Tomislav Tomašić
Andrea Demetlika
Mladen Crneković

SELF-BALANCING MOBILE ROBOT TILTER

Summary

In this project a remote controlled self-balancing mobile robot was designed, built and controlled. This paper gives a summary of the work done in the fields of mechanical design, electronics, software design, system characterization and control theory. This wide array of fields necessary for the realization of the project holds the project up as a leading example in the field of mechatronics. In the paper special focus will be on the modelling of the robotic system and the simulation results of various control methods required for the stabilization of the system.

Key words: *self-balancing robot, control theory, inverted pendulum, PID, LQR*

1. Introduction

The research on a two-wheel inverted pendulum, which is commonly known as the self-balancing robot, has gained momentum over the last decade in a number of robotic laboratories around the world. This is shown by a rapid increase in the number of journal papers, research projects and theses on the subject, which have been dealing with similar projects [1-4]. The principles of operation of such a robot are also used in a commercial electric transportation vehicle called Segway [5].

The self-balancing mobile robot on two wheels, called Tilter, works on the principle of an inverted pendulum. The robot is inherently unstable and without external control it would roll around the wheels' rotation axis and eventually fall. Driving the motors in the right direction returns the robot to the upward position.

Although the robot is inherently unstable, it has several advantages over the statically stable multi-wheeled robots – since it has only two wheels (two points touching the ground) it requires less space; since it is based on dynamic stability (it constantly needs to correct its tilt angle to remain stable) it exhibits improved dynamic behaviour and mobility. This additional manoeuvrability allows easy navigation on various terrains, turning sharp corners (it can turn on the spot) and traversing small steps or curbs.

The stages in the project realization are explained in the following text. The first part deals with the mechanical structure of the robot containing the explanation about how it differs from that of classical mobile robots.

In the second part, the selected electronic components, and the design of electrical circuits and printed circuit boards are explained. The next part gives an overview of the system modelling which was built by reference to [1], the controller design and the estimator design. The last part deals with the software implemented in the robot, the computer and the cell phone.

2. Mechanical structure

The robot is comprised of three plexiglass boards connected by threaded bars and separated by cylindrical separators. The bottom board contains a motor controller board and two motors connected to the wheels by means of a planetary gearbox. The middle board holds the control electronics and the Bluetooth communication module. On the top board there are two gel type batteries which make the main portion of the robot's mass.

Fig. 1 Self-balancing mobile robot

Unlike classical mobile robots where the centre of gravity is placed as low as possible, here it should be placed as high as possible. That is because distributing the larger part of the mass to the upper part of the robot increases its moment of inertia around the wheel axis, thus slowing down its rotation. This reduces the dynamics of the upper part in relation to the bottom part, making it easier to stabilize and control the robot.

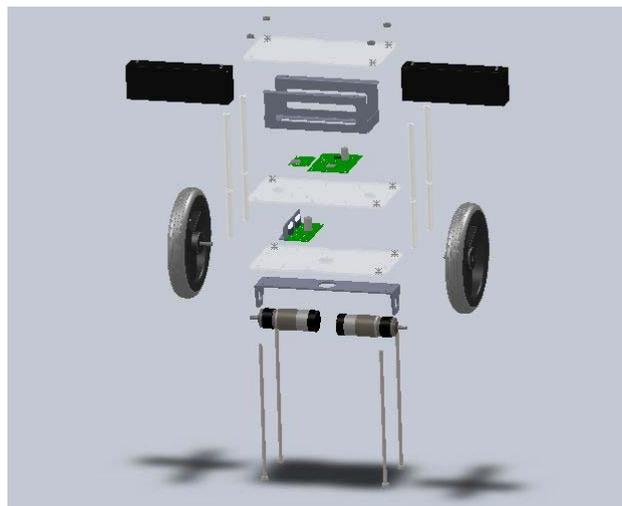Every component of the robot was first modelled in SolidWorks (Fig. 2) and then manufactured.

Fig. 2 An exploded view of the CAD model

3. Electronics

The mobile robot has three printed circuit boards, each with a different function. The first one is the main control board (Figs. 3 and 4) which controls all the other boards. It has an FT232 chip for the USB communication, a connector for sensors, connectors for the connection with other devices and most importantly, an ATmega128A microcontroller.

The microcontroller converts analogue sensor values to digital ones, and then calculates and generates an accurate PWM signal to drive the motors. Using external interrupts it counts impulses from the encoders, from which the relative position and velocity can be calculated. Using a serial connection the microcontroller communicates with the Bluetooth module.

Information about the robot's position and the tilt angle is provided by three sensors: incremental magnetic encoders, a two-axis accelerometer and a one-axis gyroscope. The encoders allow measurement of the robot's velocity and position. The accelerometer is used for measuring the components of the earth's gravity vector, from which the tilt angle can be calculated. The gyroscope measures the angular velocity of the robot around the wheels' rotation axis.

The main control board is designed in such a way that it is possible to add other devices like a camera, ultrasonic sensors, actuators etc.
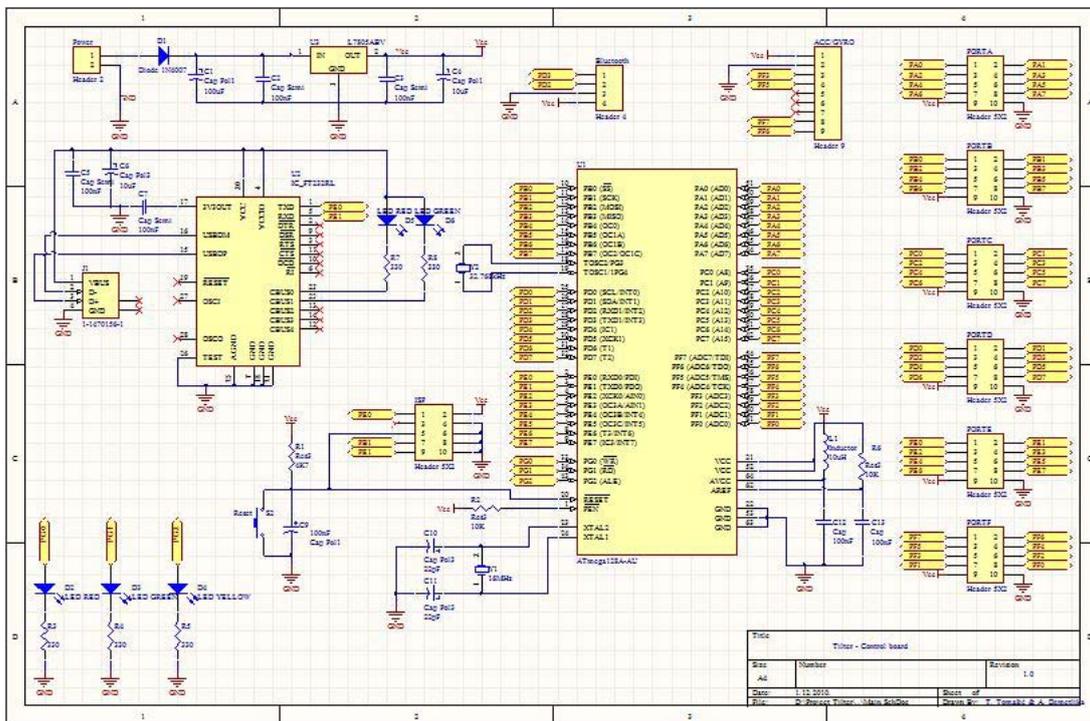

Fig. 3 Schematic of the control board

The second board is the motor controller (Fig. 5). It is a connection between the control (5 V, low current) and the energy part (12 V, high current) of the robot. It has two H-bridge LMD18200 chips which drive the motors using PWM signals sent from the microcontroller.
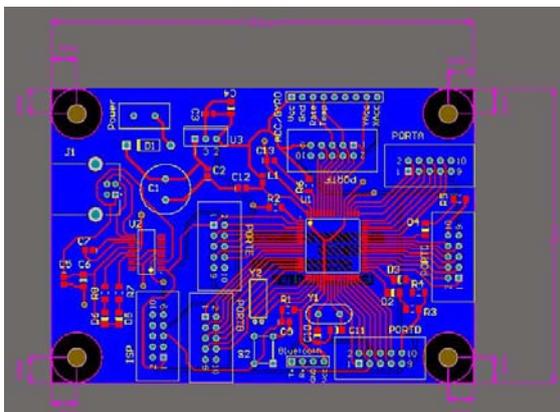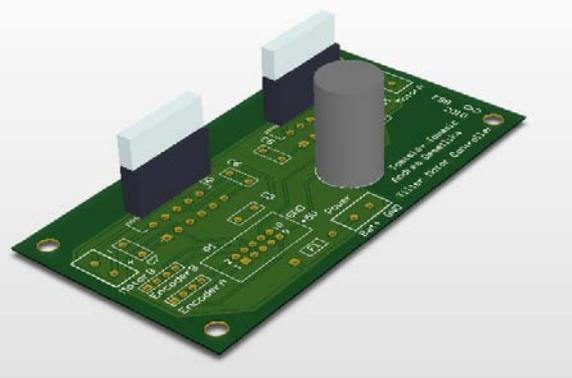

Fig. 4 Control board circuit

Fig. 5 3D representation of the motor controller board

The third board has an RN-41 Bluetooth module which communicates with the microcontroller using serial communication. The Bluetooth communication allows the robot to wirelessly communicate with devices that have Bluetooth capabilities, such as computers, cell phones, joysticks etc.

4. The modelling and the control of the robot

4.1. The modelling of the robot's dynamics

Before designing the controllers, the robot's dynamic model has to be known. As shown in [1], the main equations governing the robot's dynamics, using the symbols from Figs. 6 and 7, are:

$$\Sigma F_x = 0: \ m_k \ddot{x} = -\frac{1}{2}F_H + F_{pH} - b\dot{x}, \tag{1}$$

$$\Sigma F_y = 0: \ m_k \ddot{y} = F_{pV} - \frac{1}{2}F_V - m_k g, \tag{2}$$

$$\Sigma M_T = 0: \ J_k \ddot{\theta}_k = M_M - F_{pH} r, \tag{3}$$

which describe the dynamics of the wheel, and

$$\Sigma F_x = 0: \ m_n \ddot{x}_n = F_H, \tag{4}$$

$$\Sigma F_y = 0: \ m_n \ddot{y}_n = F_V - m_n g, \tag{5}$$

which describe the dynamics of the inverted pendulum.
After combining equations 1-5, two equations of motion are obtained:

$$(J_n + L^2 m_n)\ddot{\theta}_n = -Lm_n \ddot{x} \cos\theta + gLm_n \sin\theta - 2M_M \tag{6}$$

$$\left(m_k + \frac{J_k}{R^2} + \frac{1}{2}m_n\right)\ddot{x} = \frac{M_M}{R} - \frac{1}{2}m_n L \cos\theta \ \ddot{\theta} + \frac{1}{2}m_n L \sin\theta \ \dot{\theta}^2 - b\dot{x}. \tag{7}$$
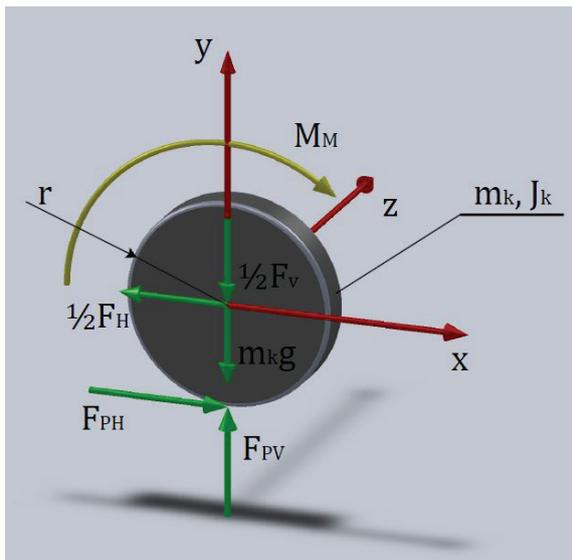

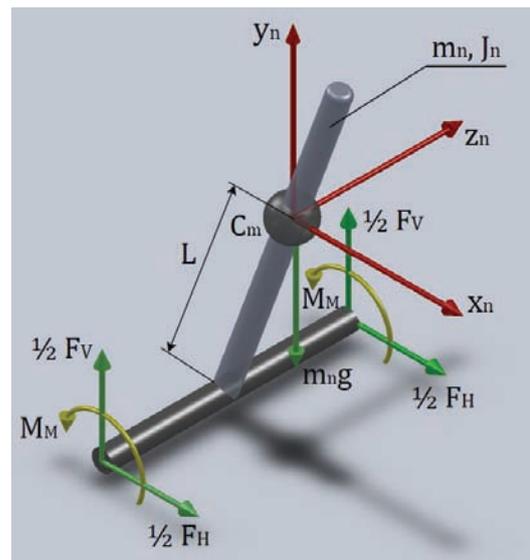Fig. 6 Forces and torques acting on the wheel

Fig. 7 Forces and torques acting on the inverted pendulum

In order to simplify the design of the controller, the nonlinear equations of motion have to be linearized. The desired state of the system is the one in which the angle is small and

close to 0 degrees. Therefore, assuming $\sin\theta = \theta$, $\cos\theta = 1$ and $\dot{\theta}^2 = 0$ we obtain the following linearized equations

$$(J_n + L^2 m_n)\ddot{\theta}_n = -Lm_n\ddot{x} + gLm_n\theta - 2M_M , \tag{8}$$

$$\left(m_k + \frac{J_k}{R^2} + \frac{1}{2}m_n\right)\ddot{x} = \frac{M_M}{R} - \frac{1}{2}m_n L\ddot{\theta} - b\dot{x} . \tag{9}$$

The linearized system equations are then represented in the state-space form

$$\dot{x} = \mathbf{A}x + \mathbf{B}u , \tag{10}$$

$$y = \mathbf{C}x + \mathbf{D}u , \tag{11}$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & a_{43} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix} u , \tag{12}$$

where the elements of matrices **A** and **B** are:

$$a_{22} = \frac{-10b}{14m_k + 5m_n} , \tag{13}$$

$$a_{23} = \frac{-5m_n}{14m_k + 5m_n} , \tag{14}$$

$$a_{42} = \frac{5b}{14Lm_k + 5Lm_n} , \tag{15}$$

$$a_{43} = \frac{7gm_k + 5gm_n}{14Lm_k + 5Lm_n} , \tag{16}$$

$$b_2 = -\frac{10L + 5r}{14m_k r + 5Lm_n r} , \tag{17}$$

$$b_4 = -\frac{5Lm_n + 7m_k r + 5m_n r}{14L^2 m_k m_n r + 5L^2 m_n^2 r} . \tag{18}$$

Fig. 8 Simulink model of the robot dynamics

These equations are then used to make a Simulink model of the robot (Fig. 8) which allows the simulation of its dynamics. In order to visualize the dynamics of the robot, the Simulink model is connected to a VRML model.

4.2. The Kalman filter

To apply the appropriate voltage to the motors, it is necessary to know the accurate value of the robot's tilt angle and position. Due to vibrations, readings from the accelerometer are noisy and have a delay, while the problem with the gyroscope is its drift and offset.

The Kalman filter was designed in order to solve these problems and combine the sensor readings. It predicts future states of the system using the robot's dynamic model, with the equations given in the references [6, 7].

4.3. Control algorithms

Several control algorithms were considered and tested for stabilizing the robot. The first one considered was the PID controller which controls the robot using only one feedback variable – the tilt angle. Simulations and testing showed that even though the robot is stable in regard to the tilt rotation, small disturbances eventually cause big changes in the position.
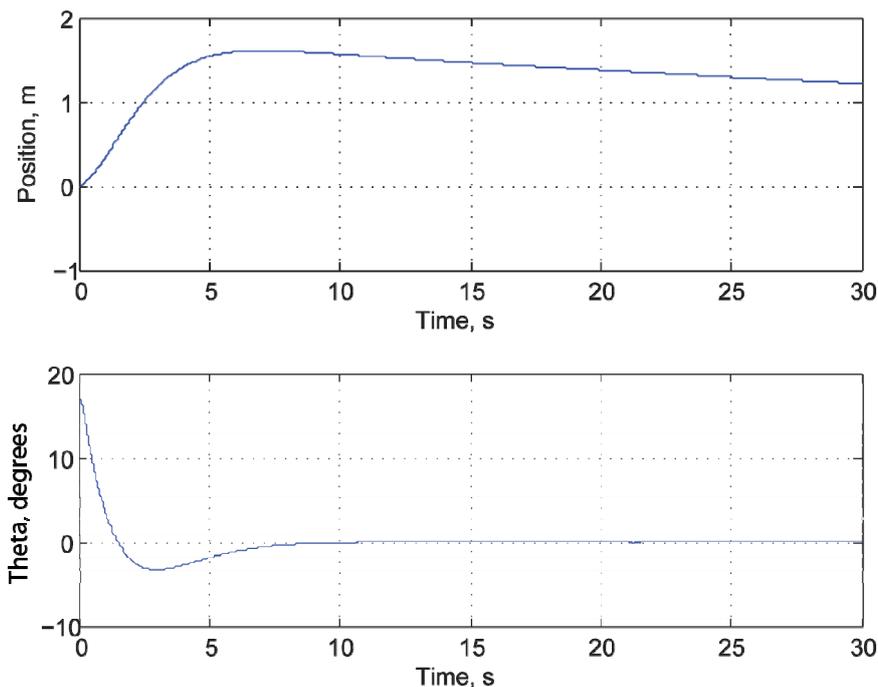

Fig. 9 Response of the system controlled by PID controller for angle initial state of 17 degrees

In order to overcome these problems the LQR controller was designed. Unlike the PID, the LQR controller uses all state variables (tilt angle, angular velocity, position and linear velocity) to calculate the control value. This allows the robot to hold the position and the tilt angle at desired values.
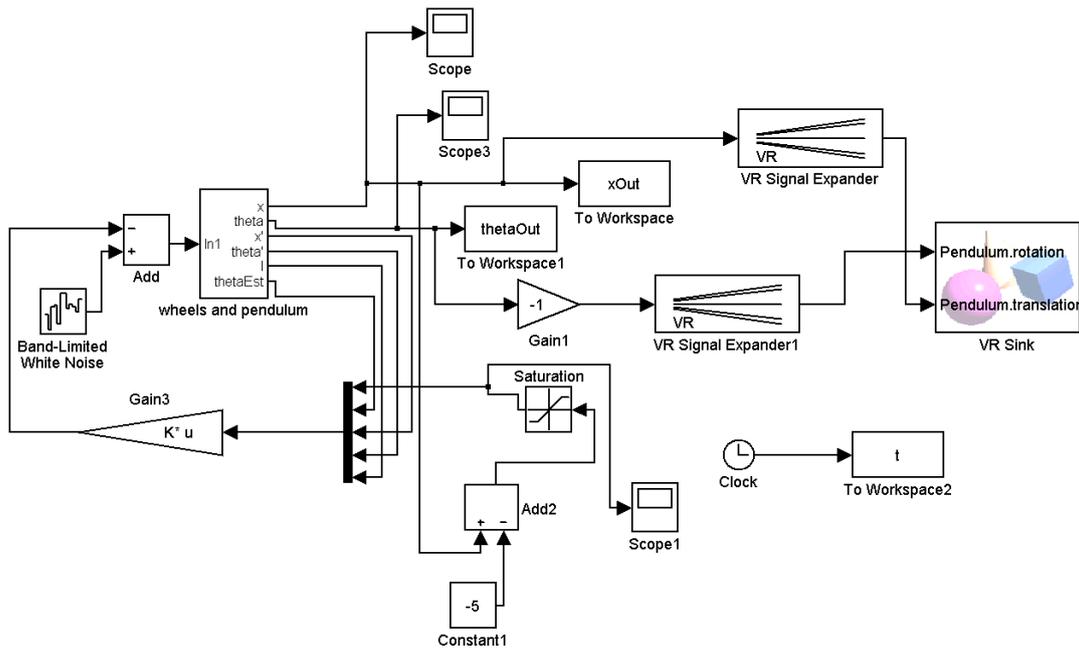
Fig. 10 LQR control algorithm blocks

The design of the LQR controller started with the transfer of the linearized model of the system into the discrete domain. Next, the **Q** and **R** matrix were set to an initial value, and latter adjusted. In the end, by using the Matlab function *dlqr*, the state feedback gain **K** was obtained. The results of the LQR controller design are shown in Fig. 11.
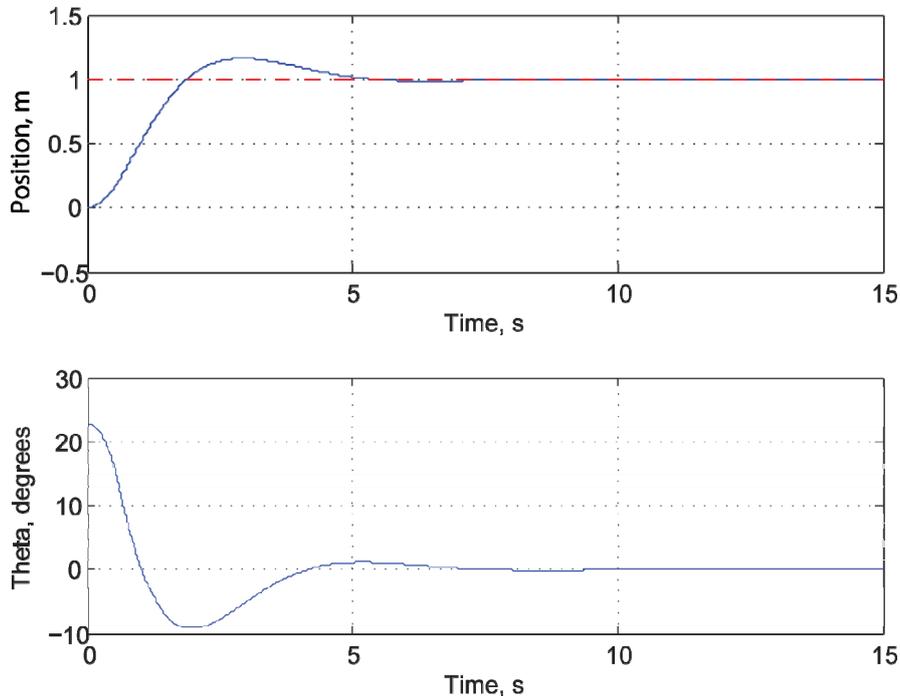

Fig. 11 Response of the system controlled by LQR controller for position reference of 1m and angle initial state of 17 degrees

The advantages of the use of the LQR controller are best observed while stabilizing the robot on a slope. The PID controller stabilizes the tilt angle, but the robot soon starts to drive down the slope and increases its speed until the control value eventually saturates and the

robot falls. The LQR controller, on the other hand, maintains the desired position with an offset and the tilt angle even when the reference value is present (Fig. 12).
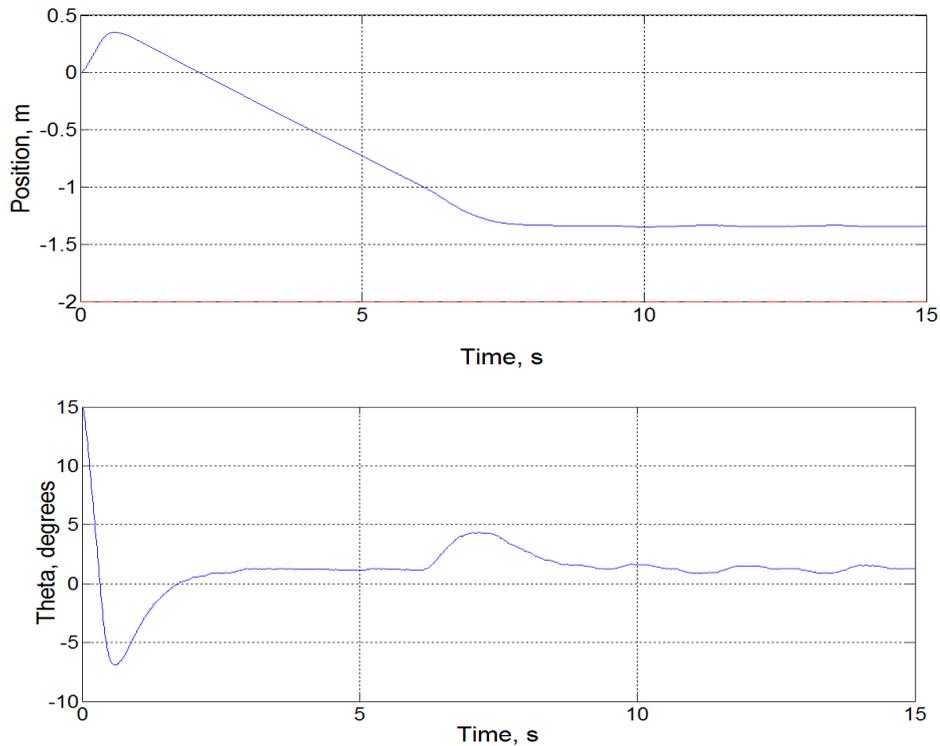

Fig. 12 Response of the system controlled by LQR controller for position reference of -2m and angle initial state of 15 degrees on a slope of 15 degrees

The fuzzy controller was successfully designed and simulated (Fig. 13), but due to the increased complexity of its application in the microcontroller environment (in relation to other controllers) it was not implemented and tested on a real platform.
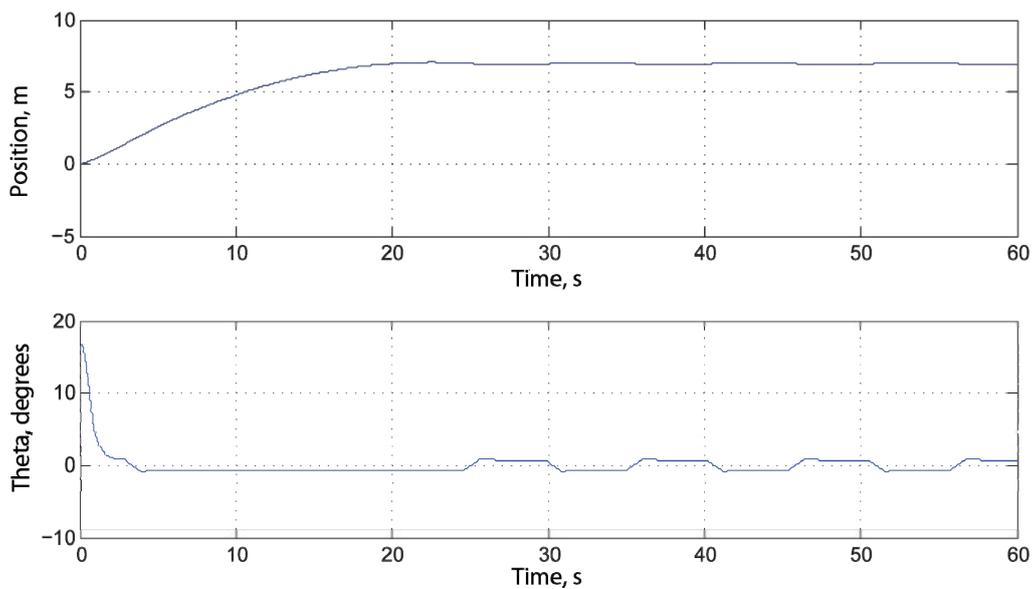

Fig. 13 Response of the system controlled by fuzzy controller for position reference of 7m and angle initial state of 17 degrees

5. Software

The robot control software is written in AVRStudio – a tool for programming Atmel's microcontrollers. It contains the code for stabilizing and controlling the robot, communicating with other devices (cell phone or laptop), performing analogue-to-digital conversion and filtering the sensor readings.

Using the Bluetooth communication the robot can be controlled by a computer or a cell phone. The application on the computer (Fig. 14) is made in Microsoft's C# development environment. Besides controlling the robot by using a graphical interface, this application displays telemetry from the robot in real time. Values that can be monitored are position, velocity, tilt angle, angular velocity, and ambient temperature.
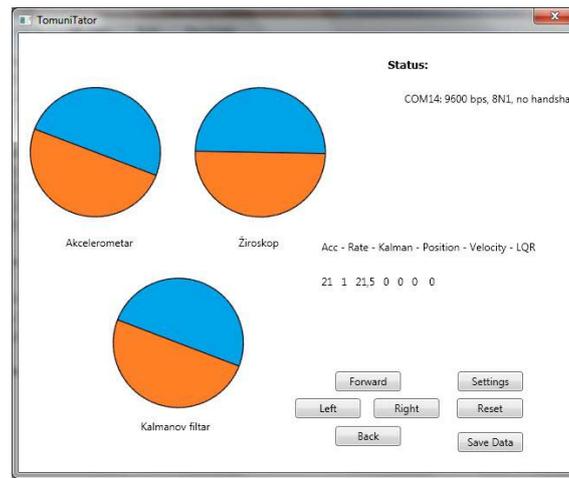

Fig. 14 Graphical interface on the computer

The android application on the cell phone (Fig. 15) uses built-in accelerometers to drive the robot in the direction in which the cell phone is tilted. To prevent any accidental tilting of the cell phone and thus unwanted control of the robot, a safety mechanism has been created – to drive the robot, the user has to press and hold two buttons at the same time.

Fig. 15 Android application used for driving the robot

6. Conclusion

As performance limits in mobile robotics are increasing, dynamic effects (and therefore dynamic stability) are becoming ever more important.

This project has resulted in building a working self-balancing mobile robot on two wheels (Fig. 16). The robot is successfully balanced and driven using a computer or a cell phone. Various control techniques have been designed and tested. It has been shown that the LQR control produces the best results, for the reasons shown in Chapter 4.3. That is why the LQR control has been implemented in the final version of the robot.

Fig. 16 Robot driving and stabilizing on a rough surface

The paper has presented the process in which the project has been carried out from the design and production of specific parts to the integration of electronic, mechanical and software sections.

Because of the need to use the knowledge in the fields of mechanics, electronics, programming and control, this project is extremely interdisciplinary and as such one of the most representative mechatronic problems.

Further work will include increasing the level of autonomy of the robot by adding a vision system, thus allowing the robot to avoid obstacles. Also, by improving the components of the robot we hope to achieve higher speeds.

REFERENCES

[1] H. Bin, L.W. Zhen, L.H. Feng: *The Kinematics Model of a Two-Wheeled Self-Balancing Autonomous Mobile Robot and Its Simulation.* ICCEA '10 Proceedings of the 2010 Second International Conference on Computer Engineering and Applications - Volume 02.

[2] N. G. M. Thao, D. H. Nghia, N. H. Phuc: *A PID Backstepping Controller For Two-Wheeled Self-Balancing Robot,* IFOST 2010 Proceedings

[3] N. M. Abdul Ghani, N. I. Mat Yatim, N. A. Azmi: *Comparative Assessment for Two Wheels Inverted Pendulum Mobile Robot Using Robust Control*, International Conference on Control, Automation and System 2010, KINTEX, Gyeonggi-do, Korea, Oct 27-30, 2010

[4] Hua Sun, Haixu Zhou, Xiang Li, Yanhui Wei and Xiao Li: *Design of Two-Wheel Self-Balanced Electric Vehicle based on MEMS*, Proceedings of the 2009 4th IEEE International Conference on Nano/Micro Engineered and Molecular Systems, Shenzhen, China, January 5-8, 2009

[5] http://www.segway.com

[6] G. Welch, G. Bishop: *An Introduction to the Kalman Filter.* 2006.

[7] J.L.C. Miranda: *Application of Kalman filtering and PID control for direct inverted pendulum control*.2009.

Tomislav Tomašić
xtomasicx@gmail.com
Andrea Demetlika
andrea.demetlika@gmail.com
Prof. dr. sc. Mladen Crneković
mladen.crnekovic@fsb.hr
University of Zagreb
Faculty of Mechanical Engineering and
Naval Architecture
Ivana Lučića 5, 10000 Zagreb, Croatia