유한 급수 전개 기법과, 식 (4.19)부터 식 (4.21)까지를 이용하여 구한 결과를 비

고속 월쉬 변환을 이용한 쌍일차계의 해석에 관한 연구

교하여 보면, 본 논문에서 제안한 고속 월쉬 변환을 사용함으로써 연산상의 많은 잇점을 얻을 수 있음을 알 수 있다.

## 5. 시뮬레이션

다음과 같이 주어지는 쌍일차계를 고려하여 본다.

$$\dot{x}(t) = Ax(t) + Bu(t) + N_0 x(t) u_0(t)$$

$$x(0) = x_0$$

$$\text{단, } A = \begin{bmatrix} -1 & 0 \\ 0 & -1.5 \end{bmatrix}, N_0 = \begin{bmatrix} 0 & 2 \\ 0.5 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$u(t) = \begin{bmatrix} u_0(t) \\ u_1(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

이 시스템에 대하여 직교 함수를 적용하여 구한 결과가 그림 5.1-그림 5.3까지 나타나 있다. 그림 5.1, 5.2, 5.3은 각각 월쉬 함수의 8항 전개, 16항 전개, 32항 전개에 의한 결과로서, 고속 월쉬 변환에 의한 상태 해석의 결과를 나타내고 있다.

직교 함수의 유한 급수 전개를 행하였을 경우에는, 그 전개 항수는 직교 함수가  $2^d$  개로 하나의 완전한 집합을 이루므로 16항, 32항, 64항 등으로 전개하여야 한다. 시스템이 2차이고 전개 항수를 64로 하면  $2 \times 64 = 128$  이 되어 결국에는  $128 \times 128$  행렬의

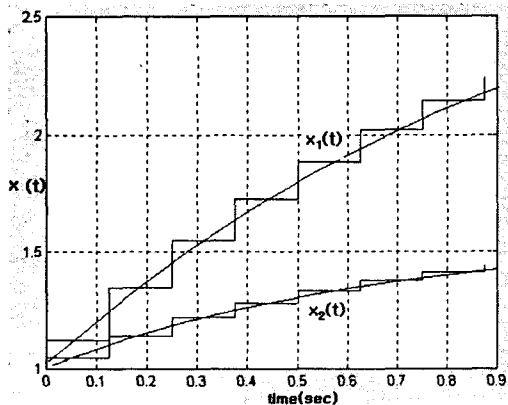


그림 5.1. 8항 전개에 의한 결과  
Fig. 5.1. The results of the expansion of Walsh function (m=8)

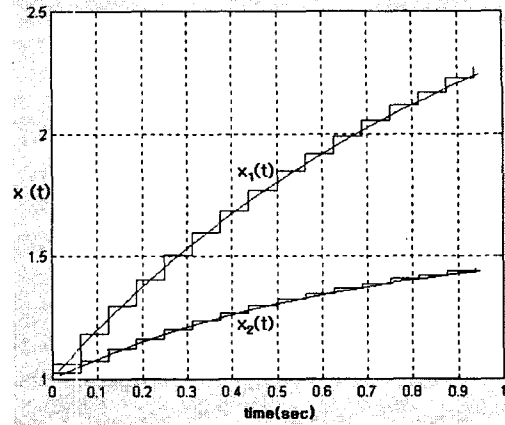


그림 5.2. 16항 전개에 의한 결과  
Fig. 5.2. The results of the expansion of Walsh function (m=16)

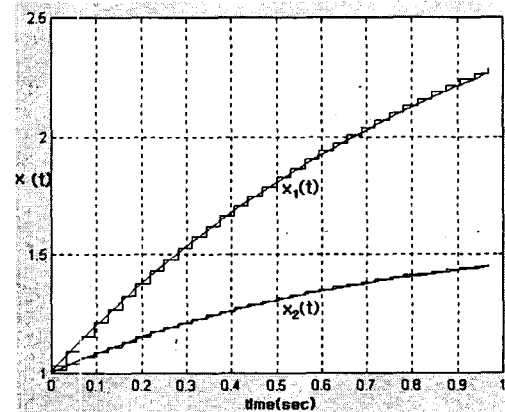


그림 5.3. 32항 전개에 의한 결과  
Fig. 5.3. The results of the expansion of Walsh function (m=32)

역변환이 따르게 되고, 이로 인하여 계산 시간이 느려지고 메모리의 용량을 많이 차지하게 된다. 그러나 고속 월쉬 변환을 이용하게 되면 반복 연산에 의하여 앞서 언급하였던 단점들을 극복할 수 있게 된다. 물론 전개 항수를 늘릴수록 정확한 결과를 얻을 수 있다.

## 6. 결론

본 연구에서는 직교 함수의 유한 급수 전개 방법과 고속 월쉬 변환 방법에 의한 쌍일차계의 상태 해

석 방법을 비교하여 제시하였으며, 수식을 통하여 직교함수의 유용성을 입증하였다.

직교 함수의 유한 급수 전개에 의한 방법은 잡음에 둔감하고 또 1회 계산에 의해 전 구간에 대한 정보를 파악할 수 있지만, 크로네커 곱에 의하여 고차 행렬의 역변환이 필요하게 되어 연산 시간이 길어지고, 오차의 누적을 피할 수 없게 된다.

이러한 문제를 해결하기 위하여 고속 월쉬 변환을 이용한 반복 연산 기법을 제시하였으며, 직교 함수의 유한 급수 전개 방법과 비교하여 볼 때 다음과 같은 사실을 알 수 있다.

1. 크로네커 곱에 의한 고차 행렬의 계산 없이도 원하는 시스템의 응답을 쉽고 간단하게 구할 수 있다.

2. 반복 연산 기법을 사용하여 연산 행렬에 포함되어 있는 많은 0요소들을 연산에서 제외시킬 수 있으므로 결국 연산의 부담을 줄일 수 있다.

References

[1] W. L. Chen and B. S. Jeng, "Analysis of piecewise constant delay systems via block-pulse functions", *Int. J. Systems Sci.*, Vol. 12, no.5, pp. 625-633, 1981.  
 [2] P. R. Roesser and M. E. Jernigan "Fast Haar transform algorithms", *IEEE Trans. Computers*, Vol. c-31, no.2, pp. 175-177, 1982.  
 [3] 안두수, 심재선, 이명규 "Walsh 급수 전개에 의한 분포 정수계의 해석에 관한 연구", *전기학회 논문집*, Vol. 35, no.3, pp. 95-101, 1986.

[4] V. R. Karanam and P. A. Frick "Bilinear system identification by Walsh functions", *IEEE. Trans. Automatic control*, Vol. AC-23, no.4, pp. 709-713, Aug., 1978.  
 [5] K. G. Palanisamy "Analysis of nonlinear systems via single term Walsh series approach", *Int. J. System Science* Vol. 13, pp. 929-935, 1982.  
 [6] 안두수, "직교 함수와 응용", 복두출판사, 1997.  
 [7] Richard J. Weider and R. J. Mulholland "Kronecker product representation for the solution of the general linear matrix equation", *IEEE. Trans. Automatic Control*, Vol. AC-25, no.3, pp. 563-564, 1980.  
 [8] J. P. Matuszewski "Suboptimal Feedback Control of Non-stationary Nonlinear Systems", *IEEE. Trans. Automatic Control*, Vol. 18, pp. 271-274, 1973.  
 [9] J. D. Pearson "Optimization Methods for Large-scale Systems", (Edited by D. A. Wismer). McGraw-Hill, New York, 1970.  
 [10] 박준훈 "블럭 펄스 변환에 의한 비선형계의 최적제어를 위한 새로운 접근 방법에 관한 연구", *성균관대학교 박사학위논문*, 1991.

◇ 저자소개 ◇

김 태 훈 (金泰勳)  
 1971년 2월 1일 생. 1995년 성균관대학교 공대 전기공학과 졸업. 1997년 성균관대학교 대학원 전기공학과 졸업(석사). 2002년 동 대학원 전기전자 및 컴퓨터 공학과 졸업(박사). 2002~현재 한국정보보호진흥원 선임연구원.

심 재 선 (沈在先)  
 1950년 7월 23일생. 1973년 성균관대학교 공대 전기공학과 졸업. 1979년 단국대학교 대학원 전기공학과 졸업(석사). 1989년 성균관대학교 대학원 전기공학과 졸업(박사). 1975~현재 삼척대학교 전기공학과 교수.