

**Research Article***Copyright © All rights are reserved by Uchechukwu C Wejinya*

Kinematic Modeling and Simulation of Inverted Pendulum for Balancing a Two-Wheeled Robot

Kathryn Remell, Kenneth Mordi, and Uchechukwu C Wejinya**Department of Mechanical Engineering, University of Arkansas – Fayetteville, Member, IEEE, USA****Corresponding author:** Uchechukwu C Wejinya, Department of Mechanical Engineering, University of Arkansas – Fayetteville, USA.**Received Date:** February 13, 2023**Published Date:** February 23, 2023**Abstract**

Two wheeled robots are highly maneuverable and capable of handling large shifts in their center of gravity through dynamic stabilization making them more versatile than four-wheeled robots and ideal for extra-terrestrial exploration. The purpose of this paper is to develop a controller for a two wheeled robot for extra-terrestrial exploration. This work develops the controller for a two wheeled robot to achieve the desired control parameters to facilitate autonomous extra-terrestrial exploration. Two lead-lag compensators were developed utilizing analytical methods combined with iteration and simulation. The analytically developed lead-lag compensator failed to meet transient design requirements, but the compensator developed using Matlab's Control System Designer met all desired specifications.

Introduction**Benefits of two-wheeled robots**

Several two wheeled robots have been designed and a variety of control methods have been proven successful at stabilizing and controlling a two wheeled pendulum base. These methods include Sliding Mode [1,2], PID [3], Linear Quadratic Gaussian [4,3], and Evolutionary Fuzzy Control through "an adaptive fusion of continuous ant colony optimization and particle swarm optimization" [5]. While all the control systems have been tested using software simulation, most have not been subjected to real world evaluation.

However, those that have been tested in the real world have demonstrated several unique capabilities and functional areas where the two wheeled mobile robot design excels: dynamic stabilization and maneuverability.

Dynamic stabilization increases a robot's ability to handle large shifts in the center of gravity [4]. This has been demonstrat

ed to lead to robots that are better at handling heavy objects with a smaller support base than a robot dependent on static stability, such as a stationary arm or a three or more wheeled robot [6]. This also improves impulse response during navigation and allows for end-effectors attached to two-wheeled robots to have a wider range of motion than end-effectors attached to four-wheeled robots of the same size. Maneuverability results from the narrower ground footprint and allows for greater performance on uneven surfaces in confined spaces [3] when compared to four-wheeled robots. Combined with dynamic stabilization, a two-wheeled robot is a robust mobile design that has the potential to be more versatile than a four-wheeled robot in most situations.

Mathematical model of two wheeled robot

As discussed in [7], a robot's physical parameters, such as the wheel diameter, the mass of the platform and wheels, and the distance between the center of rotation and the center of gravity, have

a strong influence on the robot's controllability. Therefore, the development of a two wheeled robot must begin with the mathematical model for a generic two wheeled robot. Such a model will allow combinations of parameters to be tested to inform the physical design of the robot. This will also allow the effect of a controller to be tested on various physical configurations and ideal physical ratios to be extracted.

The two major components of motion for a two-wheeled robot are navigation and balance. The mathematical model required for navigation is the kinematic model of a two wheeled robot. The mathematical model for balance is an inverted pendulum as found in [7]. This paper will focus on developing a controller to stabilize the robot's balance.

1. Nomenclature

A. Variables

- C Center
- d Vertical distance from the point of rotation to the platform's center of gravity (m)
- $D(s)$ System controller in s domain
- ess Steady state error
- g Gravity (ms^2)
- $G(s)$ System plant in s domain
- I Moment of inertia ($kg\ m^2$)
- $j\sqrt{-1}$, Imaginary number
- K Proportional gain
- Kp Proportional error constant in step response steady state error
- m Mass (kg)
- p Pole of a transfer function, root of the characteristic equation
- PO Percent overshoot (%)
- q Simplifying variable
- $q(s)$ Characteristic equation, which is the denominator of a transfer function
- r Radius of the wheel (m)

- $R(s)$ System input in s domain
- s Complex frequency, $j\omega$, generally associated with the Laplace transformation
- T_s Settling time (to within 2% of the final value)
- X x-axis (parallel to the ground, forms a horizontal plane with the y-axis)
- $Y(s)$ System output in s domain
- z Zero of a transfer function, the root of the function's numerator
- Z z-axis (vertical axis, defined as perpendicular to the ground and increasing upward)

B. Subscripts

- 1,2,... 1st, 2nd, etc. of the attached variable
- g Gravity (ms^2)
- P Platform reference frame
- R Robot base's reference frame
- $step$ Step Input

C. Greek Variables

- ϵ A very small number greater than zero which replaces a 0 in the first column of a Routh Array.
- η Angle from vertical (angle between the robot's z-axis and the platform's z-axis)
- τ Torque
- ζ Damping ratio of the system
- ω_n Natural frequency of the system (Hz)

System Definition

(Figure 1) The controller and two-wheeled robot system was defined as shown in Figure 1, with the plant, $G(s)$, describing the relationship between the motor torque and the system angle from vertical (η in Figure 2), the controller, $D(s)$, to be developed in this research, and the feedback, $H(s)=1$, to be integrated in future work. The plant transfer function was developed using the small angle approximation for linearization with zero initial conditions during the Laplace transformation [7] (Figure 2).

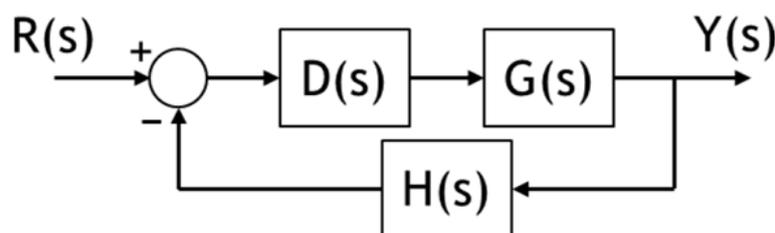


Figure 1: Transfer Function System Diagram.

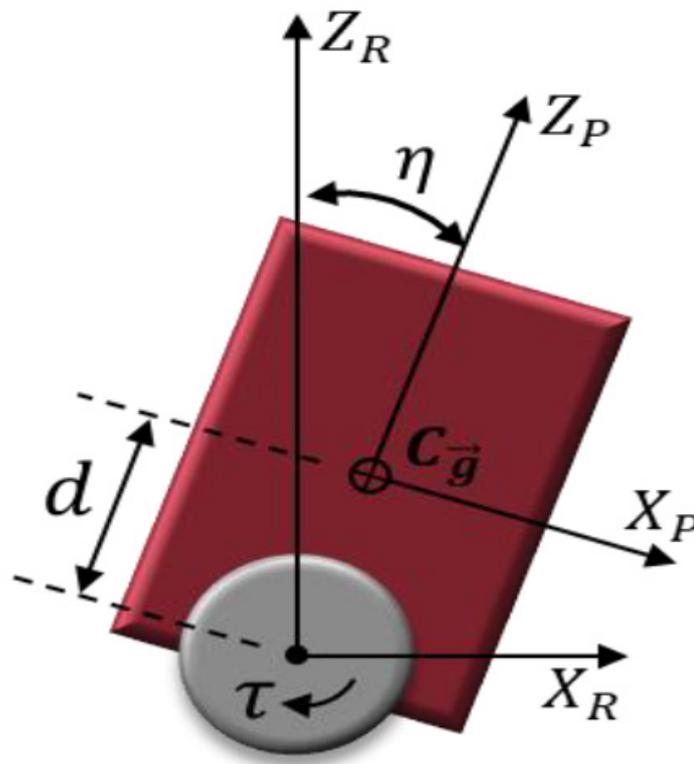


Figure 2: Two-Wheeled Robot Coordinate Systems showing Origins and Angle from Vertical (η).

Plant transfer function

As shown in [7], the plant, defined by an analysis of the system starting from first principles, is

$$G(s) = \frac{q_1 + q_5}{q_1 q_3 s^2 + q_1 q_4 - q_2 q_5} \quad (1)$$

where $q_1, q_2, q_3, q_4,$ and q_5 are simplifying variables from the two-wheeled robot's equation of motion defined as

$$\begin{aligned} q_1 &= I_R + r(m_R + m_P r) \\ q_2 &= m_P r d \\ q_3 &= I_P + m_P d^2 \\ q_4 &= m_P \bar{g} d \\ q_5 &= m_P r \end{aligned}$$

Choosing $r = .008 \text{ m}, m_R = 0.5 \text{ kg}, m_P = 1 \text{ kg},$ and $d = 0.15 \text{ m},$ the

uncompensated plant became

$$G(s) = \frac{0.3127}{0.04189s^2 + 0.3411} \quad (2)$$

which had roots at $s = 0 \pm j2.85,$ as shown in the Root Locus of the uncompensated system (Figure 3), and the system response to step input shown in Figure 4 (Figure 3).

Uncompensated system stability

As defined in [8], the Routh-Hurwitz Criteria states that there will be sign changes in the first column of the Routh array if a system's characteristic equation has roots with positive real parts and that the number of such roots will equal the number of sign changes in the first column of the array. The characteristic equation for the uncompensated system is $q(s) = 1 + D(s)G(s),$ where $D(s) = K$ and $G(s)$ is the plant transfer function defined in Error! Reference source not found. [8]. Therefore, the characteristic equation of the uncompensated system is $q(s) = 0.04189s^2 + 0.3411 + 0.3127K,$ which has the Routh array:

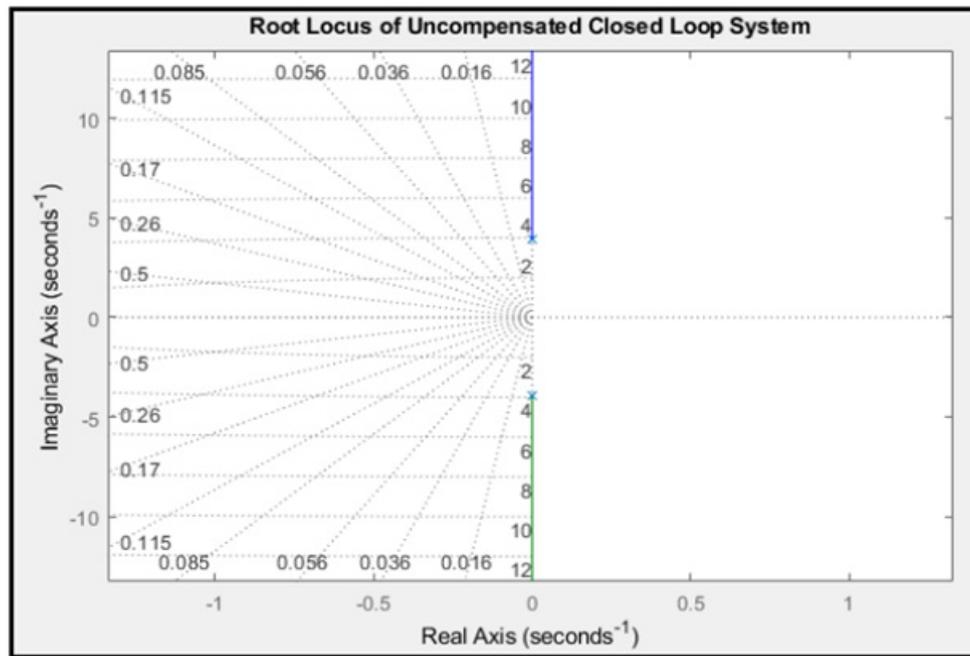


Figure 3: Root Locus of Uncompensated Closed Loop System.

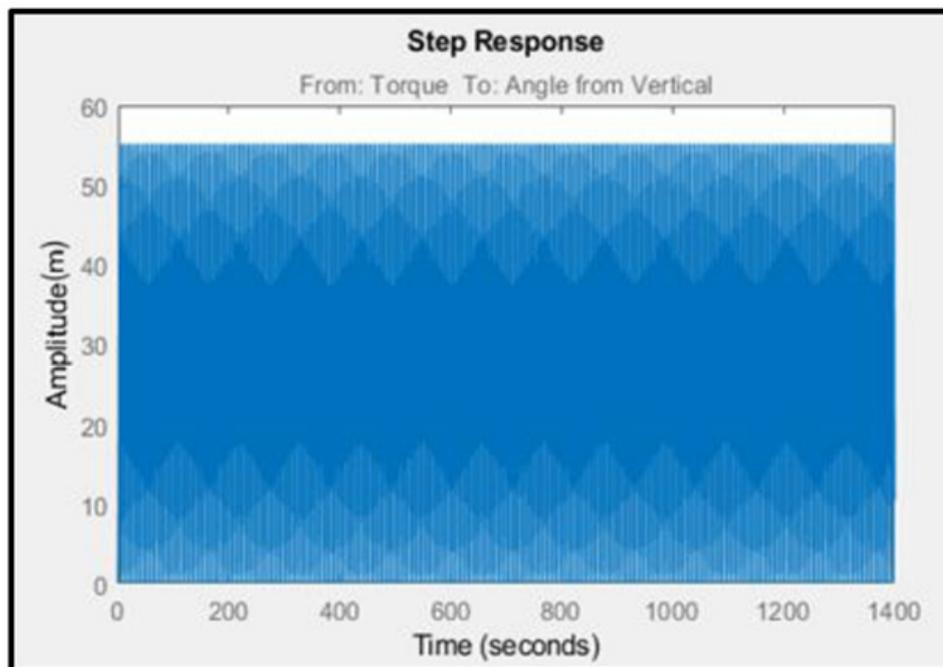


Figure 4: Plant Step Response.

$$\begin{matrix} s^2 & 0.04189 & 0.3411+0.3127K \\ s & \epsilon & \\ 1 & 0.3411+0.3127K & \end{matrix}$$

The ϵ is necessary to replace the 0 resulting from the lack of a first order term in the characteristic equation. The array demonstrates that the system is only marginally stable for any $\epsilon > 0$ and $K > -0.3411/0.3127 = -1.09$.

Controller Development

The system response to step input was examined since this most closely resembles the input likely to be experienced by the system. The desired system response should exhibit no more than 16% overshoot ($PO_{step} \leq 16\%$), have a settling time of no more than 2 seconds to reach 2% of the final value ($T_{s,step} \leq 2s$), and the final value should be no more than 5% from the desired final value ($e_{ss,step} \leq 0.05$).

Desired root location

The uncompensated system requires shifting to the left to achieve stability, so a lead compensator was developed. The minimum values of the system damping ratio and natural frequency were calculated from the design specifications. In [8], percent overshoot is defined as

$$PO = 100e^{-\left(\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)} \quad (3)$$

hence the minimum value for the damping ratio was.

$$\zeta \geq \sqrt{\frac{\ln^2(0.16)}{\pi^2 + \ln^2(0.16)}} = 0.504$$

Choosing $\zeta=0.6$ and solving the settling time (T_s) equation defined in [8],

$$T_s = \frac{4}{\zeta\omega_n} \quad (4)$$

for ω_n , the minimum value for the natural frequency was

$$\omega_n \geq \frac{4}{2*0.6} = 3.33$$

Selecting $\omega_n=4.5$ resulted in a desired root location of

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \\ = -2.7 \pm j3.6$$

Analytical development of a lead compensator

A lead compensator has the form

$$D(s) = K \frac{s+z_1}{s+p_1} \quad (5)$$

Usually, z_1 is chosen to be coincident with a plant pole to cancel the pole, which will shift the root locus left. Then p_1 is either chosen to create a more stable system or defined using the phase angle criterion if the plant has multiple poles. However, since the plant has complex roots, a value for z_1 cannot be chosen to cancel out existing poles in the plant. Therefore, $z_1=1$ and $p_1=10$ were selected through trial and error, which meant that $K=6.54$, using magnitude criterion, and the controller had the form

$$D(s) = 6.54 \frac{s+1}{s+10} \quad (6)$$

A Matlab script was used for iteration and the best step response that could be achieved utilizing only a lead compensator is shown in Figure 5.

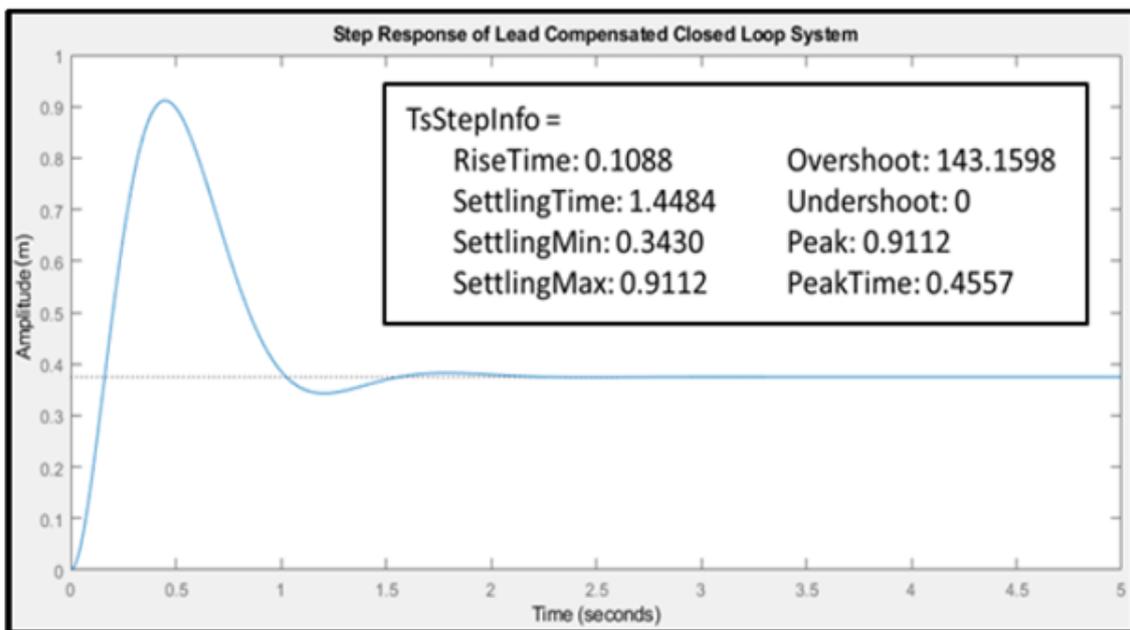


Figure 5: System Response to Step Input Using Lead Compensator.

Steady state analysis

To meet the steady state requirement the desired error constant (k_p) was defined by considering the steady state error equation from [8]

$$e_{ss} = \frac{1}{(1+k_p)} \quad (7)$$

which resulted in $k_{p,desired} \geq (1/e_{ss}) - 1 = 19$ to meet the chosen design specification. The system's error constant, again from [8], was $k_{p,system} = \lim_{s \rightarrow 0} D(s)G(s) = 0.599$ so, to achieve steady state, the system required an increase in gain of desired gain $\frac{k_{p,desired}}{k_{p,system}} = 31.70$.

D. Analytical Development of Lag Compensator

To apply the gain a lag compensator was developed. A lag compensator has the form

$$D(s) = k \frac{s+z}{s+p} \quad (8)$$

where $|z|/|p| < 1$, $|s+z|=p$, and the error constant, k_p , increases by a ratio of z/p . Keeping the previously determined gain, the lag compensator zero is defined [8] as $z=\alpha/10$ where α is the centroid of the uncompensated system. For the lead compensated plant $\alpha=-9$ ($\alpha=0$ for the uncompensated plant), so $z=-0.9$. The pole is defined as being significantly less than the ratio of the zero to the desired gain, so

$$p \ll \frac{|z|}{desired\ gain} = -0.54$$

Choosing $p=0.02$ results in the lag compensator.

$$D(s) = \frac{s+0.9}{s+0.02} \quad (9)$$

which, when applied to the lead compensated system, resulted in the step response shown in Figure 6 and the Root Locus shown in Figure 7 (Figure 6).

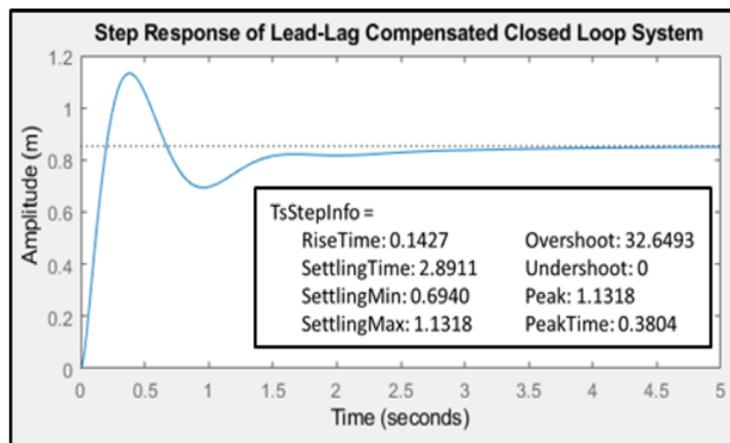


Figure 6: System Response to Step Input Using Lead-Lag Compensator.

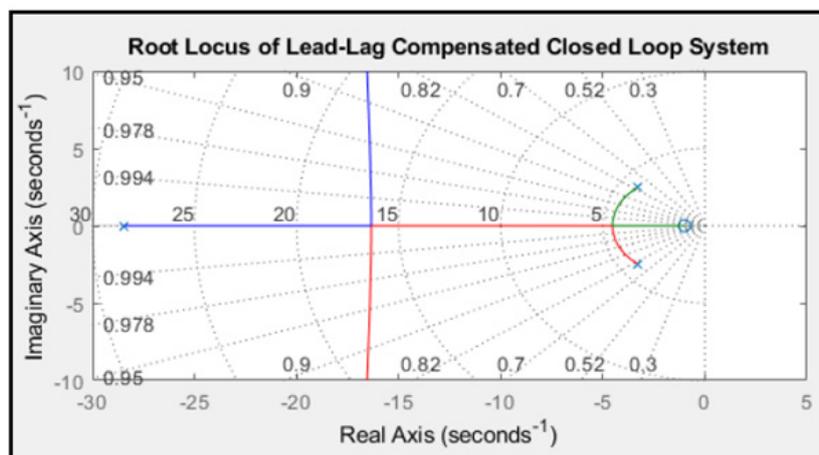


Figure 7: Root Locus of Lead-Lag Compensated System.

Simulation development of lead-lag compensator

Matlab's built in Control System Designer Optimization function was also used to generate a controller. Starting with the analytically derived gain, poles, and zeros as initial values, the optimization function was set to optimize each to achieve a settling time of less than 2 seconds and no more than 16% overshoot. The resulting lead-lag compensator was

$$D(s) = 39.688 \frac{(1 + 0.19s)(1 + 0.078s)}{(1 + 0.0063s)(1 + 0.011s)} \quad (10)$$

which had the step response shown in Figure 10. The fully compensated system had the Root Locus shown in Figure 8 and the Bode Plots shown in Figure 9 (Figure 8,9 & 10).

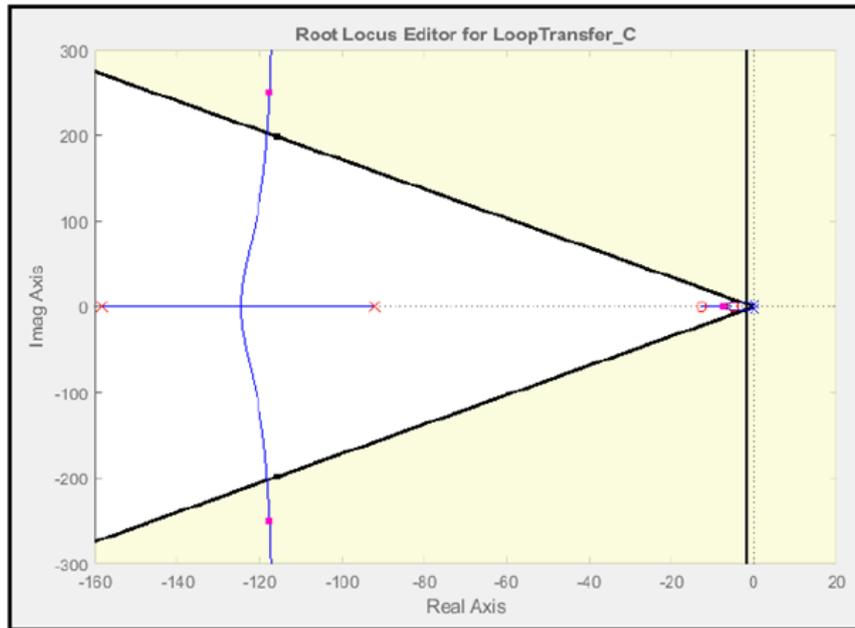


Figure 8: Root Locus of Simulation Derived Lead-Lag Compensator.

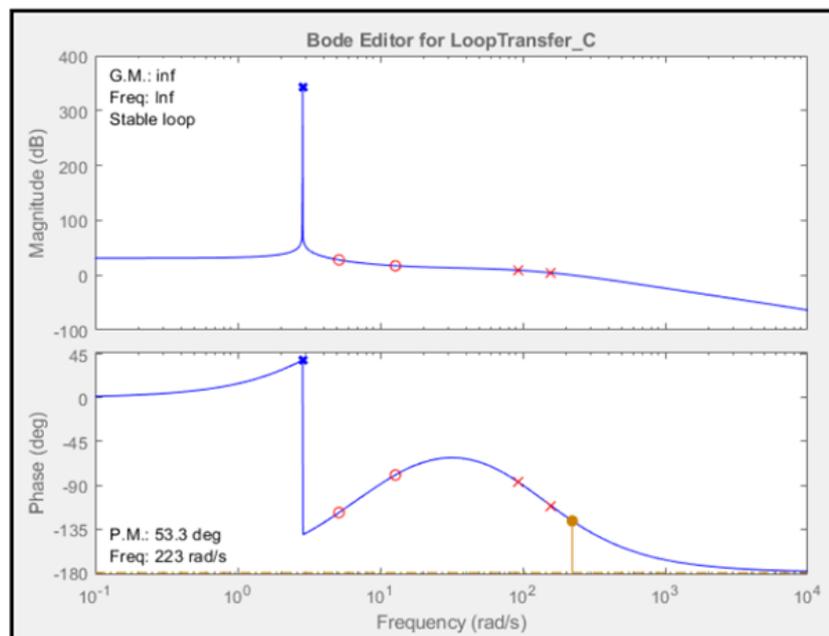


Figure 9: Bode Plots of Simulated Lead-Lag Compensator.

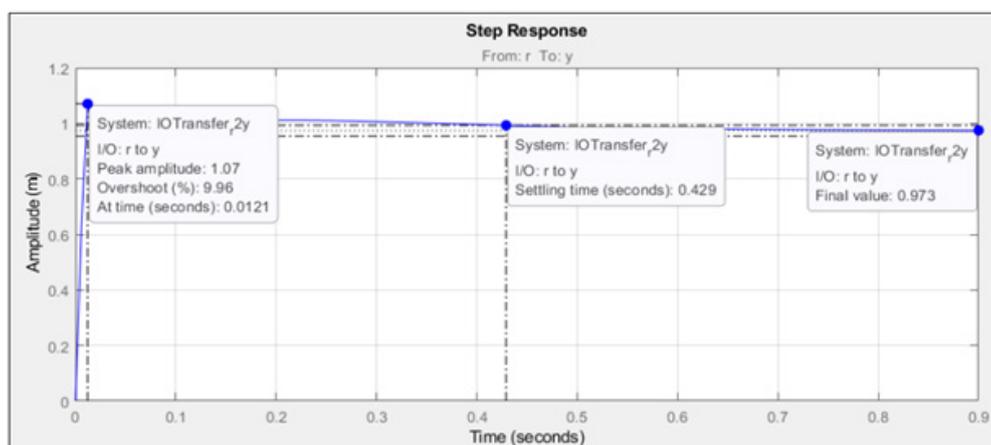


Figure 10: System Response to Step Input Using Simulation Developed Lead-Lag Compensator.

Conclusions and Discussion

Subjecting the system to step input illustrates the system response to external stimuli, such as gravity in the two-wheeled robot's case. The analytically developed lead-lag compensator resulted in a percent overshoot of $PO_{step}=32\%$ and a settling time of $T_s=2.89s$ which did not meet the transient design requirements. The lead-lag compensator developed using Matlab's Control System Designer did meet the desired design parameters with a percent overshoot of $PO_{step}=9.96\%$, a settling time of $T_s=0.43s$, and a steady state error of $e_{ss,step}=0.03$ in response to step input. Comparing the analytically developed lead-lag compensator to the one developed using Matlab's Control System Designer illustrates the challenges involved in analytically developing system controllers as well as the importance of understanding controller development methods. Due to the nonlinear nature of the plant equation more advanced controller development methods, such as sliding mode control as discussed in [1] [2], are more commonly applied to balancing a two wheeled robot than lead-lag compensators.

Future Work

Future work should focus on implementing disturbance rejection to compensate for unexpected bumps or inputs as well as integrating a sensor feedback mechanism to develop a more robust controller. It should be incorporated with a navigation system to provide balance during transit as well as while the robot is stationary. Finally, it should also include experimental testing of the simulation developed controller on a physical system to fully evaluate its ability to stabilize a two-wheeled robot.

Acknowledgement

None.

Conflict of Interest

None.

References

1. Y Zhou, Z Wang (2016) Robust motion control of a two-wheeled inverted pendulum with an input delay based on optimal integral sliding mode manifold. *Nonlinear Dynamics* 85(3): 2065-2074.
2. N Esmaili, A Alfi, H Khosravi (2017) Balancing and Trajectory Tracking of Two-Wheeled Mobile Robot Using Backstepping Sliding Mode Control: Design and Experiments. *Journal of Intelligent & Robotic Systems; Dordrecht* 87(3-4): 601-613.
3. M. Onkol, C Kasnakoglu (2015) Modeling and Control of a Robot Arm on a Two Wheeled Moving Platform. *Applied Mechanics and Materials* 789-790: 735-741.
4. C Kasnakoğlu, M Önkol (2018) Adaptive Model Predictive Control of a Two-wheeled Robot Manipulator with Varying Mass," *Measurement and Control* 51(1-2): 38-56.
5. CF Juang, MG Lai, WT Zeng (2015) Evolutionary Fuzzy Control and Navigation for Two Wheeled Robots Cooperatively Carrying an Object in Unknown Environments. *IEEE Transactions on Cybernetics* 45(9): 1731-1743.
6. M Stillman, J Olson, W Gloss (2010) Golem Krang: Dynamically stable humanoid robot for mobile manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA.
7. K Remell (2021) *Mathematical Modeling of a Two Wheeled Robotic Base (Mechanical Engineering Undergraduate Honors Theses)*, 26 May 2021. [Online]. Available: <https://scholarworks.uark.edu/meeguht/96/>. [Accessed 8 July 2020].
8. R C Dorf, R H Bishop (2011) *Modern Control Systems*, 12 ed., Upper Saddle River, New Jersey: Pearson Education.
9. MM Rahman, Ashik-E-Rasul, H Nowab Md Aminul, M Hassan, I M Hasib, et al. (2011) Development of a two wheeled self balancing robot with speech recognition and navigation algorithm in *International Conference on Mechanical Engineering, BUET, Dhaka*, 2016.