Self-Balancing Robot

"Design & develop a Self-Balancing Robot that can programmable by Matlab Simulink."

Student: Ahmad Kheirandish Gharehbagh

Supervisor: Dr.Mohammad Mansouri

Electrical Engineering Department of Shahed University

Introduction

A balancing robot is a common demonstration of controls in a dynamic system. Due to the inherent instability of the equilibrium point, appropriate controllability and observability measures must be undertaken to stabilize the system about the desired equilibrium point. We will investigate this system by developing both an analytical and experimental model of the system and analyze the dynamic characteristics. [1]

Self-balancing robots have been a topic of interest of many researchers, students and hobbyists worldwide. In essence, it is an inverted pendulum on wheels, a derivative of the inverted pendulum on a cart. Unlike traditional robots, which are in a constant state of equilibrium, the robot is a naturally unstable system [2]. Its design is more complex, as it needs to be actively controlled to maintain its upright position, however, it benefits from being able to turn on the spot.

The primary incentive of the project is to develop general understanding of control theory. For the last few decades, "the inverted pendulum has been the most popular benchmark, among others, for teaching and research in control theory and robotics [3]." Hence, developing a self-balancing robot is the ideal platform to put into practice what has been covered in Control Systems lectures. It would also be interesting to see the differences between the behaviors in practice compared to simulations. Furthermore, the material and methods learnt have a wide array of applications; for example, inverted pendulums have been used to model human locomotion, which then was used to develop bipedal robots [4].This project aims to design, construct and program a self-balancing robot in Matlab Simulink environment.

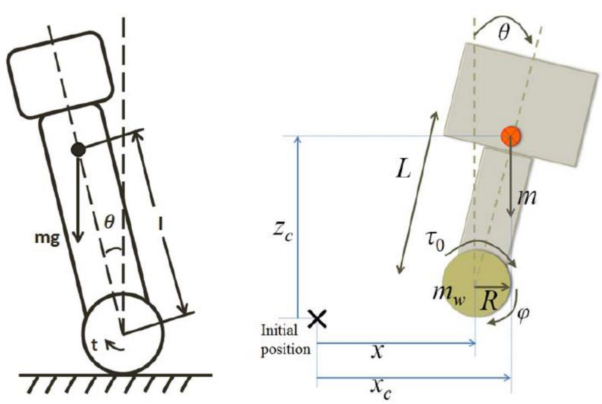


Figure 1. Free body diagram of the balancing robot and its parameters [1]

Design Self-Balancing Robot

The balancing robot that has been built is basically a simple inverted pendulum on two wheels. This is one of the most widely used examples in control classes. For this project, I simulate the system in MATLAB/Simulink.

The Equipment and electronic modules that has been used for this project is:

1. Arduino UNO Board
2. MPU 6050 module (This sensor contains a MEMS accelerometer and a gyro in a single chip)
3. Robot Chassis
4. 2 Wheels and DC motors
5. DC motor driver (L298N)

To fulfil the purpose the following method will be used:

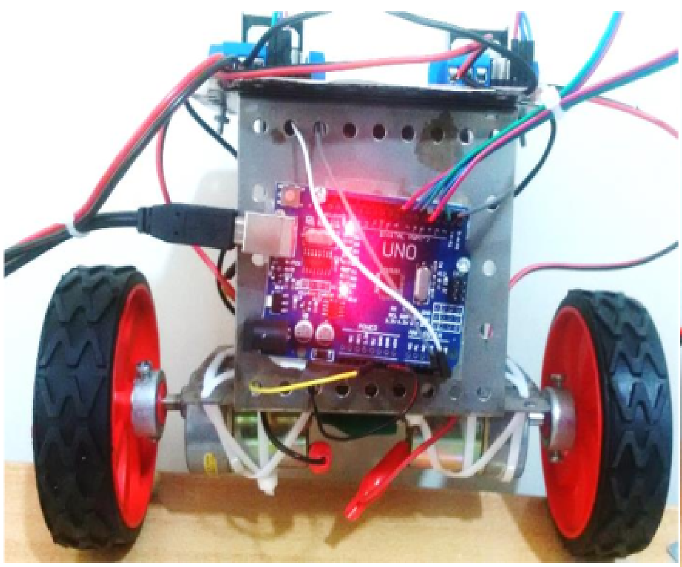
- Derive dynamical equations based on theory of the inverted pendulum
- Form transfer functions for the angle deviation and position
- Find a controller that can control these two conditions
- Set up requirements for the demonstrator
- Design a demonstrator that fulfils these requirements, investigate the boundaries of the control signal

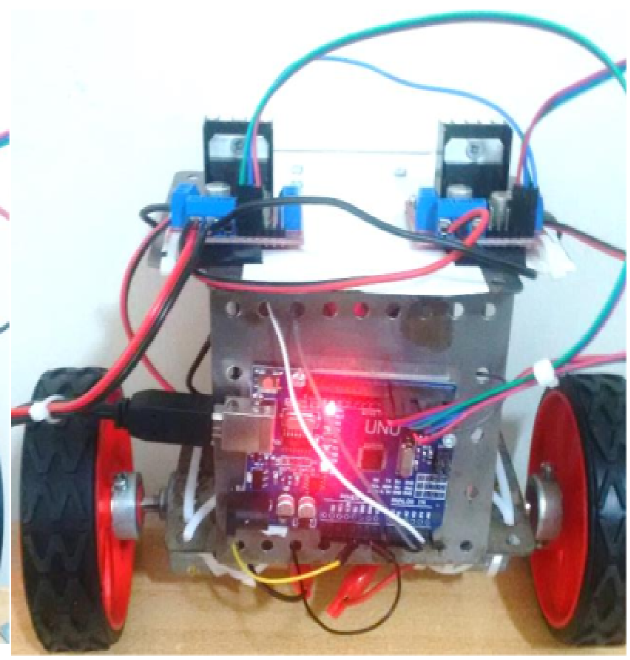
Chosen error sources will be investigated:

- Investigate the accuracy of the sensor that delivers the angular data

With an accurate model of the system and a functioning demonstrator this provides a platform for experiments in a simulated environment. The simulated model in comparison to the demonstrator will be validated by implementation of a PID-controller in both in order to compare impulse responses.

Photos of Primary Constructed Self-Balancing Robot :





Design Controller :

The control algorithm that was used to maintain it balance on the autonomous self-balancing two wheel robot was the PID controller. The proportional, integral, and derivative (PID) controller, is well known as a three term controller.

The input to the controller is the *error* from the system. The Kp, Ki, and Kd are referred as the proportional, integral, and derivative constants (the three terms get multiplied by these constants respectively). The closed loop control system is also referred to as a negative feedback system. The basic idea of a negative feedback system is that it measures the process output *y* from a sensor. The measured process output gets subtracted from the reference *set-point* value to produce an *error*. The error is then fed into the PID controller, where the error gets managed in three ways. The error will be used on the PID controller to execute the proportional term, integral term for reduction of steady state errors, and the derivative term to handle overshoots. After the PID algorithm processes the error, the controller produces a control signal *u*. The PID control signal then gets fed into the process under control.

The process under PID control is the two wheeled robot. The PID control signal will try to drive the process to the desired reference set point value. In the case of the two wheel robot, the desired set-point value is the zero degree vertical position. The PID control algorithm can be modelled in a mathematical representation.

References

1. Ye Ding1, Joshua Gafford1, Mie Kunio2 , Modeling , Simulation and Fabrication of Self Balancing Robot, Advanced System Dynamics and Control,

2. D. Caulley, N. Nehoran and S. Zaho, “Self-balancing robot,” 2015. [Online]. Available: https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2015/dc686_nn233_hz263/final_project_webpage_v2/dc686_nn233_hz263/. [Accessed 02 April 2017].

3. O. Boubaker, “The inverted pendulum: A fundamental benchmark in control theory and robotics,” IEEE, 2012.

4. D. Goldman, N. Gravish, S. Sharpe and H. Li, “Nonlinear Dynamics of Human Locomotion,” Georgia Institute of Technology, Shanghai Jiao Tong University, 2012.