

르장드르 웨이블릿을 이용한 선형 시불변 시스템의 효율적 수치 해석 방법

An Efficient Computational Method for Linear Time-invariant Systems via Legendre Wavelet

김 범 수*
(Beonsoo Kim¹)

¹Mechanical System Engineering, Gyeongsang National University

Abstract: In this paper Legendre wavelets are used to approximate the solutions of linear time-invariant system. The Legendre wavelet and its integral operational matrix are presented and an efficient algorithm to solve the Sylvester matrix equation is proposed. The algorithm is based on the decomposition of the Sylvester matrix equation and the preorder traversal algorithm. Using the special structure of the Legendre wavelet's integral operational matrix, the full order Sylvester matrix equation can be solved in terms of the solutions of pure algebraic matrix equations, which reduce the computation time remarkably. Finally a numerical example is illustrated to demonstrate the validity of the proposed algorithm.

Keywords: linear time-invariant system, Legendre wavelet, orthogonal function, Sylvester matrix equation

I. 서론

다중해상도, 직교성 및 반침(support) 특성 등을 갖고 있는 웨이블릿은 공학, 물리학, 수학 등 다양한 분야에서 많이 연구 및 응용 되고 있다[1]. 시스템의 신호를 해석하는 방법 중 하나는 잘 정의된 직교 기저 함수를 이용하여 신호를 분석하고 재구성하는 것이다. 웨이블릿은 매우 잘 정의된 직교함수를 기반으로 하고 있어서 동적 시스템, 미분 방정식, 편미분 방정식 등에서 주어진 함수를 웨이블릿 직교 기저 함수의 선형 결합으로 근사화 하여 해석하는 여러 가지 방법에 많이 적용되고 있다[2]. 이 방법들의 장점은 주어진 웨이블릿 기저 함수를 적분 또는 미분했을 때 적분 또는 미분된 함수를 다시 그 기저함수로 표현함으로써 동적 문제를 대수 행렬 방정식 문제로 변환하여 해석하는 편리성 및 효율성에 있다.

일반적으로 직교 함수들은 크게 세 개의 그룹으로 분류된다. 첫 번째 그룹은 부분 상수(piecewise constant) 함수들로서 Haar 함수, Walsh 함수, 블록 펄스(block pulse) 함수 등이 이에 속한다. 두 번째 그룹은 르장드르(Legendre) 다항식, Laguerre 다항식, Chebyshev 다항식 등이 포함된 직교 다항식(orthogonal polynomial) 그룹이며, 세 번째 그룹은 Fourier 함수, Hartley 함수 등 sine-cosine 함수들로 이루어져 있다[3]. 이중 특히 Haar 함수를 기반으로 한 Haar 웨이블릿은 동적 시스템의 해석, 최적제어 등의 문제를 해결하고자 하는 연구에 많이 적용되어 왔다[2,4,5]. Haar 웨이블릿을 이용한 방법도 주어진 시간 함수를 웨이블릿 직교 기저 함수와 이 함수를 적분한 것으로 재구성함으로써 미분 또는 편미분 방정식을 Sylvester 행렬 방정식인 대수 행렬 방정식 문제로 변환한 후

해를 근사적으로 구하는 것으로 좀 더 정확한 해를 구하기 위해서는 Haar 함수 행렬의 차원을 증가시켜야 하며 이는 계산량 증가 및 연산의 복잡성을 동반하는데 이를 효과적으로 해결하는 연구도 진행되어 왔다[4,5].

르장드르 다항식을 척도함수로 사용하여 구성된 르장드르 웨이블릿을 이용한 동적 시스템 해석 연구도 많이 진행되어 왔다[6]. 이 방법 역시 르장드르 웨이블릿 적분 연산 행렬(operational matrix of integration)을 이용하여 르장드르 웨이블릿 기저 함수를 적분했을 때 이를 다시 르장드르 웨이블릿 기저 함수로 표현함으로써 동적 시스템을 대수 행렬 방정식 문제로 변환하여 해를 근사적으로 구한다. Haar 웨이블릿을 사용한 방법과 마찬가지로 좀 더 정확한 해를 구하기 위해서는 웨이블릿의 해상도를 증가시켜야 하며 이는 Sylvester 행렬 방정식의 차원이 매우 커지며 이를 일반적인 Kronecker 곱의 역행렬을 이용해서 해를 구할 수 있지만 이는 계산량 증가 및 연산의 복잡성을 수반하게 된다. 따라서 변환된 대수 행렬방정식의 해를 단순하고 효율적으로 구할 수 있는 방법에 대한 필요성이 지적되고 있다[7].

본 연구에서는 선형 시불변 시스템을 직교 다항식 함수를 이용한 르장드르 웨이블릿 이용하여 상태 방정식을 적분하여 주어진 미분 방정식을 Sylvester 행렬 방정식으로 변환한 후 해를 구하는데 있어서 르장드르 웨이블릿의 적분 연산 행렬 구조적 특성을 사용하여 해를 효율적으로 구하는 방법을 제시한다. 즉, 변환된 Sylvester 행렬 방정식을 저차의 행렬 방정식들로 분할하는 과정은 이진 트리 구조 형태를 갖게 되며 각 노드 역시 Sylvester 행렬 방정식 구조를 갖는다. 이 트리 구조에서 전체 해는 전위 순회(preorder traversal) 알고리즘으로 구할 수 있다. 최종 분할된 말단 노드도 르장드르 웨이블릿 적분 연산 행렬 구조 때문에 Sylvester 방정식 구조를 갖게 된다. 따라서 말단 노드로부터 해를 구하기 위해서는

* 책임저자(Corresponding Author)

Manuscript received April 10, 2013 / revised May 31, 2013 / accepted June 7, 2013

김범수: 경상대학교 기계시스템공학과(kimbs@gnu.ac.kr)

Sylvester 행렬 방정식을 풀어야 하는데 실제 해에 매우 근접한 해를 구하기 위해서는 르장드르 웨이블릿의 해상도를 높여야 하고 이는 매우 많은 계산량을 필요로 한다. 여기서는 적분 연산 행렬 구조를 이용하여 말단 노드의 Sylvester 행렬 방정식을 웨이블릿에 사용된 르장드르 다항식의 차수가 M 이라 할 때 $M+1$ 개의 단순 대수 행렬 방정식으로 표현할 수 있음을 보임으로써 해를 해석적으로 구할 수 있는 효율적인 방법을 제시한다.

II. 르장드르 웨이블릿

르장드르 웨이블릿은 르장드르 다항식을 척도함수 (scaling function)로 사용하여 이 함수들을 척도구성 (dilation)과 평행 이동(translation)하여 구성한 기저 함수들의 집합이다. 르장드르 웨이블릿은 식 (1)과 같이 정의된다[8].

$$\phi_{nm}(t) = \begin{cases} \sqrt{m+\frac{1}{2}} 2^{\frac{K}{2}} P_m(2^K t - 2n + 1), & \frac{2n-2}{2^K} \leq t < \frac{2n}{2^K} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

여기서 $P_m(t)$ 는 차수가 m 인 르장드르 다항식이고, K 는 양의 정수 값을 갖는 척도 파라미터며, $n=1,2,\dots,2^{K-1}$ 은 평행 이동 구간을 나타낸다. 척도함수로 사용할 르장드르 다항식의 최대 차수가 M 이면 $m=0,1,\dots,M$ 이다.

르장드르 다항식은 다음과 같이 재귀적으로 정의 되며[8] 집합 $\{P_m(t) | m=1,2,3,\dots\}$ 은 힐베르트 공간 $L_2[-1,1)$ 에서 완전 직교(complete orthogonal) 이다.

$$\begin{aligned} P_0(t) &= 1, \\ P_1(t) &= t, \\ P_{m+1}(t) &= \frac{2m+1}{m+1} t P_m(t) - \frac{m}{m+1} P_{m-1}(t), \\ m &= 1, 2, 3, \dots \end{aligned} \quad (2)$$

구간 $[0,1)$ 에서 적분 가능한 함수 $f(t)$ 는 직교 기저 함수인 르장드르 웨이블릿으로 다음과 같이 재구성할 수 있다.

$$f(t) = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} f_{nm} \phi_{nm}(t), \quad (3)$$

여기서 계수는 $f_{nm} = \langle f(t), \phi_{nm}(t) \rangle$ 이고 $\langle \cdot, \cdot \rangle$ 는 함수의 내적을 나타낸다. 식 (3)에서 무한 급수로 전개된 함수 $f(t)$ 를 근사화 하면

$$f(t) = \sum_{n=1}^{2^{K-1}} \sum_{m=0}^M f_{nm} \phi_{nm}(t) = \mathbf{F}^T \boldsymbol{\Psi}(t) \quad (4)$$

와 같이 르장드르 웨이블릿의 선형 결합으로 나타낼 수 있다. 여기서

$$\boldsymbol{\Psi}(t) = [\underbrace{\phi_{10}(t) \phi_{11}(t) \dots \phi_{1M}(t)}_{n=1} \quad \underbrace{\phi_{20}(t) \phi_{21}(t) \dots \phi_{2M}(t)}_{n=2} \quad \dots \quad \underbrace{\phi_{2^{K-1}0}(t) \phi_{2^{K-1}1}(t) \dots \phi_{2^{K-1}M}(t)}_{n=2^{K-1}}]^T \quad (5)$$

$$\mathbf{F} = [f_{10} \ f_{11} \ \dots \ f_{1M} \ f_{20} \ f_{21} \ \dots \ f_{2M} \ \dots \ f_{2^{K-1}0} \ f_{2^{K-1}1} \ \dots \ f_{2^{K-1}M}]^T$$

이며 계수 벡터 \mathbf{F} 는 르장드르 웨이블릿이 직교 기저함수이므로 유일하게 구해진다. 식 (5)의 기저함수를 적분하여 이를

다시 르장드르 웨이블릿 기저함수로 표현하면 다음과 같다 [9].

$$\int \boldsymbol{\Psi}(\tau) d\tau = \mathbf{P}_j \boldsymbol{\Psi}(t). \quad (6)$$

식 (6)에서 적분연산행렬 \mathbf{P}_j 은 $2^{K-1}(M+1)$ 차원인 블록 상부 삼각 행렬 구조를 갖는다.

$$\mathbf{P}_j = \frac{1}{2^K} \begin{bmatrix} \mathbf{D}_{M+1} & \mathbf{U}_{M+1} & \mathbf{U}_{M+1} & & \mathbf{U}_{M+1} \\ & \mathbf{D}_{M+1} & \mathbf{U}_{M+1} & & \mathbf{U}_{M+1} \\ & & \mathbf{D}_{M+1} & & \mathbf{U}_{M+1} \\ & & & \ddots & \\ & & & & \mathbf{D}_{M+1} \end{bmatrix} \quad (7)$$

여기서 $j=2^{K-1}(M+1)$ 이고, 블록 행렬 \mathbf{D}_{M+1} 과 \mathbf{U}_{M+1} 은 다음과 같이 주어진다[7]

$$\begin{aligned} \mathbf{D}_{M+1} &= \begin{bmatrix} \frac{1}{2} & \frac{1}{\sqrt{5}} & 0 & & & \\ \frac{-1}{\sqrt{5}} & 0 & \frac{1}{\sqrt{5}} & & & \\ 0 & \frac{1}{\sqrt{5}} & 0 & & & \\ & & & \ddots & & \\ & & & & -\alpha & 0 & \beta \\ & & & & & -\beta & 0 \end{bmatrix}, \\ \alpha &= \frac{1}{\sqrt{4M^2 - 16M + 15}}, \quad \beta = \frac{1}{\sqrt{4M^2 - 8M + 3}} \quad (8) \\ \mathbf{U}_{M+1} &= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ & \ddots & \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

III. 선형 시불변 시스템 해석 알고리즘

다음과 같은 선형 시불변 시스템을 고려하자.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad (9)$$

여기서 $\mathbf{x}(t) \in R^{n \times 1}$, $\mathbf{u}(t) \in R^{r \times 1}$ 는 각각 시스템의 상태 및 입력 변수이다. 상태 변수와 입력 변수가 $L_2[0,1)$ 공간에서 적분 가능하면 르장드르 웨이블릿 기저함수를 이용해서 다음과 같이 근사적으로 표현할 수 있다.

$$\begin{aligned} \mathbf{x}(t) &\approx \mathbf{X}\boldsymbol{\Psi}(t), \\ \mathbf{u}(t) &\approx \mathbf{U}\boldsymbol{\Psi}(t). \end{aligned} \quad (10)$$

일반적으로 직교함수를 이용하여 미분 또는 편미분 방정식 문제를 대수 행렬 방정식 문제로 전환하기 위해서는 주어진 미분 방정식을 적분한 후 적분 연산 행렬을 적용한다. 여기서도 식 (9)를 적분하면

$$\mathbf{x}(t) - \mathbf{x}(0) = \mathbf{A} \int_0^t \mathbf{x}(\tau) d\tau + \mathbf{B} \int_0^t \mathbf{u}(\tau) d\tau \quad (11)$$

와 같이 되고, 여기에 (6), (10)을 대입하여 전개하면 다음과 같다.

$$\mathbf{X}\boldsymbol{\Psi}(t) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}\mathbf{P}_j\boldsymbol{\Psi}(t) + \mathbf{B}\mathbf{U}\mathbf{P}_j\boldsymbol{\Psi}(t) \quad (12)$$

여기서 $\mathbf{U} \in R^{r \times j}$, $\mathbf{P}_j \in R^{j \times j}$ 이고 $\mathbf{X} \in R^{n \times j}$ 는 구해야 할 미

지 행렬이다. $\mathbf{x}(0) = \mathbf{X}_0 \Psi(t)$ 와 같이 초기 벡터를 르장드르 웨이블릿 기저함수로 전개한 후 이를 (12)식에 대입한 후 공통 요소인 $\Psi(t)$ 를 없애면 (12)식은 다음과 같이 Sylvester 행렬 방정식으로 표현된다.

$$\mathbf{X} = \mathbf{A}\mathbf{X}\mathbf{P}_j + \mathbf{X}_0 + \mathbf{B}\mathbf{U}_j \quad (13)$$

따라서 (9)로 표현된 상태방정식의 문제는 대수 행렬 방정식을 이용해서 근사적으로 해를 구하는 문제로 귀결된다. 지금도 Sylvester 행렬 방정식의 해를 효율적으로 구하는 연구는 중요 과제이지만[9], 일반적으로 (13)으로 표현된 Sylvester 행렬 방정식의 해는 다음과 같이 Kronecker 곱을 이용하여 구하면 다음과 같다[10].

$$\text{vec}(\mathbf{X}) = (\mathbf{I} - \mathbf{P}_j^T \otimes \mathbf{A})^{-1} \text{vec}(\mathbf{X}_0 + \mathbf{B}\mathbf{U}_j), \quad (14)$$

여기서 $\text{vec}(\cdot)$ 은 다음과 같이 주어진 행렬의 열을 순차적으로 쌓아 벡터로 표현하는 연산자이다.

$$\text{vec}(\mathbf{X}) = [x_{11} \ x_{21} \ \dots \ x_{n1} \ x_{12} \ x_{22} \ \dots \ x_{n2} \ \dots \ x_{1m} \ x_{2m} \ \dots \ x_{nm}]^T.$$

르장드르 다항식의 차수가 증가하거나 웨이블릿 스케일링 파라미터가 증가하면 (14) 식의 해를 구하기 위해서는 매우 큰 역행렬 계산이 필요하고 이는 연산량, 계산 속도, 수치적 문제 등을 야기할 수 있다.

본 연구에서는 르장드르 웨이블릿 적분 연산 행렬이 블록 상부 삼각 행렬 구조를 이용하여 전위 순회 알고리즘을 사용하여 계산량 감소 및 연산 속도를 증가 시킬 수 있는 효율적인 방법을 제시한다.

식 (13)에서 $\mathbf{F} = \mathbf{X}_0 + \mathbf{B}\mathbf{U}_j$ 으로 놓으면 다음과 같이 간략히 표현할 수 있다.

$$\mathbf{X} = \mathbf{A}\mathbf{X}\mathbf{P}_j + \mathbf{F}, \quad (15)$$

여기서

$$\mathbf{X} = [\mathbf{X}_{1,1} \ \mathbf{X}_{1,2}], \quad \mathbf{P}_j = \begin{bmatrix} \mathbf{P}_{j,1} & \mathbf{U}_{j,1} \\ \mathbf{0} & \mathbf{P}_{j,2} \end{bmatrix}, \quad \mathbf{F} = [\mathbf{F}_{1,1} \ \mathbf{F}_{1,2}]$$

와 같이 행렬을 동일한 크기로 분할하면 식 (15)은 다음과 같이 2개의 저차화된 Sylvester 행렬 방정식으로 분할 된다.

$$\mathbf{X}_{1,1} = \mathbf{A}\mathbf{X}_{1,1}\mathbf{P}_{j,1} + \mathbf{F}_{1,1}, \quad (16-a)$$

$$\mathbf{X}_{1,2} = \mathbf{A}\mathbf{X}_{1,1}\mathbf{U}_{j,1} + \mathbf{A}\mathbf{X}_{1,2}\mathbf{P}_{j,2} + \mathbf{F}_{1,2}. \quad (16-b)$$

먼저 식 (16-a) 에서 $\mathbf{X}_{1,1}$ 를 구한 다음 이 값을 (16-b)에 대입한 후 $\mathbf{X}_{1,2}$ 를 구함으로써 전체 \mathbf{X} 를 구할 수 있다. 그런데 (16)의 $\mathbf{P}_{j,2}$ 도 블록 상부 삼각 행렬이므로 (16-a) 및 (16-b) 도 각각 차수가 반으로 줄어든 Sylvester 행렬 방정식들로 분리된다. 이 과정을 반복하면 최종적으로 다음과 같은 행렬 방정식들로 분리된다.

$$\mathbf{X}_{k,2i-1} = \frac{1}{2^k} \mathbf{A}\mathbf{X}_{k,2i-1} \mathbf{D} + \mathbf{F}_{k,2i-1}, \quad (17-a)$$

$$\mathbf{X}_{k,2i} = \frac{1}{2^k} \mathbf{A}\mathbf{X}_{k,2i-1} \mathbf{U} + \frac{1}{2^k} \mathbf{A}\mathbf{X}_{k,2i} \mathbf{D} + \mathbf{F}_{k,2i}. \quad (17-b)$$

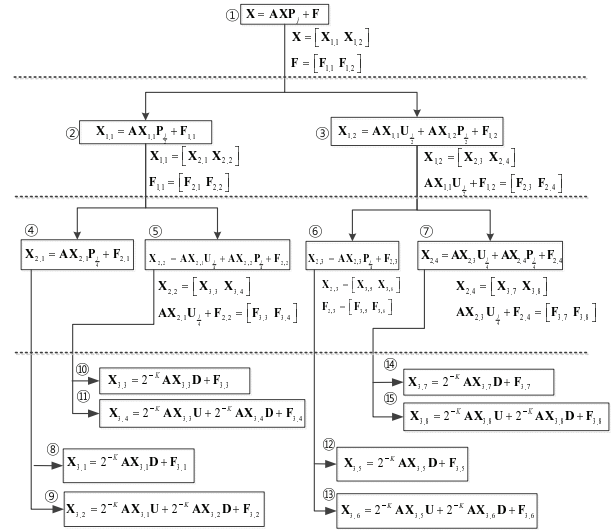


그림 1. Sylvester 행렬 방정식 분할.

Fig. 1. Decomposition of Sylvester matrix equation.

여기서 첫 번째 첨자 $k = 2^{K-1} - 1$ 는 분할된 단계이며, 두 번째 첨자는 각 분할 단계에서 전체 행렬에서 $2i-1, 2i, (i=1,2,\dots,2^{K-1})$ 번째에 해당되는 블록 행렬 인덱스이다.

예를 들어 $K=3$ 인 경우 이 분리 과정을 그림 1에 도시하였다.

식 (15)의 $\mathbf{X}, \mathbf{P}_j, \mathbf{F}$ 의 분할은 ①→②→④→⑧→⑨→⑤→⑩→⑪→③→⑥→⑫→⑬→⑦→⑭→⑮ 순서로 전위 순회를 하면서 최종 분할 단계에서는 $\mathbf{X}_{k,j}, k = 2^{K-1} - 1, j = 1, 2, \dots, 2^{K-1}$ 를 구하고 상위 단계 중 홀수 노드에서는 구해진 $\mathbf{X}_{k,j}$ 를 사용하여 $\mathbf{A}\mathbf{X}_{k,i}\mathbf{U}_{j,1} + \mathbf{F}_{k,j}$ 를 갱신하여 다시 하위 노드로 순회할 때 이 부분을 분할한다.

말단 단계인 ⑧~⑮에서 $\mathbf{X}_{k,j} \in R^{n \times M+1}$ 를 구하기 위해서는 Sylvester 행렬 방정식을 풀어야 한다. 여기서는 행렬 \mathbf{D}, \mathbf{U} 의 구조를 이용하여 단순 대수 행렬방정식으로 구할 수 있는 방법을 제시한다. (7)에 있는 적분연산 행렬의 부분 행렬인 \mathbf{U} 는 (1,1) 요소만 0이 아니고 나머지 요소가 모두 0인 단일 요소 행렬(single entry matrix) 이므로 행렬의 곱이 매우 간단해진다. 행렬 \mathbf{D} 는 (1,1) 요소를 제외하면 반대칭행렬(skew-symmetric matrix)이고, 특히 대각요소의 위아래 부분만 0이 아닌 값을 갖는다. 따라서 마지막 열은 1개의 요소만 0이 아니고, 그 외 열들은 2개의 요소만 0이 아니다. 따라서 동일한 구조를 갖는 ⑧~⑮의 Sylvester 행렬 방정식들은 다음 방법에 의해 해석적으로 구할 수 있다.

편의상 (17)에서 $\mathbf{x} := \mathbf{X}_{k,2i-1}, \mathbf{f} := \mathbf{F}_{k,2i-1}, \{\mathbf{x}, \mathbf{f}\} \in R^{n \times M+1}$ $\mathbf{A}_k := \frac{1}{2^k} \mathbf{A}$ 로 간략히 표현하고 \mathbf{x}, \mathbf{f} 를 열벡터로 전개하면 다음과 같다.

$$[\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_{M+1}] = \mathbf{A}_k [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_{M+1}] \mathbf{D} + [\mathbf{f}_1 \ \mathbf{f}_2 \ \dots \ \mathbf{f}_{M+1}]. \quad (18)$$

따라서 식 (18)은 다음과 같이 $M+1$ 개의 단순 대수 행렬 방정식들로 분할 된다.

$$\begin{aligned}
 \mathbf{x}_1 &= \mathbf{A}_K (\mathbf{x}_1 - a_2 \mathbf{x}_2) + \mathbf{f}_1, \\
 \mathbf{x}_2 &= \mathbf{A}_K (a_2 \mathbf{x}_1 - a_3 \mathbf{x}_3) + \mathbf{f}_2, \\
 &\vdots \\
 \mathbf{x}_i &= \mathbf{A}_K (a_i \mathbf{x}_{i-1} - a_{i+1} \mathbf{x}_{i+1}) + \mathbf{f}_i, \\
 &\vdots \\
 \mathbf{x}_M &= \mathbf{A}_K (a_M \mathbf{x}_{M-1} - a_{M+1} \mathbf{x}_{M+1}) + \mathbf{f}_M, \\
 \mathbf{x}_{M+1} &= a_M \mathbf{A}_K \mathbf{x}_M + \mathbf{f}_{M+1}, \\
 a_k &= (4k^2 - 8k + 3)^{-\frac{1}{2}}.
 \end{aligned} \tag{19}$$

(19)의 첫 번째 식에서 \mathbf{x}_1 을 두 번째 식에 대입하여 \mathbf{x}_2 를 \mathbf{x}_3 변수로만 표현하고 이를 다시 세 번째 식에 대입하는 과정을 진행하면 맨 마지막 식은 \mathbf{x}_{M+1} 변수 자신만으로 표현된다. 이 과정을 정리하여 다시 쓰면 다음과 같다.

$$\begin{aligned}
 \mathbf{x}_1 &= -a_2 \mathbf{k}_1 \mathbf{A}_K \mathbf{x}_2 + \mathbf{k}_1 \mathbf{g}_1, \\
 \mathbf{x}_2 &= -a_3 \mathbf{k}_2 \mathbf{A}_K \mathbf{x}_3 + \mathbf{k}_2 \mathbf{g}_2, \\
 &\vdots \\
 \mathbf{x}_i &= -a_{i+1} \mathbf{k}_i \mathbf{A}_K \mathbf{x}_{i+1} + \mathbf{k}_i \mathbf{g}_i, \\
 &\vdots \\
 \mathbf{x}_M &= -a_{M+1} \mathbf{k}_M \mathbf{A}_K \mathbf{x}_{M+1} + \mathbf{k}_M \mathbf{g}_M, \\
 \mathbf{x}_{M+1} &= \mathbf{k}_{M+1} \mathbf{g}_{M+1}.
 \end{aligned} \tag{20}$$

여기서 \mathbf{k}_i 와 \mathbf{g}_i 는 다음과 같이 각각 재귀적으로 구해진다.

$$\begin{aligned}
 \mathbf{k}_1 &:= (\mathbf{I} - \mathbf{A}_K)^{-1}, \\
 \mathbf{k}_i &:= (\mathbf{I} + a_i^2 \mathbf{A}_K \mathbf{k}_{i-1} \mathbf{A}_K)^{-1}, \quad 2 < i \leq M+1.
 \end{aligned}$$

그리고

$$\begin{aligned}
 \mathbf{g}_1 &= \mathbf{f}_1, \\
 \mathbf{g}_i &= a_i \mathbf{A}_K \mathbf{k}_{i-1} \times \mathbf{g}_{i-1} + \mathbf{f}_i, \quad i \geq 2.
 \end{aligned}$$

이다. 예를 들어 $\mathbf{g}_3 = a_3 \mathbf{A}_K \mathbf{k}_2 (a_2 \mathbf{A}_K \mathbf{k}_1 \mathbf{f}_1 + \mathbf{f}_2) + \mathbf{f}_3$ 와 같이 구해진다. 따라서 (20)에서 맨 뒤 열벡터인 \mathbf{x}_{M+1} 부터 \mathbf{x}_1 까지 역순으로 \mathbf{x}_i 를 구한 것을 배열하면 각 말단 노드에서의 행렬 $[\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_{M+1}]$ 를 해석적으로 쉽게 구할 수 있다.

IV. 수치 예

직교 다항식에 바탕을 둔 르장드르 웨이블릿 방법과 부분 상수 함수 그룹에 속한 Haar 웨이블릿 방법을 비교하기 위해서 [5]에서 사용된 선형 시스템을 고려하자.

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -20 & -75 \\ 1 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t),$$

여기서 초기치는 $\mathbf{x}_0 = [1 \ 0.5]^T$ 이다. 단위계단 입력을 인가하였을 때 이 시스템의 해석적 해는 다음과 같이 구해진다.

$$\begin{aligned}
 x_1(t) &= \frac{103}{20} e^{-15t} - \frac{83}{20} e^{-5t}, \\
 x_2(t) &= -\frac{107}{300} e^{-15t} + \frac{83}{100} e^{-5t} + \frac{1}{75}.
 \end{aligned}$$

[5]에서 사용된 Haar 웨이블릿을 이용한 모의 실험 데이터는 해상도를 $J=5$ 로 하면 32개의 이산 데이터를 갖으며 르

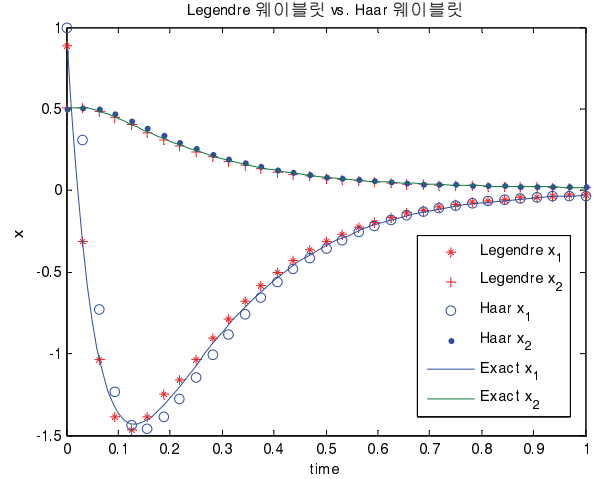


그림 2. M=3, K=3 일 때 상태 변수 궤적.
Fig. 2. State trajectories at M=3, K=3.

장드르 웨이블릿 방법에서도 동일한 이산 데이터를 갖기 위해서 M=3, K=3으로 했을 때의 결과는 그림 2와 같다. 그림 2에서 실선은 해석적 해에 대한 궤적이며 작은 원과 *는 각각 $x_1(t)$ 에 대한 Haar 웨이블릿 결과 및 르장드르 웨이블릿 결과이며, 작은 점과 + 기호는 $x_2(t)$ 에 대한 Haar 웨이블릿 결과 및 르장드르 웨이블릿 결과이다.

그림 3, 4는 정확한 해에 대한 르장드르 웨이블릿 방법과 Haar 웨이블릿 방법과의 오차이다. 그림 4에서 상태 값의 변화가 크지 않을 때는 르장드르 웨이블릿 방법과 Haar 웨이블릿 방법이 유사한 결과를 준다. 그림 3에서는 두 방법간의 차이가 보이는데 이는 해상도가 낮은 상태에서는 부분 상수 함수를 이용했을 때보다는 상대적으로 매끈한 부분을 포함하고 있는 직교 다항식 그룹이 상태 변화가 심한 부분에서 약간 낮은 결과를 준다고 사료된다.

그림 5는 M=5, K=5 일 때 수치 해석 결과이다. 척도 파라미터 K와 르장드르 다항식의 차수 M을 증가시키면 정확한 해에 매우 근접함을 오차를 도시한 그림 6, 7에서 볼 수

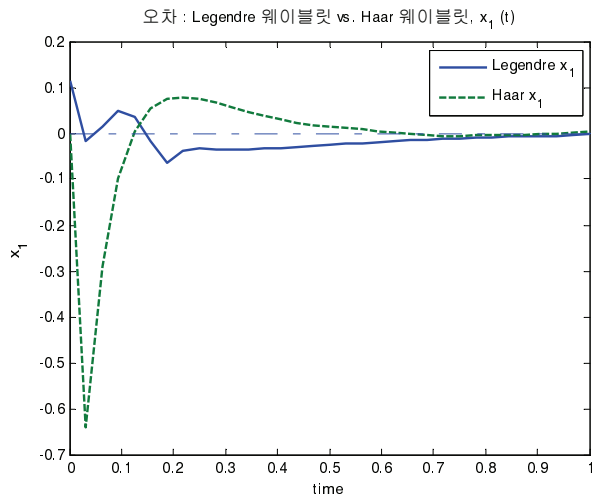


그림 3. M=3, K=3 일 때 상태 변수 $x_1(t)$ 의 오차.
Fig. 3. Error of $x_1(t)$ at M=3, K=3.

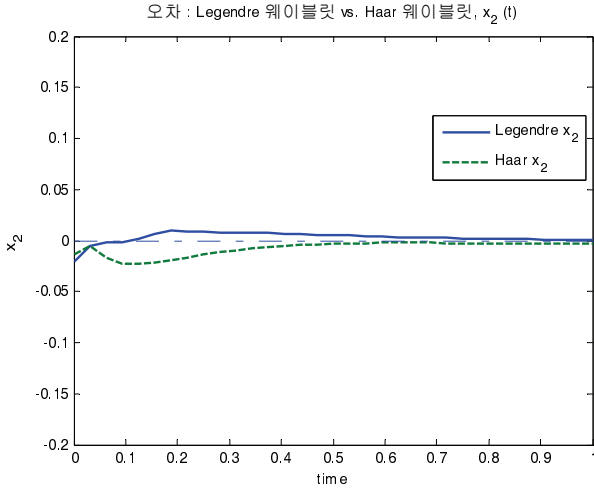


그림 4. $M=3, K=3$ 일 때 $x_2(t)$ 의 오차.
Fig. 4. Error of $x_2(t)$ at $M=3, K=3$.

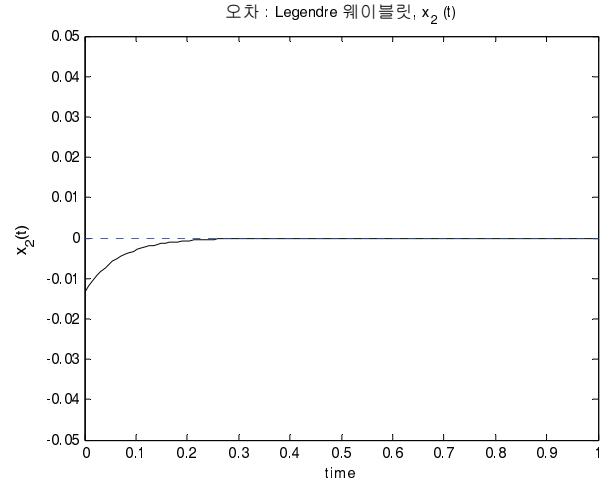


그림 7. $M=5, K=5$ 일 때 상태 변수 $x_2(t)$ 의 근사오차.
Fig. 7. Approximation error of $x_2(t)$ at $M=5, K=5$.

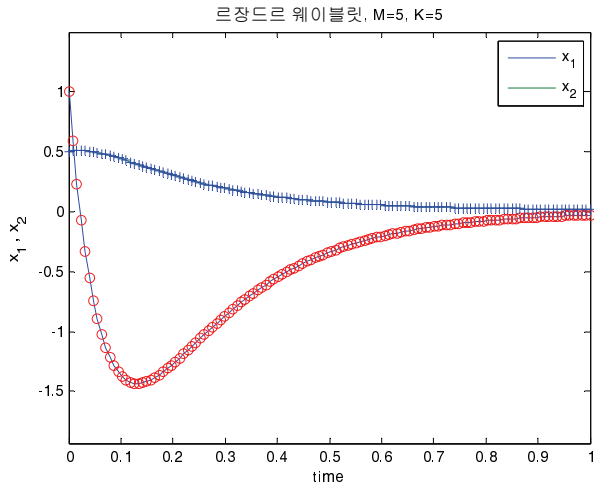


그림 5. $M=5, K=5$ 일 때 상태 변수 궤적.
Fig. 5. State trajectories at $M=5, K=5$.

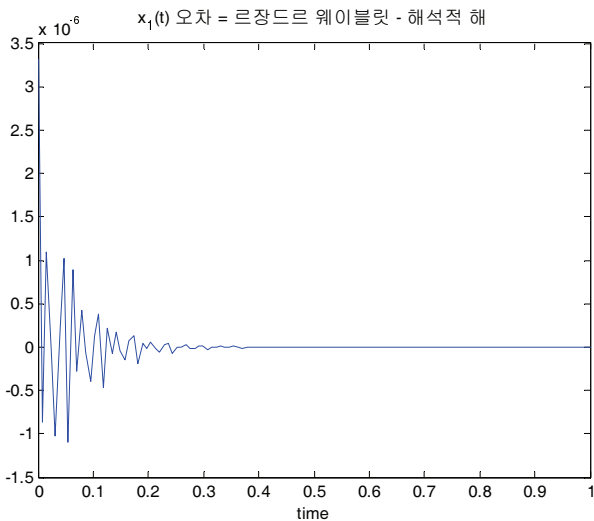


그림 6. $M=5, K=5$ 일 때 상태 변수 $x_1(t)$ 의 근사오차.
Fig. 6. Approximation error of $x_1(t)$ at $M=5, K=5$.

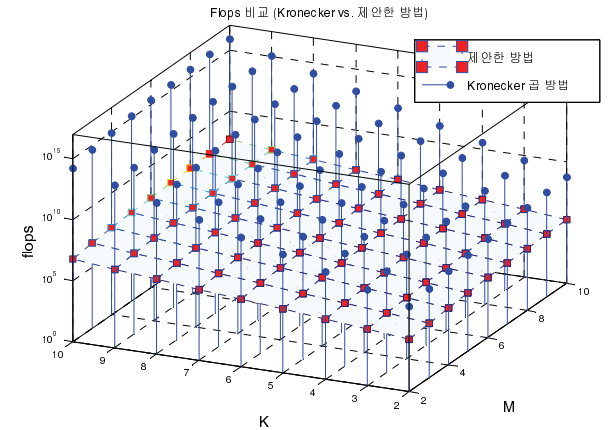


그림 8. Kronecker 곱을 이용한 방법과 제안한 방법에서의 flop 수 3차원 그림.
Fig. 8. 3D Log plot of flop counts for the Kronecker product method and the proposed method.

있다. 즉 그림 3, 4와 비교해서 그림 6, 7에서 오차가 $x_1(t)$ 변수 경우에는 10^{-6} 정도 줄어들음을 볼 수 있다. 그리고 상태 값의 변동이 상대적으로 큰 $x_2(t)$ 의 경우에는 정상상태에서 오차가 많이 줄어들음을 볼 수 있다.

Sylvester 행렬 방정식 (15)를 Kronecker 곱을 이용해서 구할 때 flop는 $O(m^3n^3 + m^2n^2)$, ($m = 2^{K-1}(M+1)$) 이고 제안된 방법에서는 $O(2^{K-1}((M+1)(3n^3 - 2n^2) + M(2n^3 + (4M+2)n^2 - (M+1)2n) + 2(n^3 - n^2)))$ 이다.

그림 8은 이 두 가지 경우에 대해 상태 변수를 4차로 하고, 르장드르 웨이블릿의 해상도는 입력 변수로서 르장드르 다항식의 차수, 척도 파라미터에 영향을 받으므로 이 둘을 반영하여 르장드르 다항식의 차수는 2차에서 10차까지, 각 웨이블릿에서 척도 파라미터는 2에서 10까지 고려했을 때의 flop 수이다. 그림 8에서 원으로 표시된 데이터는 주어진 M과 K에서의 Kronecker 곱을 이용했을 때의 flop 수이고, 바로 밑에 네모로 표시된 데이터는 제안된 방법을 사용했을 때의 flop 수이다. 르장드르 다항식의 차수 또는 척도 파라미터

가 증가할수록 양쪽 방법 모두 flop 수도 매우 크게 증가하지 만(flop 축은 log scale 임) 상대적으로 제안된 방법이 훨씬 적은 계산량을 보이고 있다.

V. 결론

선형 시불변 시스템을 르장드르 웨이블릿 및 적분 연산 행렬을 이용하여 미분 방정식의 문제를 Sylvester 행렬 방정식 문제로 변환한 후 이를 다시 저차의 행렬 방정식들로 분할하여 이진 트리 구조를 구성한 후 말단 노드로부터 해를 구하여 전체 해를 구하는 방법을 제시하였다. 또한 말단 노드의 Sylvester 행렬 방정식을 단순 행렬 방정식들로 분리하여 해석적으로 해를 구함으로써 전체 해를 계산량을 줄이면서 쉽게 구할 수 있음을 보였다.

추후 비선형 시스템 해석에 직교함수인 르장드르 웨이블릿을 적용하여 비선형 미분 방정식을 순수 행렬 방정식으로 변환하기 위해 르장드르 웨이블릿에 관련된 적분연산 행렬, 곱셈 행렬(product matrix) 등의 성질을 연구하고자 한다.



김 범 수

1987년 고려대학교 전기공학과. 1989년 고려대학교 대학원 전기공학과(공학석사). 2002년 고려대학교 대학원 전기공학과(공학박사). 1989년~1998년 LG산전 연구소 선임연구원. 2005년~현재 경상대학교 기계시스템공학과 부교수.

REFERENCES

- [1] A. Haydar, M. H. Al-Lail, and V. Covachev, "Survey on wavelet transform and application in ODE and wavelet networks," *Advances in Dynamical Systems and Applications*, pp. 129-162, 2006.
- [2] C. F. Chen and C. H. Hsiao, "Haar wavelet method for solving lumped and distributed-parameter systems," *IEE Proc. Control Theory Appl.*, vol. 144, pp. 87-94, 1997.
- [3] H. R. Marzbana and M. Razzaghi, "Solution of time-varying delay systems by hybrid functions," *Mathematics and Computers in Simulation*, vol. 64, pp. 597-607, 2004.
- [4] B. S. Kim, I. J. Shim, M. T. Lim, and Y. J. Kim, "Combined preorder and postorder traversal algorithm for the analysis of singular systems by Haar wavelets," *Mathematical Problems in Engineering*, 2008.
- [5] B. S. Kim and I. J. Shim, "study for state analysis of linear systems using haar wavelet," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 14, no. 10, pp. 977-982, 2008.
- [6] F. Khellat and S. A. Yousefi, "The linear Legendre mother wavelets operational matrix of integration and its application," *Journal of the Franklin Institute*, vol. 343, pp. 181-190, 2006.
- [7] B. M. Mohan and S. K. Kar, "Optimal control of nonlinear systems via orthogonal functions," *IEEE Internal Conferenc on Energy, Automation, and Signal*, 2011.
- [8] M. Razzaghi and S. Yousefi, "The Legendre wavelets operational matrix of integration," *International Journal of Systems Science*, vol. 32, pp. 495-502, 2001.
- [9] F. Ding, P. X. Liu, and J. Ding, "Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle," *Applied Mathematics and Computation*, vol. 197, pp. 41-50, 2008.
- [10] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Tran. on Circuits and Systems*, vol. 25, pp. 772-781, 1978.