Motivation: Rockets are the most powerful engines mankind has ever built, launching us into orbit, to the moon, to Mars, and beyond. Rockets typically have several stages, with each successive stage separating when it’s fuel is spent to reduce the mass of the remaining stages. Rocket boosters (the actual rocket) forms the first stage that is ejected. Until recently, these boosters typically fell back to Earth, sometimes burning up in the atmosphere during reentry, often guided to splash into the oceans, to be recovered by ships later for reuse. Recent advances in rocket control has enabled booster engines to re-ignite after separation to attempt powered descent and landing. If successful, this process would provide a 100x savings for rocket launch. Blue Origin and SpaceX have successfully achieved booster landings in the past year.

System Model: We will consider a planar rocket booster with COM position given by the coordinates $(y, z)$, with an angle $\theta$ to the vertical, and with the thruster at an angle $\psi$ wrt to the rocket. We will assume the mass of the rocket is $m(t)$ at time $t$. The mass decreases as more fuel is burnt for the thruster. The system thus is a 5 degree-of-freedom system. The inputs to the system are (a) the thrust that is produced, which is proportional to the mass that is being expelled (fuel burnt) and the exit velocity of the gas particles, and (b) the torque to gimbal the thrust direction. An unknown wind disturbance $d$ affects the horizontal position dynamics of the rocket. The dynamics of the system is given by

$$
\begin{bmatrix}
    m  \\
    m  \\
    J  \\
    J_T  \\
    1
\end{bmatrix}
\begin{bmatrix}
    \dot{y} \\
    \dot{z} \\
    \dot{\psi} \\
    \dot{\theta} \\
    \dot{m}
\end{bmatrix}
= \begin{bmatrix}
    0  \\
    m g  \\
    0  \\
    0  \\
    0
\end{bmatrix}
+ \begin{bmatrix}
    -\gamma \sin(\psi + \theta) & 0 \\
    \gamma \cos(\psi + \theta) & 0 \\
    -L \gamma \sin(\psi) & 0 \\
    0 & 1 \\
    0 & 0
\end{bmatrix}
\begin{bmatrix}
    f_T \\
    \tau
\end{bmatrix}
+ \begin{bmatrix}
    d
\end{bmatrix}.
$$

Problem Definition: You will be given an (unknown) initial condition, $x_0 = [y \ z \ \theta \ \psi \ \dot{y} \ \dot{z} \ \dot{\theta} \ \dot{\psi} \ m]^T$, for the system, representing the state after booster separation. Your goal is to design a controller that computes the control input $u = [f_T \ \tau]^T$ for each instant in time that will take your system from the specified initial condition to the landing pad within 60 seconds. Your controller will be tested for multiple initial conditions of increasing difficulty. For each test, you will be assigned a score based on the cost function below, with a max score of 100.

$$
J = \begin{cases} 
0, & |P_i| > M_i, \text{ for any } i \geq 2, \\
\sum \alpha_i M_i - |P_i|, & \text{else.}
\end{cases}
$$

---

1In 2014, parts of the Saturn V booster, that launched the Apollo missions in the late 60’s, were recovered from the deep ocean, see [http://www.bezosexpeditions.com/updates.html](http://www.bezosexpeditions.com/updates.html).

2See Blue Origin’s historic rocket landing [http://www.space.com/31202-blue-origin-historic-private-rocket-landing.html](http://www.space.com/31202-blue-origin-historic-private-rocket-landing.html), and SpaceX’s first successful rocket landing [https://youtu.be/CEgB0cKdA0Q](https://youtu.be/CEgB0cKdA0Q).

3Note that this system is like an inverted pendulum on a cart, but with the cart flying
We will use:

\[
P = \begin{bmatrix}
\text{Fuel used} \\
\text{Horizontal distance from center of landing pad} \\
\text{Vertical distance of the rocket bottom from the ground} \\
\text{Landing angle} \\
\text{Landing translational speed} \\
\text{Landing rotational speed}
\end{bmatrix},
M = \begin{bmatrix}
\text{Maximum fuel (150 kg)} \\
20 \text{ m} \\
L = 10 \text{ m} \\
\pi/6 \text{ rad} \\
5 \text{ m/s} \\
1 \text{ rad/s}
\end{bmatrix},
\alpha = \begin{bmatrix}
10 \\
20 \\
10 \\
10 \\
10 \\
10
\end{bmatrix}.
\]

Thus, if you land outside the \((-20, 20)m\) landing pad, or your rocket is above the ground by over 10m, or land with an angle greater than \(\pi/6\) radians, or land with a translational speed of over 5 m/s, or land with a rotational speed of over 1 rad/s, or you take over 60s, your rocket will blow up and you will get a score of zero. The total score will be the sum of all your scores, i.e., \(n \times 100\), where \(n\) is the number of tests. Only the total score will appear on the leader board. (We will use \(n = 10\) with max possible score of 1000.)

**Initial Condition and Disturbance Range:** The following are the ranges of various state variables at the start of simulation:

\[
|y(0)| \leq 100m, \quad |\dot{y}(0)| \leq 10m/s, \quad 25 \leq |z(0)| \leq 1500m, \quad |\dot{z}(0)| \leq 10m/s \\
|\theta(0)| \leq 179^\circ, \quad |\dot{\theta}(0)| \leq 179^\circ/s, \quad \psi(0) = 0, \quad \dot{\psi}(0) = 0 \\
m(0) = 100\% = (25 + 150 kg) \quad |d| \leq 10N
\]

Assume that the wind disturbance \(d\) is zero close to the ground, i.e., \(d = 0\) for \(z \leq 2L\).

**Next Steps:**

1) Download RocketLanding_matlab.zip that has all the matlab simulation code. Execute sim_rocket.m to use the default control to see a rocket animation, various plots, and the computed score. You only need to change two files: (i) student_setup.m - used to do any one-time setup at the starting of simulation, and (ii) student_control.m - used to compute the control at each instant in time.

2) Submit (using the procedure below) your controller. We recommend you submit the default controller to test out everything without doing any control design. The default controller will get you a non-zero score.

**Submission Procedure:**

1) When you are pleased with your control design in student_setup.m and student_controller.m, run the packager pack_o_matic on the matlab command prompt. This will create the submission for a few test cases and generate a file submit.zip.

2) Submit submit.zip as an attachment to autolab. If everything is good, your files will then be graded and a score entry will appear on the leader board.

**Crazy Assumptions:** (can be skipped if you are not a rocket aficionado)

A1) The acceleration due to gravity \((g)\) dramatically changes as a function of altitude. We will boldly assume \(g\) is constant here.

A2) Air drag is a major force on the rocket. These forces are tremendous during re-entry and generate extreme temperatures. We will assume we do not have to deal with air drag and re-entry.

A3) The Saturn V burns 20 tonnes (20,000 Kg) of fuel per second. We will assume our rocket manages to burn a colossal 0.25 Kg/sec.

A4) At high altitude, large lateral velocities could actually cause the rocket to keep missing Earth and essentially go into orbit. We will assume radius of Earth to be infinity.

A5) Thrust typically increases with attitude, however, we will assume this does not occur.

A6) Boost typically involves several changes leading to discontinuities in the mass, thrust, and drag, as various stages separate. We will ignore these discontinuities in the model.
A7) Most importantly, the controller typically does not know the entire state, i.e., the controller does not know the rocket horizontal and vertical positions and velocities, the rocket attitude and angular velocities, and rocket mass. These have to be estimated through various noisy inertial and flow sensors. We will assume the controller has perfect knowledge of the entire state of the rocket!

Thus, for the above reasons, doing well on this project does NOT imply you can play with one of SpaceX’s rockets.