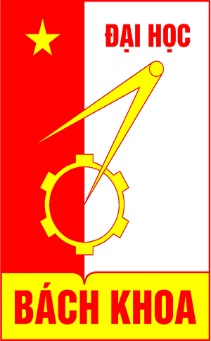
**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**



**Digital Control**

**SELF BALANCING**

**TWO WHEELERS**

**CLASS ID:** **143962 – EE4435E**

**GROUP 4:**

**ĐỖ QUANG THÀNH –** [Thanh.dq202770@sis.hust.edu.vn](mailto:Thanh.dq202770@sis.hust.edu.vn)

**NGUYỄN TUẤN DƯƠNG –** [Duong.nt202740@sis.hust.edu.vn](mailto:Duong.nt202740@sis.hust.edu.vn)

**NGUYỄN ĐÌNH TOÀN –** [Toan.nd202775@sis.hust.edu.vn](mailto:Toan.nd202775@sis.hust.edu.vn)

**VÕ KHÔI THÀNH SƠN –** [Son.vkt202768@sis.hust.edu.vn](mailto:Son.vkt202768@sis.hust.edu.vn)

**Control engineering and Automation**

|  |  |
| --- | --- |
| **Instructing teacher:** | Vu Thi Thuy Nga |
| **Department:** | Automation |
| **University:** | Hanoi University of Science and Technology |

**HÀ NỘI, 1/2024**

**Acknowledgement**

Through this course and research, we have had the chance to learn the basics of digital control. In addition, we have the opportunity to cooperate with colleagues to work on a new and interesting topic which can contribute greatly to society in the future. With this, we desire to express our gratitude towards our instructor, Mrs Vu Thi Thuy Nga.

**Abstract**

The emerging field of self-balancing two-wheelers marks a leap forward in autonomous mobility, blending cutting-edge technology with innovative control mechanisms. The heart of these vehicles are gyroscopes integrated with sophisticated control algorithms. This combination enables the vehicles to autonomously maintain balance, a critical feature for stability in motion. This report provides an in-depth exploration of the principles underlying the stability and control of these two-wheelers. It examines the role of accelerometers, gyroscopes, and advanced feedback control systems, which are essential for precise navigation and equilibrium maintenance.

This abstract offers a comprehensive overview of the technological innovations and operational dynamics of self-balancing two-wheelers. It underscores their significance in the broader context of evolving transportation technologies and their role in defining the future of autonomous mobility.

**CONTENTS**

[**I.** **Introduction** 7](#_Toc156410742)

[**II.** **Control problem** 8](#_Toc156410743)

[**III.** **System model** 9](#_Toc156410744)

[3.1. Model definition 9](#_Toc156410745)

[3.2. Calculation of the system model 9](#_Toc156410746)

[**IV.** **Control strategy** 12](#_Toc156410747)

[**V.** **Simulation and experiment result** 16](#_Toc156410748)

[5.1. Simulation result: 16](#_Toc156410749)

[5.2. Experiment result: 17](#_Toc156410750)

[**REFERENCE** 19](#_Toc156410751)

**FIGURE LIST**

[Figure 1: Simplified sketch showing the tilt angle, ψ and the position x 9](#_Toc156410797)

[Figure 2: Exposure of the simplified inverted pendulum 9](#_Toc156410798)

[Figure 3: simulation of the system model 17](file:///D:\Duong\hoc\BKA\20231_dk%20he%20thong%20so\report_group4.docx#_Toc156410799)

[Figure 4: position and angular response of the system simulation 17](file:///D:\Duong\hoc\BKA\20231_dk%20he%20thong%20so\report_group4.docx#_Toc156410800)

**TABLE LIST**

[**Table 1: Model parameters** 16](#_Toc156410813)

1. **Introduction**

Maintaining an inverted pendulum mounted on a cart in upright position by accelerating the cart back and forth represents a classic control theory problem. A self-balancing robot (SBR) is an adapted version of an inverted pendulum. The main difference is that an SBR consists of only one moving part and it can be controlled by the motors mounted at its base. The robot is described by a single input multiple output (SIMO), unstable system with nonlinear dynamics which makes it hard to control. This is a classic control theory problem that has received numerous solutions during the years. In most of these solutions the robot is actuated by geared DC motors. The report is focused on building and controlling an SBR using stepper motors. A stepper motor has two main advantages over DC motors. The first one is that no additional gears are needed due to the high output torque it can deliver. The second one is that stepper motors rotate in discrete steps which yields high position control accuracy and allows open loop control. A full state feedback, linear quadratic regulator (LQR) controller is approached since it is fully customizable on the needs of the process, making it an ideal control choice for a self-balancing robot. The mathematical modeling of an inverted pendulum and the control theory behind an LQR controller are presented in the theoretical aspects. The adaptation of the SBR to the mathematical model of the inverted pendulum is next described. The main components used for building the robot are also presented. Last but not least, an experiment is performed and the results are compared to numerical simulation data.

1. **Control problem**

The control of a self-balancing two-wheeler presents a unique set of challenges, primarily rooted in maintaining equilibrium while navigating varied terrains and responding to user inputs. At the core of this control problem is the need to constantly counteract the force of gravity to prevent the vehicle from tipping over. This task is similar to the classic control problem of an inverted pendulum, where the goal is to keep a pendulum upright on a moving base.

To achieve balance, the system relies on a combination of sensors, such as gyroscopes and accelerometers, which continuously monitor the vehicle's tilt angle and angular velocity. These sensors provide real-time feedback to a microcontroller, which computes the necessary adjustments. The primary control action involves varying the speed and direction of the wheels to generate a balancing force. For instance, if the vehicle tilts forward, the wheels must move forward to realign the center of gravity over the base. This control must be precise and rapid, as delays or inaccuracies can lead to instability.

Another layer of complexity arises from user control. The vehicle must not only balance itself but also respond to the user's direction, such as moving forward, backward, or turning, without compromising stability. This requires a sophisticated control algorithm that can seamlessly integrate balance control with user inputs. Furthermore, external factors such as terrain changes, obstacles, and dynamic loads (like shifting body weight) introduce additional variability, demanding adaptability and robustness in the control algorithm.

Overall, the control of a self-balancing two-wheeler involves a delicate and continuous process of sensing, decision-making, and actuating, all working in concert to maintain a stable and responsive ride experience.

1. **System model**
   1. Model definition

The cart is represented by the motor shafts and wheels. The pendulum is represented by the robot body (motor stators, motor brackets, threaded rods, plastic sheets, electronics). The pendulum length is measured from the axis of rotation (the robot rotates about the motor shafts) to the plane in which the Center of gravity (COG) of the robot body lies.

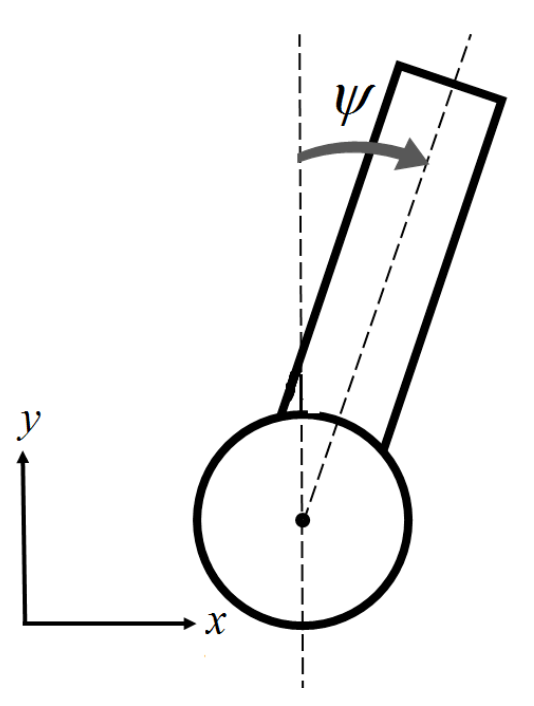


Figure : Simplified sketch showing the tilt angle, ψ and the position x

* 1. Calculation of the system model

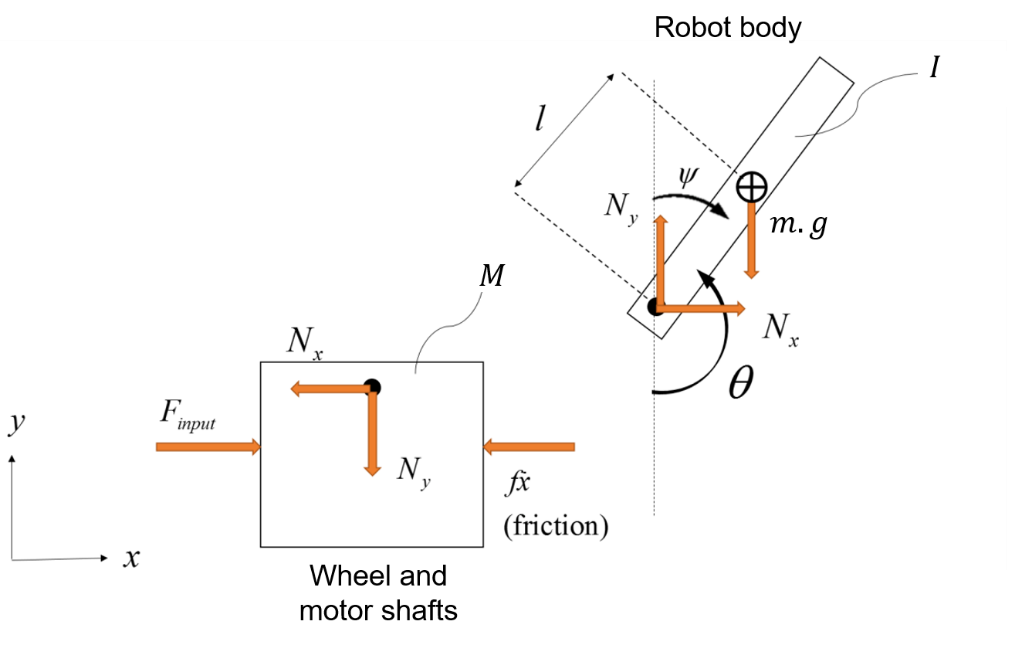


Figure : Exposure of the simplified inverted pendulum

**Equations for the wheel and motor shafts:**

The equation of the wheel and the motor shafts is:

Where, is the applied force.

𝑀 is the mass of the wheel and motor shafts.

𝑓 is the constant of friction.

is the contact force between the wheel and the robot body.

**Equations for the robot body:**

The sum of forces in the x direction on the robot body is:

Where, is the mass of the robot body.

is the distance to the center of mass.

is the angle between a vertical line and the robot body.

The sum of all forces perpendicular to the robot body is:

Where, is the force in y direction.

is the gravity constant.

Summarization the torque acting on the center of the robot body, we have:

Where, is the moment of inertia of the robot body

Combine the above equations, we obtain:

From (1) and (2), we have:

From (2) and (3), we have:

**Linearizing the equations:**

Equation (III-5) and (III-6) are necessary to get transfer functions for the position x and the angle deviation ψ. To compute the transfer functions, the equations need to be linearized. A proper equilibrium point would be when the pendulum is in upright position. The angle will represent the deviation of the pendulum from the equilibrium. The following approximations for small deviations will be used in the nonlinear equations (III-5) and (III-6).

Linearization with (III-7), (III-8) and (III-9) in (III-5) and (III-6) leads to the following approximated linear equations where has been substituted for the more general control effort .

**State space modeling:**

Rewriting equation (III-10) and (III-11), the following mathematical model obtains:

where, ;

;

;

By denoting the states of the system as:

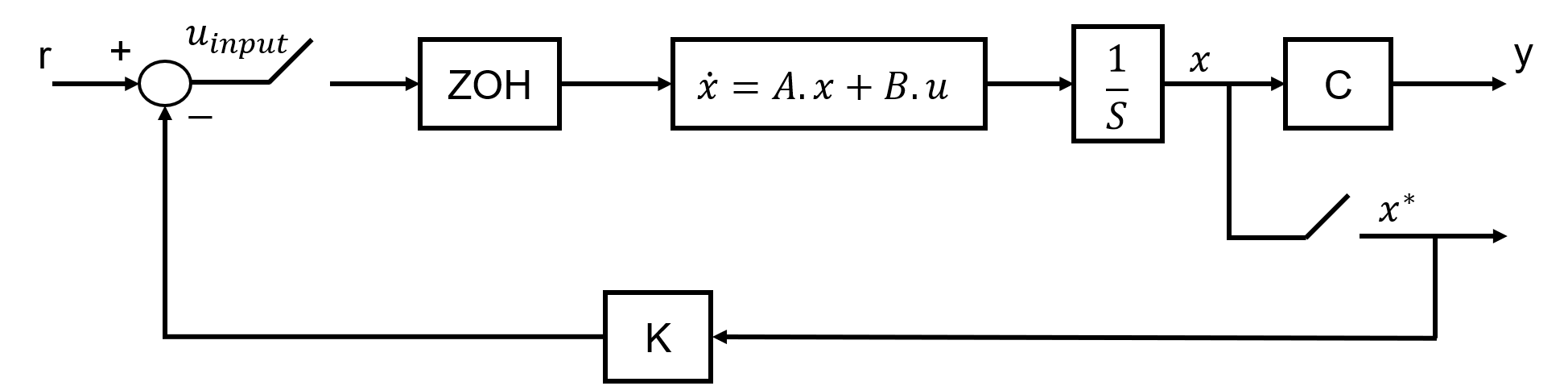
And output is:

The following state-space representation is obtained:

where, A = ; ;

# **Control strategy**

A Linear Quadratic Regulator (LQR) – controller can be applied with the states of the dynamical system described by the following state space model including the matrices A, B and C:



A cost function is set by the following equation:

Where, ;

Q is a positive definite matrix that penalizes the state vector;

R is a positive definite matrix that penalizes the control input.

The matrices Q and R are selected to minimize the value for the cost function J.

The control effort is given as:

Where, is a vector containing the set point for each state.

K is the gain matrix defined as:

Where, P is solved from the Algebraic Riccati Equation:

Since the state space system is observable and controllable, P has a unique solution so that the closed loop system poles are strictly in the right half plane.

**Discretization of continuous system:**

The state space model of the self-balancing two wheelers is given by:

; with A, B and C calculated in the previous section.

From equation (IV-5)

With

With k = 0, t = k.T = 0 x(0)

With k = 1, t = T

Hence, we obtain:

Where,

**Discrete LQR controller:**

Consider the discrete time system:

Where, , N is the length of the time horizon.

The cost function is chosen as a quadratic sum of the states and control inputs:

Where, is the weighting matrix for the states.

is the weighting matrix for the control input.

is the weighting matrix for the final state at last time step N.

The optimal control input sequence , k = 0,1,…, N-1, for linear system (IV-7) to

minimize the quadratic cost function (IV-8) is:

We use the dynamic programming (recursive) approach for solving the LQR problem. The optimal control problem is solved recursively in time using the idea of the cost-to-go function :

where, is the cost accumulated from the instant till the end.

From (IV-8) and (IV-10), we obtain: , and the optimal cost-to-go function can be computed by:

where ≥ 0 is known as the Riccati matrix

Substitute (IV-7) in (IV-12), we obtain:

(IV-13)

The optimal control input is computed by minimizing :

is obtained by equating the gradient of with respect to to zero:

where, the optimal feedback gain is:

Substitute by in (IV-13), gives the optimal cost-to-go function:

Where, is the Difference Riccati Equation (DRE) in which the Riccati matrix is computed recursively from a terminal Riccati matrix

**Control signal:**

The most common way to control the output torque of a stepper motor is by controlling the stator winding currents by applying the necessary amount of voltage using pulse width modulation. However, the drivers used to control the stepper motors mounted on the real-life platform can only be used for speed control. The algorithm for obtaining a speed from linear force will be described next. The control effort is split between the two motors driving the robot:

Where, is the force generated by one motor. Both of these forces must pull the same mass.

The linear acceleration, denoted by , is obtained as:

The integral of the linear acceleration gives the linear velocity of the SBR’s base:

A discrete time approximation of the integral used in the implementation of the control algorithm on the microcontroller is given by the following equation:

Where, k is the sample number.

This approximation can lead to an error accumulation, but it can be neglected for short running times. Knowing the linear velocity of the robot base, the impulse frequency which is equal to the motor speed measured in steps per second is computed using the following formula:

Where, is the total number of steps that the motor needs to perform a revolution. A negative value of denotes a backward rotation of the wheels, while a positive value of denotes a forward rotation of the wheels.

From the given details, we can deduce the time span associated with each step, essentially representing the duration of the PWM (Pulse Width Modulation) signal dispatched to control the stepper motor. This temporal interval serves as a crucial determinant in the precision and timing of the stepper motor's operation:

# **Simulation and experiment result**

* 1. Simulation result:

Within the context of the hardware proposal, we find a comprehensive depiction of the parameters associated with our self-balancing two-wheel cart, meticulously organized in the form of the following table. This detailed presentation encapsulates the essential specifications crucial for the proper functioning and performance evaluation of our innovative self-balancing cart:

**Table 1: Model parameters**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Symbol** | **Value** |
| Mass of the robot body | m | 0.412 kg |
| Mass of the wheels and motor shafts | M | 0.555 kg |
| Distance from rotation axis to COG | l | 0.07 m |
| Mass moment of inertia of the robot body | I | 0.0006 kg.m^2 |
| Gravitational acceleration | g | 9.81 m/s^2 |
| Wheel radius | r | 0.03 m |
| Friction | f | 0.005 |
| Step per revolution | n | 200 |

Using the parameters presented in **Table 1**, the following numerical state space representation of the SBR is obtained:

Discretize the system, we have:

Utilizing the derived system model, we proceed to simulate the system within the Matlab Simulink environment:

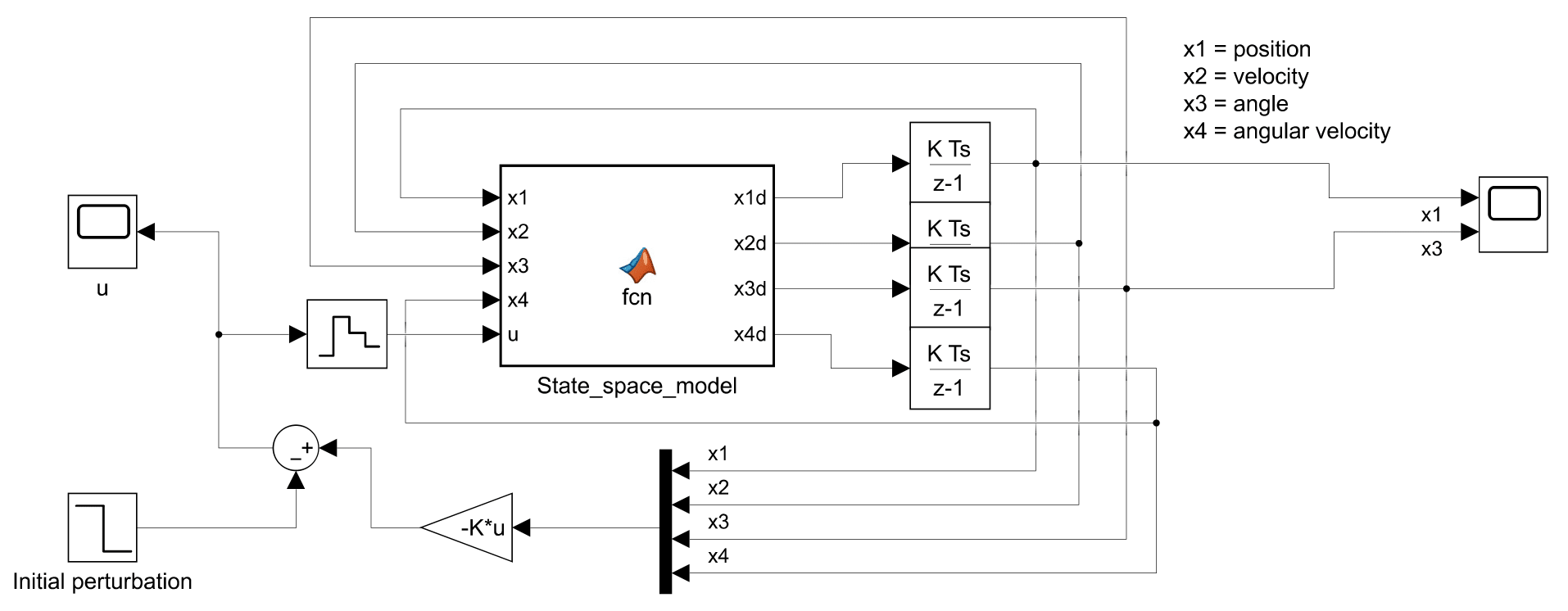


Figure : simulation of the system model

We assess the efficacy of our system model by examining the position and angular response of the model, utilizing an initial perturbation as the input to the system:

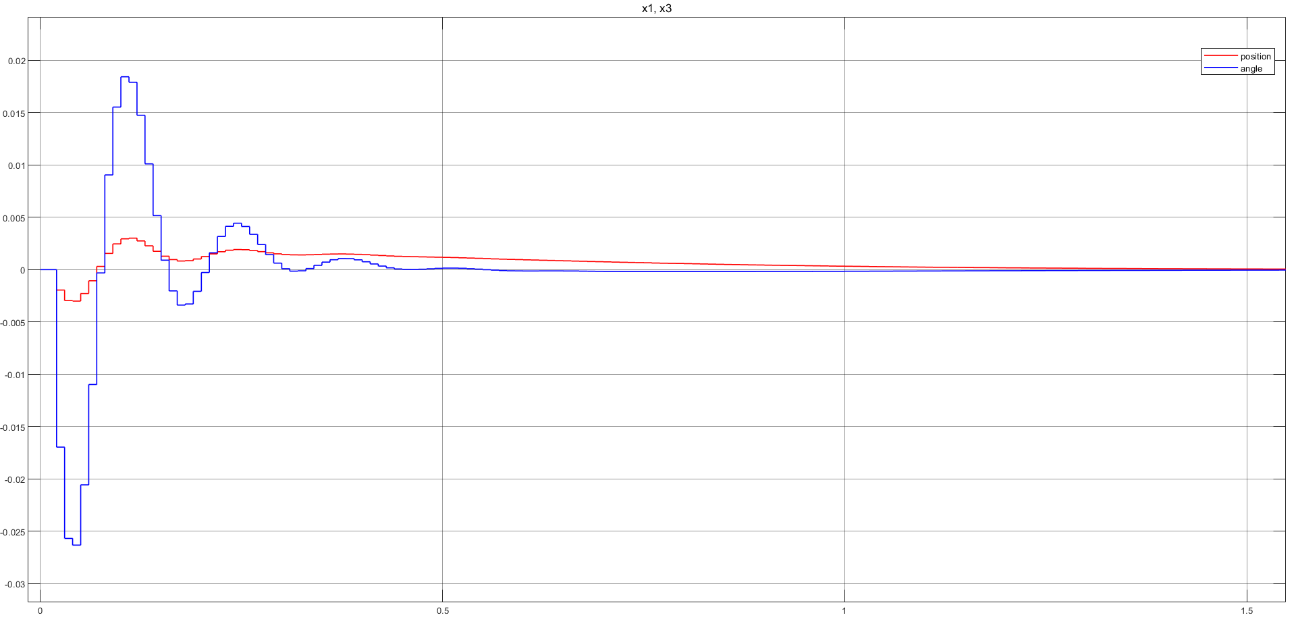
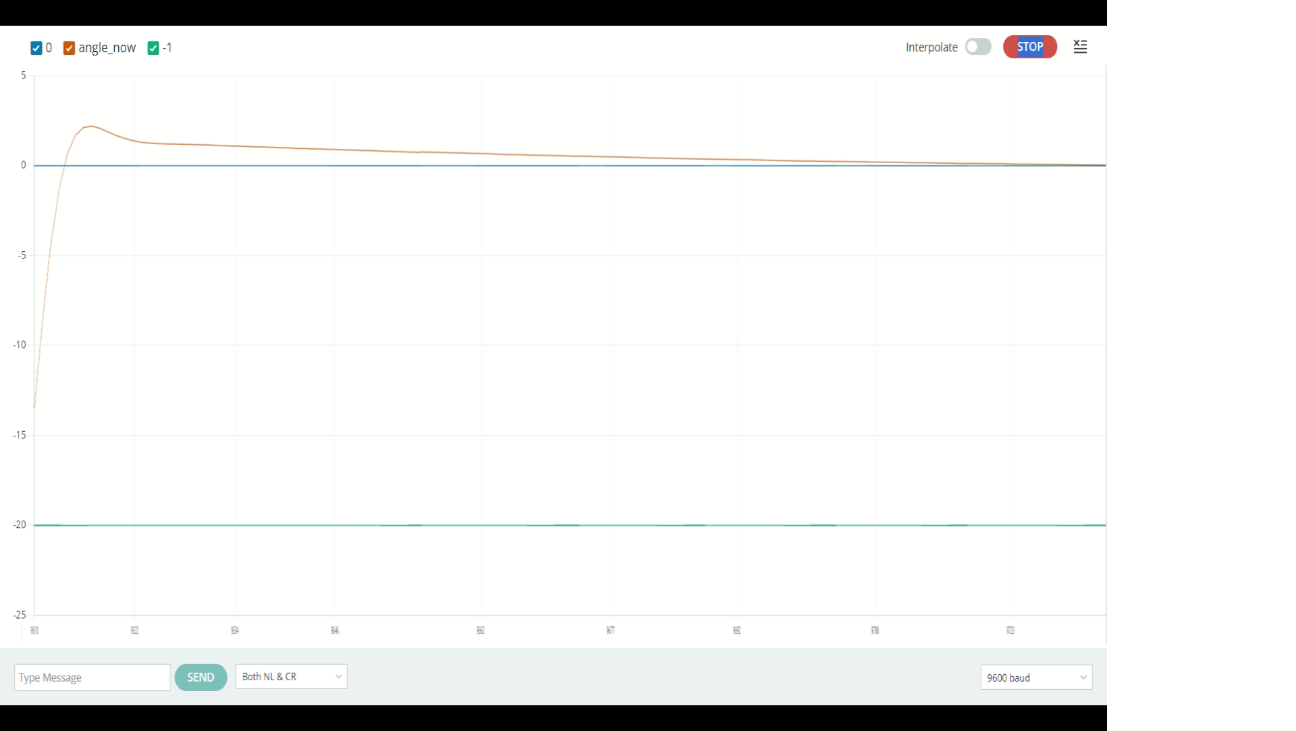


Figure : position and angular response of the system simulation

* 1. Experiment result:

The system is translated into a physical model by employing Arduino along with the proposed hardware components.



Through MATLAB Simulink model and simulation in Arduino IDE, we obtain the angular response as shown below:

* 1. Conclusion:

We are able to balance the robot with minimal change of the angle, however, the vibration problem still remains. That is the reason why a mattress is used under the robot to temporarily remove the vibration. In the future, we hope to improve our product by eliminating this issue completely.

# **REFERENCE**

[1] Linear and angular position control of a custom built stepper motor driven self-balancing robot, Casian-George Iacob Automation Department Technical University of Cluj-Napoca, Cluj-Napoca, Romania, 2020.

[2] Two-Wheeled Self-Balancing Robot, Hanna Hellman, Henrik Sunnerman, 2015.