MTRE 4002 Feedback Control Laboratory

Lab 7 Report – Control of a Front-Wheel-Steered Robot for Moving to a Point and Line Following Problems

By
Levi J. Vande Kerkhoff
Chase Lolley

College of Engineering
Kennesaw State University
Marietta, GA

Experiment Objective

Students were required to design the equivalent control system for the kinematic model of a four-wheeled, front-wheel steered mobile robot. The kinematic model's equivalent control system was implemented as a subsystem within two mobile robot control systems: one that sends the robot to a desired location and the other causes the robot to follow a given line.

Background Knowledge

The following figures show the designed MATLAB Simulink models that controlled the theoretical kinematic control system model of the mobile robot. The robot's velocity is proportional to its distance from the goal location (x*, y*) and is given by the following equation:

$$v^* = K_v\sqrt{(x^* - x)^2 + (y^* - y)^2}$$

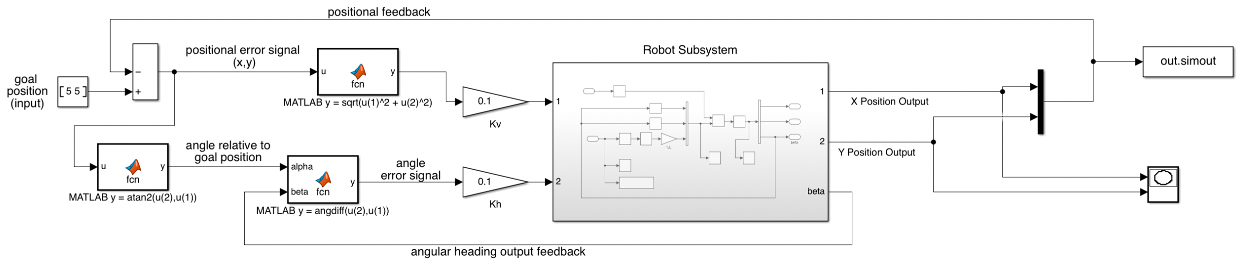
K_v is the controller gain associated with the velocity of the robot. To control the heading of the robot relative to the goal position, the following equation is employed:

$$\theta^* = tan^{-1}\left(\frac{y^* - y}{x^* - x}\right)$$

The steering angle is defined by the following statement:

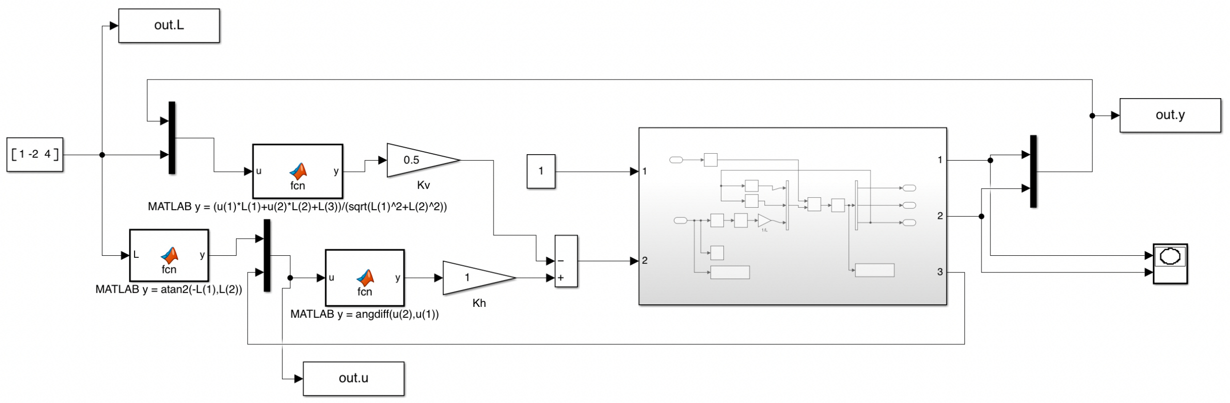
$$\gamma = K_h(\theta^* \ominus \theta), K_h > 0$$

The operator is equivalent to the angdiff() function (found in the Robotic System Toolbox).



positional feedback
goal position (input)
[5 5]
positional error signal (x,y)
MATLAB y = sqrt(u(1)^2 + u(2)^2)
0.1
Kv
Robot Subsystem
X Position Output
Y Position Output
out.simout
angle relative to goal position
MATLAB y = atan2(u(2),u(1))
angle error signal
MATLAB y = angdiff(u(2),u(1))
0.1
Kh
beta
angular heading output feedback

Figure 1: Mobile Robot Control System – Experiment 1, Drive to Goal Point



out.L
[1 -2 4]
MATLAB y = (u(1)*L(1)+u(2)*L(2)+L(3))/(sqrt(L(1)^2+L(2)^2))
0.5
Kv
1
out.y
MATLAB y = atan2(-L(1),L(2))
MATLAB y = angdiff(u(2),u(1))
1
Kh
out.u

Figure 2: Mobile Robot Control System – Experiment 2, Follow Parallel Line

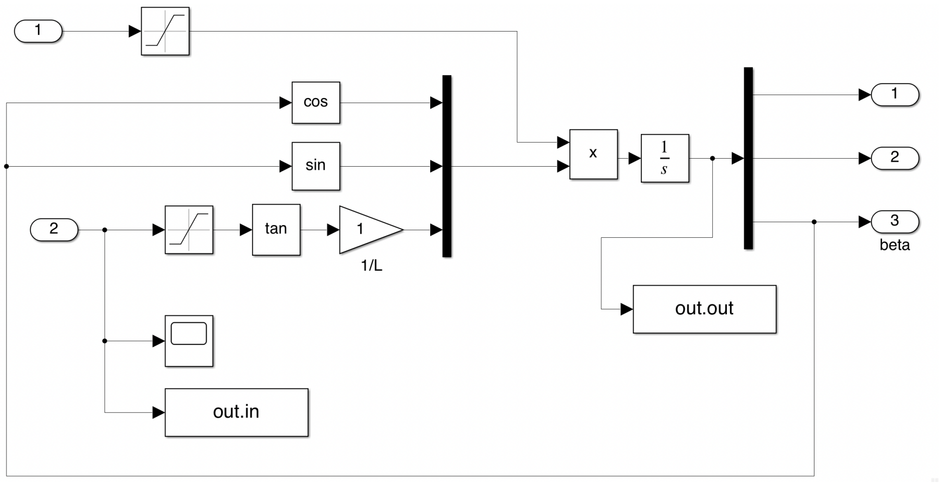


Figure 3: Mobile Robot Control Subsystem – Experiment 1 & 2

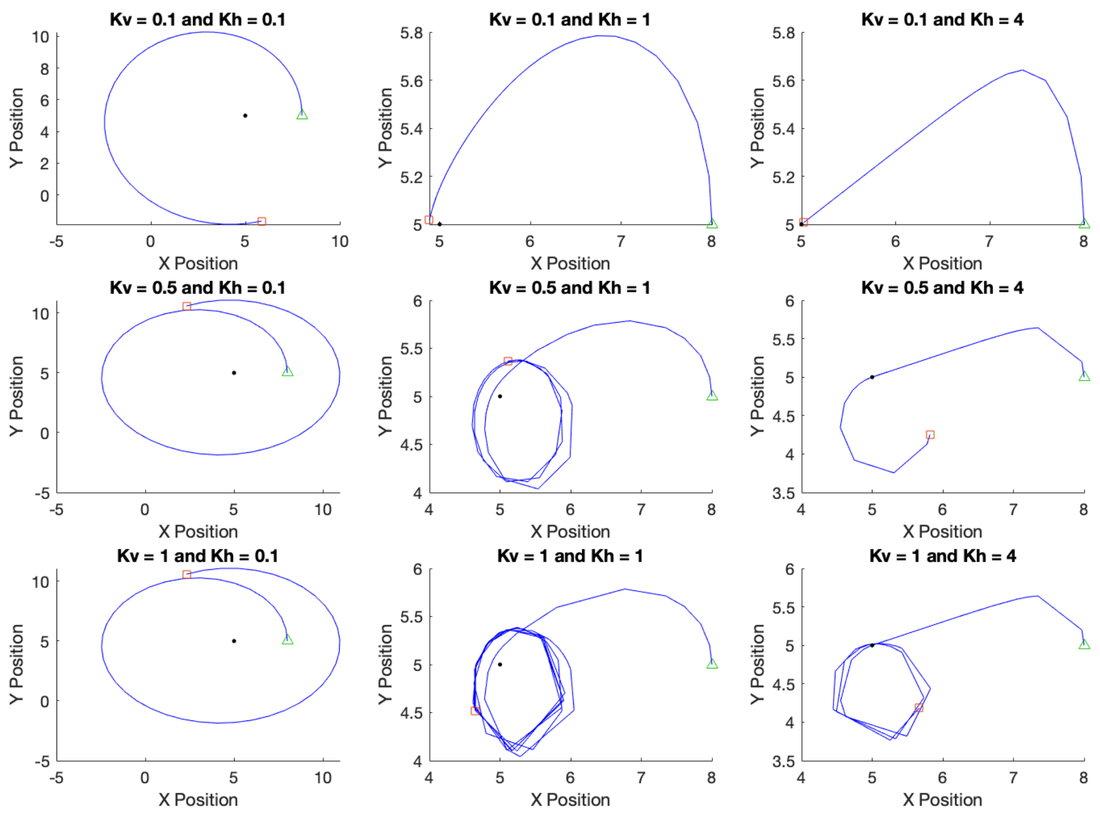


Figure 4: Plots Detailing the Robot Motion to Goal Point Trajectory

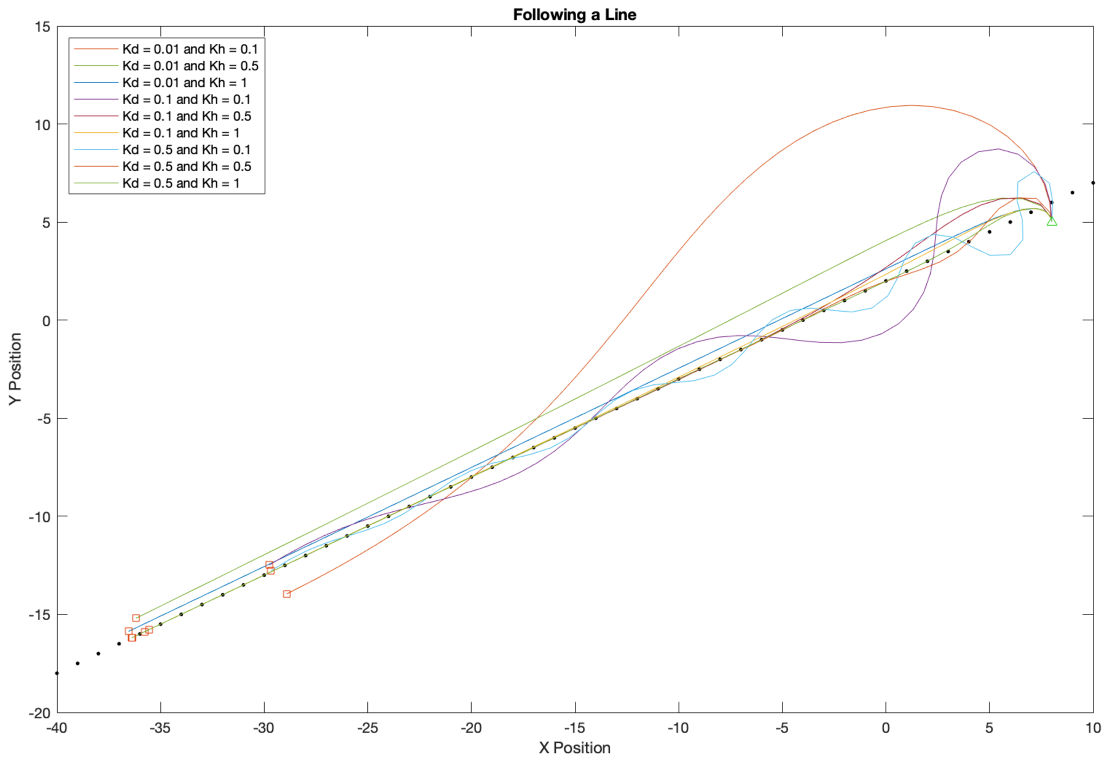


Figure 5: Plot Detailing Robot Line Following Trajectory

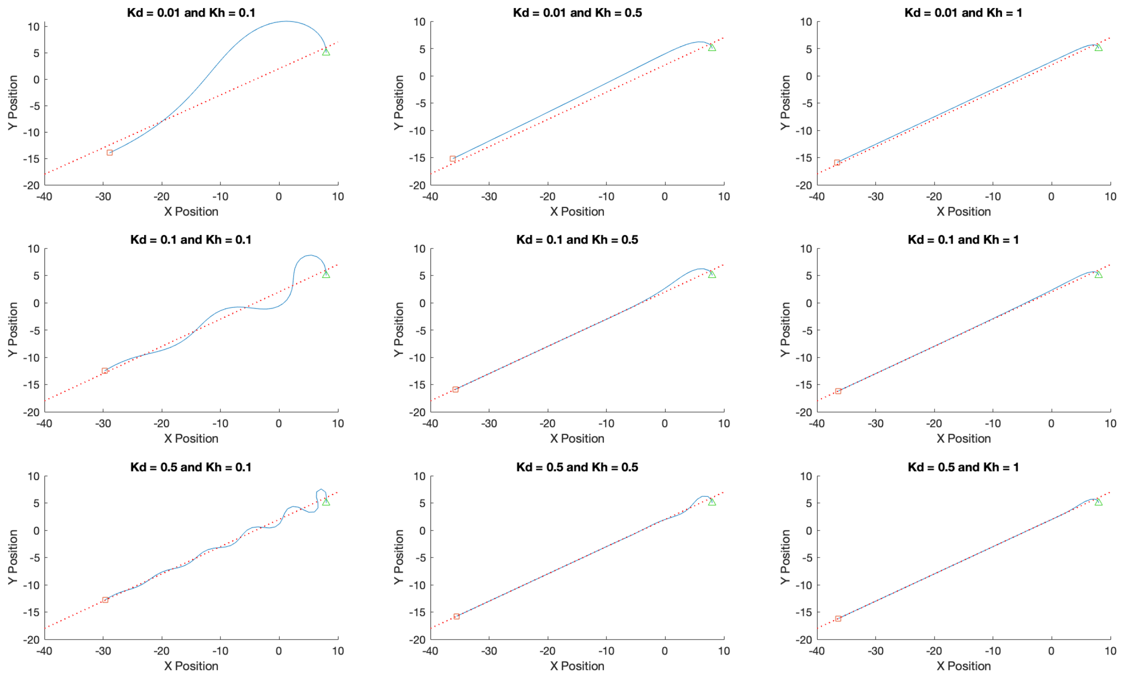


Figure 6: Subplots Detailing Robot Line Following Trajectory

Conclusion

The best controller gain values for Experiment 1 were 0.1 and 4 for K_v and K_h, respectively. These gain values produced a trajectory that was highly accurate with a quick turnaround; the distance required to change the robot's heading from $\frac{\pi}{2}$ to π was only 0.6, where, in other experiments, it was as high as 5. Other controller value combinations produced inaccurate trajectories that orbited the goal point. The best controller gain values for Experiment 2 were 0.5 and 1 for K_d and K_h, respectively. This produced the fastest and most accurate trajectory to the line.

Additional Resources: MATLAB Code Used to Retrieve Data From Workspace Files

```
%% Go to Location
figure(1)
i=1;

for kv_value = ["0.1", "0.5", "1"]
    for kh_value = ["0.1", "1", "4"]
        % the following line uses the for loop variables to write the file
        % name being loaded.
        file = "Lab07A_kv"+kv_value+"_kh"+kh_value+".mat";
        load(file); % this loads the file
        % assigns the data in the struct generated by simulink to a variable
        % for easier accessibility
        coord = out.simout.Data(1:end,1:end);
        x = coord(:,1);
        y = coord(:,2);
        subplot(3,3,i)
        hold on
        plot(8, 5, '^','Color','#26CA15') % start point
        plot(x, y, 'b') % robot path
        plot(x(end), y(end), 's','Color','#DC3C09') % end point
        plot(5,5,'.k') % goal location
        % the following line writes the title name in a string
        PlotTitle = 'Kv = ' + kv_value + ' and Kh = ' + kh_value;
        ylabel("Y Position")
        xlabel("X Position")
        title(PlotTitle)
        i = i + 1;
    end
end


%% Follow a Line
figure(2)
i = 1;
x = -40:10;
y = 0.5*x + 2;
plot(x,y, '.k', 'LineWidth', 1) % line to follow

for kd_value = ["0.01", "0.1", "0.5"]
    for kh_value = ["0.1", "0.5", "1"]
        file = "Lab07B_kd"+kd_value+"_kh"+kh_value+".mat";
        load(file);
        coord = out.y.Data(1:end,1:end);
        x = coord(:,1);
        y = coord(:,2);
        hold on
        myPlots(i) = plot(x, y); % saves the plots in an array of plots
        plot(8, 5, '^','Color','#26CA15')
        plot(x(end), y(end), 's','Color','#DC3C09')
        title("Following a Line")
        ylabel("Y Position")
        xlabel("X Position")
        % the following line writes the legend names as strings in a string
        % array indexed by our counter variable 'i'.
        LegendName(i) = 'Kd = ' + kd_value + ' and Kh = ' + kh_value;
        i = i + 1;
    end
end
% the following line only creates legends according to the path plots
% created under the variable 'MyPlots'.
legend(myPlots(1:9), LegendName, 'Location', 'northwest')
```