

경로예측이 가능한 이동물체와 이동로봇간의 Rendezvous Point에 관한 연구

윤정훈* . 이기성

A Study on Rendezvous Point between the Mobile Robot and Predicted Moving Objects

Junghoon-Youn* . Keeseong-Lee

Dep. Electrical and Control Engineering of Hong-Ik Univ.

Abstract - A new navigation method is developed and implemented for mobile robot. The mobile robot navigation problem has traditionally been decomposed into the path planning and path following. Unlike tracking-based system, which minimize intercept time and moved mobile robot distance for optimal rendezvous point selection. To research of random moving object uses algorithm of Adaptive Control using Auto-regressive Model. A fine motion tracking object's trajectory is predicted of Auto-regressive Algorithm. Thus, the mobile robot can travel faster than the target within the robot's workspace. We can select optimal rendezvous point of various intercept time.

체와의 거리를 측정 후 이동물체와의 최적 점을 계산하여 로봇이 이동한다. 본 논문에서는 camera를 통해 모든 정보를 입수한다는 가정 하에 camera와 이동로봇간의 통신 delay는 무시할 수 있을 정도의 값으로 정하였다. 또한, 이동로봇과 이동물체의 움직임은 camera를 통해 입수되므로 로봇 및 이동물체는 모두 평면에 위치해 있는 것처럼 여겨질 것이다. 따라서, 3차원 상에서 이루어지는 것이 아니므로 계산량이 현저히 줄어드는 것을 알 수 있고, 결국 이동물체의 움직임을 파악하여 이동로봇이 그만큼 빠른 시간 내에 이동할 시간적 여유를 가질 수 있다. 이동물체의 움직임을 사용자가 정한 몇 가지 경우에 대하여 이동물체의 이동에 따른 이동로봇의 적응력을 높일 것이다. 각각의 경우에 대하여 최적의 rendezvous point를 구할 때 이동물체와 이동로봇과의 만나는 시간 및 좌표계에서의 점들을 살펴볼 것이다.

1. 서 론

현재, 인간이 3D(Dirty, Dangerous, Difficult) 직종을 기피하게 되면서, 스스로 판단하고 행동하는 로봇이 개발되어 투입되고 있다. 하지만, 작업환경이 제약되어 있는 상황에서의 업무를 하던 매니플레이터(Manipulator)와 같은 로봇만으로는 인간이 원하는 것을 모두 만족시키지는 못했다. 결국, 작업환경이 비교적 자유롭고, 활동성이 필요한 업무를 위해 자율이동로봇(Autonomous Mobile Robot)에 더욱 많은 관심과 연구가 되어 왔다.

Random하게 이동하는 이동물체의 움직임을 과거부터 현재까지의 궤적에 대한 data를 계산하고, 앞으로의 움직임을 예상하여 predicted rendezvous point를 연산을 통해 robot이 이동물체를 intercept 한다.

본 논문에서는 이동로봇이 임의의 궤적을 추종하여 비선형적으로 변화하는 로봇의 동적 특성을 효과적으로 밝히기 위해 Adaptive Control 알고리즘을 적용한다.

본 논문에서 제안된 알고리즘을 이용한 Simulation을 통하여 알고리즘의 유용성을 보여줄 것이다.

2. 본 론

2.1 Predicted Trajectory of Objects

본 논문에서는 Rendezvous Point를 찾는 것이 목표이다. 로봇은 주어진 환경 속에서 이동물체의 예상된 경로를 계산하여 로봇을 이동시킨다. 따라서, 로봇이 이동물체를 추적하기 위해 필요한 작업에 대해 먼저 설명할 것이다. 먼저, 로봇은 다음과 같은 작업에 의해 이동물체의 예상경로와 rendezvous point를 찾아 이동할 것이다.

자율이동로봇은 한 곳에 고정되어 있지 않고, 스스로 환경을 판단하여 자신의 위치를 이동하는 능력을 가진 로봇으로, 매니플레이터에 비해 가장 큰 장점이라고 부각시킬 수 있는 점은 바로 작업환경의 제약이 상당히 적어진다는 점이다. 하지만, 그만큼 로봇의 능력도 더욱 요구되고 있다.

일반적인 자율 주행 로봇 연구의 궁극적인 목적 중 하나는 위험하고 밀집된 환경에서 안전하게 주어진 임무를 수행할 수 있는 로봇을 만드는 것이라 할 수 있다. 예를 들어, 실내 환경에서 인간을 보조하는 서비스 로봇(Service Robot)은 예측하지 못한 주변 환경의 변화에 대해 빠르게 반응할 수 있어야 하며, 외부 환경이 다양하더라도 동작할 수 있어야만 한다.

먼저, Vision에 의해 물체를 인식하여 주위환경에 대한 정보를 지도작성으로 표현한다. 이때, 정적 장애물의 여부를 확인하게 된다. 먼저, 장애물이 없는 경우는 로봇의 위치를 확인 후, rendezvous point를 계산하여 바로 예상된 위치로 이동한다. 하지만, 정적 장애물이 존재할 때는 장애물을 회피한 후에 다시 rendezvous point를 잡고 예상된 위치로 이동하게 된다. 목표된 지점으로 이동하여 두 물체가 만난 점을 찾고, 이 때까지 소요된 시간을 알게 된다.

이런 점을 감안하여 이동로봇의 작업환경 상에 이동물체가 발견되었을 때, 로봇이 이 물체의 움직임에 대해 대처할 수 있는 능력을 만들기 위해 rendezvous point에 대한 문제점을 제기하게 되었다.

본 논문에서 가장 중점을 두고자 한 부분은 이동물체와 이동로봇간의 최적의 rendezvous point를 찾는 것이다. 이때 simulation 상의 공간을 직교 좌표계를 사용할 것이다. 이동물체의 정보를 camera를 통해서 monitoring한 data, 즉 이동물체의 직교 좌표계에서의 현 위치와 속도 및 방향 등의 data를 이동로봇에 보내 이동로봇과 이동물

2.1.1 Adaptive Control using Auto-Regressive Model

Kuvio 와 Guo에 의해 제안된 방법으로,

manipulator의 입력과 출력에 맞도록 autoregressive controller를 사용하여 적응력이 있고, 스스로 튜닝(tuning) 할 수 있는 알고리즘이다. 이 control 알고리즘에서는 joint사이의 interaction force를 무시하였다.

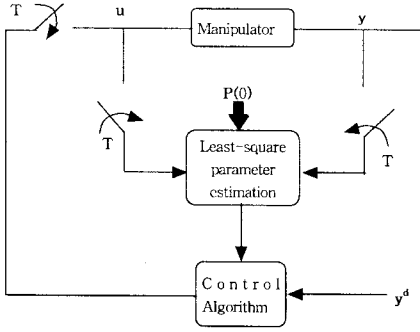


Fig. 1 Autoregressive Model을 이용한 Adaptive Control

이 control system의 block diagram을 Fig. 1에서 보여주고 있다.

여기서, u_i 는 i 번째 조인트의 입력 토크(torque)이고, y_i 는 manipulator의 angular position을 의미한다.

여기서, autoregressive model에 의해 입력과 출력을 각각 u_i 와 y_i 로 나타내면, 출력 y_i 는 아래와 같이 나타낼 수 있다.

$$y_i(k) = \sum_{m=1}^n [a_i^m y_i(k-m) + b_i^m u_i(k-m)] + a_i^0 + e_i(k) \quad (1)$$

여기서, a_i^0 는 constant forcing term이고, $e_i(k)$ 는 u_i 와 $y_i(k-m)$ 이 독립적이고, zero 평균을 가진 화이트 가우시안 노이즈(White Gaussian Noise)로 가정될 때의 모델링 error이다.

a_i 를 i 번째 parameter vector라 하면,

$$a_i = (a_i^0, a_i^1, \dots, a_i^n, b_i^0, b_i^1, \dots, b_i^n)^T \quad (2)$$

식(2)와 같이 matrix로 표현될 수 있다.

Manipulator에서 사용된 이 adaptive control을 mobile robot에도 적용시켜 비선형적인 움직임을 보이는 이동물체와 robot의 rendezvous point를 찾고, 비선형적인 움직임을 보이는 물체에 대한 이동로봇의 적응력을 높일 것이다.

2.2 Random한 Moving Object의 Predicted Trajectory

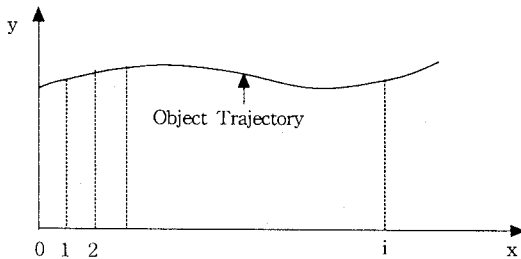


Fig. 2 Random한 Object Trajectory

여기서, x : time(sec)

y : object position(x_0, y_0)

1장에서 언급했던 내용을 mobile robot에 적용하기 위해 Fig2처럼 물체가 움직이는 비선형 물체라고 가정해 보자.

본 논문에서는 Work Space를 Vision System에 의해서, monitoring하기 때문에 이동물체의 초기 위치와 초당 물체의 움직임을 계산해 낼 수 있다. 식(1)의 parameter중에서 입력변수(u)는 본 논문에서 고려하는 상황에서는 무의미하므로 제거할 수 있다. 따라서, 본 논문에서 random하게 움직이는 물체의 궤적을 표현하기 위해 다음과 같이 변형하였다.

이 궤적이 바로 우리가 찾고자하는 rendezvous point의 집합체이기 때문이다. 이 비선형 물체 궤적을 구할 수 있다면, 로봇의 위치와 속도 및 장애물의 여부를 고려하여 rendezvous point를 계산해 낼 수 있다.

$$y(k) = \sum_{m=0}^n a_m * y(k-m) \quad (3)$$

식(3)을 다시 표현하면 아래와 같다.

$$y(k) = \beta(k-1) * a(k) \quad (4)$$

식(3)에서 i 번째에서의 $a(k)$ 와 $\beta(k-1)$ 은 아래와 같다.

$$\beta(k-1) = (a_1, a_2, a_3, \dots, a_n)^T \quad (5)$$

$$a(k) = (y(k-1), y(k-2), \dots, y(k-n)) \quad (6)$$

여기서, 현재 object의 상태를 나타낸 식이다. 따라서, 이 식을 이용하여, 미래 상태의 $y(k+1)$ 값도 예측할 수 있다. 미래의 값 또한, 과거의 상태들의 집합이기 때문이다.

이제 우리는 이 시스템에 대한 parameter를 결정해 주어야 한다. Parameter를 결정하기 위한 식은 아래와 같다. (Recursive Identification Method)

$$S(k) = \frac{1}{\mu} [S(k-1) - \frac{S(k-1)\alpha(k)\alpha^T(k)S(k-1)}{\mu + \alpha^T(k)S(k-1)\alpha(k)}] \quad (7)$$

$$\hat{\beta}(k) = \hat{\beta}(k-1) + S(k)\alpha(k)[y(k) - \hat{\beta}^T(k-1)\alpha(k)] \quad (8)$$

여기서, μ 는 forgetting vector로서, $0 < \mu \leq 1$ 구간에서 존재하게 된다. 현 시점 $y(k)$ 에서 볼 때 과거의 값들, 즉 $y(k-1), y(k-2), \dots, y(k-n)$ 의 값들의 가중치가 다르다는 것을 의미한다. 가까운 과거의 값일수록 바로 예측될 미래의 값에 더 많은 영향을 준다는 것을 의미한다.

$\hat{\beta}$ 로 표시되는 것은 미래시점의 예측되어질 값들이다. 식(8)에서 $[y(k) - \hat{\beta}^T(k-1)\alpha(k)]$ 값은 error에 대한 수식적인 값이다.

여기서, $S(k)$ 는 least square parameter로서 error 값을 최소화시킬 수 있는 parameter의 값이다.

$\hat{\beta} = (\hat{a}_1, \hat{a}_2, \hat{a}_3, \dots, \hat{a}_n)$ 로 정의될 수 있다. $\hat{\beta}$ 의 각각의 element들도 미래의 값들이다.

2.3 Rendezvous Point 계산

여러 개의 rendezvous point가 존재할 수 있지만, optimal한 point를 찾기 위해서는 point를 찾는데 중요한 요소가 되는 것은 intercept하는 데 소요되는 시간과 로봇이 이동한 궤적의 거리가 모두 고려되어야 한다. 로봇의 위치와 속도 및 Work Space 상에 처음 나타났을 때의 이동물체의 위치와 시간에 따른 위치변화를

과약하여 계산을 하게 된다. 여기서, 이동물체의 변화하는 경로는 앞에서 서술했던 Auto-Regressive Model 에 이용된 Adaptive Control을 사용하게 된다.

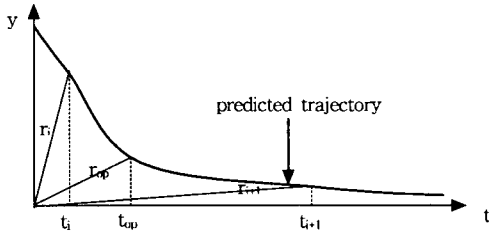


Fig.3 optimal time & position에 대한 고려

이에 따른 물체의 경로를 과약한 후에, 이동물체를 intercept할 수 있는 시간을 계산하게 되는 것이다.

초기의 이동로봇의 위치를 (x_r, y_r) 라하고, predicted rendezvous point를 (x_i, y_i) 라 하자. 또한, 속도에 대해서도 고려해야 한다. 이동로봇의 속도는 v_r 로 이미 알고 있다. 또한, 이동물체의 속도도 앞장에서 설명한 식들에 의해 단위시간당 이동물체가 움직인 거리와 방향을 계산해 낼 수 있다. 이 속도를 v_o 라 하자.

그러면, 우리는 여러 개의 intercept time을 알 수 있고, 그 중에서 소요된 시간과 로봇이 이동한 거리를 고려해 보면, optimal한 rendezvous point를 구할 수가 있다.

$$t_i = \frac{\sqrt{(x_i - x_r)^2 + (y_i - y_r)^2}}{v_r - v_i} \quad (9)$$

여기서, 시간이 i 인 점을 지날 때 만난다고 가정하면, v_i 는 이동물체의 속도를 의미하는 것이다. 이때, t_i 는 intercept를 할 수 있는 시간이다. 몇 가지의 경우가 존재할 수 있다. 이때, 그 중에서 가장 optimal한 조건에 맞는 것을 선택하여 rendezvous point로 선택하여야 할 것이다. 이동로봇의 속도(v_r)은 항상 이동물체의 속도(v_i)보다 빨라야 한다. 로봇과 이동물체는 등속도 운동을 한다는 가정 하에 Simulation을 시행하였다.

여기서, optimal rendezvous point는 시간과 거리가 가장 적은 값을 갖는 point에서 갖게 된다.

2.4 모의 실험 및 결과

모의실험을 위해 아래와 같이 가정을 하였다.

1. 이동로봇은 좌표계에 원점에서 항상 시작한다는 가정 하에 모의실험을 하였다.

실험 결과치	1회	2회	3회	4회	5회	
Interception Time(Sec)	2.56	1.71	1.57	1.24	1.12	
Rendezvous Point(x, y)	x	6.86	4.35	3.67	2.36	2.03
	y	2.68	2.75	2.96	2.89	2.71

- 이동로봇의 속도는 v_r 이고, 이미 알고 있는 값이다.
- 이동물체의 초기위치와 과거의 위치 data 값을 임의로 정해주었다.
- 이동물체와 이동로봇의 속도는 등속이다. ($v_r > v_i$)

기존의 Data와의 비교 분석한 결과가 아래와 같다.

	1회	2회	3회	4회	5회
본 논문	2.56	1.71	1.57	1.24	1.12
기존 방법		1.69	1.25	1.24	1.18

본 논문에서 Simulation한 결과가 위와 같이 나타났다. 기존 방법으로 제시된 data는 매니플레이터로 실험한 결과이지만, 비슷한 상황설정으로 본 논문에서 제시한 Auto-Regressive 에 의한 방법이 Update하면 할수록 결과가 좋게 나옴을 알 수 있다. 본 논문에서 제시한 방법 또한 기존에 제시한 알고리즘에 결코 뒤처지지 않음을 알 수 있다.

3. 결 론

본 논문에서는 되도록 산업현장에서 실제로 발생할 수 있는 상황에 적용하기 위해 random한 움직임을 보이는 이동물체를 고려하여 Adaptive Control에 대해 논하고, Simulation 해 보았다.

여기서, 적응력을 높이기 위해 Off-line상에서의 이동물체의 움직임에 따른 로봇의 움직임을 고려하여 환경에 대한 적응력을 높일 수 있었다. On-line 상에서의 이동물체 출현에 대해 적응하도록 하여 최적의 rendezvous point를 찾는 데 성공하였다.

하지만, predicted rendezvous point를 계산하는데 걸리는 시간에 의해 시간적인 면에서 비교적 만족스럽지 못했다. Adaptive control을 이용해서 rendezvous point를 구하는 데는 성공하였다.

좀 더 좋은 결과를 얻기 위해서는 좀더 많은 회수의 반복이 필요하다. 또한 data의 update과정이 중요시된다.

Simulation을 하기 위한 Work Space를 작게 두어서 좋은 값을 얻을 수 있었다.

[참 고 문 헌]

- IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 28, No. 2, April, 1998.
- K.S.FU, R.C.Gonzalez, C.S.G.Lee, "Robotics P244-2 46, 1986.
- Ahrikencheikj Seireg, "Optimized-Motion Planning," wiley_inter science Publishers.
- Kikuo Fujimura and Hanan Samet, "Motion Planning in a Dynamic Domain", IEEE International Conf. on Robotics & Automation, pp.324-330, 1990.
- M.D.Adams, Huosheng Hu and P.J. Probert, "Towards A Real-Time Architecture for Obstacle Avoidance and Path Planning in Mobile Robots", in IEEE J.Robotics and Automation, pp.584-589, 1990.
- 이주호, "자율 주행 로봇의 새로운 경로 계획법과 주행 알고리즘에 관한 연구", 고려대학교 석사학위논문, 1995.
- 윤재원, "로봇의 제적 추종을 위한 안정한 관측기에 기초한 제어기 구현", 광주 과학 기술원 석사학위논문, 1999.
- 이재원, "여유 자유도 로봇의 시간 최소 경로 계획", 한국 과학 기술원 석사학위논문, 1997.
- 김병기, "휠 구동 이동로봇의 경로추적 제어기 구현", 포항공대 대학원 석사학위논문, 1998.
- 조덕영, "이동 로봇의 조타 제어기 설계와 직선 추종 주행 실험에 관한 연구", 한국 과학 기술원 석사학위논문, 1987.