Bug Algorithms

Jasen Carroll

Bug1

%% USAR Bug1

clear

clc

start=[0;0];

qgoal = [-8;-8];

%% wall-following numbers

follow = 0; %bug knows when it is/is not following an obstacle

perim = [0 0]; %first is total perimiter; seconcd is perimiter since last closest point to the goal

qhit = [0 0]'; %so the bug knows when it has circled the obstacle

qleave = [0 0]';% closest point to goal around the obstacle

hug = 0.5; %stay about 1m away from wall

rightleft = 1; %1=hang a left at an obstacle; -1=turn right

closest = 0; %stores closest distance to goal around the perimeter of an obstacle

pts = zeros(2,0);

m = .75;

turnConst = 80; %movement constants

moveConst = 40;

%% initialize robot

rob = initializeRobot('rob','P2AT',[start' 1.8],[0 0 0]);

pause(2);

%% start the loop

t = -linspace(-pi,pi,180)';

INS = getINSReadings(rob);

pos = INS.Position(1:2);

orient = INS.Orientation(3);

lastpos = pos;

v = [];

while(norm(pos - qgoal) > 1)

lrf = getLaserSensorReadings(rob);

data = lrf.Scans;

if (follow==0)

if data(90) < hug % if the sensor reading ahead is less than 0.75

qhit = pos;

qleave = pos;

follow = 1; % assign 1 to follow

flag = 0;

end

v = (qgoal-pos)/norm(qgoal-pos); % direction and speed toward goal

else

[C I] = min(data);

theta = t(I) + orient+pi/2;

ostart = pos;

if ((C) < hug)

% add in correction factor?

v = [cos(theta);sin(theta)];

elseif ((C) > hug)

v = [cos(theta);sin(theta)];

end

if follow == 1

q = pos-qgoal;

if (norm(q) < norm(qgoal - qleave))

qleave = pos;

end

if (norm(pos-qhit) < 0.5 && flag)

follow=2;

elseif (norm(pos-qhit) > 0.75)

flag = 1;

end

elseif follow == 2 %follow == 2, we know the closest place to the goal and we're moving there

if (norm(pos-qleave) < .5);

follow = 0;

end

end

end

% "v" should now be assigned. The robot will attempt to move in that

% direction.a

vH = [cos(orient);sin(orient)];

omega = turnConst\*v'\*[0 -1;1 0]\*vH;

V = moveConst\*v'\*vH;

sendDriveCommand(rob,V+[omega,-omega]);

pause(.05);

lastpos = pos;

INS = getINSReadings(rob);

pos = INS.Position(1:2);

orient = INS.Orientation(3);

end

shutdownRobot(rob);

Bug2

%% USAR Bug2

clear

clc

start=[0;0];

qgoal = [-8;-8];

%% wall-following numbers

follow = 0; %bug knows when it is/is not following an obstacle

perim = [0 0]; %first is total perimiter; seconcd is perimiter since last closest point to the goal

qhit = [0 0]'; %so the bug knows when it has circled the obstacle

qleave = [0 0]';% closest point to goal around the obstacle

hug = 0.5; %stay about 1m away from wall

rightleft = 1; %1=hang a left at an obstacle; -1=turn right

closest = 0; %stores closest distance to goal around the perimeter of an obstacle

pts = zeros(2,0);

m = .75;

turnConst = 80; %movement constants

moveConst = 40;

%% initialize robot

rob = initializeRobot('rob','P2AT',[start' 1.8],[0 0 0]);

pause(2);

%% start the loop

t = -linspace(-pi,pi,180)';

INS = getINSReadings(rob);

pos = INS.Position(1:2);

orient = INS.Orientation(3);

lastpos = pos;

v = [];

while(norm(pos - qgoal) > 1)

lrf = getLaserSensorReadings(rob);

data = lrf.Scans;

if (follow==0)

if data(90) < hug % if the sensor reading ahead is less than 0.75

qhit = pos;

qleave = pos;

follow = 1; % assign 1 to follow

flag = 0;

end

v = (qgoal-pos)/norm(qgoal-pos); % direction and speed toward goal

else

[C I] = min(data);

theta = t(I) + orient+pi/2;

ostart = pos;

if ((C) < hug)

% add in correction factor?

theta = theta - 20\*(pi/180);

v = [cos(theta);sin(theta)];

elseif ((C) > hug)

theta = theta + 20\*(pi/180);

v = [cos(theta);sin(theta)];

end

if follow == 1

q = pos-qgoal;

if (norm(q) < norm(qgoal - qleave))

qleave = pos;

end

if (qleave ~= pos)

follow=0;

end

end

end

% "v" should now be assigned. The robot will attempt to move in that

% direction.a

vH = [cos(orient);sin(orient)];

omega = turnConst\*v'\*[0 -1;1 0]\*vH;

V = moveConst\*v'\*vH;

sendDriveCommand(rob,V+[omega,-omega]);

pause(.05);

lastpos = pos;

INS = getINSReadings(rob);

pos = INS.Position(1:2);

orient = INS.Orientation(3);

end

shutdownRobot(rob);

TBUG

%% USAR TBUG

clear

clc

start=[0;0];

qgoal = [-8;-8];

%% wall-following numbers

follow = 0; %bug knows when it is/is not following an obstacle

perim = [0 0]; %first is total perimiter; seconcd is perimiter since last closest point to the goal

qhit = [0 0]'; %so the bug knows when it has circled the obstacle

qleave = [0 0]';% closest point to goal around the obstacle

hug = 1; %stay about 1m away from wall

rightleft = 1; %1=hang a left at an obstacle; -1=turn right

closest = 0; %stores closest distance to goal around the perimeter of an obstacle

pts = zeros(2,0);

m = .75;

turnConst = 80; %movement constants

moveConst = 40;

%% initialize robot

rob = initializeRobot('rob','P2AT',[start' 1.8],[0 0 0]);

pause(2);

%% start the loop

t = -linspace(-pi,pi,180)';

INS = getINSReadings(rob);

pos = INS.Position(1:2);

orient = INS.Orientation(3);

lastpos = pos;

v = [];

while(norm(pos - qgoal) > 1)

lrf = getLaserSensorReadings(rob);

data = lrf.Scans;

if (follow==0)

if (data(90) < hug) % if the sensor reading ahead is less than hug

follow = 1; % assign 1 to follow

end

v = (qgoal-pos)/norm(qgoal-pos); % direction and speed toward goal

else

[C I] = min(data);

theta = t(I) + orient+pi/2;

if ((C) < hug)

%add in correction factor?

v = [cos(theta);sin(theta)];

elseif ((C) > hug)

v = [cos(theta);sin(theta)];

end

x = qgoal(1)-pos(1);

y = qgoal(2)-pos(2);

phi = atan(y/x);

alpha = phi - orient;

beta = 20\*pi/180;

index = t >= alpha-beta & t <= alpha+beta;

d = data(index);

if (all(d > 2.5) && data(90) >= 2.5)

follow = 0;

end

end

% "v" should now be assigned. The robot will attempt to move in that

% direction.a

vH = [cos(orient);sin(orient)];

omega = turnConst\*v'\*[0 -1;1 0]\*vH;

V = moveConst\*v'\*vH;

sendDriveCommand(rob,V+[omega,-omega]);

pause(.05);

lastpos = pos;

INS = getINSReadings(rob);

pos = INS.Position(1:2);

orient = INS.Orientation(3);

end

shutdownRobot(rob);