```
syms x;
%membership sets for distance
veryClose(x) = piecewise(x<10, 1-x/10, x>10, 0);
somewhatClose(x) = piecewise(x<20, x/20, (20<x)& (x<30), 3-x/10, x>30, 0);
notFar(x) = piecewise(x<20, 0, (20<x)& (x<30), x/10-2, (30<x) & (x<40), 4-
x/10, x>40, 0);
somewhatFar(x) = piecewise(x<30, 0, (30<x)& (x<40), x/10-3, (40<x) & (x<50),
 5-x/10, x>50, 0);
veryFar(x) = piecewise(x<40,0,(40<x)&(x<50),x/10-4,x>50,1);
a=0;
b=pi/2;
c=pi;
d=3*pi/2;
e = 2*pi;
%membership sets for angles
slightLeft(x) = piecewise(x<b, 1-x/b, x>=b,0);
strongLeft(x) = piecewise(x<b,x/b,(b<=x)&(x<c),2-x/b, x>=c,0);
uturn(x) = piecewise(x<b,0,(x>=b)&(x<c),x/b-1,(x>=c)&(x<d),3-x/b,x>=d,0);
strongRight(x) = piecewise(x<c,0,(x>=c)&(x<d),x/b-2,(x>=d)&(x<e),4-x/b);
slightRight(x) = piecewise(x<d,0,(x>=d)&(x<e),x/b-3);
goalx = input('Enter goal x coordinate');
goaly = input('Enter goal y coordinate');
currentx = 0;
currenty = 0;
distance = sqrt((goalx - currentx)^2+(goaly-currenty)^2);
%robot starts facing straight down the x axis while at the origin
currentTheta = 0;
time = 0;
%loop to be repeated while not essentially at destination
while distance > .1
    time = time+1;
    %fuzzifier for distance
    fuzzyDvc = double(veryClose(distance));
    fuzzyDsc = double(somewhatClose(distance));
    fuzzyDnf = double(notFar(distance));
    fuzzyDsf = double(somewhatFar(distance));
    fuzzyDvf = double(veryFar(distance));
    %defuzzifier for distance
    speed = (fuzzyDsc*10+fuzzyDnf*20+fuzzyDsf*30+fuzzyDvf*40)/(fuzzyDvc
+fuzzyDsc+fuzzyDnf+fuzzyDsf+fuzzyDvf);
    goalTheta = mod((atan((goaly-currenty)/(goalx-currentx))),(2*pi));
    %fuzzifier for angle robot needs to travel
    fuzzyTslg = double(slightLeft(goalTheta));
    fuzzyTstlg = double(strongLeft(goalTheta));
    fuzzyTutg = double(uturn(goalTheta));
    fuzzyTstrg = double(strongRight(goalTheta));
    fuzzyTsrg = double(slightRight(goalTheta));
    %fuzzifier for current angle robot is facing
    fuzzyTslc = double(slightLeft(currentTheta));
    fuzzyTstlc = double(strongLeft(currentTheta));
    fuzzyTutc = double(uturn(currentTheta));
    fuzzyTstrc = double(strongRight(currentTheta));
```

```
fuzzyTsrc = double(slightRight(currentTheta));
thetaAdjust = 0;
%determines which fuzzy sets are dominant for the two fuzzy angles.
ctmemvalues = [fuzzyTslc,fuzzyTstlc,fuzzyTutc,fuzzyTstrc,fuzzyTsrc];
[blank,fuzzdomc] = max(ctmemvalues);
gtmemvalues = [fuzzyTslg,fuzzyTstlg,fuzzyTutg,fuzzyTstrg,fuzzyTsrg];
[blank2,fuzzdomg] = max(gtmemvalues);
%FIE for angle to travel vs current angle, approximates turn needed to
%the nearest pi/2 radians
if (fuzzdomc==1) || (fuzzdomc == 5)
    if (fuzzdomg ==1) || (fuzzdomg == 5)
        thetaAdjust = 0;
    end
    if fuzzdomg == 2
        thetaAdjust = pi/2;
    end
    if fuzzdomg == 3
        thetaAdjust = pi;
    end
    if fuzzdomg == 4
        thetaAdjust = -pi/2;
    end
end
if fuzzdomc==2
    if (fuzzdomg ==1) || (fuzzdomg == 5)
        thetaAdjust = -pi/2;
    end
    if fuzzdomg == 2
        thetaAdjust = 0;
    end
    if fuzzdomg == 3
        thetaAdjust = pi/2;
    end
    if fuzzdomg == 4
        thetaAdjust = pi;
    end
end
if fuzzdomc==3
    if (fuzzdomg ==1) || (fuzzdomg == 5)
        thetaAdjust = pi;
    end
    if fuzzdomg == 2
        thetaAdjust = -pi/2;
    end
    if fuzzdomg == 3
        thetaAdjust = 0;
    end
    if fuzzdomg == 4
        thetaAdjust = pi/2;
    end
end
if fuzzdomc==4
    if (fuzzdomg ==1) || (fuzzdomg == 5)
        thetaAdjust = pi/2;
        end
        if fuzzdomg == 2
            thetaAdjust = pi;
        end
        if fuzzdomg == 3
            thetaAdjust = -pi/2;
        end
        if fuzzdomg == 4
            thetaAdjust = 0;
        end
    end
    %adjust theta
    currentTheta = mod((currentTheta+thetaAdjust),(2*pi));
    %adjust location assuming speed is in units of m/s
    currentx = currentx + speed*.5*cos(currentTheta);
    currenty = currenty + speed*.5*sin(currentTheta);
    %see how far away from goal we now are
    distance = sqrt((goalx - currentx)^2+(goaly-currenty)^2);
    disp('time: ' + time);
    disp('membership functions');
    disp('Very Close: ' + fuzzyDvc);
    disp('Somewhat Close: ' + fuzzyDsc);
    disp('Not Far: ' + fuzzyDnf);
    disp('Somewhat Far: ' + fuzzyDsf);
    disp('Very Far: ' + fuzzyDvf);
    disp('Slight Left: ' + fuzzyTslg)
    disp('Slight Right: '+ fuzzyTsrg)
    disp('Strong Left: ' + fuzzyTstlg)
    disp('Strong Right: ' + fuzzyTstrg)
    disp('U-Turn: ' + fuzzyTut)
    disp('New Angular Velocity' + currentTheta)
    disp('New Forward Force' + speed)
end
disp('job done')
```

```
Error using input
Cannot call INPUT from EVALC.

Error in fuzzyprojectr (line 19)
goalx = input('Enter goal x coordinate');
```

Published with MATLAB® R2021b