AE353: Design Problem 01

Cole Rodgers

October 7, 2020

1 Goal

The code provided in DesignProblem01 simulates the 2D dynamics of a simplified powered descent vehicle for delivering a payload on Martian surface. The descent and landing phase of such a vehicle involves using onboard thrusters to reduce velocity and establish an upright position to hover above the ground, followed by employing a crane mechanism to safely deploy the payload onto the ground.

Our idealized delivery vehicle has two thrusters that can apply thrust to control the motion in three degrees of freedom: horizontal motion, vertical motion, and rotation. It also has sensors—e.g., an inertial measurement unit (IMU), a radar, and/or an optical flow sensor—that can be used to estimate the vehicle's altitude and horizontal position, horizontal and vertical velocities, orientation, and angular velocity. The powered descent starts at some initial altitude and orientation. The goal is to achieve a particular state that is ideal for satisfying the mission requirements.

2 Model

A free body diagram of the forces acting on the vehicle are given in the problem statement. The sky crane—which we are not attempting to control—has a maximum reach of 40 meters. Activating the sky crane mechanism when the vehicle is not stationary or too close to the ground could lead to instabilities in the system and damage the payload. We assume that the ground is flat, and the ground frame of reference is outlined in the problem statement.

3 Requirements

3.1 System Equations of Motion

Equations of motion as derived from the model (with respect to ground frame of reference):

$$\ddot{x} = \frac{F_L}{m}\cos(\theta + 45^\circ) - \frac{F_R}{m}\sin(\theta + 45^\circ)$$

$$\ddot{z} = \frac{F_L}{m}\sin(\theta + 45^\circ) + \frac{F_R}{m}\cos(\theta + 45^\circ) - g$$

$$\ddot{\theta} = \frac{a}{J\sqrt{2}}(F_L - F_R) + \frac{b}{J\sqrt{2}}(F_R - F_L)$$

Expressing these equations as a system of first order ODEs:

$$s = \begin{bmatrix} x - x_e \\ \dot{x} - \dot{x}_e \\ z - z_e \\ \dot{z} - \dot{z}_e \\ \theta - \theta_e \\ \dot{\theta} - \dot{\theta}_e \end{bmatrix}, \quad \dot{s} = \begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \dot{s}_3 \\ \dot{s}_4 \\ \dot{s}_5 \\ \dot{s}_6 \end{bmatrix} = \begin{bmatrix} \dot{x} - \dot{x}_e \\ \ddot{x} - \ddot{x}_e \\ \dot{z} - \dot{z}_e \\ \ddot{z} - \ddot{z}_e \\ \dot{\theta} - \dot{\theta}_e \\ \ddot{\theta} - \ddot{\theta}_e \end{bmatrix} = \begin{bmatrix} s_2 \\ \frac{F_L}{m}\cos(\theta + 45^\circ) - \frac{F_R}{m}\sin(\theta + 45^\circ) \\ s_4 \\ \frac{F_L}{m}\sin(\theta + 45^\circ) + \frac{F_R}{m}\cos(\theta + 45^\circ) - g \\ s_6 \\ \frac{a}{J\sqrt{2}}(F_L - F_R) + \frac{b}{J\sqrt{2}}(F_R - F_L) \end{bmatrix}$$

3.2 State Variables and Desired State

As shown above, my state variables s are defined as:

$$s = \begin{bmatrix} x - x_e & \dot{x} - \dot{x}_e & z - z_e & \dot{z} - \dot{z}_e & \theta - \theta_e & \dot{\theta} - \dot{\theta}_e \end{bmatrix}^T$$

Based on the mission requirements, the desired states are defined as:

$$\text{equilibrium} = \begin{bmatrix} x_e \\ \dot{x}_e \\ z_e \\ \dot{z}_e \\ \theta_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} 250 \\ 0 \\ 400 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } F_{Le} = F_{Re} = \frac{mg}{\sqrt{2}}$$

Note: x_e was not directly specified so I chose 250 as my equilibrium x-position.

3.3 Linearization

In order to express my model in standard state-space form, I needed to linearize the system: $\dot{s} = As + Bu$

$y = Cs + Du$ where,

$A =$

0 1 0 0 0 0 0 0

0 0 0 0 −g 0 0 0

0 0 0 0 0 0 0 1

0 0 0 0 0 0 0 0

0 0 0 0 0 0

$B =$

0 0 7.443×10^{-4} 7.443×10^{-4} 0 0
-7.443×10^{-4} 0 -1.95×10^{-4}
0 1.95×10^{-4}
7.443×10^{-4}

D
$C =$ $= [0]$

1
0 0 0 0 0
0 1 0 0 0 0 0 0
1 0 0 0 0 0 0 1
0 0 0 0 0 0 1 0
0 0 0 0 0 1

2

3.4 Zero Input System

For zero input on the linearized model, the equation $\dot{s} = As + Bu$ becomes $\dot{s} = As$. Further more, A has 6 eigenvalues all equaling 0, meaning the system is not asymptotically stable. All eigenvalues need to be negative for the system to be asymptotically stable.

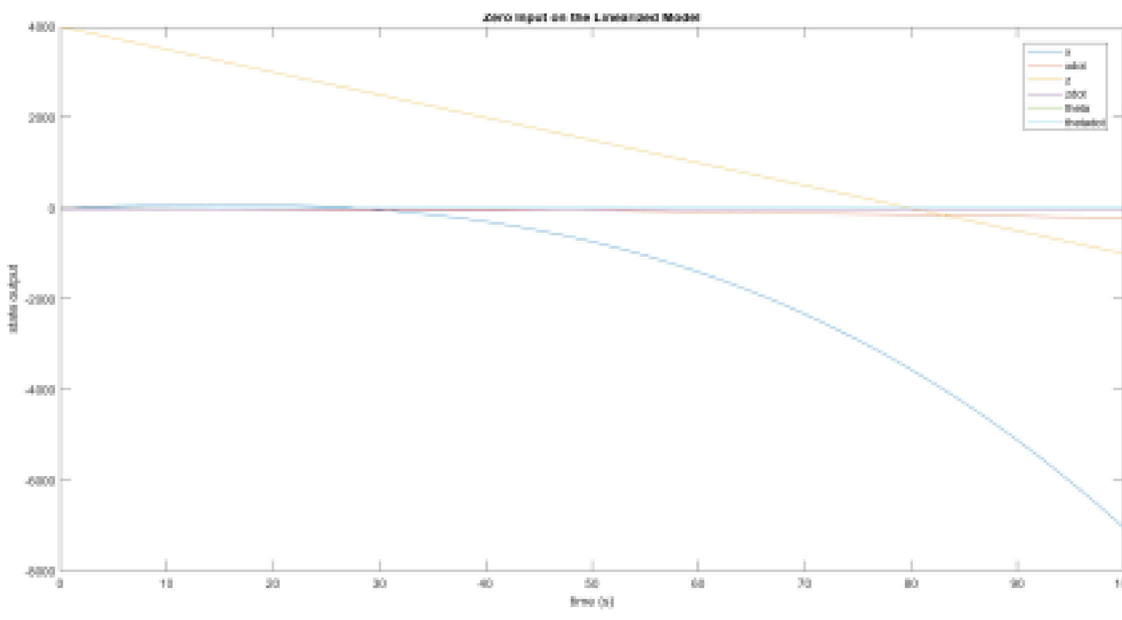

Figure 1: It can be observed from this plot that when zero input is applied to the linearized model, the system is not asymptotically stable, namely z (in yellow) and x (in blue).

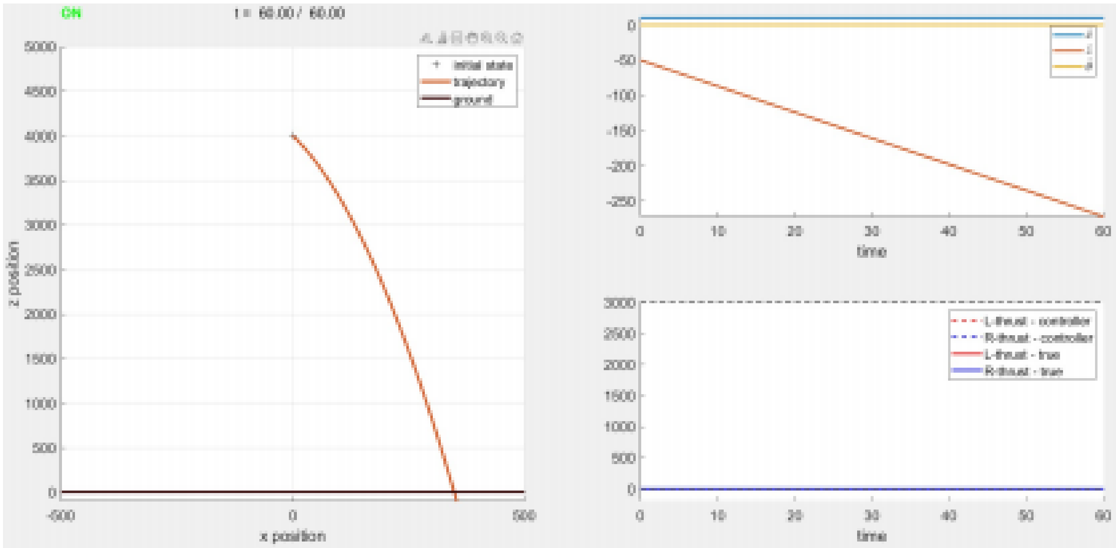

Figure 2: It can be observed from this plot that when zero input is applied to the non-linear model, the system is still not asymptotically stable, namely z (in red) and x (in blue).

3.5 State Feedback System

For a state feedback system, the equation $\dot{s} = As + Bu$ becomes $\dot{s} = (A - BK)s$ when the controller $u = -Ks$ is implemented, where

K= 2103 −145 −1739 129 637 18074 9877 −124 −1412 124 627 12820

Note: this is just one possible choice for K

Using this K, the eigenvalues of $(A - BK)$ are all negative, meaning this system is asymptotically stable.

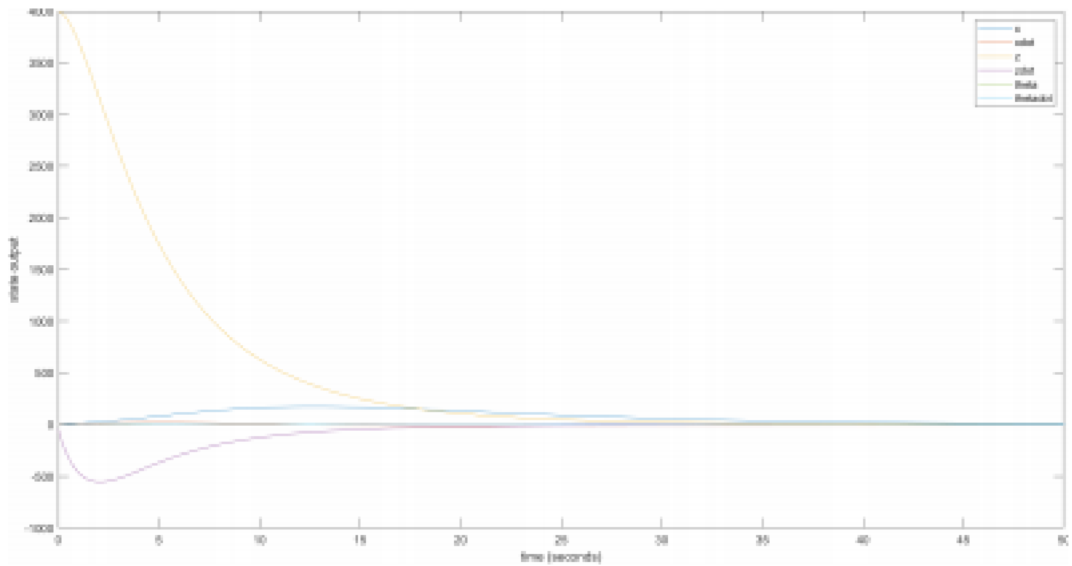

Figure 3: When a controller that applies state feedback is applied to the linearized model, the resulting system is indeed asymptotically stable. All values (x=blue, $\dot{x}$=red, z=yellow, $\dot{z}$=purple, θ=green, $\dot{\theta}$=light blue) converge to zero.

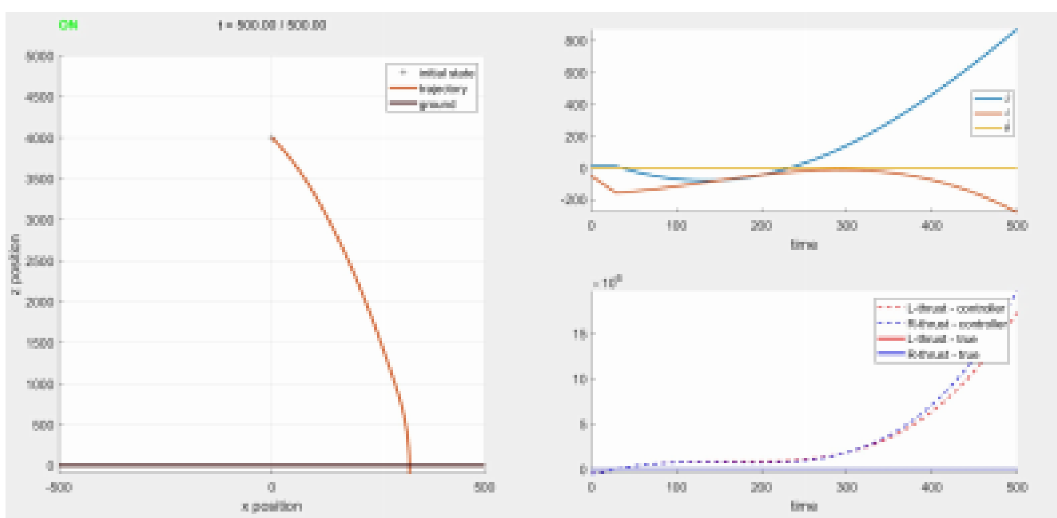

Figure 4: When the same controller that applies state feedback is applied to the non-linear model, the resulting system is not asymptotically stable.

3.6 Comparing Linearized Model and Nonlinear Simulation

For the zero input plots, the linearized model shows that z decreases linearly, i.e. $\dot{z}$ is constant. However, in the nonlinear simulation, $\dot{z}$ decreases over time, meaning z decreases non-linearly. This suggests that the linear model with zero input doesn't properly account for gravity, while the nonlinear simulation does. Also, the linearized model plot shows that x decreases non-linearly while in the nonlinear simulation $\dot{x}$ stays constant at its initial value.

For the state feedback plots, the linearized model shows that all values converge to 0. However, in the nonlinear simulation plots, no values converge to 0, besides $\dot{\theta}$. This difference is largely due to the fact that there is a maximum thrust value inputted into the simulation. Once the controller wants to use more than this maximum thrust value, the system will still use the maximum thrust value. As the vehicle gets further away from its equilibrium point, it wants to use more thrust but can’t. This cycle continues and the vehicle will never reach the desired equilibrium.