Controlling The Landing of a SpaceX Starship Rocket on Mars

Michael Kelly (ID: 3032501278), Ross Alexander (ID: 3035294926), Eric Stern (ID: 3035289154)
Rebecca Schwartz (ID: 3035289947), Jeremy Hamlin (ID: 3035295368)

December 17th 2019

Abstract

The goal of this project is to use optimization based control algorithms to simulate the landing of the second stage portion of the SpaceX Starship rocket on Mars. The Starship's dynamics were modeled to simulate a rocket landing using a Model Predictive Controller (MPC). Varying initial conditions and environmental disturbances were tested to ensure robustness.

See the video presentation here:
https://youtu.be/lYbNoOgsdFc

1 Background

The idea of traveling to Mars has captured humanity's imagination for ages. In recent years, SpaceX has made headlines with its ambitious goal of bringing humans to Mars with its rocket, Starship. The rocket modeled in this report uses Starship's specifications, which are shown below.

Starship Specifications				
Height	Diam.	Dry Mass	Main Thrust	Side Thrust
50 m	9 m	120,000 kg	14.4 MN	146 kN

The modeled rocket has 6 Raptor engines, each capable of outputting 24,000 kN of thrust. To account for side thrust, two of SpaceX's Draco engines were used on each side.

2 Dynamic Model

Eq. 1-6 below are nonlinear equations which govern the states of the modeled rocket, where $\dot{\theta}$ and $\dot{\omega}$ are the angular velocity and acceleration about the primary vertical axis, respectively; $\dot{h}$ and $\dot{v}$ are the vertical velocity and acceleration in the positive z axis, respectively; $\dot{x}$ and $\dot{v}_x$ are the horizontal velocity and acceleration in the positive x axis, respectively.

$$\dot{\theta} = \omega \tag{1}$$

$$\dot{\omega} = \frac{L}{J}\left[\frac{F_{TH}}{4} - \frac{F_{TL}}{2}\right] \tag{2}$$

$$\dot{h} = v \tag{3}$$

$$\dot{v} = \frac{F_E cos(\theta) - F_{TH} sin(\theta) - F_{TL} sin(\theta)}{m} - g \tag{4}$$

$$\dot{x} = v_x \tag{5}$$

$$\dot{v_x} = \frac{-F_E sin(\theta) - F_{TH} cos(\theta) - F_{TL} cos(\theta)}{m} \tag{6}$$

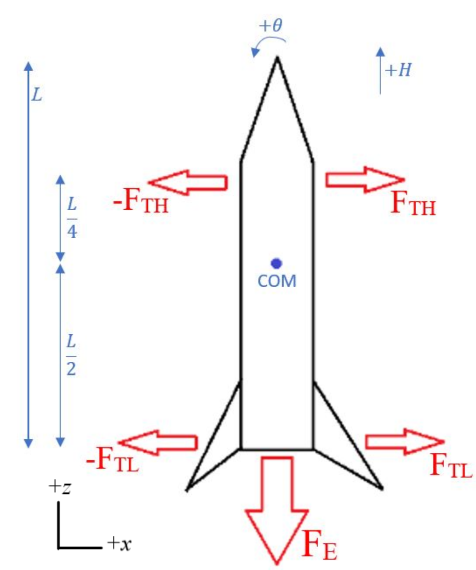

Figure 1: Rocket Schematic

2.1 Non-Linear Discretized Equations

To simulate the rocket, the equations of motion are discretized and the small angle approximation ($cos(\theta) = 1, sin(\theta) = \theta$) is used. See Eq. 7-12.

$$\theta(t+1) = \theta(t) + T_s * \omega(t) \tag{7}$$

$$\omega(t+1) = \omega(t) + T_s * \frac{L}{J}[\frac{F_{TH}(t)}{4} - \frac{F_{TL}(t)}{2}] \tag{8}$$

$$h(t+1) = h(t) + T_s * v(t) \tag{9}$$

$$v(t+1) = v(t) + \frac{T_s}{m}[-F_E(t) - F_{TH}(t) * \theta(t) - F_{TL} * \theta(t) - mg] \tag{10}$$

$$x(t+1) = x(t) + T_s * v_x(t) \tag{11}$$

$$v_x(t+1) = v_x(t) + \frac{T_s}{m}\Big[- F_E(t) * \theta(t) - F_{TH}(t) - F_{TL}(t)\Big] \tag{12}$$

2.2 Linearized Discretized Dynamics

To improve computation speed, linearized dynamics were used in the MPC trajectory tracking algorithm. The Jacobian method was used to linearize the dynamics around the equilibrium point $z_{eq} = [0, 0, 0, 0, 0, 0]$, where the corresponding inputs to ensure a stable equilibrium point are $u_{eq} = [mg, 0, 0]$. This equilibrium point represents the target landing state, where the rocket is on the ground, stationary, and vertically aligned. After linearization, the dynamic formulation in Eq. 13 was found, with the matrices defined in Eq. 14 and 15.

2.3 Environmental Disturbances

To test the robustness of the algorithm, constant force disturbances were added to the simulation to push the rocket off course. Horizontal forces up to 50 kN were tested. The rocket simulation presented in the results section incorporates a crosswind of 11 mph (4000 N horizontal force) while landing.

$$z_{k+1} = Az_k + Bu_k + C \tag{13}$$

$$A = \begin{bmatrix} 1 & T_s & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; z_k = \begin{bmatrix} \theta_k \\ \omega_k \\ h_k \\ v_k \\ x_k \\ v_{xk} \end{bmatrix} \tag{14}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{T_s L}{4J} & -\frac{T_s L}{2J} \\ 0 & 0 & 0 \\ \frac{T_s}{m} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{T_s}{m} & -\frac{T_s}{m} \end{bmatrix}, \; u_k = \begin{bmatrix} F_{Ek} \\ F_{THk} \\ F_{TLk} \end{bmatrix}, \; C = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -g \\ 0 \\ 0 \end{bmatrix} \tag{15}$$

3 Constraints

To correctly model the rocket landing, it was simulated at an initial height of 1500 m traveling downwards at 300 m/s. The rocket started at an initial angle of 0 degrees off its primary axis and horizontal offset 50 m from its landing position.

The following limits were applied to the controller. The rocket was required to be angled between -2 and 2 degrees and have a rotation no more than 10 rad/s. Additionally it could not travel faster than 500 m/s vertically or faster than 100 m/s horizontally.

Instead of landing on a target ship, the rocket was set to land on a 0.4 m diameter pad on the ground. To ensure a proper landing the rocket would have to land within 1 degree off vertical with a rotation of $\leq$ 1 deg/s. The rocket would have to make a soft landing and be within 0.2 m in height of the pad with a vertical velocity of $\leq$ 0.1 m/s and horizontal velocity of $\leq$ 0.1 m/s.

4 MPC Strategy

To allow for a stable and reliable starship landing, an MPC strategy that tracks an optimal path was precomputed using the batch approach for the nonlinear dynamic model. This strategy allows the system to guide itself to a safe landing, even under unexpected conditions like the high speed cross-winds that forced Mark Wattney's crew to leave him behind on the red planet in *The Martian*.

The optimal reference trajectory was precomputed 15 seconds out ($\Delta t = 0.1s$) using the nonlinear dynamics, system constraints, and landing pad as the terminal set using the IPOPT optimization solver.

Next the MPC strategy took over, using a quadratic cost function with terminal cost matrix P, stage cost matrix Q, and input cost matrix R (Eq. 16). These cost matrices apply increasing penalties to the cost function when the states or inputs get further away from the reference trajectory. This difference between state (z_k) and reference state (z_{traj-k}) is $\hat{z}_k$ and the difference between input (u_k) and reference input (u_{traj-k}) is $\hat{u}_k$.

$$
\begin{aligned}
\min_{z_0,\,\dots,\,z_N,\,u_0,\,\dots,\,u_{N-1}} \quad & \hat{z}_N^T P \hat{z}_N + \sum_{k=0}^{N-1} \hat{z}_k^T Q \hat{z}_k + \hat{u}_k^T R \hat{u}_k \\
\text{subject to} \quad & z_{k+1} = A z_k + B u_k + C \\
& u_{Min} \leq u_k \leq u_{Max} \\
& z_{Min} \leq z_k \leq z_{Max} \\
& \hat{z}_k = z_{traj-k} - z_k \\
& \hat{u}_k = u_{traj-k} - u_k \\
& z_0 = z(t) \\
& z_N = z_{traj-N}
\end{aligned} \tag{16}
$$

By solving problem 16 at each time step, an optimal next input and the projected states and inputs N steps ahead are calculated. This approach aims for the reference trajectory value at time $t + N$ as the terminal constraint for the receding horizon, and makes predictions based on linear dynamics to allow for the use of quadprog to solve the optimization problem. After the optimal next input is found, it is applied to the system and the model evolves under nonlinear dynamics and environmental disturbances to the next state, where the process is repeated until the Starship lands. In a real world scenario state estimators would provide feedback controls through sensors located on the ship.

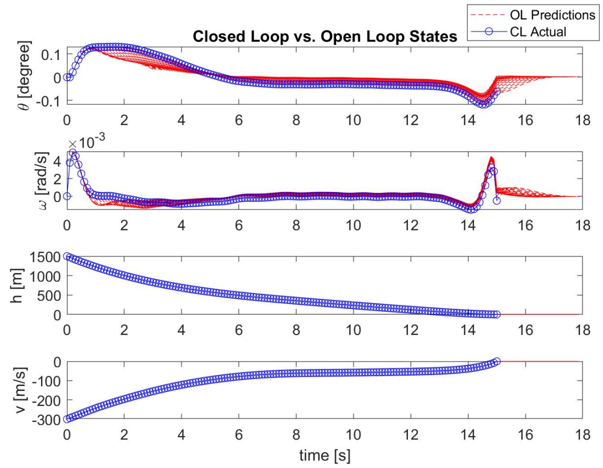

Figure 2: Closed Loop states versus Open Loop predictions for θ, ω, h, and v.

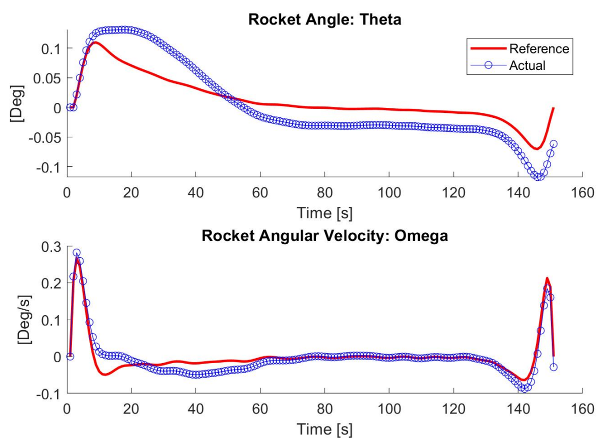

Figure 3: Reference and actual trajectories.

5 Results

Figures 2 and 3 show the rocket states during landing. Although the landing was a success there is disagreement between the predicted and actual states shown in Figure 2. There is also a small amount of tracking error between the optimal reference trajectory and the MPC strategy solution shown in Figure 3.

The linearized dynamics used by the MPC strategy is the cause of these errors. Specifically, the loss of the $F_e * \theta$ term causes the linearized dynamics to over or under-compensate with horizontal thrusters when θ is nonzero. This results in incorrect tracking of θ since the horizontal thrusters affect the angular velocity of the rocket.

6 Conclusions

The combined approach presented in this paper was able simulate Starship's landing for a range of initial conditions and disturbances. Of the initial conditions, only the starting rocket angle θ and horizontal position x were varied independently. When the rocket starting position was vertically aligned with the target it was able to land successfully with θ between ± 2 degrees and disturbance forces up to 50 kN. With a θ of 0 degrees and a 4000 N horizontal disturbance, the horizontal starting position could be varied up to 60 m away from the target before the algorithm could no longer find a feasible solution. The greatest limitation of this method is that the optimal path must keep θ small for the MPC solution to correctly track it. This requires more thruster usage than a regular rocket which would simply tilt its orientation to move horizontally.

References

[1] Francesco Borrelli. Model predictive control algorithm, feasibility and stability. UC Berkeley, ME Dept., ME 231A. PPT.

[2] Francesco Borrelli. MPC: Tracking, soft constraints, move-blocking. UC Berkeley, ME Dept., ME 231A. PPT.

[3] Mark Mueller. Quadcopter dynamics. UC Berkeley, ME Dept., ME 236. PPT.