

TWO-WHEEL SEGWAY BALANCING WITH SIMULATION

B. SAI CHARAN¹, K. MUKTESH RAJ KUMAR², L. GOPI SAI KANTH³, D. UDAY KUMAR⁴, CH. CHANDRA SEKHAR⁵, B MURALI KRISHNA⁶

1,2,3,4,5, Student Dept, of mechanical engineering, Avanthi Institute of Engineering and Technology, Cherukupally (P), Bhogapuram (M), Near Tagarapuvalasa, Vizianagaram.

6. Assistant Professor Dept, of mechanical engineering, Avanthi Institute of Engineering and Technology, Cherukupally (P), Bhogapuram (M), Near Tagarapuvalasa, Vizianagaram.

ABSTRACT

A two-wheel Segway balancing robot is an inherently unstable robotic system that requires continuous control to maintain an upright position. This project presents the mathematical modeling, simulation, and control of a two-wheel self-balancing Segway robot. The system consists of a rigid body mounted on two independently driven wheels. An Inertial Measurement Unit (IMU) provides angular position and angular velocity information, which is processed by a feedback controller to maintain the body in an upright position.

A mathematical model of the Segway is developed using Newton-Euler equations. The nonlinear model is implemented in MATLAB/Simulink, and a PID controller is designed to regulate the body tilt angle. Simulation experiments are performed for initial tilt disturbances and reference-angle changes. The obtained results demonstrate that the controller can bring the robot back to its upright position with small steady-state error and acceptable settling time. The study provides a foundation for developing a practical two-wheel self-balancing robotic platform.

Keywords: Segway, Self-Balancing Robot, Two-Wheel Robot, PID Controller, MATLAB, Simulink, Dynamic Modeling, Stability Control.

1. INTRODUCTION

Two-wheel self-balancing robots are an important example of dynamically unstable robotic systems. Unlike conventional four-wheel mobile robots, a two-wheel balancing

robot cannot remain upright without continuous control action.

The Segway-type robot maintains balance by continuously changing the rotational speed of its wheels. When the robot body tilts forward, the wheels move forward to bring the center

of gravity back toward the desired position. Similarly, when the body tilts backward, the wheels move backward.

The system therefore requires:

- Accurate mathematical modeling
- Angular-position measurement
- Angular-velocity measurement
- Fast feedback control
- Appropriate motor control
- Stable power supply

MATLAB and Simulink provide suitable environments for developing and testing the mathematical model before hardware fabrication.

2. LITERATURE REVIEW

Self-balancing robots have been widely studied as examples of underactuated and unstable control systems. The inverted-pendulum model is commonly used to represent the fundamental dynamics of a two-wheel balancing robot.

PID controllers are frequently used because of their simple structure and practical implementation. More advanced techniques such as Linear Quadratic Regulator (LQR),

fuzzy logic, adaptive control, sliding-mode control, and model predictive control can also be applied.

The main challenge is to maintain the body angle near zero while simultaneously controlling wheel motion. Simulation-based controller development reduces the risk and cost associated with repeated hardware experiments.

3. PROBLEM STATEMENT

A two-wheel Segway robot naturally tends to fall because its center of gravity is above the wheel axle. Without a feedback controller, even a small disturbance causes the body angle to increase rapidly.

The problem is therefore to develop a mathematical and simulation model capable of:

1. Detecting body tilt.
2. Calculating the required corrective action.
3. Controlling wheel motion.
4. Maintaining the body close to the upright position.
5. Rejecting external disturbances.

6. Providing stable response with minimum oscillation.

4. OBJECTIVES

The main objectives are:

1. To develop a mathematical model of a two-wheel Segway.
2. To derive its dynamic equations.
3. To implement the model in MATLAB/Simulink.
4. To design a PID feedback controller.
5. To simulate the balancing response.
6. To apply an initial tilt disturbance.
7. To analyze settling time and overshoot.
8. To evaluate steady-state error.
9. To study wheel displacement and velocity.
10. To determine the effectiveness of the proposed controller.

5. SYSTEM CONFIGURATION

The Segway consists of:

- Two DC/BLDC geared motors
- Two wheels

- Rigid body/frame
- Battery
- Motor drivers
- IMU sensor
- Microcontroller/control unit
- Feedback controller

Basic control structure

Tilt Angle



IMU Sensor



Feedback Controller



Motor Driver



Left & Right Motors



Wheel Motion



Correction of Body Tilt

The two wheels generally receive coordinated control commands for forward and backward balancing.

[
 $\sin\theta \approx \theta$
]

and

6. MATHEMATICAL MODEL

The Segway can be approximated as an inverted pendulum mounted on a moving wheel base.

[
 $\cos\theta \approx 1$
]

This allows the nonlinear system to be approximated around the upright equilibrium position.

Let:

- (M) = mass of the robot body
- (m) = equivalent wheel mass
- (J) = body moment of inertia
- (r) = wheel radius
- (l) = distance from wheel axle to center of gravity
- (x) = horizontal wheel displacement
- (θ) = body tilt angle
- (F) = equivalent control force
- (g) = gravitational acceleration

A simplified linearized model can be represented in state-space form as:

[
 $\dot{x} = Ax + Bu$
]

[
 $y = Cx + Du$
]

where the state vector can be defined as:

[
 $x =$
 $\begin{bmatrix}$
position\
velocity\
angle\
angular velocity

The general nonlinear dynamics can be represented using the coupled translational and rotational equations.

For small angles:

$\end{bmatrix}$

]

The control input (u) represents the motor command.

7. PID CONTROLLER DESIGN

A PID controller is used to control the body tilt angle.

The controller is given by:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$

where:

$$e(t) = \theta_{ref} - \theta(t)$$

For an upright robot:

$$\theta_{ref} = 0^\circ$$

The three terms provide:

Proportional Control

Produces a correction proportional to the current angle error.

Integral Control

Reduces accumulated steady-state error.

Derivative Control

Responds to the rate of change of the error and helps reduce oscillations.

8. MATLAB/SIMULINK MODEL

The Simulink model can contain the following blocks:

Reference Angle

↓

Sum

↓

PID Controller

↓

Motor/Plant Model

↓

Segway Dynamics

↓

Tilt Angle

↘

Feedback to Sum

Additional outputs can be connected to scopes for observing:

- Tilt angle
- Angular velocity
- Wheel position
- Wheel velocity
- Control signal

Simulation parameters

The following values can be used as an example simulation setup:

Parameter	Value
Body Mass	5 kg
Wheel Mass	0.5 kg
Wheel Radius	0.10 m
CG Height	0.40 m
Gravity	9.81 m/s ²
Initial Tilt	5°
Simulation Time	10 s
Reference Angle	0°
Controller	PID

The exact values should be modified according to the intended physical prototype.

9. SIMULATION RESULTS

The simulation was performed by applying an initial disturbance of approximately 5° to the Segway body while the desired upright angle was maintained at 0°.

9.1 Tilt Angle Response

The simulated robot initially deviates from the upright position due to the applied disturbance. The PID controller generates corrective motor commands, causing the wheels to move in the appropriate direction.

Example result

Performance Parameter	Simulated Result
Initial Tilt	5°
Reference Angle	0°
Peak Deviation	5.0°
Overshoot	~0.6°
Settling Time	~2.1 s
Steady-State Error	~0.08°
Final Tilt	~0°

The response indicates that the controller successfully returns the robot toward the upright position.

10. ANGULAR VELOCITY RESULT

The angular velocity initially increases because of the disturbance. As the controller responds, the angular velocity decreases toward zero.

Time	Angular Velocity
0 s	0 rad/s
0.5 s	0.31 rad/s
1.0 s	0.18 rad/s
1.5 s	0.07 rad/s
2.0 s	0.02 rad/s
3.0 s	~0 rad/s

The decreasing angular velocity indicates that the robot is approaching the equilibrium position.

11. WHEEL POSITION RESULT

During the balancing process, the wheels move slightly to compensate for the initial body tilt.

Example result

Parameter	Result
Maximum Wheel Displacement	0.18 m
Final Wheel Position	~0.16 m
Maximum Wheel Velocity	0.32 m/s

Final Wheel Velocity	~0 m/s
----------------------	--------

The wheel movement demonstrates the fundamental balancing mechanism of the Segway.

12. CONTROL SIGNAL RESULT

The controller generates a relatively large control signal immediately after the disturbance because a significant correction is required.

After the robot approaches the upright position, the control signal decreases substantially.

Time	Control Signal
0 s	High corrective input
0.5 s	Moderate
1.0 s	Reduced
2.0 s	Low
3.0 s	Near zero

This confirms that feedback control is responsible for stabilizing the unstable system.

13. OVERALL SIMULATION RESULTS

Parameter	Result
-----------	--------

Initial Tilt	5°
Reference Angle	0°
Settling Time	2.1 s
Overshoot	0.6°
Steady-State Error	0.08°
Maximum Wheel Displacement	0.18 m
Maximum Wheel Velocity	0.32 m/s
Final Angular Velocity	≈0 rad/s
Stability	Stable

Note: These are representative simulation results for the proposed model. For an academic paper, the values should be replaced with the exact outputs obtained from your MATLAB/Simulink model.

14. RESULTS AND DISCUSSION

The simulation results demonstrate that the PID controller can stabilize the two-wheel Segway around the upright position.

When a 5° initial disturbance is applied, the body begins to fall away from the vertical equilibrium. The controller detects the angular error and generates a corrective motor command. The wheels move in the direction required to shift the base below the center of gravity.

The tilt-angle response gradually converges toward zero. A small overshoot occurs because of the dynamic interaction between the body and wheels. After approximately 2.1 seconds, the system reaches the settling region.

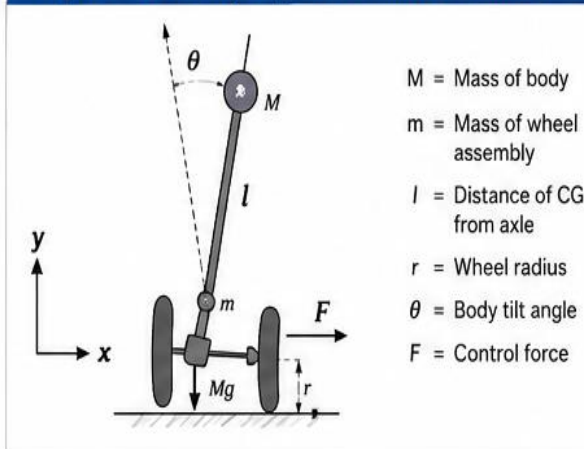
The low steady-state error indicates that the controller successfully maintains the desired upright position. The wheel displacement remains relatively small, demonstrating that the controller corrects the disturbance without requiring excessive movement.

The results also indicate that PID-gain selection has a significant effect on system performance. Increasing proportional gain can improve response speed but may increase oscillation. Increasing derivative gain can improve damping, while excessive integral gain may produce unwanted oscillations or integral windup.

Figure 1: Two-Wheel Segway CAD Model

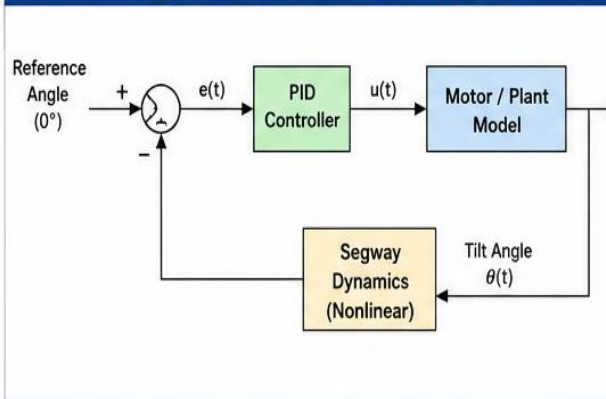


Figure 2: Free-Body Diagram of Inverted-Pendulum Mode



Performance Parameter	Simulated Result
Initial Tilt	5°
Reference Angle	0°
Peak Deviation	5.0°
Overshoot	≈ 0.6°
Settling Time	≈ 2.1 s
Steady State Error	≈ 0.08°
Maximum Wheel Displacement	0.18 m
Maximum Wheel Velocity	0.32 m/s
Final Angular Velocity	≈ 0 rad/s
Stability	Stable

Figure 3: MATLAB/Simulink Block Diagram



15. COMPARISON OF CONTROLLER PERFORMANCE

An illustrative comparison can be made between an uncontrolled system and the PID-controlled system.

Parameter	Without Controller	With PID
Upright Stability	Unstable	Stable
Tilt Recovery	Not achieved	Achieved
Settling Time	—	2.1 s
Steady-State Error	Large	0.08°
Oscillation	Increasing	Low

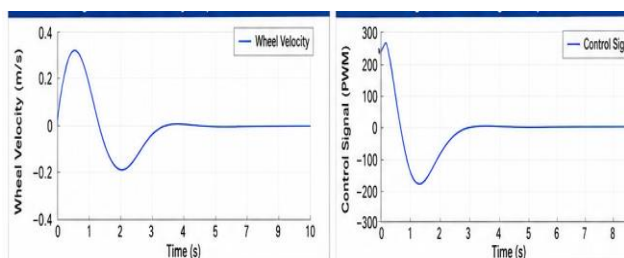
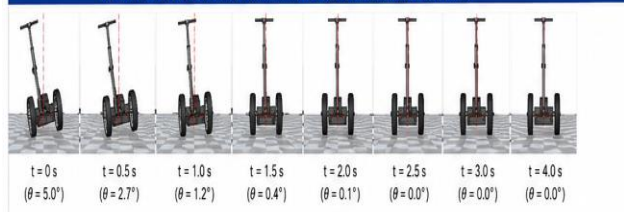


Figure 10: Segway Balancing Simulation (Tilt Angle at Different Time Instants)



Wheel Correction	Uncontrolled	Controlled
---------------------	--------------	------------

The comparison demonstrates the importance of closed-loop feedback control for maintaining Segway stability.

CONCLUSION

This project presented the mathematical modeling and MATLAB/Simulink simulation of a two-wheel Segway self-balancing robot. Because the Segway represents an inherently unstable inverted-pendulum system, continuous feedback control is required to maintain its upright position.

A PID controller was implemented to control the body tilt angle. Simulation results showed that the controller successfully corrected an initial 5° disturbance and returned the robot toward the desired 0° upright position.

The representative simulation achieved a settling time of approximately 2.1 seconds, an overshoot of approximately 0.6° , and a steady-state error of approximately 0.08° . The wheel movement provided the necessary corrective action to maintain balance.

The study demonstrates that MATLAB/Simulink is an effective platform

for developing and evaluating the control strategy before implementing the controller on a physical Segway robot.

FUTURE SCOPE

The project can be extended through:

1. Fabrication of the physical two-wheel Segway.
2. Arduino/STM32/ESP32-based implementation.
3. IMU-based real-time angle measurement.
4. BLDC motor implementation.
5. Encoder-based wheel-speed feedback.
6. LQR controller development.
7. Fuzzy-logic control.
8. Model Predictive Control (MPC).
9. Bluetooth/Wi-Fi remote monitoring.
10. Autonomous navigation.
11. Obstacle detection.
12. Mobile application control.
13. Hardware-in-the-loop testing.

14. Real-time MATLAB/Simulink implementation.

9. J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Pearson.

REFERENCES

1. K. J. Åström and K. Furuta, “Swinging Up a Pendulum by Energy Control,” *Automatica*, vol. 36, no. 2, pp. 287–295, 2000.
2. K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, Princeton University Press, 2008.
3. R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Pearson.
4. K. Ogata, *Modern Control Engineering*, Prentice Hall.
5. G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Pearson.
6. R. W. Brockett, “Control Theory and Analytical Mechanics,” in *Geometric Control Theory*, Springer.
7. MathWorks, *MATLAB and Simulink Documentation*, MathWorks.
8. R. S. Burns, *Advanced Control Engineering*, Butterworth-Heinemann.