Embedded controller for balancing type robot

Jakub Tutro*, Krzysztof Wesołowski, Mariusz Pauluk*, Dariusz Marchewka***

*AGH University of Science and Technology
**Technika Obliczeniowa, Kraków

Abstract: The article is a summary of the Segway two-wheel balancing robot project. The project included design of the mechanical platform and implementation of control algorithm on the controller. The robot contains digital gyroscopes and DC motors with encoders. The controller based on measurements from the sensors stabilizes the robot in an upright position and allows it to move. The mechanical part was designed using CAD (SolidWorks), and then imported into Simulink environment. The design techniques based on models (Model Based Design) were used.

Keywords: balancing robot, model based design, Segway, control design

1. Introduction

A development of science and related to this progress of informatics and technology favours creating more and more advanced systems with highly sophisticated control algorithms. A level of advancement of the new technologies causes problems with assuring reliability of work, both on the hardware as well on the software layer. Currently, a lot of efforts are laid on improving sureness of the achieved final solutions. A traditional concept of developing systems, often called a V-diagram has many drawbacks. The paper presents the process of constructing a controller for the balancing type robots basing on the so called Model Based Design technique (MBD) [5]. The MBD name comes from referring to the mathematical model of the constructing system through the whole design process, what gives better effectiveness in detecting potential errors and elasticity in correcting them. The basic advantages (features) of the MBD:

- presence of mathematical model of the designed system,
- constructing mathematical model at a stage of working over the concept,
- simulation of the system behavior,
- simulation during the phase of constructing a controller,
- numerical controller parameters optimization with a plant model,
- controller verification and cover tests,
- access to the following techniques: real time simulation involving rapid prototyping, hardware in the loop,
- automatic code generation,

The paper presents a balancing type robot (Segway style [1, 9]) that is being developed at the AGH University of Science and Technology. An aim of the project is to design and construct a mascot of the Students Science Association. Students and scientific workers from different departments of the AGH take part in this project.

Below there are presented early-stage results of designing an embedded controller that should stabilize and control movement of the robot. During the whole development process a Model Based Design approach was being applied. There are also presented models, that were created during development and their application for controller implementation purposes.

2. Robot construction details

Currently the robot is in a form of a platform ready to build upon. This paper describes work with initial mechanical platform. The second platform was constructed meantime and is illustrated in fig. 1. Two DC motors with planetary gearboxes are used to drive platform and stabilize robot in upright position. Motors are controlled with H-bridges,

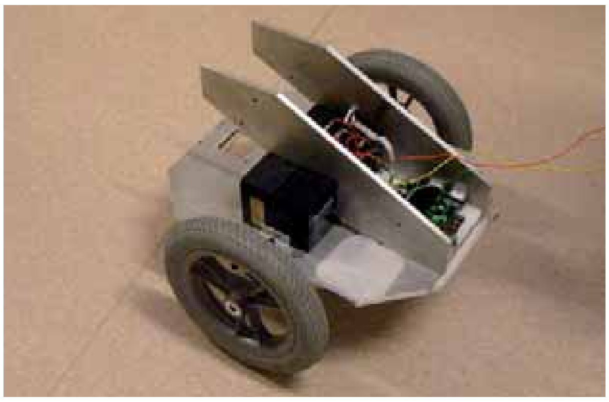

Fig. 1. Latest photo of the robot, with the new mechanical platform
Rys. 1. Najnowsze zdjęcie robota z nową platformą mechaniczną

fed by microcontroller generated PWM signals [4]. The motors are equipped with encoders, so the controller can measure a position of the wheels and respectively their speed. A Micro Electro-Mechanical Systems type (MEMS) [2] gyroscope is used for measuring angular coordinates of the robot body. Data from sensors is processed by the STM32 microcontroller [3]. The DC motors are controlled accordingly to the processing results.

3. Modelling

A traditional modelling phase relies on: analysing forces acting in the physical model and then, basing on the Newton's law, constructing differential equations describing dynamic of the system. An alternative and equivalent method basis on the Lagrange equations [7].

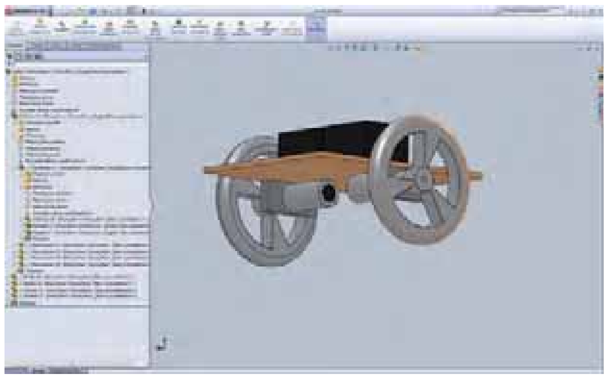

Fig. 2. SolidWorks model of the robot
Rys. 2. Model robota w środowisku SolidWorks

The mentioned above process of obtaining mathematical equations usually takes a lot of time. It may be shortened and simplified by using a software for modelling dynamic of the physical objects. In the presented case, the SimMechanics software helped to omit this process, by enabling creation of the model in software dedicated for mechanical modelling purposes language that is supported by appropriate libraries and features a graphical interface.

At the beginning of the project, no physical model of the robot was available. The virtual model was prepared in SolidWorks [10]. The masses were assumed according to material characteristics or documentation, and a moment of inertia tensor was calculated by SolidWorks automatically. Then, the robot model was exported to Simulink environment. Masses, inertia tensors and shapes were preserved during the export. Fig. 2 presents the robot model built in the SolidWorks software and fig. 3 illustrates the model imported into the Simulink environment.

SolidWorks supports modelling much more sophisticated objects, that may be exported into SimMechanics [8]. However, after exporting there was a need to model a movement of the robot in reference to the floor. There is no straightforward support in SimMechanics for defining such interaction.

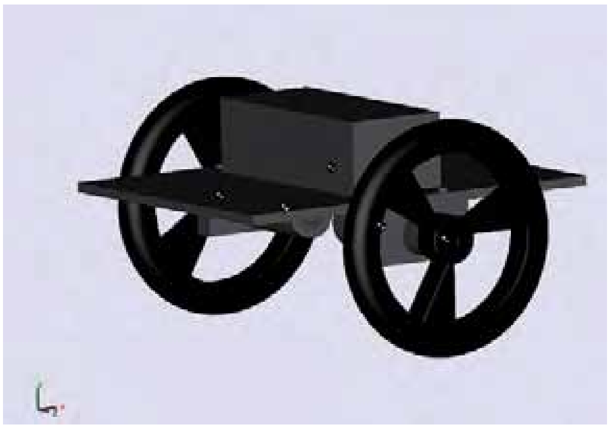

Fig. 3. Robot imported into the Simulink environment
Rys. 3. Robot po zaimportowaniu do środowiska Simulink

The problem is modelled with two joints with two degrees of freedom: prismatic along an X axis and revolute around a Z axis.

An extra velocity constraint forces them to spin with a speed proportional to a linear velocity.

In the described case, robot behaviour may be simulated only in two dimensions: X, Y axes and a rotation around a Z axis).

4. Modelling DC motor with planetary gearbox

A mechanical model of DC motors and wheels consists of several parts. Both a motor shaft and an output shaft are attached to the platform with revolute joints.

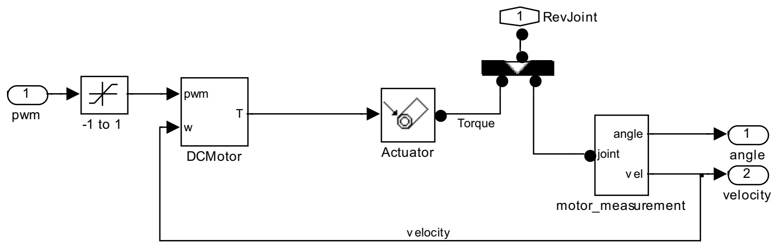

Fig. 4. A connection between the DC motor electric model and SimMechanics
Rys. 4. Połączenie modelu elektrycznego silnika DC z modelem mechanicznym w SimMechanics

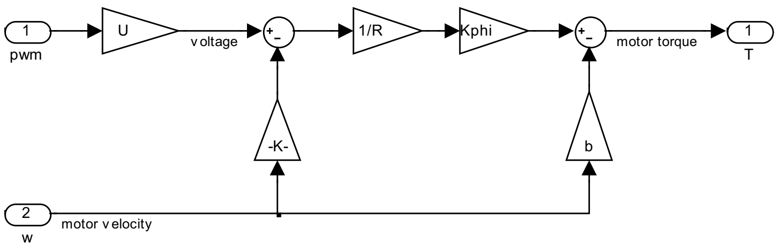

Fig. 5. The DC motor electric model
Rys. 5. Model elektryczny silnika DC

A joint between the body and the motor shaft is controlled by using an external torque. A planetary gearbox is modelled with a gear constraint between the motor shaft and the output shaft. Details of a connection between the motor and the mechanics are illustrated in fig. 4.

It was assumed, that the electrical time constant of the DC motor is small enough to omit its influence. The model has two inputs: a PWM control value, that is rescaled respectively to a voltage value and a value of the shaft velocity. The DC motor torque is calculated according to the model presented in fig. 5.

Speed and torque constants were assumed to be equal to the values in a data sheet. A motor viscous friction is also available in a documentation but it does not take the planetary gearbox into account. The viscous friction of the motor with the planetary gearbox was identified basing on encoder measurements gathered during experiments.

5. Modelling sensor errors and characteristics

Presented models do not calculate all useful states. It is often possible to assume that sensors are ideal. However, in case of encoders it might be not always reasonable, e.g. for movements with a small velocity. Modelling its behavior can improve efficiency of a control algorithm and accuracy of simulations based on the mathematical model.

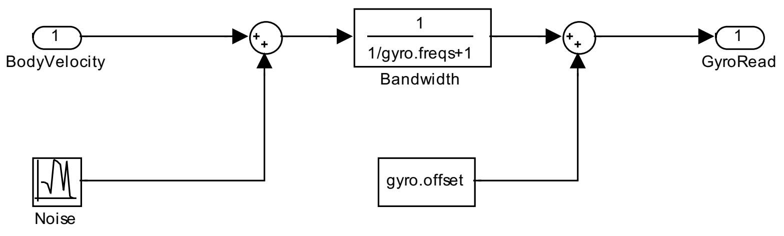

Fig. 6. Model of the MEMS gyroscope
Rys. 6. Model żyroskopu wykonanego w technologii MEMS

Application of the MEMS gyroscopes except benefits, relates also to problems with a measurement. The gyro sensors feature: limited bandwidth, significant measurement offset and also noisy output signal. Fig. 6 presents the model of the gyro regarding the mentioned problems. A gyroscope inertia was ignored during a model linearization process and is thus also ignored in the controller structure. However controller parameters were tuned with inertia enabled. Parameters for the gyro model illustrated in fig. 6 were taken from a manufacturer datasheet, as they require complex experiments to identify them properly.

Output signal of the gyroscope is scaled to 12-bit integer variable, to reflect an Analog to Digital Converter (ADC) resolution. The encoder attached to the motor shaft generates 512 cycles per revolution. Errors of quantization were also concerned and modeled but tests showed, it did not affect a model behavior.

6. Modelling digital controller

The central hardware of the embedded controller is the STM32 microcontroller. Due to implementation requirements, all calculations are processed with a single precision floating point arithmetic.

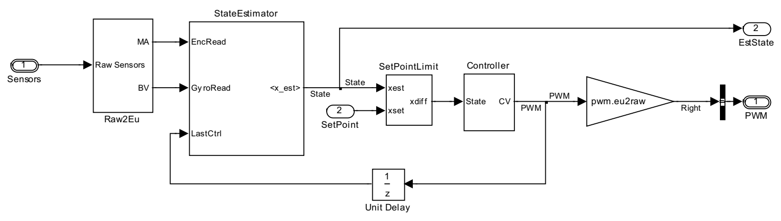

Fig. 7. Internal structure of the controller model
Rys. 7. Wewnętrzna struktura modelu sterownika

Fig. 7 illustrates structure of the controller model. It is assumed, data from sensors is in a raw format, same as in registers of microcontroller. For simulations purposes the controller is fed by data from sensor models, earlier described.

The *Raw2Eu* block converts data to floating point format and scales to SI units. The converted sensor readings are connected as input to the *StateEstimator* block.

The *StateEstimator* block allows choose one of two etimators: simple derivative/integral and Kalman filter [11]. However, the tests showed the first one gives better results and requires less efforts on parameters optimization to estimate the model state correctly.

Difference between a set point and a current state is calculated and also limited in *SetPointLimit* block. A control error feeds *Controller* block. An LQ (*Linear Quadratic*) regulator calculates control value. The value is converted into raw units and write down into PWM (*Pulse Width Modulation*) registers.

7. Example simulation result

Stabilizing robot in an upright position is rather an easy task. The system is linearized in the equilibrium point and if a distance of the system is close enough to the upper point, the system may be treated as a linear one. Segway type robots usually have to handle two typical control problems. The first task relies on disturbance rejection.

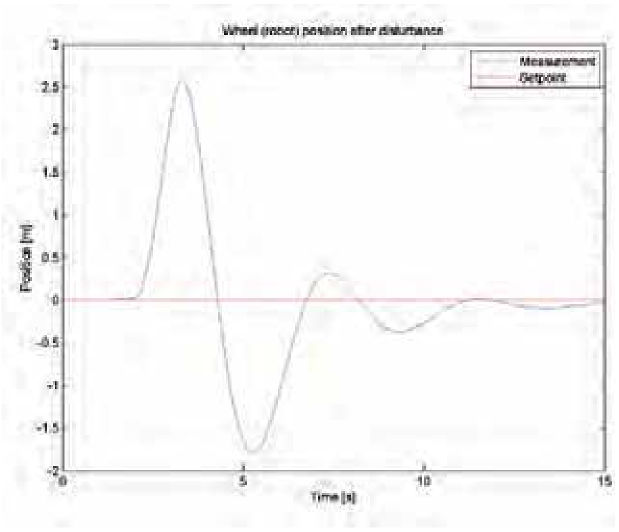

Fig. 8. Changes of the robot position in response to appliance of the external force

Rys. 8. Zmiany położenia robota w odpowiedzi na zewnętrzną siłę

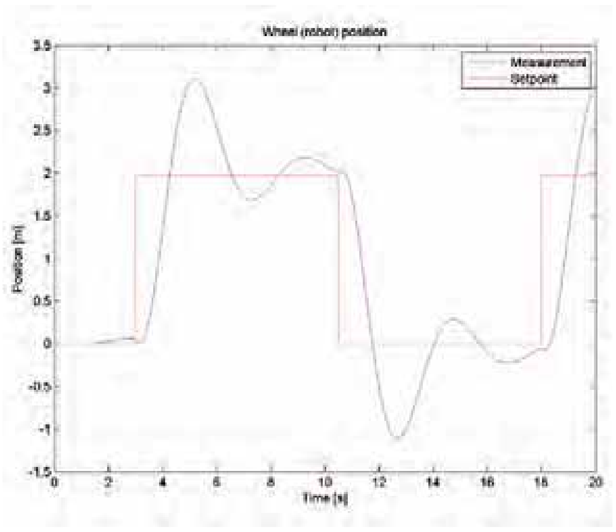

Fig. 10. Robot relocation without difference limit

Rys. 10. Przemieszczanie robota – sterownik bez ograniczenia uchybu

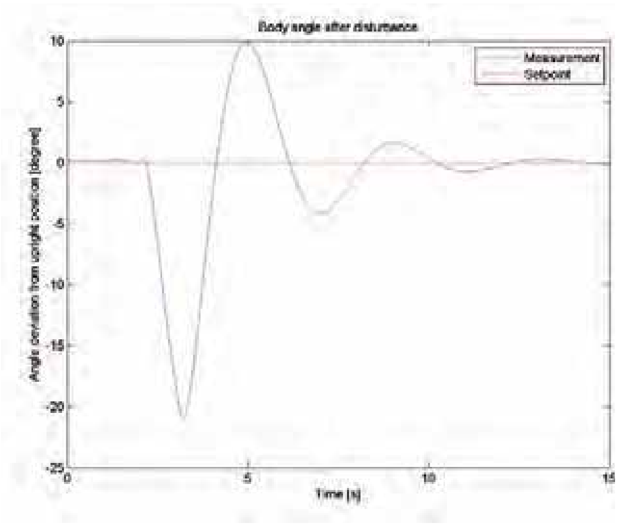

Fig. 9. Robot body angle response to external force

Rys. 9. Zmiany pochylenia robota w odpowiedzi na zewnętrzną siłę

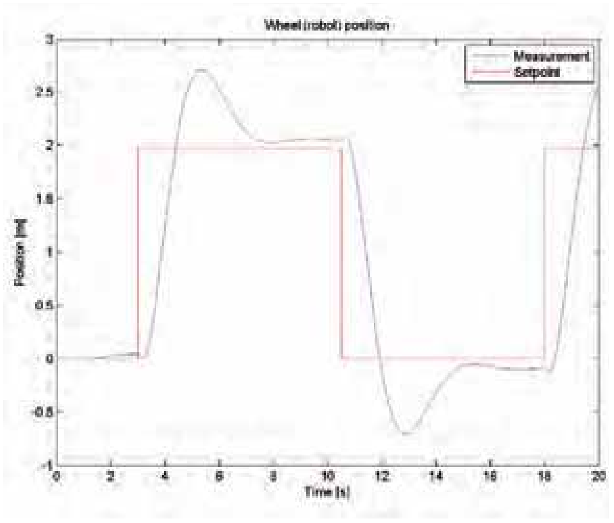

Fig. 11. Robot movement when a saturation block is applied

Rys. 11. Przemieszczanie robota – sterownik z dodatkowym blokiem ograniczającym

When robot is pushed, e.g. it has to stay upright and keep its initial position. The second task relies on stable movement of the robot.

Below, two simulations are presented. In the first test, a horizontally directed disturbance force equal 30 N, was applied for 1 s to the robot body. Fig. 8 and fig. 9 illustrate results of the test.

At the beginning of the experiment, between the 3rd and 4th second, the distance of the robot from the desired value achieves approximately 2,5 m value. During the next twelve seconds the distance is minimized to value 0.

Respectively, the angle deviation of the system achieves in the 3rd second maximal value equal 20 degree and the angular position stabilizes near the 15th second of the experiment.

The second experiment simulates behavior of the robot during a simple movement without external disturbances. The robot should change its position in a smoothly way and stop stable in a new (desired) position. The smooth movement is realized by giving a new set point. It should be taken into account, that the constructed LQ controller is as long valid, as model linearization conditions are fulfilled. If the robot angular position is too distant from the vertical position, the LQ controller stops being optimal, and the system may even become unstable. Such situation is presented in fig. 10. A big value of the linear distance error caused a large value of control and the robot swing overcrossed the space of linearization validity.

To avoid such situation the max value of the angle position error was limited by using a saturation block, so even when the robot should relocate on a big distance, the too big control force does not destabilize the system. Fig. 11 shows that oscillations are significantly diminished after adding the saturation. It should be noted, when the saturation block takes active part, the LQ controller does not calculate an optimal control. To assure optimality, a changes in the LQ objective function matrix are necessary. The changes should be done in such a way, the saturation block does not have to be active during the control process. However, tuning in this way the matrix of the LQ objective function not always turned into being effective in practice.

8. Controller code generation and deployment

Fig. 12 illustrates controller code generation and deployment workflow.

This model of controller created during simulations was used for code generation purposes [6]. The result of such generation consist of four parts.

Global input variables are equivalent to controller input ports. They are used by generated software and has to be read from IO drivers periodically.

Global output variables are equivalent to controller output ports. They are updated by generated software and has to be written to IO drivers,

1. Initialization function, called at program startup, allowing controller to operate with same initial conditions as during simulations,
2. Controller step function, which has to be executed periodically, with same period as during simulation.

Input and output signals are in raw, microcontroller friendly, format. Conversion from raw values to floating point SI units is part of model, so it was easier to test is in simulations.

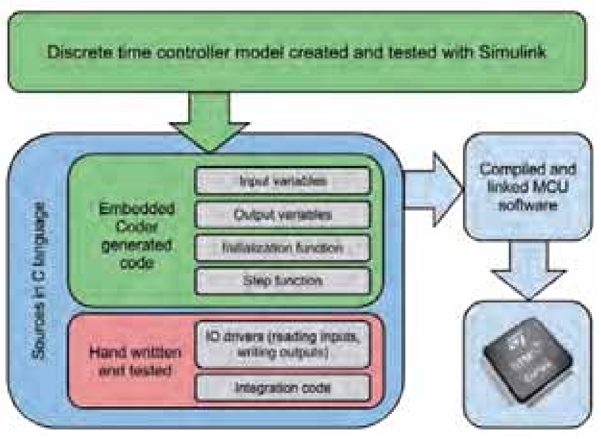
Discrete time controller model created and tested with Simulink
Sources in C language
Embedded Coder generated code
Input variables
Output variables
Initialization function
Step function
Hand written and tested
IO drivers (reading inputs, writing outputs)
Integration code
Compiled and linked MCU software

Fig. 12. Controller code generation and deployment workflow
Rys. 12. Przebieg procesu generacji kodu i kompilacji oprogramowania sterownika

Fig 13 illustrates how code generated from our model was integrated with IO drivers to create complete software. Because calculation time is not negligible, we decided to use (and model) delayed version of control loop.

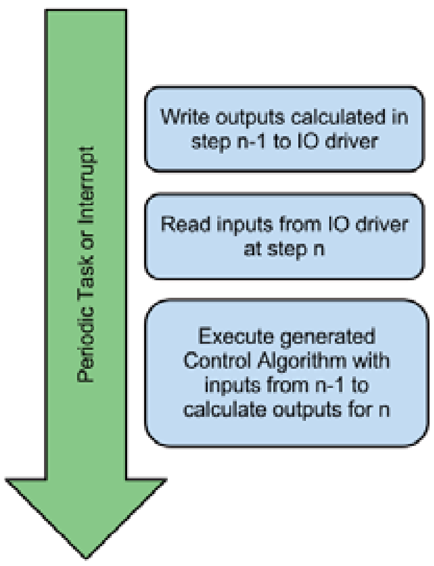
Periodic Task or Interrupt
Write outputs calculated in step n-1 to IO driver
Read inputs from IO driver at step n
Execute generated Control Algorithm with inputs from n-1 to calculate outputs for n

Fig. 13. Control algorithm step function integration
Rys. 13. Integracja algorytmu sterującego z pozostałą częścią oprogramowania

7. Summary

During the phase of designing the controller, many models were developed, tested and integrated in one simulation environment. The MBD approach allowed to design and to test a control algorithm without an access to the real, physically built robot. When the robot was ready and hardware drivers were tested, a final version of the software was downloaded to a target processor. Mathematical models proved to be accurate enough to realize in practice with success a stand up task and to stabilize the real robot in the upright position. Tests performed on the real object allowed further improvements, both in the plant model and in the control algorithm.

Acknowledgments

The results described in this paper were partly financed from the AGH grant for Student Science Associations.

We would like also to express our gratitude for ONT – Krakow company, for giving us an access to a trial version of a software.

Bibliography

1. Grasser F., D'Arrigo A., Colombi S., Rufer A.C., *JOE: A Mobile, Inverted Pendulum,* "IEEE Transactions On Industrial Electronics", Vol. 49, No. 1, February 2002, 107–114.
2. Ghodssi R., Lin P., *MEMS Materials and Processes Handbook.* Springer, Berlin 2011.
3. Paprocki K., *Mikrokontrolery STM32 w praktyce*, Wydawnictwo BTC, Legionowo 2009.

4. Pauluk M., *Projektowanie algorytmów sterujących w układach sterowanych napięciowo i prądowo* [*Designing of the control algorithms for systems controlled by voltage or current*], „Pomiary Automatyka Robotyka"; 2005 Vol. 8, No. 12, 37–45.
5. Nicolescu G., Mosterman P.J., *Model-Based Design for Embedded Systems* (*Computational Analysis, Synthesis, and Design of Dynamic Systems*), CRC Press Taylor & Francis Group, USA 2010.
6. Piątek P., Marchewka D., Pauluk M., *Automatyczna generacja kodu regulatora dla wbudowanego sterownika układu magnetycznej lewitacji* — [*Automatic code generation of embedded controller algorithm for magnetic levitation system purposes*], [in:] *Projektowanie, analiza i implementacja systemów czasu rzeczywistego*, praca zbiorowa (red. Trybus L., Samolej S.,), Wydawnictwa Komunikacji i Łączności, Warszawa 2011, 187–196.
7. Spong M.W., Vidyasagar M., *Dynamics and Control of Robots*, WNT, Warszawa 1997.
8. [www.mathworks.com].
9. [www.segway.com].
10. [www.solidworks.com].
11. Welch G., Bishop G., *Course 8 – An Introduction to the Kalman Filter*, University of North Carolina at Chapel Hill, Siggraph 2001.

Sterownik wbudowany robota balansującego

Streszczenie: Artykuł jest podsumowaniem prac nad projektem robota balansującego typu Segway. W ramach prac zaprojektowano platformę mechaniczną, sterownik elektroniczny oraz zaimplementowano algorytm sterowania robotem. Robot wyposażony jest w cyfrowy żyroskop i silniki prądu stałego z enkoderami. Sterownik w oparciu o odczyty z czujników steruje ruchem robota równocześnie stabilizując jego położenie. Część mechaniczną projektu wykonano przy użyciu oprogramowania CAD (SolidWorks). Zbudowane modele mechaniczne zaimportowano do środowiska MATLAB/Simulink, w którym opracowano i przetestowano sterownik. W trakcie prac wykorzystano techniki projektowania w oparciu o modele (ang. Model Based Design).

Słowa kluczowe: robot balansujący, Segway, sterowanie, czas rzeczywisty, modelowanie, projektowanie oparte na modelach

Krzysztof Wesołowski, Eng.

In 2011 completed first degree Studies in Automation and Robotics. During studies he worked on mobile robots as member of the Students Science Association "INTEGRA". Currently he is writing Master Thesis about balancing robots controller design. Since 2011 Krzysztof cooperates with Technika Obliczeniowa. Cooperation is focused on controllers for mechatronics devices.

e-mail: krzysztof.wesolowski@rainlabs.pl

Jakub Tutro, Eng.

In 2011 completed first degree Studies in Automation and Robotics. Currently he is working on his Master Thesis. Jakub has both academic and practical experience with electronic hardware design and mechatronics. He specializes in analog and mixed signal design, with emphasis on circuits applicable in real time control systems.

e-mail: jakub.tutro@rainlabs.pl

Dariusz Marchewka, PhD

He completed studies in Automation and Robotics at AGH University of Science and Technology in 1996. First he worked as assistant in Department of Automatics. He sucessfully defended his PhD in 2006. Since that he has worked as an assistant professor at the Department of Automatics and Biomedical Engineering on AGH University of Science and Technology. Actually his research focused on advanced real-time control algorithms of the mechatronics systems, developing of the embedded control systems. In addition, he is interested in industrial distributed control systems and mobile robots. Since 2008 he is a supervisor of the Students Team 'INTEGRA'

e-mail: dmar@agh.edu.pl

Mariusz Pauluk, PhD

In 1992 graduated from the AGH University of Science and Technology . Since that he was working as an assistant at the AGH department of automation and robotics. In 2001 received a PhD degree in the automation and robotics area. Since that he has worked as an assistant professor at the AGH automation and biomedical engineering cathedral. His research focuses on DSP technology, algorithms for controlling in real time and model based design techniques concerning multidimensional problems described by the non-linear differential equations.

e-mail: mp@agh.edu.pl