A rocket taking off

Description automatically generated with medium confidence

Image credit: www.spacex.com

Project One

Using Optimization and gradient methods to represent the landing of a rocket

David Wittman | MAE598 | 10/20/22

# Formulation

The representation of a rocket landing can be modeled in a discrete timeframe.

The closed-loop controller aspect of the model specifies control input as:

from above is a neural network with parameters , which are derived from optimization. Implementing loss based on cumulative and short-term incentives, while also defining proper landing conditions we are optimizing:

This formulation is unconstrained with respect to .

Using the Pytorch toolkit, the problem can be further enhanced with realistic conditions i.e. A landing platform, more advanced dynamics, or optimization given multiple initial conditions (position only).

# Programming

Please see the [GitHub Repository](https://github.com/Davwittma/MAE-598-Design-Optimization/blob/main/Project%201.py) (with a large credit to Prof. Ren) for the explicit code. Here we will outline the implementation of the additions to our model, concretely outlined below:

* Landing Platform (acceptable range for the landing criteria)
* Advanced Dynamics
  + Drag
  + Orientation
* Batched Initial Conditions

**Landing Platform:**

Implementing the landing platform puts a further constraint on , the finial position of the rocket. In order to create some level of simplicity, it is beneficial for us to have , although adding a minimum velocity for landing would also increase the “realism” as a dynamic system rarely ever has .

**Advanced Dynamics:**

To account for a multitude of differing variables, like orientation and drag, the initial conditions and the dynamic representation (in and ) need to accommodate these effects. By adding length to the dynamic tensor ( and

**Batched Initial Conditions:**

Creating a batch of initial conditions means that all points in which the initial position affects ( and the action set of both and ) need to be in a tensor form that can be differentiable. Note: This addition to the code requires Stochastic Gradient Descent, SGD for short, to accommodate the large batch size.

# Result Analysis

Here the outcome of the optimization methods will be evaluated.