MATLAB-based Tools for Modelling and Control of Underactuated Mechanical Systems

Slávka Jadlovská [1], Lukáš Koska [2] and Matej Kentoš [3]

[1) 2) 3)] Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Košice, Slovakia

e-mail: [1)] *slavka.jadlovska@tuke.sk*, [2)] *lukas.koska@student.tuke.sk*, [3)] *matej.kentos@gmail.com*

Abstract — **Underactuated systems, defined as nonlinear mechanical systems with fewer control inputs than degrees of freedom, appear in a broad range of applications including robotics, aerospace, marine and locomotive systems. Studying the complex low-order nonlinear dynamics of appropriate benchmark underactuated systems often enables us to gain insight into the principles of modelling and control of advanced, higher-order underactuated systems. Such benchmarks include the Acrobot, Pendubot and the reaction (inertia) wheel pendulum. The aim of this paper is to introduce novel MATLAB-based tools which were developed to provide complex software support for modelling and control of these three benchmark systems. The presented tools include a Simulink block library, a set of demo simulation schemes and several innovative functions for mathematical and simulation model generation.**

Keywords — underactuated mechanical systems, nonlinear systems, Acrobot, Pendubot, Reaction Wheel Pendulum

I. Introduction

Underactuated systems represent an important group of mechanical systems which are highly nonlinear and have fewer independent control actuators than the degrees of freedom (DoFs) to be controlled. They appear in a broad range of applications in areas such as robotics, aerospace, marine and locomotive systems. Examples of the underactuated systems include underwater vehicles, surface vessels, helicopters, space robots, underactuated manipulators and mobile, notably legged robots. We can therefore see the significant practical use in the research of their properties [1].

The ultimate motivation behind the research into underactuation is the ability to control nonlinear systems without complete control authority by exploiting their natural dynamics. This is similar to how biological systems execute motions involving a loss of instantaneous control authority. The underactuated devices are therefore expected to be more efficient, simpler and more reliable than their fully actuated alternatives [2]. However, control of the underactuated devices is more complicated to design theoretically. This is caused by several structural properties, such as feedback linearizability, that are lost with the decreasing number of actuators. Building this type of control systems requires deep understanding of model dynamics, which is the reason behind the need to obtain high-quality and accurate mathematical models. Studying the complex low-order nonlinear dynamics of the appropriated benchmark underactuated systems often enables us to gain insight into the principles of modelling and control of advanced, higher-order underactuated systems.

Inverted pendulum systems have a prominent position among the benchmark underactuated systems. Stabilization of a physical pendulum or a system of interconnected pendulum links in the unstable upright position is a frequently solved problem in nonlinear control theory [3]. Knowing this, we spent several years developing and improving a custom Simulink block library, Inverted Pendula Modelling and Control. The intermediate results were published in 2009 [4], 2011 [5] and 2014 [6]. The library now includes a number of blocks and applications for modelling and control of the inverted pendulum systems attached either to a cart or rotary arm. The library centers on an algorithmic procedure, implemented using Symbolic Math Toolbox, which generates the mathematical model for a system with a given number of pendulum links. The procedure is based on the concept of a generalized (n-link) inverted pendulum system, which allows us to treat an arbitrary system of interconnected inverted pendula as a particular instance of the system of n pendula attached to a given stabilizing base. In [6], the procedure was expanded to cover all possible combinations of underlying assumptions for the reference pendulum angle value and reference direction of the pendulum rotation.

In recent years, mechatronic systems such as the *Acrobot*, *Pendubot*, and *Reaction (Inertia) Wheel Pendulum* have been regularly employed in control structures as examples of unstable nonlinear underactuated systems. All are lower-degree underactuated systems which correspond to different configurations of the inverted pendula (Fig. 1) with a torque input. The *Acrobot* and *Pendubot* were designed as two-link planar robots which share the same inertia matrix. In the case of the *Acrobot*, the actuator is placed at the elbow, while the Pendubot is actuated at the shoulder. The *reaction wheel pendulum* is composed of a physical pendulum with a rotating uniform inertia wheel at the end of the pendulum rod which is not directly actuated: in order to stabilize the pendulum in the upright equilibrium, the system has to be controlled via the rotating wheel. Each of these systems was artificially developed for the purpose of better understanding of the underactuated dynamics: the *Acrobot*, patterned after a gymnast (acrobat) performing on a single parallel bar, was presented by Murray and Hauser in [7], while the *Pendubot* and the *Reaction Wheel Pendulum* were introduced, originally for educational purposes, by M. Spong et al. in [8] and [9], respectively.

TELEN2017003
DOI 10.14311/TEE.2017.3.056

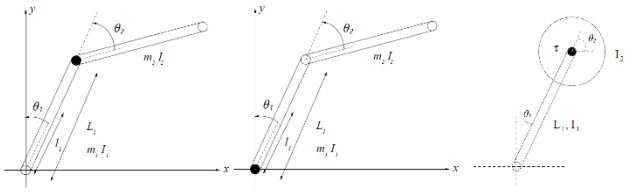

Fig. 1. Acrobot, Pendubot and Reaction Wheel Pendulum.

This paper focuses on the presentation of a novel *Simulink* block library, *Underactuated Systems*, which has been developed as a comprehensive software framework for the analysis and control problems of three presented benchmark underactuated systems. It follows the steps of its predecessor, the *Inverted Pendula Modelling and Control* (*IPMaC*) block library, developed for classical and rotary inverted pendulum systems. The reader is referred to [5] for a detailed description of the *IPMaC* functionality. The libraries have several aspects in common, e.g. an algorithmic procedure for generating mathematical models, or a set of links to demo simulation schemes which illustrate the ways of interconnecting library blocks to solve a variety of problems.

II. Simulink Block Library – *Underactuated Systems*

The *Underactuated Systems* block library was developed in MATLAB/Simulink and can be used exclusively in this program environment. MATLAB versions 7.12.0 (R2011a) and 8.1 (R2013a) were used for the development and compatibility testing. The installation process consists of unzipping the provided package into a desired directory and running the included installation script *slblocks.m* which adds the library to the *Simulink Library Browser*. As a result, we can open the browser and navigate to our library in *Libraries* section where all our blocks and tools are available. Naturally, all blocks are fully compatible with the blocks from the rest of Simulink built-in libraries.

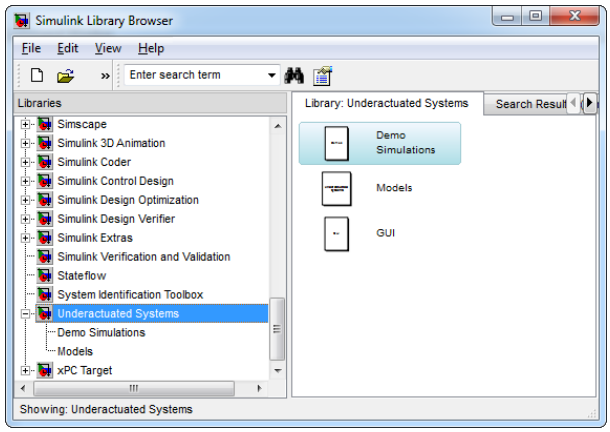

Fig. 2. *Underactuated Systems* block library, installed and active in *Simulink Library Browser*.

A. Automatic Mathematical Model Generation for Benchmark Underactuated Systems

As was the case of the *IPMaC*, the *Underactuated Systems* block library centers around an algorithmic procedure which generates the mathematical model for a selected underactuated system using the Lagrange equations of the second kind, defined as:

$$\frac{d}{dt}\left(\frac{\partial L(t)}{\partial \dot{\boldsymbol{\theta}}(t)}\right)-\frac{\partial L(t)}{\partial \boldsymbol{\theta}(t)}+\frac{\partial D(t)}{\partial \dot{\boldsymbol{\theta}}(t)}=\mathbf{Q}^{*}(t), \tag{1}$$

where $L(t)$ is the difference between the multibody system kinetic and potential energies, $D(t)$ stands for the dissipation properties and $\mathbf{Q}^{*}(t)$ is the vector of generalized external inputs. In all three cases, the vector of generalized coordinates is defined as $\boldsymbol{\theta}(t)=\left(\theta_1(t) \quad \theta_2(t)\right)$, which corresponds to the two degrees of freedom of each system: the link angles of the *Acrobot/Pendubot*, or the link and rotor angle of the *reaction wheel pendulum*. The procedure of the mathematical model derivation was implemented for all systems using *Symbolic Math Toolbox* as the *underactuated_modelling.m* function. The underlying physical formulae are derived and presented in detail in [11]. All possible combinations of the reference positions of individual links as well as the reference directions of the link rotation are considered. This approach was motivated by our ambition to ultimately cover any configuration of the motion equations found in the relevant sources.

The results of this function can optionally be depicted in the user-friendly, well-arranged format provided by a novel application with the graphical user interface: *Automatic Modelling for Underactuated Systems*. Figure 3 shows an example preview of the application window which contains the generated motion equations for the *Acrobot* system. Four reference positions for the pendulum links (up, down, right, left) and two reference directions of the link rotation (clockwise, counter-clockwise) are available for the *Acrobot*, meaning that eight separate sets of the motion equations could be derived to describe it.

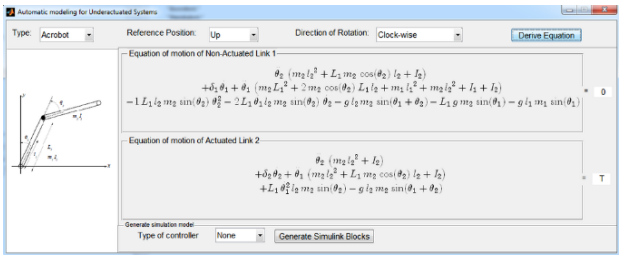

Fig. 3. *Automatic Modelling for Underactuated Systems* – GUI application for generating.

Finally, we developed a function (*ode_form.m*) which parses the symbolic variables at the output of the *underactuated_modelling.m* into the standard minimal (ordinary differential equation – ODE) form, defined as

$$\mathbf{M}\left(\boldsymbol{\theta}(t)\right)\ddot{\boldsymbol{\theta}}(t)+\mathbf{N}\left(\dot{\boldsymbol{\theta}}(t),\boldsymbol{\theta}(t)\right)\dot{\boldsymbol{\theta}}(t)+\mathbf{P}\left(\boldsymbol{\theta}(t)\right)=\mathbf{V}(t)u(t) \tag{2}$$

which is a prerequisite for linearization and control design for the mechatronic systems [12].

B. Simulation Models of Underactuated Systems – Implementation and Verification

As shown above, the developed application for the automatic mathematical model generation lets the user choose from several combinations of initial assumptions about the models. Instead of building a separate *Simulink* block using each combination of the reference position and direction of rotation, we implemented two advanced approaches for developing the simulation models.

TELEN2017003
DOI 10.14311/TEE.2017.3.056

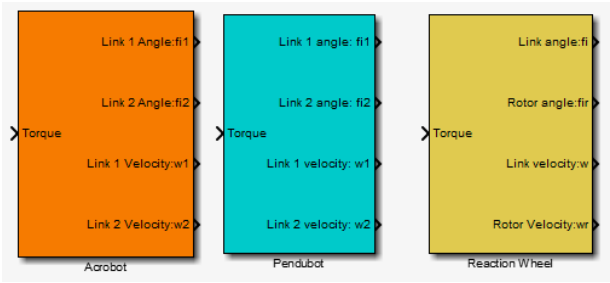

Fig. 4. Simulation models of Acrobot, Pendubot and Reaction Wheel Pendulum.

The first approach is represented by the pre-prepared model blocks, which are shipped as a part of the library and can be dynamically modified to reflect the selected combination of the initial assumptions. The hierarchical structure of all blocks consists of three levels. At the highest level (Fig. 4), the system mask allows us to set the reference angle value, reference direction of rotation, physical parameters and the required number of the input and output ports. The middle level is represented by the *Link 1/Link 2* subsystems in the case of the *Acrobot*/*Pendubot*, or *Link*/*Rotor* in the case of the *Reaction Wheel Pendulum*. Connections between the individual subsystems represent their mutual physical impact. The lowest level is represented by the underlying differential equations of the system in form of the interconnected built-in and custom Simulink blocks. The structure of the simulation models at this level is dynamically adjusted to mirror the initial assumptions using a callback that modifies the chain of the signs in the *Sum* block and the goniometric functions in the blocks *d and e*, as can be seen in Fig. 4 example [11].

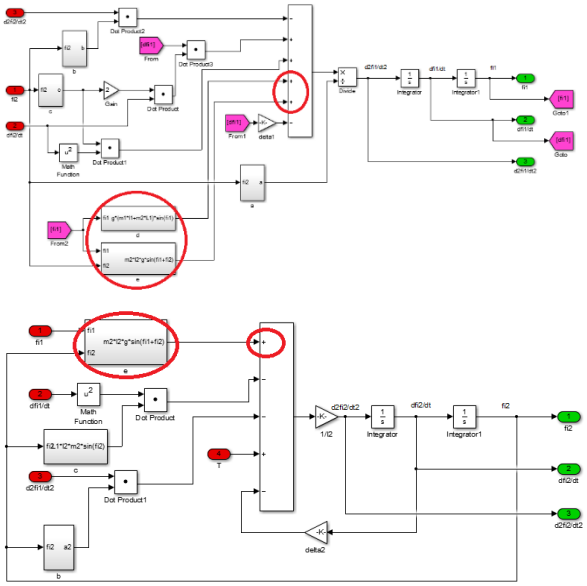

Fig. 5. Low level architecture of the *Link 1/Link 2* subsystems of the *Acrobot* simulation model.

The second approach is based on the MATLAB feature which allows to add and to interconnect blocks in the simulation schemes from the MATLAB code. Once the mathematical model has been successfully generated, we can select the *Generate Simulink Blocks* button in the *Automatic Modelling for Underactuated Systems* application. The *scheme.m* function is then called which builds up a simulation model from the lowest to the highest level by gradually translating the motion equations generated for a given set of the initial assumptions into the language of *Simulink* blocks, allowing the user to easily build the simulation models from the scratch. The resulting model structure looks analogical to the one in Figs. 4, 5 [12].

Using appropriate simulation experiments from the *demo simulations* section of the *Underactuated Systems* library, we will now evaluate the open-loop behavior of the considered systems, as well as the influence of the initial assumptions on their response [11].

The comparison of the time behavior of the Acrobot and Pendubot system is depicted in Fig. 6. In both cases, after applying an impulse input, both links fall from the upright into the downward equilibrium through a damped oscillatory transient state and stabilize there, which agrees with the empirical observations of the pendula behavior. (Note: $\theta_2(t)$ is determined *relative* to $\theta_1(t)$).

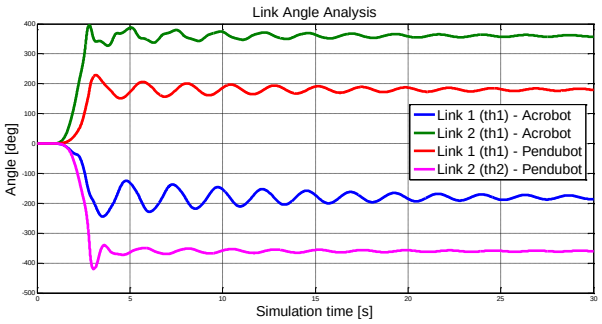

Fig. 6. Open-loop behavior of Acrobot and Pendubot.

Figure 7 depicts the open-loop dynamical behavior of the *Pendubot* simulation model after selecting four reference positions which vary by 90 ° (up, left, down, right). The link angle is determined clockwise in all cases. It is clear that changes in the initial assumptions have no effect on the dynamics of the system, and only the numerical representation of the link behavior is subject to change – in all simulations, the system starts in the upright position, but the corresponding angle value changes for every new reference position.

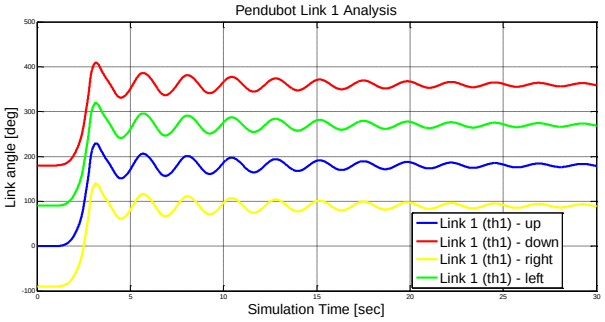

Fig. 7. Simulations of a Pendubot first link – effect of the changing reference positions.

The reasonable behavior of the open-loop responses of all simulation models means that under all criteria, systems described by the generated motion equations can be considered accurate enough to serve as a reliable testbed for the verification of the linear and nonlinear control algorithms.

TELEN2017003
DOI 10.14311/TEE.2017.3.056

C. Swing-up and Stabilization of Benchmark Underactuated Systems

The principal control objective for all considered nonlinear underactuated systems is the *stabilization in the unstable equilibrium*, i.e. in the vertical upright (inverted) position of all links of the system [3]. We will now consider two frequently solved model situations, in which a varying *initial position of the system links* calls for a corresponding control setup. Both situations will be illustrated by simulation experiments from the *demo simulations* section of the library. On the one hand, it is sufficient to design a stabilizing state-feedback controller if the pendulum links are kept in the proximity of the upright position (i.e. we are solving an initial deflection or a disturbance compensation problem). On the other hand, the pendulum links in the natural hanging position require an additional mechanism which swings them up to the upright position before they can be stabilized, as well as a mechanism to switch between the two controllers. The underactuated benchmarks can therefore be also considered as suitable testbed systems for the verification of the hybrid control approaches.

The goal of the optimal control design for a linear, time-invariant dynamic system is to determine such feedback control so that a given criterion of optimality is achieved. In case the considered linear system is actually a linear approximation of a nonlinear system around a given equilibrium point, then the optimal control techniques designed for linear systems yield an approximate, locally near-optimal stabilizing solution to the problem with the guaranteed closed-loop stability and robustness. Assuming a continuous-time linear system that represents a linear approximation of the nonlinear system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t)) \tag{3}$$

in a given equilibrium, then the goal of the optimal LQR control design is to determine such a feedback gain vector $\boldsymbol{k}$ so that the resulting feedback control law, defined as

$$u(t) = -\mathbf{k}\mathbf{x}(t) \tag{4}$$

would minimize the quadratic criterion given as

$$J(u) = \int_0^{\infty} \left(\mathbf{x}^T(t)\mathbf{Q}\,\mathbf{x}(t) + u^T(t)Ru(t)\right)dt \tag{5}$$

The application of the *LQR optimal state-feedback control techniques* in the stabilizing control of inverted pendulum systems (as a specific case of nonlinear underactuated systems) has already been extensively studied in [5], [13] using the model and controller blocks from the *IPMaC*. For this paper, the controller blocks and control structures from the *IPMaC* have not only been successfully reused, but a custom function has been implemented which builds up a whole control structure in *Simulink* in the same way as in the system modelling and computes the linearized model for the given equilibrium.

Figure 8 illustrates the results of applying the state-feedback control law based on the continuous-time LQR algorithm on the *Acrobot* system which has been steered away from the upright position. The effect of two weight matrices on the time behavior of the system has been compared.

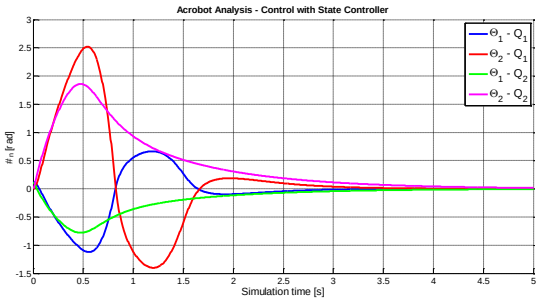

Fig. 8. Stabilization of the *Acrobot* in the upright unstable position.

If the initial position of the system link is equivalent to the natural hanging position, then the additional control problem of swinging it upwards leads to a hybrid control setup which consists of a swing-up controller, stabilizing (balancing) controller and transition (switching) mechanism which intercepts the system links when they near the upright position (i.e. cross the borderline of the balancing region) and switches to a state-feedback stabilizing control algorithm described above. Such a switching control structure is depicted in Fig. 9. The objective is to swing up and stabilize the Acrobot in the unstable upright position.

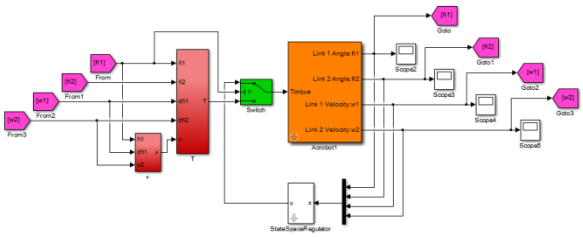

Fig. 9. Swing-up and subsequent stabilization of the Acrobot in the upright position.

By implementing the *partial feedback linearization technique* as the swing-up control algorithm we are able to make use of the natural dynamics of the model in the closed loop [11], [12]. Since the underactuated systems are not feedback linearizable, part of the dynamics corresponding to the actuated degrees of freedom linearized with a nonlinear feedback. As it can be seen from Fig. 10, the links of the system acquire energy with every swing until the *Acrobot* enters the area surrounding the unstable equilibrium. The control law then switches to the balancing control which successfully stabilizes both links of the system.

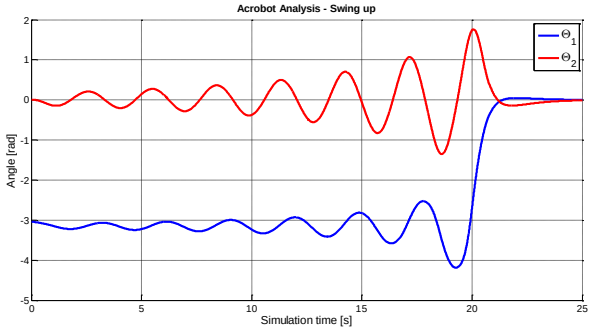

Fig. 10. Swing-up and subsequent stabilization of the Acrobot in the upright position.

TELEN2017003
DOI 10.14311/TEE.2017.3.056

In laboratory applications of the underactuated systems, a controllable DC motor can be the source of the torque input. As a result, the dynamics of the DC motor should be reflected in the mathematical model of the system. In [14], the mathematical model of the DC motor was obtained in form of a voltage-to-torque conversion relationship. This can next be substituted into the generated *torque model* of either the Acrobot, Pendubot or Reaction Wheel Pendulum system, resulting in the *voltage model* of the underactuated system. Figure 11 depicts the results of the stabilizing control applied on the Acrobot voltage model; this time, the LQR algorithm generates the desired voltage for the DC motor.

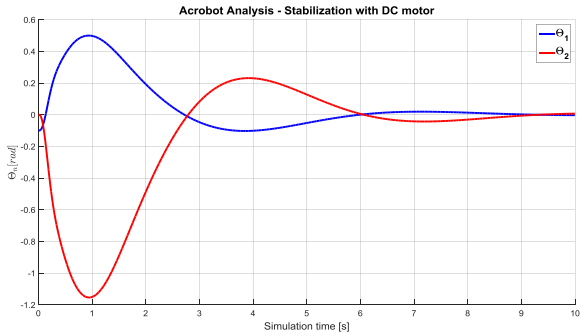

Fig. 11. Stabilization of the Acrobot with a DC motor in the upright position.

Alternative approaches for the model-based control of the laboratory models of the underactuated benchmarks with a more complicated actuating mechanism include a combination of experimental and analytical identifications. We already employed this approach in [15] to control a laboratory model of a single inverted pendulum on a cart with a linear synchronous motor.

III. Conclusion

The purpose of this paper was to describe novel MATLAB-based tools for modelling and control of some important benchmark underactuated systems – two-link planar robots *Acrobot* and *Pendubot* and the *Reaction Wheel Pendulum*. The block library presented in the paper provides the control engineer with a readily available collection of highly accurate mathematical and simulation models of the selected benchmark underactuated systems in various configurations. A mathematical model for a selected underactuated system is generated by an algorithmic procedure which has been implemented in form of a symbolic MATLAB function with an optional graphical user interface. Two advanced approaches for developing the simulation models are supported. The first approach involves manually-built Simulink blocks, which can be dynamically modified to reflect the selected reference position and direction of the system link rotation. The second approach is based on a MATLAB function which builds up a simulation model by gradually translating the motion equations generated for a given set of the initial assumptions into the language of *Simulink* blocks.

The developed models have been used in a number of simulation experiments designed to explore their properties and to verify appropriate control design strategies. The validity and accuracy of generated motion equations are confirmed by evaluating the open-loop responses of the simulation models, and the control objective of stabilization in the unstable upright position with an optional swing-up from the downward into the upright equilibrium (using a switching control structure) was successfully implemented. The obtained results have provided us with a starting point for the research of advanced underactuated systems, notably the *biped robots and the mechanism of robot walking*. Walking robots are often based on the benchmark underactuated systems and manage to exploit their dynamics to create naturally-looking gait. The hybrid models and switching control structures are useful if event-based dynamics of a legged locomotion is considered.

Acknowledgment

This publication is the result of the project implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology - 2nd Phase, ITMS: 313011D232 (40 %) and the "Centre of Information and Communication Technologies for Knowledge Systems", ITMS: 26220120020 (20 %), both supported by the Operational Programme Research & Development funded by the ERDF, grant TUKE FEI-2015-33: Research Laboratory of Nonlinear Underactuated Systems (30 %) and the project KEGA - 001TUKE-4/2015 (10 %).

References

[1] M. W. Spong, Underactuated Mechanical Systems: Control Problems in Robotics and Automation. In: Lecture Notes in Control and Information Sciences, vol. 230, 1998, pp. 135-150.

[2] R. Tedrake, Underactuated Robotics: Learning, Planning and Control for Efficient and Agile Machines. Course Notes for MIT 6.8., Cambridge: Massachusetts Institute of Technology, 2009.

[3] S. Jadlovská, J. Sarnovský. Modelling of Classical and Rotary Inverted Pendulum Systems – a Generalized Approach. In: Journal of Electrical Engineering, vol. 64, no. 1, 2013, pp. 12–19, ISSN 1335-3632.

[4] S. Jadlovská, A. Jadlovská. A Simulink Library for Inverted Pendula Modelling and Simulation. Proc. of the 17th Int. Scient. Conf. Technical Computing 2009, Prague, Czech Rep., Nov. 19, 2009, ISBN 978-80-7080-733-0.

[5] S. Jadlovská, J. Sarnovský. An Extended Simulink Library for Modelling and Control of Inverted Pendula Systems. Proc. of the 19th Int. Scient. Conf. Technical Computing 2011, Prague, Czech Rep., Nov. 8, 2011, ISBN 978-80-7080-794-1.

[6] S. Jadlovská, J. Sarnovský, J. Vojtek, D. Vošček. Advanced Generalized Modelling of Classical Inverted Pendulum Systems. In: Advances in Intelligent Systems and Computing: Emergent Trends in Robotics and Intelligent Systems. Vol. 316, Switzerland : Springer, 2014 p. 255-264, ISBN 978-3-319-10782-0, ISSN 2194-5357

[7] R. M. Murray, J. Hauser. A Case Study in Approximate Linearization: The Acrobot Example. Proc. of the American Control Conference, 1991

[8] M. W. Spong. The Pendubot: a Mechatronic System for Control Research and Education. Proc. of the 34th IEEE Conf. on Decision and Control, New Orleans, USA, 1995 https://doi.org/10.1109/CDC.1995.478951

[9] M. W. Spong, P. Corke, R. Lozano. Nonlinear Control of the Reaction Wheel Pendulum. Automatica, Vol. 37, No. 11, pp. 1845–1851 https://doi.org/10.1016/S0005-1098(01)00145-5

[10] H. Goldstein, Ch. Poole, J. Safko. Classical Mechanics, 3rd ed. Addison-Wesley, 2001. 680 p.

[11] M. Kentoš. Modelling and Control of Underactuated Mechanical Systems [Modelovanie a riadenie podaktuovaných mechanických systémov]. Bachelor Thesis. Supervisor: prof. Ing. J. Sarnovský, CSc, consultant: Ing. S. Jadlovská, FEEI-TU, 2014.

TELEN2017003
DOI 10.14311/TEE.2017.3.056

[12] L. Koska. Modelling, Simulation and Control of Nonlinear Mechanical Systems [Modelovanie, simulácia a riadenie nelineárnych mechanických systémov]. Bachelor Thesis. Supervisor: prof. Ing. J. Sarnovský, CSc, consultant: Ing. S. Jadlovská, FEEI-TU, 2015.

[13] S. Jadlovská, J. Sarnovský. A Complex Overview of Modelling and Control of the Rotary Single Inverted Pendulum System. In: Advances in Electrical and Electronic Engineering, vol. 11, no. 2, 2013, pp. 73–85, ISSN 1336-1376.

[14] S. Jadlovská. Modelling and Optimal Control of Nonlinear Underactuated Dynamical Systems [Modelovanie a optimálne riadenie nelineárnych podaktuovaných dynamických systémov]. Dissertation Thesis. Supervisor: prof. Ing. J. Sarnovský, CSc, FEEI-TU, 2015.

[15] A. Jadlovská, S. Jadlovská, D. Vošček. Cyber-physical System Implementation into the Distributed Control System. In: ScienceDirect : IFAC-PapersOnLine. - Amsterdam: Elsevier, 2016 Vol. 49, no. 25 (2016), p. 031-036. - ISSN 2405-8963.

TELEN2017003
DOI 10.14311/TEE.2017.3.056