AA 274: Principles of Robotic Autonomy
Problem Set 1

Name: Tariq Zahroof
SUID: tzahroof

Problem 1: Optimal Control

Derive the Hamiltonian and conditions for optimality and formulate the problem as a 2P-BVP. Make sure you explain your steps, and that you include boundary conditions.

(i) The function has the form:

$$\begin{aligned} J &= 0 + \int_0^{t_f} [\lambda + V^2 + w^2] dt \\ &= h(x_f, t_f) + \int_0^{t_f} [g(x, u, t)] dt \end{aligned}$$

The Hamiltonian equations can then be found by:

$$\dot{x} = a(x, u, t) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V\cos\theta \\ V\sin\theta \\ w \end{bmatrix}$$

$$H = g(x, u, t) + p^T a(x, u, t)$$

$$\implies H = \lambda + v^2 + w^2 + \begin{bmatrix} p1 & p2 & p3 \end{bmatrix} \begin{bmatrix} V\cos\theta \\ V\sin\theta \\ w \end{bmatrix}$$

$$\dot{p} = -\frac{\partial H}{\partial x} = \begin{bmatrix} 0 \\ 0 \\ p_1 V \sin\theta - p_2 V \cos\theta \end{bmatrix}$$

$$0 = \frac{\partial H}{\partial u} = \begin{bmatrix} 2V + p_1\cos\theta + p_2\sin\theta \\ 2w + p_3 \end{bmatrix}$$

$$\therefore w = \frac{-p_3}{2}, V = \frac{(-p_1\cos\theta + p_2\sin\theta)}{2}$$

The following change of variables is used for guaranteeing a τ between 0, 1 for programming implementation:

$$r = t_f, \dot{r} = 0$$

Thus $\frac{\partial z}{\partial \tau}$ becomes the following, with the V and w substitution given above:

$$\frac{\partial z}{\partial \tau} = t_f * \frac{\partial z}{\partial t} = \begin{bmatrix} \frac{\partial x}{\partial \tau} \\ \frac{\partial y}{\partial \tau} \\ \frac{\partial \theta}{\partial \tau} \\ \frac{\partial p_1}{\partial \tau} \\ \frac{\partial p_2}{\partial \tau} \\ \frac{\partial p_3}{\partial \tau} \\ \frac{\partial r}{\partial \tau} \end{bmatrix} = \begin{bmatrix} rV\cos\theta \\ rV\sin\theta \\ rw \\ 0 \\ 0 \\ r(p_1 V\sin\theta - p_2 V\cos\theta) \\ 0 \end{bmatrix}$$

Since we have a fixed x_f, but a free t_f, the following BC is incorporated:

$$H + \frac{\partial h}{\partial t} = 0 \implies H = 0$$

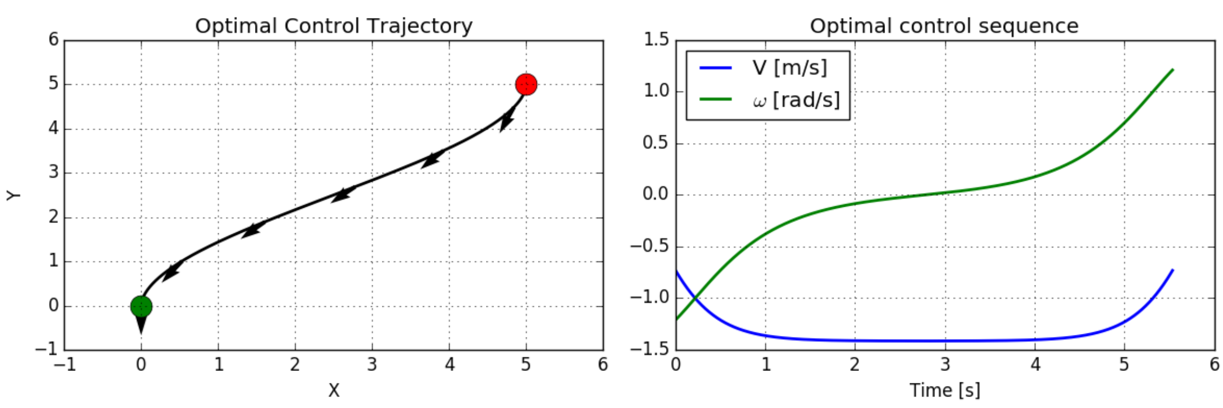
The following boundary values are used to solve the equation.

$$x(0) = 0, x(1) = 5$$
$$y(0) = 0, y(1) = 5$$
$$\theta(0) = \frac{\pi}{2}, \theta(1) = \frac{\pi}{2}$$
$$H_f = \lambda + v^2 + w^2 + \begin{bmatrix} p1 & p2 & p3 \end{bmatrix} \begin{bmatrix} V\cos\theta \\ V\sin\theta \\ w \end{bmatrix} = 0$$

(ii) Complete `P1_optimal_control.py`.

Code attached in zip file

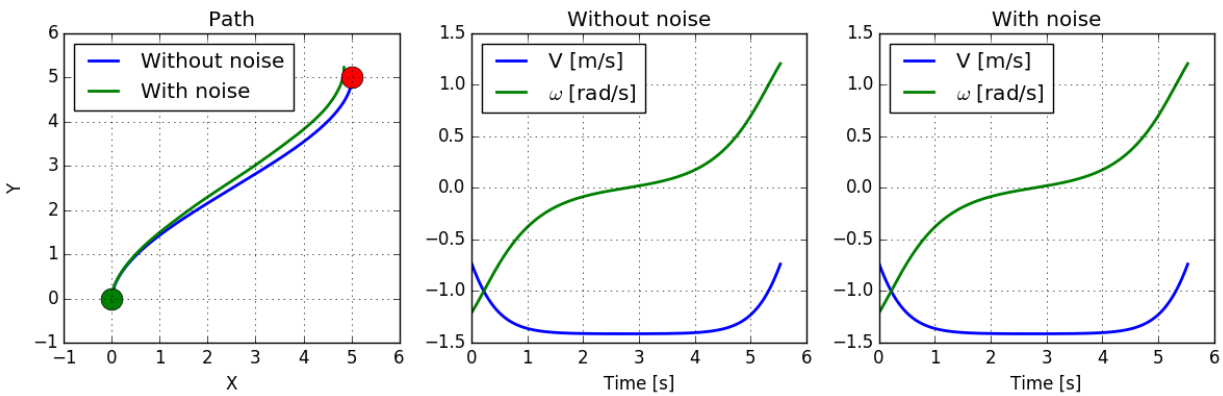
(iii) Include `optimal_control.png`



(iv) Explain the significance of using the largest feasible λ.

λ signifies the penalty on time. Having a large λ penalty increases total trajectory cost the longer the controller makes decisions. Thus, a large λ forces the trajectory to exert more control to reach the goal region faster (and therefore produces a lower t_f.

(v) Simulate car with open-loop optimal control.



Problem 2: Differential Flatness

(i) Write a set of linear equations to express the initial and final conditions.

The following shows the important equations for problem setup

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{V} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} V\cos\theta \\ V\sin\theta \\ a \\ w \end{bmatrix}$$
$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -V\sin\theta \\ \sin\theta & V\cos\theta \end{bmatrix} = \begin{bmatrix} a \\ w \end{bmatrix}$$

By differential flatness, x and y can be written as follows:

$$x = \sum_{1}^{n} x_i \psi_i$$
$$y = \sum_{1}^{n} y_i \psi_i$$

Thus x and $\dot{x}$ can be written as follows (can be applied to y):

$$\psi_1 = 1, \psi_2 = t, \psi_3 = t^2, \psi_4 = t^3$$
$$x = x_1(1) + x_2(t) + x_3(t^2) + x_4(t^3)$$
$$\dot{x} = x_2 + 2 * x_3(t) + 3 * x_4(t^2)$$

Then, x and y may be determined by the following format ($A_x = A_y$). Note that A is determined by the coefficients of the x_i,y_i variables from the previous equation.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 15 & 15^2 & 15^3 \\ 0 & 1 & 30 & 675 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \end{bmatrix} = \begin{bmatrix} x_0 \\ V_0 \cos\theta_i \\ x_f \\ V_f \cos\theta_i \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 15 & 15^2 & 15^3 \\ 0 & 1 & 30 & 675 \end{bmatrix} \begin{bmatrix} y1 \\ y2 \\ y3 \\ y4 \end{bmatrix} = \begin{bmatrix} y_0 \\ V_0 \sin\theta_f \\ y_f \\ V_f \sin\theta_f \end{bmatrix}$$

with the following initial and final conditions:

$$x_0 = 0, y_0 = 0, V_0 = 0.5, \theta_0 = -\frac{\pi}{2}$$
$$x_f = 5, y_f = 5, V_f = 0.5, \theta_0 = -\frac{\pi}{2}$$

(ii) Why can one not set $V(t_f) = 0$?

Matrix J is defined as follows:

$$J = \begin{bmatrix} \cos\theta & -V\sin\theta \\ \sin\theta & V\cos\theta \end{bmatrix}$$

As such, when V_f becomes 0, the matrix is no longer invertible. In other words, the system loses the ability to map the control space (a, u) to the input space (x, y), and thus loses the properties of differential flatness (and the previous solutions no longer apply).

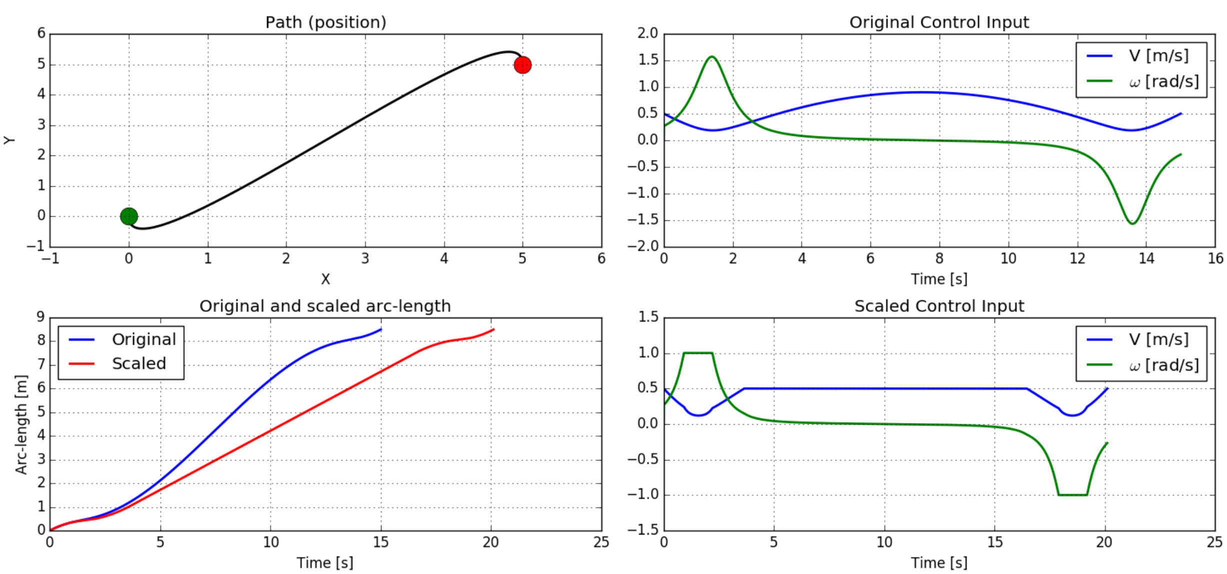
(iii) Complete the function: `differential_flatness_trajectory`

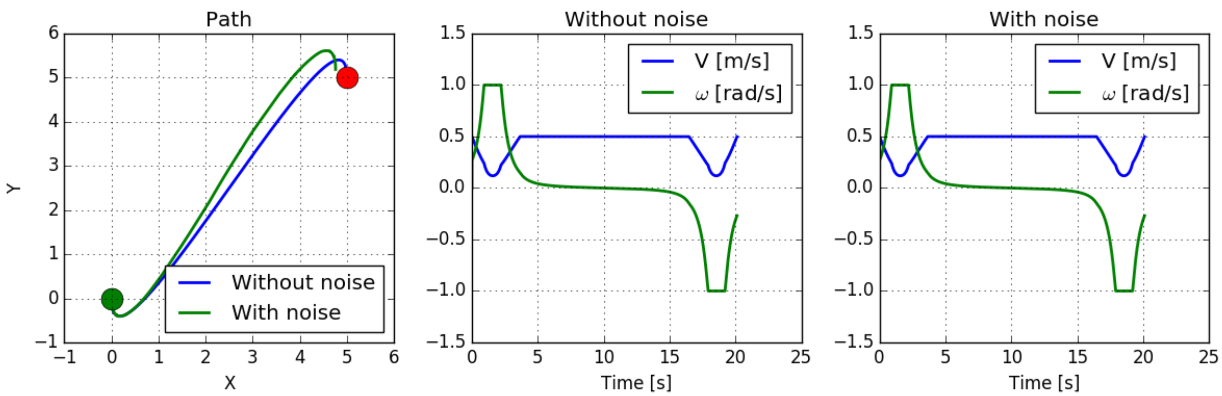
See code in zip.

(iv) Complete the rest of: `P2_differential_flatness.py`

See code in zip

(v) Validate work by running open-loop differential flatness law





Problem 3: Closed Loop Control I

(i) Complete the function `ctrl_pose`

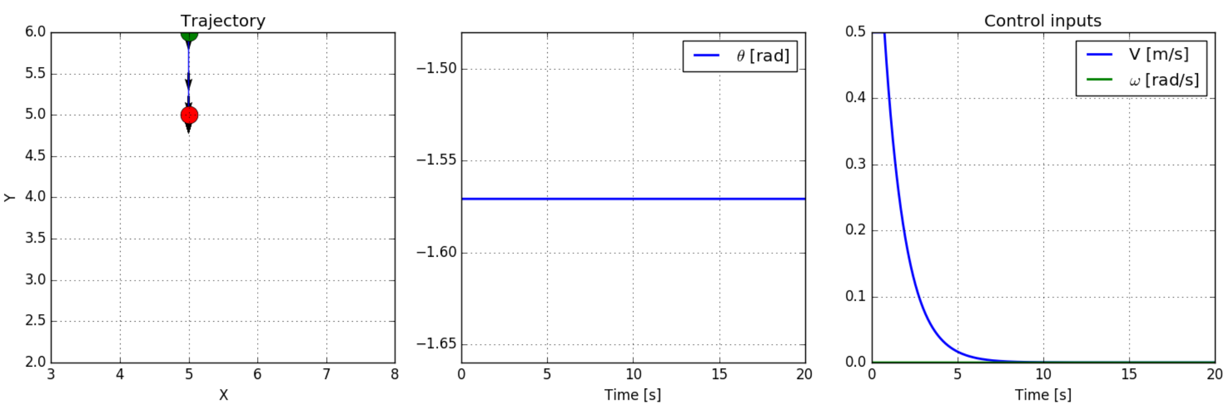
Code included in zip file

(ii) Validate `ctrl_pose`

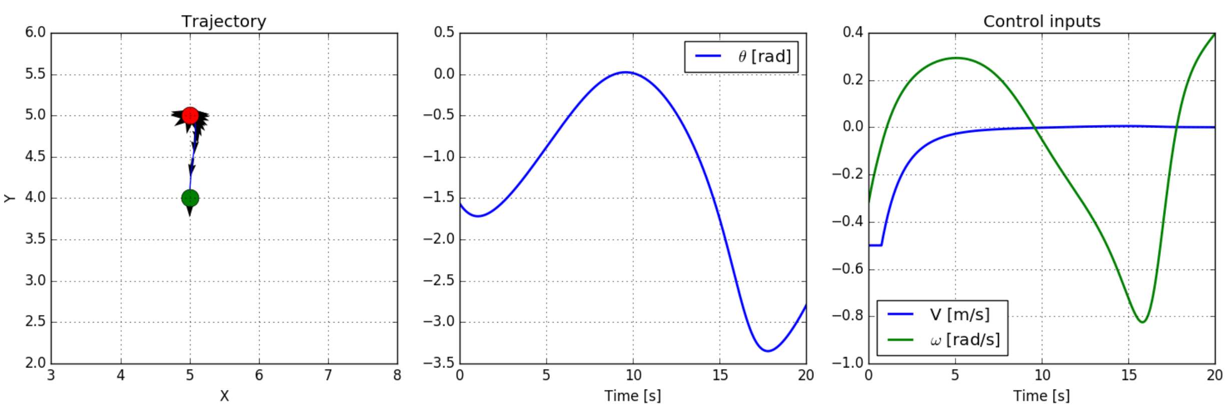
Code included in zip file

(iii) Include plots from validating closed-loop control law.

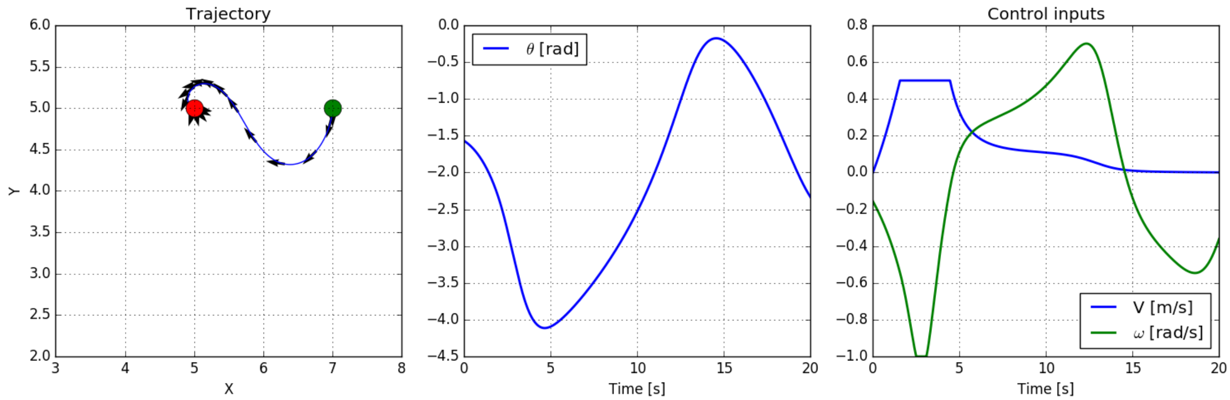
Forwards:



Reverse:



Parallel:



Problem 4: Closed Loop Control II

(i) Write down a system of equations for computing the true control inputs (V, u) in terms of the virtual controls and the vehicle state

From Problem 2, we have the controls equation:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -V\sin\theta \\ \sin\theta & V\cos\theta \end{bmatrix} \begin{bmatrix} a \\ w \end{bmatrix}$$

$$\begin{bmatrix} \dot{V} \\ w \end{bmatrix} = \begin{bmatrix} a \\ w \end{bmatrix}$$

Therefore, by rearranging, we can obtain the (V, w) controls in terms of $((u_1, u_2) = (\ddot{x}, \ddot{y}))$

$$\begin{bmatrix} \dot{V} \\ w \end{bmatrix} = \begin{bmatrix} \cos\theta & -V\sin\theta \\ \sin\theta & V\cos\theta \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

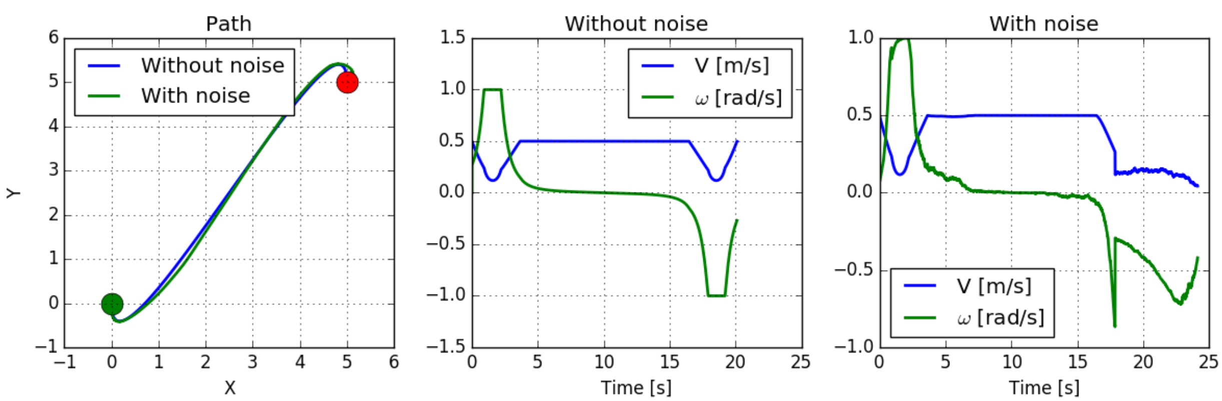
(ii) Complete the function `ctrl_traj`

See code in zip

(iii) Modify controller to switch to pose stabilization controller when sufficiently close to goal.

See code in zip.

(iv) Validate work (closed loop control on differential flatness)



Problem 5: ROS

(i) Record the outputs of the publisher publishing your name.

The publisher was recorded into `name.bag`

(ii) What is command to play back a rosbag?

```
$ rosbag play bagname.bag
```

(iii) What command must be ran after writing a python node?

The following command must be ran to ensure that the python node is executable (and thus usable by ROS)

```
$ chmod +X node.py
```

(iv) What is the command to show information about active topics?

`rostopic`

(v) Complete the code in `controller.py` and record the control outputs.

See code in zip. The controller outputs were stored in `gazebo.bag`.