# Revolutionary Fire Tower Stabilizer: Reaching New Heights with Robotics

1st Jaden Stanshall
*University of Toronto Division of Engineering Science*
*ROB310: Mathematics for Robotics — UTIAS*
Toronto, Ontario
1008792905

## I. Introduction

Fire lookout towers are critical for spotting wildfires in remote and forested areas. Traditionally, these towers are limited in height as their stability in windy and seismic environments relies solely on the structural nature of the tower. However, by integrating robotic stabilization mechanisms, a new generation of fire towers can be built much taller, dramatically increasing visibility and coverage. In short, we can greatly reduce structural requirements of fire towers by off-loading a portion of the stabilizing to a robotic control system, allowing fire towers to be built taller and with less materials; the main difference being no more need for a foundation.

## II. The Model

In this project, we are tasked with simulating the dynamic stabilization system in one dimension by modeling the fire tower as an inverted pendulum mounted on a motorized wheeled platform. As the tower sways, force is applied to the platform (by turning the platform's wheels) to move itself forwards or backwards and maintain the stability of the tower. Given our simulation results, we must comment on whether this new balancing fire lookout tower idea is plausible.

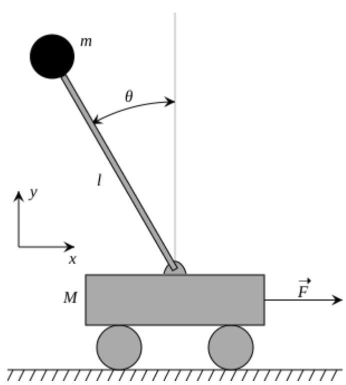


Fig. 1. Diagram of the model we will use to simulate the fire tower stabilization [1].

- $m$ = mass of the tower
- $l$ = half the height of the tower
- $M$ = mass of the platform
- $\theta$ = angle between the tower and the vertical axis
- $F$ = force applied to the platform

The following dynamics describe the continuous motion of the system [2]:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{mg}{M} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2g(m+M)}{Ml} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{2}{ml} \end{bmatrix} \begin{bmatrix} F \end{bmatrix} \quad (1)$$

A derivation of these dynamics can be found at [2]. Note, the following assumptions were made to simplify the problem:

- Perfect motors apply exactly the amount of force to the platform as our controls specify.
- Friction and air resistance are neglected in the system dynamics.
- The dynamics have been linearized by assuming the angles involved are small enough such that $\sin\theta = 0$ and $\cos\theta = 1$.
- The tower has a uniform cross-section along its height and a constant density throughout; that is to say the center of mass of the tower is located half way up the height of the tower.

As your controller, you will minimize a cost function of the following form:

$$\sum_{k=1}^{N} (\mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{R} F_k^2) \quad (2)$$

Where $\mathbf{Q} \in \mathbb{R}^{4\times 4}$ and $\mathbf{R} \in \mathbb{R}^{1\times 1}$ are diagonal matrices.

## III. Project Requirements

The primary objective of this project is to determine a set of control inputs $\mathbf{F} = [F_1, F_2, \ldots, F_N]$ which control the system to stability ($x_k \to 0, \theta_k \to 0$) given the following initial conditions:

$$x_0 = 0m, \quad \dot{x}_0 = 0\frac{m}{s}, \quad \theta_0 = 0.25, \quad \dot{\theta}_0 = 0 \quad (3)$$

Additionally, you are expected to select appropriate values for $\mathbf{Q}$ and $\mathbf{R}$ and briefly justify your choice.

This marks the end of the project description.

## IV. Solution Approach

All report content which follows would not be available to a student assigned this project.

We can break the project into manageable steps:

*1) **Robot Modeling**:*

- First we derive a discrete-time representation of the system. For this project, deriving the $A_D$ and $B_D$ matrices by hand is not necessary; we can use `scipy expm`.

*2) **Planning and Optimization**:*

- Given our discrete system dynamics and cost function, we formulate the optimization problem. We select values for $\mathbf{Q}$ and $\mathbf{R}$ such that the correction of $\theta$ is very highly prioritized compared to all other state parameters, that is, we would like to ensure $\theta$ is weighted very highly in the cost function.
- We can also show this optimization problem is convex (using the second derivative test and noticing the convexity preserving properties of the operations involved), thus its solution is globally optimal.

*3) **Simulation and Interpretation**:*

- We solve the optimization problem in `Python` using the `cvxpy` library. Our simulation/optimization `Python` function keeps a record of the system state and control at each time step.
- We plot $x$, $\theta$, and $F$ over an appropriate time horizon.
- At this point, we can test different values for our initial conditions and our $\mathbf{Q}$ and $\mathbf{R}$ matrices to see how they impact our results.

## V. Results and Discussion

Given the specified initial conditions and using realistic values for mass, we have the following results. $m$ was estimated to be roughly 21,000kg assuming wooden construction. A Ford F350 was selected which is capable of carrying 21,000kg, and the mass of the truck, 3000kg was used as $M$. The height of the tower is assumed to be 18m.

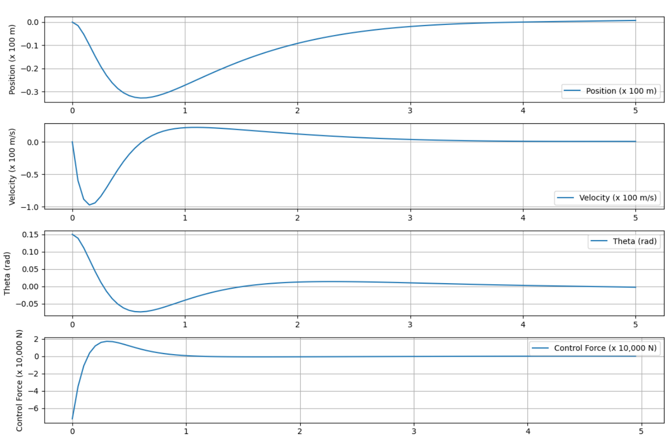


Fig. 2. $x_k$, $\dot{x}_k$, $\theta_k$, and $F_k$ plotted against time

We can see in Figure 2 how the control force is applied to the wheeled platform to balance the pendulum while also moving the cart to the desired location; all values eventually converging to 0, achieving stability for the system.

It's evident that given the results of our simulation, the robotic balancing fire tower redesign is not plausible as the platform (truck) is required to achieve speeds exceeding 150m/s, around $540\frac{km}{hr}$. By adjusting the $\mathbf{Q}$ matrix such that the velocity is weighted higher in the cost function we can achieve maximum velocities around $75\frac{m}{s} = 270\frac{km}{hr}$ which is still too high.

Based on our model the redesign is not plausible, but we should investigate how accurate our model even is. Recall our linearizing assumption, $\sin\theta = 0$ and $\cos\theta = 1$.

Consider the new initial conditions:

$$\mathbf{x}_0 = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \theta_0 \\ \dot{\theta}_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{2\pi}{3} \\ 0 \end{bmatrix} \tag{4}$$

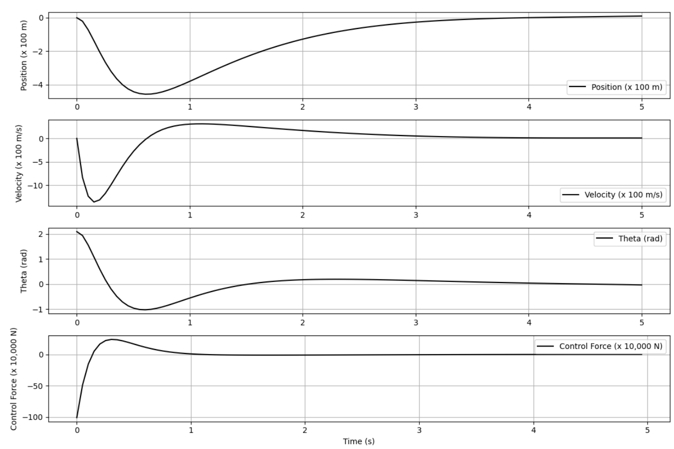


Fig. 3. $x_k$, $\dot{x}_k$, $\theta_k$, and $F_k$ plotted against time given initial conditions (4)

In this case we have given an initial angle greater than $\frac{\pi}{2}$, meaning the pendulum has fallen below the horizon and our small angle assumption no longer holds and our model is invalid. In this case, the wheeled platform would need to perform a complex maneuver to position the pendulum back above the platform in its balancing position, but this is not possible given our linearized system dynamics.

And with that, as our original simulation (Figure 2) involved sufficiently small angles, the errors due to linearization are not as severe, and we can still trust our original results; the robotic redesign of the fire lookout towers is not possible.

## References

[1] "Inverted pendulum," Wikipedia. [Online]. [Accessed: Dec. 04, 2024].

[2] University of Michigan, "Inverted Pendulum: System Modeling," Control Tutorials for MATLAB and Simulink (CTMS), [Online]. [Accessed: 04-Dec-2024].