목차

# 1. 전체 알고리즘 구상

# 2. 함수 설명

# 3. 콘솔창

# 4. 더 보충&고려해야 할 것

# 5. 다른 팀이랑 머지하게 되면 고려해야 할 것

# 전체 알고리즘 구상 (path\_planning part)

전략 stone위치 받아오기->

전략 robot 위치 생성하기->

다음 robot 좌표로 이동하기 위한 각도 거리 파라미터 생성->

실제 stone의 위치를 영상팀에서 받아오기->

--------------------------------------반복문--------------------------------------------

레이더 센서에서 로봇을 기준으로 상대적인 로봇의 위치를 계산하기->

영상팀에서 받아온 실제 stone의 위치로부터 바람직한 robot formation 좌표를 계산하기->

다음 stone의 경로 좌표 생성->

다음 stone 경로로부터 robot좌표 생성->

다음 robot 좌표로 이동하기 위한 각도 거리 파라미터 생성

# 함수 설명(main문 순서대로)

get\_trajectory\_plan  
속도가 존재하는 파일(1.txt)로부터 전략경로를 받아옴  
j\_x,y,v\_stone\_plan에 저장(처음부터 끝까지)  
이 파일의 인덱스는 변수w에 저장됨

w=3489

ideal\_robot\_trajectory  
j\_x\_stone\_plan에 저장된 스톤의 저장경로로부터 로봇의 전략경로를 생성함  
x,y\_robot\_plan\_1에 저장(처음부터 끝까지)

plan\_distance\_angle\_two

처음부터 끝까지 전략경로(robot\_ plan[j]->[j+1])로 이동하기 위한 거리(distance\_R1[j])와 각도(angle\_R1[j])를 생성함

get\_real\_stone\_traj

속도가 존재하는 파일(3.txt)로부터 현재 스톤 위치(x,y,v\_stone\_real)를 받아옴(처음부터 끝까지)  
이 파일의 인덱스는 변수real\_index에 저장됨

real\_index=3920

-----------------------------------------반복문-----------------------------------------------

velocity\_cal

v\_stone\_real\_y[index] = v\_stone\_real[index] \* tan(angle\_R1[index]);

y축 stone의 속도 계산<-다음 위치를 알기 위해서 필요함

find\_stone

용규함수//ang, rad 값 읽어옴

real\_formation

x\_stone\_real[ju]위치로부터 ang,rad 값을 이용해서 현재 로봇위치(x,y\_robot\_real\_1[ju]) 추정

ideal\_formation

x\_stone\_real[j]위치로부터 정해진 formation 위치를 x,y\_robot\_ideal\_1[j]에 저장

path\_prediction

if (index < 5)//robot출발후 얼마 안된 시간이므로 전략 경로 대로 갈 것이라고 생각함

{

x\_stone\_predict[index] = j\_x\_stone\_plan[index + 1];

y\_stone\_predict[index] = j\_y\_stone\_plan[index + 1];

return 0;

}

실제 로봇의 위치(x,y\_robot\_real\_1)[index])를 이용해서 기울기(m1,m2)를 추정하고 다음좌표 추정함

y\_stone\_predict[index] = y\_stone\_real[index - 1] + v\_stone\_real\_y[index - 1] \* freq;

x\_stone\_predict[index] = w1\*x\_stone\_real[index-1] + w2 \* v\_stone\_real\_y[index - 1] \* freq / m3;

// stone\_real 값으로부터 stone \_predict를 추정

**x\_stone\_predict[index] =0.7\* x\_stone\_predict[index]+ 0.2\*j\_x\_stone\_plan[index+1]+ 0.1\*x\_stone\_real[index];**

**y\_stone\_predict[index]=0.7\* y\_stone\_predict[index] + 0.2\*j\_y\_stone\_plan[index+1] + 0.1\*y\_stone\_real[index];**//ideal+ real+predict 혼합해서 robot predict 추정

//stone\_predict값, stone\_plan의 다음 값(전략경로를 반영하기 위해), stone\_real값(현재 stone의 위치를 고려하기 위해 )에 가중치를 반영해서 stone\_predict을 추정한다.

plan\_distance\_angle

현재 로봇의 위치(x,y\_robot\_real[j])에서 로봇의 다음 위치(x,y\_predict[j+1])로 이동하기 위한 거리(distance\_R1[j])과 각도(angle\_R1[j])생성

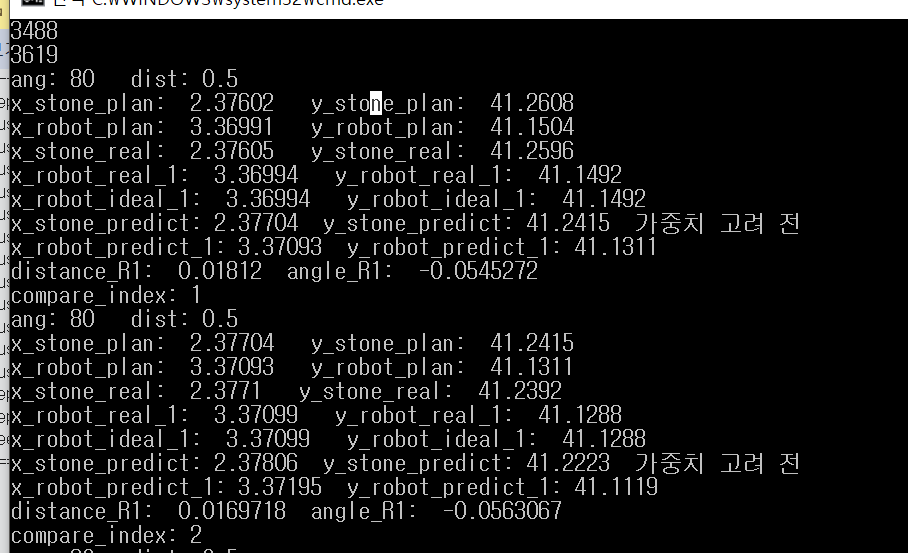
~~robot\_trajectory->ideal\_formation 함수로 바꿈!!~~

~~x,y\_stone\_real[j]로부터 로봇의 formation 위치(x,y\_robot\_real[j])를 생성함~~

# 콘솔창

1. 용규에게 받아오는 값이 이상적이고 1번텍스트가plan, 3번텍스트가 real일 때

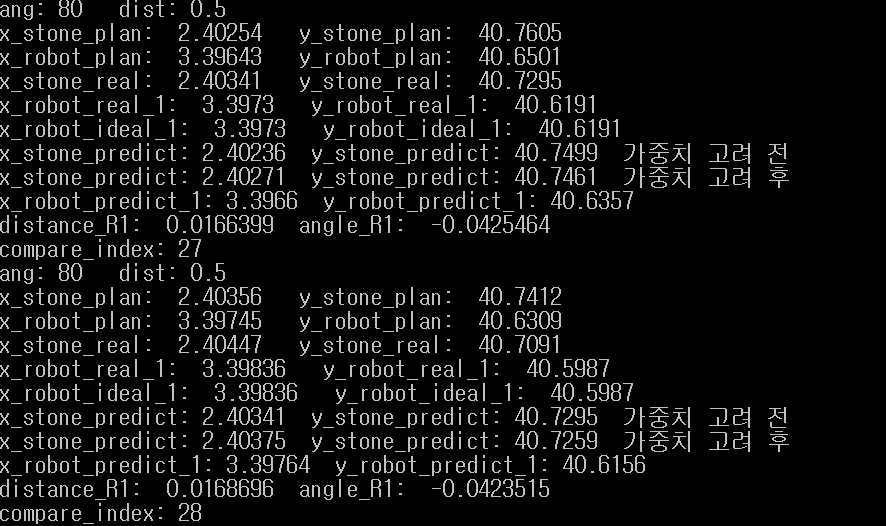
ang:80, dist:0.5m

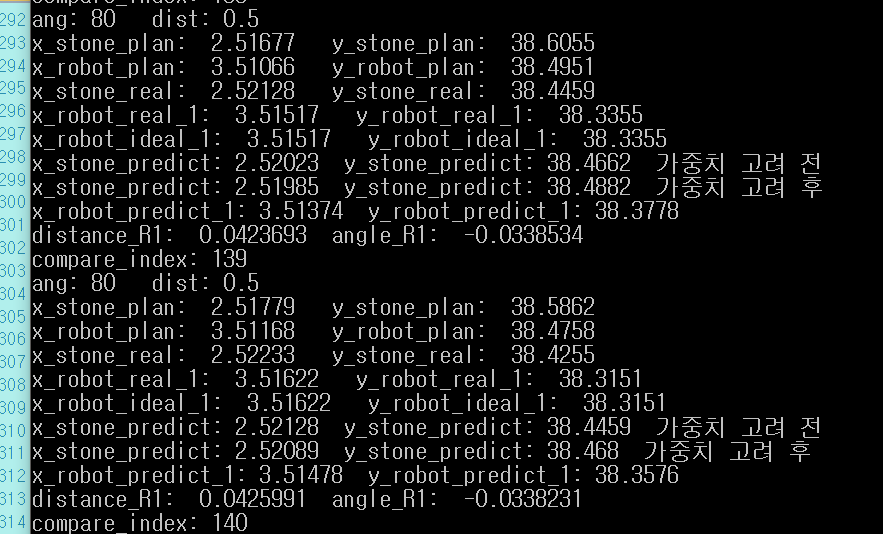


1번텍스트에서의 index(plan), 3번텍스트에서의 index(real),

반복문 들어온 이후 저장된 배열값을 출력한다.

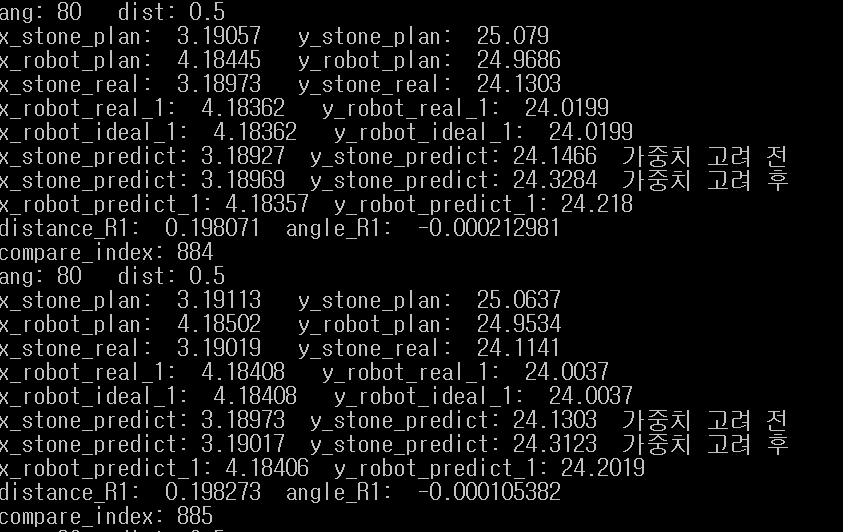
인덱스가 5보다 작을때는 predict값에 real 값을 대입한 후 계산해서 출력한다.



인덱스가 5보다 클 때는 이전 real 값들을 고려해서 predict값을 찾는다.

index가 증가할수록 predict값의 오차가 증가하는 것을 확인할 수 있다.

x는 0.002m정도 y는 0.1m차이

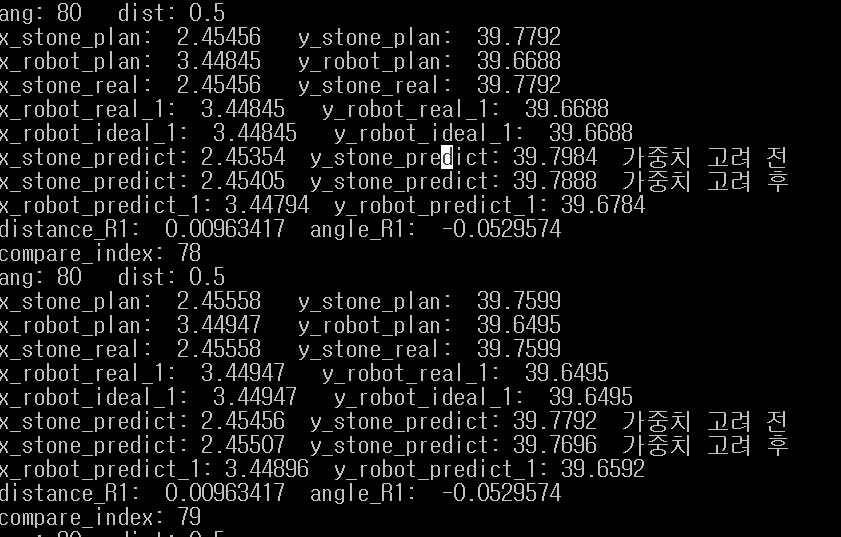


x는 0.01m정도 y는 1m차이

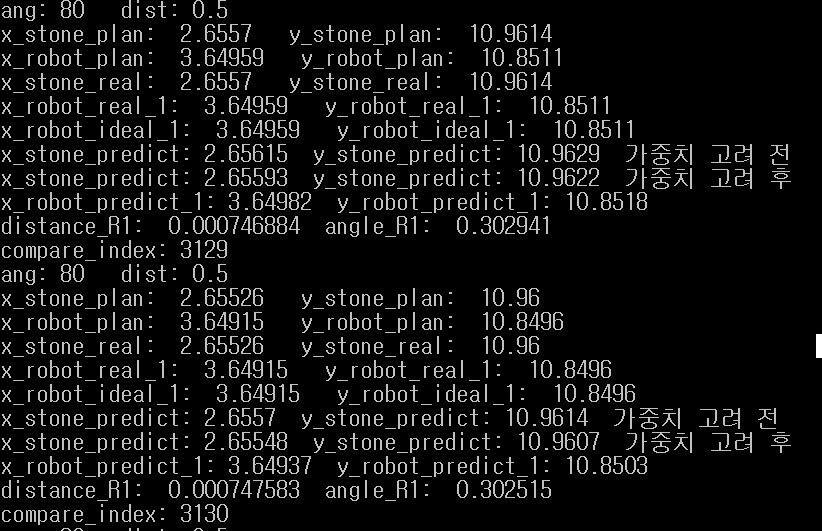
index가 끝값과 비슷해질수록 2m 가까이 차이가 나게 된다.

1. 용규에게 받아오는 값이 이상적이고 1번텍스트가 plan, 1번텍스트가 real일 때

ang:80, dist:0.5m



x는 0.01m정도 y는 0.03m차이



인덱스가 커져도 별로 차이가 나지 않는다.

# 4. 더 보충&고려해야 할 것

1. velocity\_cal함수  
   tan함수를 쓰는 것이 적당한가?  
   용규꺼를 이상적 값으로 가정: v\_stone\_real\_y가 18정도 나옴  
   용규보편: v\_stone\_real\_y가 -7정도 나옴
2. 만약 경로 복귀했을 때는?(경로와 오차한계를 얼마로 지정하기)  
   아직까지는 반복문 무조건 돈다고 생각했음  
   test\_Pre.j\_x\_stone\_plan[compare\_index] != test\_Pre.x\_stone\_real[compare\_index] || test\_Pre.j\_y\_stone\_plan[compare\_index] != test\_Pre.y\_stone\_real[compare\_index]
3. 난희 idea:70퍼센트 지점까지(w개수의 70퍼센트)는 real과 plan의 오차 줄이기하고 나머지 구간은 경로복귀에 집중하기
4. 현수T: 속도, 앵글을 직접 조정하기  
   초기에 포메이션이 형성이 안되었을땐 x, y 좌표 신경쓰지 말고 스톤에 관해 거리랑 각도만 보고 거리 스톤이랑 거리 각도 계속 비교 하며 거리 멀면 속도 줄여서 일정 거리 안으로 들어오면 그때 정확하게 맞추고 각도도 마찬가지로.  
   빠르게 맞추는게 좋으니 거리각도 같이 맞추는게 좋긴한데 그건 시뮬레이션 해보고 결정하자 거리가 20cm이상 차이나면 현재 속도의 10%정도 줄이거나 늘리고 20cm이하로 오차 범위가 들어오면 세세하게 맞추기
5. 속도를 넘겨주는 것은 distance, angle, hz 고려해서 식을 만들어서 전송할 것\*\*  
   맨끝에..속도가 줄지를 않고 오히려 증가해버림..  
   w가 거의 끝일때는 속도를 강제로 낮추기….

# 다른 팀이랑 머지하게 되면 고려해야 할 것

1. real stone 좌표 받아오는거 패킷으로 만들기
2. thread

-주행

-formation

-레이더

-sweeping