
이륜 역진자 로봇의 각도 및 속도 제어를 위한 신경회로망 PID 제어기

김영두* · 안태희* · 정건우* · 최영규**

Neural Network PID Controller for Angle and Speed Control of Two Wheeled Inverted Pendulum Robot

Young-doo Kim* · Tae-hee An* · Gun-oo Jung* · Young-kiu Choi**

요 약

본 논문에서는 최근 편리하고 간편한 이동수단으로 각광받고 있는 Segway 형태의 이륜 역진자 로봇에 대해 기존의 방법보다 더 안정적인 밸런싱과 빠른 속도제어가 가능하도록 제어기를 설계하였다. 먼저 널리 사용되는 PID 제어 구조를 이륜 역진자 로봇에 적용하고, 몇 단계로 지정된 탑승자의 각 몸무게에 대해 적절한 PID 제어기 이득을 시행착오적으로 선택하여 밸런싱과 속도제어가 잘 이루어지도록 하였다. 앞에서 지정된 몸무게 이외의 임의의 몸무게에 대한 PID 제어기 이득값을 구하기 위해 보간 개념으로 신경회로망을 사용하였으며 앞에서 시행착오적으로 구한 제어 이득값을 학습데이터로 사용하였다. 이와 같이 신경회로망을 이용하여 설계된 제어기의 성능을 확인하기 위해서 시뮬레이션 연구를 수행하였으며, 기존의 PID 제어기보다 빨리 밸런싱과 속도제어가 됨을 확인할 수 있었다.

ABSTRACT

In this paper, a controller for two wheeled inverted pendulum robot, i.e., Segway type robot that is a convenient and easily handled vehicle is designed to have more stable balancing and faster velocity control compared to the conventional method. First, a widely used PID control structure is applied to the two wheeled inverted pendulum robot and proper PID control gains for some specified weights of users are obtained to get accurate balancing and velocity control by use of experimental trial-and-error method. Next, neural network is employed to generate appropriate PID control gains for arbitrarily selected weight. Here the PID gains based on the trial-and-error method are used as training data. Simulation study has been carried out to find that the performance of the designed controller using the neural network is more excellent than the conventional PID controller in terms of faster balancing and velocity control.

키워드

이륜 역진자형 로봇, 신경회로망, 밸런싱 제어, PID 제어이득

Key words

two wheeled inverted pendulum type mobile robot, neural network, balancing control, PID control gain

* 준회원 : 부산대학교 대학원 전자전기공학과

접수일자 : 2011. 05. 19

** 정회원 : 부산대학교 전자전기공학부 (교신저자, ykichoi@pusan.ac.kr)

심사완료일자 : 2011. 07. 27

I. 서론

인류는 고대부터 지금까지 새로운 이동수단에 대해 끊임없이 연구해왔으며 또한 발전시켜왔다. 바퀴의 발명으로부터 시작된 이동수단의 역사는 내연기관의 발전에 따라 비약적으로 발전했으며 현재는 많은 종류의 이동수단들이 산업분야 뿐만 아니라 일상생활에서 그 수요와 목적에 따라 깊숙이 자리 잡았다. 특히 최근 사회에서는 친환경적이고 안전하며 고효율의 이동수단을 요구하고 있다.

이러한 추세에 맞춰서 각광 받고 있는 이동수단이 Segway이다. Segway는 이동로봇의 한 종류로 2륜형이 일반적이긴 하나 그림 1과 같은 물건의 운반을 목적으로 하는 로봇식 이동형 플랫폼(RMP)과 같이 4륜형으로 구성된 로봇도 있다[1]. Segway는 그 활용범위가 다양한데 기본적으로 개인 주행기기로 활용될 뿐만 아니라 그림 2에서 보는 것과 같이 창고에서 활용할 수 있는 운반용으로도 이용되며, 상업 경비, 골프, 레저, 긴급의료 서비스와 같이 다양한 분야에서 이미 많이 활용되고 있다. 그에 따라서 Segway에 관련된 연구가 활발히 진행되고 있다.

특히 Segway에 탑승자가 탑승했을 경우 여러 가지 알고리즘을 적용하여 밸런싱 시키는 연구들이 많이 이루어지고 있으며 널리 알려진 제어 알고리즘인 fuzzy 알고리즘을 이용하거나 신경회로망 알고리즘을 이용한 제어들을 볼 수 있다[2, 3]. 또한 Segway의 속도제어와 관련된 연구도 활발히 진행되고 있다[4].

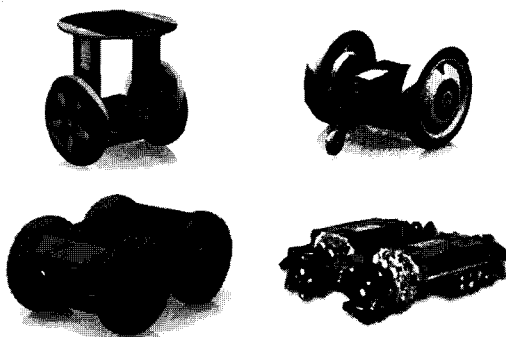


그림 1. 다양한 종류의 로봇식 이동형 플랫폼(RMP)
Fig. 1 Various kinds of robot movable platform



그림 2. 세그웨이의 다양한 활용
Fig. 2 Various applications of Segway

본 논문에서는 Segway를 제어하기 위하여 다양한 제어 방식중 하나인 PID 제어 방식을 선택하였다. 탑승자가 바뀌면 그 몸무게 값이 달라짐에 따라 제어 성능에 변화가 일어나는 등 영향을 미치게 되는데, 이를 보완하기 위해 신경회로망을 이용한 PID 제어를 제안하고 몸무게 변화에 따라 적절한 이득값을 구해 제어함으로써 기존 제어기보다 우수한 성능을 갖게 한다. 성능 확인을 위하여 시뮬레이션 상에서 이륜 역진자형 로봇을 구성하고 일반 PID 제어를 적용한 경우와 제시된 신경회로망 PID 제어를 적용한 경우를 비교 확인한다.

II. 이륜 역진자형 로봇의 모델링

이륜 역진자형 로봇은 그림 3과 같이 3-자유도(3-DOF)를 가지는 형태의 로봇으로 동역학적으로 모델링해서 해석하기 위하여 그림 4와 같은 형태로 나타내고 동역학적인 변수와 파라미터 값들을 결정하여 제어기 설계를 위한 상태 방정식을 유도한다[5]. 상태방정식 유도를 위해 필요한 여러 가지 파라미터 값들은 표 1을 통해서 정의하였다.

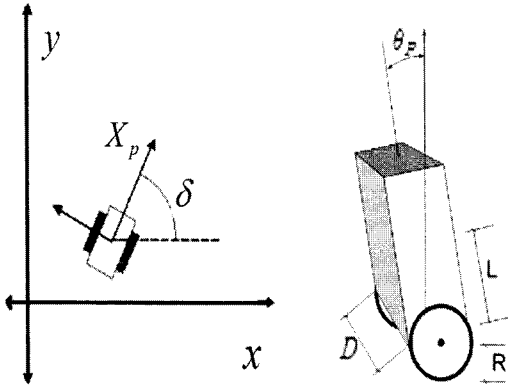


그림 3. 이륜 역진자 로봇
Fig. 3. Two wheeled inverted pendulum robot

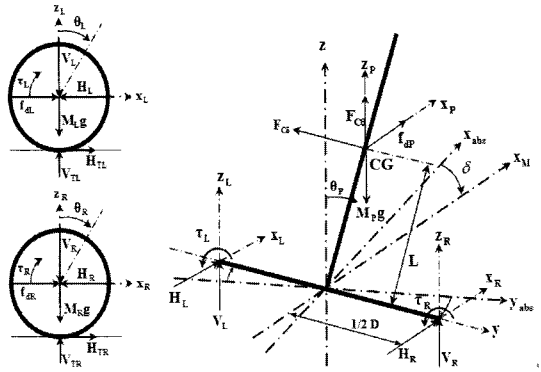


그림 4. 이륜 역진자형 로봇의 모델링
Fig. 4 Modeling of two wheeled inverted pendulum robot

표 1. 파라미터값들의 정의
Table. 1 Definitions of parameters

J_L, J_R	Y축에 대한 좌·우 바퀴의 관성 모멘트
M_L, M_R	좌·우 바퀴의 질량
$J_{P\theta}$	Y축에 대한 물체의 관성 모멘트
$J_{P\delta}$	Z축에 대한 물체의 관성 모멘트
M_p	물체의 질량
R	바퀴의 반지름
D	바퀴의 축간 거리
L	물체의 무게 중심과 Y축간의 거리
H_L, H_R	좌·우 바퀴중심으로 수평 방향 반작용
V_L, V_R	좌·우 바퀴중심으로 수직 방향 반작용
H_{TL}, H_{TR}	좌·우 바퀴 접촉면의 수평방향 반작용
V_{TL}, V_{TR}	좌·우 바퀴 접촉면의 수직방향 반작용

주어진 파라미터 값들을 바탕으로 왼쪽 바퀴에 해당하는 수평 운동 방정식을 유도해서 보면 식(1)과 같이 나타낼 수 있다.

$$M_L \ddot{x}_L = f_{dL} - H_L + H_{TL} \quad (1)$$

또한 수직 방향에 대해서도 운동 방정식을 유도해서 나타내보면 식(2)와 같이 나타낼 수 있다.

$$M_L \ddot{z}_L = V_{TL} - M_L g - V_L \quad (2)$$

마지막으로 회전 운동 방정식을 유도해보면 식(3)과 같이 나타낼 수 있다.

$$J_L \ddot{\theta}_L = \tau_L - H_{TL} R \quad (3)$$

식(1)~(3)으로 표현된 운동 방정식은 오른쪽 바퀴에도 동일한 형태로 적용 된다. 또한 물체에 관한 수평, 수직, 회전 운동방정식을 유도하여 나타내면 아래의 식(4)~(7)과 같이 표현할 수 있다.

$$M_P \ddot{x}_P = f_{dP} + H_R + H_L \quad (4)$$

$$M_P \ddot{z}_P = V_R + V_L - M_P g + F_{C\theta} \quad (5)$$

$$J_{P\theta} \ddot{\theta}_P = (V_L + V_R)L \sin(\theta_P) - (H_L + H_R)L \cos(\theta_P) - (\tau_L + \tau_R) \quad (6)$$

$$J_{R\delta} \ddot{\delta} = \frac{D}{2}(H_L - H_R) \quad (7)$$

제어기 설계에 있어 단순화 시키고 오차를 줄이기 위해서 원심력에 의한 미끌림 현상을 배제 시켰으며 각각의 바퀴가 동일하게 지면에 밀착되어 있다는 조건을 두었고 모터의 시정수는 로봇의 시정수에 비해 매우 작으므로 모터의 동역학적 요소 또한 배제 시킬 수 있다.

또한 θ_P 의 경우 매우 작은 값을 가지므로 0으로 근사화 시킬 수 있는데 그에 따라 $J_{P\theta}$ 는 $\theta_P=0$ 에 대응하는 관성 모멘트로 근사화 시킬 수 있다. 위의 조건들을 바탕으로 $\theta_P=0, \delta_P=0$ 일 때 선형 상태방정식을 구하면 식 (8)과 같이 나타낼 수 있다.

$$\begin{aligned}
 \begin{bmatrix} \dot{x}_M \\ \dot{v}_M \\ \dot{\theta}_P \\ \dot{\omega}_P \\ \dot{\delta} \\ \ddot{\delta} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & A_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_M \\ v_M \\ \theta_P \\ \omega_P \\ \delta \\ \dot{\delta} \end{bmatrix} \\
 + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ B_{21} & B_{22} & B_{23} & B_{24} & B_{25} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ B_{41} & B_{42} & B_{43} & B_{44} & B_{45} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ B_{61} & B_{62} & B_{63} & B_{64} & B_{65} & 0 \end{bmatrix} \begin{bmatrix} \tau_L \\ \tau_R \\ f_{dL} \\ f_{dR} \\ f \end{bmatrix}
 \end{aligned} \tag{8}$$

상태공간을 표현한 식(8)을 이용하여 선형화된 제어기 설계가 가능하다.

III. 제어기 설계

이륜역진자형 로봇의 동역학 모델링을 통하여 구한 상태방정식을 기초로 하여 제어기를 설계하여 구성하면 그림 5와 같은 블록선도로 나타낼 수 있다. 설계된 제어기에서 목표값들은 로봇의 목표 위치값 x_{Ml} , 목표 속도값 v_{Ml} , 목표 밸런싱 각도값 θ_{Ml} , 목표 각속도의 값 ω_{Ml} 로 정의 되었고, 실제값들은 로봇의 위치값 x_M , 로봇의 속도값 v_M , 밸런싱 각도값 θ_M , 밸런싱 각속도값 ω_M 으로 나타낼 수 있다.

이러한 실제 데이터 값들은 로봇의 센서를 통해서 구

할 수 있는데 로봇의 위치와 속도값은 모터의 엔코더에서 알 수 있으며, 밸런싱 각도와 각속도는 자이로와 각도 센서를 통해서 구할 수 있다. 설계된 제어기의 입력값은 목표값과 실제 측정값의 차인 오차값으로 적절한 PID 이득값을 적용시킴으로써 양쪽 바퀴의 출력 토크를 결정한다.

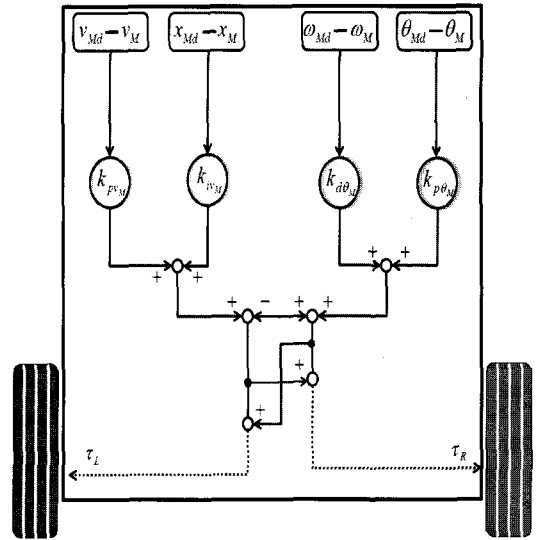


그림 5. 제어기 블록선도
Fig. 5. Controller block diagram

제어기에서 제어하고자 하는 대상에 따라서 입력된 에러값에 각각의 PID 이득값을 곱해주고 결과값에 따라 모터제어에 필요한 최종적인 토크값을 출력한다. 이 때 사용되는 PID 이득값들은 입력 무게값에 따라서 다르게 적용되는데 입력 무게값에 따라서 시행착오적으로 구한 이득값들을 적용한다.

그러나 시행착오적으로 구한 이득값에 정확히 일치하는 무게값이 아닌 임의의 무게값이 입력으로 들어 오게 되면 새로운 입력에 대한 적절한 이득값 선택에 따른 문제가 발생한다. 만약, 기존의 방법을 적용하면 임의의 무게값이 입력되었을 경우 시행착오적 방법으로 획득한 이득값 중 새로운 정보의 입력 무게와 가장 근사한 무게에 해당하는 이득값을 선택하여 사용한다. 그러나 근사한 이득값을 적용하였을 경우 빠르고 정확한 밸런싱 각도 및 속도 제어결과를 기대하기는 어렵다.

본 논문에서는 앞에서 시행착오적으로 구한 이득값들로 신경회로망을 학습시키고 임의의 무게값이 들어왔을 경우 적절한 제어 이득값을 신경회로망으로 출력시켜 기존의 방법보다 좀 더 좋은 결과를 보일 수 있었다. 제어시스템의 알고리즘을 블록선도로 나타내면 그림 6과 같다.

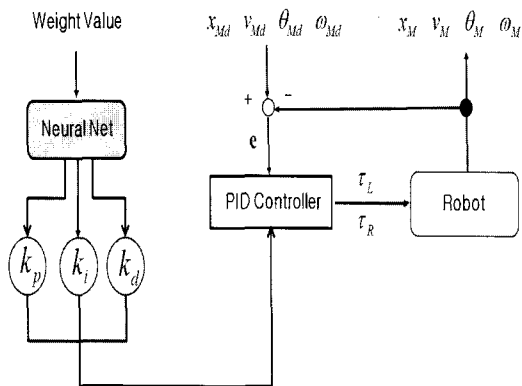


그림 6. 제어시스템 블록선도
Fig. 6. Control system block diagram

IV. 신경회로망 학습

생물 개체내의 신경신호 전달 및 신호처리 과정을 알고리즘으로 모사한 방법을 신경회로망 알고리즘이라고 한다. 또한 신경회로망 알고리즘의 학습알고리즘을 통해서 보다 나은 출력을 구할 수 있다. 본 논문에서는 지도 학습법(supervised learning)을 이용한 알고리즘을 사용하였다. 이 방법은 실제 출력값과 원하는 값의 오차를 구하여 오차 정보를 신경회로망에 전달하는 과정을 반복함으로써 실제 출력값의 오차를 최대한 줄이는 방법이다. 그 밖에도 학습의 방법에 따라 강화학습법 (reinforcement learning)을 이용한 알고리즘과 비지도 학습법(unsupervised learning)을 이용한 알고리즘들이 있다.

본 논문에서 제시하는 신경회로망 알고리즘은, 기존의 PID 제어기에서 실험을 통해 우수한 성능을 갖는 PID 제어기 이득 값 K_p , K_i , K_d 값들을 구하며 그 값들을 신경회로망의 목표치로 사용하고 학습을 통해 보간(interpolation) 함으로써 연속적인 PID 이득 값

들을 획득하는 구조이다. 신경회로망의 구조는 그림 7과 같다. 신경회로망 구조를 살펴보면 입력층, 은닉층, 출력층으로 이루어진 다층(multilayer)구조로 이루어져 있다.

특히, 입력층과 출력층 사이의 모든 중간층을 은닉층이라고 하는데 여러 개의 은닉층이 사용될 수 있다. 또한, 여러 개의 노드가 각각의 층별로 배열되며, 하나의 층으로부터 노드들의 출력은 층간의 가중치를 통해서 확대 또는 감소되어 다른 층의 노드로 전달된다. 이와 같이 처리 노드는 층을 구성하고 다음 층에 일반적으로 완전히 상호 연결되지만 같은 층 내에서는 상호연결이 없다[6].

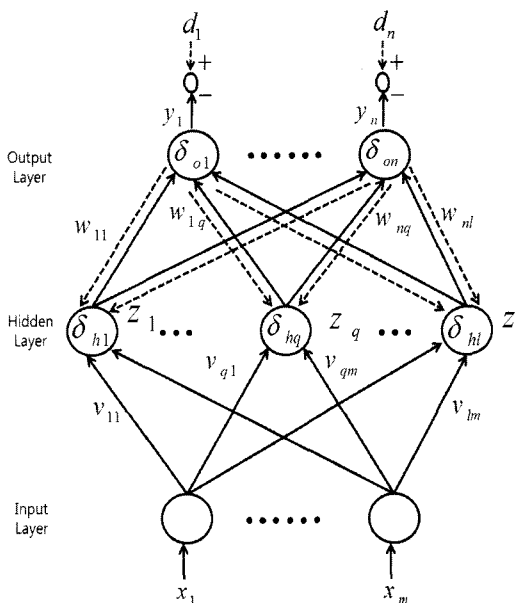


그림 7. 신경 회로망의 구조
Fig. 7. Structure of neural network

그림 7에서 임의의 입력을 x_m 이라 하고 은닉층의 임의의 출력을 z_q , 출력층의 임의의 출력을 y_n 라 정의할 때, 출력 값들을 식(9), (10)과 같이 표현 할 수 있다.

$$z_q = a(\text{net}_q) \tag{9}$$

$$\text{net}_q = \sum_{j=1}^m v_{qj} x_j$$

$$y_n = a(\text{net}_n) \quad (10)$$

$$\text{net}_i = \sum_{q=1}^l w_{nq} z_q$$

본 논문에서 사용하는 역전파 알고리즘은 다층(multilayer)으로 구성된 feedforward 신경망에서 사용되는 학습 알고리즘이며, 학습의 방법은 지도 학습법이다. 간단히 학습의 개념을 살펴보면 입력의 결과값인 출력에서 원하는 값만큼의 오차가 발생할 때 오차에 비례하여 출력층의 가중치를 갱신하고, 그 다음 은닉층의 가중치를 갱신한다.

이렇게 갱신된 가중치를 적용했을 때 모든 학습 데이터의 오차가 적정 수준으로 감소할 때까지 이 과정을 반복한다.

그림 7과 같은 구조의 신경회로망을 역전파 알고리즘으로 학습시키면 계산된 입력에 따른 계산된 출력과 원하는 출력간의 오차를 계산하고, 오차를 줄이기 위해 가중치의 증감 여부를 결정한다. 그리고 각각의 가중치를 얼마나 변화시킬 것인가를 결정하고 결정된 값으로 가중치를 갱신한다. 이 과정들을 학습 데이터에 대해 오차가 적정 수준으로 감소하기까지 반복하게 된다.

이러한 학습과정을 통하여 갱신된 가중치를 구하는 과정을 수식으로 나타내어 보면 식(11)~(14)로 나타낼 수 있다[7].

$$\delta_{oi} = (d_i - y_i)(y_i(1 - y_i)) \quad (11)$$

$$\delta_{hq} = \left(\sum_{i=1}^n \delta_{oi} w_{iq} \right) (z_q(1 - z_q)) \quad (12)$$

$$\Delta v_{qi} = \eta(\delta_{hq})x_i, v_{qi} = v_{qi} + \Delta v_{qi} \quad (13)$$

$$\Delta w_{jq} = \eta(\delta_{oj})x_q, w_{jq} = w_{jq} + \Delta w_{jq} \quad (14)$$

V. 시뮬레이션

제안된 알고리즘에 의해 구해진 이득값들을 제어기에 적용하였을 때, 기존의 방식보다 더 나은 성능을 보임을 확인하기 위하여 시뮬레이션을 시행하였다. 먼저, Matlab 프로그램을 사용한 실험을 통해 구한 PID 이득값과 신경회로망으로 튜닝한 PID 이득값을 그래프를 통해 비교해서 나타낼 것이다. 그리고 각각의 이득값을 이용하여 설계한 제어기를 Simulink 프로그램으로 모델링한 로봇에 적용시켜 밸런싱 각도와 로봇의 위치정보에 따른 속도를 비교할 것이다.

5.1 PID 제어기 이득값

몸무게에 따른 PID 이득값들의 범위를 표2에 나타내었다. PID 이득의 목표값들은 5kg 간격으로 실험을 통해 시행착오적으로 구한 가장 좋은 값들로 구성된다. PID 이득의 목표값과 신경회로망을 통해 학습된 이득값을 그래프로 비교한다.

신경회로망의 입력 x_m 은 실수값을 갖는 단일 입력이 되고 사용자의 몸무게를 나타내며 로드셀(load cell) 등을 이용하여 입력시킬 수 있다. 신경회로망 출력값 y_n 에서 n은 3이 되고 출력값은 PID 이득 Kp, Kd, Ki 값이며 그의 목표값은 5kg 간격으로 실험을 통해 가장 좋은 값을 선택하였다. 뉴런의 개수는 13개로 구성하였다. 신경회로망 입력인 몸무게는 탑승자의 몸무게를 일반적으로 고려하여 40kg부터 5kg의 단위로 90kg까지 설정하였으며, 임의로 두 개의 몸무게 값으로 시뮬레이션 하였다.

그림 8에서 각도 이득 Kp, Kd를 시행착오적으로 구한 목표 이득값과 학습된 이득값을 나타내었다. 실험적으로 구한 목표 이득값들은 5kg 단위로 지정된 반면 신경회로망에서 학습된 이득값들은 몸무게에 대해서 연속적으로 나타난다. 즉, 신경회로망의 interpolation 기능을 사용한다. 또한, 그림 9에서 위치 이득 Kp, Ki를 시행착오적으로 구한 목표 이득값과 학습된 이득값을 나타내었다. 각각의 Kp, Kd, Ki의 경우 몸무게의 값에 따라 각각 다른 값을 가지며 무게가 늘어남에 따라 변화 하게 된다. 변화의 정도는 그림 9에서 확인할 수 있다.

표 2. 몸무게 범위에 따른 PID 이득값 범위
Table. 2 PID gains range for weight

	각도 이득값 범위	위치 이득값 범위
40~50 (kg)	$K_p=196\sim 200$ $K_d=12\sim 13$	$K_i=17\sim 18$ $K_p=13\sim 15$
50~60 (kg)	$K_p=200\sim 207$ $K_d=13\sim 15.2$	$K_i=18\sim 19.7$ $K_p=15\sim 18$
60~70 (kg)	$K_p=207\sim 221$ $K_d=15.2\sim 18$	$K_i=19.7\sim 23$ $K_p=18\sim 21$
70~80 (kg)	$K_p=221\sim 238$ $K_d=18\sim 21$	$K_i=23\sim 27$ $K_p=21\sim 23$
80~90 (kg)	$K_p=238\sim 262$ $K_d=21\sim 24$	$K_i=27\sim 33$ $K_p=23\sim 27$

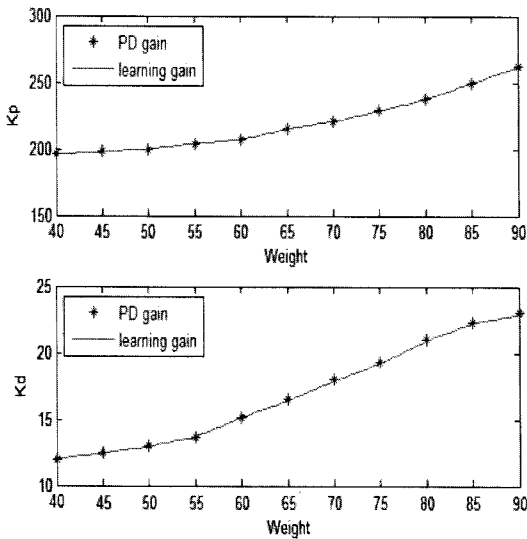


그림 8. 몸무게에 따른 각도 이득
Fig. 8 Angle gains for weights

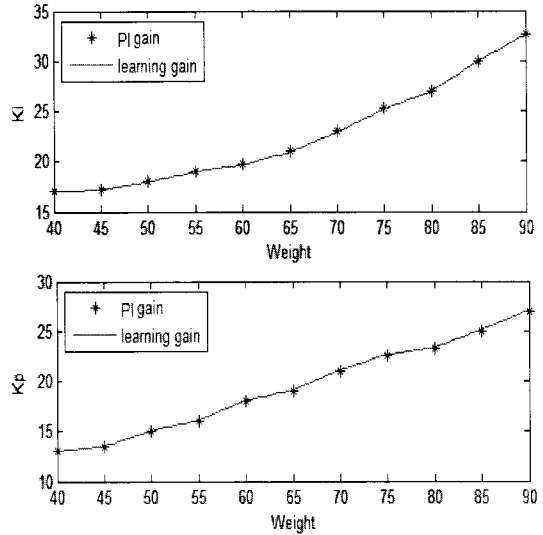


그림 9. 몸무게에 따른 위치 이득
Fig. 9 Position gains for weights

5.2 시뮬레이션 과정 및 결과

시행착오적 방법으로 구한 기존의 PID 이득값과 신경회로망 학습을 통해 구한 PID 이득값의 성능을 비교하기 위해서 Simulink 프로그램으로 설계한 이론 역진자형 로봇에 적용해 보았다. 로봇에는 각도와 속도 측정을 위한 센서들을 부착하였다. 가속도, 자이로, 각도 센서로부터 실시간으로 측정값을 제어기에 전송할 수 있도록 설계되었다. 로봇의 높이는 1.5m, 바퀴의 반지름은 0.35m, 두 바퀴의 무게는 각각 5kg으로 설계하였다.

시뮬레이션을 위해 로봇의 전체 무게는 시행착오적으로 구한 이득값에 해당하지 않는 무게값으로 67kg과 83kg으로 설정하여 실행하였다. 각 무게의 시뮬레이션 결과들은 그림 10~13에서 나타내었다. 입력 무게값이 67kg일 경우 밸런싱 각도 변화의 결과값은 그림 10과 같이 나타난다. 그래프에서 알 수 있듯이 기존의 PID 이득값을 적용했을 경우에 비해서 신경회로망 학습을 통해서 얻은 PID 이득값을 적용한 경우가 더 나은 성능을 보이는 것을 알 수 있다. 기존의 경우에 비해서 학습된 이득값을 적용했을 경우에 밸런싱 각도의 제어시간이 단축되는 것을 알 수 있다.

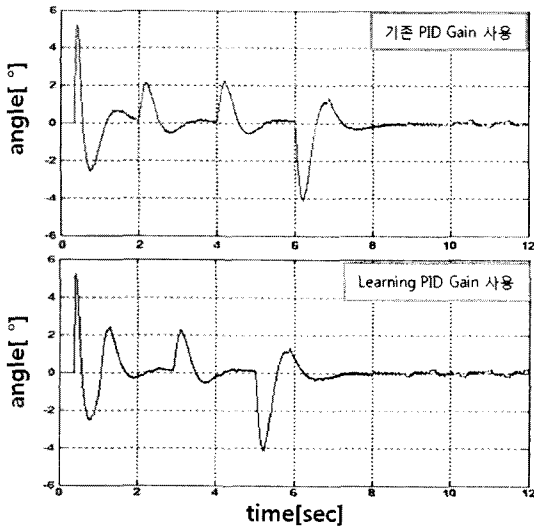


그림 10. 몸무게가 67kg일 경우의 각도 궤적
Fig. 10 Angle trajectory for weight of 67kg

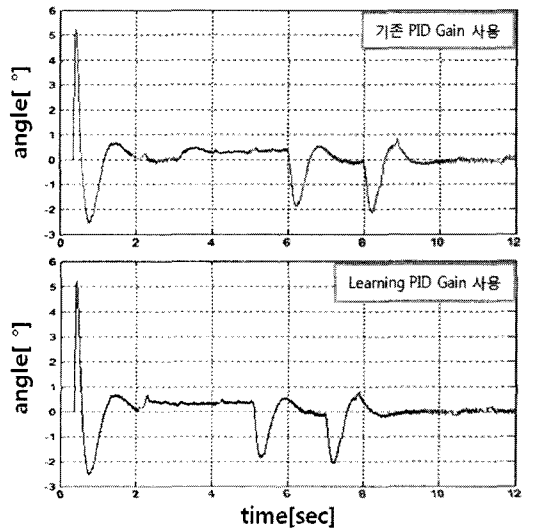


그림 12. 몸무게가 83kg일 경우의 각도 궤적
Fig. 12 Angle trajectory for weight of 83kg

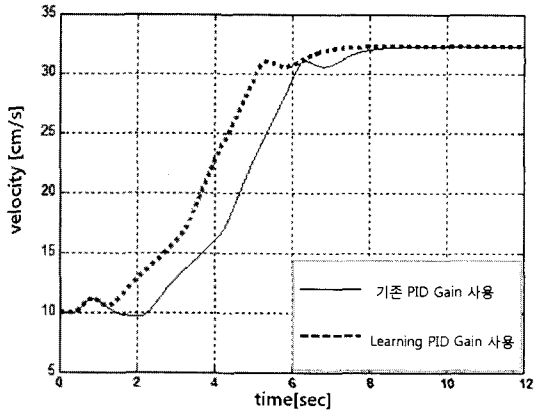


그림 11. 몸무게가 67kg일 경우의 속도 궤적
Fig. 11 Velocity trajectory for weight of 67kg

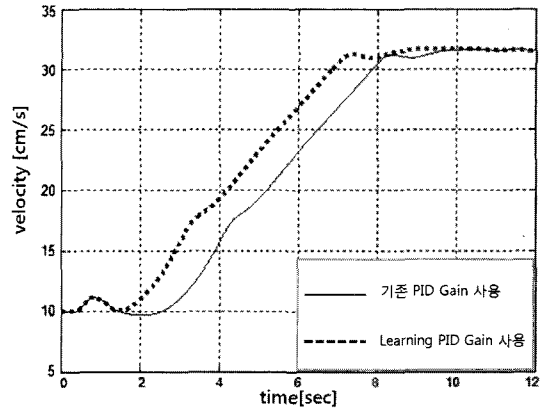


그림 13. 몸무게가 83kg일 경우의 속도 궤적
Fig. 13 Velocity trajectory for weight of 83kg

그림 11은 입력 무게값이 67kg일 때 학습된 이득값을 적용시켰을 때와 기존의 이득값을 적용했을 때의 속도 궤적을 나타낸다. 로봇이 안정적으로 등속도 주행하는 시점에 도달하는 시간이 기존 이득값을 적용했을 때보다 학습된 이득값을 적용했을 때가 빠른 것을 알 수 있다. 수치상으로 정리를 하여 보면 67kg일 때 기존의 제어기를 사용한 로봇보다 제안된 제어기를 사용한 로봇이 약 1.2초 빠르게 밸런싱 되어 등속도 운동함을 그래프 상으로 볼 수가 있다.

입력 무게값을 83kg으로 적용했을 경우 결과를 그림 12~13를 통해 나타내었다. 앞선 시물레이션 결과와 마찬가지로 기존 이득값을 제어기에 적용했을 때보다 학습된 이득값을 적용하였을 경우 밸런싱 시간이 단축되었음을 알 수 있다. 또한 로봇의 등속도 도달 시간이, 제안된 신경회로망으로 학습된 이득값을 적용시켰을 때가 기존의 이득값을 적용시켰을 때보다 67kg과 마찬가지로 약 1.2초 단축되었음을 그림 13에서 알 수 있다. 시물레이션 결과에서 알 수 있듯이 제안된 학습 이득값을

적용하였을 경우, PID제어에서 기존의 이득값을 적용했을 때보다 우수한 성능을 보임을 시뮬레이션을 통하여 확인할 수 있다.

VI. 결 론

본 논문에서 이륜 역진자 로봇 탑승자의 몸무게가 변화하여도 좋은 제어성을 갖는 제어기를 설계하기 위해, 일반적인 PID 제어기의 이득값인 K_p , K_i , K_d 의 값을 몸무게에 따라 시행착오적으로 구하고, 이것들을 학습데이터로 하여 신경회로망 학습 알고리즘 학습을 통해서 연속적인 이득값을 찾아서 PID 제어기에 적용시켰다.

신경회로망 학습을 통해 얻어진 이득값을 적용한 제어기 성능을 평가하기 위해서 시행착오적으로 구한 기존의 이득값을 적용한 제어기와 시뮬레이션으로 비교하였다. 시뮬레이션은 사용자 몸무게에 따라서 67kg, 83kg의 2가지 경우를 시행하였고, 입력 무게값이 클수록 각도의 변화량과 제어시간이, 무게값이 작을 때보다 약간 커지지만 두 경우 모두 본 논문에서 제시한 제어기가 기존의 제어기보다 우수한 성능을 보였다.

참고문헌

- [1] "Segway", <http://www.segway-korea.com>
- [2] H. Azizan, M. Jafarinasab, S. Behbahani, and M. Danesh, "Fuzzy Control on LMI Approach and Fuzzy Interpretation of the Rider Input For Two Wheeled Balancing Human Transporter," Proc. of the 2010 IEEE International Conference on Control and Automation, Xiamen, China, June 9-11, 2010
- [3] Ching-Chih Tsai, Hsu-Chih, and Shui-Chun Lin, "Adaptive Neural Network Control of a Self-Balancing Two-Wheeled Scooter," IEEE Transactions on Industrial Electronics, vol. 57, no. 4, April, 2010
- [4] Kaustubh Pathak, Jaume Franch, and Sunil K. Agrawal, "Velocity Control of a Wheeled Inverted Pendulum by Partial Feedback Linearization," Proc. of 43rd IEEE Conference on Decision and Control Atlantis, Paradise Island, Bahamas December 14-17, 2004
- [5] Felix Grasser, Aldo D'Arrigo, Silvio Colombi and Alfred C. Rufer, "JOE: A Mobile, Inverted Pendulum," IEEE Transactions on Industrial Electronics, vol. 49, no. 1, April, 2002
- [6] Vojislav Kecman, *Learning and Soft Computing*, The MIT Press, Cambridge, MA, 2001
- [7] M.I. El Adawy, M. E. Aboul-wafa, H.A. Keshk and M.M. El Tayeb, "A Soft -Backpropagation Algorithm for Training Neural Network," Proc. of 19th National Radio Science Conference, Alexandria, March 19-21, 2002

저자소개



김영두(Young-Doo Kim)

2009년 부경대학교 전기공학과
공학사
2011년 부산대학교
전자전기공학과 공학석사

※ 관심분야: 로봇제어, 퍼지제어, 유전알고리즘 등



안태희(Tae-Hee An)

2009년 부경대학교 전기공학과
공학사
2011년 부산대학교
전자전기공학과 공학석사

2011년~현재 부산대학교 전자전기공학과 박사과정
※ 관심분야: Robot Control, Neural Network, Fuzzy Algorithm



정건우(Gun-Oo Jung)

2009년 부산대학교
전자전기공학과 공학사
2010년~현재 부산대학교
전자전기공학과 석사과정

※관심분야: 퍼지제어, 지능로봇, 응용제어



최영규(Young-Kiu Choi)

1980년 서울대학교 전기공학과
(공학사)
1982년 KAIST 전기 및
전자공학과(공학석사)

1987년 KAIST 전기 및 전자공학과(공학박사)
1986년~현재 부산대학교 전자전기공학부 교수.
※관심분야: 로봇제어, 지능제어, 신경회로망 등