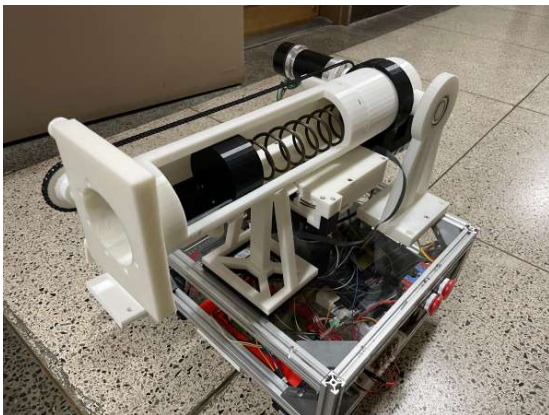


화재진압 로봇의 작업 알고리즘

지도 교수: 최 정 수
영남대학교 로봇기계공학과

1. 서 론

최근 대전 현대 아울렛 화재, 화성 제약 회사 공장 화재 등 대형화재의 발생이 잦아지는 추세이다. 대형화재는 인명피해와 물질적 손실의 규모가 크기 때문에 대형화재를 방지하기 위해 초기 화재 진압이 중요하다. 초기 화재의 경우 소화기 한 대로 진압할 수 있지만, 일정 시간 경과 후에는 소화기로 진압이 힘들어진다. 또한 효과적인 화재 진압을 위한 골든 타임은 5분으로 알려져 있지만, 실제로는 7분 내로도 도착하지 못한다. 이를 방지하기 위해 국내외 다양한 화재 진압 로봇이 존재한다. 하지만 실제 현장 투입 비율은 높지 않고, 사람이 직접 외부에서 조작해주는 형태로 실용성에 의문이 제기된다. 따라서 위에서 언급한 다양한 문제점들을 해결하기 위해 위 그림과 같은 화재 진압 로봇을 제작하였다. 본 논문에서는 화재 진압로봇의 작업 수행 알고리즘을 제안한다.



작품 사진

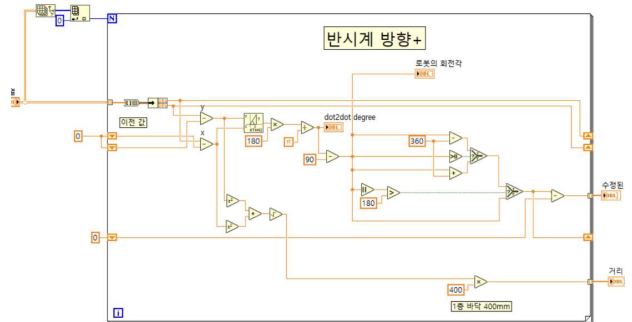
제작한 로봇은 화재를 탐지한 CCTV의 위치 정보를 토대로 발화 지점까지 스스로 이동한다. 카메라에 화재가 탐지되면, 화재까지의 거리를 이용해 발사 각도를 조정한 후 소화용액을 투척하여 불을 끈다. 이동을 위한 바퀴 모터 2개, 각도 조정을 위한 모터 1개, 발사를 위한 모터 1개로 4개의 모터를 사용하였고 화재 감지 및 거리 측정을 위해 depth camera를 사용하였다.



동작 순서로는 'wheel' sub VI에서 A* 알고리즘을 이용하여 목적지까지 이동하고, 'angle'에서 카메라가 부착된 포신의 각도를 조정한다. 'cannon'에서는 탐지된 불과의 거리를 측정하여 설정된 각도에 따라 필요한 힘만큼 스프링을 당겨준다. 그 후 포신이 들리면서 미세하게 변경된 각도를 바퀴 구동을 통해 조정해준다. 'trig'는 스프링을 잡고 있던 부분을 모터를 이용해 풀어주어 투척용 소화기가 발사되며 수행이 완료된다.

2. 알고리즘

2.1 map 읽기



labview 코드



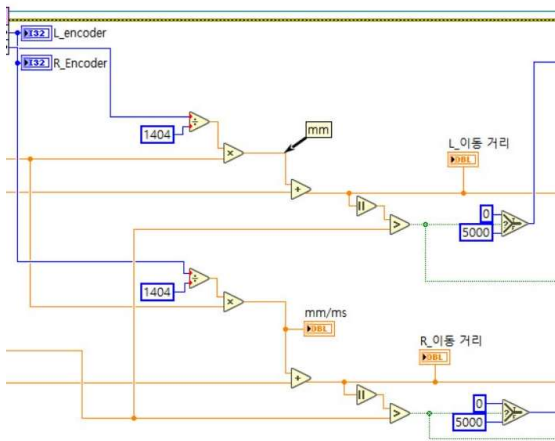
그림1

그림2

input 데이터는 A*의 연산을 통한 nx2 행렬이고, output 데이터는 각 픽셀로 이동하기 위한 각도와 각 좌표 사이의 거리 정보를 담은 행렬이다. X를 두 좌표의 x축 거리차, Y를 두 좌표의 y거리차로 두고, $\text{atan2}(y,x)$ 로 두 점의 각도를 계산한다. 이때 차체가 회전을 하지 않은 상태는 직교 좌표계의 90도에 해당하는 offset 및 이전 좌표

계에서 회전한 상태의 각도가 있기때문에 atan2의 결과에 추가적인 연산이 필요하다. atan2의 연산 결과는 주각 $(-\pi, \pi]$ 형태로 나오지만 offset값을 뺀을 때는 주각을 벗어나는 경우가 생긴다. 따라서 주각을 벗어나고, 0보다 큰 경우 360을 빼주고, 주각을 벗어나고, 0보다 작은 경우 360을 더해주어 차체의 회전각을 주각으로 만들어준다. 만약 그림1과 같이 (1, 1), (0, 2), (0, 3)의 atan2를 이용해 각도를 구하면 45, 135, 90으로 로봇이 현재 회전되어 있는 상태는 고려하지 않은 값이 출력된다. 따라서 로봇의 각도를 고려하면 그림2와 같이 -45, 90, +45로 올바른 회전각도가 구해지는 것을 확인할 수 있었다.

2.2 바퀴 구동



labview 코드

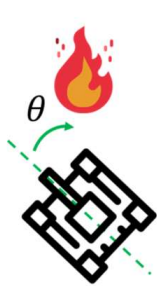


그림1 정렬 전



그림2 정렬 후

바퀴 구동 불을 발견하기 전과 후로 두가지 구동모드가 있다. 첫번째로 화재 발견 전의 구동상태는 2.1에서 구한 좌표 사이에서의 각도와 거리대로 구동하는 것이다. 이때 엔코더의 A, B상을 4체배로 연산하여 정확도를 높여주었고 0.25m/s의 속도로 이동하도록 pwm을 맞춰주었다. 체배 연산은 xor(배타적 논리합)을 이용하

여 A상과 B상 rising, falling을 얻는다. 만약 A상에서 rising, falling이 발생한다면 B상의 값과 비교하여 바퀴의 구동방향 또한 얻을 수 있게 된다. 코드 상에서 timed loop로 돌면서 양 바퀴가 각각 자신이 이동해야 할 만큼 구동이 끝나면 pwm이 0, 구동이 끝나지 않은 상태이면 5000으로 입력된다. 원하는 pwm을 input으로 넣으면 FPGA내에서 20000 중 input만큼 true, $(20000 - \text{input})$ 만큼 false가 되어 바퀴가 구동된다.

두번째 구동모드는 화재를 발견한 이후이다. 화재 발견 후에는 Jetson Nano에서 신호가 들어와 더 이상의 경로 이동을 진행하지 않고 투척 시 정확성을 높이기 위해 불과의 각도를 맞춘다. 포신에 부착되어 있는 depth 카메라를 이용해 픽셀 칸마다의 거리를 실제 거리와 비교하여 (a)그림의 θ 를 구하고 해당 값만큼 로봇을 미세하게 조정시켜준다.

2.3 대포 각도 조정



labview 코드1

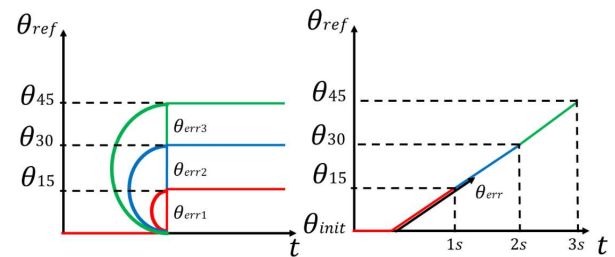


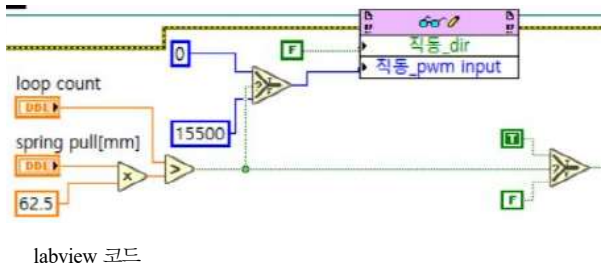
그림1 reference 수정 전

그림2 reference 수정 후

바퀴의 정렬이 끝나면 포신의 각도를 특정 각도에서 유지한 상태로 나머지 동작을 수행해야한다. 스프링 힘이 고정되어 있기때문에 불과의 거리가 달라질 때마다 대포의 각도가 달라졌고, 따라서 적정 각도에서 머무를 수 있도록 힘제어를 해주었다. input으로 특정 각도가 들어오면 해당 값만큼 대포 각을 조정하기위해 PID제어를 사용하였다. 그림1과 같이 reference를 시간에 상관없이 단순히 내가 원하는 각도로 설정하면 input angle이 너무 작을 시

에는 움직이지 않는 경우가 발생하였고, 각도가 너무크면 reference가 계단형으로 갑자기 튀는 형태로 힘을 이기지 못하고 차체가 파손되는 경우가 발생하였다. 따라서 timed loop를 사용하여 그림2와 같이 시간에 따라 reference가 0부터 서서히 증가하는 형태로 입력하였다. 그림2와 같은 reference를 사용하면 모터가 회전하는 속도가 일정하게 증가하고, input angle과 같은 각도에 도달하면 reference가 더 이상 증가하지 않아 힘을 유지한 상태로 각도를 유지할 수 있다. 코드 상에서 보았을 때 설정한 기울기와 현재 시간을 곱하여 angle값을 구하고 해당 angle값과 원하는 angle값을 비교하여 PID error가 되는 것이다. 이때 제어 계수는 $P=80$, $D=0.01$ 로 설정하였으며 실험 결과 흔들림이나 큰 오차없이 각도를 잘 유지하여 I제어는 하지않았다.

2.4 스프링 힘 조정

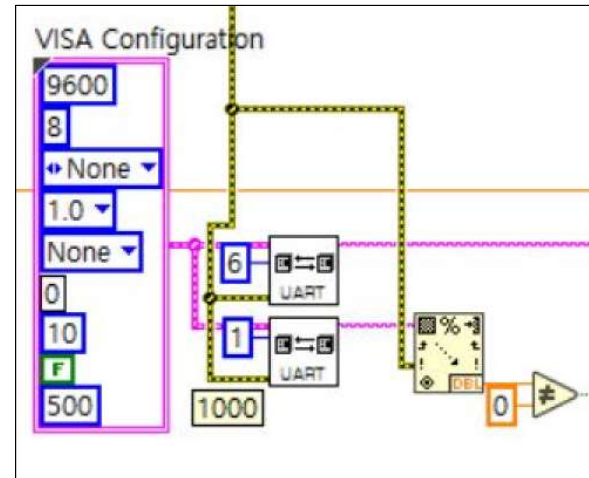


	2.5s	5.0s	7.5s
당길 때	4.0cm	8.0cm	12.0cm
밀 때	4.0cm	8.0cm	12.0cm

표1 실험 결과

스프링을 당길 때 사용하는 모터에는 엔코더가 부착되어 있지않기 때문에 labview의 timed loop로 모터를 구동하여 시간당 이동거리를 측정하여 사용하였다. 따라서 실험을 통해 스프링을 밀고 당길 때 실제로 스트로크가 얼마나 이동되는지 알아보았다. 그 결과 표1과 같이 2.5초 당 4cm씩 당겨지고, 7.5초는 12cm가 측정되어 모터의 힘이 스프링을 당기기에 충분하며, 선형적으로 움직이는 것을 알 수 있었다. 스프링 상수 $k = 1667\text{N/m}$ 에 따라 12cm를 이동하였을 때의 힘이 200N이 생기게 된다.

2.5 카메라 통신



불이 인식되었다면 로봇은 이동을 중단하여야한다. 이때 불 인식은 카메라가 하기 때문에 VISA함수를 이용하여 myRIO와 Jetson Nano를 UART 통신시켰다. 불을 인식했다는 신호를 보낸다. 위 이미지에서 1과 6은 몇 글자를 읽을 것인지를 나타내는데 인식되기 전에는 한글자만 읽어 0이 들어오면 불 인식안됨으로 주행을 계속하게 된다. 1이 들어오면 불 인식됨으로 주행을 중단하고 대포각 조정 단계로 넘어가는 간다. 이때부터 카메라에서 보내주는 6글자는 불의 중간 지점과 카메라 중간 지점(대포의 중간지점)의 수평, 수직 차를 계산하여 필요한 회전값을 전송받는다. 이 값만큼 회전을 시키면 로봇이 정확히 정면으로 불을 바라볼 수 있게 된다.

참고문헌

- [1] Kiam Heong Ang, G. Chong and Yun Li, "PID control system analysis, design, and technology," in IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pp.559–576, July 2005