

이족로봇용 신경망 지능 제어기

김 성 주, 김용택*, 고재양*, 서재용**, 전홍태
 중앙대학교, *한국전력기술, **한국기술교육대학
 전화 : 031-289-4695 / 핸드폰 : 017-224-4695

The Intelligent Controller for Biped Robot Using Neural Network

Sung Joo Kim, Yong Taek Kin*, Jae Yang Ko*, Hyun Chan Cho**, Hong Tae Jeon
 CAU, *KOPEC, **KUT
 E-mail : kimyt@kopec.co.kr

Abstract

This paper proposes the controller for biped robot using intelligent control algorithm. The main purpose of this paper is to design the robot controller using Hierarchical Mixture of Experts(HME). The neural network direct control method will be applied to the control scheme for the biped robot and neural network will learn the dynamics of biped robot.

The learning scheme using a intelligent controller to biped robot is developed. The learning scheme uses a HME controller combined with a inverse biped robot model. The controller provides the control signals at each control time instant. Simulation results are reported for a seven-link biped robot.

I. 서론

인간의 존엄성과 가치에 대한 관심이 높아짐에 따라 원자력 발전소나 유해가스가 존재하는 장소에서 인간이 수행할 작업을 대신할 수 있는 이족로봇에 대한 요구가 증가하게 되었다. 이족로봇은 바퀴 구동식 이족로봇과 다리 구동식 보행로봇 등이 있다. 바퀴 구동식 이족로봇은 제작 및 제어가 쉽고 에너지 효율은 높으나 작업장의 환경에 많은 영향을 받으며, 특히 바닥면의 굴곡과 요철이 존재하는 계단의 경우에는 이족로봇의 고유한 특성을 발휘할 수 없게 된다. 인간중심으로 만들어진 작업환경에서 바퀴 구동식 이족로봇의 적용은 제한받게 된다.

본 논문의 목적은 동력학의 역 모델링 기법을 이용한 이족로봇의 제어기를 설계하는 것이다. 이를 위해

계층적 모듈라 신경망(Hierarchical Mixture of Experts, HME)을 적용한다. 이족로봇을 4개 관절그룹으로 모듈화 한다. 모듈화된 제어대상을 제어하기위해 통합한 제어기형태로 모듈화와 계층적 구조를 가진 HME를 적용하는 것이 바람직하다.

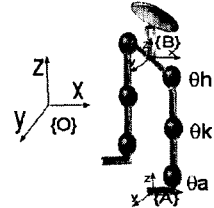


그림 1 이족보행로봇

II. 이족로봇 동력학

2.1 일족보행로봇의 기구학 및 동력학

그림 1과 같이 발끝 축의 프레임(A)와 본체 골반의 프레임(B)와의 기구학적 연결은 아래와 같다. 관절공간상의 변수 $\theta_h, \theta_a, \theta_k$ 및 θ_h , 는 가상 발끝관절(Toe), 발목(Ankle) 관절, 무릎(Knee) 관절과 골반(Hip) 관절의 각도 변수 값들이며, L_1 과 L_2 는 각 관절의 길이를 나타낸다. L_h 는 뒷굽의 높이를 나타낸다. 프레임 {O}는 기준좌표계이며 평면바다 가정 하에 보행로봇의 좌표계 {A}와의 관계는 ${}^O R_A = I$ 로 정한다. {A}에서 {B}로의 기구학을 정리하면

$${}^A X = \begin{pmatrix} x \\ z \\ \theta \end{pmatrix} = \begin{bmatrix} -L_{11}c_{\theta_1} - L_h s_{\theta_1} - L_1 s_{\theta_1 + \theta_a} - L_2 s_{\theta_1 + \theta_a + \theta_k} \\ -L_{11}s_{\theta_1} - L_h c_{\theta_1} - L_1 c_{\theta_1 + \theta_a} - L_2 c_{\theta_1 + \theta_a + \theta_k} \\ -\theta_h - \theta_k - \theta_a - \theta_t \end{bmatrix} \quad (1)$$

여기서 $c_{\theta} \equiv \cos(\theta)$, $s_{\theta} \equiv \sin(\theta)$ 이다. 이를 미분하여 자코비안(jacobian) 행렬을 만들면 아래와 같다.

$${}^A J = \begin{bmatrix} J_{1,1} & J_{1,2} & J_{1,3} & 0 \\ J_{2,1} & J_{2,2} & J_{2,3} & 0 \\ -1 & -1 & -1 & -1 \end{bmatrix}, \quad (2)$$

자코비안 행렬은 프레임 (A)와 프레임 (B)사이의 관절 속도와 직교좌표계 속도와의 관계를 가지고 있다. ${}^B A X = {}^B A \theta$ 이며 관절 토크의 힘의 관계식은 아래와 같다.

$$\tau = ({}^B A J)^T ({}^B A F) \quad (3)$$

τ 는 관절토크벡터이며, F 는 힘 벡터이다. 발끝 관절에는 모터가 존재치 않기 때문에 $\tau_i = 0$ 이다. 이러한 제약 조건을 가지고 식 (1)에 적용하면 아래와 같다.

$$\begin{bmatrix} 0 \\ \tau_o \\ \tau_k \\ \tau_k \end{bmatrix} = \begin{bmatrix} J_{1,1} & J_{2,1} & -1 \\ J_{1,2} & J_{2,2} & -1 \\ J_{1,3} & J_{2,3} & -1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_o \end{bmatrix} \quad (4)$$

f_x 와 f_o 를 열거하여 f_x 로 식을 전개하면 다음과 같은 수식을 얻을 수 있다.

$$f_x = \begin{bmatrix} -J_{2,1} \\ J_{1,1} \end{bmatrix} \begin{bmatrix} f_x \\ f_o \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} \tau_o \\ \tau_k \\ \tau_k \end{bmatrix} = \begin{bmatrix} -J_{1,2}J_{2,1} + j_{2,2} & \frac{j_{1,2}}{J_{1,1}} - 1 \\ -J_{1,3}J_{2,1} + j_{2,2} & \frac{j_{1,2}}{J_{1,1}} - 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} f_x \\ f_o \end{bmatrix} \quad (6)$$

여기서 모든 관절각도는 계측 가능하다고 가정하며, 발끝에 액츄에이터가 없으므로 “가상의” 라는 단어를 사용하였으며, 발끝의 각도는 아래의 식으로 계산된다.

$$\theta_i = -\theta - \theta_k - \theta_k - \theta_o \quad (7)$$

2.2 이족로봇 완성

전절에서는 일족 보행을 구현하여 보았으며, 이족 보행에 관한 내용을 구현하면 아래와 같다. 기존의 J. pratt 의 논문[2]에서 제한행렬(constraint matrix)를 구축하고 임의의 힘벡터 F_x, F_z 를 가지고 이족간의 힘분배 문제를 해결하였다.

$$\begin{bmatrix} f_x \\ f_o \end{bmatrix}_{left} = \alpha \begin{bmatrix} F_x \\ F_o \end{bmatrix}, \quad \begin{bmatrix} f_x \\ f_o \end{bmatrix}_{right} = (1-\alpha) \begin{bmatrix} F_x \\ F_o \end{bmatrix} \quad (8)$$

여기서 α 는 $0 < \alpha < 1$ 의 범위의 임의의 상수이다. 일족의 경우에서처럼 x방향의 힘을 줄 수는 없지만 z와 θ 의 방향으로 힘벡터를 조정할 수 있으며 이족간의 힘분배를 결정하고 식 (8)을 이용하여 각 관절의 토크를 계산할 수 있다.

III. 신경망을 이용한 로봇 제어기

3.1 계층적 모듈라 신경망

N 집단에 속해 있는 입력 벡터 $x^{(i)}$ 를 관찰할 때 원하는 출력벡터 $y^{(i)}$ 를 구축하면 다음의 식을 얻는다.

$$y = \sum_{i=1}^N g_i \sum_{j=1}^M g_{ji} \mu_{ji} \quad (9)$$

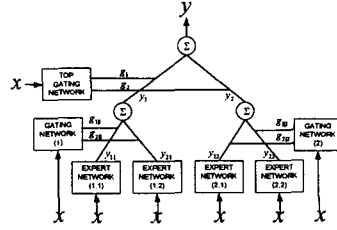


그림 2. 계층적 모듈라 신경망(HME)

여기서, μ_{ji} 는 하위 단계의 i번째 모듈의 j번째 Expert Network(EN)의 출력이며, g_{ji} 와 g_i 는 Gating Network(GN)의 출력이다. M은 하위단계 수준 모듈에서 어느 클러스터 안의 EN의 개수이고, N은 상위 수준 모듈의 노드의 개수이며 클러스터의 개수이다. EN은 일반적인 다층신경회로망을 사용한다. EN의 입력 x에 대하여 EN의 출력은 다음과 같다.

$$\mu_{ji} = f(U_{ji}, x) \quad (10)$$

여기서, U_{ji} 는 EN의 가중치행렬이며, $f(\cdot)$ 는 활성화 함수이다. 반대로 GN의 출력은 아래와 같이 얻을 수 있다.

$$\xi_{ij} = v_{ij} x \quad (11)$$

$$g_{ij} = \frac{\exp(\xi_{ij})}{\sum \exp(\xi_{ij})} \quad (12)$$

여기서, v_{ij} 는 GN의 가중치 벡터를 나타내며, 학습이 끝난 후 g_i 와 g_{ji} 는 후진부 확률 $P(i|x)$ 와 $P(j|i, x)$ 를 제공한다. 식 (9)에 식 (10)을 적용하면 아래 식이 된다.

$$y = \left(\sum_{i=1}^N g_i \sum_{j=1}^M g_{ji} U_{ji} \right) x \quad (13)$$

만약 g_i 와 g_{ji} 가 상수라면, $\sum_{i=1}^N g_i \sum_{j=1}^M g_{ji} U_{ji}$ 는 HME의 자코비안 행렬로 볼 수 있다. HME가 시스템의 전방향 모델 역할을 한다. 역 모델링 학습이 가능하고 학습이 되었다면 우리는 원하는 출력신호(y^d)를 얻을 수 있다. 그러므로 식 (13)의 방정식을 만족하는 $\sum_{i=1}^N g_i \sum_{j=1}^M g_{ji} U_{ji}$ 의 값을 얻을 수 있다면, 신경망의 정확한 오차 값을 얻을 수 있을 것이다.

$$y^d = \left(\sum_{i=1}^N g_i^d \sum_{j=1}^M g_{ji}^d U_{ji} \right) x^d \quad (14)$$

그림 2에서 상위 GN의 i 번째 신경망 활성화함수의 출력 g_i 는 다음과 같이 정의된다.

$$g_i = \frac{\exp(u_i)}{\sum_{j=1}^K \exp(u_j)}, \quad i = 1, 2, \dots, K, \quad (15)$$

여기서 u_i 는 입력벡터 x에 대한 가중치 합이다. 하

위 GN의 i 번째 클러스터의 신경망 출력함수는 Soft-Max 함수로 다음과 같다.

$$g_{1i} = \frac{\exp(u_{1i})}{\sum_{j=1}^L \exp(u_{ji})}, \begin{cases} i=1,2,\dots,L \\ j=1,2,\dots,K \end{cases}, \quad (16)$$

u_{ji} 는 i 번째 클러스터의 특정 뉴런에 적용된 입력의 가중치 합이다. 그림 2의 각 클러스터의 EN들은 상기에서 언급한 모듈라 신경망의 EN을 사용한다.

$$y_i = \sum_{j=1}^L g_{ji} y_{ij}, \quad i=1,2,\dots,K \quad (17)$$

HME의 출력벡터는 아래의 식과 같다.

$$y = \sum_{i=1}^K g_i y_i \quad (18)$$

3.2 확률해석

학습데이터 $\{x, d\}$ 는 Jordan & Jacobian의 논문[3]에서 제안한 nested regressive process에 의해 만든다. 기준응답 d 는 다음의 회귀응답으로부터 발생된다.

$$d = F_{ji}(x) + \epsilon, \quad \begin{cases} i=1,2,\dots,K \\ j=1,2,\dots,L \end{cases} \quad (19)$$

여기서 F_{ji} 는 임의의 표본함수의 과거의 값을 이용하여 미래의 값을 정확하게 예측할 수 있는 결정론적인 함수이다. ϵ 는 영점 평균인 가우시안 분포 확률벡터이다. 학습데이터 $\{x, d\}$ 에 대한 확률분포를 모델링하는 것이 목적이다. 이 목적함수가 직접적인 학습문제로 근사되기 위해 동반 가우시안 혼합모델의 확장된 형태로서 정의된 로그가능도함수(log likelihood function)를 이용하여 아래의 식과 같다.

$$l = \ln \sum_{i=1}^K g_i \sum_{j=1}^L g_{ji} \exp\left(-\frac{1}{2} \|d - y_{ji}\|^2\right) \quad (20)$$

여기서 d 는 입력벡터 x 에 대한 기준응답벡터이며 나머지 항들은 상기에서 언급하였으므로 설명을 피한다. 회귀의 경우는 확률모델은 가우시안분포라는 가정을 한다. 각 EN을 위한 $\sigma^2 I$ 의 형태의 식별 공분산 행렬은 다음의 계층적 확률모델을 만든다.

$$P(y|x, \theta) = \frac{1}{(2\pi)^{j/2} \sigma^j} \sum_{i=1}^K g_i \sum_{j=1}^L g_{ji} e^{-\frac{1}{2\sigma^2} (y - y_{ji})^T (y - y_{ji})} \quad (21)$$

g_i, g_{ji} , 및 y_{ji} 는 미지의 값들이고 미지의 조건부 확률이며 모든 확률변량은 입력벡터 x 에 대하여 조건부이다.

첫 번째는 i 번째 클러스터에서 기준응답벡터 d 를 만드는 조건부 사후확률을 정의하며 아래의 식과 같다.

$$h_i = \frac{g_i \sum_{j=1}^L g_{ji} \exp\left(-\frac{1}{2} \|d - y_{ji}\|^2\right)}{\sum_{i=1}^K g_i \sum_{j=1}^L g_{ji} \exp\left(-\frac{1}{2} \|d - y_{ji}\|^2\right)}, \quad i=1,\dots,K, \quad (22)$$

두 번째는 i 번째 클러스터의 j 번째 EN에서 기준응답벡터 d 를 만드는 조건부 사후확률을 정의하며 아래

의 식과 같다.

$$h_{ji} = \frac{g_{ji} \exp\left(-\frac{1}{2} \|d - y_{ji}\|^2\right)}{\sum_{j=1}^L g_{ji} \exp\left(-\frac{1}{2} \|d - y_{ji}\|^2\right)}, \begin{cases} i=1,\dots,K \\ j=1,\dots,L \end{cases}, \quad (23)$$

세 번째로 h_i 와 h_{ji} 사이의 곱을 집합 사후확률(Joint a Posterior Probability)로 정의하며 아래의 식과 같다.

$$h_{ji} = h_i \times h_{ji} = \frac{g_i g_{ji} \exp\left(-\frac{1}{2} \|d - y_{ji}\|^2\right)}{\sum_{i=1}^K g_i \sum_{j=1}^L g_{ji} \exp\left(-\frac{1}{2} \|d - y_{ji}\|^2\right)}, \begin{cases} i=1,2,\dots,K \\ j=1,2,\dots,L \end{cases} \quad (24)$$

집합 사후확률 h_{ji} 는 아래의 조건을 만족하여야 한다.

$$\sum_{j=1}^L \sum_{i=1}^K h_{ji} = 1, \quad 0 \leq h_{ji} \leq 1 \text{ for all } (i, j) \quad (25)$$

사후확률 h_i, h_{ji} 는 입력벡터 x 와 기준응답벡터 d 에 대하여 조건부 확률이다. 출력벡터 y_{ji} 와 g_i, g_{ji} 는 EN과 GN을 구성하고 있는 신경망의 시냅스 가중치 행렬에 의존한다. 로그가능도함수 l 은 미지의 파라미터로써 신경망 시냅스 가중치로 표현 가능한 로그가능도함수로 볼 수 있다. 이런 이유로 로그가능도함수를 최대화 한다는 것은 파라미터 최대가용도추종을 이끌어낸다. 시스템에서 모든 시냅스 가중치에 즉각적으로 작용하는 기울기상승기법(Gradient Ascent Method)을 이용하여 l 함수의 최대화는 수행할 수 있다. Jordan & Jacob의 방식[4]을 이용하여 시냅스의 조정을 계산한다.

학습알고리즘 개발은 다음과 같은 $\frac{\partial l}{\partial a_i}, \frac{\partial l}{\partial c_{ji}}$ 및

$\frac{\partial l}{\partial w_{ji}^{(m)}}$ 등의 요소들을 결정하여야 한다. 여기서 a_i 는 상

위 GN의 i 번째 출력 신경망의 가중치 벡터이고, c_{ji} 는 i 번째 클러스터 GN의 j 번째 출력 뉴런의 시냅스 가중치 벡터로 정의한다. $w_{ji}^{(m)}$ 은 i 번째 클러스터의 j 번째 EN의 q 개의 출력 뉴런 중 m 번째 출력 뉴런 가중치 벡터로 정의한다.

$$\frac{\partial l}{\partial a_i} = \frac{\partial l}{\partial u_i} \frac{\partial u_i}{\partial a_i}, \quad i=1,2,\dots,K \quad (26)$$

위의 식을 유도하면 아래의 식으로 정리된다.

$$\Delta a_i = \eta \sum_{j=1}^L (h_{ji}^{(t)} - g_i^{(t)}) x^{(t)} \quad (27)$$

로그가능도함수를 i 번째 클러스터 GN의 j 번째 출력 뉴런의 시냅스 가중치 벡터로 편미분하면

$$\frac{\partial l}{\partial c_{ji}} = \frac{\partial l}{\partial u_{ji}} \frac{\partial u_{ji}}{\partial c_{ji}}, \quad \begin{cases} i=1,2,\dots,K \\ j=1,2,\dots,L \end{cases} \quad (28)$$

위의 식을 유도하면 아래의 식으로 정리된다.

$$\Delta c_{ji} = \eta \sum_{q=1}^Q h_{ji}^{(t)} (h_{ji}^{(t)} - g_{ji}^{(t)}) x^{(t)} \quad (29)$$

로그가용도함수를 i 번째 클러스터의 j 번째 EN의 q 개의 출력 뉴런 중 m 번째 출력 뉴런 가중치 벡터로 편미분하면,

$$\frac{\partial l}{\partial w_{ji}^{(m)}} = \frac{\partial l}{\partial y_j^{(i)}} \frac{\partial y_j^{(i)}}{\partial w_{ji}^{(m)}}, \begin{cases} i = 1, 2, \dots, K \\ j = 1, 2, \dots, L \\ m = 1, 2, \dots, q \end{cases} \quad (30)$$

$$\Delta w_{ji}^{(m)} = \eta \sum h_i^{(l)} h_j^{(l)} (y_j^{(l)} - \mu^{(l)}) x_i^{(l)T} \quad (31)$$

여기서 η 는 학습율이고, 상기 식을 통하여 학습에 필요한 요소들을 결정할 수 있다.

IV. 이족로봇의 제어기

4.1 HME 제어기

HME는 시스템의 오차신호를 자코비안에 의해 역 모델링 된 신호로부터 사후확률을 III장의 절차에 따라 구한다. 사후확률과 사전확률과의 차이가 감소되도록 EM알고리즘을 사용하여 시스템의 오차가 줄어드는 방향으로 학습한다.

$$y = N(Y_d, W) \quad (32)$$

이고, 이족로봇의 시스템 함수는 아래와 같다.

$$y_s = S(y) \quad (33)$$

HME의 출력을 입력으로 받고 HME는 위의 식의 관계에 대한 역 모델링을 수행한다. 역 모델링을 수행하기 위해 오차 $e(= y_s - y_d)$ 를 가지고 오차를 시스템의 자코비안 관계로 표현하면 아래와 같이 나타낼 수 있으며 실행지수 E 를 아래와 같이 정의한다.

$$E = \|y' - y^d\|^2 \quad (34)$$

위의 실행지수 E 를 기울기 감소 가중치 갱신 규칙을 사용하면 아래와 같은 식으로 유도된다.

$$\Delta w = -\eta \left(\frac{\partial N}{\partial w} \right) \Delta y \quad (35)$$

이고 Δy 는 아래와 같이 쓸 수 있다.

$$\Delta y = \frac{d(S^T)}{dt} (y_s - y_d) = J(y_s)^T (y^d - y^s) \quad (36)$$

V. 모의실험

5.1 보행실험 결과

이족로봇의 보행실험 결과는 아래의 그림 3, 4와 같다. 그림 3은 이족로봇의 기준 관절계절을 나타내고, 그림 4는 HME 제어기로 구동된 이족로봇의 오차 곡선을 나타낸다.

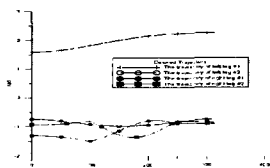


그림 3 이족로봇의 관절 기준계절

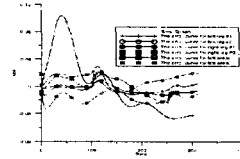


그림 4 이족로봇 관절 오차곡선

VI. 결론

본 논문에서는 이족로봇의 신경망 제어기 설계를 제안하였다. 기존의 이족로봇제어방식은 다중제어방식으로 다수의 제어기로 구성되어 있다. 보행을 위헤선 전문가 지식에 의존하며 고전적인 제어기를 채택했을 때는 제어기 마다 동특성에 맞는 파라미터를 각각 설정하여야 하는 문제가 있다. 이런 문제를 HME를 이용하여 해결하였으며 이족로봇 시뮬레이터를 이용하여 검증하였다. 또한, HME 신경망의 빠른 수렴속도를 이용하여 보행 안정성을 보장하였다.

참고문헌

- [1] M Vukobratovic and A. A. Frank, "Legged locomotion studies", Advances in External Control of Human Extremities, M. H. Gavrilovic and A. B. Wilson Jr., eds. Belgrade: Yugoslav Committee for Electronics and Automation, pp. 407-430, 1969.
- [2] Pratt, J., Torres, A., Dilworth, P. & Pratt, G., "Virtual Actuator Control", IEEE International Conference on Intelligent Robots and Systems, 1996. Simon Haykin, Neural Networks - A Comprehensive Foundation, Macmillian College Publishing Company Inc., 1994.
- [3] Michael I. Jordan., and R.A. Jacobs, "Hierarchies of Adaptive experts" In Advanced in Neural Information Processing System 4 (J.E. Moody, S.J. Hanson and R.P. Lippmann, eds) San Mateo, CA: Morgan Kaufmann, pp.985-992. 1992.
- [4] Michael I. Jordan and Robert A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," Neural Computation, Vol. 6, No. 1, pp. 181-214, 1994.