// Nawaf Abdullah

// CMPSC 201 SU 2016

// Date 8-3-16

// HW #8B

// MATLAB Robot Simulation

**1.0 Problem Statement**

* Part One
* Part Two

**2.0 Map Used**

* Screenshot of the Maze
* Maze script

**3.0 MATLAB Functions**

* Without Bump Sensor
* Using Bump Sensor

**4.0 Test Cases**

* Without Bump Sensor
* Using Bump Sensor

**1.0 Problem Statement**

* Part One

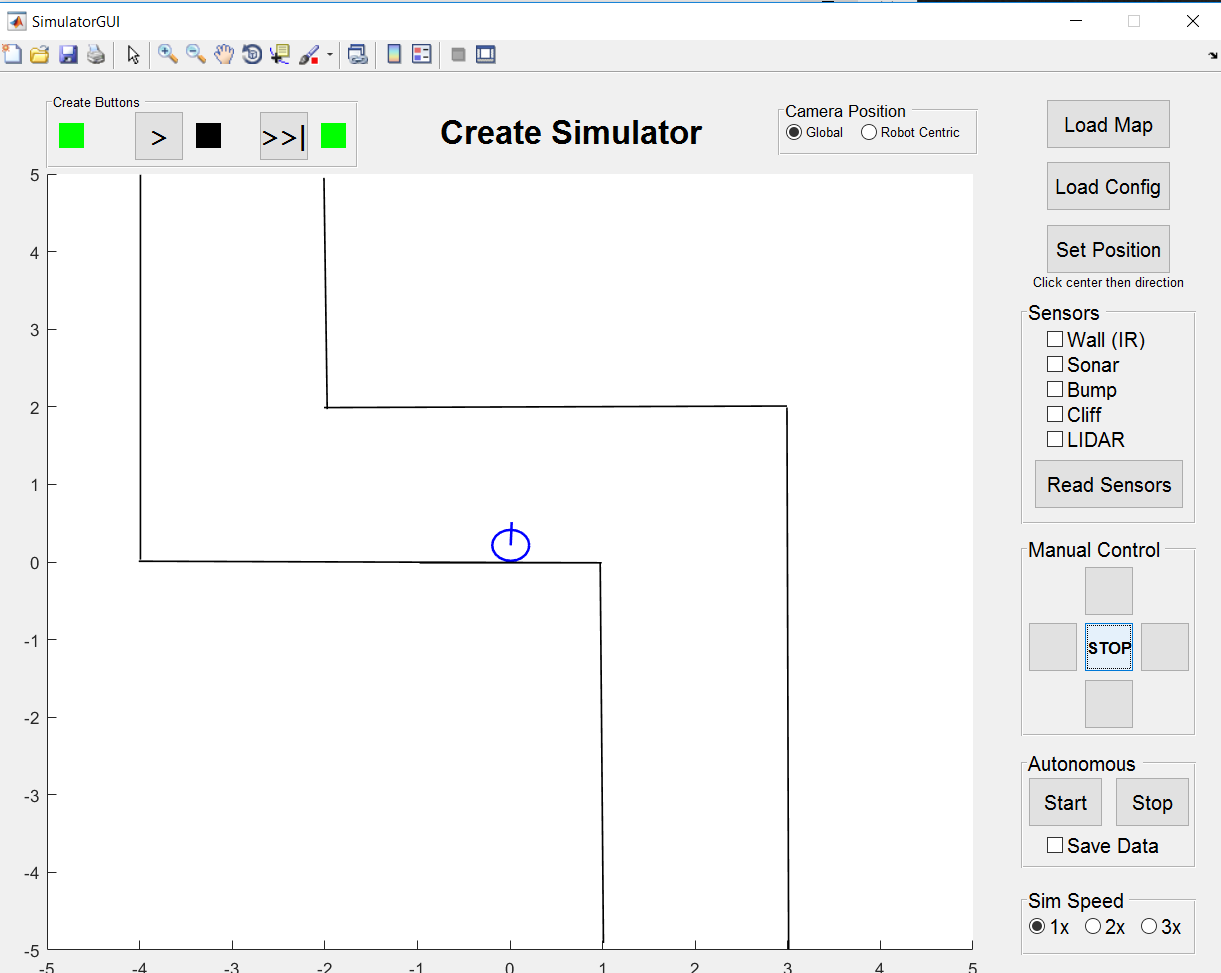
Write a script in MATLAB to move the robot simulation through a simple maze without using any sensors.

* Part Two

Write a script in MATLAB to move the robot simulation through a simple maze using the bump sensor.

**2.0 Map Used**

* Screenshot of the Maze



* Maze script

% File containing map information

% Formatting:

% wall x1 y1 x2 y2

% Order does not matter between the points

% line x1 y1 x2 y2

% beacon x y [r g b] ID\_tag

% [r g b] is the red-green-blue color vector

% virtwall x y theta

% Virtual walls emit from a location, not like real walls

% theta is the angle relative to the positive x-axis

wall 1.009 -4.910 0.974 -0.010

wall 0.991 -0.010 -0.974 -0.010

wall 0.957 -0.010 -4.009 0.010

wall -3.991 0.030 -3.991 4.990

wall 3.009 -4.990 2.991 1.990

wall 2.991 2.010 -2.009 1.990

wall -1.974 1.970 -2.009 4.950

**3.0 MATLAB Functions**

* Without Bump Sensor

function squarePath(serPort)

% Robot moves along a path that forms a square

% serPort is the serial port number (for controlling the actual robot).

%%%% DO NOT MODIFY CODE ABOVE %%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

travelDist(serPort, .3, 6) %% Travel at 0.3m/s for 6 meters.

turnAngle(serPort, .2, 90) %% Turn 90 degrees counterclockwise.

travelDist(serPort, .3, 5) %% Travel at 0.3m/s for 5 meters.

turnAngle(serPort, .2, -90) %% Turn 90 degrees clockwise.

travelDist(serPort, .3, 5) %% Travel at 0.3m/s for 5 meters.

* Using Bump Sensor

function driveForwardUntilWall(serPort)

% Robot drives forward until it bumps a wall.

% serPort is the serial port number (for controlling the actual robot).

[BumpRight,BumpLeft,WheDropRight,WheDropLeft,WheDropCaster,BumpFront] = ... %% Move forward if there are no obstacles.

BumpsWheelDropsSensorsRoomba(serPort);

while ~BumpRight && ~BumpLeft && ~BumpFront

SetDriveWheelsCreate(serPort, .5,.5)

pause(.1)

[BumpRight,BumpLeft,WheDropRight,WheDropLeft,WheDropCaster,BumpFront] = ...

BumpsWheelDropsSensorsRoomba(serPort);

end

StopCreate(serPort)

Signal()

turnAngle(serPort, .3, 90) %% Turn 90 Degrees counterclockwise after bumbing into a wall.

[BumpRight,BumpLeft,WheDropRight,WheDropLeft,WheDropCaster,BumpFront] = ... %% Move forward if there are no obstacles.

BumpsWheelDropsSensorsRoomba(serPort);

while ~BumpRight && ~BumpLeft && ~BumpFront

SetDriveWheelsCreate(serPort, .5,.5)

pause(.1)

[BumpRight,BumpLeft,WheDropRight,WheDropLeft,WheDropCaster,BumpFront] = ...

BumpsWheelDropsSensorsRoomba(serPort);

end

StopCreate(serPort)

Signal()

turnAngle(serPort, .3, -90) %% Turn 90 Degrees clockwise after bumbing into a wall.

[BumpRight,BumpLeft,WheDropRight,WheDropLeft,WheDropCaster,BumpFront] = ... %% Move forward if there are no obstacles.

BumpsWheelDropsSensorsRoomba(serPort);

while ~BumpRight && ~BumpLeft && ~BumpFront

SetDriveWheelsCreate(serPort, .5,.5)

pause(.1)

[BumpRight,BumpLeft,WheDropRight,WheDropLeft,WheDropCaster,BumpFront] = ...

BumpsWheelDropsSensorsRoomba(serPort);

end

StopCreate(serPort)

Signal()

%%%%%%%%%%%%%%%%%%%%%%%%

function StopCreate(serPort)

% Stop the robot

% serPort is the serial port number (for controlling the actual robot).

SetDriveWheelsCreate(serPort, 0,0)

%%%%%%%%%%%%%%%%%%%%%%%%

function Signal()

% Make signal sound (4 beeps)

n= 4;

for k=1:4

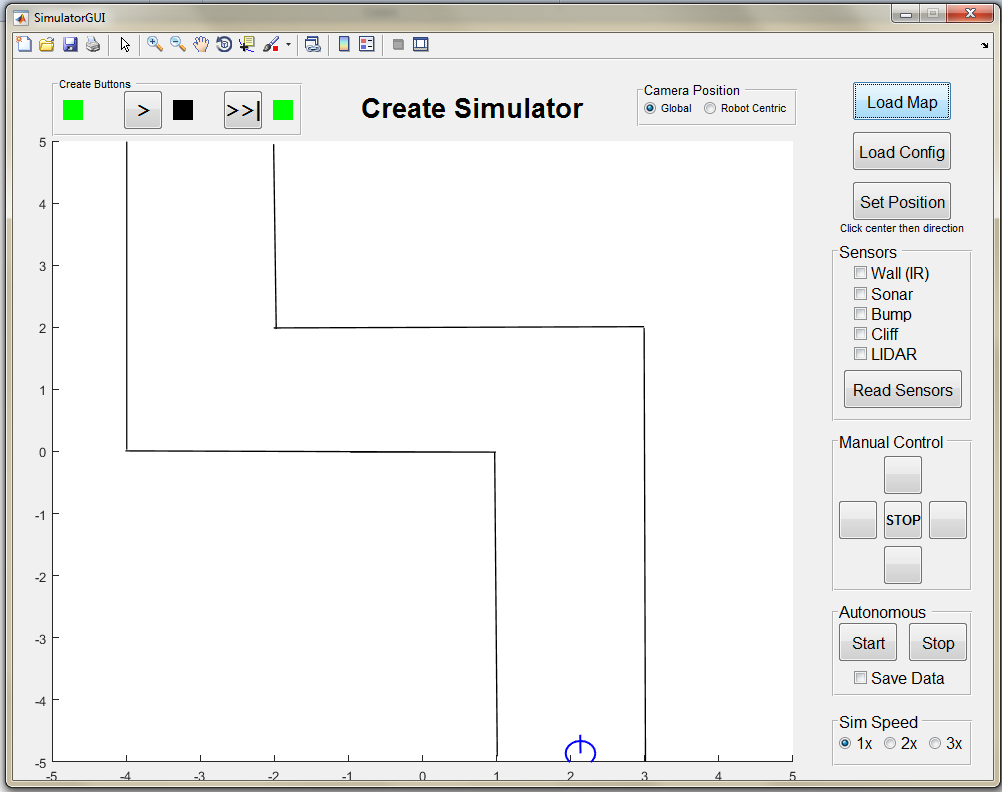
beep

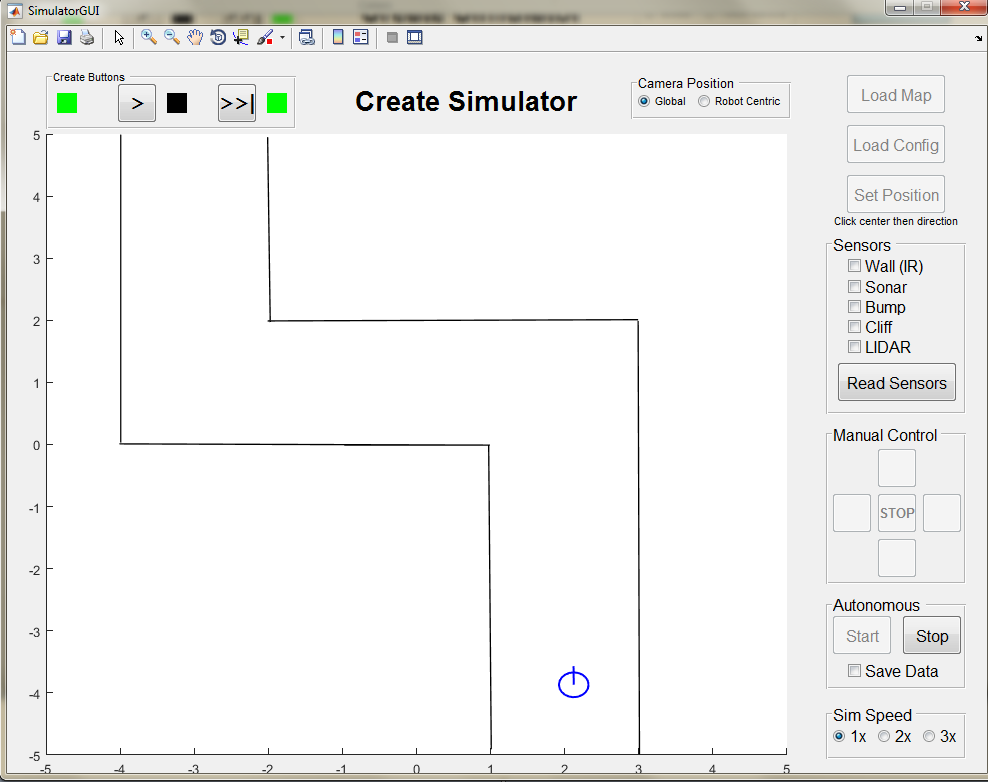
pause(.2)

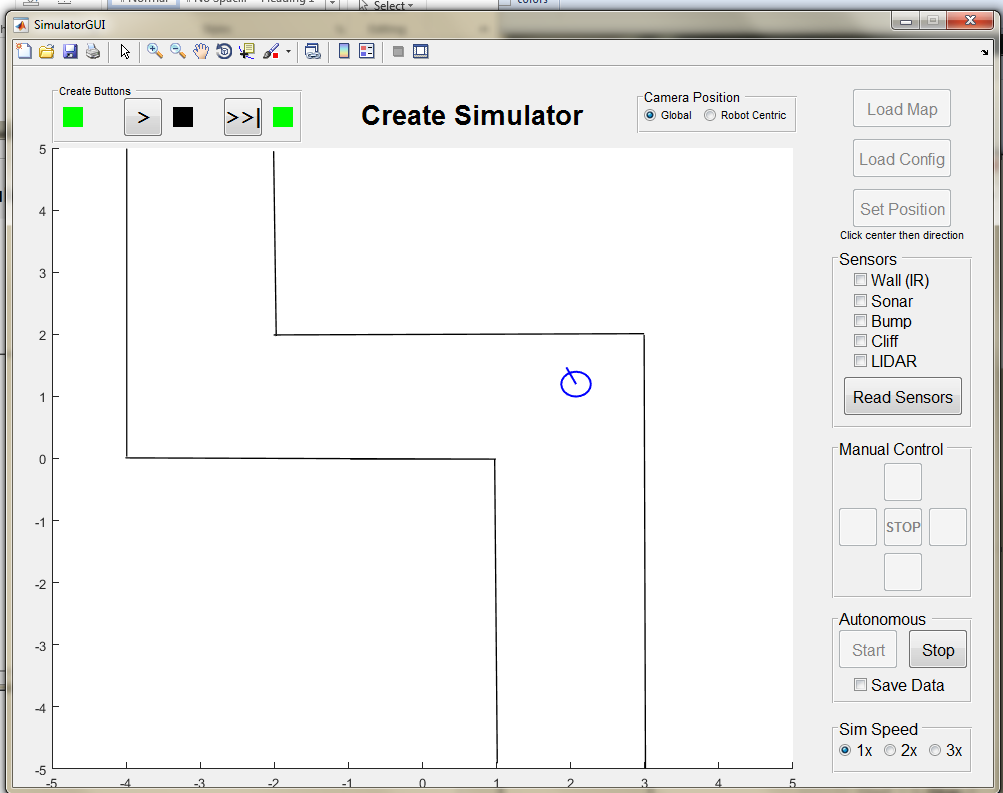
end

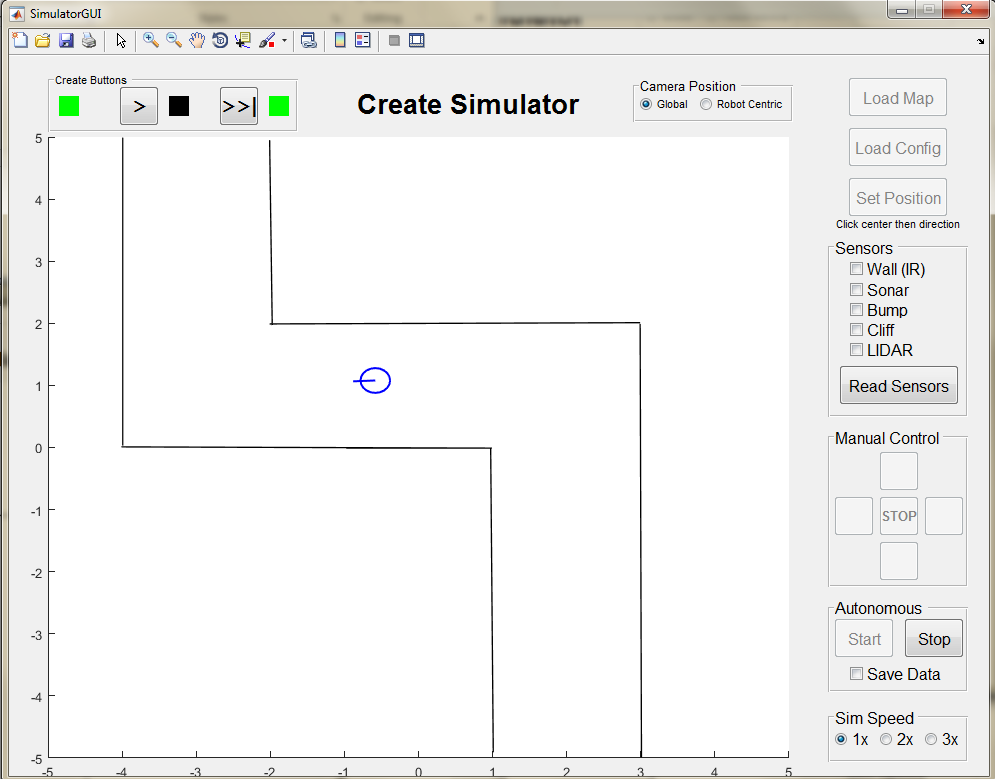
**4.0 Test Cases**

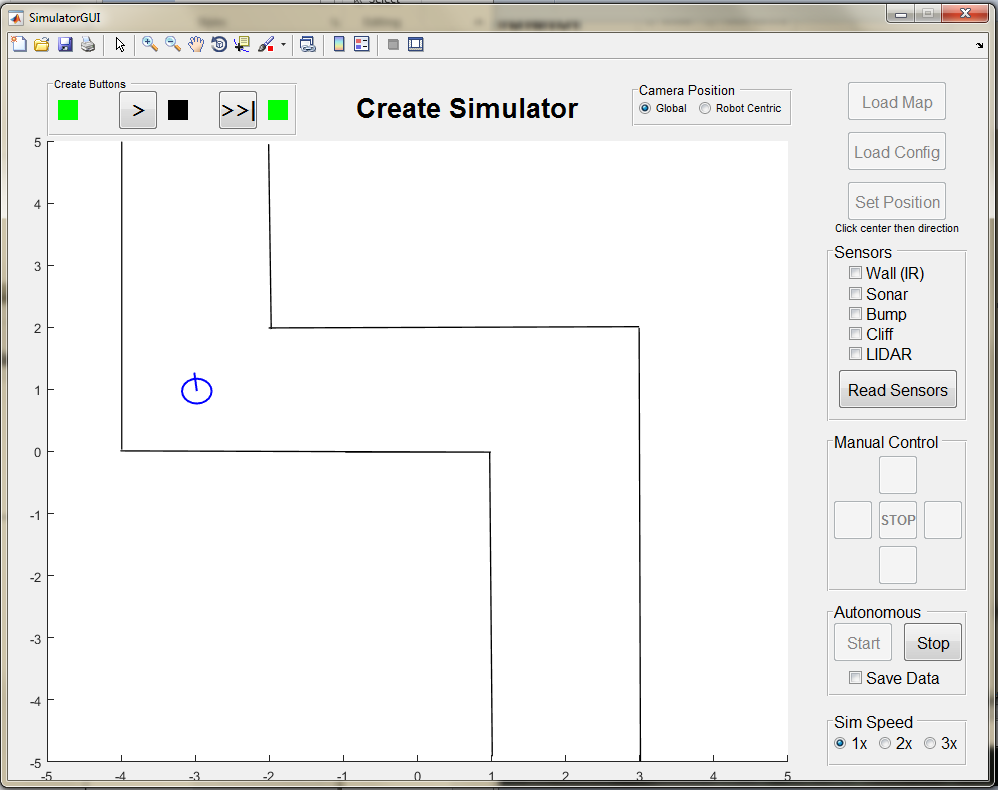
* Without Bump Sensor



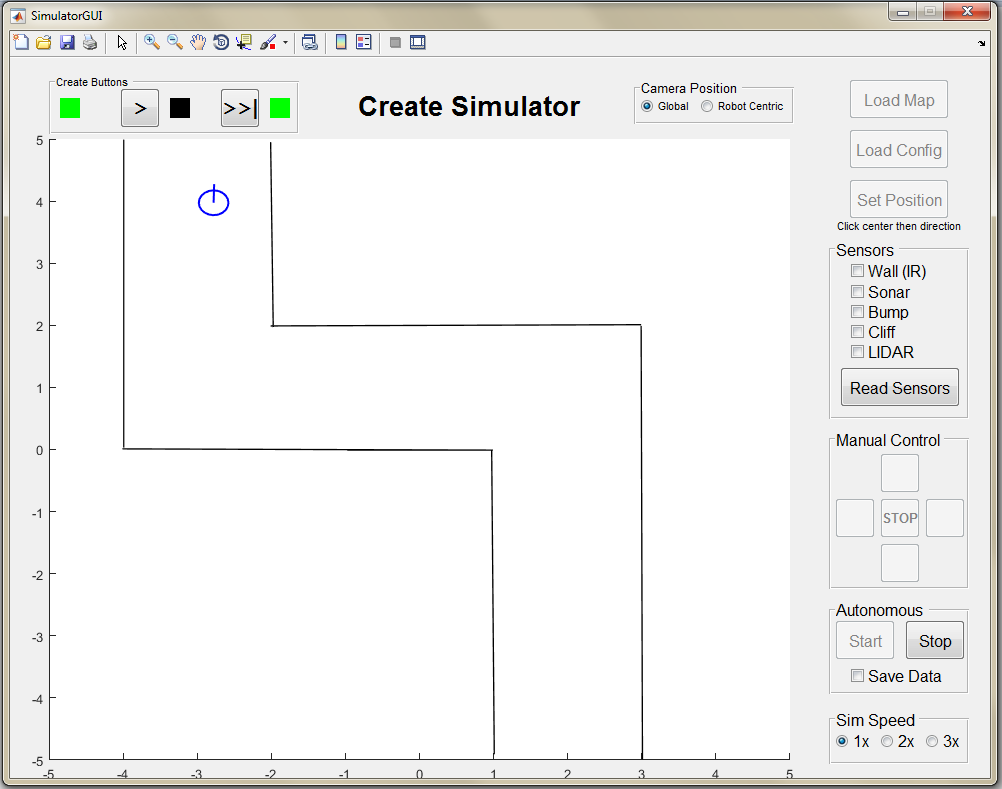




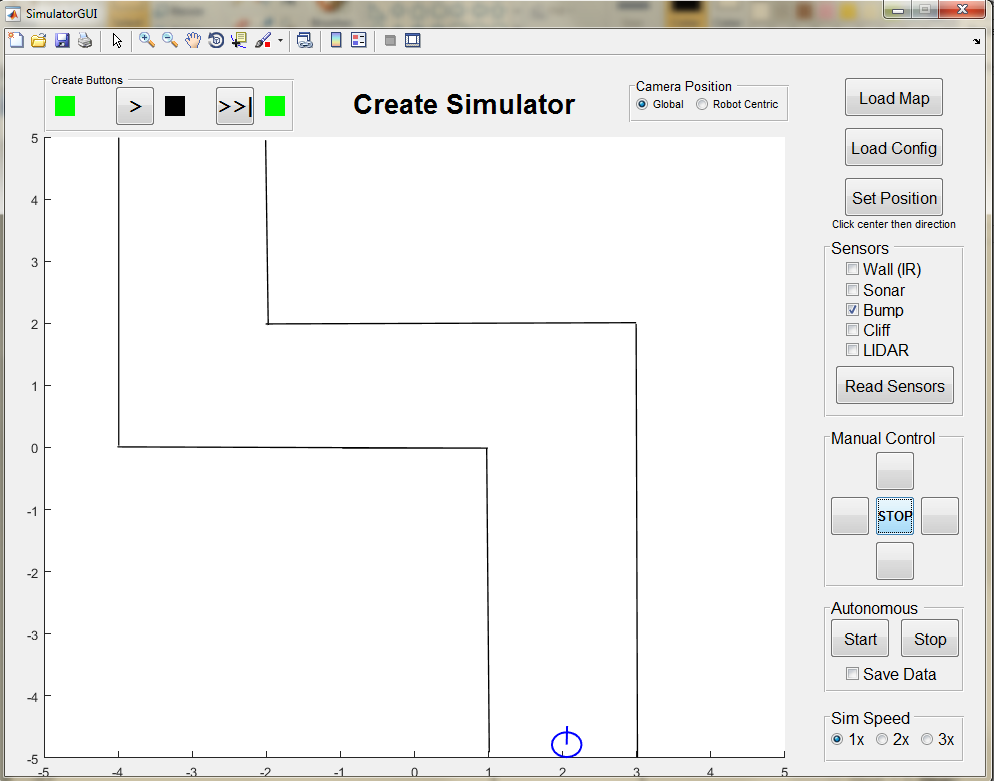


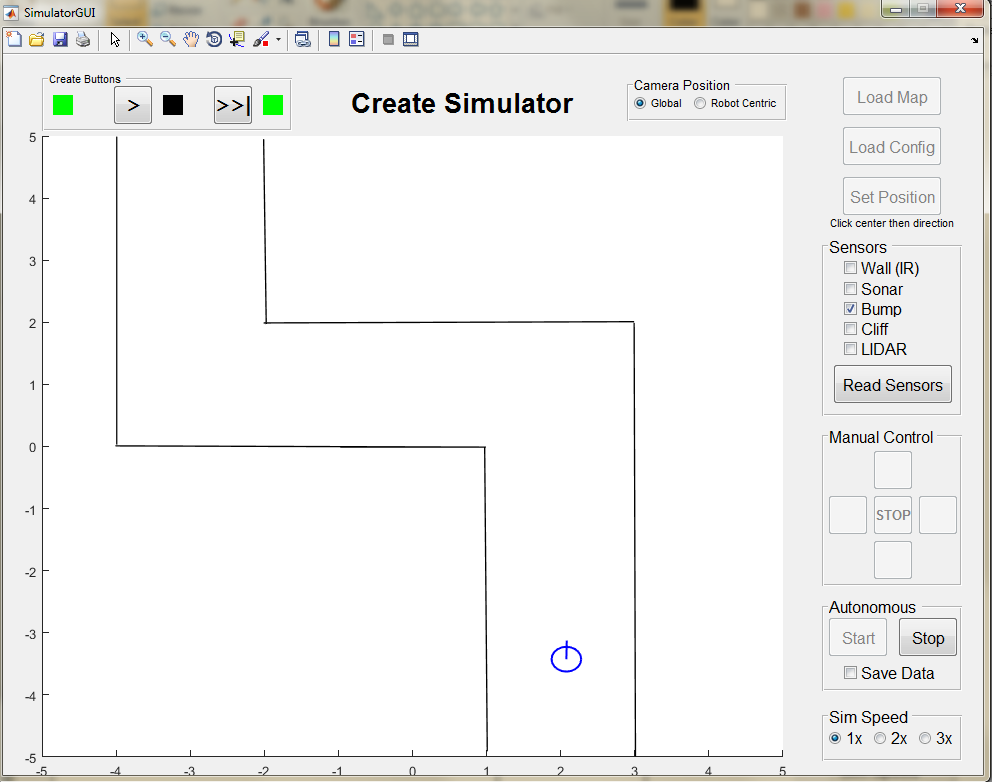


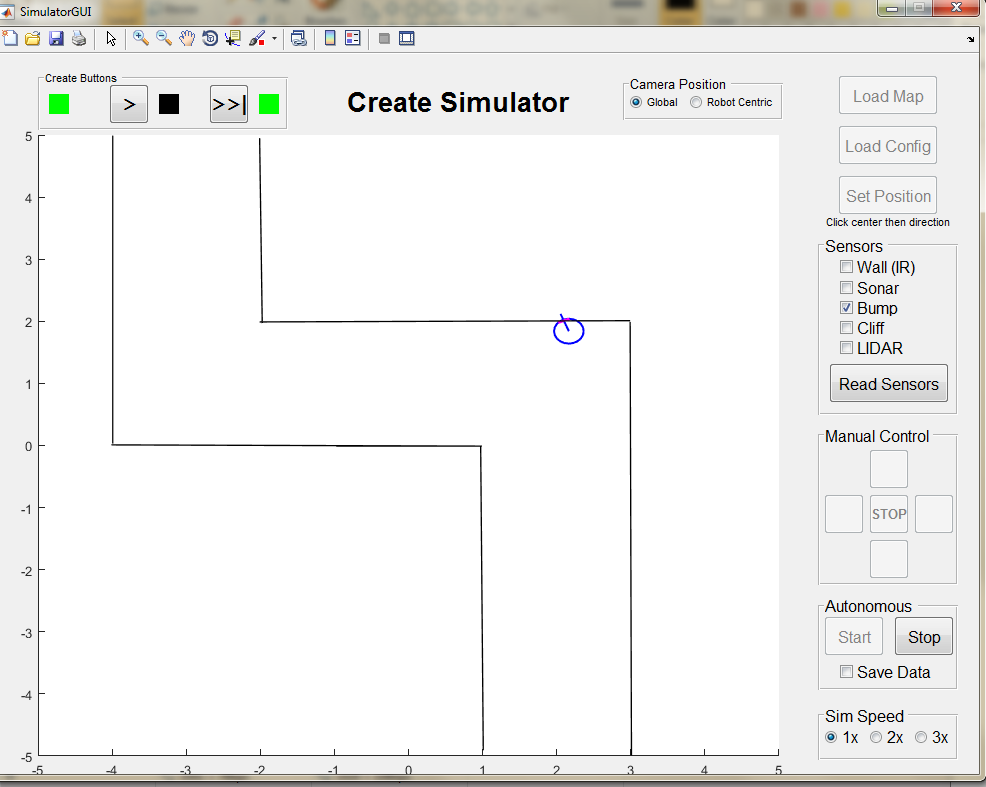
(This picture is from a different trial)

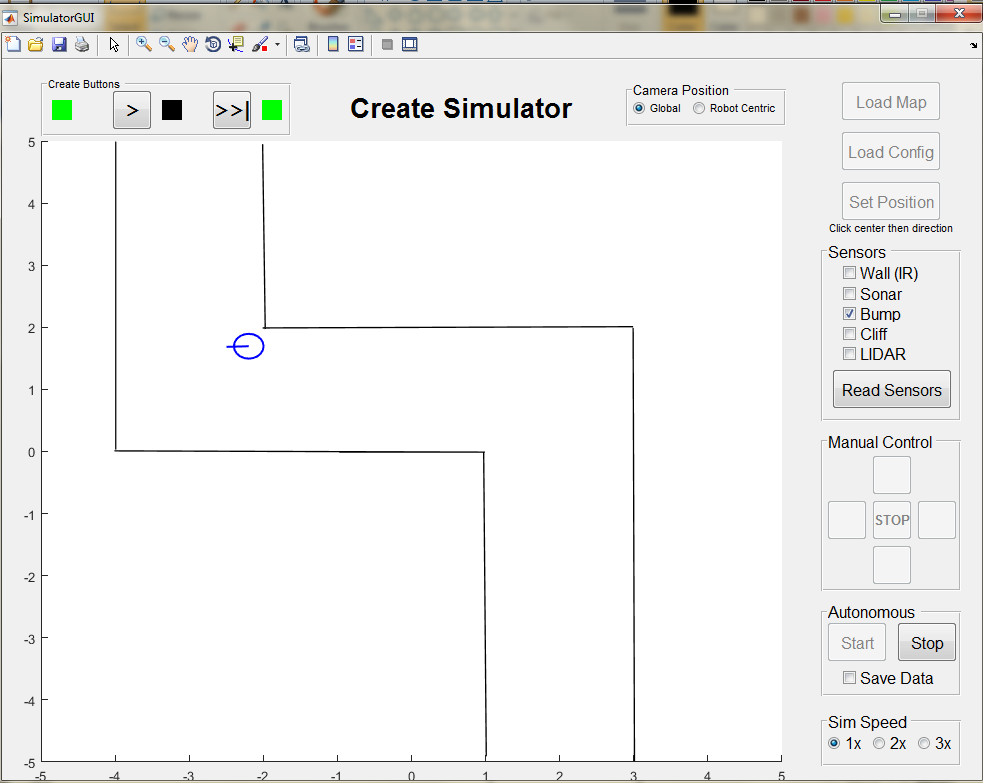


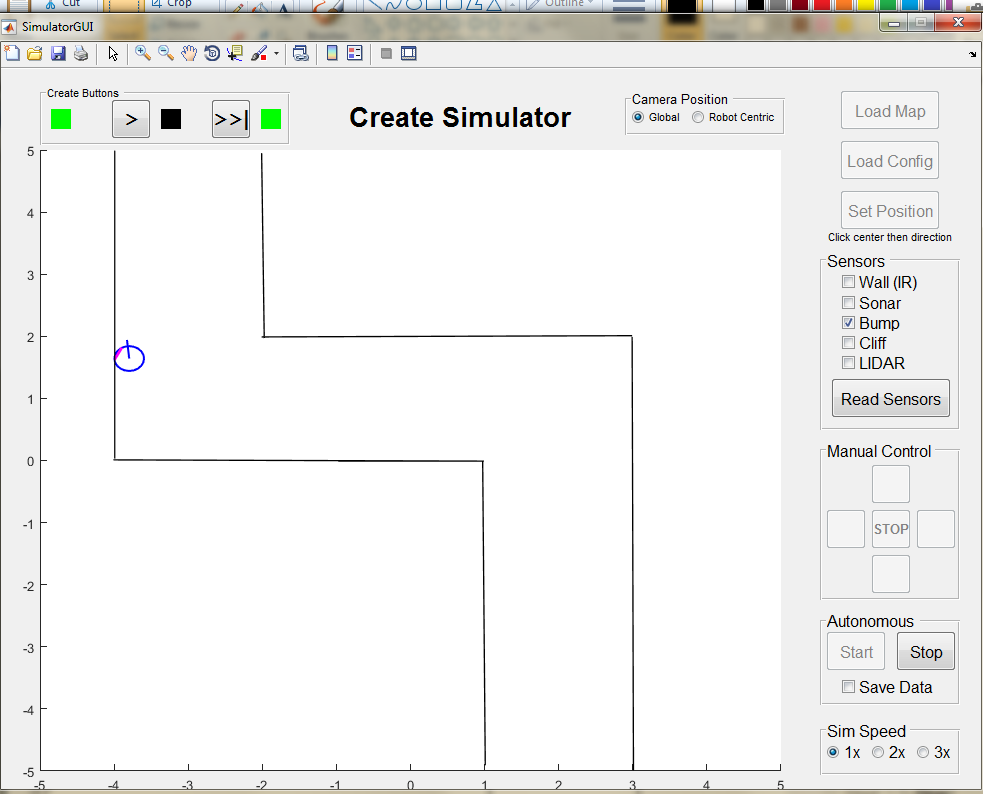
* Using Bump Sensor

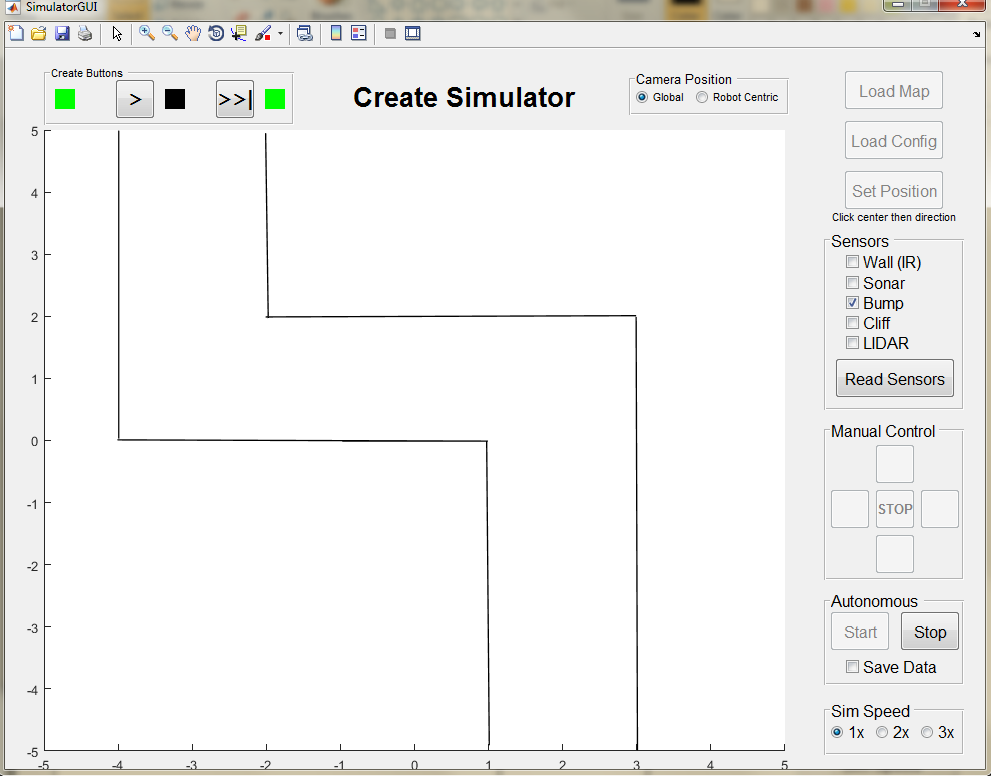
****

****

****

****

****

****