Open Loop Simulation of a Thrust Vector Controlled Rocket, (Kalman-TVC)

Course Project Part 2
AESE-404
Joshua Kraus (jjk226@lehigh.edu)

1. Vehicle and Purpose

The "Kalman" thrust vector control (TVC) rocket is a finless solid propellant rocket running on an AeroTech I49N-P motor. The vehicle is approximately 1.44 m long and has a liftoff mass of approximately 1.73 kg. It is steered only by gimballing the motor about a pivot that sits $l = 0.46\,\mathrm{m}$ behind the center of gravity (CG).

The purpose of the flight is to demonstrate a fully autonomous controlled ascent using pure thrust vector control and internal reaction systems. The demonstrator vehicle gives us a comprehensive testbed for developing code that can be used to further the research and development of the Lehigh University rocketry team's liquid bipropellant rocket line.

The purpose of this report is to cover the open loop simulation of the Kalman TVC rocket. The thrust curve and the gimbal angles are chosen in advance, and the response of the rocket does not feed back to change any input. Sensors, estimation, and feedback are not within the scope of this report.

2. Implemented Equations of Motion

2.1 Frames and State

Three coordinate systems are used (Figure 1). The ground frame has its origin at the launch pad and its axes point North, East, and Down (NED). The body frame is fixed to the rocket with its origin at the CG. The x_b axis points out of the nose, y_b points to the right, and z_b completes the right handed set. The orientation of the body frame is given by the Euler angles φ, θ, ψ, which are three successive turns that carry one set of axes into another (Weisstein, n.d.). The turns are roll, pitch, and yaw in the 3-2-1 sequence used in flight dynamics (Stevens et al., 2016), which means they are made about the 3rd, 2nd, and 1st axes in that order. They are measured from a launch reference. The launch reference sits at the pad and uses the same directions as the ground frame, but its axes are listed as Up, East, and North, so all three angles are zero when the nose points straight up. The 12 states are the position (P_N, P_E, P_D) of the CG in the ground frame, the velocity (u, v, w) of the CG in body axes, the Euler angles (φ, θ, ψ), and the body angular rates (p, q, r).

Equation (8) contains $\tan\theta$ and $1/\cos\theta$, and both become infinite at $\theta = 90^\circ$. At that pitch the roll axis and the yaw axis line up, so the roll and yaw angles can no longer be told apart. Aircraft measure pitch from the horizon, which would put a rocket standing on the pad at exactly 90°, and the equations would fail before the simulation began. Measuring from the launch reference puts the pad at $\theta = 0$ and moves the trouble spot to $\theta = \pm 90^\circ$, which now means the nose is level with the ground. That is horizontal flight, which for this vehicle is already an extreme failure and far outside the intended flight. Regarding the Part 1 equations, nothing else changes except for the weight (3) and the position rates (9), which are written in the launch frame reference. As a safety measure the code keeps $\cos\theta$ from reaching exactly zero, and the plot script prints a warning when the pitch angle passes 80°.

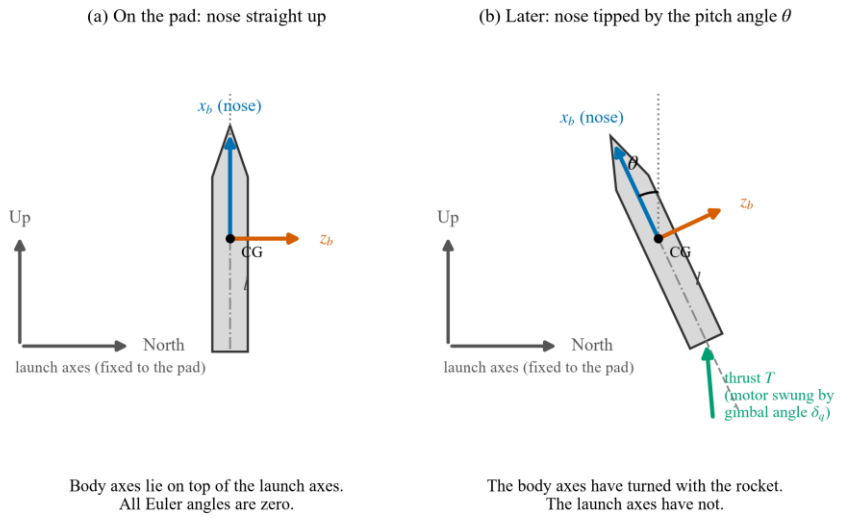

Figure 1. Body axes (fixed to the rocket) and launch axes (fixed to the pad), drawn in the pitch plane. The Euler angles describe how far the body axes have turned from the launch axes.

2.2 Forces and Moments

The thrust magnitude $T(t)$ is taken from the published AeroTech I49N thrust curve (ThrustCurve.org, n.d.), which is certification data published in 2009. The thrust vector control assembly (TVCA) swings the thrust in the pitch plane by δ_q and in the yaw plane by δ_r. Each angle is limited to $\pm 5^{\circ}$, a value set by the measured limits of the mechanism (Kraus, 2026b). Positive angles tilt the thrust toward $+z_b$ and $+y_b$. In body axes the thrust force, and its moment about the CG through the lever arm l, are

$$\mathbf{F}_T = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = T \begin{bmatrix} \cos\delta_q \cos\delta_r \\ \cos\delta_q \sin\delta_r \\ \sin\delta_q \end{bmatrix} \tag{1}$$

$$M_x = M_{\text{roll}} = 0, \qquad M_y = l\,T \sin\delta_q, \qquad M_z = -\,l\,T \cos\delta_q \sin\delta_r \tag{2}$$

On the pad $+y_b$ points East and $+z_b$ points North. A positive pitch gimbal therefore pushes the tail North and tips the nose South, and a positive yaw gimbal pushes the tail East and tips the nose West.

Weight and aerodynamic drag complete the forces. The weight resolved into body axes depends on the Euler angles. Drag opposes the velocity, whose size is $V = \sqrt{u^2 + v^2 + w^2}$.

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = -g \begin{bmatrix} \cos\theta \cos\psi \\ \sin\varphi \sin\theta \cos\psi - \cos\varphi \sin\psi \\ \cos\varphi \sin\theta \cos\psi + \sin\varphi \sin\psi \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = -\frac{1}{2}\,\rho\,V S\,C_d(V) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4}$$

The air density $\rho = 1.225\,\text{kg/m}^3$ is kept constant because the flight stays below 1 km. This assumption was tested, see Appendix B.

The reference area $S = 70.88\,\text{cm}^2$ is the circular area at the largest diameter of the vehicle, $D = 9.5\,\text{cm}$ at the nose and avionics section, so $S = \pi D^2/4$. The drag coefficient $C_d(V)$ comes from the drag table of the OpenRocket simulation (Kraus, 2026a), shown in Table 1. It falls from 0.35 to 0.30 as the speed rises from 7 to 33 m/s. It should be noted that OpenRocket is not designed for finless flight. It stopped the run when the stage began to tumble under thrust (Figure 2), so the table ends at 33 m/s. Thus the drag coefficient is held at its last value for higher speeds, which is an assumption. The OpenRocket data still gave valuable insight into an approximation of the drag.

Table 1. Drag coefficient data from the OpenRocket simulation (Kraus, 2026a). The values are listed at selected speeds out of 34 rows.

Speed (m/s)	Mach	Cd total	Cd friction	Cd pressure	Cd base
7.1	0.022	0.350	0.229	0.099	0.021
10.1	0.030	0.335	0.214	0.099	0.021
15.7	0.046	0.321	0.200	0.099	0.021
20.1	0.059	0.315	0.194	0.099	0.021
25.6	0.075	0.309	0.188	0.099	0.021
30.0	0.088	0.306	0.185	0.100	0.021
33.2	0.097	0.304	0.182	0.100	0.021



Figure 2. The OpenRocket model of the Kalman TVC rocket (Kraus, 2026a). The Barrowman estimate puts the center of pressure at −389 cm, which is ahead of the nose tip and not physically meaningful. OpenRocket aborted the run when the finless stage began to tumble under thrust.

The aerodynamics have two limits. First, the reference area is held constant. When the vehicle flies at an angle of attack the frontal and wetted area change, and the drag should change with them, so (4) is only accurate for flight with the nose pointed along the velocity. Finding the frontal and wetted area at different angles of attack, and using it to correct the drag, is within the scope of the overall project. The next step is to develop an ANSYS CFD model to determine the drag coefficient at different angles of attack.

Second, the location of the center of pressure (CP) is not known. The CP is the point where the total aerodynamic force acts. A stable rocket has its CP behind its CG, and fins are what provide that stability (Bigwood et al., 2021, pp. 46 to 47). This vehicle has no fins, so its CP is expected to sit ahead of the CG. At a small angle of attack (the angle between the nose axis and the direction of travel) the aerodynamic force then makes a moment about the CG that tips the rocket further over. This moment enters the rotational equations through M_{aero} in (7) and is given by

$$M_{\text{aero},y} = -\frac{1}{2}\,\rho\,V S D\,C_{m\alpha}\,w, \qquad M_{\text{aero},z} = \frac{1}{2}\,\rho\,V S D\,C_{m\alpha}\,v \tag{5}$$

where $C_{m\alpha}$ is the moment slope. Once the CP is known from the ANSYS simulation, the moment slope follows from the normal force slope $C_{N\alpha}$ and the positions of the CG and CP measured from the nose tip. It is negative when the CP sits ahead of the CG.

$$C_{m\alpha} = -\,C_{N\alpha}\,\frac{x_{cg} - x_{cp}}{D}$$

A placeholder value of $C_{m\alpha} = -18.7$ per radian is stored in the initialization file, but it is not used in any run in this report because the aerodynamic moment is switched off. It is an assumption, taken from a straight line fit to the OpenRocket data (Kraus, 2026a). That run only covered angles of attack from 6.7 to 17.4 degrees before it aborted, so the fit is rough, and it will be replaced by the CFD result.

OpenRocket finds the CP with the Barrowman equations (Niskanen, 2013, pp. 21 to 23). The method assumes the angle of attack is very close to zero, treats a cylindrical body tube as adding no normal force, and assumes the CP is mostly set by the fins. This vehicle has no fins, and the OpenRocket run saw angles of attack of 6.7 to 17.4 degrees, so the CP estimate is not usable, as the −389 cm in Figure 2 shows. An ANSYS CFD model will therefore be used to find the CP of the vehicle, assumed symmetric, at different angles of attack. A similar approach was used by Bigwood et al. (2021, p. 105), who ran Ansys Fluent at angles of attack from 0 to 70 degrees to get the center of pressure and the aerodynamic forces on the descent stage of their rocket. Until those results exist the aerodynamic moment is switched off in the code with `aero_moment_enable = 0`. It is included in the equations as a placeholder while the coefficients are determined as part of the scope of this project.

2.3 Equations of Motion

The equations of motion are the standard rigid body equations written in the body axes defined above, as in Stevens et al. (2016), and they are the same equations written in Part 1. Bigwood et al. (2021, p. 154) use the same split into a center of mass model and an attitude model. Newton's second law written in the rotating body axes gives the translational equations (6), where m is the mass. Euler's equations give the rotational equations (7), where I_x, I_y, and I_z are the principal moments of inertia. The Euler angle kinematics (8) turn the body angular rates into the rates of change of the angles, and (9) turns the body velocity into velocity over the ground.

$$\begin{aligned} \dot{u} &= \frac{F_x + D_x}{m} + g_x - q\,w + r\,v \\ \dot{v} &= \frac{F_y + D_y}{m} + g_y - r\,u + p\,w \\ \dot{w} &= \frac{F_z + D_z}{m} + g_z - p\,v + q\,u \end{aligned} \tag{6}$$

$$\begin{aligned} \dot{p} &= \frac{M_x - (I_z - I_y)\,q\,r}{I_x} \\ \dot{q} &= \frac{M_y + M_{\mathrm{aero},y} - (I_x - I_z)\,p\,r}{I_y} \\ \dot{r} &= \frac{M_z + M_{\mathrm{aero},z} - (I_y - I_x)\,p\,q}{I_z} \end{aligned} \tag{7}$$

$$\begin{aligned} \dot{\varphi} &= p + (q \sin\varphi + r \cos\varphi) \tan\theta \\ \dot{\theta} &= q \cos\varphi - r \sin\varphi \\ \dot{\psi} &= \frac{q \sin\varphi + r \cos\varphi}{\cos\theta} \end{aligned} \tag{8}$$

$$\begin{aligned} \dot{P}_N &= u(-\sin\theta) + v \sin\varphi \cos\theta + w \cos\varphi \cos\theta \\ \dot{P}_E &= u \cos\theta \sin\psi + v(\sin\varphi \sin\theta \sin\psi + \cos\varphi \cos\psi) \\ &\quad + w(\cos\varphi \sin\theta \sin\psi - \sin\varphi \cos\psi) \\ \dot{P}_D &= -u \cos\theta \cos\psi - v(\sin\varphi \sin\theta \cos\psi - \cos\varphi \sin\psi) \\ &\quad - w(\cos\varphi \sin\theta \cos\psi + \sin\varphi \sin\psi) \end{aligned} \tag{9}$$

The equations are solved in Simulink with a fixed step fourth order Runge-Kutta method (ode4) at 0.01 s. The scaffold provided for this course used an Euler method (ode1), but ode4 was chosen for accuracy. As a rule of thumb, a fourth order method is more accurate than Euler for the same step, and the equations here are nonlinear with the states depending on each other.

The equations exist in one MATLAB Function block inside the Vehicle Dynamics subsystem, with four integrators that turn each rate back into a state (Figures 3 and 4). The translational equations (6) in the code are

```
u_dot = (F_input(1) + Dx)/mass + gx - q*w + r*v;
v_dot = (F_input(2) + Dy)/mass + gy - r*u + p*w;
w_dot = (F_input(3) + Dz)/mass + gz - p*v + q*u;
```

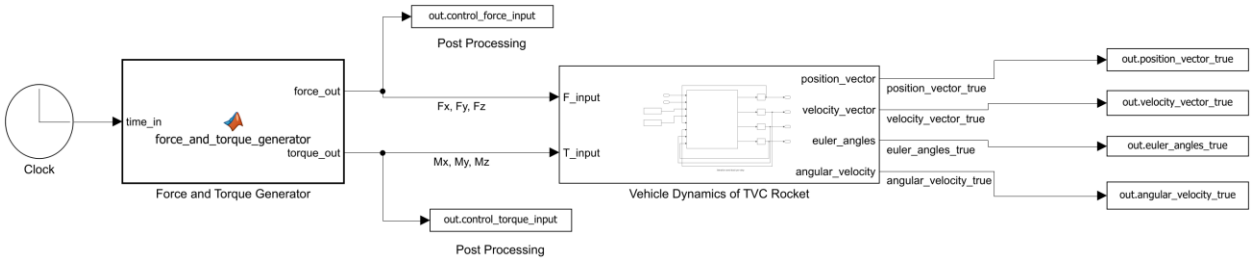

Figure 3. Top level of the Simulink model. The clock feeds the Force and Torque Generator, and its force and torque feed the Vehicle Dynamics block. The four outputs are saved for plotting.

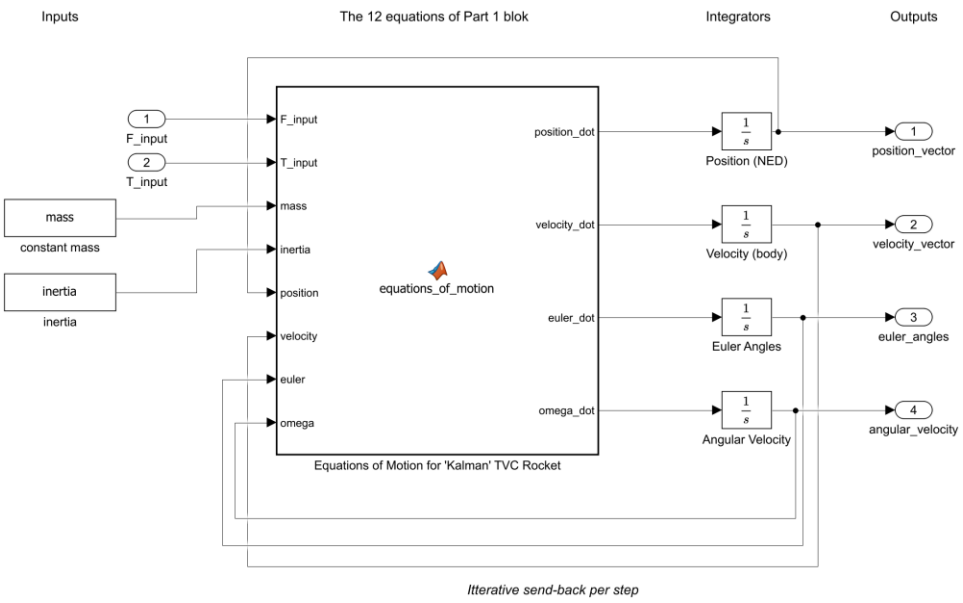

Figure 4. Inside the Vehicle Dynamics block. The equations of motion block returns the rate of change of each state, each integrator turns a rate into a state, and the states are sent back to the block at every step.

2.4 Model Parameters

The parameters in Table 2 are defined at the top of the initialization file.

Table 2. Model parameters.

Quantity	Value	Source
Liftoff mass, m	1.7293 kg	Measured (weighed at about 1.7 kg) and OpenRocket
CG to gimbal pivot, l	0.46 m	Measured at pad weight
Roll inertia, I_x	0.002 kg·m²	OpenRocket
Pitch and yaw inertia, $I_y = I_z$	0.487 kg·m²	OpenRocket
Gimbal limit	±5°	Measured (4.97°)
Thrust curve	7.94 s burn, 384 N·s, 63.4 N peak	ThrustCurve.org
Drag coefficient, C_d	0.35 to 0.30	OpenRocket
Reference area, S	70.88 cm²	OpenRocket
Reference diameter, D	9.5 cm (largest diameter)	OpenRocket
Gravity, g	9.80665 m/s²	Standard value
Air density, ρ	1.225 kg/m³	Standard sea level value

Initial conditions are set in the initialization file. The rocket starts on the pad at position zero, at rest with velocity zero, pointing straight up with the Euler angles zero, and with no rotation so the angular rates are zero. These are the variables `x0`, `v0`, `euler_angles0`, and `angular_velocity0`.

2.5 Modeling Assumptions

Table 3. Modeling assumptions and their justifications.

Assumption	Justification
Rigid body, flat non rotating Earth, constant gravity	The flight stays below 1 km, so the curvature and rotation of the Earth and the change in gravity are negligible. Bigwood et al. (2021, p. 154) also use constant gravity and neglect the rotation of the Earth.

Assumption	Justification
Constant mass, CG, and inertia	The handout does not require time varying parameters. The burn removes 0.205 kg, 12 percent of the liftoff mass. The mass study in Section 4 shows the size of this effect.
Symmetric vehicle with $I_y = I_z$ and no products of inertia	The airframe is a body of revolution and is effectively symmetric. Any small asymmetry is assumed not to change the moments of inertia in a significant way.
No roll torque, $M_x = 0$	The vehicle is modeled as perfectly symmetric with no thrust misalignment, so nothing makes it spin up. Roll control is covered in Section 2.6 and Appendix A.
Gimbal angle is a direct input	The handout excludes actuator models. Thrust acts along the motor axis through the pivot and the motion of the motor's own mass is neglected. A simple moment arm is accurate for small gimbal angles (Stepp, 2024) and the range here is within 5 degrees.
Thrust follows the published curve and does not change with altitude	The certification curve is the best data available for this motor.
Drag from a speed table with a constant reference area	Drag opposes the velocity with no wind. The drag table covers speeds up to 33 m/s and its last value is held above that. The area does not change with angle of attack, which is inaccurate and is listed as future work.
Constant air density	The flight stays below 1 km, where the density falls by 9 percent at most (Appendix B).
No center of pressure and no aerodynamic moment	This is the largest assumption in the model. The CP of a finless body is unknown and the Barrowman estimate is not usable, so the moment is switched off until the ANSYS CFD results are available. A destabilizing moment would make any small tilt grow.
No wind, ground, sensors, disturbances, or feedback	The handout excludes these items. The simulation has no ground, so the plot script cuts the plots when the rocket returns to the height of the pad. The runs in this report stop at 20 s, before that happens.

2.6 Hand Calculation for Roll Control

As noted in Part 1, the model assumes no roll torque, but the real vehicle has misalignments such as thrust misalignment, the bars at the base of the TVC assembly, and the camera bumps on the nose, which will likely spin it up. Roll will be controlled by a reaction disc inside the airframe that is spun against the vehicle. The hand calculation that sizes the disc is in Appendix A.

3. How the Inputs Work

3.1 The Inputs

Two kinds of input drive the simulation. The thrust curve is a force that changes with time and is read from a data file. The two gimbal angles are lists of time and angle points in the initialization file. At every 0.01 s step the model reads the lists, draws a straight line and interpolates between the points, and limits the range to $\pm 5^{\circ}$.

The test cases in the initialization file fill the lists. A hold keeps a constant angle between a start time and an end time. A sine swings the angle plus and minus its size over one period. A ramp rises in a straight line to its size over one period and then holds until the end time. Outside the window the gimbal sits at zero, which is straight flight. The Force and Torque Generator turns these inputs into a force (1) and a torque (2). The key lines of its code are

```
T  = interp1(thrust_t, thrust_N, time_in, 'linear', 0);   % thrust from the curve
dq = interp1(gimbal_pitch_time, gimbal_pitch_angle, time_in);   % pitch gimbal
dr = interp1(gimbal_yaw_time, gimbal_yaw_angle, time_in);       % yaw gimbal
dq = min(max(dq, -gimbal_limit), gimbal_limit);   % mechanism limit, same for dr
force_out  = T * [cos(dq)*cos(dr); cos(dq)*sin(dr); sin(dq)];   % equation (1)
torque_out = [M_roll; l_arm*force_out(3); -l_arm*force_out(2)]; % equation (2)
```

A similar thrust vector controlled rocket model is available from MathWorks (2020) and is discussed in a video by BPS.space (2020). It also uses a gimbal angle to steer the thrust. In this project the equations of motion are written out in full from Part 1.

3.2 From Inputs to Motion

Figure 5 shows the whole chain, and the steps below follow it from left to right.

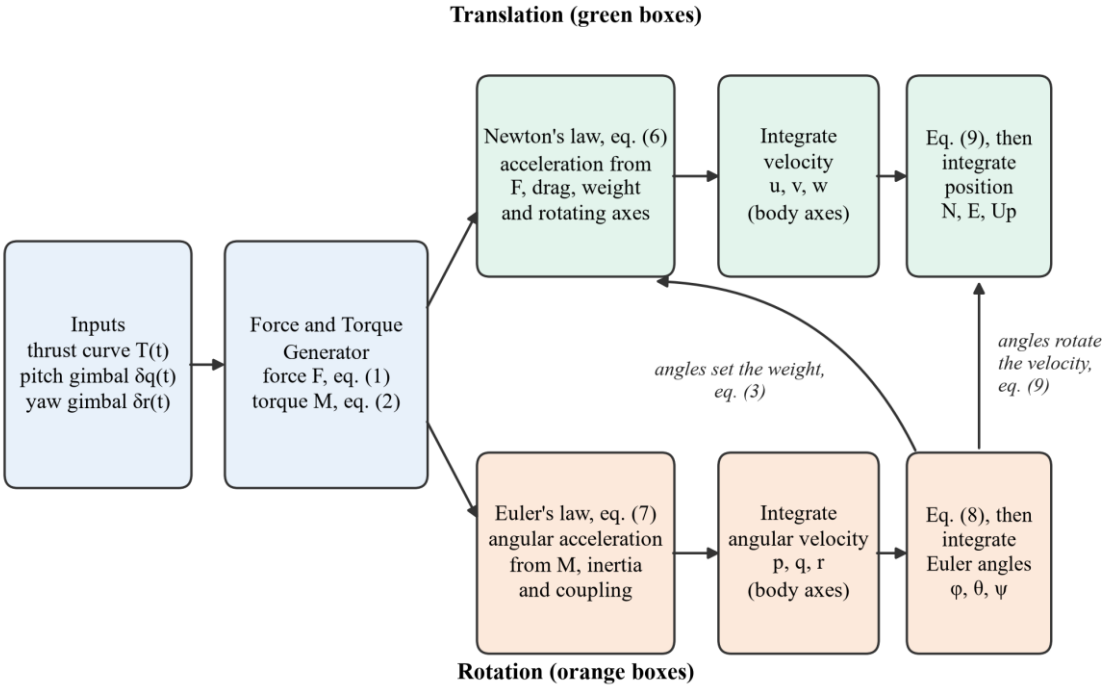

Figure 5. How the inputs become motion. Blue boxes are the inputs, green boxes are the translation equations, and orange boxes are the rotation equations. An integrator adds up a rate over time to give the state. This figure was generated with Claude AI and placed in Microsoft Word.

- Step 1. The generator reads the time and looks up the thrust T and the two gimbal angles. Thrust always points along the motor axis, so the gimbal angles only tilt the force away from the nose axis (1).
- Step 2. The same block finds the torque as the lever arm crossed with the force (2). The force acts at the pivot, a distance l behind the CG, so any sideways piece of the force has leverage about the CG.
- Step 3. For translation, force divided by mass is acceleration (6). Drag, weight, and the extra terms that come from using rotating axes ($-q\,w + r\,v$ in the u row and the matching terms in the other rows) are added in. An integrator adds the acceleration up over time to give the velocity u, v, w.
- Step 4. For rotation, torque divided by inertia is angular acceleration (7). An integrator gives the angular velocity p, q, r.
- Step 5. The angular velocity is turned into the rates of change of the Euler angles by (8). An integrator gives the Euler angles, which determine how the rocket is pointed.
- Step 6. The Euler angles feed back in two places. They decide how much of the weight lies along each body axis (3), and they rotate the body velocity into North, East, and Down velocity (9). A last integrator gives the position.

The model repeats these steps every 0.01 s. At each step it uses the current states to find the rates, and the integrators add a small piece of each rate to its state.

3.3 The Two Frames

The thrust is fixed to the motor, so the force, the torque, and the angular rates are written in body axes. Position is measured from the pad, thus it is written in the ground frame. Velocity is kept in body axes as that makes the force balance (6) simpler. The thrust and the drag already point along the body axes, so only the weight has to be turned into them using the Euler angles. The Euler angles connect the two frames, and they enter the equations in only two places, the weight (3) and the position rates (9).

3.4 Hand Calc of a Gimbal Command

Take a thrust of $T = 60\,\text{N}$ and a pitch gimbal of 0.1°. The sideways force, the torque about the CG, and the pitch acceleration are

$$F_z = T \sin \delta_q = 60 \sin 0.1^\circ = 0.105\,\text{N}$$

$$M_y = l\,F_z = (0.46)(0.105) = 0.048\,\text{N}\cdot\text{m}$$

$$\dot{q} = \frac{M_y}{I_y} = \frac{0.048}{0.487} = 0.099\,\text{rad/s}^2 \approx 5.7\,\text{deg/s}^2$$

When the gimbal returns to zero, $\dot{q}$ returns to zero but q does not. Nothing in the model damps a rotation, so the rocket keeps turning at whatever rate it had.

4. Simulation Results

4.1 Test Matrix and What Was Entered

Every result in this section comes from the same model. The test box at the top of the initialization file was changed between runs, and the matrix in Table 4 describes each case. All runs were stopped at `t_final = 20` s. The mass study used `test_case = 0` and `mass_scale` values of 0.88, 1.0, and 1.2. As an example, these are the lines entered for the demo case in Section 4.4.

```
test_case       = 2;
pitch_amp_deg   = 0.1;
yaw_amp_deg     = 0.05;
gimbal_start_s  = 2;
gimbal_end_s    = 5;
gimbal_period_s = 3;
t_final         = 20;
```

Table 4. Lines entered in the initialization file for each case.

Case	`test_case`	`pitch_amp_deg`	`yaw_amp_deg`	`gimbal_start_s`	`gimbal_end_s`	`gimbal_period_s`
No gimbal input	0	n/a	n/a	n/a	n/a	n/a
Hold	1	0.04	0	2	3	n/a
Sine	2	0.1	0	2	5	3
Ramp	3	0.04	0	2	5	3
Demo case	2	0.1	0.05	2	5	3
Gimbal size study	1	0.02, 0.04, 0.06	0	2	3	n/a

4.2 Straight Flight With No Gimbal Input

With no gimbal angle the thrust points along the nose axis through the CG, so there is no torque and the rocket does not rotate. The motion is then just force balance along a vertical line. The motor produces up to 63 N and 48 N on average against a weight of 17 N, an average thrust to weight ratio of 2.85. Speed peaks at 112 m/s at 7.1 s, just before burnout at 7.94 s. At that moment the thrust has fallen to about 33 N, which equals the drag plus the weight, so the speed stops increasing. After that the thrust keeps falling and the rocket slows down. It coasts up to an apogee of 988 m at 16.5 s. North and East stay at zero and all the angular rates are zero. This is the middle curve in Figure 13.

4.3 Hold, Sine, and Ramp

Figure 6 shows the three gimbal test cases from Table 4. Each curve is a separate simulation run, overlaid on the same axes. The top plot is the pitch gimbal angle that was entered. The bottom plot is the pitch angular velocity q that the rocket ends up with. In these runs $p = 0$ and $I_y = I_z$, so the cross terms in (7), the ones that multiply two angular rates, are exactly zero and the pitch acceleration is $\dot{q} = M_y / I_y$. From (2) the torque is $M_y = l\,T \sin \delta_q$. So the angular velocity rises while the gimbal is off center and stays constant once the gimbal returns to zero.

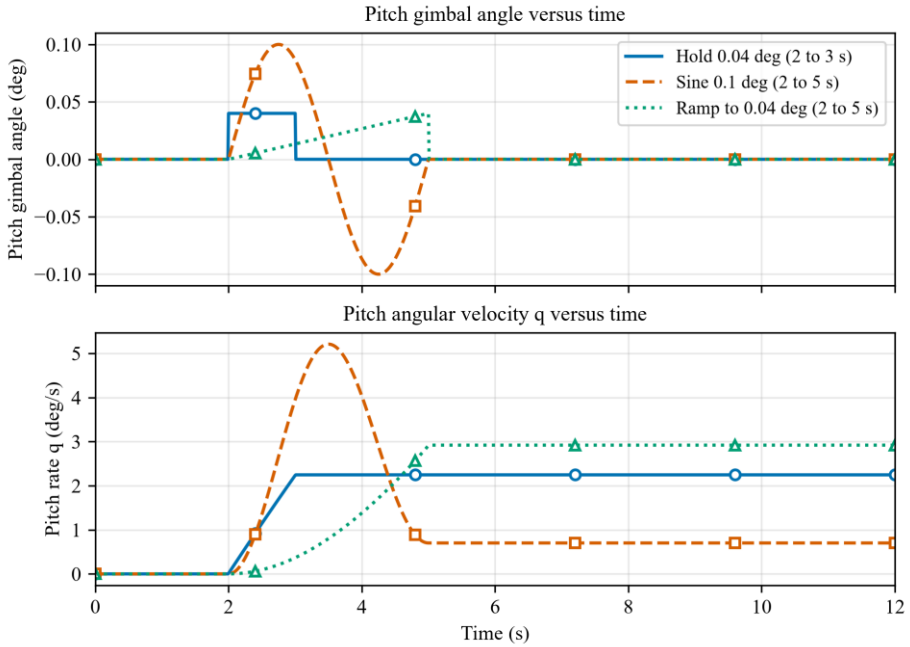

Figure 6. The three gimbal test cases (top) and the pitch angular velocity each one produces (bottom). Each curve is a separate run overlaid on the same axes. The rate does not return to zero once the command ends.

- The hold (solid blue line with circles) is 0.04° from 2 to 3 s. The torque is constant, so q rises in a straight line to 2.25 deg/s at 3 s and then stays there.
- The sine (dashed orange line with squares) swings from +0.1° to −0.1° between 2 and 5 s. The rate climbs while the angle is positive and peaks at 5.2 deg/s at 3.5 s when the angle crosses zero. It then falls while the angle is negative and settles at 0.70 deg/s. The two halves nearly cancel. A small rate is left over because the thrust is lower in the negative half, so the negative torque is weaker.
- The ramp (dotted green line with triangles) grows steadily to 0.04° at 5 s. The rate climbs in a curve and reaches 2.92 deg/s at 5 s, which is where it stays.

A constant angular velocity means the tilt keeps growing. After the hold the rocket is tipped 39 degrees from vertical by 20 s. The tilted thrust then pushes the rocket sideways, so the flight path bends.

4.4 Demo Case

The demo case applies a sine of 0.1° in pitch and 0.05° in yaw from 2 to 5 s (Table 4). Figures 7 to 11 are the plots of this run. Figure 7 shows the position. North and East both go negative, so the rocket drifts South and West and ends 157 m South and 78 m West of the pad at 20 s. Height peaks at 977 m at 16.4 s.

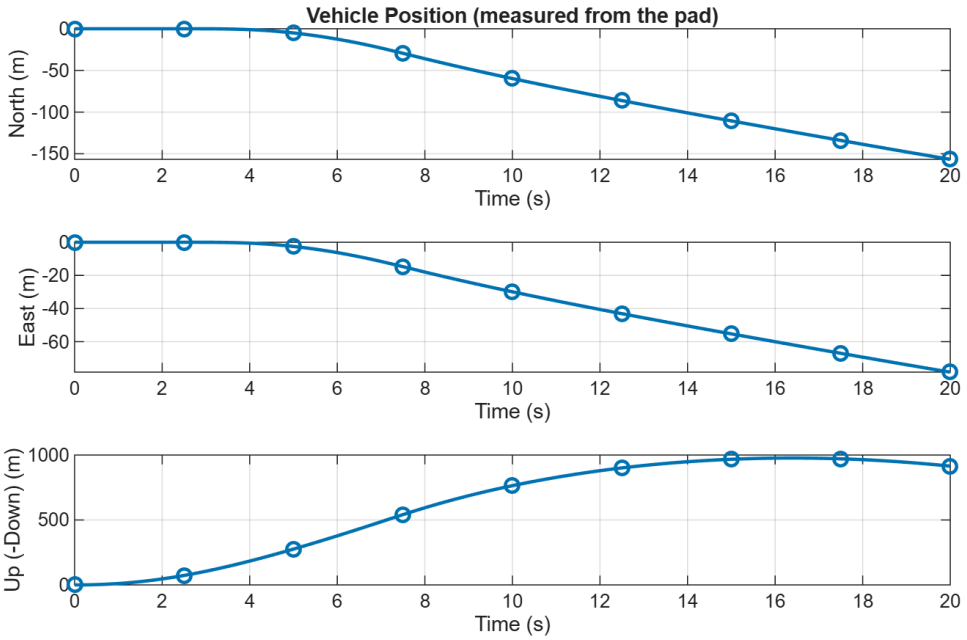

Figure 7. Position of the CG in the ground frame for the demo case. North and East are negative, so the rocket drifts South and West.

Figure 8 shows the velocity in body axes. The component along the nose, u, peaks at 111 m/s at 7.1 s. The sideways components v and w start to grow when the gimbal moves at 2 s.

Figure 9 shows the Euler angles. At 20 s pitch is 18.8 degrees, yaw is −9.8 degrees, and roll is −1.6 degrees.

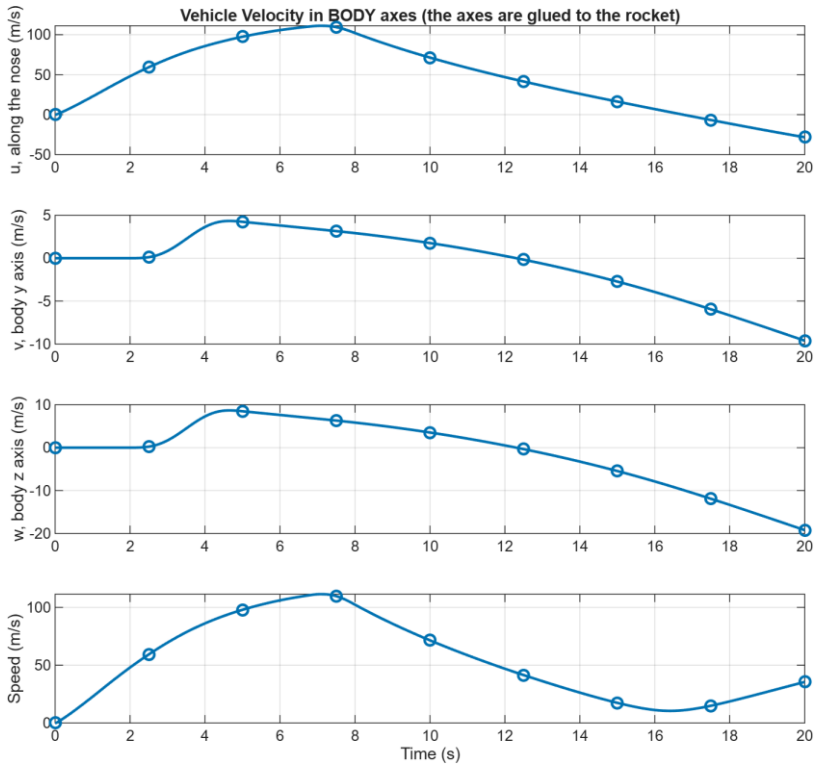

Figure 8. Velocity of the CG in body axes for the demo case. The bottom plot is the speed, which is the size of the velocity vector.

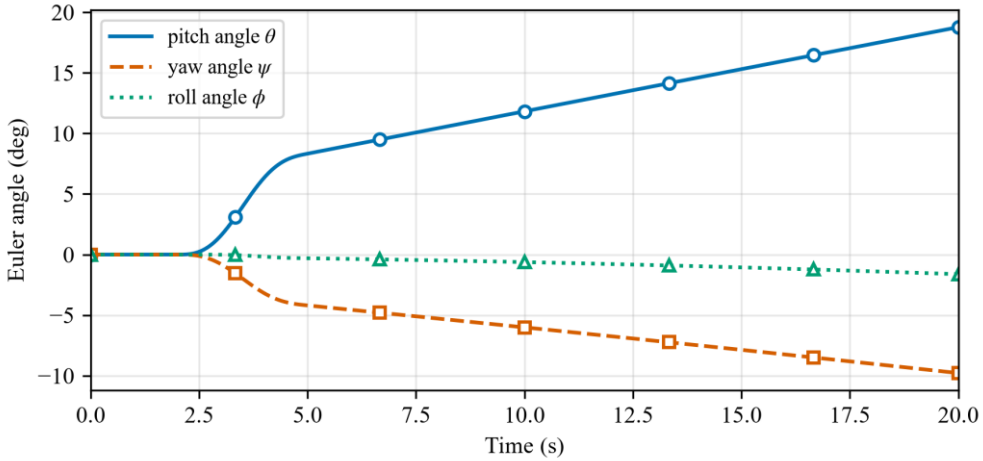

Figure 9. Euler angles for the demo case. These are the orientation states and they come from the same run as Figures 7, 8, 10, and 11.

Figure 10 has four plots. The first three are inputs to the dynamics block and only the fourth is a result. The gimbal angle, the force, and the torque are three views of the same command at different steps of the chain in Figure 5, and none of them is a separate command.

- Gimbal angles (top left). This is the command and the only thing we told the rocket to do. The pitch gimbal swings through ±0.1° and the yaw gimbal through ±0.05° between 2 and 5 s.
- Thrust force (top right). F_x follows the thrust curve and falls to zero at burnout. F_y and F_z are the tiny sideways pieces of the force, never more than 0.11 N, so they sit on top of zero at this scale.

- Torque about the CG (bottom left). The generator calculates it from the force and the lever arm, $M_y = lF_z$ and $M_z = -lF_y$, so it has the same shape as the gimbal angles. M_x is zero.
- Angular velocity (bottom right). This is the response of the rocket and the first thing the equations of motion calculate from the torque. The pitch and yaw rates rise while the torque acts and stay constant afterward, and the roll rate p stays at zero.

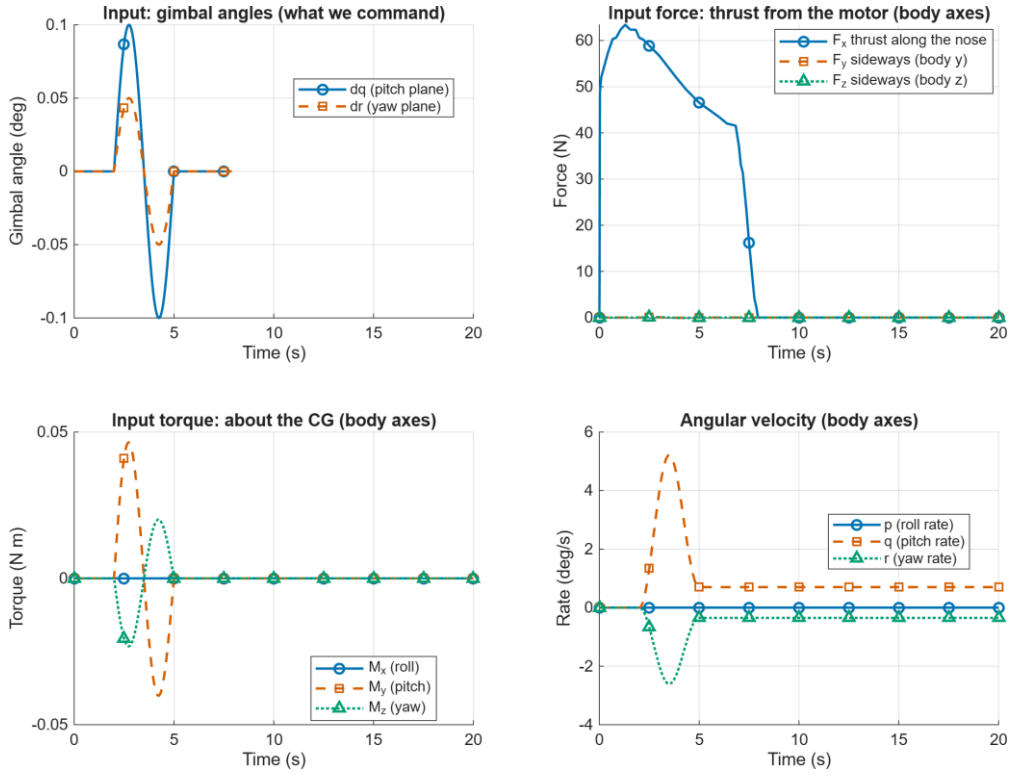

Figure 10. Inputs and angular velocity for the demo case. The first three plots are inputs and the fourth is the response of the rocket.

Figure 11 shows the same flight as a 3-D path. The path bends away from vertical toward South and West as the nose tips over.

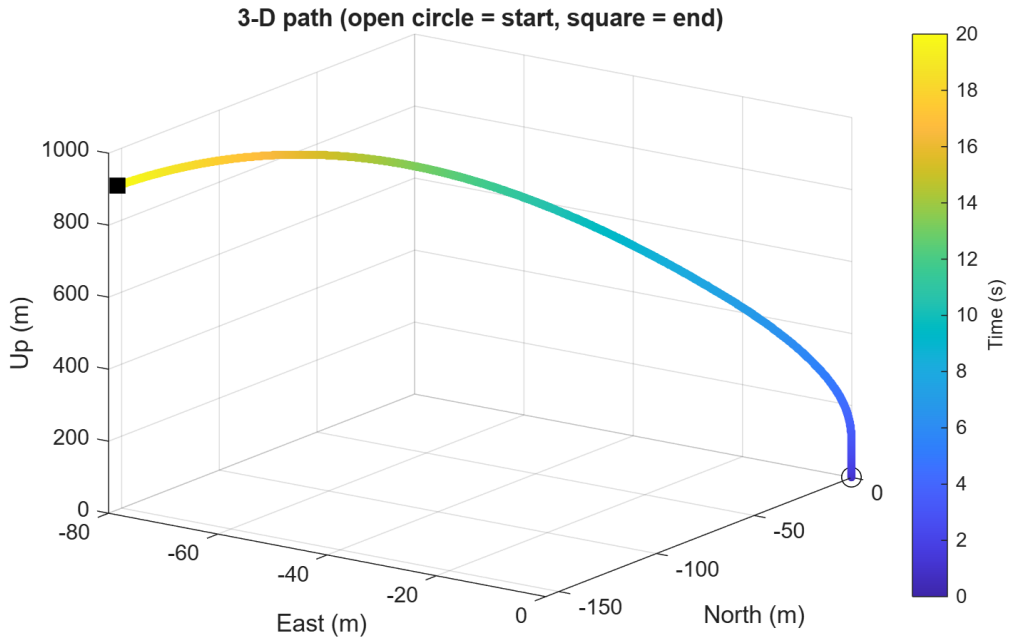

Figure 11. 3-D trajectory for the demo case, colored by time.

4.5 Varying Key Parameters

The first parameter is the size of the pitch gimbal angle. Holds of 0.02, 0.04, and 0.06 degrees from 2 to 3 s (Figure 12) leave the rocket turning at 1.12, 2.25, and 3.37 deg/s and 61, 122, and 183 m South of the pad at 20 s. Apogee falls from 986 m to 975 m. For small angles the torque in (2) is proportional to the gimbal angle, so doubling the angle doubles the rate and the drift. Each curve is a separate run overlaid on the same axes.

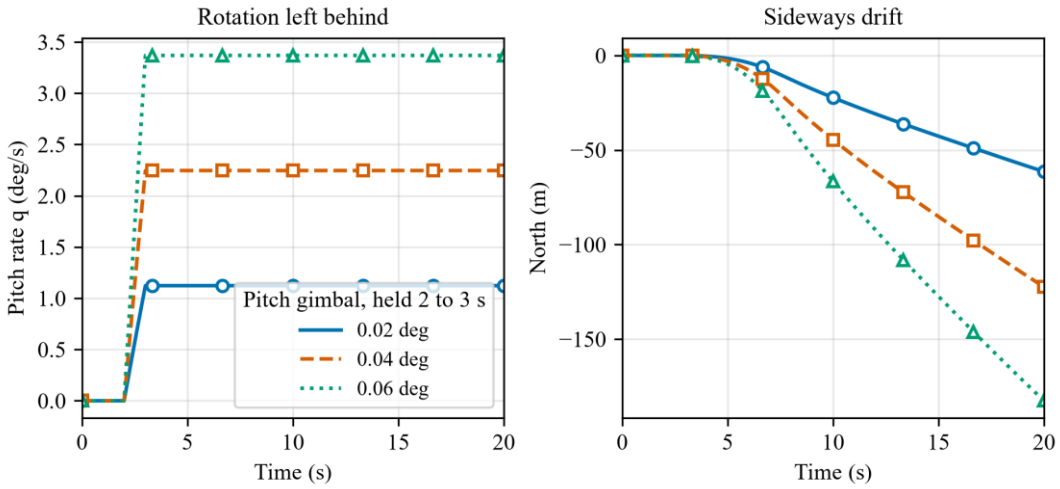

Figure 12. Effect of the pitch gimbal size (held from 2 to 3 s). Each curve is a separate run overlaid on the same axes.

The second parameter is the vehicle mass, scaled by 0.88, 1.0, and 1.2 with the inertia left unchanged (Figure 13). A lighter rocket accelerates harder and climbs higher. Apogee is 1120 m at 0.88 times the mass (about the burnout mass) and 776 m at 1.2 times the mass, with peak speeds of 125 and 91 m/s against 112 m/s for the nominal mass. These runs have no gimbal input, so there is no rotation to compare. The real rocket loses 0.205 kg during the burn, so the real apogee would be higher than the 988 m predicted here.

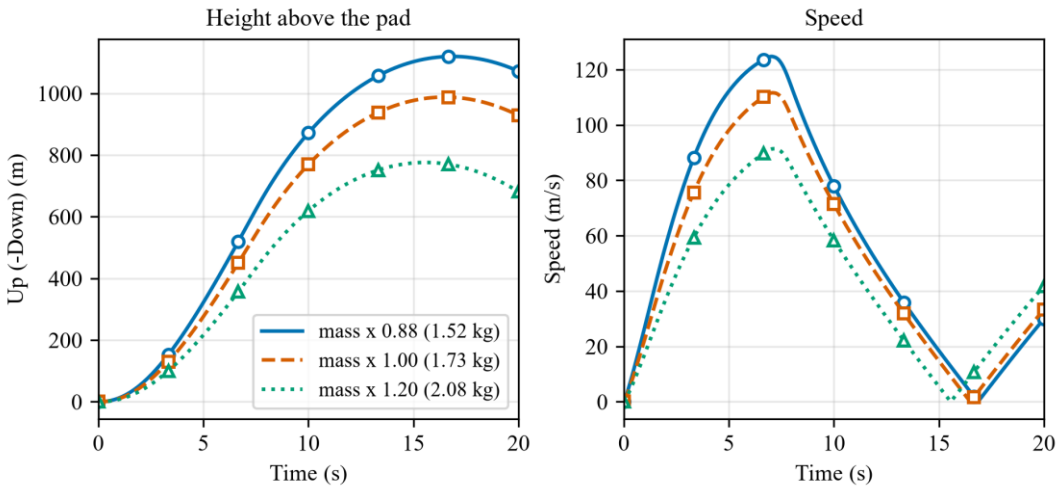

Figure 13. Effect of the vehicle mass with no gimbal input. Each curve is a separate run overlaid on the same axes. Only the mass is scaled and the inertia is unchanged.

With no damping, a rotation persists, so the open loop rocket does not correct itself. A gimbal angle of 0.04 degrees, less than 1 percent of the 5 degree range, is enough to tip the rocket 39 degrees in 20 s. The largest pitch angle in any case shown is 59 degrees, for the 0.06 degree hold. The largest assumption behind all of these results is that there is no destabilizing aerodynamic moment, so the real rocket would tip over faster than these results show.

5. Disturbances and Uncertainties

No disturbances act on the simulated rocket. There is no wind, no crosswind, no gusts, no launch rail tip off, no thrust misalignment, no offset of the CG from the roll axis, and no spin. OpenRocket uses an average wind of 2 m/s in its launch conditions, and a crosswind of that size is the first disturbance to add. There is also no backlash in the TVC motor, which does exist in the physical hardware. The aerodynamic moment is switched off, and an ANSYS CFD model to bring it back is within the scope of the project. On the real vehicle the camera bumps on the nose and the bars at the base of the TVC assembly add a little drag and some asymmetry (Kraus, 2026b).

The gimbal angles are not directly driven on the real vehicle. They come from a servo rotational design that still needs to be characterized. The model also assumes that the gimbal angle changes instantly, whereas the real servo has a time delay that needs to be characterized. The drag at different angles of attack needs to be characterized, along with the forces that act on the rocket at those angles, since the deflection angles in this study are quite small. The center of pressure cannot be taken from OpenRocket for a finless body. The gimbal geometry is less accurate at larger angles (Stepp, 2024).

A gimbal angle of a few hundredths of a degree already turns the rocket, so the vehicle is very sensitive to the gimbal. It would help to find ways to increase the gimbal angle that is needed for control, which would make the vehicle less sensitive. One way is to move the CG lower on the vehicle. That shortens the arm l in (2), so more gimbal angle is needed for the same torque. It also puts the CG farther behind the CP, which would increase the aerodynamic moment, so this tradeoff needs to be studied once the CP is known.

It is also within scope to add a scheduled mass reduction to follow the burn. Over the burn the mass drops from 1.7293 kg to 1.5243 kg, the CG moves 8 cm toward the nose, and the pitch inertia falls by roughly 17 percent. The CG moves closer to the CP, which changes the stability relationship, and the arm between the CG and the gimbal pivot gets longer, which makes the vehicle more sensitive.

On the hardware side, a possible next step is to characterize the mechanical response of the TVC assembly in a hardware in the loop model. An MPU6050 inertial sensor would be mounted at the exhaust end of the motor, and the live servo positions would be fed into the simulation. On the analysis side, a next step is developing an ANSYS CFD model to characterize the aerodynamic coefficients.

Works Cited

Bigwood, J., Connor, R., Gilbert, C., Kuhnle, C., Mitkov, A., Rutkowski, N., Schrader, C. M., & Ternent, J. (2021). *Design, analysis, and test of a high-powered model rocket-2* (Major Qualifying Project No. JB3-2102). Worcester Polytechnic Institute.

BPS.space. (2020, October 14). *Modeling a thrust vectored rocket in Simulink* [Video]. YouTube. https://www.youtube.com/watch?v=nwgd1CV__rs

Kraus, J. (2026a). *OpenRocket model and simulation of the Kalman TVC rocket* [Unpublished simulation data].

Kraus, J. (2026b). *Measured parameters of the Kalman TVC rocket* [Unpublished raw data].

MathWorks. (2020). *Modeling a thrust vector controlled rocket in Simulink* [Source code]. GitHub. https://github.com/mathworks/Modeling-TVCd-Rocket-in-Simulink

Niskanen, S. (2013). *OpenRocket technical documentation* (Version 13.05). https://github.com/openrocket/openrocket/releases/download/OpenRocket_technical_documentation-v13.05/OpenRocket_technical_documentation-v13.05.pdf

Stepp, N. A. (2024). *MSFC-TVC TB-03, Derivation of thrust vector control (TVC) actuator-force / gimbal-torque transformation matrix* (NASA/TM-20240012525). NASA Marshall Space Flight Center. https://ntrs.nasa.gov/citations/20240012525

Stevens, B. L., Lewis, F. L., & Johnson, E. N. (2016). *Aircraft control and simulation. Dynamics, controls design, and autonomous systems* (3rd ed.). Wiley.

ThrustCurve.org. (n.d.). *AeroTech I49N* [Motor data]. https://www.thrustcurve.org/motors/AeroTech/I49N

Weisstein, E. W. (n.d.). *Euler angles*. In *MathWorld, A Wolfram Resource*. https://mathworld.wolfram.com/EulerAngles.html

Appendix A. Reaction Disc Sizing

The roll inertia of the vehicle is found by treating it as a thick walled cylinder with the mass of the whole vehicle, an outer radius of 4 cm (the 8 cm airframe), and an inner radius of 2 cm. The length of the vehicle drops out of the roll inertia. The result is inside the OpenRocket range of 0.0015 to 0.0025 for its rounded value of 0.002 (Kraus, 2026a).

$$I_x = \frac{1}{2}\, m\,(R_o^2 + R_i^2) = \frac{1}{2}\,(1.7293)\,(0.04^2 + 0.02^2) = 0.00173\,\text{kg}\cdot\text{m}^2$$

The reaction disc is sized for a Ready To Sky RS2212 920 KV motor, which has a working range of 4000 to 9000 rpm. The disc idles at 4000 rpm, so $\omega_{min} = 419\,\text{rad/s}$, and it takes up roll by speeding up to 9000 rpm, so $\omega_{max} = 942\,\text{rad/s}$. A disturbance torque would give the vehicle an angular momentum $I_x\, p$ about the roll axis if it were left alone. To hold the vehicle still the disc has to take up that same angular momentum.

$$I_x\, p = I_d\,(\omega_{max} - \omega_{min}) \quad \Rightarrow \quad I_d = \frac{I_x\, p}{\omega_{max} - \omega_{min}}$$

The roll rate to hold off is not known yet and still needs to be characterized. For ease it is assumed to be one revolution per second, $p = 2\pi\,\text{rad/s}$, as a baseline. Then

$$I_d = \frac{(0.00173)(6.28)}{942 - 419} = 2.08 \times 10^{-5}\,\text{kg}\cdot\text{m}^2$$

The disc is a ring with an outer diameter of 71 mm, a wall thickness of 10 mm, and a height of 12 mm. Its outer radius is $R_o = 35.5\,\text{mm}$ and its inner radius is $R_i = 25.5\,\text{mm}$. The shaft and the connection to the motor are assumed to add a negligible amount of inertia. For a ring $I_d = \frac{1}{2}\, m_d (R_o^2 + R_i^2)$, so the mass that is needed is

$$m_d = \frac{2\, I_d}{R_o^2 + R_i^2} = \frac{2\,(2.08 \times 10^{-5})}{0.0355^2 + 0.0255^2} = 0.0217\,\text{kg}$$

So the ring needs a mass of about 22 g. The ring has a volume of 23 cm³, so 22 g means a density of about 945 kg/m³, which is the density of a plastic. A metal ring of the same size is heavier and holds more spin. Aluminum (2700 kg/m³) gives 62 g and can take up about 2.9 revolutions per second of roll, and steel (7850 kg/m³) gives 181 g and can take up about 8 revolutions per second. Taking up spin in both directions would need the disc to idle in the middle of the working range, which halves these numbers. This design is in scope and will be built into the rocket to enable roll control, or at least to oppose spin up.

Appendix B. Air Density With Height

The flight stays below 1 km, so the air density was held at $\rho = 1.225\,\text{kg/m}^3$. This assumption was tested. In the standard atmosphere the density falls with height h as

$$\rho(h) = \rho_0 \left(1 - \frac{\Gamma\, h}{T_s}\right)^{\frac{g}{R\Gamma} - 1}$$

with a lapse rate $\Gamma = 0.0065\,\text{K/m}$, a sea level temperature $T_s = 288.15\,\text{K}$, and $R = 287.05\,\text{J/(kg}\cdot\text{K)}$. At $h = 1000\,\text{m}$ this gives $1.112\,\text{kg/m}^3$, which is 9 percent lower than at the pad. At the fastest point of the

flight (112 m/s at about 500 m) the density is only 5 percent lower. A separate vertical flight check that uses this density curve raised the apogee from 988 m to 1003 m, a change of 1.5 percent, so a constant density is a small error for this flight.

Appendix C. Figures and Tools

Figures 1, 5, 6, 9, 12, and 13 were drawn with Claude AI and placed in Microsoft Word. Figure 2 is a screenshot of the OpenRocket model, Figures 3 and 4 are screenshots of the Simulink model, and Figures 7, 8, 10, and 11 are the plot windows that the Simulink model produces. The equations were typeset in Microsoft Word. Where Claude AI helped with the MATLAB files, it is noted in the code.