# INTRODUCTION AND BACKGROUND

Controlling a rocket’s orientation and velocity to safely land upright is not a trivial task. As such, in this project, gradient-based algorithms and differentiable programming are utilized to solve this optimal control problem. In particular, a neural network is designed to control the rocket thrusters to ensure the rocket lands upright with zero linear and angular velocity.

A rocket’s state at time t, x(t), is described mathematically in Eq. (1) and a possible initial state, x(0), is shown visually in Figure 1. The distance to the ground is captured by the variable d(t), the vertical velocity is v(t), the angle relative to vertical is ϕ(t), and the angular velocity is ω(t). Notice the rocket has thrusters on both sides, indicating that either a positive or negative ω(t) is possible. However, there is only one vertical thruster, whose acceleration is used to balance against the downward acceleration due to gravity.

Diagram

Description automatically generated

**Figure 1**: Initial rocket state, x(0)

The desired rocket state, at the end of the analysis period (consisting of T timepoints), x(T), should be zero, as shown in Figure 2. As such, a loss function, l(x(T), a(T), α(T)), is defined in Eq. (2) to quantify how close to zero x(T) is at the final timepoint T. The variable a(T) describes the vertical acceleration, considering both the downward acceleration due to gravity and the upward acceleration due to rocket thrust. The variable α(T) denotes the angular acceleration due to the side thrusters (can be positive or negative). Note that a solution with both rotational thrusters on with equal magnitude is possible, which is why it was mentioned that ω(t) could be either positive or negative, indicating that either the right or left thruster is on, but not both.

Shape

Description automatically generated

**Figure 2**: Final rocket state, x(T)

# METHODOLOGY

The optimal control problem has the objective of minimizing the loss function, as shown in Eq. (3). The components of x(t) are determined by the dynamics in Eq. (4-9). Eq. (4) determines the distance from the ground, based on the velocity (determined in Eq. (5)), which is determined from the acceleration (Eq. (6)), the first action output of the neural network (hence the index 0). Similarly, Eq. (9) determines the angular acceleration (the second output of the neural network, index 1), which will determine the angular velocity in Eq. (8). Then, this angular velocity will determine the angle relative to vertical in Eq. (7). Clearly, this is an unconstrained optimization problem with respect to θ, since all states are determined by the output of the neural network.

The neural network programmed in this problem has 4 layers.

# RESULTS AND DISCUSSION