Abstract

This project presents the development and validation of a mathematical and simulation-based model of a two-wheeled human transporter, similar to the Segway Ninebot S Max. With the rising prominence of self-balancing electric vehicles in micro-mobility, understanding the dynamic stability of such systems is critical for ensuring user safety and optimizing design. The transporter is modeled as a planar, two-wheeled inverted pendulum system constrained to forward translational and rotational motion. The modeling process is divided into three phases: analytical modeling using Newtonian mechanics, linearization and state-space representation for numerical simulation in MATLAB Simulink, and experimental modeling in SimulationX.

In the analytical phase, the transporter is divided into two subsystems: the wheel-base and the user-handlebar. The equations of motion are derived, accounting for thrust, user lean, and inertia. The equations are then linearized using small-angle approximations and implemented in Simulink to determine conditions necessary for balance and to identify operational constraints like force thresholds and tipping points. Subsequently, SimulationX is used to build and test a three-dimensional model, allowing investigation into how design parameters such as mass distribution, thrust force, lean angles, and friction coefficients affect stability and distance traveled before tipping occurs. Experimental findings highlight the trade-offs between thrust, mass, and stability, and underscore the sensitivity of the system to initial conditions and physical parameters. The integration of analytical and simulation techniques in this study yields insights into both passive and actively balanced transporter systems, offering recommendations for robust design and control of self-balancing human transporters.

Table of Contents

List of Figures

List of Tables

1 Introduction

The goal of this project is to develop, analyze, and validate a mathematical model of a two-wheeled human transporter, similar in structure to the Segway Ninebot S Max. These transporters are increasingly common in urban environments and university campuses, providing short-range personal transportation. Unlike bicycles, they use side-by-side wheels and respond to the user's lean to control forward or backward motion through motorized torque. However, their narrow base and high center of mass make them inherently unstable, requiring careful balance to avoid tipping. This instability presents an opportunity to explore dynamic modeling and simulation techniques.

In this project, we model the transporter as a planar two-wheeled inverted pendulum, constrained to forward translational and rotational motion in a 2D plane. Our objective is to determine the conditions and forces necessary to keep the system upright during straight-line travel, and to evaluate how long it can maintain balance under different initial conditions and input parameters.

The project consists of three primary phases. In Part 1, we derive the equations of motion analytically using Newtonian mechanics, accounting for the interaction between the base-wheel subsystem and the user-handlebar subsystem. In Part 2, we linearize the resulting nonlinear differential equations and solve them numerically using MATLAB's Simulink. Finally, in Part 3, we build a 3D experimental model in SimulationX to test different thrust forces and user configurations, with the goal of maximizing the distance traveled before tipping.

Through this process, we aim to bridge analytical modeling and simulation techniques to understand real-world balancing challenges and assess how design parameters such as user mass, lean angle, and thrust force affect stability and performance.

2 Background

Human transporters, such as the original Segway and modern devices like the Ninebot S Max, belong to a class of self-balancing electric vehicles designed for short-distance personal mobility. Initially explored in robotics in the 1990s and popularized by the launch of the Segway Personal Transporter (PT) in the early 2000s, these systems introduced a novel control scheme in which rider lean is used to control acceleration and braking [1]. The onboard control system interprets the rider's forward or backward tilt to generate corresponding wheel motions, enabling intuitive and hands-free navigation.

These vehicles gained popularity for their compact form factor, energy efficiency, and potential to reduce traffic congestion by replacing short car trips. They are widely used in urban commuting, tourism, and campus patrols, and are increasingly recognized as a key component in the future of micro-mobility [2].

From a dynamics perspective, human transporters can be modeled as two-wheeled inverted pendulums, a canonical problem in control systems engineering. Unlike bicycles, which maintain balance through gyroscopic and centrifugal effects, these systems are statically unstable and require continuous active stabilization. Their center of mass lies above the wheel axle, making them prone to tipping without feedback control. Modern devices rely on gyroscopes, accelerometers, and embedded processors that apply control algorithms (e.g., PID or state-space controllers) to maintain balance in real time [3].

This project abstracts away from real-time control and sensor feedback to study the passive dynamics of a transporter modeled in 2D. By simplifying the system into two subsystems,the wheel-base and the user-handlebar, and analyzing their interactions through Newtonian mechanics, we gain insight into how physical parameters like user mass, thrust force, and lean angle influence stability. Such modeling forms the foundation for understanding balance in self-balancing transporters and informs future design and control system development.

3 Methodology

3.1 FBD and KD of the Subsystems

Handlebar-User Subsystem

The handlebar-user subsystem consists of the user and the handlebar modeled as a single rigid body of mass M_u. The handlebar of the transporter is fixed and cannot be titled sideways. The user and handlebar are pinned onto the base and aligned with the wheel's center of mass O, so the user and handlebar contact with the base are modeled as a pin joint. The FBD and KD are illustrated in Figure 1. The key forces acting on this subsystem are:

- $M_u g$ — Gravitational force acting downward at the wheel's centre of mass.
- O_x — Horizontal reaction force that arises at the pin joint O where the user-handlebar subsystem is attached to the base.
- O_y — Vertical reaction force that arises at the pin joint O where the user-handlebar subsystem is attached to the base.

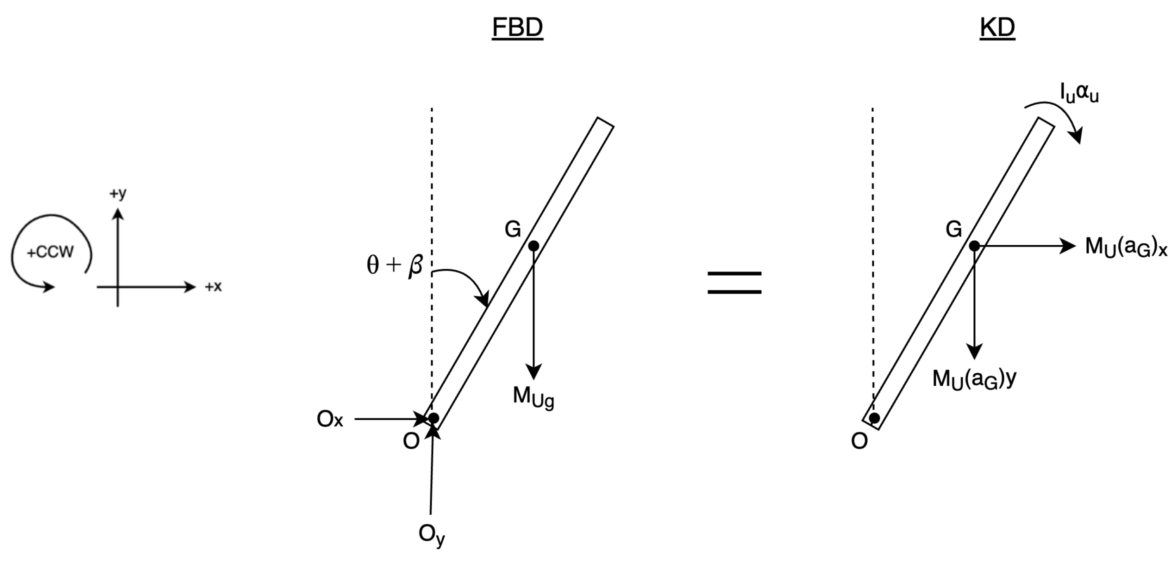


Figure 1. FBD and KD of Handlebar-User Subsystem

Wheel-Base Subsystem

The wheel-base subsystem consists of the wheels, axles, and platform base, modeled as a single rigid body of mass M. The system is constrained to translate in the horizontal direction $x(t)$, and rotates via the wheels without slipping. The FBD and KD are illustrated in Figure 2. The key forces acting on this subsystem are:

- F_T — Thrust force applied at the wheel centre O, generated by the internal motor, pushing the transporter forward.
- F_f — Friction force at the wheel-ground contact point, opposing slipping and enabling rolling without slipping.
- N — Normal force acting upward from the ground, balancing the weight of the base.
- Mg — Gravitational force acting downward at the wheel's centre of mass.
- O_x — Horizontal reaction force that arises at the pin joint O where the user-handlebar subsystem is attached to the base.
- O_y — Vertical reaction force that arises at the pin joint O where the user-handlebar subsystem is attached to the base.

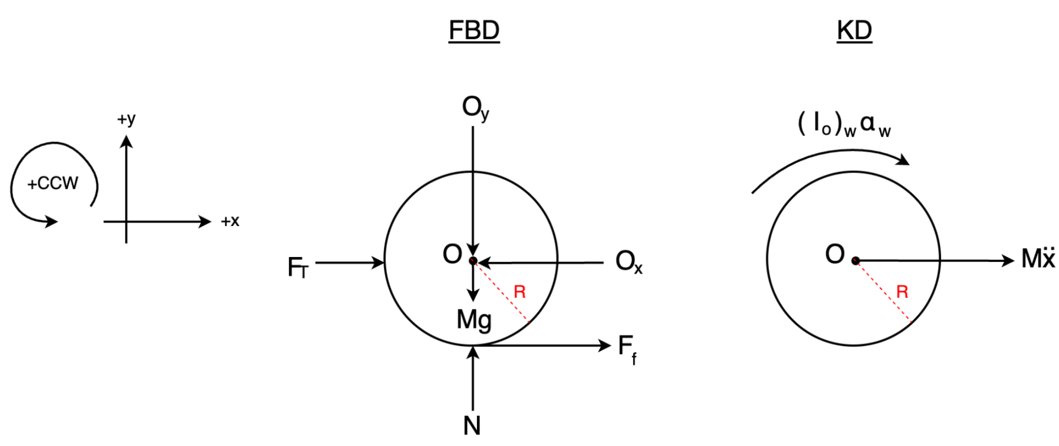


Figure 2. FBD and KD of Wheel-Base Subsystem

3.2 Mathematical Model of the Transporter

Recall,

$$\mathbf{a_G} = \left(\ddot{x} + L_G cos(\theta+\beta)\ddot{\theta} - L_G sin(\theta+\beta)\dot{\theta}^2\right)\hat{\mathbf{i}} + \left(-L_G sin(\theta+\beta)\ddot{\theta} - L_G cos(\theta+\beta)\dot{\theta}^2\right)\hat{\mathbf{j}}.$$

Derivation of Equation of Motion for Handlebar-User Subsystem

Applying Newton's Second Law in the x-direction and substituting in the horizontal component of a_G, we get an equation for O_x:

$$\sum F_x = M_u (a_G)_x$$

$$O_x = M_u\left(\ddot{x} + L_G\cos(\theta+\beta)\ddot{\theta} - L_G\sin(\theta+\beta)\dot{\theta}^2\right)$$

Applying Newton’s Second Law in the y-direction and substituting in the vertical component of a_G, we get an equation for O_y:

$$\sum F_y = M_u (a_G)_y$$

$$O_y - M_u g = M_u\left(-L_G\sin(\theta+\beta)\ddot{\theta} - L_G\cos(\theta+\beta)\dot{\theta}^2\right)$$

$$O_y = M_u\left(-L_G\sin(\theta+\beta)\ddot{\theta} - L_G\cos(\theta+\beta)\dot{\theta}^2 + g\right)$$

Taking the moment about G:

$$\sum M_G = -(I_G)_u\ddot{\theta}$$

$$L_G\cos(\theta+\beta)O_x - L_G\sin(\theta+\beta)O_y = -(I_G)_u\ddot{\theta}$$

Substituting in O_x and O_y:

$$L_G\cos(\theta+\beta)M_u\left(\ddot{x} + L_G\cos(\theta+\beta)\ddot{\theta} - L_G\sin(\theta+\beta)\dot{\theta}^2\right) - L_G\sin(\theta+\beta)\left(-L_G\sin(\theta+\beta)\ddot{\theta} - L_G\cos(\theta+\beta)\dot{\theta}^2 + g\right)M_u = -(I_G)_u\ddot{\theta}$$

Expanding:

$$M_u L_G \cos(\theta+\beta)\ddot{x} + M_u L_G^2 \cos^2(\theta+\beta)\ddot{\theta} - M_u L_G^2 \sin(\theta+\beta)\cos(\theta+\beta)\dot{\theta}^2 +$$

$$M_u L_G^2 \sin^2(\theta+\beta)\ddot{\theta} + M_u L_G^2 \sin(\theta+\beta)\cos(\theta+\beta)\dot{\theta}^2 - M_u L_G g\sin(\theta+\beta)\ddot{\theta} = -\left(I_G\right)_u \ddot{\theta}$$

Notice the $\dot{\theta}^2$ terms cancel out and the following trig $\sin^2\theta + \cos^2\theta = 1$ identity: , we further simplify to achieve the final differential equation:

$$\boxed{\left(M_u L_G^2 + \left(I_G\right)_u\right)\ddot{\theta} + M_u L_G \cos(\theta+\beta)\ddot{x} - M_u L_G g\sin(\theta+\beta)\ddot{\theta} = 0}$$

The coefficients are as follows:

$$\boxed{C_4 = M_u L_G^2 + \left(I_G\right)_u \ ,\ C_5 = M_u L_G \cos(\theta+\beta)\ ,\ C_6 = -M_u L_G g\sin(\theta+\beta)}$$

Derivation of Equation of Motion for Wheel Subsystem

Applying Newton's Second Law for rotation about the wheel center O:

$$\sum Mo = F_f R = -\left(I_O\right)_w \alpha_w$$

$$F_f = \frac{-\left(I_O\right)_w \alpha_w}{R}$$

Using the no-slip condition, we substitute:

$$\ddot{x} = R\alpha_w$$

$$\alpha_w = \frac{\ddot{x}}{R}$$

$$F_f = \frac{-\left(I_O\right)_w \ddot{x}}{R^2}$$

From the handlebar-user subsystem, we found:

$$O_x = M_u\left(\ddot{x} + L_G\cos(\theta+\beta)\,\ddot{\theta} - L_G\sin(\theta+\beta)\,\dot{\theta}^2\right)$$

Applying Newton's Second Law in the x-direction and substituting in F_f and O_x:

$$\sum F_x = Ma_x$$

$$F_T + F_f - O_x = M\ddot{x}$$

$$F_T - \frac{(I_O)_w\,\ddot{x}}{R^2} - M_u\left(\ddot{x} + L_G\cos(\theta+\beta)\,\ddot{\theta} - L_G\sin(\theta+\beta)\,\dot{\theta}^2\right) = M\ddot{x}$$

Isolating for F_T, we obtain:

$$F_T = M\ddot{x} + \frac{(I_O)_w\,\ddot{x}}{R^2} + M_u\left(\ddot{x} + L_G\cos(\theta+\beta)\,\ddot{\theta} - L_G\sin(\theta+\beta)\,\dot{\theta}^2\right)$$

$$\boxed{F_T = \left(M + \frac{(I_O)_w}{R^2} + M_u\right)\ddot{x} + M_uL_G\cos(\theta+\beta)\,\ddot{\theta} - M_uL_G\sin(\theta+\beta)\,\dot{\theta}^2}$$

The coefficients are as follows:

$$\boxed{C_1 = M + \frac{(I_O)_w}{R^2} + M_u\ ,\ C_2 = M_uL_G\cos(\theta+\beta)\ ,\ C_3 = -M_uL_Gg\sin(\theta+\beta)\ ,\ C_7 = 1}$$

3.3 Linearization and Standard State-Variable Model

We begin with the nonlinear equations of motion derived in 3.2:

(1) $$F_T = \left(M + \frac{(I_O)_w}{R^2} + M_u\right)\ddot{x} + M_uL_G\cos(\theta+\beta)\,\ddot{\theta} - M_uL_G\sin(\theta+\beta)\,\dot{\theta}^2$$

(2) $$\left(M_uL_G^2 + (I_G)_u\right)\ddot{\theta} + M_uL_G\cos(\theta+\beta)\,\ddot{x} - M_uL_Gg\sin(\theta+\beta) = 0$$

Using small-angle approximations:

$$\cos(\theta+\beta) \approx 1,\quad \sin(\theta+\beta) \approx \theta+\beta,\quad \dot{\theta}^2 \approx 0$$

Equations (1) and (2) become:

(3) $$F_T = \left(M + \frac{(I_O)_w}{R^2} + M_u \right)\ddot{x} + M_u L_G \ddot{\theta}$$

(4) $$\left(M_u L_G^2 + (I_G)_u \right)\ddot{\theta} + M_u L_G \ddot{x} - M_u L_G g(\theta + \beta) = 0$$

The general state-space form is written as: $\dot{x} = Ax + Bu$, where:

- x is the state vector
- u is the input vector

We set $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$ so that $\dot{x} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} x_2 \\ \ddot{x} \\ x_4 \\ \ddot{\theta} \end{bmatrix}$ so we need to solve for $\ddot{x}$ and $\ddot{\theta}$.

We can put equations (3) and (4) into matrix form: $\begin{bmatrix} a_1 & a_2 \\ a_4 & a_3 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} F_T \\ -a_5(x_3 + \beta) \end{bmatrix}$. Inverting and rearranging we find:

$$\ddot{x} = \frac{(M_u L_G^2 + (I_G)_u)F_T + M_u^2 L_G^2 g(x_3 + \beta)}{\left(M + \frac{(I_O)_w}{R^2} + M_u \right)(M_u L_G^2 + (I_G)_u) - M_u^2 L_G^2}$$

$$\ddot{\theta} = \frac{-M_u L_G F_T - (M + \frac{(I_O)_w}{R^2} + M_u)(M_u L_G g(x_3 + \beta))}{\left(M + \frac{(I_O)_w}{R^2} + M_u \right)(M_u L_G^2 + (I_G)_u) - M_u^2 L_G^2}$$

We can organize these two values into the state derivative vector, written as $\dot{x}$. Next, we separate the system into standard control theory elements:

- The system dynamics matrix (A), which captures how the current state affects the rate of change of the system ($\dot{x}$).
- The input matrix (B), which describes how the external inputs influence the system.

In this case, we are defining the input vector $[F_T \quad \beta]^T$, where F_T and β are the actual control inputs going into the system. This means that in our system model, F_T and β are treated as external inputs, rather than as part of the internal state. The final state-space form clearly shows how changes in the state and the inputs together determine the evolution of the system over time:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{M_u^2 L_G^2 g}{\left(M+\frac{(I_O)_w}{R^2}+M_u\right)\left(M_u L_G^2+(I_G)_u\right)-M_u^2 L_G^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{\left(M+\frac{(I_O)_w}{R^2}+M_u\right)M_u L_G g}{\left(M+\frac{(I_O)_w}{R^2}+M_u\right)\left(M_u L_G^2+(I_G)_u\right)-M_u^2 L_G^2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 \\ \frac{(M_u L_G^2+(I_G)_u)}{\left(M+\frac{(I_O)_w}{R^2}+M_u\right)\left(M_u L_G^2+(I_G)_u\right)-M_u^2 L_G^2} & \frac{M_u^2 L_G^2 g}{\left(M+\frac{(I_O)_w}{R^2}+M_u\right)\left(M_u L_G^2+(I_G)_u\right)-M_u^2 L_G^2} \\ 0 & 0 \\ \frac{-M_u L_G}{\left(M+\frac{(I_O)_w}{R^2}+M_u\right)\left(M_u L_G^2+(I_G)_u\right)-M_u^2 L_G^2} & \frac{-\left(M+\frac{(I_O)_w}{R^2}+M_u\right)M_u L_G g}{\left(M+\frac{(I_O)_w}{R^2}+M_u\right)\left(M_u L_G^2+(I_G)_u\right)-M_u^2 L_G^2} \end{bmatrix} \begin{bmatrix} F_T \\ \beta \end{bmatrix}$$

3.4 Solve the Linearized Transporter Model Numerically using Simulink

To numerically simulate the transporter dynamics, we used equations (3) and (4), rewritten in terms of five derived constants, $A,\ B,\ C,\ D,\ E$:

$$F_T = A\ddot{x} + B\ddot{\theta}$$

$$C\ddot{\theta} + D\ddot{x} - E(\theta + \beta) = 0$$

Where the constants are defined as:

$$A = M + \frac{(I_O)_w}{R^2} + M_u,\ B = M_u L_G,\ C = M_u L_G^2 + (I_G)_u,\ D = M_u L_G,\ E = M_u L_G g$$

Rearranging the equations, we isolate each acceleration term:

$$\ddot{x} = \frac{1}{A}\left(F_T - B\ddot{\theta}\right)$$

$$\ddot{\theta} = \frac{1}{C}\left(-D\ddot{x} + E(\theta + \beta)\right)$$

3.5 Modelling the Transporter Model using SimulationX

The transporter model was modeled in SimulationX using the Multi-Body System (MBS) library (Figure 3). The coordinate system was established with the following conventions:

- X-axis: Forward motion direction of the transporter
- Y-axis: Perpendicular to the wheel face (lateral axis)
- Z-axis: Vertical axis representing height

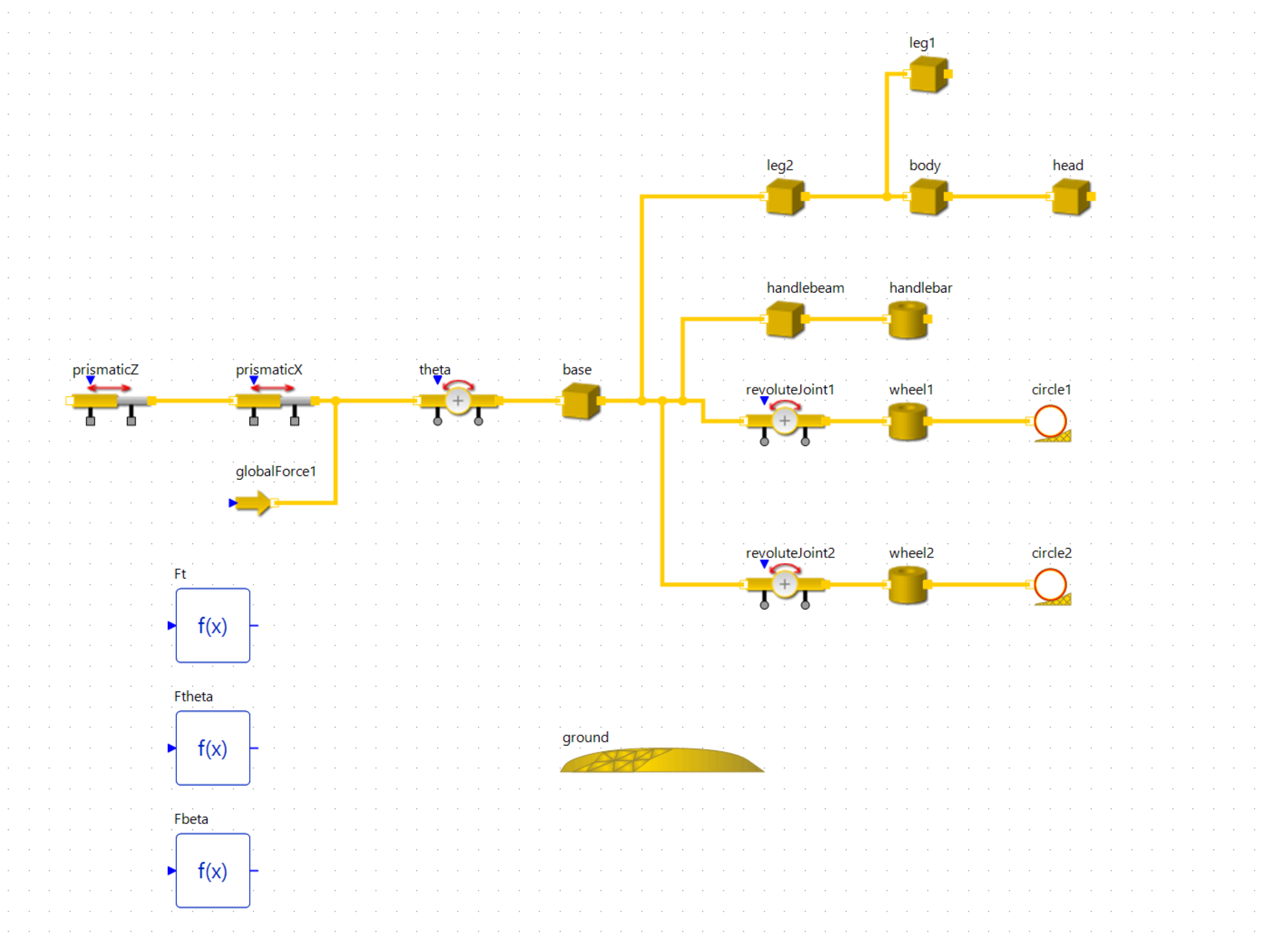


Figure 3. SimulationX Model Components of the Transporter and User

Transporter

The handlebar of the transporter was modelled as two parts: handlebeam and handlebar. These components were fixed to the base. Two identical wheels were connected to the base via revolute joints about the Y-axis. This allows the wheels to freely rotate in the XZ plane, simulating forward and backward rolling motion. The base was modeled as a cuboid with a symmetric mass distribution centered between the wheels. Two prismatic joints were used to constrain and guide the motion of the base:

- A prismatic joint in the X-direction permits translational movement along the ground plane, allowing the base to respond to applied thrust [4].
- A prismatic joint in the Z-direction allows the system to move vertically and respond to wheel-contact forces. When we ran the simulation without it, we observed that the entire structure translated horizontally as a rigid body, and the wheels failed to exhibit rotation or ground interaction. This indicated that the model was over-constrained and that ground contact forces were not correctly applied [4].

Dimensions for all components are summarized in Table 1 and the model can be seen in Figure 4.

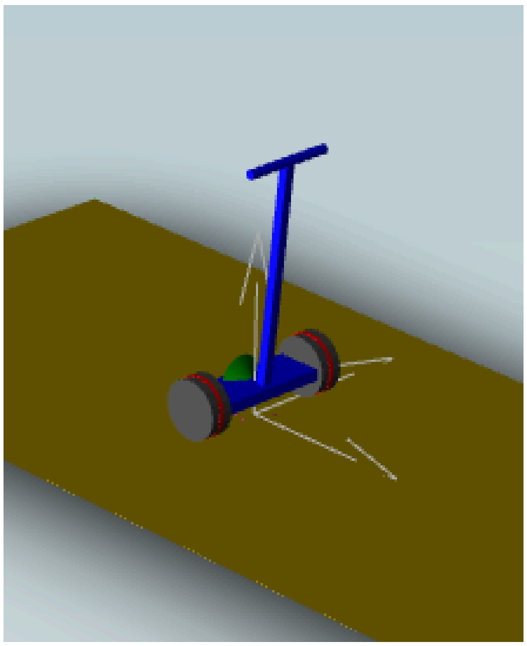


Figure 4. Model of the Transporter

Ground and Contact Modelling

To model contact with the environment, circle objects were placed under each wheel and connected to a fixed ground block. Circle-Line contacts were used to define wheel-ground interactions, as required by the MBS library [12]. All components were aligned and joined using the "Position" tab to ensure correct placement and joint behavior.

Force and Tilt Measurement

A global force block was applied in the X-direction to simulate an external thrust. A global force is applied consistently in the global reference frame regardless of component orientation, which is essential for a transporter where the base and upper structure can tilt during simulation [13]. In contrast, a body force would rotate with the component to which it is attached, potentially misaligning with the intended thrust direction [14]. A theta block was added to monitor and analyze tilt during simulation.

Initial Testing of the Transporter Without User

In the initial simulation, the revolute joint for θ caused the base and handlebar to rotate with the wheels. To isolate wheel behavior, the joint was temporarily removed. As shown in Figure 5, the system exhibited angular acceleration and forward X-motion, confirming correct wheel configuration and force application. However, the wheels appeared to sink into the ground (Figure 6) due to the use of a massless, fixed triangulated surface. This behavior is common when using idealized, non-compliant ground surfaces [15]. This can be addressed in future iterations with compliant contact or by assigning mass and stiffness to the ground.

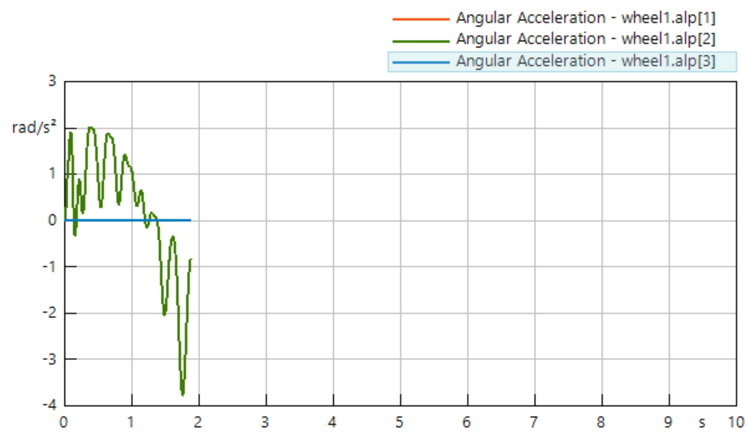


Figure 5. Graph of Angular Acceleration vs Time for the Transporter

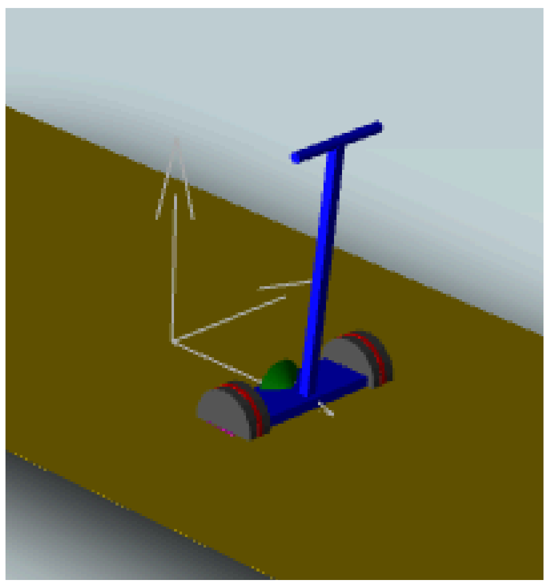


Figure 6. Transporter Sunk Into Ground During Simulation

User

The human rider was modeled using three components: legs, torso, and head. The entire rider was fixed to the base of the transporter. Dimensions for all components are summarized in Table 1 and the model can be seen in Figure 7.

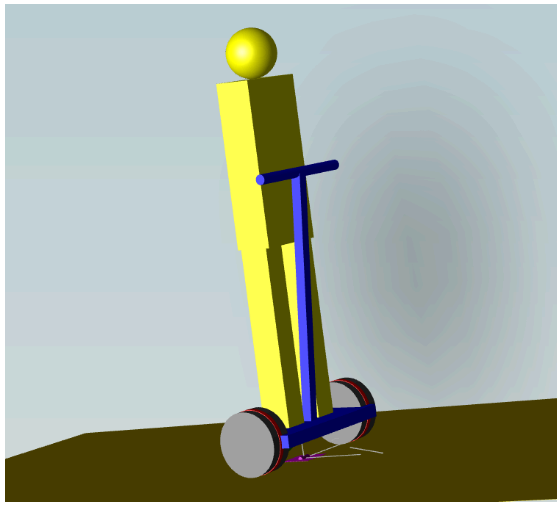


Figure 7. Model of the User

Component	Shape	Dimensions (m)	Mass (kg)	Comments
Base	Cuboid	X: 0.32 Y: 0.60 Z: 0.05	25.0	The total mass of the wheel-base subsystem is given as 35 kg, distributed as 25 kg in the base and 5 kg for each wheel. The average male foot length is approximately 28 cm [5]. To allow for comfortable standing room, we selected a base length of 32 cm. The average male shoulder width is 41 cm [6]. Since users typically stand with their feet shoulder-width apart while riding a transporter, we chose 60 cm. Roughly aligns with Ninebot S Max overall dimensions.
Wheel (x2)	Cylinder	Radius: 0.127 (given) Thickness: 0.10	5.0	We choose 10 cm based on observed proportions.
Handlebeam	Cuboid	X: 0.03 Y: 0.03 Z: 1.00 (given)	1.0	The total mass of the handlebar assembly is given as 1.5 kg, distributed as 1.0 kg in the handlebeam and 0.5 kg in the handlebar. This allocation reflects the larger volume of the handlebeam relative to the cylindrical bar.

				The X and Y dimensions match the handlebar dimensions.
Handlebar	Cylinder	Radius: 0.015 Height: 0.50	0.5	The radius is based on typical motorcycle handlebars, which have diameters ranging from 25 mm to 32 mm [7]. The width was set to slightly larger than the average male shoulder width of approximately 40 cm for ergonomic comfort [6].
Head	Sphere	Radius: 0.09	4.0	Based on the average head circumference of 56.5 cm, which is equivalent to a radius of 9 cm [8]. The given mass of the user is 51.5 kg. According to anthropometric data, the head accounts for around 8% of the total body weight, which is 4 kg [9].
Torso	Cuboid	X: 0.15 Y: 0.30 Z: 0.65	20.5	We are using a female rider of height 165 cm, which is the average height for females in Canada [10]. The values were chosen to approximate the average women's torso dimensions of this height [6]. The mass is set to 20.5 kg to match standard trunk mass, which is around 40% of body mass [9].
Leg (x2)	Cuboid	X: 0.12 Y: 0.10 Z: 0.80	13.5	Dimensions reflect thigh geometry based on standard adult female leg circumference, which is around 55 cm [11]. The mass is set to 13.5 kg per leg, which is around 26% of body mass per leg, consistent with anatomical data [9].

Table 1. Dimensions of SimulationX Transporter and User Components

3.6 Testing the Model

To automatically stop the simulation when the handle bar would hit the ground, we added a terminating condition. This was accomplished by setting a condition in the model properties, under General Parameters, such that the simulation stops when abs(theta - phiRel) > $\pi/2$. Initially, the simulation exhibited significant wheel bouncing, resulting in the wheels losing contact with the ground and slipping. The point of contact would have a nonzero velocity during bounce, causing the model to skid and limiting its displacement to only 0.09 meters (Figure 8) .

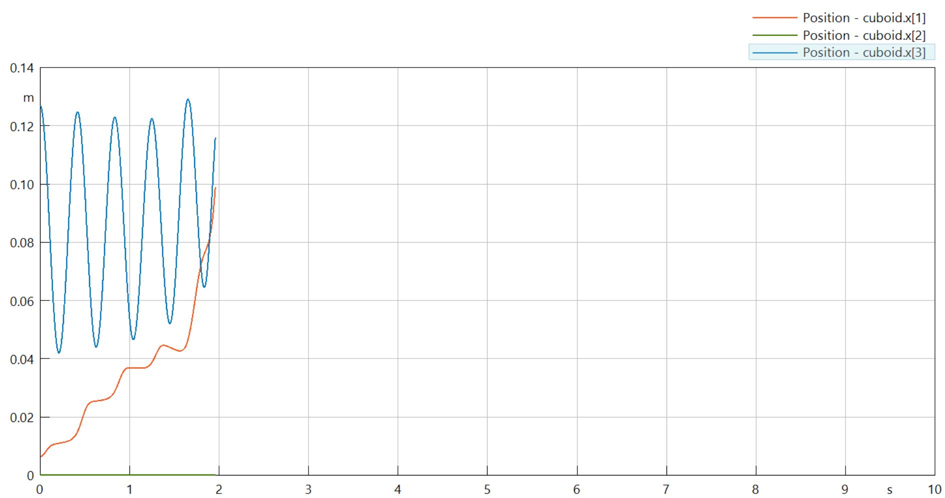


Figure 8. Position Plotted With Default Friction And Damping Values

To mitigate this, we added a rolling resistance coefficient (muR = 0.01) to the wheel-ground interaction. This value was determined through testing and by monitoring the vertical displacement of the wheels, helping to reduce bouncing. We also changed the sliding coefficient of friction in the tangential direction from the default of 0.2 to 0.05.

After reducing bouncing, we tested the simulation using $\theta = 0.05$, $\beta = -0.08$, and $F_T = 80$N loosely based on Simulink. With these changes, the transporter now travels approximately 1.024 meters before falling backward (Figure 7). Falling backwards means that the center of mass moves behind the wheel axle. Although there is still some slight displacement in the z-direction, the overall simulation has improved:

the wheels are spinning correctly and the person falls as expected, which verifies that the adjustments are working as intended. We verified the wheels were spinning by looking at the angular velocity and acceleration.

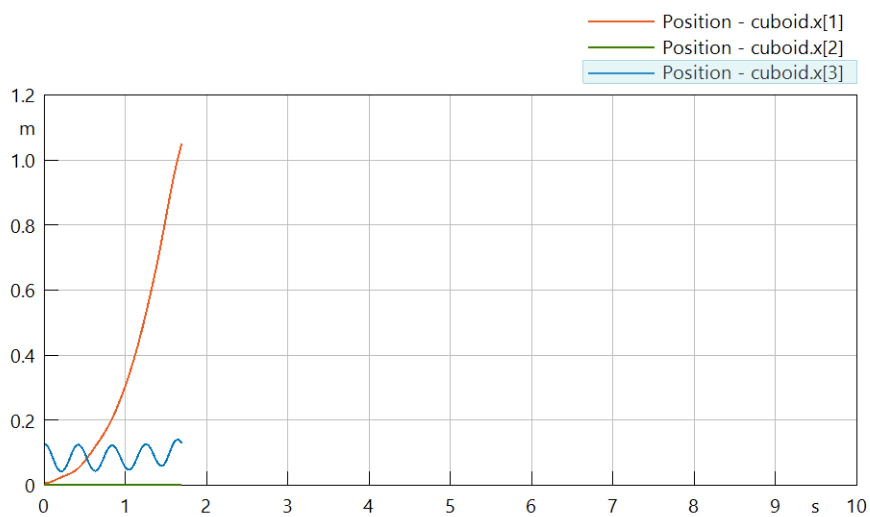


Figure 9. Increased Displacement After Changing Friction And Rolling Resistance Coefficient

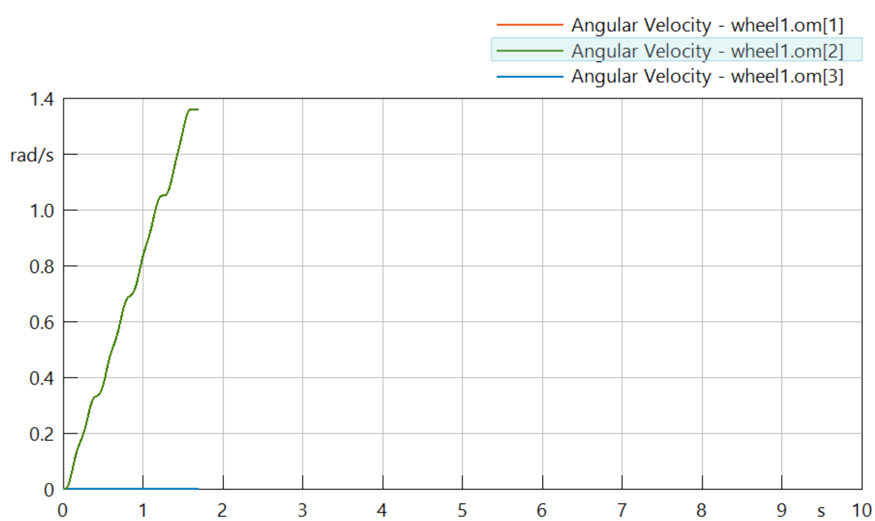


Figure 10. Angular Velocity of Whee1 After Tuning Rolling Resistance and Friction of Contact

4 Results and discussion

4.1 Simulink Results

Using the equations in 3.4, the system was implemented in Simulink using integrator, gain, and summation blocks. As shown in Figure 11, there are two parallel integration chains, one for $x(t)$ and one for $\theta(t)$. All physical constants (see Figure 12) were defined in the MATLAB Model Workspace for modularity and ease of parameter tuning.

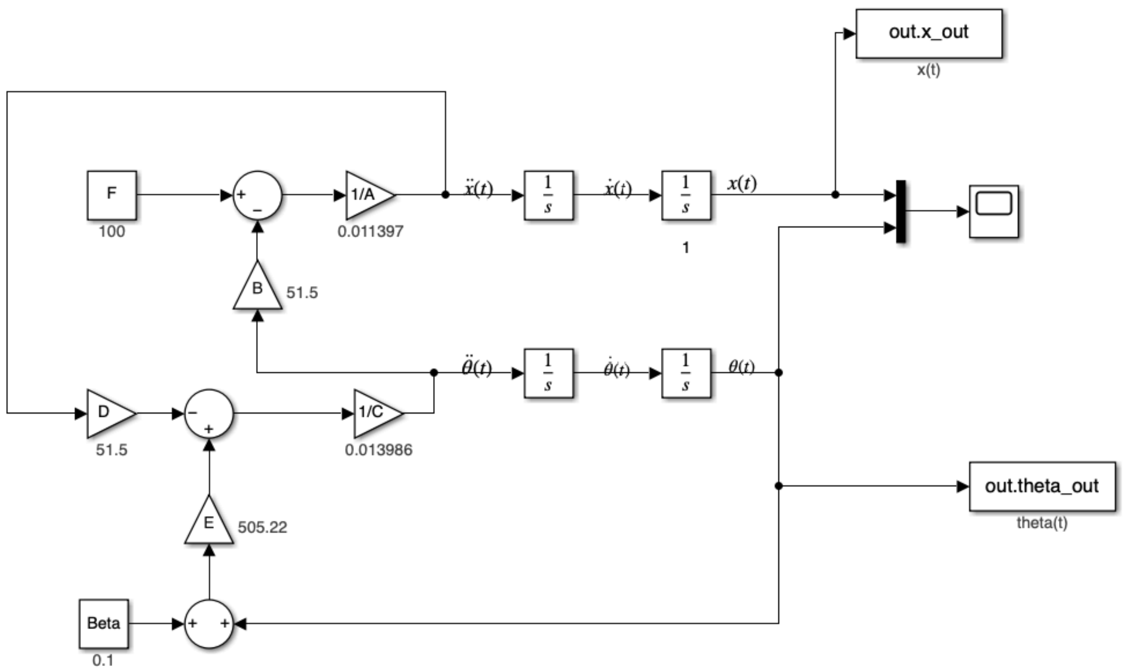


Figure 11. Simulink Model

Data source: MATLAB Code

MATLAB Code:

```
% Define base constants
M = 35;
Mu = 50 + 0.5 * 3;            % = 51.5
Io_w = 0.02;
Ig_u = 20;
R = 0.127;
g = 9.81;
Lg = 1;

% Derived constants
A = M + (Io_w / R^2) + Mu;
B = Mu * Lg;
C = Mu * Lg^2 + Ig_u;
D = Mu * Lg;
E = Mu * Lg * g;

F = 100;
Beta = 0.1;
```

Reinitialize from Source

Figure 12. Base and Derived Constants Used in Simulink Model

We simulated the system's response to a constant thrust force $F_T = 100\,N$ and an initial lean angle $\beta = 0.1\,rad$ over 1 second. The results are plotted using Matlab in Figure 13. From this plot, we observe that the base position $x(t)$ increases smoothly and nonlinearly from rest, while the tilt angle $\theta(t)$ decreases steadily into negative values, in the clockwise direction. This indicates that the applied thrust is accelerating the base forward, but the system is not maintaining its vertical alignment. Instead, the body begins to lean forward. Although the 1-second simulation captures only the initial phase of instability, the trajectory of both signals suggests that the system will continue to destabilize over time.

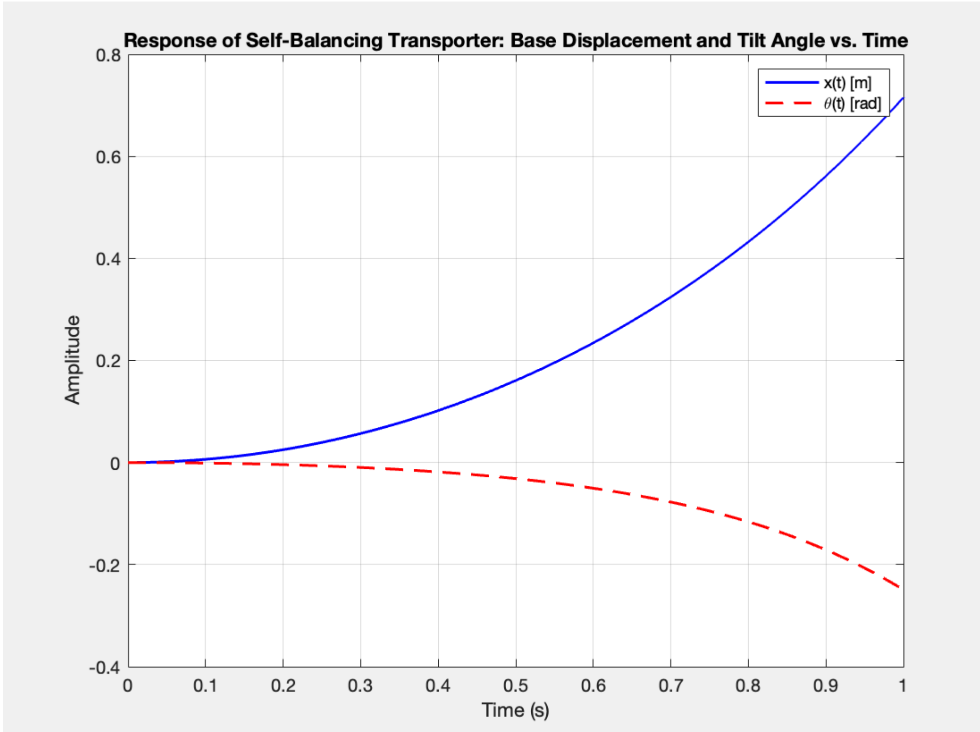


Figure 13. Simulated Response of $x(t)$ and $\theta(t)$ Over 1 Second

To analyze the balancing performance of the self-balancing transporter, we first tuned the thrust force F_T to find a value that allows the system to travel between 3 to 10 meters before tipping over. For this analysis, the initial tilt β was fixed at 0.1 rad, and tipping was defined as occurring when $\theta(t)$ exceeded $\pm \frac{\pi}{2}$ radians ($\pm\ 90^o$), indicating complete loss of balance.

A custom MATLAB function block was created within Simulink to detect when tipping occurs by monitoring $\theta(t)$. Upon tipping, the simulation automatically stops, and relevant outputs, including the time to tip, final displacement $x(t)$ and the tilt angle $\theta(t)$ are logged. This Simulink setup is illustrated in Figure 14, with the Matlab function shown in Figure 15.

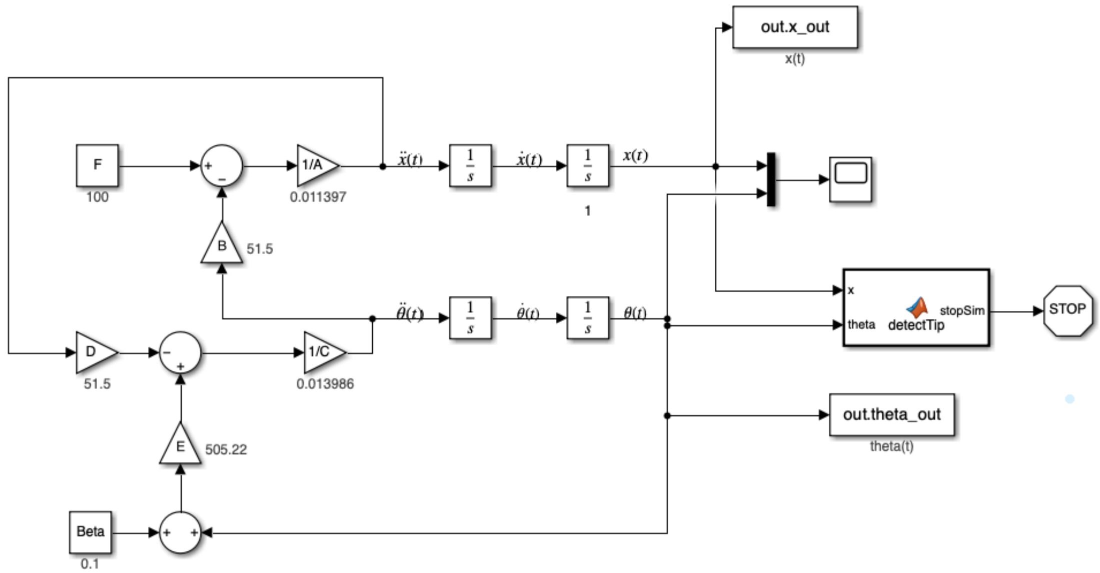
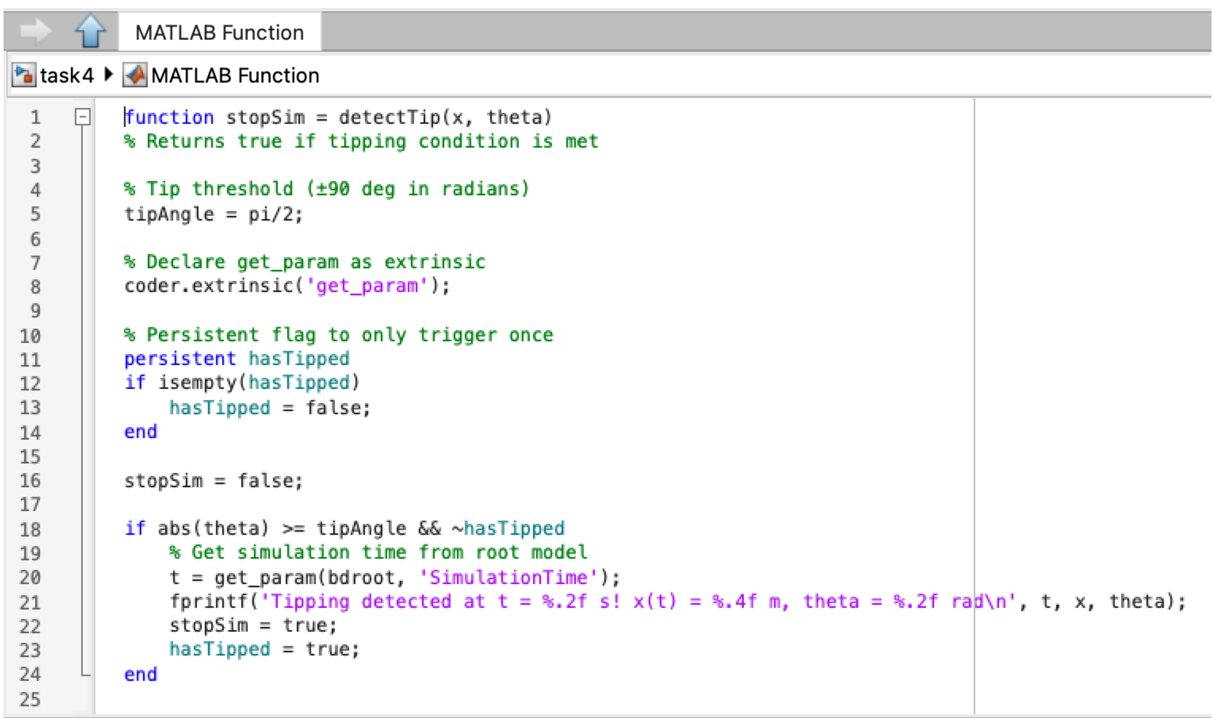


Figure 14. Simulink Model with Matlab Function Block to Detect Tip



MATLAB Function

task4 ▸ MATLAB Function

```
function stopSim = detectTip(x, theta)
% Returns true if tipping condition is met

% Tip threshold (±90 deg in radians)
tipAngle = pi/2;

% Declare get_param as extrinsic
coder.extrinsic('get_param');

% Persistent flag to only trigger once
persistent hasTipped
if isempty(hasTipped)
    hasTipped = false;
end

stopSim = false;

if abs(theta) >= tipAngle && ~hasTipped
    % Get simulation time from root model
    t = get_param(bdroot, 'SimulationTime');
    fprintf('Tipping detected at t = %.2f s! x(t) = %.4f m, theta = %.2f rad\n', t, x, theta);
    stopSim = true;
    hasTipped = true;
end
```

Figure 15. Matlab Function to Detect Tip

We conducted a sweep of F_T values to identify the ideal thrust that would achieve a target displacement within the desired 3-10 meter range. The experimental results are summarized in Table 2. From these trials, we observed that:

- A thrust of 88 N resulted in overshooting the 10-meter upper bound, reaching approximately 13.426 m before tipping.
- Tuning downward, a thrust of 86.07374 N achieved 9.9841 m, while 86.07373 N yielded a final displacement of 10.0022 m.

F_T **(N)**	x **(m)**	$Time$ **(s)**	**θ (rad)**
50	-0.4847	1.24	1.57
80	0.4642	1.75	1.57
90	2.7151	1.87	-1.57
100	2.2203	1.51	-1.57
88	3.0763	2.07	-1.57
86	3.5093	3.01	1.57
86.09	6.6726	3.42	-1.57
86.081	7.4120	3.64	-1.57
86.073	13.4261	5.05	-1.57
86.074	9.6520	4.22	-1.57
86.07375	9.9732	4.29	-1.57
86.07374	9.9841	4.3	-1.57
86.07373	10.0022	4.3	-1.57

Table 2. Simulation Results for Varying Thrust Force F_T

The corresponding simulation plot for $F_T = 86.07374\ N$ is presented in Figure 16. We can see that the displacement increases steadily while the tilt angle approaches the tipping threshold. The system remains upright just until the 10 m mark is reached.

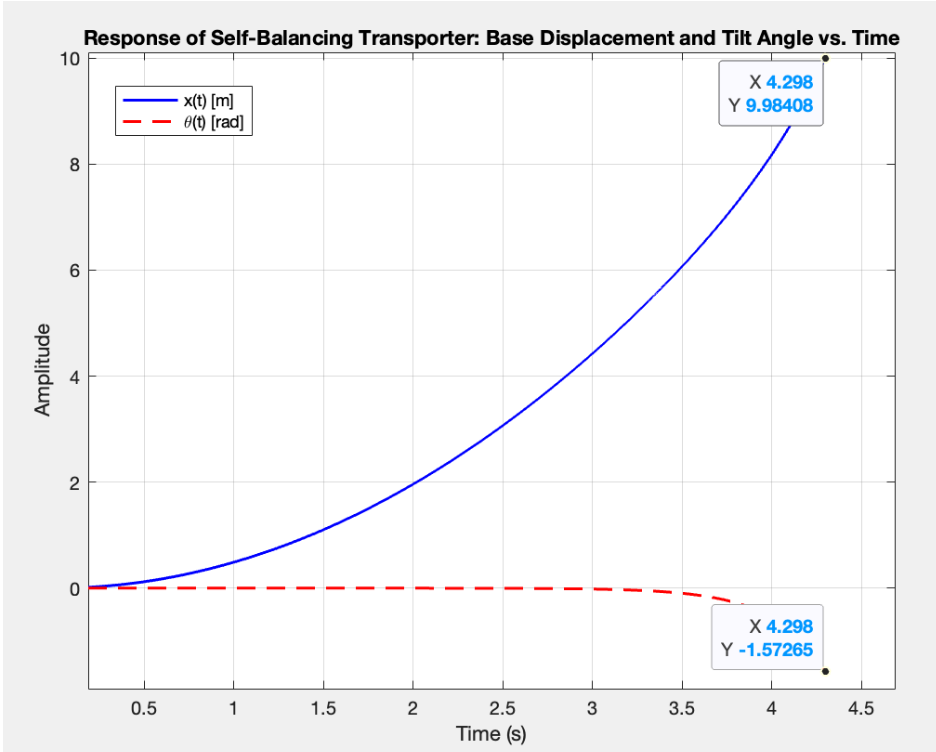


Figure 16. Simulated Response with $F_T = 86.07374\ N$, Reaching 9.98 m Before Tipping

Next, we investigated the effect of varying the base mass M while holding the thrust force at $F_T = 86.07374\ N$. The mass of the base appears only in the derived constant A, which influences $\ddot{x}$. This means that changing M directly affects the linear acceleration and inertial dynamics of the transporter.

Table 3 summarizes the simulation outcomes for varying base masses. The results indicate that increasing the mass of the base reduces the displacement achieved before tipping occurs. For instance, the default configuration with a mass of 35 kg reached a displacement of 9.98 m, whereas increasing the mass to 40 kg reduced the displacement significantly to 0.73 m. This effect is attributed to the increased inertia of the system: a larger mass resists acceleration, causing the base to respond more slowly to the same thrust force. As a result, the base travels a shorter distance before the system becomes unstable and tips.

Interestingly, reducing the base mass (e.g., to 33 kg) also decreased the displacement, to 3.04 m. In this case, the lower inertia allows the base to accelerate rapidly in response to the thrust. However, the upper body (modeled as an inverted pendulum) cannot keep up with the fast-moving base, leading to a loss of stability and earlier tipping. This highlights that both excessively high and low base masses result in suboptimal performance. The sign of the tilt angle θ also changes with mass. Heavier bases tend to result in positive θ, corresponding to a counter-clockwise (backward) tilt. This occurs because the slow-moving base lags behind the pendulum's natural forward motion, causing it to rotate backward relative to the base. Conversely, lighter bases cause a negative θ, indicating a clockwise (forward) tilt, as the base accelerates faster than the pendulum, effectively outrunning it. Figures 17 and 18 show the tipping trajectories for masses of 34 kg and 36 kg, respectively.

M **(kg)**	x **(m)**	$Time$ **(s)**	θ **(rad)**
35	9.9841	4.3	-1.57
50	0.3081	1.62	1.58
20	2.1597	1.33	-1.58
34	3.45125	2.252	-1.57
36	1.60242	2.278	-1.57
34.9	5.1091	2.92	-1.57
35.1	3.2739	2.93	1.58
34.95	5.6848	3.12	-1.57
35.05	3.8686	3.13	1.57

Table 3. Simulation Results for Varying Base Mass M

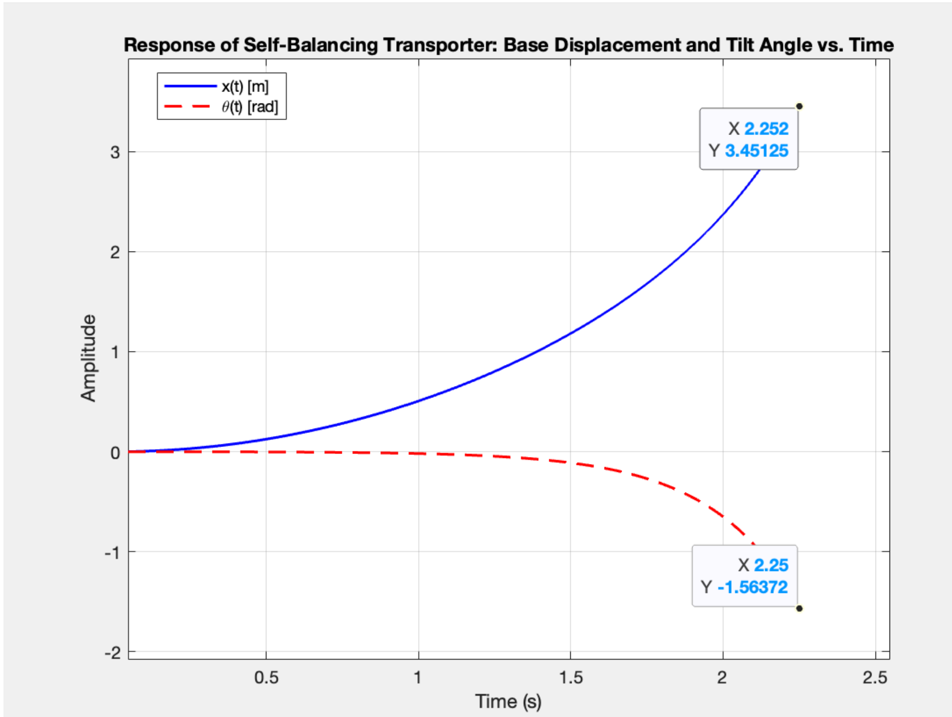


Figure 17. Simulated Response with $M = 34\ kg$, Forwards Tilt

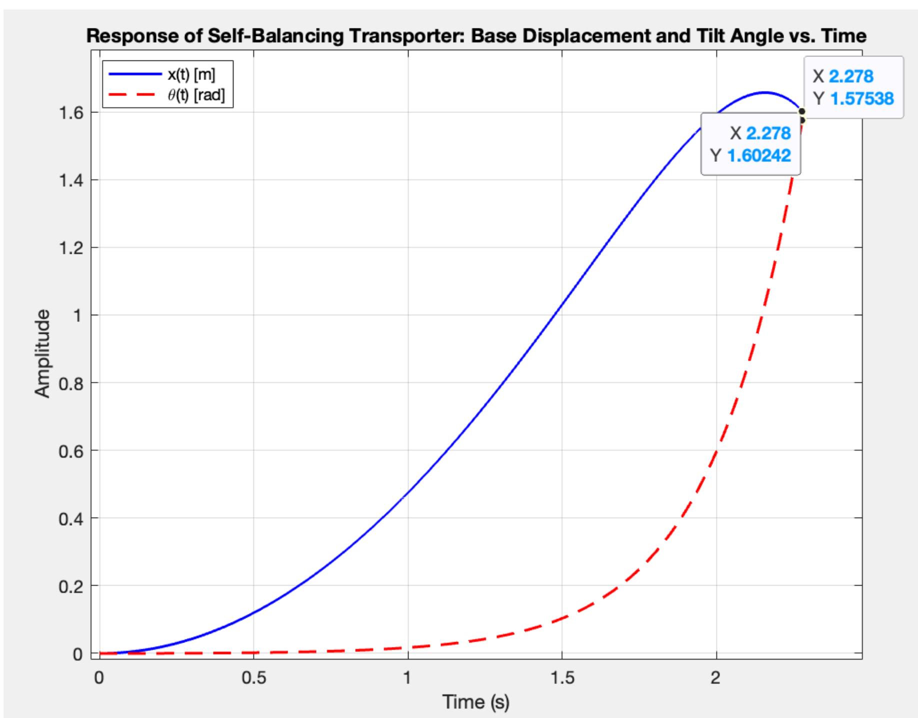


Figure 18. Simulated Response with $M = 36\ kg$, Backwards Tilt

4.2 Tuning the Model in SimulationX

Following Task 8, we tuned the initial angles of theta and beta using Variant Wizard to 0.0509 and -0.098 respectively, to achieve a distance of over 4 metres as shown in Figure 19.

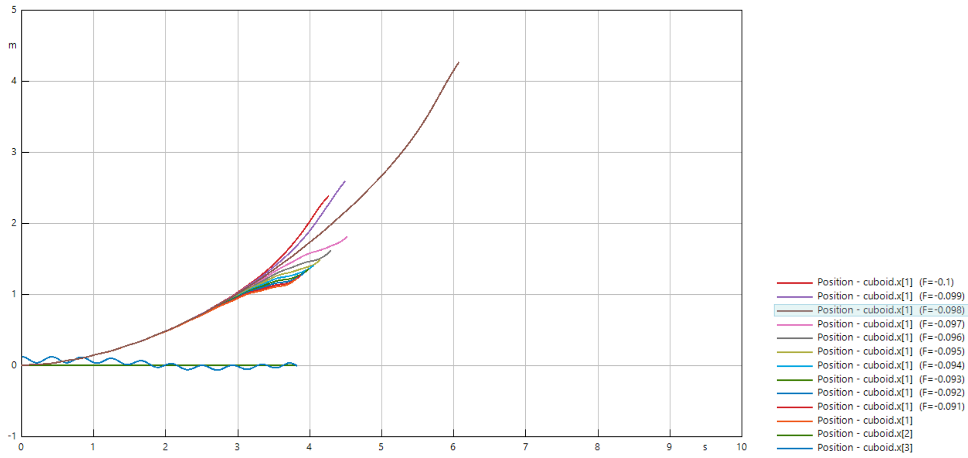


Figure 19. Horizontal Displacement Of Platform With Varying Beta Angles

We continued to tune theta and beta so that the transporter could stay upright for as long as possible, until both angles were at 9 decimals. Then, keeping beta constant, we tuned theta until it got to 16 decimal places, reaching a beta of -0.097994845 rad and theta of 0.0509000297999901 rad, and horizontal distance of around 8.25 metres as shown in Figure 20.

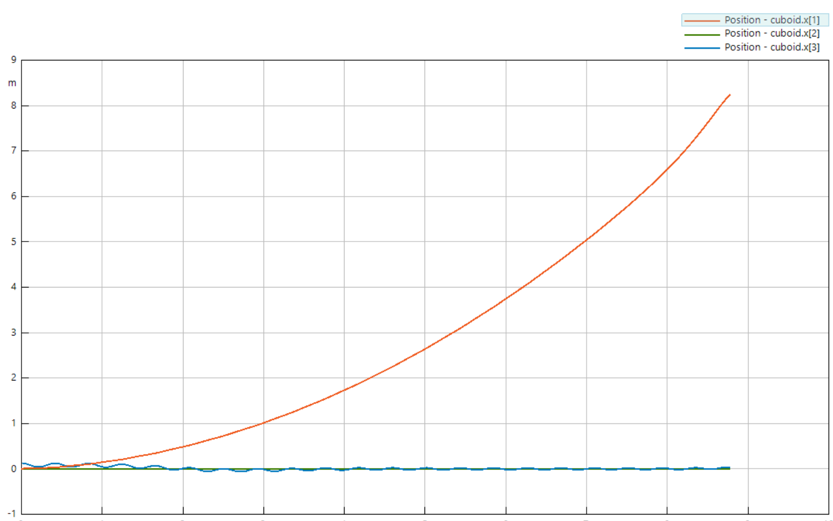


Figure 20. Horizontal Displacement Of Platform With Theta = 0.0509000297999901 Rad

With further tuning of the angles and keeping F_T constant, we were able to achieve a distance of over 20 metres, as seen in Figure 21. Theta was 0.0509000297999901 rad and beta was -0.0979948420329166889 rad.

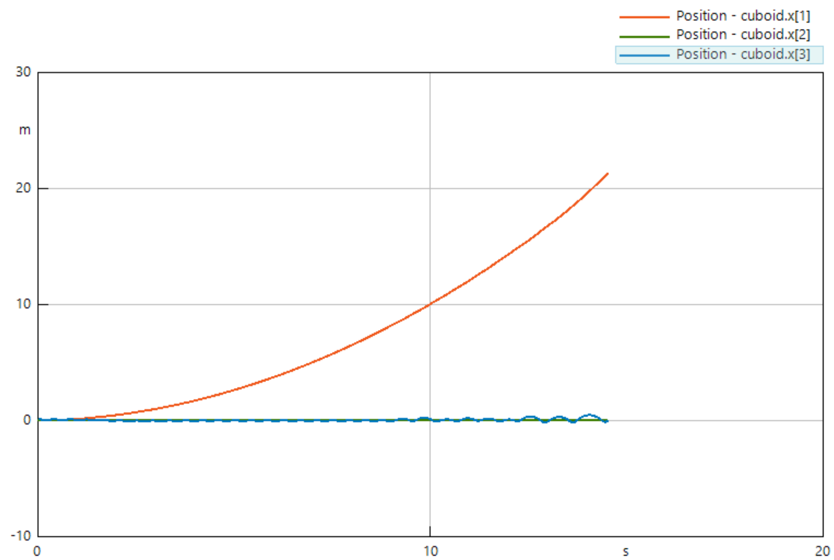


Figure 21. Horizontal Displacement Of Over 20 Metres With Further Tuning Of Angles

Although the transporter travelled much further, upon inspection of the wheels' angular velocity, we noticed the distance from 4 to 9 seconds was just the wheels skidding (seen in Figure 22).

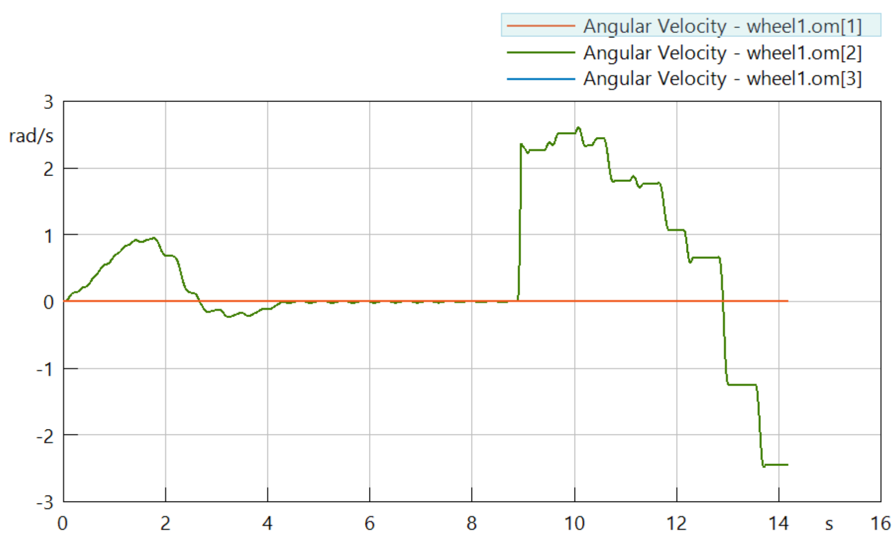


Figure 22. Angular velocity of the wheels, demonstrating skidding

We hypothesized that this might be due to the lowered sliding friction coefficient that allowed for too much sliding instead of rolling. To combat this, we increased the sliding friction coefficient to 0.1, still lower than the default (0.2), but greater than the previous value (0.05). We also decreased rolling resistance in the wheel contacts from 0.01 to 0.001 to allow the wheels to roll more easily. In order to minimize bouncing, we increased damping of contact in the wheels to 150. Finally, in order to maximize distance, we wanted to maximize force, so we chose an initial F_T of 99.99 N. We then tuned theta and beta so that it was just falling forward, which would allow us to continue to increase F_T. This time, we didn't want to overtune our angles, so we capped θ at 5 decimal places and beta at 3, giving us angles of 0.08135 and -0.092, respectively. Now, to tune F_T, we simply tuned one decimal point at a time, increasing and decreasing the number depending on whether the transporter tipped backwards or forwards. We did multiple iterations of this until the horizontal distance converged so that each additional decimal to F_T was no longer visibly changing the displacement. We reached a force of 99.996933376525497294551312 N and final horizontal displacement of 42.2601642570015 metres, as seen in Figure 23.

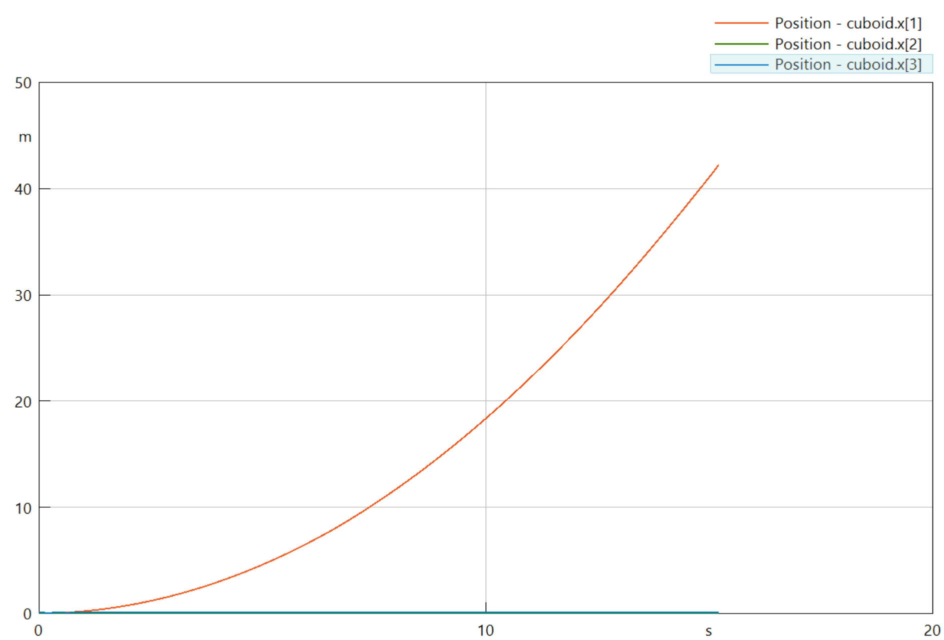


Figure 23. Horizontal displacement of 42.2601642570015 metres

Looking at the angular velocity of the wheel in Figure 24, we can be assured there was no skidding and all the distance was contributed by rolling.

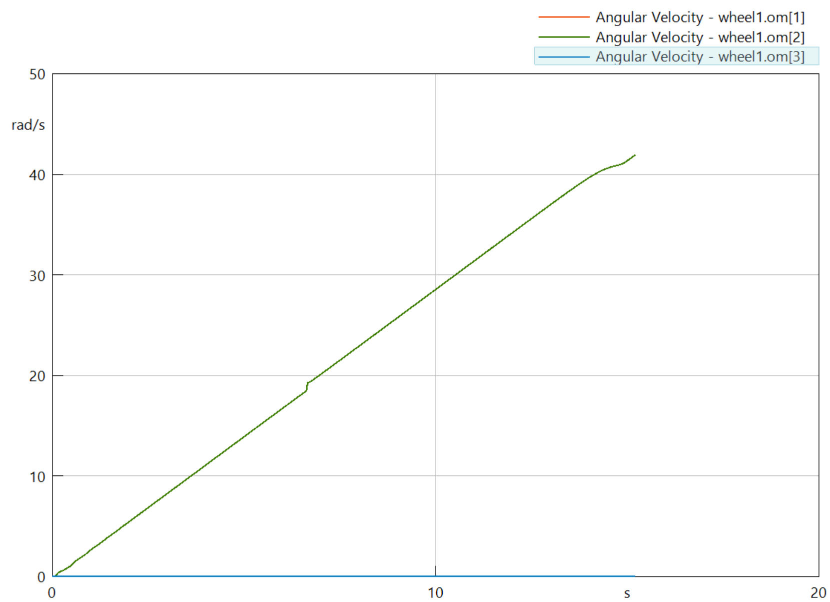


Figure 24. Angular Velocity Of Wheel For Final Model

In Figure 25, you can see a sample of the results obtained from the different beta, theta, and force values we tested.

Beta (rad)	Theta (rad)	Force (N)	Horizontal Displacement (m)
-0.098	0.0509	80	~4.25
-0.097994845	0.0509000297999901	80	~8.25
-0.0979948420329166889	0.0509000297999901	80	~22
-0.092	0.08135	99.996933376525497294551312	**42.2601642570015**

Table 4. Table Of Angles And Forces With Respective Final Horizontal Displacements

4.3 Comparison of Simulink and SimulationX Model

In Simulink, the thrust force F_T was specifically tuned to achieve a final displacement of approximately 10 meters at a tilt angle $\beta = 0.1\ rad$ before tipping occurred. This was successfully achieved with a thrust of 86.07374 N. In contrast, the objective in SimulationX was to maximize displacement while keeping the transporter upright for as long as possible, resulting in a different optimal thrust value of 99.997 N. Unlike Simulink, SimulationX incorporates physical effects like sliding friction, rolling resistance, and contact compliance, which introduce a high degree of sensitivity, making the tuning process more complex; small changes in parameter values resulted in large variations in system behavior.

To ensure the transporter remains upright throughout the simulation, we propose several improvements. First, removing the revolute joint between the base and the handlebar ("theta" in SimulationX model) would eliminate internal rotational freedom, effectively making the structure rigid and preventing tipping. Alternatively, a more realistic solution would involve adding an active control system that dynamically adjusts the handlebar tilt angle β to stabilize the transporter in real time, similar to how real-world Segways operate [16].

Additionally, lowering the rider's center of mass, either by moving mass downward or backward, would reduce the tipping moment and enhance system stability.

5 Conclusions and Recommendations

This study comprehensively investigated the dynamics and stability of a two-wheeled human transporter by integrating analytical modeling and multi-physics simulation. Modeling the transporter as a two-wheel inverted pendulum, we employed both linearized numerical simulations (in Simulink) and detailed physical modeling (in SimulationX) to capture key behaviors and constraints of upright travel.

Our findings reveal that the system is very sensitive to initial conditions and parameter choices such as thrust force, user and base mass, lean angles, friction, and rolling resistance. Open-loop simulations in Simulink confirmed a narrow range of thrust values where the transporter can travel upright without tipping: too little thrust leads to forward tipping; too much, to backward tipping as the base moves out from under the user. The optimized model in SimulationX, which captured realistic friction, ground contact compliance, and rolling resistance, demonstrated additional complexities, including wheel skidding and a far more restricted "safe" parameter space due to nonlinear physical effects.

Through iterative tuning, we achieved upright travel distances exceeding 42m before tipping, but this result depended on extremely precise settings of lean angle and thrust. Unrealistic friction or resistance settings led to non-physical behaviors, underscoring the importance of real-world parameterization. The results highlight critical differences between open-loop configurations and commercial transporters such as the Segway Ninebot S Max [16]. Real Segways use inertial sensors (gyroscopes and accelerometers) and microprocessor-based feedback systems to detect tilt and instantly adjust wheel torque for stabilization[17]. This feedback allows them to remain upright and stable over long distances and variable terrain, something impossible with an open-loop approach, as confirmed by both our modeling and established literature on inverted pendulum robotics [18].

Based on our sensitivity analysis and results, several strategies are recommended for improving modeling fidelity and transporter design. If we were to make minor changes to the existing model, we could further tune the θ and β to more decimal plans. However more major changes outside the scope of this assignment would be to implement active feedback control such as a PID or state-space controller should be prioritized and would emulate the real-time pitch and velocity regulation used in commercial Segways [16]. Our study focused on open-loop platforms where thrust, lean angles, and other parameters are set at the beginning, and the transporter evolves according to the model's dynamics, without any adjustments based on how much the platform actually tilts or how much it has drifted from balance. Open-loop platforms have no feedback controller that measures the tilt or velocity in real time to correct for errors and maintain balance unlike the closed-loop (feedback) system, used in real Segways and the Ninebot S Max, which use gyroscopes and accelerometers to continuously monitor tilt and adjust wheel torque accordingly to stabilize the device [19]. Thus, our model exhibits extreme sensitivity to minor parameter variations and is not robust against disturbances unlike closed-loop systems . Comprehensive sensitivity analysis across all critical parameters (thrust, mass, lean angles, friction, compliance) is also vital to identify robust operational zones and improve model predictiveness. The physical model can be enhanced with empirically derived friction laws and compliant ground models to reduce artifacts such as skidding, since our model still bounces and skids. Further lowering the effective center of mass, a key design principle seen in commercial products [16], could provide additional tipping resistance in both simulation and hardware.

6 References

[1] Y. Kim, S. H. Kim, and Y. K. Kwak, "Dynamic Analysis of a Nonholonomic Two-Wheeled Inverted Pendulum Robot," Journal of Intelligent and Robotic Systems, vol. 44, no. 1, pp. 25–46, Sep. 2005, doi: https://doi.org/10.1007/s10846-005-9022-4.

[2] A. Li and R. Ando, "Measuring the Acceptability of Self-Balancing Two-Wheeled Personal Mobility Vehicles," 2013. Accessed: Jul. 15, 2025. [Online]. Available: https://easts.info/on-line/proceedings/vol9/PDF/P95.pdf

[3] K. Nader and D. Sarsri, "Modelling and Control of a Two-Wheel Inverted Pendulum Using Fuzzy-PID-Modified State Feedback," Journal of Robotics, vol. 2023, pp. 1–13, Jun. 2023, doi: https://doi.org/10.1155/2023/4178227. [4] "Prismatic Joint," Simulationx.com, 2025. https://doc.simulationx.com/4.5/1033/Content/Libraries/Mechanics/MechanicsMBS/Joints/PrismaticJoint.htm (accessed Jul. 24, 2025).

[5] "Know your foot size," | Orazo | Premium Motorcycle Riding Boots, May 19, 2022. https://orazosafety.com/index.php/shoe-sizing/ (accessed Jul. 24, 2025).

[6] K. Watson, "What's an Average Shoulder Width?," Healthline, Oct. 26, 2018. https://www.healthline.com/health/average-shoulder-width#measuring-your-shoulders

[7] "Find My Handlebars: Fitment & Size Guide," BarCraft Handlebars, 2022. https://barcrafthandlebars.com.au/pages/handlebar-fitment-chart (accessed Jul. 24, 2025).

[8] "Head Circumference Chart | Welcome to the Craft Yarn Council," www.craftyarncouncil.com. https://www.craftyarncouncil.com/standards/head-circumference-chart

[9] "Body Segment Data," Exrx.net, 2012. https://exrx.net/Kinesiology/Segments

[10] "What Is the Average Height for Women?," Cleveland Clinic, Oct. 16, 2024. https://health.clevelandclinic.org/what-is-the-average-height-for-women

[11] C. Marshall, "Ideal Thigh Size Breakdown For Men & Women 2024," Endomondo, Oct. 2024. https://www.endomondo.com/training/ideal-thigh-size

[12] "Contact with Circle," Simulationx.com, 2025. https://doc.simulationx.com/4.5/1033/Content/Libraries/Mechanics/MechanicsMBS/Contacts/SurfaceContacts/Circle.htm (accessed Jul. 24, 2025).

[13] "Global Force," Simulationx.com, 2025. https://doc.simulationx.com/4.5/1033/Content/Libraries/Mechanics/MechanicsMBS/Forces/GlobalForce.htm (accessed Jul. 24, 2025).

[14] "Body Force," Simulationx.com, 2025. https://doc.simulationx.com/4.5/1033/Content/Libraries/Mechanics/MechanicsMBS/Forces/BodyForce.htm (accessed Jul. 24, 2025).

[15] "Fixed Triangulated Surface," Simulationx.com, 2022. https://doc.simulationx.com/4.6/1033/Content/Libraries/VehiclesMBS/Components/Wheels/SXTire/Ground.htm?Highlight=triangular (accessed Jul. 24, 2025).

[16] “Ninebot S Max,” Segway, https://www.segway.com/self-balancing/products/ninebot-s-max.html (accessed Jul. 27, 2025).

[17] F. Grasser, A. D’Arrigo, S. Colombi, and A. Rufer, “JOE: A Mobile, Inverted Pendulum,” IEEE Transactions on Industrial Electronics, vol. 49, no. 1, pp. 107–114, Feb. 2002.

[18] P. Neelamegam, et al., “Control and Estimation for Segway-Type Robots,” IEEE Transactions on Robotics, 2020.

[19] M. Ragno, “Open-loop vs closed-loop control systems: Features, examples, and applications,” RT Engineering, https://www.rteng.com/blog/open-loop-vs-closed-loop-control-systems (accessed Jul. 27, 2025).