Wrocław University of Science and Technology
Faculty of Electronics

Two-Wheeled Self-Balancing Robot - design and control

Author: Jakub Rotkiewicz

Class: Intermediate Project

Supervisor: Ph.D. Witold Paluszyński

Date: February 2, 2021

Abstract

The aim of the project was to build a small, two-wheeled robot using the off-the-shelf components, as well as designing and implementing the required controller for the self-balance capability. The goal was to get a robust system for the future further expansion and experiments.

The project is not completely successful yet. Although it will be, because of previously mentioned aim to further expanding and experimenting with the project. Up until now the robot was built and the PID controller was successfully implemented. The only thing left is fine tuning the parameters.

1 Introduction

Self-balancing robots are already widely investigated subject by scientists as well as hobbyists. The subject is interesting because it shows the great possibilities of control engineering. There are plenty approaches to this particular project, presented in the literature as well as in the Internet. Generally, the robots must consist of two wheels with two electrical motors mounted to some base, the motors' drivers, direct or indirect pitch angle sensor and obviously a power supply. Additionally, there is no one unique way to design the controller. People often use PID controller because it is the simplest one. Although the system is non-linear, it can be linearized around an equilibrium point. Other linear, model-based controllers such as Pole Placement or Linear Quadratic Regulator are also used in many applications. It is up to the designer what parts and what controller one will choose.

This project was created basing on stepper motors and their drivers. The pitch angle was calculated indirectly from acceleration and rotation speed provided from the IMU sensor. The Kalman Filter was implemented to merge both sensors' data. The controller used in the project was simple PID.

2 Materials and Methods

The following things were done in the project:

- full construction of the robot
- soldered prototype board with sockets for all modules - allows to reuse/change everything in the robot
- 11.1V Li-Ion 3s1p battery pack with Battery Management System was built and tested
- implemented and tested for various parameters Kalman filter
- implemented and tested for various parameters closed-loop control - PID
- software for stepper drivers, IMU, Bluetooth module was written and tested
- whole system was deployed on STM32 NUCLEO-F411RE
- Android application for online parameters tuning

2.1 Construction

The robot consists of two levels. The first one holds the prototype board with all the components and the second one is empty on top, but Li-Ion battery pack is mounted on the bottom of it. That way one can experiment with different weights that can be put on top. Both levels are connected to stepper motors with four M4 threaded rods, nuts and washers. The prototype board includes sensors and modules listed below:

- STM32 NUCLEO-F411RE
- two A4988 Stepper motor drivers

- HC-05 Bluetooth module
- MPU6050 IMU sensor
- 3.3V to 5V logic converter for I2C
- 7805 Voltage Regulator IC

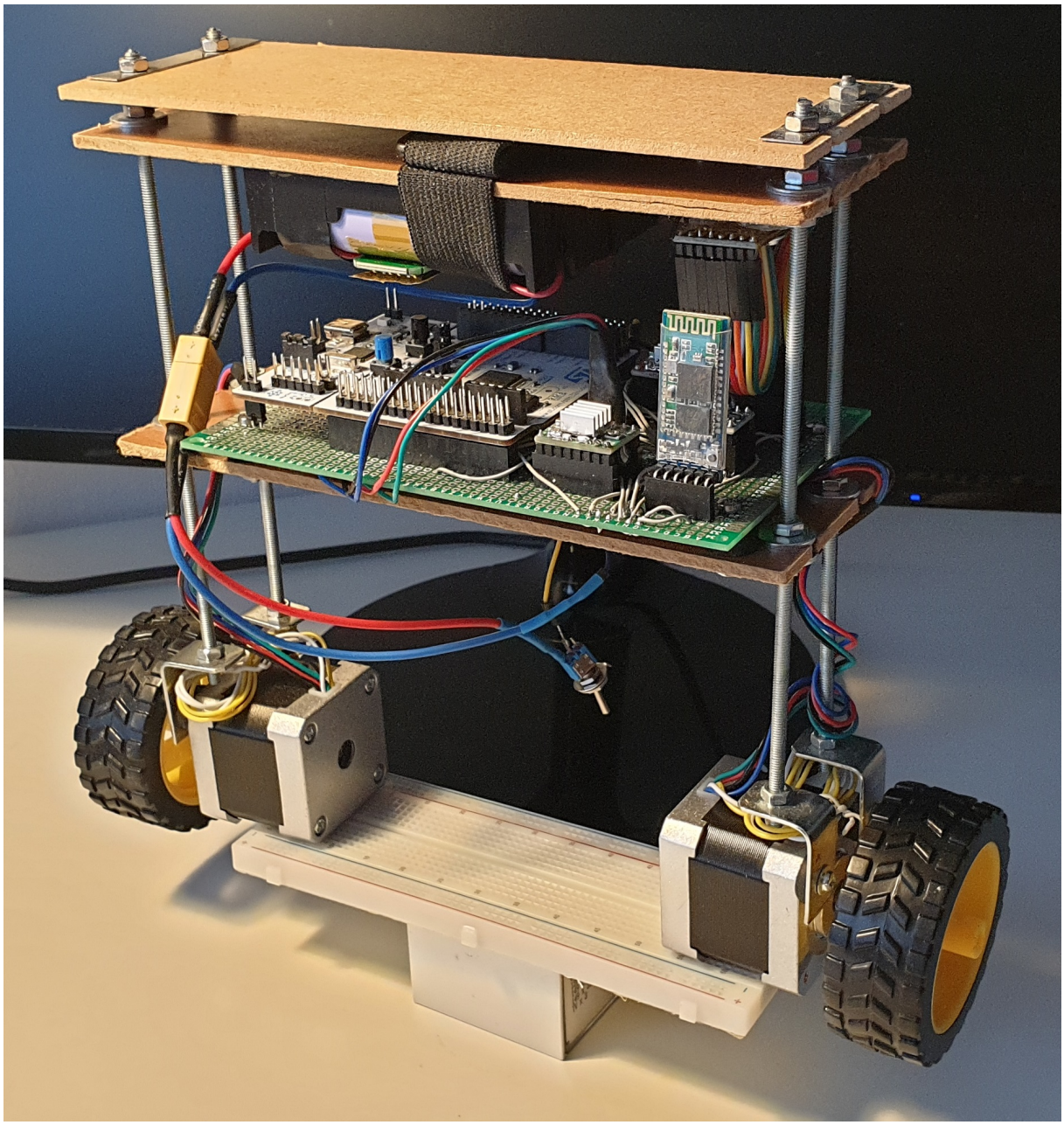

Figure 1 – Self-balancing robot

Before mentioned Li-Ion battery pack was assembled from scratch. First try was with 7.4V 2s1p Li-Ion battery pack, two of them were created. Unfortunately it was a mistake, since Stepper motor drivers require minimum 6V to run the motors. The full potential of the battery pack was not used, because when it went below 6V the drivers stopped working properly. So next battery pack was made. This time 11.1V, which allows the system to work properly from the fully charged to fully discharged battery, without stopping in the middle.

2.2 Kalman filter

To achieve the best pitch angle estimation the sensor fusion was applied in the system. Acceleration and rotation speed from IMU were merged together in the Kalman filter. The code for the filter was taken from Lauszus [2012] and rewritten from C++ into C. The very similar approach was also presented in article Radin Charel et al. [2016]. In the main program the angle is calculated 200 times a second. The problem was to find proper coefficients for the filter, but basing on the before mentioned article a decent ones were found.

2.3 PID control

The PID controller is responsible for calculating the input signal for the stepper motors that will stabilize the robot in upward position and remain it like that despite some small external disturbances. A very good explanation as well as the code was found here Lab [2020]. The PID is calculated at frequency 500Hz. Even though the proper gains weren't found yet.

2.4 Testing

The robot required six different parameters to work properly. P, I, D for the controller and Q_{angle}, , Q_{bias}, $R_{measure}$ (in the application A, B and R respectively) for the Kalman filter. More about filter gains can be found in cited articles. To make life easier a simple Android application was created using website MIT to send parameters to the robot via Bluetooth. It is shown in the Figure 2. To monitor the results the *STMStudio* project was made, Figure 3.

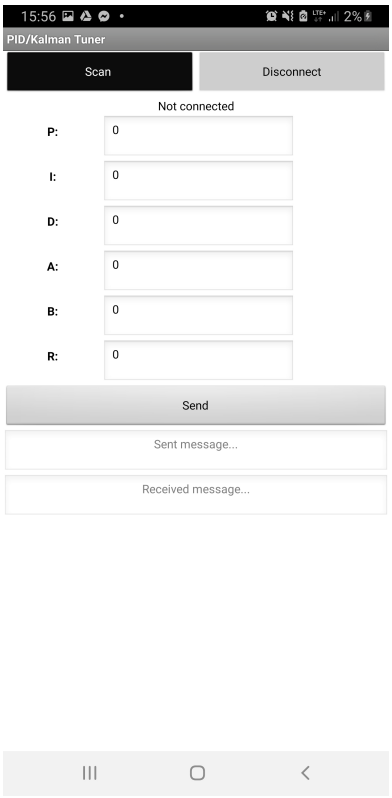

Figure 2 – PID/Kalman Tuner

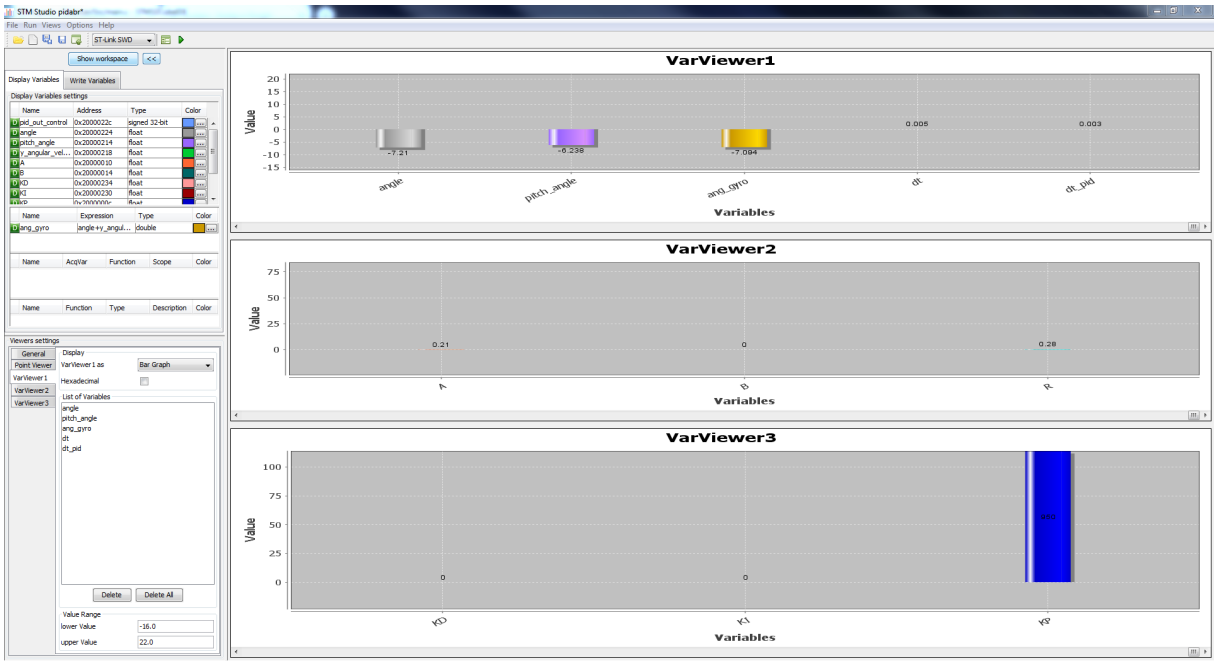

Figure 3 – Parameters monitoring

3 Results

The robot does not balance yet. One can say that it is a failure and probably would be right. Nevertheless, the huge amount of work was done and many mistakes were fixed on the way. The system is powered up correctly. All the sensors and modules work properly. The communication (I2C, Bluetooth with USART) work perfectly fine. All the testing equipment is reliable and works as it should. The one thing left is to find out why tuning the parameters is so hard for this particular robot. Maybe the construction is unfortunate or I am missing something. Although, the project will be continued, as stated at the beginning, the aim is further expansion, experiments and investigation, so it will be done.

References

MIT App Inventor. Accessible in the internet: https://appinventor.mit.edu/.

Phil's Lab. *PID Controller Implementation in Software*, May 2020. Accessible in the internet: https://youtu.be/zOByx3Izf5U.

Kristian Lauszus. *A practical approach to Kalman filter and how to implement it*, September 2012. Accessible in the internet: http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/.

S. E. Radin Charel, E. H. Binugroho, M. A. Rosyidi, R. S. Dewanto, and D. Pramadihanto. Kalman filter for angle estimation using dual inertial measurement units on unicycle robot. In *2016 International Electronics Symposium (IES)*, pages 256–261, 2016. doi:10.1109/ELECSYM.2016.7861013.