

슬라이딩 평면 예측에 기반한 이동 로봇의 경로 추종 제어

문써레*, 최운호**, 박진배*
연세대학교*, 경기대학교**

Sliding Mode Prediction Based Tracking Control for Mobile Robots

Ssurey Moon*, Yoon Ho Choi**, Jin Bae Park*
Yonsei University*, Kyonggi University**

요약 - 본 논문에서는 이동 로봇의 경로 추종을 위해서, 비선형 모델 예측 슬라이딩 모드 제어(nonlinear model predictive sliding mode control) 기법을 제안한다. 본 논문에서 제안한 방법에서는 미래의 슬라이딩 평면을 예측하고, 이에 따른 최적화된 제어를 유도함으로써 슬라이딩 모드 제어기 단독으로 사용하는 제어 시스템에 비해 성능을 향상시킬 수 있다. 마지막으로 컴퓨터 시뮬레이션을 통해 본 논문에서 제안한 제어기의 성능을 검증하고자 한다.

1. 서 론

비 홀로노믹(nonholonomic) 이동 로봇은 강한 비선형적 특성을 가진 시스템이기 때문에 비선형 시스템에 적합한 제어기를 사용해야 한다. 슬라이딩 모드 제어(sliding mode control: SMC) 기법은 외란과 불확실성의 영향을 줄이는데 효과적인 특성 때문에 선형 시스템 뿐만 아니라 비선형 시스템을 다루기 위해 사용되고, 특히 이동 로봇의 제어를 위해 널리 사용되고 있다. 그러나 기존의 SMC 기법은 슬라이딩 평면 근처에서 채터링(chattering) 현상이 발생해서 시스템의 성능을 저하시키는 경향이 있다[1].

모델 예측 제어(model predictive control: MPC) 기법은 최적화 제어 기법 중의 하나로서 많은 장점을 가지고 있다. MPC 기법은 미래의 출력들과 입력들을 예측하고 그에 따른 최적의 입력을 계산한다. 그러나 MPC 기법은 계산량이 많다는 단점이 있다. 많은 계산량에 의해 제어 입력이 늦게 도출되기 때문에 빠른 응답을 필요로 하는 시스템 보다는, 반응속도가 느린 화학 플랜트와 같은 분야에서 많이 사용되었다. 그러나 컴퓨터 과학의 발전에 의해 빠른 연산이 가능해짐에 따라 이동 로봇 같이 빠른 입력 계산을 필요로 하는 시스템에도 예측 제어 기법이 적용되고 있다[2].

한편, SMC와 MPC를 결합한 제어 기법들이 많이 발표되고 있다. 그러한 제어 기법의 장점은 MPC를 이용하여 미래의 슬라이딩 평면을 예측하고, 입력의 변화를 최적화 시킴으로써, 슬라이딩 평면 근처에서의 채터링 현상을 제거할 수 있다는 점이다. 뿐만 아니라 최적 제어의 특성상 시스템의 전체적인 성능 향상도 이루어진다[3].

본 논문에서는 이동 로봇의 경로 추종을 위해서 비선형 모델 예측 슬라이딩 모드 제어 기법을 제안한다. 제안된 방법에서는 미래의 슬라이딩 평면을 예측하고, 이에 따른 최적화된 제어를 유도한다. 마지막으로 컴퓨터 시뮬레이션을 통해 본 논문에서 제안한 제어기의 성능을 검증하고자 한다.

2. 이동 로봇 모델 및 제어기 설계

2.1 이동 로봇 시스템 모델

본 논문에서 사용하는 비 홀로노믹 이동로봇의 이산 시간 모델은 식 (1)과 같다[4].

$$\begin{aligned} x_{k+1} &= x_k + T_c v_k \cos(\theta_k) + T_c \eta_k \sin(\theta_k) \\ y_{k+1} &= y_k + T_c v_k \sin(\theta_k) - T_c \eta_k \cos(\theta_k) \\ \theta_{k+1} &= \theta_k + T_c w_k \end{aligned} \quad (1)$$

여기서 (x_k, y_k) 와 (θ_k) 는 각각 k 스텝(step)에서의 이동 로봇의 중심점의 위치와 방위각이며, 선형 속도(linear velocity) v_k 와 회전 속도(rotational velocity) w_k 가 제어 입력으로 사용된다. 이때, T_c 는 샘플링(sampling) 시간이다. 또한 미끄러짐에 의한 외란을 가지고 있으며 $|\eta_k| < \rho$ 의 범위를 가지고 있다. 식 (1)을 벡터 형태로 나타내면 식 (2)와 같다.

$$X_{k+1} = f(X_k, U_k) \quad (2)$$

여기서 $X_k = [x_k, y_k, \theta_k]^T$, $U_k = [v_k, w_k]^T$ 이다.

2.2 슬라이딩 평면 모델

이동 로봇의 이산시간 모델이 식 (1)로 주어졌을 때, 이동 로봇의 각 상태 x, y, θ 에 대해 슬라이딩 평면을 식 (4)와 같이 설정하자.

$$S_k = \begin{bmatrix} s_k^x \\ s_k^y \\ s_k^\theta \end{bmatrix} = \begin{bmatrix} e_{k+1}^x + \gamma_x e_k^x \\ e_{k+1}^y + \gamma_y e_k^y \\ e_{k+1}^\theta + \gamma_\theta e_k^\theta \end{bmatrix} \quad (3)$$

여기서, 각 상태 오차는 $e_k^x = x_k - x_k^d, e_k^y = y_k - y_k^d, e_k^\theta = \theta_k - \theta_k^d$ 로 정의한다.

이동 로봇의 상태가 목표 슬라이딩 평면 $S=0$ 에 도달했을 때, 상태 오차가 0으로 수렴하려면, 식 (3)의 매개변수 γ 가 $-1 < \gamma < 0$ 범위 내에 존재해야 한다. 이러한 특성 때문에 $|S| \rightarrow 0$ 을 만족하는 입력으로 상태 오차도 0으로 수렴시킬 수 있다.

MPC에서는 입력에 따른 슬라이딩 평면의 변화량이 필요하기 때문에 식 (1)을 이용하여 식 (3)에서 주어진 슬라이딩 평면 모델을 상태와 입력에 대한 모델로 나타내면 식 (4)와 같이 된다.

$$S_k(x_k, u_k) = \begin{bmatrix} (1 + \gamma_x)e_k^x - (x_{k+1}^d - x_k^d) + T_s v_k \cos(\theta_k) + T_s \eta_k \sin(\theta_k) \\ (1 + \gamma_y)e_k^y - (y_{k+1}^d - y_k^d) + T_s v_k \sin(\theta_k) - T_s \eta_k \cos(\theta_k) \\ (1 + \gamma_\theta)e_k^\theta - (\theta_{k+1}^d - \theta_k^d) + T_s w_k \end{bmatrix} \quad (4)$$

2.3 기울기 감소 방법을 이용한 제어기 설계

예측된 슬라이딩 평면을 0으로 수렴 시키고, 입력의 변화량도 고려해주기 위해 비용 함수를 식 (5)와 같이 설정할 수 있다.

$$J_k = S_{k+N}^T Q_N S_{k+N} + \sum_{i=k+1}^{k+N-1} S_i^T Q_i S_i + \Delta U_i^T R \Delta U_i \quad (5)$$

여기서 $\Delta U_i = U_i - U_{i-1}$ 이다.

k 는 로봇의 이동 구간에 대한 각 스텝이며, 각 k 에 대해서 구간 N 을 잡아 최적화 시키는 이동 구간 제어(receding horizon control)의 개념이 이용되었다.

모델 예측 제어 문제에 효과적으로 기울기 감소(gradient descent: GD) 방법을 적용하기 위해 Lagrange 승수(multiplier)와 Hamiltonian 방정식을 이용한 접근 방법이 사용된다. 다음은 Lagrange 승수와 Hamiltonian 방정식을 이용하여 비용 함수를 감소시키는 최적화된 입력을 도출하는 과정이다 [5],[6].

Lagrange 승수 λ 에 식 (2)의 조건 $X_{k+1} - f(X_k, U_k) = 0$ 을 곱하여 식 (5)에 더하면 식(6)과 같이 된다.

$$J_k = S_{k+N}^T Q_N S_{k+N} + \sum_{i=k}^{k+N-1} S_i^T Q_i S_i + \Delta U_i^T R \Delta U_i + \lambda_{i+1} [f(X_i, U_i) - X_{i+1}] \quad (6)$$

간단한 계산을 위해, k 는 0으로 두고 Hamiltonian 함수를 식(7)과 같이 정의하자.

$$H_i = S_i^T Q_i S_i + \Delta U_i^T R \Delta U_i + \lambda_{i+1} f(X_i, U_i) \quad (7)$$

식 (5)의 비용 함수 J 의 증분은 식 (8)로 나타내어진다.

$$dJ = \left[\left(\frac{\partial S_N}{\partial X_N} \right)^T Q_N S_N - \lambda_N \right] dX_N + \frac{\partial H_0}{\partial X_0} dX_0 + \frac{\partial H_0}{\partial S_0} dS_0 + \frac{\partial H_0}{\partial U_0} dU_0 + \sum_{i=1}^{N-1} \left[\left(\frac{\partial H_i}{\partial X_i} - \lambda_i \right) dX_i + \frac{\partial H_i}{\partial S_i} dS_i + \frac{\partial H_i}{\partial U_i} dU_i \right] \quad (8)$$

Lagrange 승수 λ 는 원래 0에 곱해져 있던 상수이기 때문에 임의의 값으로 정해질 수 있다.

$$\lambda_N = \left(\frac{\partial S_N}{\partial X_N} \right)^T Q_N S_N \quad (9)$$

$$\lambda_i = \frac{\partial H_i}{\partial X_i} + \frac{\partial H_i}{\partial S_i} \frac{\partial S_i}{\partial X_i} \quad (10)$$

λ 의 경계 조건이 식 (9)에 의해 유도되므로, 식 (10)을 이용하여 모든 시간 ($i=1, \dots, N$)에 대한 λ_i 를 유도할 수 있다. λ_i 를 식 (8)에 대입하면 식(11)과 같은 비용 함수의 증분이 구해진다.

$$dJ = \sum_{i=0}^{N-1} \frac{\partial H_i}{\partial U_i} dU_i + A \quad (11)$$

여기서 $\frac{\partial H_i}{\partial U_i} = \left(\frac{\partial S_i}{\partial X_i} \right)^T Q S_i + R \Delta U_i + \left(\frac{\partial f_i}{\partial U_i} \right) \lambda_{i+1}$ (12)

이며, A 은 dJ 의 나머지 항들이다.

한편, 제어 입력은 비용함수 J 를 감소시키는 방향으로 변해야 하며, 식 (13)과 같이 유도된다.

$$U_{i+1} = U_i + \Delta U_i = U_i + \epsilon \left(- \frac{\partial H_i}{\partial U_i} \right) \quad (13)$$

이때, 식 (12)와 (13)에 의해, 제어 입력의 변화량 ΔU 는 식 (14)와 같이 구할 수 있다.

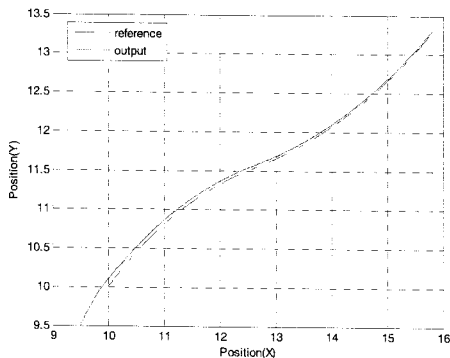
$$\Delta U_i = -\epsilon (I + \epsilon R)^{-1} \left[\left(\frac{\partial S_i}{\partial X_i} \right)^T Q S_i + \left(\frac{\partial f_i}{\partial U_i} \right) \lambda_{i+1} \right] \quad (14)$$

식 (14)를 통해 구해진 최적화 입력 중에서 초기의 입력인 U_{k+0} 만 실제 이동 로봇을 제어하는데 사용되며, 나머지 입력 값들 ($U_{k+1}, \dots, U_{k+N-1}$)을 이용하여 다음 $k+1$ 스텝의 미래 출력 값들을 예측하여, 위의 연산을 반복한다.

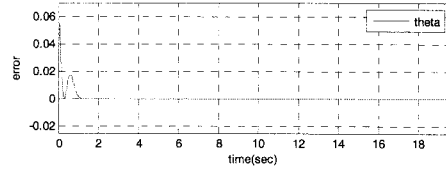
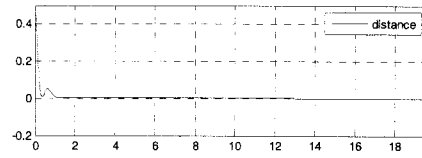
2.4 컴퓨터 모의 실험 및 결과 고찰

본 논문에서 제안한 제어기의 성능을 입증하기 위해 MATLAB을 이용하여 컴퓨터 모의 실험을 수행하였다. $\rho=0.2$, 샘플링 시간이 0.01(sec)인 환경에서 실험이 진행되었으며, 예측 범위(N)는 0.5(sec)로 설정하였다. 나머지 매개변수의 값들은 $Q=4I$, $R=I$, $\epsilon=0.5$ 이다. 또한 기준 궤도의 초기 상태는 $x=10(cm)$, $y=10(cm)$, $\theta=\pi/4(rad)$ 이며 시간 $t=10(sec)$ 일 때를 기점으로 기준 속도 v_x, v_y 이 변한다. 실제 이동 로봇의 초기 상태는 $x=9.5(cm)$, $y=9.5(cm)$, $\theta=\pi/3(rad)$ 이다.

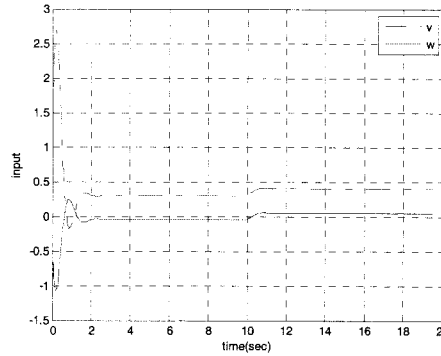
그림 1은 기준 궤도와 이동 로봇의 위치를 나타내고, 그림 2는 기준 모델과 실제 이동 로봇 간의 위치 오차와 각도 오차를 나타낸다. 두 그림의 결과들을 통해 이동 로봇이 기준 궤도와 오차를 극복하고 잘 추종하는 것을 확인할 수 있다. 그림 3은 시간에 따른 이동 로봇의 속도, 즉 제어 입력 신호를 나타내는 그래프이다. 처음 $t=2(sec)$ 이내에 제어 입력이 안정되며, $t=10(sec)$ 일 때, 기준 속도가 변한 후 1(sec) 이내에 제어 입력이 다시 새로운 값으로 수렴한다. 입력이 안정화되기 때문에, 채터링 현상도 발생하지 않음을 확인할 수 있다.



〈그림 1〉 기준 궤도와 이동 로봇의 위치



〈그림 2〉 거리 오차($e_x^2 + e_y^2$)와 각도 오차(e_θ^2)



〈그림 3〉 선형 속도(v)와 회전 속도(w)

3. 결 론

본 논문에서는 이동 로봇의 경로 추종을 위해서, 비선형 모델 예측 슬라이딩 모드 제어기를 제안하였다. 제안한 제어기에서는 비선형 시스템의 제어와 미끄러짐에 의한 외란을 제거하기 위해 슬라이딩 모드 제어기를 사용하였으며, 슬라이딩 모드의 예측을 통해, 채터링을 제거하고, 전체적인 시스템의 성능을 향상시켰다. 또한 컴퓨터 모의 실험을 통해 이동 로봇의 경로 추종 시의 성능을 확인할 수 있었다.

[참고 문헌]

- [1] J.-J. E. Slotine and W. Li, Applied nonlinear control, Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [2] S. J. Yoo, Y. H. Choi, and J. B. Park, "Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: adaptive learning rates approach", IEEE Trans. Circuits and Systems, vol. 53, no. 6, pp. 1381-139, 2006.
- [3] L. Xiao, H. Su, X. Zhang and J. Chu, "Discrete variable structure control algorithm for nonlinear systems via sliding mode prediction", in Proc. Conf. American Control Conference, pp. 4712-4717, June 2006.
- [4] M.L. Corradini, T. Leo, G. Orlando, "Experimental testing of a discrete-time sliding mode controller for trajectory tracking of a wheeled mobile robot in the presence of skidding effects", Int. J. Robot. Syst., vol. 19, pp. 177 - 188, 2002.
- [5] Allgower, F and Zheng, A, Nonlinear model predictive control, Progress in Syst and Control Theory, vol. 26, 2000.
- [6] H. J. Kim, D. H. Shim and S. Sastry, "Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles", in Proc. Conf. American Control Conference, pp. 3576 - 3581, 2002.