ME 586: **Biology-inspired robotics**

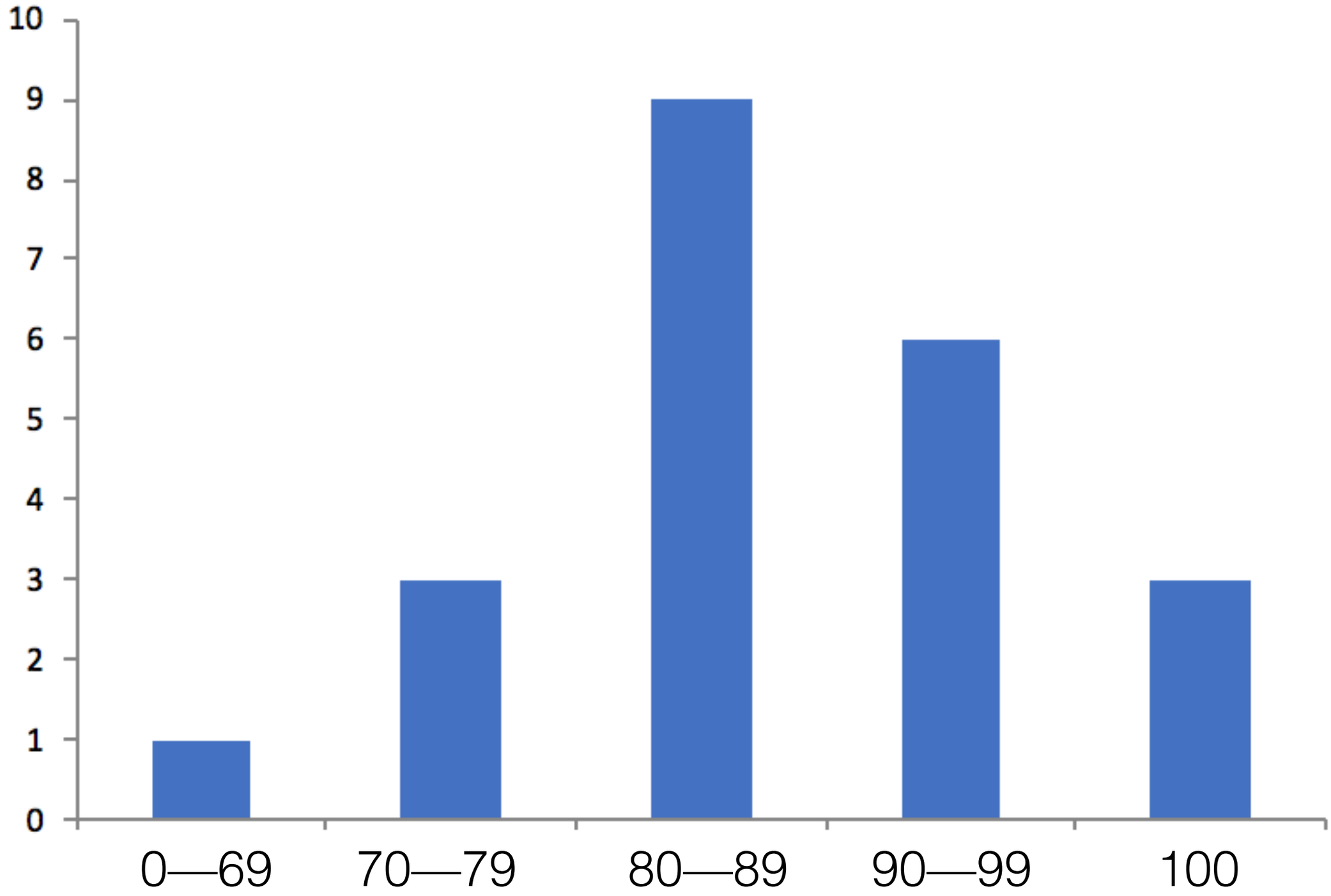
Prof. Sawyer B. Fuller

Goals:

- return quiz
- cool problem set results
- watch robot videos

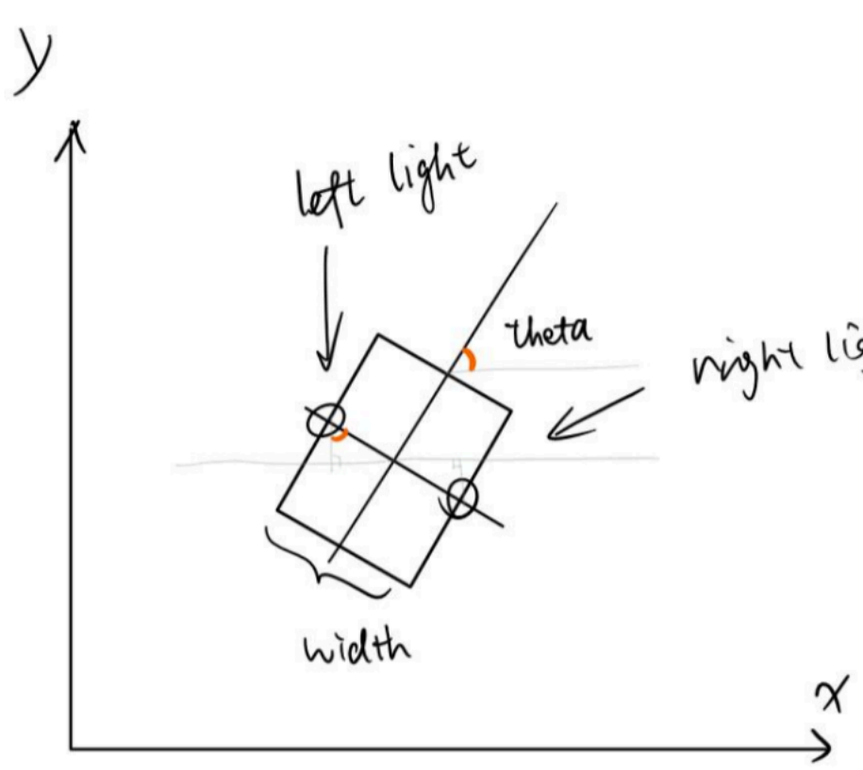
UNIVERSITY of WASHINGTON

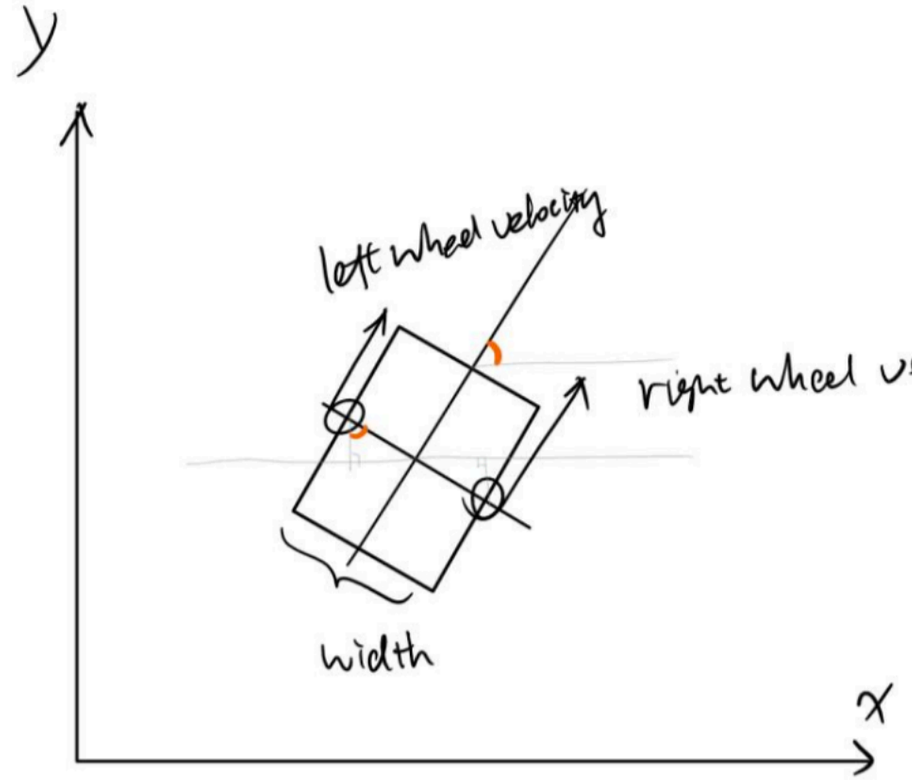
quiz histogram
mean: 85.1

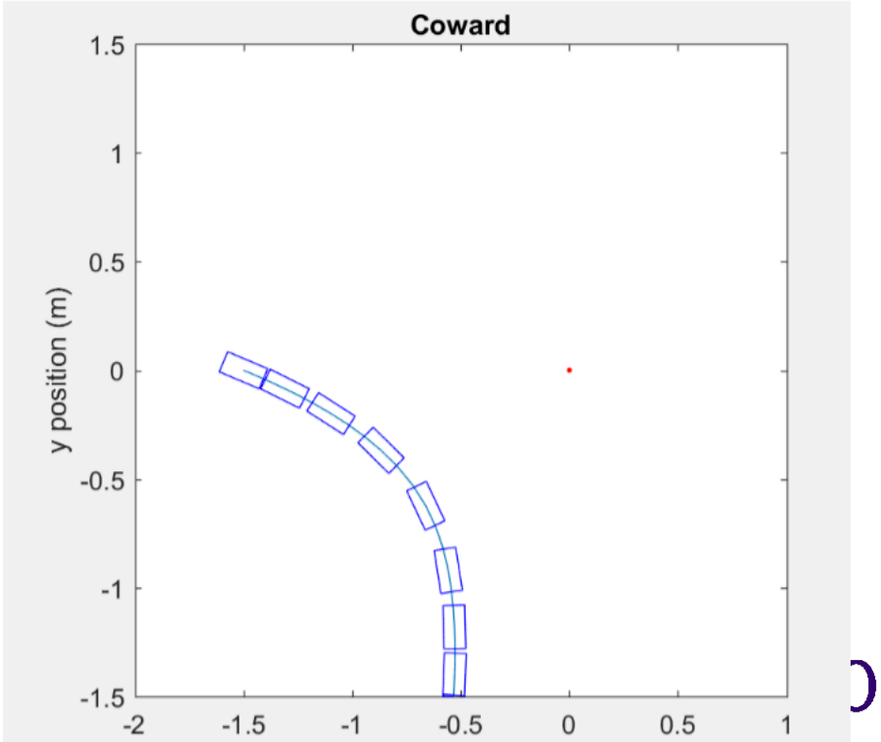


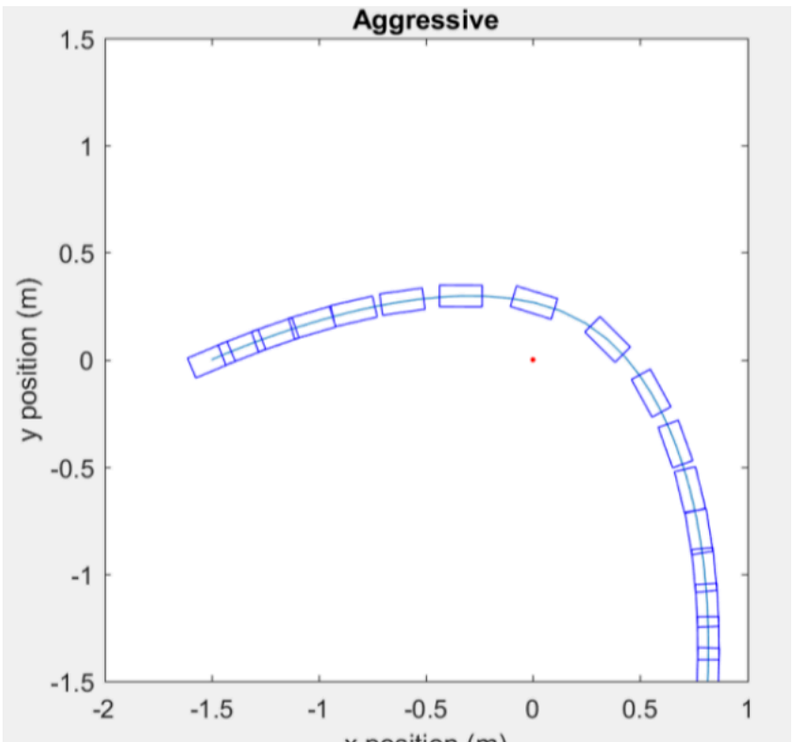
problem set
mean: 3.8/4

from X. Yang


y
left light
theta
right light
width
x


y
left wheel velocity
right wheel velocity
width
x

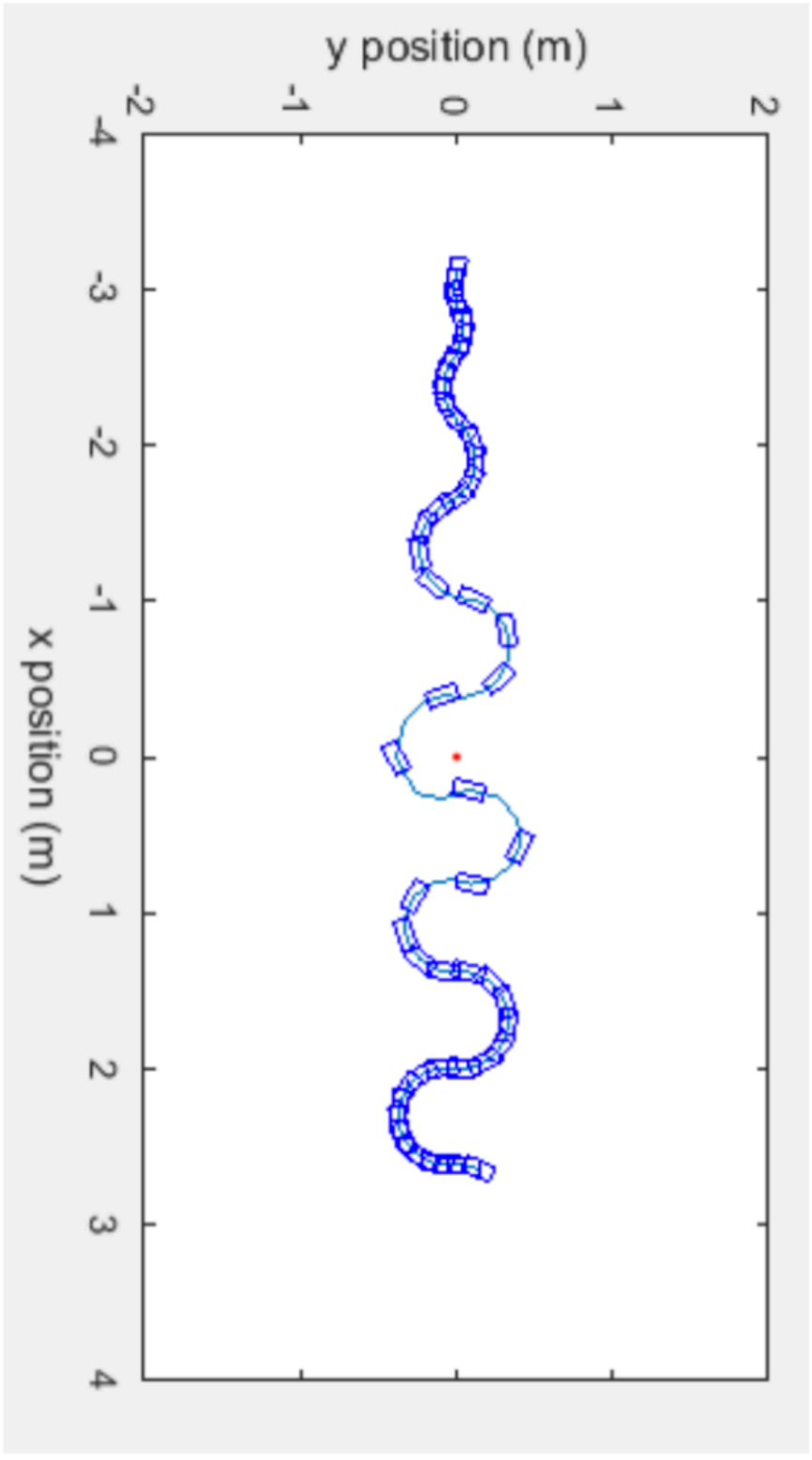

Coward
y position (m)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1


Aggressive
y position (m)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1

interesting problem set solutions

UNIVERSITY of WASHINGTON

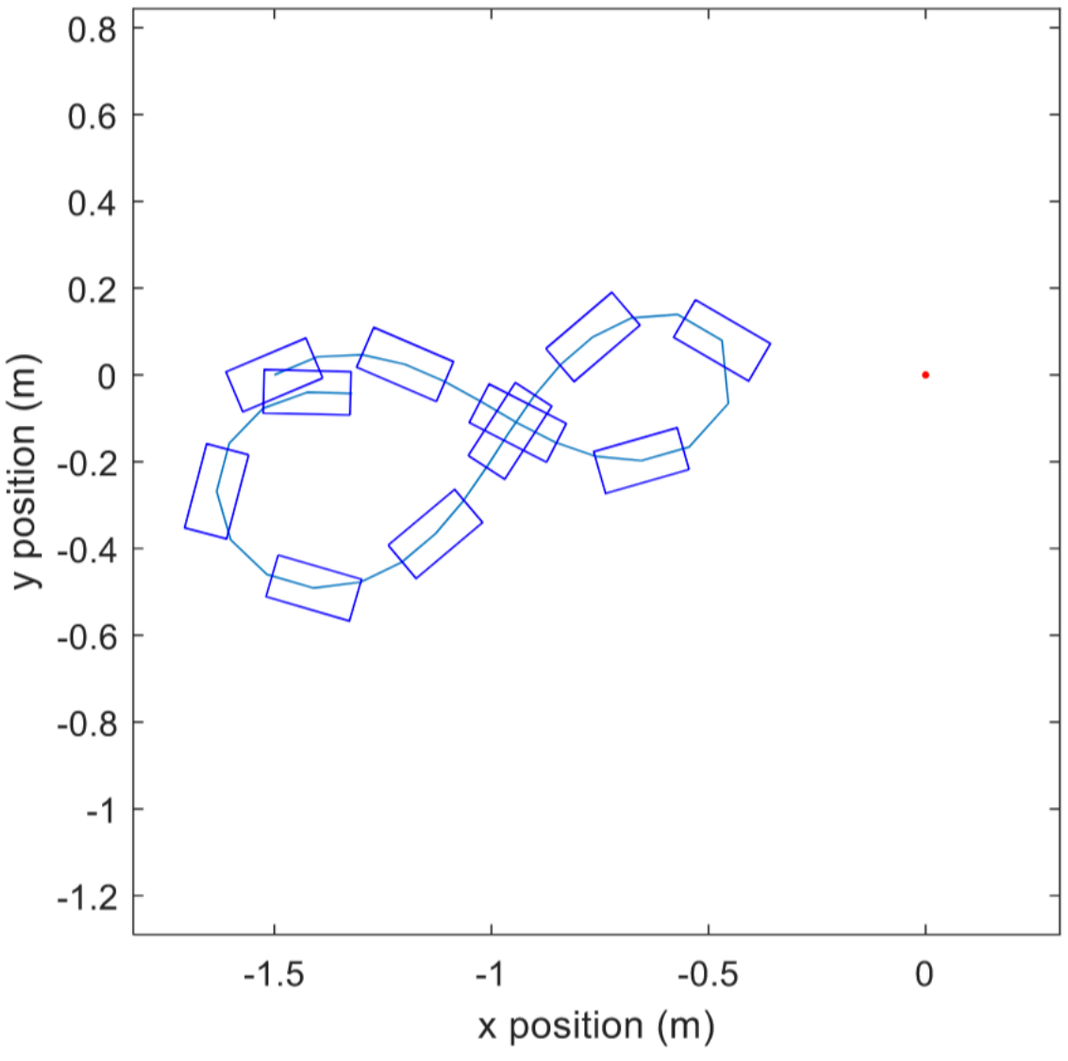
Jing Lu




```
function u = light_response(distances,p) % Interesting Behavior
    distance_to_left_sensor = distances(1);
    distance_to_right_sensor = distances(2);
    LWV = 0.7 / (distance_to_left_sensor );
    RWV = 0.7 / (distance_to_right_sensor );
    %Gaussian
    I = [LWV,RWV];
    v_initial = 0.5; %initial velocity, kick start
    miu = 0.7;
    sigma = 0.3;
    u = exp(-(I  - miu).^2 / (sigma^2)) + v_initial;
    %Other
    if distance_to_left_sensor > distance_to_right_sensor
       LWV = 1.38 * RWV;
    elseif distance_to_left_sensor < distance_to_right_sensor
       RWV = 1.38 * LWV;
    elseif distance_to_left_sensor > distance_to_right_sensor
       LWV = 0;
       RWV = 0;
    end
    u = [LWV; RWV];
end
```

(sensors positioned at front corners of robot)

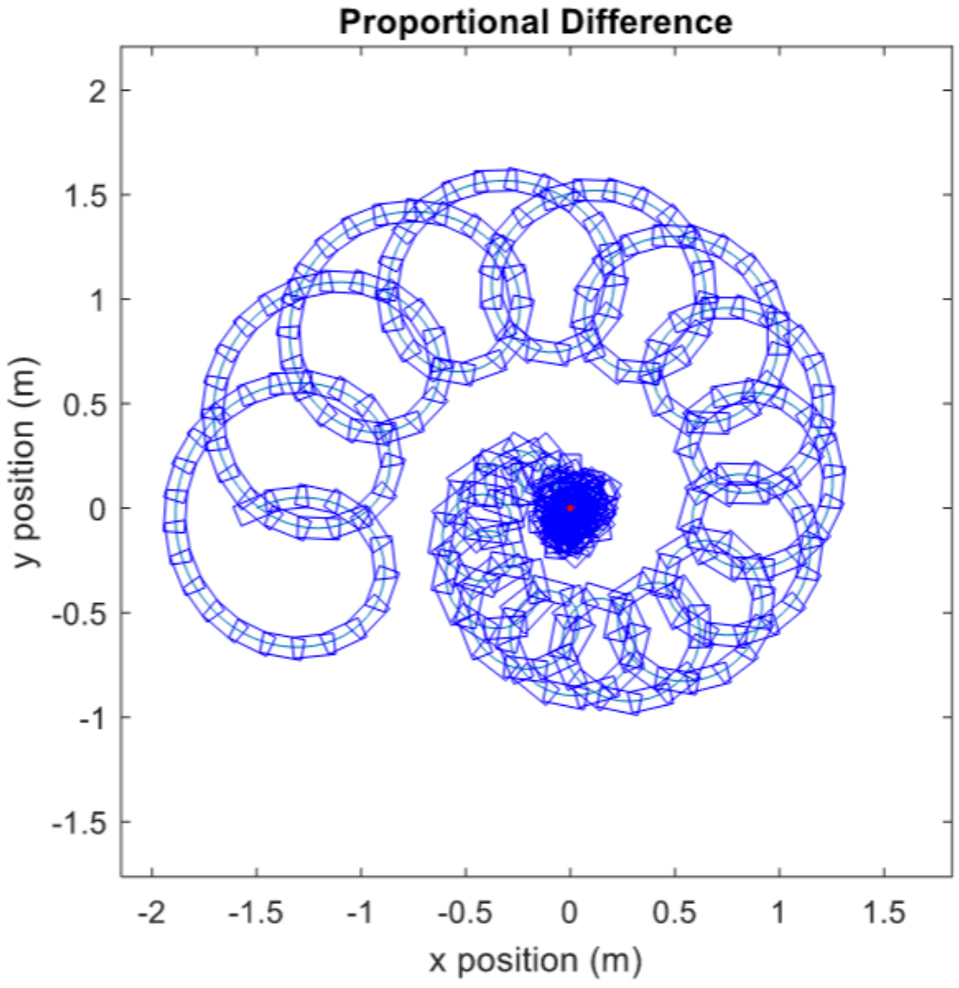
Saunak Panda:
asymmetric response function




```
function u = light_response(distances, p)
    distance_to_left_sensor = distances(1);
    distance_to_right_sensor = distances(2);

    % fill this in ...

     left_wheel_velocity =  0.5*distance_to_left_sensor
     right_wheel_velocity = 0.5/distance_to_right_sensor

    u = [left_wheel_velocity; right_wheel_velocity];
end
```

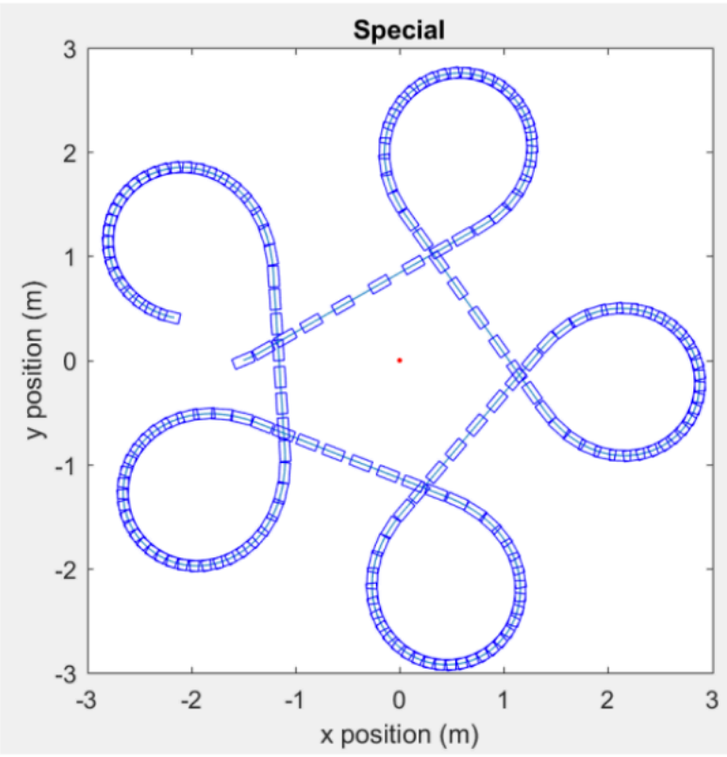
(sensors positioned at front corners of robot)

Haonan Peng:




```
I=[1/distances(1),1/distances(2)];
    norm_prop=0.5;
    v_diff = 0.1;
u = (1-norm_prop*norm(I))*0.5 + norm_prop*norm(I)*[0.5+v_diff,0.5-v_diff];
```

Xingjiang Yang: clover




```
function u = light_response(distances, p, q)
    distance_to_left_sensor = distances(1);
    distance_to_right_sensor = distances(2);
    distance = sqrt(distances(1)^2 + distances(2)^2);
    % fill this in ...
    factor=1;
    if distance >= 2
        a = 1;
        left = distance_to_right_sensor;
        right = distance_to_left_sensor*0.895;
    elseif distance <=0.1
        left = distance_to_left_sensor;
        right = distance_to_right_sensor;
    else
        left = (distance_to_left_sensor+distance_to_right_sensor)/2;
        right = left;
    end
    left_wheel_velocity =  factor/left; %coward
    right_wheel_velocity = factor/right;
    u = [left_wheel_velocity; right_wheel_velocity];
end
```