development of a Flight Control system for a 3D VTVL Rocket

Khelifi touhami achraf

In this paper I will discuss the dynamics, control, for a three degree- of-freedom rocket. This is a final project report for "Multivariable Linear Systems Control " module.

Nomenclature

$\boldsymbol{x}$	System State
$\boldsymbol{y}$	Observed State
$\boldsymbol{u}$	Control Input
$\boldsymbol{r}$	Reference State
A	Linearized Continuous-Time State Transition Matrix
B	Control Matrix
C	Observation Matrix

1 Introduction

In my graduate studies, I worked to develop a Flight control system for 3DOF VTVL Rocket(inspired by SpaceX). However, due to time restrictions, I decided to model the problem in 3DOF rather than the 6DOF.

2 Theory

Dynamics

The two forces considered in the dynamics of the vehicle are gravity and the thrust from the propulsion system. A free body diagram can be drawn and analyzed to derive equations of motion for the system.

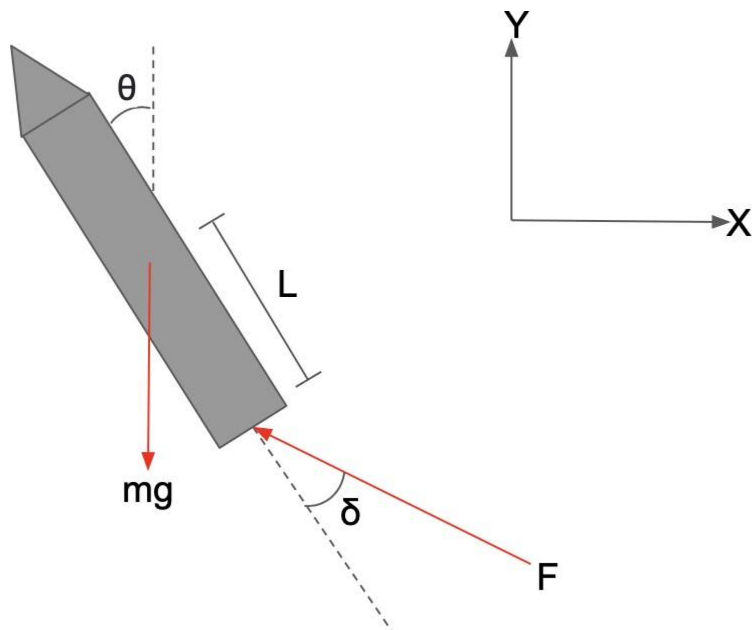

Figure 1: Free Body Diagram

m	vehicle mass
I	rotational inertia
g	gravitational acceleration
x	horizonatal displacement
y	vertical displacement
ϑ	angular displacement
F	magnitude of thrust
f	magnitude of thrust in excess of vehicle weight ($f = F - mg$)
δ	angle of thrust vector
L	moment arm

By summing forces in the x and y directions and summing moments, the equations of motion can be derived.

$$\ddot{x} = -\frac{Fsin(\vartheta + \delta)}{m} \quad (1)$$

$$\ddot{y} = \frac{Fcos(\vartheta + \delta)}{m} - g \quad (2)$$

$$\ddot{\vartheta} = -\frac{FLsin(\delta)}{I} \quad (3)$$

By introducing f, the equations can be rewritten

$$\ddot{x} = -g\sin(\vartheta + \delta) - \frac{f}{m}\sin(\vartheta + \delta) \tag{4}$$

$$\ddot{y} = \frac{f}{m}\cos(\vartheta + \delta) \tag{5}$$

$$\ddot{\vartheta} = -\frac{mgL}{I}\sin(\delta) - \frac{fL}{I}\sin(\delta) \tag{6}$$

These equations will now be linearized assuming ϑ,δ, and f are small.

$$\ddot{x} = -g\vartheta - g\delta \tag{7}$$

$$\ddot{y} = \frac{f}{m} \tag{8}$$

$$\ddot{\vartheta} = -\frac{mgL}{I}\delta \tag{9}$$

The dynamics can now be written in canonical state space form.

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$$

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -g \\ \frac{1}{m} & 0 \\ 0 & -\frac{mgL}{I} \end{pmatrix} \begin{pmatrix} f \\ \delta \end{pmatrix} \tag{10}$$

Controller Design

Firstly, I checked controllability of the system in Matlab. Intuitively it can be seen that with variable thrust and thrust angle, the system can reach any state in the state space. Sure enough, the controllability matrix is full rank, and thus the eigenvalues of the closed-loop system can be placed anywhere with a judicious choice of K_c. The control law I utilized takes the following form.

$$\boldsymbol{u} = K_c(\boldsymbol{r} - \boldsymbol{x}) \tag{11}$$

After some experimenting, I decided on eigenvalues of -1, -2, -3, -4, -5, and -6, as they resulted in good stability while still being conservative enough to keep the system in the linear regime.

3 Simulation

I made a Simulink model for the system's dynamics, control, and state estimation. In this simulation, I attempted to make the vehicle follow a simple ascent, translation, descent trajectory,I used the following vehicle parameters.

m	1 kg
I	0.002 kg m^2
L	10 cm

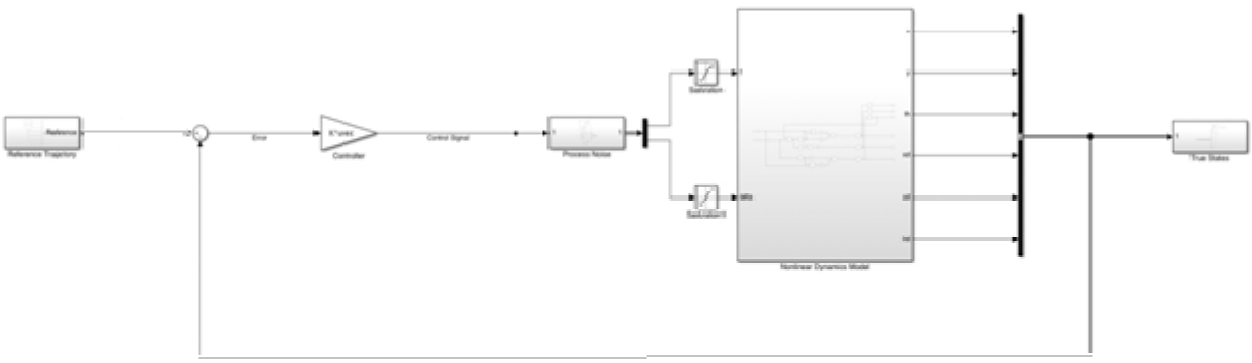

Figure 2: Full Block Diagram

I utilized two saturation blocks for the actuators. I limited the thrust vector angle to within 10 degrees.

Reference Trajectory

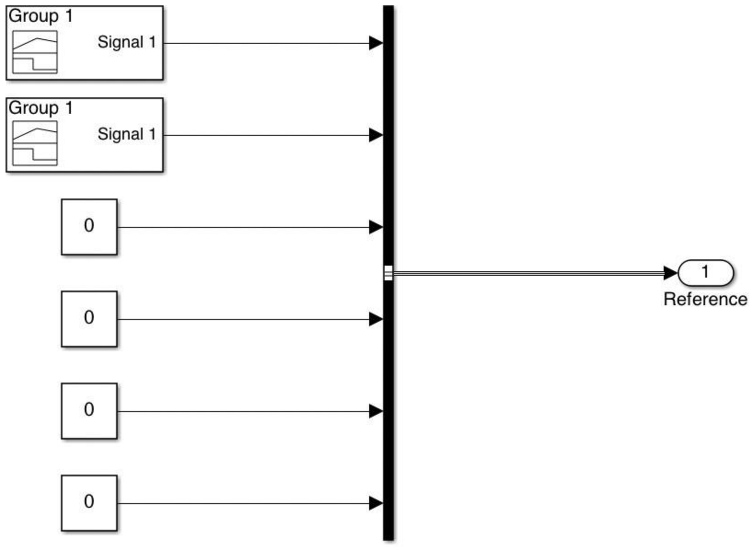

Figure 3: Reference Trajectory Block

The first two signals describe the reference *x* and *y* positions. I designed these signals to generate the ascent, translation, descent trajectory that I want. The rest of the reference signals are set to zero. The exact shape of the *x* and *y* reference trajectories .

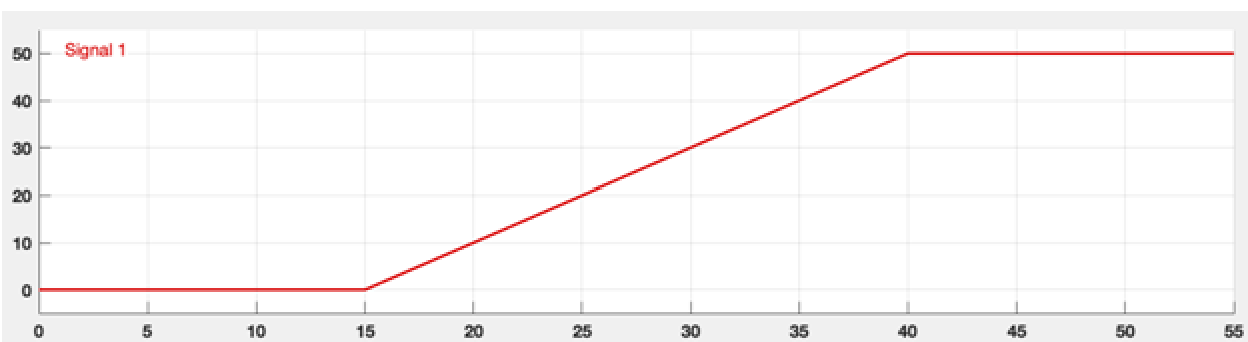

Figure 12: Reference X-Position

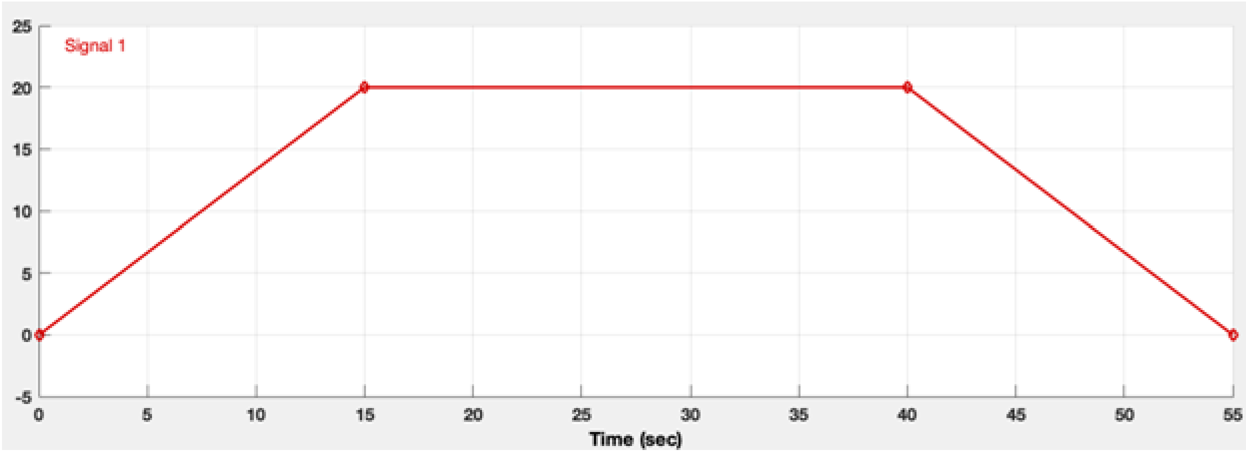

Figure 12: Reference Y-Position

Controller

The feedback controller inputs the state error and outputs the desired control commands. To design the controller, I used the place command in Matlab to place the poles at -1, -2, -3, -4, -5, and -6 for reasons mentioned in the Controller Design section. The controller gain matrix is shown below.

$$K_c = \begin{matrix} -10.66 & 13.12 & 45.94 & -13.68 & 7.59 & 4.81 \\ 0.0127 & -0.0015 & -0.1245 & 0.0226 & -3.65e-4 & -0.0278 \end{matrix}$$

Dynamics Model

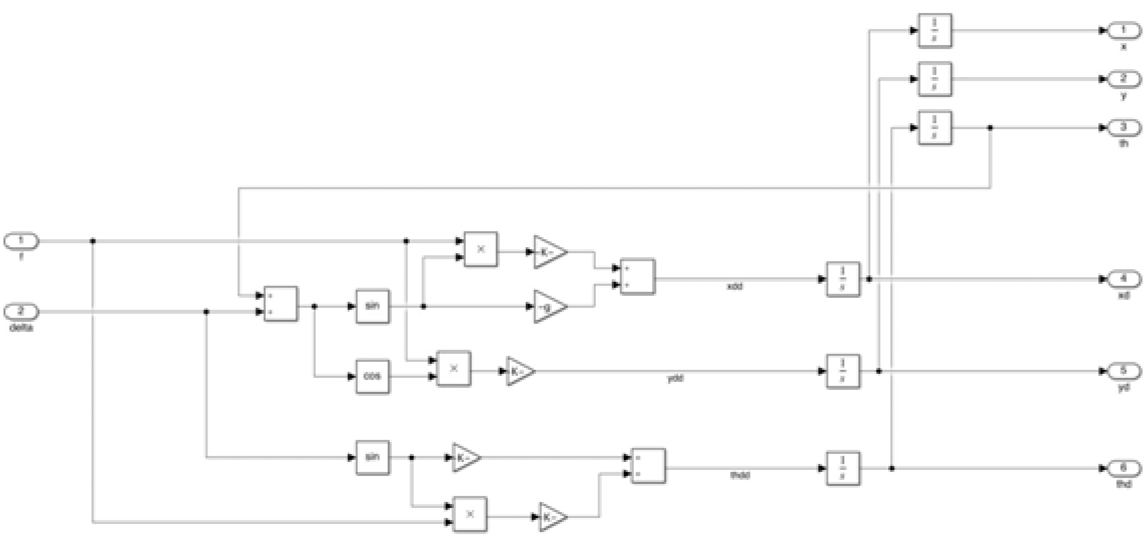

Figure 4: Nonlinear Dynamics Block

When I designed this block I used the full nonlinear equations of motion (Equations 4, 5, and 6) in order to see how effective the controller based on the linearized model would work to control the full nonlinear plant. The dynamics block does all computations in continuous time to mimic real dynamics.

4 Results

In this section I will present plots that show the performance of the trajectory tracking.

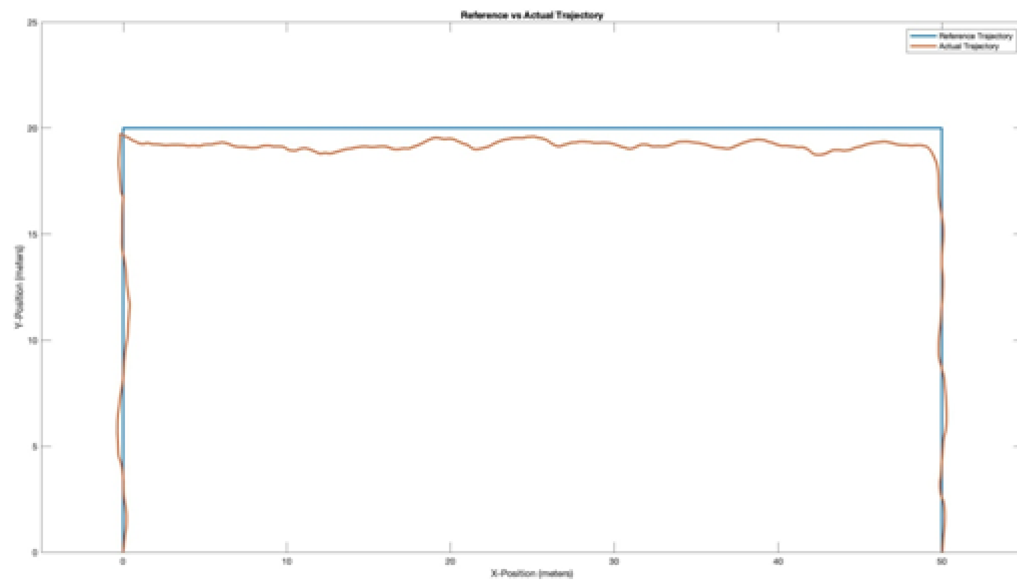

Figure 10: Reference vs Actual Trajectory

As can be seem from the above plot, the vehicle does a great job of tracking to within 35 centimeter on ascent and descent. The main problem that can be seen is that the vehicle has about 5% steady-state error in the y-direction. This is somewhat expected, given that this controller did not include an integrator or integrated states.

5 Further Work

This project was a great learning experience for me: it was the first time I learned about aerospace systems dynamics and knowing more about GNC/AOCS, I think that some natural next steps for me would be to learn convex optimization, trajectory optimization and trajectory tracking.

Perhaps I'll try to design a 6DOF model of the system, FDIR (faults diagnosis and detection) system to detect the vehicles sensors and actuators faults. also try to improve the controller performance by redesign the controller using advanced control methods such as SMC (sliding mode control) ,optimal and nonlinear control,etc.