

< 자율 이동 로봇의 허접한 기구학 >

로봇을 만들려면, 기본적으로 그 기구적 특성 정도는 알고 있는 것이 좋다고 생각합니다.

물론 기구학이란게 우습게 보이지만 사실은 천재성이 요구되는(?) 심오한데가 있는 학문이므로, 저도 별로 자신은 없지만 말입니다.

교과서적인 나열보다는, 좀 쉽게 제가 이해하고 있는 정도를 소개해 봅니다.

1. 자유도 (DOF : Degree Of Freedom)

보통 3차원 공간상에서 물체의 위치를 표시하기 위해서는 직교 좌표계를 사용하지요.

(x, y, z)

이렇게 말입니다. 여기에 더하여, 물체의 자세(방향) 까지 완벽하게 나타낼려면 세 개의 정보가 더 필요할 것입니다.

$(x, y, z, \alpha, \beta, \gamma)$

이렇게요. 음... 감마 문자가 이상하게 찍히고 있군요...

항공역학 쪽 사람들은, 이렇게 추가된 자세에 관한 정보 세 가지를 각각 요잉(Yawing)/피칭(Pitching)/틸팅(Tilting) 으로 말하고....

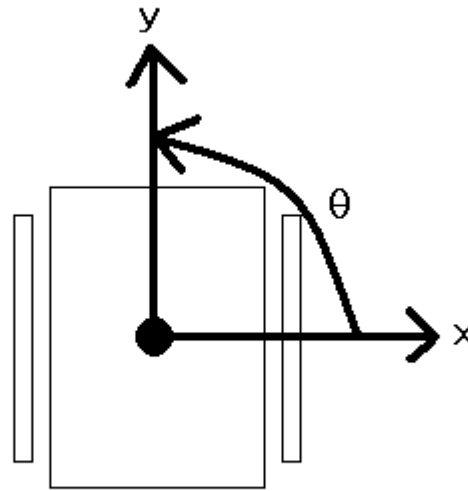
영화찍는 카메라맨 아자씨들은 뭐 팬(Pan)/롤(Roll)/틸트(Tilt) 라고 부르더군요.

암튼 전부 같은 소리인 듯 하고, 여기서는 별 상관없는 거니까...

그나저나, 이렇게 6개의 좌표 정보가 필요한 경우... 우리는 "위치및 자세 제어를 위해 최소 6자유도가 필요하다"고 말합니다.

'자유도'란 용어의 정확한 정의에 대해서는 생략합니다. 책에선 너무 어려운 말들이 튀어나오더군요.

Pocket-bot 의 경우, 3차원 공간이라기 보다는 2차원 평면상에서 움직인다고 가정할 때... 완벽한 움직임을 보장하기 위해서는 최소 3자유도가 필요합니다.

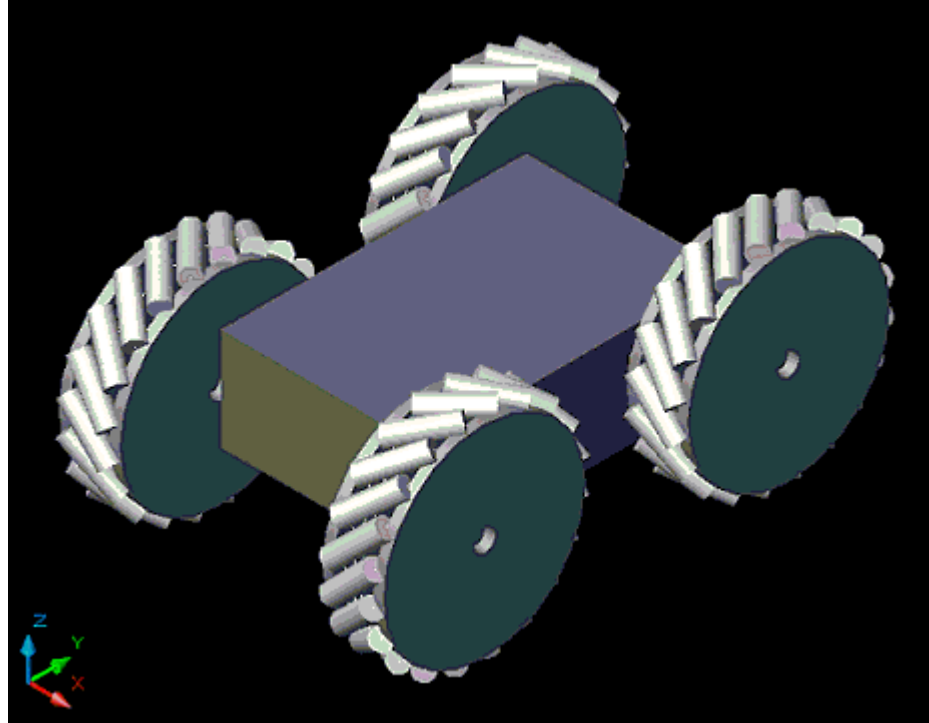


만일 이렇게 x 축, y 축 그리고 θ 방향이 독립적으로 움직일 수 있는 기구적 구조가 존재한다면, 정말이지 마음대로 위치 및 자세가 제어될 것입니다. 그런데... 불행히도 그런 구조는 실질적으로 구현하기가 힘들죠.

우리가 타는 자동차... 스티어링 휠로 방향을 조절하죠. 이놈은, y 축 방향으로 자유롭게 움직일 수 있지만... x 축 및 θ 방향으로의 운동은 완전히 독립되질 못하고 구속되어 있습니다. 즉 완벽한 3자유도의 기구 구조가 아닙니다. 그래서인지, 주차장에서 땀빼는 사람들이 많겠죠?

그럼 무한궤도가 달린 탱크 식의 기구 구조는 어떨까요? 자동차 보다는 훨씬 낫죠... 제자리에서 회전도 되고, 즉 y 축 및 θ 방향의 운동은 독립적으로 확보가 됩니다. 그러나 역시 x 축 방향으로의 운동은 불가능합니다. 역시 이런 형태도 완벽한 3자유도의 확보에는 실패입니다.

그럼 정녕 3자유도 확보된 바퀴달린 기구는 없는 걸까요? 그런게 존재합니다. 실용화 된 것도 있구요. 아래 그림을 보시면 이해가 되시리라 생각합니다.



많이 보시던 형태의 휠 구조이죠. 뭐 일본인이 만든 마이크로 마우스 사진에서 본 것도 같구요. 이런 구조를 처음 고안한 곳은 미국의 카네기 멜론 대학인 것으로 알고 있는데... 70년대 중반인가 그럴 겁니다. 4개의 바퀴축에 독립적으로 구동 모터를 달아놓고, 회전방향을 여러가지로 바꾸어 보면... 이 로봇은 완전히 바닥에서 뚫뚫 떠서 다니는 듯한 느낌이 듭니다. 좌우측으로(x축 방향) 그대로 움직여 버리니 신기하죠.

이런 휠 구조로 만든 AGV 위에 올라타본 경험이 있는데... 승차감이 장난 아닙니다. 꿀렁꿀렁~ 멀미날 정도... 즉 에너지 손실이 많다는 소리겠죠. 그래서, 많이 사용되고 있지는 않습니다. 뭐 실험용 정도로 사용될까...

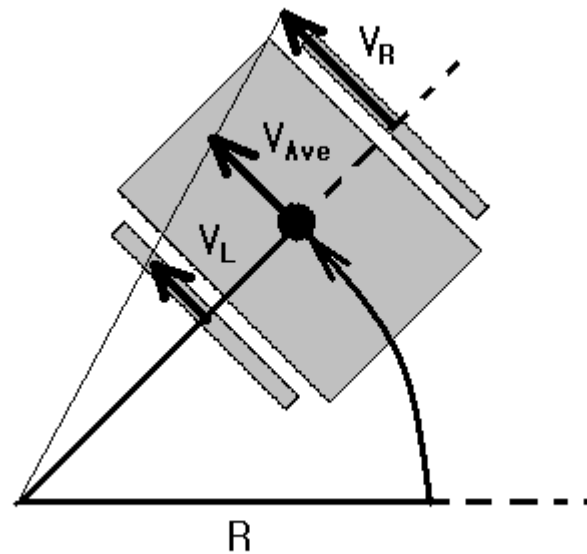
게다가, 3자유도 구현을 위해 4개의 구동 모터를 사용하여야 한다는 점도, 적극적인 채용에 제동을 거는 요인인 듯 합니다.

2. Pocket-bot 의 기구

Pocket-bot 의 경우, 타이밍 벨트를 무한궤도 삼아 돌리는 식이므로, 위에서 소개한 것 중 두 번째 경우에 해당하는군요.

(x,y,θ) 절대 좌표계에서, A (x_1,y_1,θ_1) 에서 B (x_2,y_2,θ_2) 로 움직이려면, 해답이 없는 경우(특이점)가 생긴다는 이야깁니다. 이래가지고선 단순한 선형 제어기만 가지고 원활한 활동을 보장하기 힘들어 지겠군요. 그래서 행동의 시퀀스를 준다든지 하는 보완책이 필요해 질 것입니다.

비록 그런 안좋은 경우가 있음에도 불구하고... 어느정도 일반화된 기구적 운동의 플랜은 가지고 있어야 할 것이 아닌가 생각합니다. 그래서 나름대로 이런 기구의 운동에 관한 일반적인 모델링을 시도해 봅니다.



윈도우 그림판으로 너저분 하게 그려 보았습니다. 로봇이 좌측으로 턴하는 상황을 나타내고 있습니다. 이 그림이 나타내는 내용은, 좌우측 바퀴의 속도차이에 의해 회전반경 R 이 결정된다는 것입니다.

한편... 이 그림의 내용을 확장하여 생각해 보면, 회전반경 $R < 0$ 일때는 우측으로 턴하는 경우가 될 테고, $R = 0$ 일때는 제자리 회전이며, $R \rightarrow \infty$ 라면 직진하는 상황을 나타낸다고 보시면 되겠습니다. 즉 이 기구에서 발휘될 수 있는 모든 경우의 운동을 하나의 식만 가지고 풀이해 낼 수 있겠다는 것입니다.

턴 중심과, 속도 벡터 화살표가 그려주는 직각삼각형이 보이시지요? 그렇죠... V_L 과 V_R 의 차이에 의해 이 직각삼각형의 모양이 결정될 테고, 따라서 R 이 결정된다

는... 그런 단순한 논리.. 헤헤..

자.. 그럼, 이런 아이디어를 이용해서 어떤 함수를 만들어 놓으면, 아규먼트로 R 이랑 V_{Ave} 만 주면 그 함수는 연산 결과로써 V_L 과 V_R 이 튀어나와서 모터를 돌려준다는 것이죠. 물론 이때 서보 모터는 속도제어 모드로 해 놓아야겠군요!

이걸 라인 트레이서 만들때의 경우로 특화시켜 생각해 봅시다.

서울시립대 라인트레이서 대회 규정을 보면... 라인의 턴 반경이 표준으로 정해져 있군요? 아님 마이크로 마우스의 경우도, 턴 반경이 거의 정해져 있구요... 경기장 규격이 있으니. 그럼 R 은 몇 몇 경우 밖에 안될테니, 결정하기 쉬운 것일테구... 로봇의 속도야 상황에 맞게 맞추어 주면 될 꺼구요. 구현이 어렵지 않은 않을 것으로 생각합니다.

< 끝! >

여기서 조금 더 발전될 이야기의 주제는, 로봇이 어떤 궤적을 그리면서 나아가야 목표 좌표에 도달할 수 있을까 하는 역기구학(Inverse Kinematics) 비스무레한 것입니다. 그 이야기는 필요할때 나중에 계속 하도록 하겠습니다.

그때는, 비록 속도제어 모드이긴 하지만 그 속도 데이터를 계속 적분해 나가면서 현재 위치를 파악해 나가는... 뭐랄까 좌표변환? 그런 이야기도 할 것 같습니다.

너무 허접한 이야기라고 비웃지만 마시구요... 혹시 모르는 분들이 있을까봐 정리 차원에서 올려봅니다.