

신경회로망을 이용한 동적 시스템의 상태 공간 인식 모델에 관한 연구

A Study on the State Space Identification Model of the Dynamic
System using Neural Networks

*이 재 현, 강 성 인, 이 상 배
한국해양대학교 전자통신공학과 퍼지-뉴로 연구실
부산광역시 영도구 동삼동 1번지 한국해양대학교 전자통신공학과
E-mail : jhlee@kmucc.kmaritime.ac.kr

Abstract

System identification is the task of inferring a mathematical description of a dynamic system from a series of measurements on the system. There are several motives, for establishing mathematical descriptions of dynamic systems. Typical applications encompass simulation, prediction, fault diagnostics, and control system design. The paper demonstrates that neural networks can be used effective for the identification of nonlinear dynamical systems. The content of this paper concerns dynamic neural network models, where not all inputs to and outputs from the networks are measurable. Only one model type is treated, the well-known Innovation State Space model(Kalman Predictor). The identification is based only on input/output measurements, so in fact a non-linear Extended Kalman Filter problem is solved. Even for linear models this is a non-linear problem without any assurance of convergence, and in spite of this fact an attempt is made to apply the principles from linear models, and extend them to non-linear models. Computer simulation results reveal that the identification scheme suggested are practically feasible.

I. 서론

시스템 인식(System Identification)은 시스템의 연속적인 측정물로부터 동적 시스템의 수학적 표현으로 추측하는 작업이다. 동적 시스템을 수학적 표현으로 인식하여, 그 인식된 시스템이 응용되어지는 분야는 다양하다.

전형적인 응용은 시뮬레이션(simulation), 예측(prediction), 결점 진단(fault diagnostic) 그리고 제어 시스템의 설계 등을 포함한다.

특히, 제어 분야에서는 플랜트(plant)의 모델링이 매우 중요하다. 시스템 인식방식으로 신경

회로망을 사용할 경우, 플랜트의 출력과 신경회로망의 출력이 동일하도록 신경회로망의 웨이트(weight)가 조정되면 신경회로망은 플랜트를 정확히 인식하게 된다. 이러한 신경회로망에 의한 시스템 인식 방식은 기존 방식과 비교해 볼 때 신경회로망은 범용성을 갖고 있기 때문에 선형함수 뿐만 아니라 복잡한 비선형 함수를 용이하게 구현할 수 있다. 이러한 시스템 인식 방법에는 많은 인식 방법들이 존재한다. 그러나 여기서는 상태 공간 모델법을 사용한다. 왜냐하

면, 현대의 복잡한 세이시스템은 여러 개의 입·출력을 가질 수 있으며, 이들 입·출력으로서 복잡하게 관련되어 있다. 이러한 시스템을 해석하기 위해서는 수학적 표현이 간결하여야 하며, 해석에 필요한 많은 양의 계산들을 컴퓨터에 의존하는 것이 필수적이다. 이러한 관점에서 상태공간 접근방법이 시스템 해석에 가장 적합하다고 할 수 있다. 따라서, 이 논문에서는 상태 공간 방법으로 시스템을 인식할 수 있는 신경회로망에 대한 알고리즘을 제시하였다.

II. θ -벡터와 ψ -행렬의 정의

일반적으로 오류 역전파 알고리즘 (Error Backpropagation Algorithm)은 신경회로망의 웨이트 행렬 W_1 과 W_2 를 가지고 동작한다. 반면에, 예측 오차를 기초로 하는 인식 방법은 전통적으로 하나의 긴 행 벡터 θ 에서 수집되어진 모든 파라미터를 가지고 동작한다. 여기서는 어떻게 두 행렬 W_1 과 W_2 를 하나의 긴 행 벡터와 같은 구조로 되어지는가를 설명한다.

일반적으로 하나의 은닉층을 갖는 다층 퍼셉트론은 다음과 같다. 여기서 은닉층 활성화함수는 비선형 함수, 출력층 활성화함수는 선형함수를 사용한다.

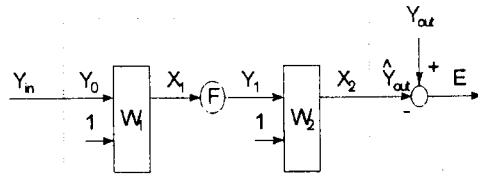


그림 1. 행렬 기호로 표시된 하나의 은닉층을 가진 MLP의 구조

$$n_0 = m_0 + 1 : \text{입력의 개수} + 1$$

$$n_1 = m_1 : \text{은닉층의 뉴런의 수}$$

$$n_2 = m_2 : \text{출력층의 뉴런의 수}$$

θ 의 요소 $\theta = (\theta_1 \theta_2 \theta_3 \dots \theta_{n_0+n_1+n_2})^T$ 은 다음과 같이 된다.

$$W_1 = \begin{bmatrix} W_1(1, \cdot) \\ W_1(2, \cdot) \\ \vdots \\ W_1(n_1, \cdot) \end{bmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_{n_0} \\ \theta_{n_0+1} & \theta_{n_0+2} & \dots & \theta_{2n_0} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{(n_1-1)n_0+1} & \theta_{(n_1-1)n_0+2} & \dots & \theta_{n_1n_0} \end{bmatrix}$$

$$W_2 = \begin{bmatrix} W_2(1, \cdot) \\ W_2(2, \cdot) \\ \vdots \\ W_2(n_2, \cdot) \end{bmatrix} = \begin{bmatrix} \theta_{n_0+n_1+1} & \theta_{n_0+n_1+2} & \dots & \theta_{n_0+n_1+n_1} \\ \theta_{n_0+n_1+n_1+1} & \theta_{n_0+n_1+n_1+2} & \dots & \theta_{n_0+n_1+2n_1} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{n_0+n_1+(n_2-1)n_1+1} & \theta_{n_0+n_1+(n_2-1)n_1+2} & \dots & \theta_{n_0+n_1+n_2n_1} \end{bmatrix}$$

θ -벡터는 W_1 과 W_2 로부터 열로 채워져 있고, θ -벡터는 다음과 같이 배치되어진다.

$$\theta = \begin{bmatrix} W_1(1, \cdot)^T \\ \vdots \\ W_1(n_1, \cdot)^T \\ W_2(1, \cdot)^T \\ \vdots \\ W_2(n_2, \cdot)^T \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{n_1} \\ \theta_1^2 \\ \vdots \\ \theta_{n_2}^2 \end{bmatrix} = \begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix}$$

ψ -행렬은 다음과 같이 배치되어진다.

$$\psi = \frac{d\hat{Y}_{out}^T}{d\theta} = \begin{bmatrix} \frac{d\hat{Y}_{out}^T}{dW_1(1, \cdot)^T} \\ \vdots \\ \frac{d\hat{Y}_{out}^T}{dW_1(n_1, \cdot)^T} \\ \frac{d\hat{Y}_{out}^T}{dW_2(1, \cdot)^T} \\ \vdots \\ \frac{d\hat{Y}_{out}^T}{dW_2(n_2, \cdot)^T} \end{bmatrix} = \begin{bmatrix} \psi_1^1 \\ \vdots \\ \psi_{n_1}^1 \\ \psi_1^2 \\ \vdots \\ \psi_{n_2}^2 \end{bmatrix} = \begin{bmatrix} \psi^1 \\ \psi^2 \end{bmatrix} \quad (1)$$

< 순환 신경회로망 >

순환 신경회로망은 입력이 신경회로망의 출력으로부터 시간 지연 되먹임이 존재하는 것을 말하고, 예를 들어 다음과 같이 계산되어진다.

$$\hat{Y}_{out}(k, \theta) = F(Y_{in}(k), \hat{Y}_{out}(k-1, \theta), \theta) \quad (2)$$

수식 (2)로부터 ψ 는 다음과 같이 된다.

$$\psi(k, \theta) = \frac{d\hat{Y}_{out}^T(k, \theta)}{d\theta} = \frac{\partial \hat{Y}_{out}^T(k, \theta)}{\partial \theta} + \frac{d\hat{Y}_{out}^T(k-1, \theta)}{d\theta} \frac{\partial \hat{Y}_{out}^T(k, \theta)}{\partial \hat{Y}_{out}(k-1, \theta)} = \frac{\partial \hat{Y}_{out}^T(k, \theta)}{\partial \theta} + \psi(k-1, \theta) \frac{\partial \hat{Y}_{out}^T(k, \theta)}{\partial \hat{Y}_{out}(k-1, \theta)}$$

여기서 $\phi(k, \theta) = \frac{\partial \hat{Y}_{out}^T(k, \theta)}{\partial \theta}$ 로 정의하면 순환 신경회로망에 대해서 다음과 같이 된다.

$$\psi(k, \theta) = \phi(k, \theta) + \psi(k-1, \theta) \frac{\partial \hat{Y}_{out}^T(k, \theta)}{\partial \hat{Y}_{out}(k-1, \theta)}$$

$\phi = \frac{\partial \hat{Y}_{out}^T(k, \theta)}{\partial \theta}$ 는 두 부분으로 나누어져 있

다. 즉, 이 첫 번째 $n_1 n_0$ 의 ϕ 요소(ϕ^1)는 W_1 과 관계가 있고, 반면에 마지막 $n_2 n_0$ 의 ϕ 요소(ϕ^2)는 W_2 와 관계가 있다.

< ϕ^1 의 계산 >

$$\begin{aligned} \phi^1 &= \frac{\partial \hat{Y}_{out}^T}{\partial \theta^1} \\ &= \frac{\partial (W_2 F(X_1(\theta)))^T}{\partial \theta^1} \\ &= \frac{\partial X_1^T(\theta)}{\partial \theta^1} \frac{\partial Y_{out}^T}{\partial X_1} W_2^T \\ &= \begin{bmatrix} \begin{matrix} \{Y_0\} & 0 & \dots & 0 \\ 1 & & & \end{matrix} \\ 0 & \begin{matrix} \{Y_0\} \\ 1 \end{matrix} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & \begin{matrix} \{Y_0\} \\ 1 \end{matrix} \end{bmatrix} F'(X_1) W_2^T \\ &= M_1 F'(X_1) W_2^T \end{aligned}$$

여기서 M_1 는 $(n_1 n_0) \times n_1$ 행렬이다.

< ϕ^2 의 계산 >

$$\phi^2 = \frac{\partial \hat{Y}_{out}^T}{\partial \theta^2} = \begin{bmatrix} Y_1 & 0 & \dots & 0 \\ 0 & Y_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Y_1 \end{bmatrix} = M_2$$

여기서 M_2 는 $(n_2 n_1) \times n_2$ 행렬이다.

< ϕ 의 계산 >

$$\phi = \begin{bmatrix} \phi^1 \\ \phi^2 \end{bmatrix} = \begin{bmatrix} M_1 F'(X_1) W_2^T \\ M_2 \end{bmatrix} \quad (3)$$

정적 신경회로망은 $\phi = \phi^1$ 이고, 순환 신경회로망은 ϕ 가 순환적으로 계산되어진다.

III. 가우스-뉴턴 탐색법

가우스-뉴턴 탐색법은 좋은 수렴 특징을 갖는 매우 유용한 최소화 방법이다. 이러한 회귀적 알고리즘에서 최소화된 오차 함수는 다음과 같다.

$$\begin{aligned} V(k, \theta) &= \sum_{i=0}^k \lambda^{k-i} \frac{1}{2} E^T(i, \theta) E(i, \theta) \\ E(i, \theta) &= Y_{out}(i) - \hat{Y}_{out}(i, \theta) \end{aligned} \quad (4)$$

여기서 θ 는 신경회로망의 모든 웨이트와 바이어스를 포함하는 열 벡터이다. 망각 요소(forgetting factor) λ 는 오래된 값을 점차적으로 잊어버리는 용도로 소개된다. 그러므로 파라미터 변화를 매우 늦게 따라가는 인식을 실행한다. 학습 쌍에 몇 번의 반복(iteration)동안 회귀

적 오프라인 학습을 실행한 것에 의해서 망각 요소는 매우 1에 근접하는 $\lambda = 0.995 \dots 0.999$ 값으로 선택된다.

보편적으로 쓰이는 회귀적인 가우스-뉴턴 최소화 방법은 다음과 같이 θ 벡터의 회귀적 갱신을 사용한다.

$$\theta(k) = \theta(k-1) - H^{-1}(k, \theta(k-1)) G(k, \theta(k-1)) \quad (5)$$

여기서 G (경사)와 H (Hessian 행렬)는 각각 파라미터 벡터 θ 에 관한 오차함수 V 의 1차와 2차 도함수이다.

G 는 다음과 같다.

$$\begin{aligned} G(k, \theta) &= \frac{dV(k, \theta)}{d\theta} \\ \phi(k, \theta) &= \frac{d\hat{Y}_{out}^T(k, \theta)}{d\theta} \end{aligned}$$

H 행렬은 다음과 같다.

$$H(k, \theta) = \frac{dG(k, \theta)}{d\theta^T}$$

일반적으로 근사화된 Hessian 행렬은 다음과 같이 R 로 표현된다.

$$R(k, \theta(k-1)) = \lambda R(k-1, \theta(k-1)) + \phi(k, \theta(k-1)) \phi^T(k, \theta(k-1))$$

가우스-뉴턴 탐색 방법을 사용한 회귀 예측 오차법은 아래에 요약되어진다. 그리고 간단하게 하기 위하여 모든 독립변수 ' θ '는 무시한다.

- 1) $\hat{Y}_{out}(k)$ 예측
- 2) $\phi(k)$ 예측
- 3) $E(k) = Y_{out}(k) - \hat{Y}_{out}(k)$ (6)
- 4) $R(k) = \lambda R(k-1) + \phi(k) \phi^T(k)$
- 5) $\theta(k) = \theta(k-1) + R^{-1}(k) \phi(k) E(k)$

IV. 비선형 동적 시스템의 상태 공간 인식 모델 구조

m 입력, n 상태, 그리고 p 출력을 갖는 선형 처리에 가장 일반적인 모델은 다음과 같이 주어진다.

$$\begin{aligned} \hat{x}(k) &= A \hat{x}(k-1) + Bu(k-1) + Ke(k-1) \\ y(k) &= C \hat{x}(k) + e(k) \end{aligned} \quad (7)$$

여기서 \hat{x} 는 n 차 상태벡터이고 u 는 m 차 입력 벡터, y 는 p 차 출력 벡터, 그리고 e 는 p 차 예측 오차 벡터이다. A, B, K, C 는 각각 $n \times n, n \times m, n \times p, p \times n$ 차 상수 행렬이다. \hat{x} 는 모두 다 측정되지는 않는다고 가정하자.

간단하게 하기 위해 행렬 C 는 $C = [I_{p,p} O_{p,n-p}]$

로 선택되어지고, 여기서 $I_{p,p}$ 는 $p \times p$ 단위행렬이고 $O_{p,n-p}$ 는 $p \times (n-p)$ 영행렬이다. 이것은 상태벡터 \hat{x} 의 첫 번째 p 요소가 \hat{y} 로 채워진다는 것을 의미한다. $k=1 \dots N$ 인 입력 추정값 $u(k)$ 와 바라는 출력 추정값 $y(k)$ 의 쌍이 주어졌을 때 이 모델의 학습은 확장 칼만 필터를 풀이하는 것과 같다.

이러한 선형 상태 공간 모델로부터의 착상을 가지고 비선형 상태 공간 모델은 자연스럽게 다음과 같이 정의된다.

$$\begin{aligned} \hat{X}(k) &= F(\hat{X}(k-1), U(k-1), E(k-1), \theta) \\ Y(k) &= C \hat{X}(k) + E(k) \end{aligned} \quad (8)$$

여기서 F 는 비선형 벡터 함수이고, θ 는 파라미터를 나타내고, $E(k)$ 는 예측 오차이다. 신경회로망 모델에서 입력벡터 $Y_{in}(k)$ 는 다음과 같고, 그림 2에 신경회로망의 구조를 나타내었다

$$Y_{in}(k) = \begin{bmatrix} \hat{X}(k-1) \\ U(k-1) \\ E(k-1) \end{bmatrix} \quad (9)$$

신경회로망의 입력은 모두 다 측정되지 않는다. 왜냐하면 이 모델은 신경회로망 주위에 되먹임 루프를 포함하기 때문이다. 그리고 신경회로망으로부터의 출력은 모두 다 측정되지 않는다. 왜냐하면 단지 출력벡터 $Y(k)$ 만이 측정되기 때문이다.

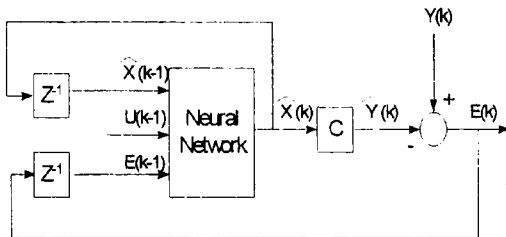


그림 2. 비선형 이노베이션 상태 공간 모델

1) 이득 행렬 $Q(k)$

$$\begin{aligned} Q(k) &= \frac{d\hat{X}(k)}{dY_{in}^T(k)} \\ &= \frac{d\hat{X}(k)}{d\{ \hat{X}^T(k-1)U^T(k-1)E^T(k-1) \}} \\ &= \left\{ \begin{array}{ccc} \frac{\partial \hat{X}(k)}{\partial \hat{X}^T(k-1)} & \frac{\partial \hat{X}(k)}{\partial U^T(k-1)} & \frac{\partial \hat{X}(k)}{\partial E^T(k-1)} \end{array} \right\} \\ &= \{ \hat{A}(k) \hat{B}(k) \hat{R}(k) \} \end{aligned} \quad (10)$$

2) $\psi(k)$ 의 계산

$$\begin{aligned} \psi(k) &= \frac{d\hat{Y}^T(k)}{d\theta} \psi_x(k)C^T \\ \psi_x(k) &= \frac{d\hat{X}^T(k)}{d\theta} \\ &= \frac{\partial \hat{X}^T(k)}{\partial \theta} + \frac{d\hat{X}^T(k-1)}{d\theta} \frac{\partial \hat{X}^T(k)}{\partial \hat{X}^T(k-1)} \\ &\quad + \frac{dE^T(k-1)}{d\theta} \frac{\partial \hat{X}^T(k)}{\partial E(k-1)} \\ &= \frac{\partial \hat{X}^T(k)}{\partial \theta} + \frac{d\hat{X}^T(k-1)}{d\theta} \frac{\partial \hat{X}^T(k)}{\partial \hat{X}^T(k-1)} \\ &\quad + \frac{d\hat{X}^T(k-1)}{d\theta} C^T \frac{\partial \hat{X}^T(k)}{\partial E(k-1)} \\ &= \psi(k) + \psi_x(k-1) \{ \hat{A}^T(k) - C^T \hat{R}^T(k) \} \end{aligned} \quad (11)$$

V. 컴퓨터 시뮬레이션

그림 3에서 시소 시스템은 두 개의 긴 막대가 지지대 위에 경첩되어있다. 축은 시소의 각 편차를 측정하기 위해 사용되어지는 3 : 1 기어율에 의하여 전위차계(potentiometer)와 결합되어져 있다. 이 실험의 목적은 시소 시스템의 시소가 수평으로 유지되기 위해 cart의 위치를 제어하는 것이다.

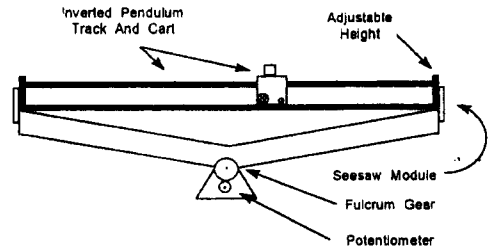


그림 3. 시소 시스템의 구성

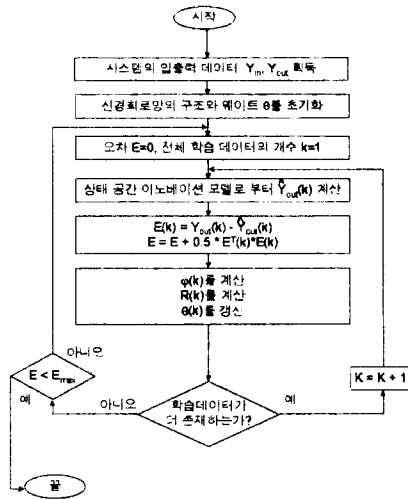
시스템의 상태는 아래와 같이 정의된다.

- x : track의 중심으로부터 cart의 위치 (m)
- \dot{x} : cart의 속도 (m/sec)
- θ : 수직으로부터 시소의 각 (rad)
- $\dot{\theta}$: 시소의 각 속도 (rad/sec)

전압 입력으로 시스템 파라미터를 표현하면 다음과 같이 된다.

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1.45 & 9.18 & -17.2 & 0 \\ 10.44 & 4.38 & 2.5 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 3.85 \\ -0.56 \end{bmatrix} u \quad (12)$$

여기서 시소 시스템의 상태 공간 인식 모델 절차는 아래와 같이 수행되어진다.



시스템의 입·출력 데이터 Y_{in} , Y_{out} 은 수식 (12)에 의해서 획득되어진다. 그때 입출력 데이터는 다음과 같다.

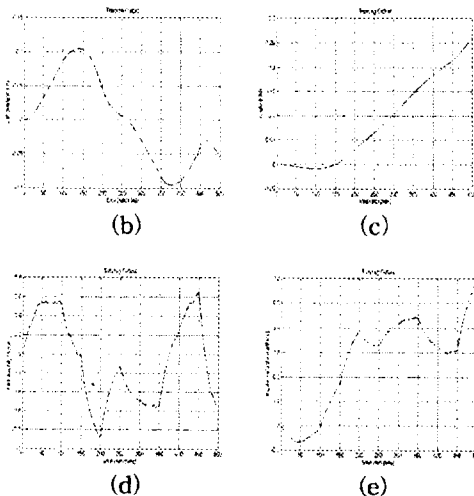
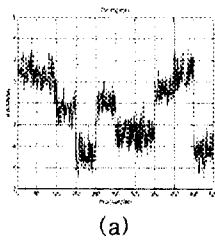


그림 4. 시소 시스템의 입·출력 데이터 : (a) 입력 전압(u), (b) cart의 위치(m), (c) 시소의 각(rad), (d) cart의 속도(m/sec), (e) 시소의 각속도(rad/sec)

그림 4의 입·출력 데이터를 가지고 신경회로망을 학습한다. 시소 시스템을 인식하기 위한 신경회로망의 구조는 그림 5와 같이 선택되어졌고, 신경회로망의 웨이트는 -0.05에서 0.05사이의 임의의 값으로 초기화하였다.

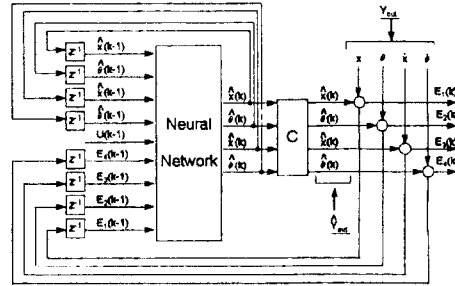


그림 5. 추정용 위한 신경회로망의 구성

그림 5에서 신경회로망 입력은 시소 시스템 4개의 상태값(x , θ , \dot{x} , $\dot{\theta}$)과 전압 입력 u , 각각의 예측 오차 4개 (E_1 , E_2 , E_3 , E_4), 즉 총 9개의 입력을 가지고 있으며, 은닉층의 뉴런은 6개를 사용하였으며, 출력층은 모든 상태값들을 추정하기 위해 4개의 뉴런을 사용하였다. 그리고, C 행렬은 모든 상태가 측정 가능하다는 전제하에서 다음과 같은 행렬이 된다.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

신경회로망으로부터의 입력은 모두 다 측정되지는 않는다. 즉, 초기에는 신경회로망으로 되먹임 되는 상태값(x , θ , \dot{x} , $\dot{\theta}$)과 예측 오차 4개 (E_1 , E_2 , E_3 , E_4)는 0으로 초기화하고, 학습을 진행하면서 자동적으로 이전에 값들이 되먹임 되어지며, 상태공간 이노베이션 모델로부터 $\hat{Y}_{out}(k)$ 를 계산하고, 그리고 $\phi(k)$, $R(k)$,

$\theta(k)$ 는 수식 (6)과 (11)에 의하여 계산한다. 전체 학습 데이터가 다 주어지고 그때 전체 오차 E가 우리가 원하는 E_{max} 보다 적으면 학습을 종료하게 되고 그렇지 않은 경우에는 학습을 계속한다. 이 시뮬레이션에서는 E_{max} 를 0.0001로 정의하여 사용하였다. 학습을 종료한 후 시소 시스템을 상태 공간 모델 형태로 정확하게 인식되었는지 확인하기 위해 그림 6에서 (a)의 테스트 입력 데이터를 입력한다.

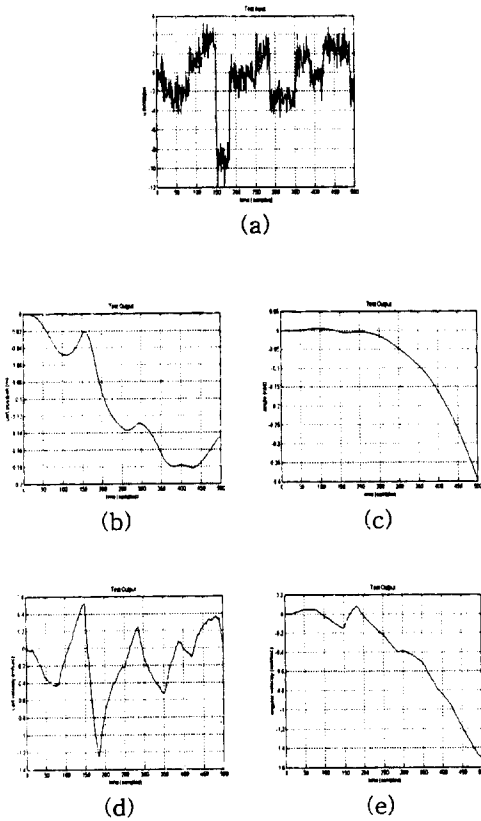


그림 6. 시소 시스템의 입·출력 데이터 : (a) 입력 전압(u), (b) cart의 위치(m), (c) 시소의 각(rad), (d) cart의 속도(m/sec), (e) 시소의 각속도(rad/sec)

그림 6의 입·출력 데이터는 수식 (12)에 의해서 획득했고, 신경회로망이 정확하게 인식되었는지 확인하기 위해 사용된다. 정확하게 인식되었다면 그림 6에서 (a)의 입력 전압을 신경회로망의 입력으로 사용할 경우 학습하지 않은 데이터에 대해서도 적절하게 반응해야 한다. 그때, 신경회로망에서 추정된 출력은 그림 7과 같다.

이 시뮬레이션 결과에서도 알수 있듯이 그림 6과 그림 7을 비교해 보면 약간의 오차를 가지고 있지만 시소 시스템을 정확하게 인식하였다는 것을 보여준다.

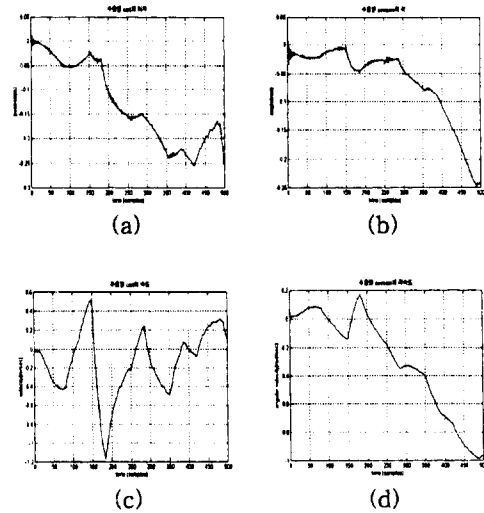


그림 7. 신경회로망에 의해 추정된 데이터 : (a) cart의 위치(m), (b) 시소의 각(rad), (c) cart의 속도(m/sec), (d) 시소의 각속도(rad/sec)

VI. 결론

본 논문에서는 신경회로망을 이용하여 동적 시스템을 상태 공간 모델로 정확하게 인식함에 따라 동적 시스템의 내부 파라미터(A, B, K)를 추정할 수 있는 비선형 이노베이션 상태 공간 인식 모델의 구조와 알고리즘을 제시하였다.

제안된 신경회로망을 이용하여 시소 시스템을 인식하기 위해 시소 시스템의 입·출력 데이터를 가지고 학습하였으며, 이 신경회로망이 시소 시스템을 정확하게 인식하였는지 확인하기 위해 테스트 데이터를 가지고 시소 시스템을 추정한 결과 만족할 만한 출력결과를 얻을 수 있었다.

VII. 참조논문

- [1] Jer-Nan Juang, Applied System Identification, Prentice-Hall, 1993.
- [2] Landau, System Identification and Control Design, Prentice-Hall, 1993.
- [3] S. Chen et al., "Non-linear system identification using neural networks", Int. J. Control, vol. 6, pp. 1191-1214, 1990.
- [4] Lennart Ljung, System identification, theory for the user, Prentice-Hall, first edition, 1987.