

THE DEVELOPMENT OF A SELF-BALANCING VEHICLE: A PLATFORM FOR EDUCATION IN MECHATRONICS

Grepl, R., Zouhar, F., Štěpánek, J., Horák, P.

Mechatronics Laboratory, www.mechlab.cz,
Faculty of Mechanical Engineering, Brno University of Technology

Abstract

This paper deals with the development of a low cost self-balancing vehicle aimed to be used as a teaching tool. Matlab/Simulink tools intensively supported the development process. According to the model-based design approach, an electro-mechanical model was created as an initial step. The mechanical part was implemented into SimMechanics, and a DC motor was considered as a static system. Next, an MF624 multifunction I/O card together with a Real-Time Toolbox were used for data acquisition. Furthermore, the Simulink Parameter Estimation blockset used the measured data for the estimation of the simulation model parameters. In the following step, several sensory processing and control strategies were tested through offline simulations. The finally adopted controller consists of a complementary filter and two independent PIDs with additional feedforward friction compensation. The C-code for the target hardware (16-bit dsPIC33 micro-controller) was automatically generated from the Simulink model via Real Time Workshop, Real Time Embedded Coder and Kerhuel blockset. The resulting platform will be used as a demonstrator in the teaching of several Mechatronics courses and further improved through the next Master and Bachelor theses.

1 Introduction

The Segway[®] [1] balancing vehicle is an ideal platform for the demonstration of most aspects of Mechatronics, including the modelling and identification of the electromechanical system and its embedded control.

Mechatronics is usually defined as a combination of mechanics, electronics and computer control. Often [2] an interaction with an environment is emphasized: the mechatronic system senses the information from the environment; next it processes it and finally reacts through actuators. Furthermore, the system must have a significantly higher level of functionality or even a completely new functionality must be enabled.

Today's teaching of technical courses must also focus on the motivation of students. One of the most useful approaches is the implementation of hands-on laboratory exercises as part of the study. The inverted pendulum (the most widely used educational model in control engineering [3]), the helicopter, magnetic levitation and drive system control are examples of proven objects for the teaching of modelling, identification and control [4]. Also, the control of a Segway[®]-like unstable balancing vehicle is very attractive and thus could be successfully used in the teaching of Mechatronics.

This paper briefly presents the development of a balancing vehicle including mechanical parts, electronics and embedded control. The aim of the work was to build an open (both hardware and software) experimental platform for the study of algorithms and approaches (including fault diagnostics and other advanced model-based techniques) for vehicle control.

2 First experiments

The development of the balancing vehicle started with the first testing platform made of components already available in our laboratory. The wheels were driven by two 350W DC motors by means of a chain transmission, as it is very common in many DIY constructions.

The very first experiments were aimed at the control of the balancing plate only (Fig. 1). H-bridges and sensors were connected to the MF624 I/O card [5] and all experiments were performed directly in the Simulink with the Real-Time Toolbox (RTT). The very important advantage of this solution is no need for compilation and thus very comfortable and fast development is allowed (see [4] for more information).

In the next steps, many signal processing and control algorithms were tested on the “wooden” platform leading to the fully functional vehicle capable of balancing with a human, yet still connected via a bunch of wires to a PC (Fig. 2).

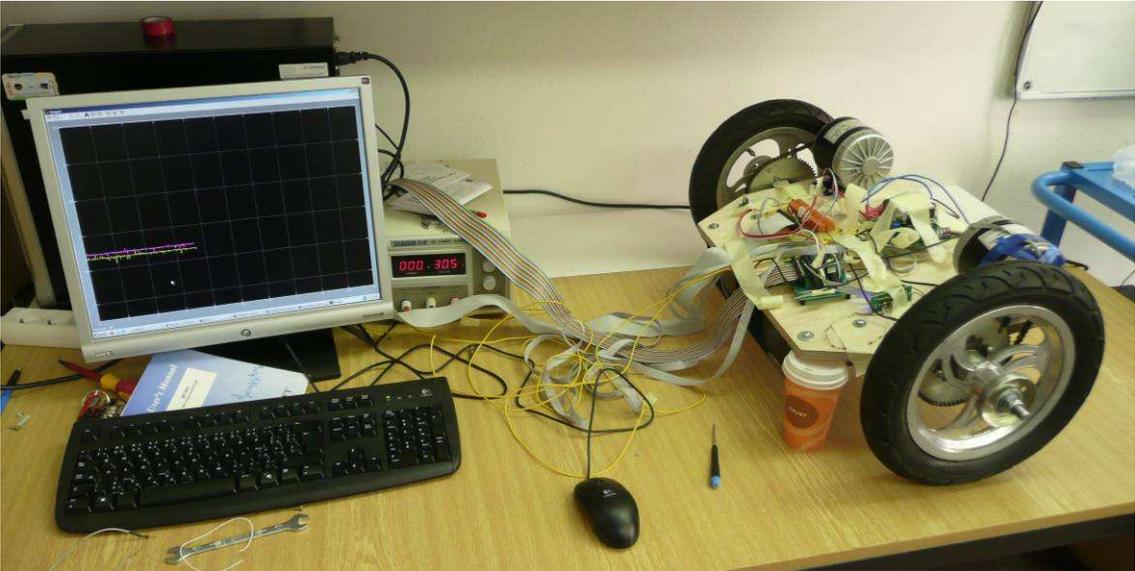


Figure 1: Testing platform at very early stage connected to Simulink via MF624 and supported by an empty cup of coffee



Figure 2: Self-balancing H1 controlled using MF624 and Real-Time Toolbox directly from Simulink (left); Detail of DC motor, H-bridge and gyro (right)

The model of the dynamics of the mechanical subsystem was implemented in SimMechanics. The rolling contact (wheel – road) cannot be directly modelled in SimMechanics and therefore it was replaced by the combination of Prismatic and Revolute joints together with the appropriate compensation of the driving torque. The resulting model is shown in Fig. 4.

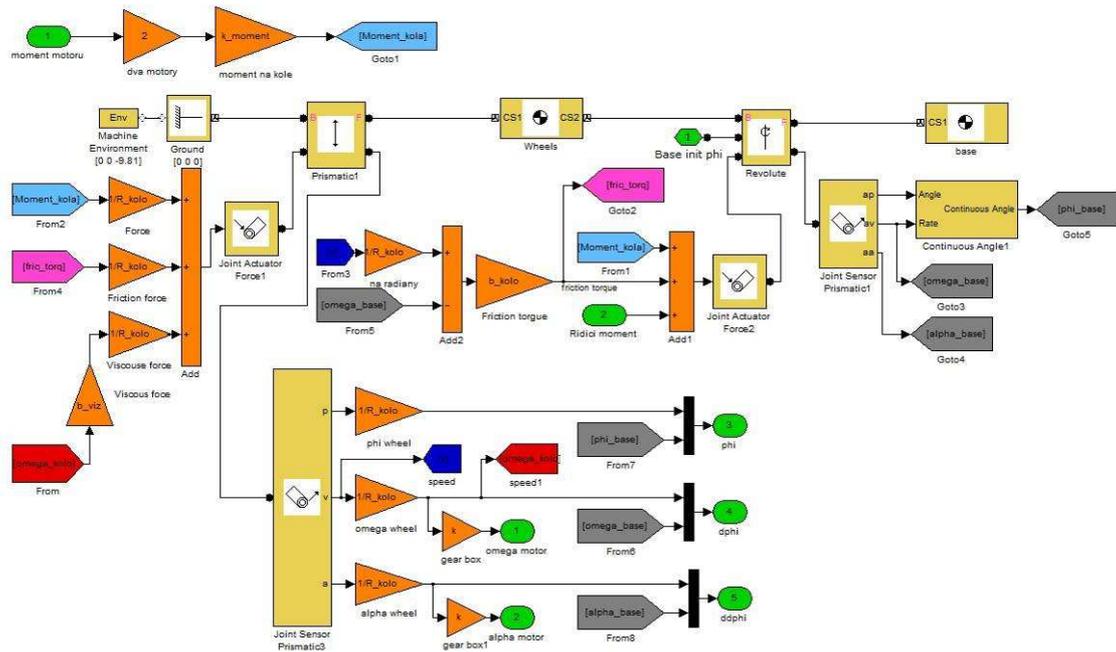


Figure 4: Model of mechanical subsystem implemented in Simulink/SimMechanics

The DC motor is usually modelled using the first order ODE. However, the dynamics of the electrical subsystem can be neglected and thus the static equation was used.

3.2 Estimation of parameters

The key problem in modelling is the determination of the parameters of the system. Some of them are relatively easy to obtain (weight, dimensions) but, for example, the viscous or dry friction cannot be measured directly.

There are two main approaches with related software tools available in the MATLAB/Simulink environment. Both of them require measured data of a particular real system but differ as follows:

- **System Identification Toolbox** uses the measured data to create (usually linear) models (e.g. ARX, OE) without any knowledge of the physical substance of the system.
- **Simulink Parameter Estimation** uses the measured data to find the parameters of the given Simulink model constructed using physical laws or other knowledge of the system.

Clearly the second option was used in our case. All the measured data was acquired via the MF624 I/O card. The estimation process was done in steps starting with the parameters of the DC motor, proceeding to the parameters of the whole vehicle.

4 Control algorithm and hardware

The final construction of the vehicle is equipped with the following sensors:

- Gyro for balancing (ADXRS150EB)
- 3D accelerometer for balancing (MMA 7260QT)
- Gyro for turning
- 2 encoders on motors (20 pulses per wheel revolution)
- Potentiometer for turning (doubled)
- 2 current sensors on motors

- 1 emergency stop button
- Battery voltage measurement.

The present control algorithm does not use encoders (turning is measured using the gyro) since their resolution is very raw. However, the additional information can be used, for example, for diagnostics or fault detection algorithms in the future.

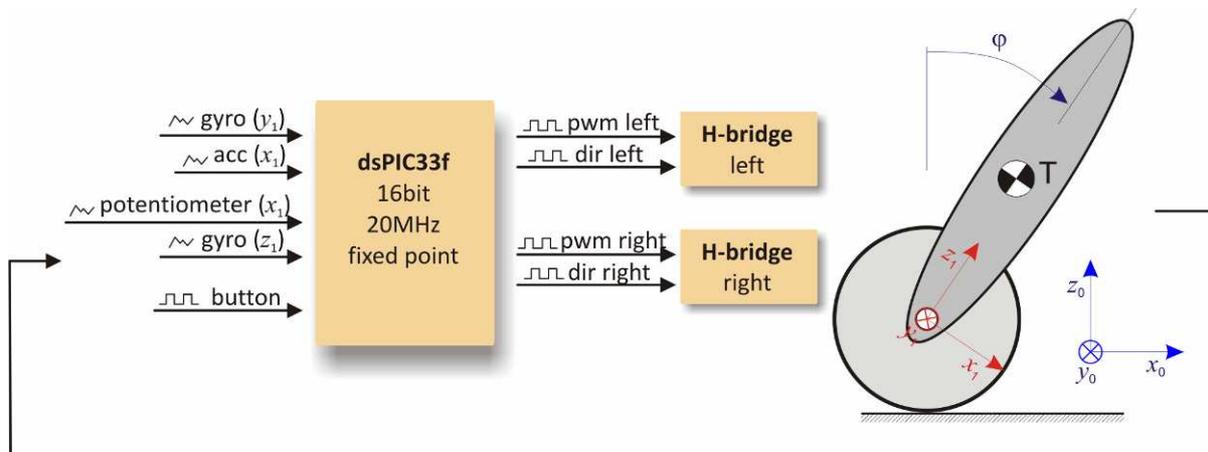


Figure 5: Schematic of inputs and outputs to/from the dsPIC micro-controller

4.1 Complementary filter

The most important and most interesting task of the control is the determination of the real inclination angle of the platform. In comparison with the classical inverted pendulum, the angle is not directly measurable. It must be obtained indirectly using one of these three options:

- **Accelerometer** – the inclination is computed as a projection of the vector of gravity into the horizontal axis of the sensor. But also the forward acceleration is projected into the measured signal and thus the angle can be computed very incorrectly.
- **Gyro** – the angle is obtained as the integration of the measured angular velocity. The problem is the drift of the gyro.
- **Combination of accelerometer and gyro** - the right practical option. Besides the famous Kalman Filter, the very simple Complementary filter [7] can be used:

$$\varphi_k = (1 - c)(\varphi_{k-1} + \dot{\varphi}_{GYRO}T_s) + c\varphi_{ACC},$$

where $\dot{\varphi}_{GYRO}$ is the measured angular velocity, and φ_{ACC} is the (static) angle computed from the accelerometer signal. The tuning of the filter lies in the selection of the constant c representing the weight of the accelerometer input. The present control algorithm runs with $T_s = 10\text{ms}$ and uses $c = 0,02$. Thus the gyro is a significantly major input to the filter and the small influence of the accelerometer is just used for drift compensation.

4.2 Controller and automatically generated C-code

The resulting controller structure is very simple and consists of PID for balancing, P regulator for turning and the complementary filter. Additionally, the constant feedforward friction compensation is used.

While the Real-Time Toolbox and the MF624 I/O card were used in the rapid control prototyping phase (Fig. 2, 3), the Microchip 16-bit dsPIC33 microcontroller is used to control the final autonomous vehicle. We developed our own modular system with the universal processor module and ad-hoc developed connector board (Fig. 6).

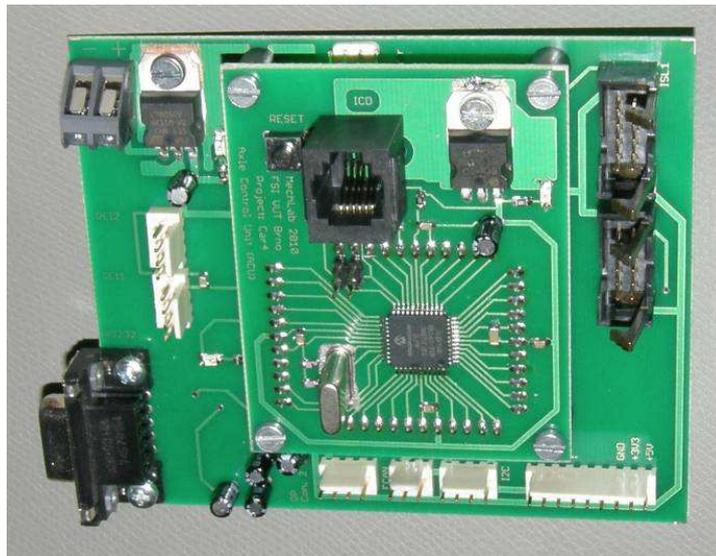


Figure 6: Modular board with dsPIC microcontroller

The C-code for dsPIC33 was generated automatically from the Simulink. The following software tools are employed:

- Real-Time Workshop (renamed to Simulink Coder)
- Real-Time Workshop Embedded Coder (renamed to Embedded Coder)
- Kerhuel blockset [8] for support of the subset of the Microchip devices
- MPLAB compiler.

The simulation model used in the rapid control prototyping phase can be used only with minor adjustment (related to fixed step solver) and new I/O blocks (Fig. 7).

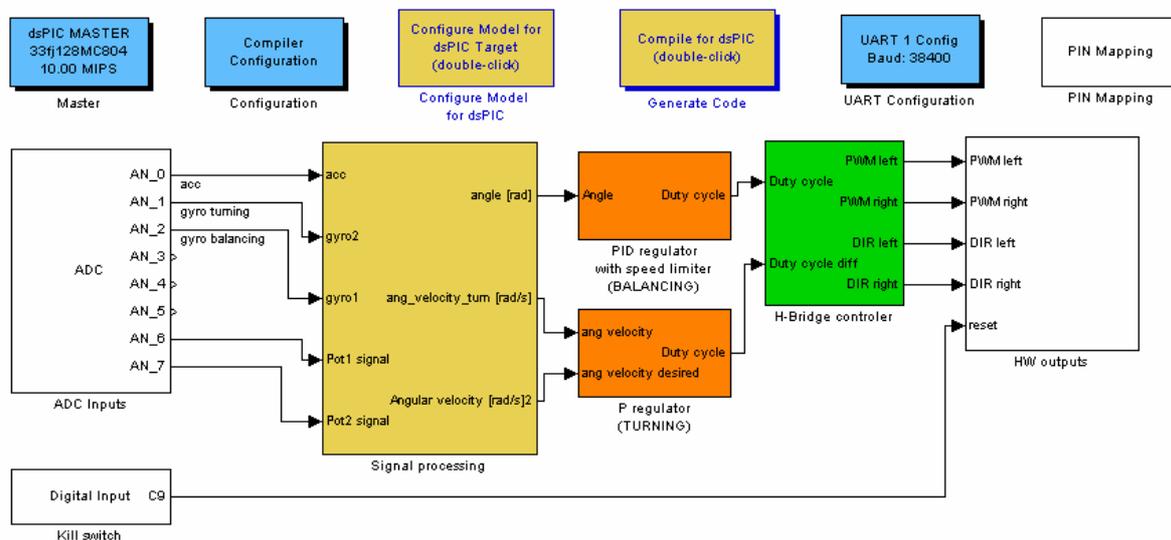


Figure 7: Simulation model used for the C-code generation for dsPIC using Kerhuel blockset

5 Conclusion

Three students spent 1,5 years working on the vehicle during they Master's thesis work [9,10,11]. The final variant has robust and matured mechanical construction (Fig. 8), can be operated for approx. 30min. and has an integrated battery charger. The stabilization algorithm works perfectly but a subjective impression of the drive is that it's a little bit less comfortable compared to a Segway®.

Modern software tools based on MATLAB/Simulink were used during the development process for the modelling, parameter estimation and control tuning. The Real-Time Toolbox and MF624 I/O card were used in the majority of experiments (without code generation). Furthermore, the dsPIC33 control hardware was programmed using the automatically generated C-code.

Many control and signal processing algorithms were tested. However, the final implemented algorithm is very simple, consisting of PID, P regulator and Complementary filter.

The resulting vehicle will be used as a demonstrator in several courses of mechatronics taught at Brno University of Technology.

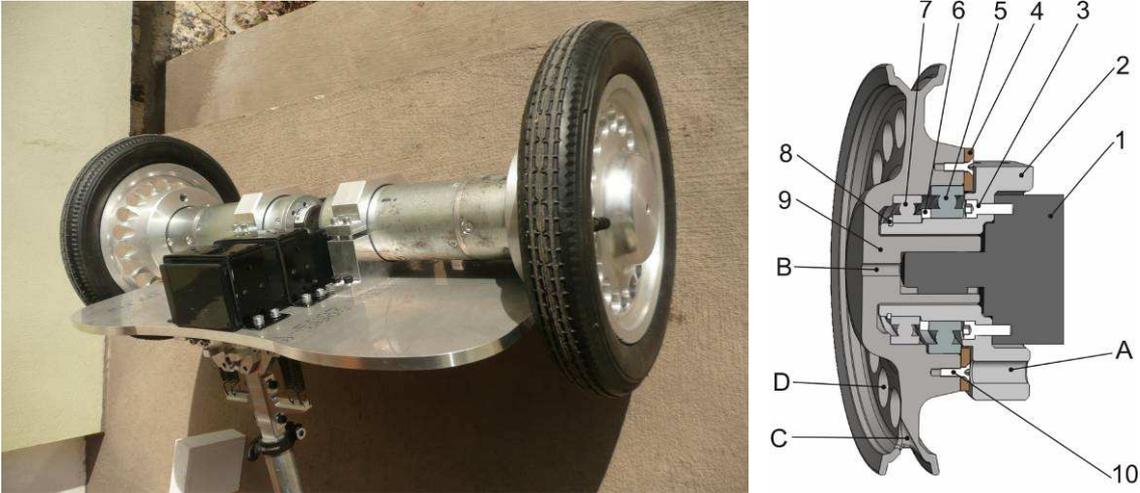


Figure 8: Detailed view and cross-section of DC motors and wheel mounting (without electronics)



Figure 9: Balancing vehicle with one of its author

Acknowledgement

The work presented in this paper has been supported by research project FSI-S-11-15 "Design, testing and implementation of control algorithms with use of non-linear models of mechatronics systems".

References

- [1] <http://www.segway.com/>
- [2] http://www.mcgs.ch/mechatronics_definition.html
- [3] Lin, S., Tsai, Ch.: *Development of a Self-Balancing Human Transportation Vehicle for the Teaching of Feedback Control*, IEEE TRANSACTIONS ON EDUCATION, VOL. 52, NO. 1, FEBRUARY 2009
- [4] Grepl, R.; , "Real-Time Control Prototyping in MATLAB/Simulink: Review of tools for research and education in mechatronics," *Mechatronics (ICM), 2011 IEEE International Conference on* , vol., no., pp.881-886, 13-15 April 2011
- [5] <http://www.humusoft.com/produkty/datacq/mf624/>
- [6] Grasser, F.; D'Arrigo, A.; Colombi, S.; Rufer, A.C.; , "JOE: a mobile, inverted pendulum," *Industrial Electronics, IEEE Transactions on* , vol.49, no.1, pp.107-114, Feb 2002
- [7] Colton, S.: The balance filter – A simple solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing problem, report, web.mit.edu/scolton/www/filter.pdf
- [8] http://www.kerhuel.eu/wiki/Simulink_-_Embedded_Target_for_PIC
- [9] Zouhar, F.: Návrh konstrukce, řízení a elektroniky pro nestabilní balancující vozidlo, DP 2011, VUT v Brně
- [10] Horák, P.: Řízení laboratorního modelu nestabilního balancujícího vozidla, DP 2011, VUT v Brně
- [11] Štěpánek, J.: Identifikace systému, sensorika a implementace řídicího algoritmu pro nestabilní balancující vozidlo, DP 2011, VUT v Brně

assoc. prof. Robert Grepl, Ph.D.

grepl@fme.vutbr.cz,

www.mechlab.cz

Mechatronics laboratory,

Institut of Institute of Solid Mechanics, Mechatronics and Biomechanics,

Faculty of Mechanical Engineering, Brno University of Technology