신경회로망을 이용한 선박의 최단시간 궤적제어

최영규* · 박진현**

Minimum-Time Trajectory Control of Ships Using Neural Networks

Young-Kiu Choi* · Jin-Hyun Park**

이 논문은 2012년도 경남과학기술대학교 기성회연구비 지원에 의하여 연구되었음.

요 약

위치에 따라 변화하는 해류가 있는 해상에서 선박이 일정속도로 움직이면서 진행각도를 제어하여 특정목표점에 최단시간으로 도착하고자 한다. 이러한 문제는 Zermelo의 항해문제라고 불리고 있으며 calculus of variation에 의해 최단시간제어의 근사해를 구할 수 있지만 매우 복잡한 비선형방정식의 역관계를 테이블 형태로 이용하여야 하므로 그 정확도가 높지 않다. 본 논문에서는 이러한 근사해의 정확도를 높이기 위해 신경회로망을 이용하여 비선형방정식의 역관계식을 표현하였고, 신경회로망의 보간 기능으로 인해 높은 정확도의 최단시간제어가 가능하였다. 여러 가지 항해 상황에 대한 컴퓨터 시뮬레이션을 통해서 본 논문에서 제안한 방법이 기존의 방법보다 우수함을 확인할 수 있었다.

ABSTRACT

A ship is intended to reach a specified target point in the minimum-time when it travels with a constant speed through a region of strong currents and its heading angle is the control variable. This is called the Zermelo's navigation problem. Its approximate solution for the minimum-time control may be found using the calculus of variation. However, the accuracy of its approximate solution is not high since the solution is based on a table form of inverse relations for some complicated nonlinear equations. To enhance the accuracy, this paper employs the neural network to represent the inverse relation of the complicated nonlinear equations. The accurate minimum-time control is possible with the interpolation property of the neural network. Through the computer simulation study we have found that the proposed method is superior to the conventional ones.

키워드

Zermelo 항해 문제, 최단시간제어, 신경회로망

Key word

Zermelo navigation problem, minimum-time control, neural network

* 정회원 : 부산대학교 공과대학 전기공학과

접수일자 : 2012. 08. 21

** 종신회원 : 경남과학기술대학교 메카트로닉스공학과
(교신저자, uabut@gntech.ac.kr)

심사완료일자 : 2012. 10. 11

I. 서 론

오래 전부터 선박의 기동 및 제어는 힘든 일이었으며 많은 연구자들의 관심을 받아 왔다[1]. 이론과 기술이 발전함에 따라 제어방법은 점차 복잡해지고, 기존의 간단한 제어방법들에서 최적제어와 같은 고급 제어방법들로 바뀌고 있다[2]. 이러한 고급 제어방법들이 실제로 실시시간으로 선박제어에 적용되려면 고도의 계산능력과 정확한 상태측정이 요구된다.

최적제어는 시스템 모델이 선형이나 또는 비선형이냐에 따라 여러 다른 이론들을 포괄하고 있다[3]. 하지만 이러한 이론들은 한 가지 공통점을 갖고 있으며, 그것은 대상 시스템의 중요 파라미터들을 최적화시킨다는 점이다. 선박 기동 및 제어에서 그러한 최적화는 최소시간제어, 방향타와 같은 제어 작동기 사용의 최소화, 최종 상태에 도달하는 궤적 중 가장 부드러운 궤적 생성 등으로 귀결된다[4]. 선박의 최소 기동시간 제어는 Zermelo 항해문제에서 시작되었다[5-8]. 강한 해류가 있는 바다에서 선박이 일정속도로 움직이면서 진행 각도(heading angle)를 제어 각도로 하고 가변시켜 임의의 초기위치에서 목표위치로 최대한 빠르게 항해하는 문제를 Zermelo 항해문제라고 한다. 이러한 문제는 calculus of variation의 기법을 이용하여 그 해를 구할 수 있으며, 초기 위치에 대해서 최적제어 각도를 그래프(또는 테이블) 형태로 근사치를 구할 수 있다[3]. 이러한 그래프 형태로 주어지는 최적제어 각도 근사치는 그래프를 세분화 할수록 그 정확도가 향상되지만 계산상의 이유로 무한히 세분화할 수 없으므로 근사오차가 발생하게 된다.

본 논문에서 이러한 근사오차를 감소시키기 위해 보간(interpolation) 기능을 갖는 신경회로망을 선택하여 최적제어 각도의 정확도를 높였고 그로 인해 선박이 최단시간으로 주어진 목표점에 도달하도록 한다. 본 논문에서 제안한 신경회로망을 이용한 최적제어기의 성능을 확인하기 위해서, 여러 가지 경우의 초기 선박 위치에 대해서 제어 및 항해 상황을 컴퓨터 시뮬레이션 하였다. 그 결과 본 논문에서 제안한 방법이 기존의 방법보다 우수함을 확인할 수 있었다.

II. Zermelo 항해 문제

그림 1과 같이 배가 거센 해류가 있는 지역을 항해하고 있다고 하자. 배의 속도는 일정 값 V 이고, 방향타를 조절하여 항해각도 θ 를 제어변수(control variable)로 삼는다고 하자. 이때의 해류 속도는 x 축 방향으로 u 이고, $u = (V/h)y$ 와 같이 배의 y 좌표에 비례하며 h 는 주어진 상수이다.

그림 1. Zermelo 문제의 형상
Fig. 1 Geometry of Zermelo's problem

배의 초기 위치 $[x(t_0), y(t_0)]$ 가 주어질 때, 최단시간으로 원점 $[0, 0]$ 에 도달하게 하는 시간최적제어(time-optimal control 또는 minimum-time control) $\theta(t)^*$ 를 구하는 것이 문제이다. 단, t_0 는 초기시간이며 통상 $t_0 = 0$ 으로 놓는다.

배의 운동방정식은 아래 식과 같다.

$$\dot{x} = V \cos(\theta) + (V/h)y \quad (1)$$

$$\dot{y} = V \sin(\theta) \quad (2)$$

최단시간 문제이므로 성능지수(performance index)는

$$J = \int_{t_0}^T 1 dt = (T - t_0) \quad (3)$$

단, T 는 최종 시간이다.

즉, 운항시간 J 를 최소화하여 가장 빠른 시간으로 초기 위치 $[x(t_0), y(t_0)]$ 에서 원점 $[0, 0]$ 에 도달하게 하

는 제어량 $\theta(t)$ 를 구한다.

식(1), (2)의 스칼라 식을 벡터 식으로 바꾸어 표현하면

$$\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} V \cos(\theta) + (V/h)y \\ V \sin(\theta) \end{bmatrix} \quad (4)$$

위의 식(4)를 일반적 형태로 표현하면 아래와 같다.

$$\frac{d}{dt} \mathbf{z} = \mathbf{f}(\mathbf{z}, \theta) \quad (5)$$

단, 상태 $\mathbf{z} = [x \ y]^T$ 로 정의한다.

최적제어를 구하기 위한 일반 조건식을 유도하면 다음과 같다.

Hamiltonian $H(\mathbf{z}, \theta)$ 는

$$\begin{aligned} H(\mathbf{z}, \theta) &= L + \Lambda^T \mathbf{f}(\mathbf{z}, \theta) \\ &= 1 + [\lambda_x(t) \ \lambda_y(t)] \begin{bmatrix} V \cos(\theta) + (V/h)y \\ V \sin(\theta) \end{bmatrix} \\ &= 1 + \lambda_x(t)(V \cos(\theta) + (V/h)y) + \lambda_y(t) V \sin(\theta) \end{aligned} \quad (6)$$

Costate equation은 다음 식(7)과 같으며, 각 성분 별 방정식은 식(8), (9)와 같다.

$$-\begin{bmatrix} \dot{\lambda}_x(t) \\ \dot{\lambda}_y(t) \end{bmatrix} = \begin{bmatrix} \frac{\partial H}{\partial x} \\ \frac{\partial H}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda_x(t) V/h \end{bmatrix} \quad (7)$$

$$-\dot{\lambda}_x(t) = 0 \quad (8)$$

$$-\dot{\lambda}_y(t) = \lambda_x(t) V/h \quad (9)$$

윗식을 시간 t 에 대해 적분하면

$$\lambda_x(t) = \lambda_x = \text{constant} \quad (10)$$

$$\lambda_y(t) = \lambda_y(T) + (\lambda_x V/h)(T-t) \quad (11)$$

Stationarity condition은 다음과 같다.

$$0 = \frac{\partial H}{\partial \theta} = -\lambda_x(t) V \sin(\theta) + \lambda_y(t) V \cos(\theta) \quad (12)$$

최종시간 T 는 고정되어 있지 않고, 최종 상태 $\mathbf{z}(T) = [x(T) \ y(T)]^T = 0$ 으로 고정되어 있으므로 $H(t) = 0$ 이다. 단, $t_0 \leq t \leq T$ 이다.

따라서 $H(t)$ 는 다음과 같이 정리된다.

$$\begin{aligned} H(t) &= 1 + \lambda_x(t)(V \cos(\theta) + (V/h)y) \\ &\quad + \lambda_y(t) V \sin(\theta) = 0 \end{aligned} \quad (13)$$

식(12)와 (13)의 연립방정식을 $\lambda_x(t)$, $\lambda_y(t)$ 에 대해서 풀기 위해 vector-matrix 의 형태로 표현하면 아래와 같다.

$$\begin{bmatrix} V \cos(\theta) + (V/h)y & V \sin(\theta) \\ -V \sin(\theta) & V \cos(\theta) \end{bmatrix} \begin{bmatrix} \lambda_x(t) \\ \lambda_y(t) \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad (14)$$

따라서

$$\lambda_x(t) = \frac{-\cos(\theta)}{V + V(y/h)\cos(\theta)} \quad (15)$$

$$\lambda_y(t) = \frac{-\sin(\theta)}{V + V(y/h)\cos(\theta)} \quad (16)$$

식(10)에서 $\lambda_x(t)$ 가 일정하므로 $\lambda_x(t) = \lambda_x(T)$ 이다. 윗 식과 식(15)에서 아래와 같이 놓으면 식(17)과 같은 관계가 성립한다.

$$\frac{-\cos(\theta(t))}{V + V(y(t)/h)\cos(\theta(t))} = \frac{-\cos(\theta(T))}{V + V(y(T)/h)\cos(\theta(T))} \quad (17)$$

최종시간 T 에서 $y(T) = 0$ 이므로

$$\cos(\theta(t)) = \frac{\cos(\theta(T))}{1 - (y(t)/h)\cos(\theta(T))} \quad (18)$$

윗 식을 $y(t)$ 에 대하여 정리하면

$$\frac{y(t)}{h} = \sec(\theta(T)) - \sec(\theta(t)) \quad (19)$$

비슷한 방법으로 $x(t)$ 를 구해보면 아래 식 (20)과 같다.

$$\frac{x(t)}{h} = \frac{1}{2} [\sec(\theta(T)) \{ \tan(\theta(T)) - \tan(\theta(t)) \} - \tan(\theta(t)) \{ \sec(\theta(T)) - \sec(\theta(t)) \} + \ln \left(\frac{\sec(\theta(T)) + \tan(\theta(T))}{\sec(\theta(t)) + \tan(\theta(t))} \right)] \quad (20)$$

위의 식(19)과 (20)에서 배의 좌표 $x(t)$, $y(t)$ 가 주어지면 시간 t 에서 time-optimal control $\theta(t)$ 를 구할 수 있지만 아주 복잡한 비선형 방정식을 풀어야 하고 그 정확도에 문제가 있으므로, 최종 시간 T 에서의 $\theta(T)$ 와 시간 t 에서의 $\theta(t)$ 를 각각 어떤 주어진 구간 내에서 세분화하고, 각 세분화된 $\theta(T)$ 와 $\theta(t)$ 값들을 식(19), (20)에 대입하여 $x(t)$, $y(t)$ 를 구하고 table 또는 graph 형태로 표현한다[3].

시간 t 에서 위치 $[x(t), y(t)]$ 가 주어지면 $\theta(t)$ 를 위의 graph 또는 table의 역관계에서 근사 값 형태로 구하여 운동방정식에 해당하는 식(1), (2)에 대입하여 미분방정식을 푼다.

그런데 위의 graph 또는 table의 정확도가 떨어지면 시간 t 에서 정확한 시간 최적 제어량 $\theta(t)$ 가 구해지기 어려우므로 최소시간으로 정확히 원점에 도달하지 못하게 된다.

따라서 위의 table을 정확하게 만드는 것이 중요하나, 입력구간을 무한히 잘게 세분화하기 어려우므로 본 논문에서 table을 사용하는 대신 보간 기능을 갖는 신경회로망을 사용하여 높은 정확도의 시간 최적 제어량 $\theta(t)$ 를 구하고자 한다.

III. 신경회로망을 이용한 최적제어

생물 개체 내의 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 학습(supervised learning) 형태의 역전파 학습알고리즘(back propagation learning algorithm)을 사용하였다[9]. 이 방법은 실제 출력값과 원하는 값의 오차를 구하여 오차 정보를 신경회로망에 전달하는 과

정을 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 신경회로망은 보간 기능을 사용하여 위의 식(19)와 (20)의 역관계를 표현한다. 즉, 선박의 위치 좌표 $x(t)$ 와 $y(t)$ 가 주어지면 이에 해당하는 최적제어 $\theta(t)$ 와 $\theta(T)$ 를 구한다. 이러한 신경회로망을 학습시키기 위해서 최종 시간 T 에서의 $\theta(T)$ 와 시간 t 에서의 $\theta(t)$ 를 각각 어떤 주어진 구간 내에서 세분화하고 각 세분화된 $\theta(T)$ 와 $\theta(t)$ 값들을 식(19), (20)에 대입하여 $x(t)$, $y(t)$ 를 구한다. 아래에서 θ 는 $\theta(t)$ 를, x 와 y 는 $x(t)$ 와 $y(t)$ 를 각각 나타낸다.

$$\text{식 (19)} : \begin{bmatrix} \theta \\ \theta(T) \end{bmatrix} \Rightarrow y$$

$$\text{식 (20)} : \begin{bmatrix} \theta \\ \theta(T) \end{bmatrix} \Rightarrow x$$

입력 출력

그림 2. 최적제어에 대한 선박위치 계산
Fig. 2 Computation of the ship position for the optimal control

적절히 세분화된 θ , $\theta(T)$ 들을 식(19), (20)에 대입하여 x , y 값들을 구한 뒤, 입출력 관계를 뒤집어 x 와 y 를 신경회로망의 입력데이터로 삼고 θ 와 $\theta(T)$ 를 출력데이터 즉, 학습데이터(training data)로 삼는다. 그림 3은 최적제어를 위한 신경회로망의 학습데이터 입출력 관계를 표시하였다.

그림 3. 최적제어를 위한 학습데이터
Fig. 3 Learning data for the optimal control

θ 와 $\theta(T)$ 를 0° 에서 360° 구간에서 각각 n 개의 격자점(grid point)으로 각각 나눈다면 총 n^2 개의 2차원 공

간 격자점들이 발생하고, 각 격자점에서 식(19), (20)에 의해 x , y 값들을 계산한다.

위에서 계산된 x , y 를 입력으로, θ 와 $\theta(T)$ 를 출력으로 순서를 뒤바꾸어 신경회로망의 학습데이터를 만들고 신경회로망을 적절하게 학습시킨 뒤, 임의의 x , y 에 대해 θ , $\theta(T)$ 를 신경회로망에 의해 구한다.

또한 신경회로망의 출력 θ , $\theta(T)$ 중 실제 제어기에 사용되어지는 값은 θ 이므로 단순히 출력 θ 만을 갖는 신경회로망 제어기도 구성할 수 있다. 이는 출력 뉴런의 개수가 적어 신경회로망의 학습이 쉬우며 제어기 설계가 간단하다는 장점을 갖는다. 그림 4는 한 개의 출력을 갖는 신경회로망 제어기를 표시하였다.

그림 4. 한 개의 출력을 갖는 신경회로망
Fig. 4 Neural network with one output

IV. 시뮬레이션 및 결과

4.1. 신경회로망의 학습

선박의 선속도 V 는 $30knots$ ($15.4 m/s$)로 하고 h 는 $1000m$, 초기 시간 t_0 는 0초, θ 의 단위는 degree로 한다. 신경회로망의 학습을 위해 θ 와 $\theta(T)$ 를 0° 에서 360° 를 5° 간격으로 72^2 개의 격자점으로 나누어 식(19), (20)에 대입하여 x , y 값들을 구한다. 이렇게 구한 x , y 값들 중 $\tan$ 함수와 $\sec$ 함수에 의한 singular point들을 제외하고 신경회로망 후보 학습쌍을 구성한다. 그림 5는 θ , $\theta(T)$ 에 대한 x/h 와 y/h 를 나타낸다.

그림 5에서 구한 데이터를 신경회로망 후보 학습쌍으로 x/h , y/h 를 입력으로 θ 와 $\theta(T)$ 를 출력으로 구성하여 학습하려고 하나, $[x/h, y/h] = [0, 0]$ 에서 신경회로망에서 함수 관계가 존재하지 않으므로 학습쌍으로 선택할 수 없다. 따라서 신경회로망의 후보 학습쌍들의 데이터를 축소하여 학습쌍들을 구성하였다.

(a)

(b)

그림 5. θ , $\theta(T)$ 에 대한 x/h 와 y/h 값

(a) x/h (b) y/h

Fig. 5 Values of x/h and y/h for θ , $\theta(T)$

(a) x/h (b) y/h

선박의 출발점을 항상 1상한이 되도록 좌표축을 잡고, x/h , y/h 의 값에도 한계값을 설정하여 학습이 가능한 데이터쌍들을 구성하였다. 학습 데이터 범위를 다음과 같이 $90^\circ < \theta < 270^\circ$, $90^\circ < \theta_T < 180^\circ$, $-8 < x/h < 17.5$, $-4 < y/h < 4$ 로 설정하였다.

그림 6은 위와 같이 학습데이터 범위를 축소하여 설정된 신경회로망 학습을 위한 데이터쌍들을 표시하였다.

(a)

(b)

그림 6. 신경회로망 학습을 위한 데이터
(a) x/h (b) y/h
Fig. 6 Data for neural networks training
(a) x/h (b) y/h

그림 7은 그림 6의 신경회로망 학습을 위한 데이터들을 입출력 관계를 뒤집어 학습데이터로 삼았다. 표 1과 같이 신경회로망의 파라미터들을 설정하여 학습 후, 그림 8에 각 학습쌍의 퍼센트 오차를 표시하였다.

표 1. 신경회로망 파라미터
Table. 1 Neural network parameters

신경회로망 파라미터	
입력	2개
출력	2개
Hidden layers	Hidden 1 : 30 neurons Hidden 2 : 40 neurons
Activation functions	Tangent sigmoid
Learning rule	Back propagation

(a)

(b)

그림 7. 신경회로망의 recall data (a) θ (b) θ_T
Fig. 7 Recall data of neural networks (a) θ (b) θ_T

그림 8. 퍼센트 오차
Fig. 8 Percent errors

한 개의 출력 θ 만을 갖는 신경회로망 제어기의 파라미터들은 앞선 신경회로망의 파라미터들과 같이 구

성하였다. 앞의 2개의 출력을 갖는 신경회로망에 비하여 신경회로망의 학습 시간이 매우 줄어들을 알 수 있었다.

4.2. 신경회로망 제어 성능 평가를 위한 제어방법

신경회로망을 이용한 시간 최적제어의 성능평가를 위해서 다음과 같이 2가지로 근사화된 최적제어방법들을 사용하였다.

θ , $\theta(T)$ 에서 일정간격 격자구조로 세분화된 값을 넣지만, 식(19), (20)이 비선형 함수이므로 수식에 의해 출력된 x , y 에 의한 θ , $\theta(T)$ 는 그림 7과 같은 데이터 구조를 갖는다. 따라서 배의 좌표(x , y)가 정확히 격자점에 있지 않을 경우,

- i) 근사 격자점 최적제어 방법 : (x , y)에 가장 가까운 격자점에 해당하는 θ 를 구하여 적용한다.
- ii) 선형근사 최적제어 방법 : (x , y)에 가장 가까운 3개의 격자점 L_1, L_2, L_3 를 찾아서 선형 근사식을 통하여 θ 를 구하여 적용한다.

선박의 위치(x , y)에 가장 가까운 3개의 격자점 L_1, L_2, L_3 의 좌표들을 그림 9에서와 같이 (x_1, y_1) , (x_2, y_2) , (x_3, y_3) 라고 하고, 해당되는 θ 를 각각 $\theta_1, \theta_2, \theta_3$ 라고 하자. 3개의 격자점을 지나는 평면방정식을 $\theta = ax + by + c$ 의 형태로 나타낼 수 있고, 파라미터 a, b, c 는 3 격자점의 좌표들을 대입하여 연립방정식을 풀어서 구할 수 있다.

그림 9. 최적제어의 선형근사

Fig. 9 Linear approximation for the optimal control

따라서 선박의 위치좌표 (x , y)가 격자점이 아닌 경우, 선형근사식 $\theta = ax + by + c$ 식에 의하여 θ 를 구할 수 있다.

4.3. 신경회로망 제어기의 성능 평가

신경회로망 제어기의 성능 평가를 위해서 4.2절에 설명한 가장 가까운 격자점에 해당하는 최적제어방법과 선형 근사방법을 사용한 최적제어방법과 비교하여 최종위치 $[x, y] = [0, 0]$ 근처에 도달하는 시간과 수렴정도를 살펴보고자 한다. 수렴정도는 제어대상인 선박이 $[x, y] = [0, 0]$ 에 가장 가까이 접근했을 때의 좌표의 거리(Euclidean distance)로 하며, 그 때의 시간을 수렴 시간으로 측정하였다.

그림 10은 배의 초기위치가 $[x(t_0), y(t_0)] = [2400, 500]$ 일 때의 3가지 제어기의 성능을 표시하였다. 수렴 시간은 선형 근사방법을 사용한 최적제어 성능이 가장 좋으나, 최종 수렴 위치가 매우 멀리 떨어져있음을 알 수 있다.

또한 신경회로망 제어기법을 사용한 제어기는 출력 θ 가 매우 부드럽게 연속적으로 나타났으나, 근사 격자점 방법은 급격한 변화를 보임을 알 수 있다. 이는 신경회로망 제어기법이 학습되지 않은 데이터들에 대하여 일반화 기능 즉 보간 기능을 가지므로 적절한 출력 데이터를 생성함을 나타낸다.

그림 11은 초기위치를 $[3800, 2150]$ 로 두었을 때의 결과를 나타내며, 그림 12는 초기위치를 $[5200, 2500]$ 로 두었을 때의 결과를 표시하였다. 그림 11과 12에서 신경회로망 제어기는 수렴 시간과 수렴 정도에서 다른 제어방법에 비하여 매우 뛰어남을 알 수 있다. 또한 보간 기능에 의해 매우 부드러운 신경회로망 출력을 나타내었다.

근사 격자점 제어방법의 수렴시간은 신경회로망의 방법과 비슷하나, 수렴정도는 그림 11의 경우 1.76배, 그림 12의 경우 2.32배 정도 큼을 알 수 있었다. 선형 근사방법은 수렴성은 좋으나 최종위치까지 도달하는데 걸리는 수렴 시간이 신경회로망의 방법에 비하여 1.31~2.4배 정도 길어짐을 알 수 있다.

(a)

(b)

(c)

그림 10. 제어기 성능 비교
 $([x(t_0), y(t_0)] = [2400, 500])$

- (a) 신경회로망의 경우 (b) 근사 격자 최적제어 경우
 (c) 선형근사 최적제어 경우

Fig. 10 The comparison of controllers performances $([x(t_0), y(t_0)] = [2400, 500])$

- (a) Neural network case (b) Approximate grid case
 (c) Linear interpolation case

(a)

(b)

(c)

그림 11. 제어기 성능 비교
 $([x(t_0), y(t_0)] = [3800, 2150])$

- (a) 신경회로망의 경우 (b) 근사 격자 최적제어 경우
 (c) 선형근사 최적제어 경우

Fig. 11 The comparison of controllers performances $([x(t_0), y(t_0)] = [3800, 2150])$

- (a) Neural network case (b) Approximate grid case
 (c) Linear interpolation case

(a)

(b)

(c)

그림 12. 제어기 성능 비교
 $([x(t_0), y(t_0)] = [5200, 2500])$

- (a) 신경회로망의 경우 (b) 근사 격자 최적제어 경우
 (c) 선형근사 최적제어 경우

Fig. 12. The comparison of controllers performances $([x(t_0), y(t_0)] = [5200, 2500])$

- (a) Neural network case (b) Approximate grid case
 (c) Linear interpolation case

신경회로망에 의한 최적제어기의 성능이 다른 최적 제어기들의 성능에 비하여 수렴 시간과 수렴 정도에서 매우 안정적임을 알 수 있다. 근사 격자점 최적제어기의 수렴시간은 신경회로망 제어기에 비하여 비슷하게 나타났으나 수렴 정도가 조금 떨어지며, 선형근사 최적제어기의 경우 수렴 정도는 매우 높게 나타났으나 수렴 시간이 매우 길다는 단점이 있다.

V. 결 론

강한 해류가 있는 바다에서 선박이 일정속도로 움직이면서 진행 각도를 제어량으로 가변시켜 임의의 초기 위치에서 목표위치로 최대한 빠르게 향해하는 Zermelo 항해문제는 calculus of variation의 기법을 이용하여 그 해를 구할 수 있으며, 초기 위치에 대해서 최적제어 각도를 그래프 또는 테이블 형태로 근사치를 구할 수 있다. 이러한 테이블 형태로 주어지는 최적제어 각도 근사치는 테이블을 세분화 할수록 그 정확도가 향상되지만 계산상의 이유로 무한히 세분화할 수 없으므로 근사오차가 발생하게 된다. 본 논문에서 이러한 근사오차를 감소시키기 위해 보간기능을 갖는 신경회로망을 선택하여 최적제어 각도의 정확도를 높였고 그로 인해 선박이 최단시간으로 주어진 목표점에 도달하도록 하였다. 여러 가지 경우의 항해 상황에 대해 컴퓨터 시뮬레이션한 결과 본 논문에서 제안한 방법이 기존의 방법보다 우수함을 확인할 수 있었다.

감사의 글

이 논문은 2012년도 경남과학기술대학교 기성회 연구비 지원에 의하여 연구되었음.

참고문헌

- [1] T. I. Fossen, *Guidance and Control of Ocean Vehicles*, England: John Wiley & Sons Inc., 1994.

- [2] A. E. Bryson, *Dynamic Optimization*, Addison Wesley Longman Inc., 1999.
- [3] F. L. Lewis and V. L. Syrmos, *Optimal Control: 2nd Ed*, John Wiley & Sons Inc., 1995.
- [4] K. Ohtsu, K. Shioji and T. Okazaki, "Minimum-time maneuvering of a ship with wind disturbances," *Control Engineering Practice*, vol. 4, no. 3, pp. 385-392, 1996.
- [5] A. Enes and W. Book, "Blended shared control of Zermelo's navigation problem," *Proc. of American Control Conference*, pp. 4307-4312, June 30, 2010.
- [6] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*, New York: Hemisphere 1975.
- [7] D. Bao, C. Robles and Z. Shen, "Zermelo navigation on Riemannian manifolds," *Journal of Differential Geometry*, vol. 66, pp. 377-435, 2004.
- [8] E. Bakolas and P. Tsiotras, "Time-optimal synthesis for the Zermelo-Markov-Dubins problem: the constant wind case," *Proc. of American Control Conference*, pp. 6163-6168, June 30, 2010.
- [9] Vojislav Kecman, *Learning and Soft Computing*, The MIT Press, Cambridge, MA, 2001.

저자소개

최영규(Young-Kiu Choi)

한국정보통신학회 논문지
제16권 제3호 참조

박진현(Jin-Hyun Park)

한국정보통신학회 논문지
제16권 제3호 참조