A Complex Overview of the Rotary Single Inverted Pendulum System

Slávka Jadlovská* and Ján Sarnovský*

* Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Košice, Slovak Republic

e-mail: *slavka.jadlovska@tuke.sk, jan.sarnovsky@tuke.sk*

***Abstract*— The purpose of this paper is to perform an in-depth analysis of the rotary single inverted pendulum system using a *Simulink* block library designed by the authors of this paper – the *Inverted Pendula Modeling and Control* (*IPMaC*), which offers comprehensive program support for modeling, simulation and control of classical and rotary inverted pendula systems. With the aid of appropriate function blocks, GUI applications and demonstration schemes from the *IPMaC*, the rotary single inverted pendulum system is analyzed, modeled and successfully stabilized in the unstable inverted position.**

***Keywords*—rotary single inverted pendulum, custom block library, automatic model generation, state-feedback control**

I. Introduction

Inverted **p**endula **s**ystems (IPS) represent a significant class of nonlinear underactuated mechanical systems, well-suited for the verification and practice of ideas emerging in control theory and robotics. Stabilization of a pendulum rod in the unstable upright position is considered a benchmark control problem which has been solved by attaching the pendulum to a base that moves in a controlled linear manner (*classical IPS*) or in a rotary manner in a horizontal plane (*rotary IPS*).

The *Inverted Pendula Modeling and Control (IPMaC)* is a structured *Simulink* block library which was developed by the authors of this paper and provides complex software support for the analysis and control of both classical and rotary IPS [1][2]. Strong emphasis is placed on the generalized approach to system modeling [1][3], allowing the library to handle systems which differ by the number of pendulum links attached to the base, such as single [2][3], double [2][4] and triple IPS.

As an underactuated, unstable and yet controllable system, the *rotary single inverted pendulum* has been an attractive testbed system for linear and nonlinear control law verification ever since it was introduced to feedback control community by Katsuhisa Furuta, Professor at the Tokyo Institute of Technology [5]. This paper aims to present an overview of this popular system, covering every significant step from mathematical model derivation to examples of control algorithm design. The individual steps of the process will be demonstrated using suitable function blocks or GUI applications from the *IPMaC*. All simulation experiments mentioned in this paper can be run from the *IPMaC*/*Demo Simulations* section which contains links to corresponding schemes.

II. Mathematical Modeling and Simulation of the Rotary Single Inverted Pendulum

The considered *rotary single inverted pendulum system* (Fig. 1) is composed of a homogenous pendulum rod attached to an arm which is free to rotate in a horizontal plane. Since the number of actuators is lower than the number of system links, the system is underactuated: the only input (the torque $M(t)$ applied on the arm) is used to control the two degrees of freedom of the system: arm angle $\theta_0(t)$ [rad] and pendulum angle $\theta_1(t)$ [rad].

A. Automatic Derivation of Motion Equations

Manual, step-by-step derivation of motion equations is the prevailing approach to inverted pendula modeling which can be found in the accessible works, e.g.[6][7][8]. However, in this paper, the mathematical model of the system will be generated *automatically* using the *Inverted Pendula Model Equation Derivator* (Fig. 2), a MATLAB GUI application from the *IPMaC*. The *Derivator* generates the motion equations for user-chosen types of IPS (classical/rotary, single/double) with the aid of an original procedure of mathematical model derivation for the generalized (*n-link*) system which was implemented in MATLAB using *Symbolic Math Toolbox* [1][3].

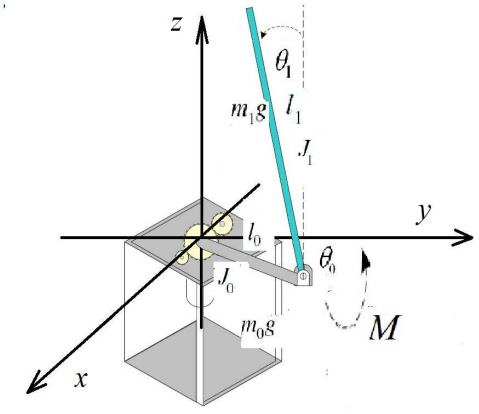

Fig. 1. Rotary single inverted pendulum – scheme and parameter nomenclature.

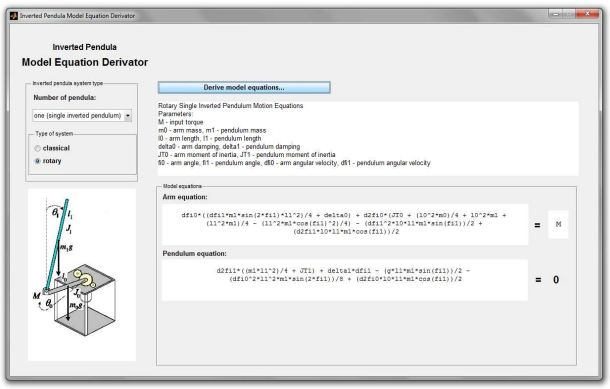

Fig. 2. Automatic derivation of the rotary single inverted pendulum motion equations using *Inverted Pendula Model Equation Derivator*.

As it can be seen in the preview of the *Derivator* window (Fig. 2), the mathematical model of the *rotary single inverted pendulum system* is composed of two second-order nonlinear differential equations which respectively describe the dynamic behavior of the rotary arm and the pendulum. If the equations are rearranged into the so-called *standard minimal ODE (ordinary differential equation)* matrix form:

$$\boldsymbol{M}\left(\boldsymbol{\theta}(t)\right)\ddot{\boldsymbol{\theta}}(t)+\boldsymbol{N}\left(\boldsymbol{\theta}(t),\dot{\boldsymbol{\theta}}(t)\right)\dot{\boldsymbol{\theta}}(t)+\boldsymbol{P}\left(\boldsymbol{\theta}(t)\right)=\boldsymbol{V}(t), \quad (1)$$

given that $\boldsymbol{\theta}(t)=\left(\theta_0(t) \quad \theta_1(t)\right)^T$, the following mathematical model of the system is obtained:

$$\begin{pmatrix} J_0+m_1l_0^2+\frac{1}{4}m_1l_1^2\sin^2\theta_1(t) & \frac{1}{2}m_1l_0l_1\cos\theta_1(t) \\ \frac{1}{2}m_1l_0l_1\cos\theta_1(t) & J_1 \end{pmatrix}\begin{pmatrix}\ddot{\theta}_0(t)\\ \ddot{\theta}_1(t)\end{pmatrix}+$$
$$+\begin{pmatrix} \delta_0+\frac{1}{4}m_1l_1^2\dot{\theta}_1(t)\sin 2\theta_1(t) & -\frac{1}{2}m_1l_0l_1\dot{\theta}_1(t)\sin\theta_1(t) \\ -\frac{1}{8}m_1l_1^2\dot{\theta}_0(t)\sin 2\theta_1(t) & \delta_1 \end{pmatrix}\begin{pmatrix}\dot{\theta}_0(t)\\ \dot{\theta}_1(t)\end{pmatrix}+ \quad (2)$$
$$+\begin{pmatrix} 0 \\ -\frac{1}{2}m_1gl_1\sin\theta_1(t)\end{pmatrix}=\begin{pmatrix}M(t)\\ 0\end{pmatrix}$$

where m_0, m_1 stand for the masses of the arm and the pendulum, l_0, l_1 are their respective lengths, δ_0, δ_1 are the damping constants in the joints of the arm and pendulum, $J_0=\frac{1}{3}m_0l_0^2$ and $J_1=\frac{1}{3}m_1l_1^2$ are the moments of inertia of the arm and pendulum with respect to their pivot points and $M(t)$ is the input torque applied upon the rotary arm. The generated model will hereafter be referred to as a *torque model* of a rotary single inverted pendulum system, to distinguish it from a *voltage model* of the system, which will be presented later.

A number of differences have been noticed between the generated model and the models in the referenced works. The general procedure used by the *Derivator* assumes that all motion of the system is bound to a standard right-handed three-dimensional coordinate system (and most differences can be attributed to the choice of coordinate system or angle orientation) and that the rotary motion of the pendulum takes place in a vertical plane that is always perpendicular to both the horizontal arm plane and the arm itself – as a result, the actual plane the pendulum is rotating in is different in every instant, which brings additional complexity as well as high accuracy to the generated mathematical model. In the referenced works, the pendulum is often assumed to be rotating in a constant plane [6][7] or the rotary motion of the arm is neglected altogether [8].

B. *Open-Loop Dynamical Analysis*

The *Inverted Pendula Models* sublibrary of the *IPMaC* contains a library block *Rotary Single Inverted Pendulum* (*RSIP*), which implements the mathematical model (2) derived above. The block is equipped with a dynamic block mask [2] which enables the user to edit the numeric parameters and initial conditions, to enable or disable the torque input port and to flexibly adjust the number of the block's output ports (Fig. 3).

The open-loop dynamical behavior of the rotary single inverted pendulum system was verified in a simulation experiment involving the *RSIP* block (Fig. 4). The system was actuated from an initial – upright position of the

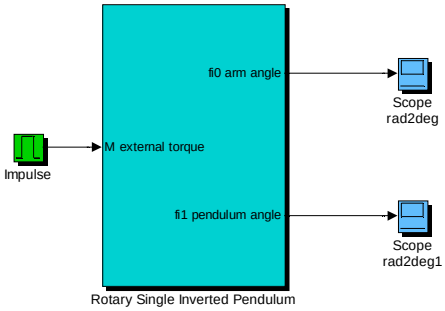

Fig. 3 Open-loop dynamical analysis of the rotary single inverted pendulum (torque model) - simulation setup.

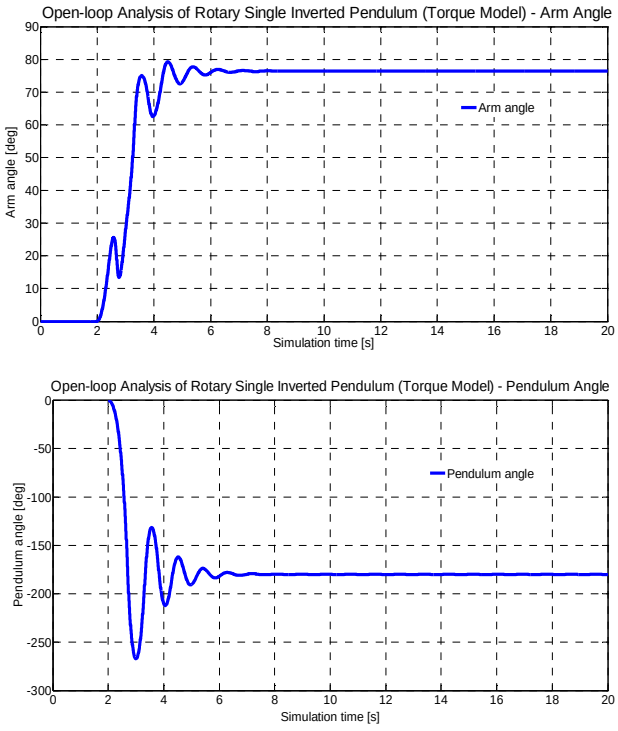

Fig. 4. Rotary inverted pendulum (*torque model*) – open-loop time behavior of the arm and pendulum angles.

pendulum by a torque impulse of 0,4 Nm lasting 1s. The numeric parameters of the simulated system were selected to be $m_0 = 0.5kg$, $m_1 = 0.275kg$, $l_0 = 0.6m$, $l_1 = 0.5m$, $\delta_0 = 0.3kgs^{-1}$, $\delta_1 = 0.011458kgm^2s^{-1}$.

Reasonable behavior of the open-loop response (damped oscillatory transient state, system reaching the stable equilibrium point with the pendulum pointing downward, visible backward impact of the pendulum on the base) means that the simulation model implemented as the *RSIP* library block can be considered accurate enough to serve as a reliable testbed system for linear and nonlinear control algorithms.

C. *Actuating Mechanism Implementation*

To provide the simulation models of IPS with a model of the most frequently used actuating mechanism, a library block *DC Motor for Inverted Pendula Systems* was included into the the *Inverted Pendula Motors* sublibrary of the *IPMaC*. The block implements the mathematical model of a brushed direct-current (DC)

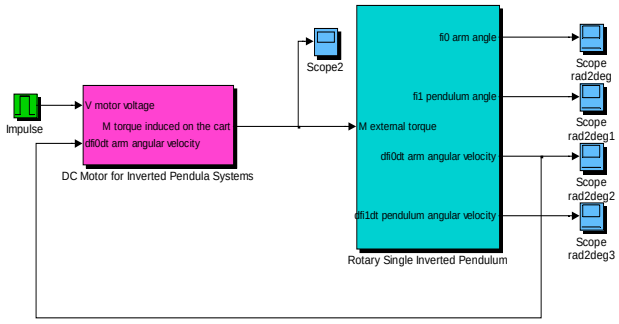

Fig. 5 Open-loop dynamical analysis of the rotary single inverted pendulum (voltage model) - simulation setup.

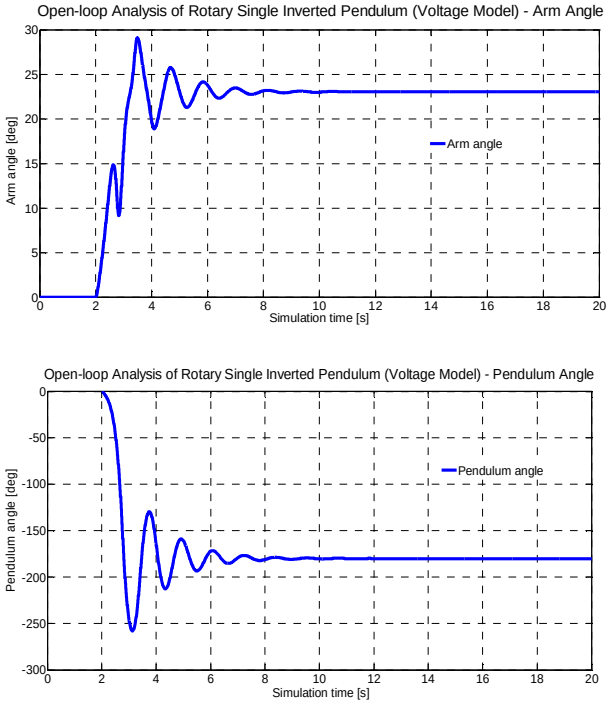

Fig. 6 Rotary single inverted pendulum (voltage model) – open-loop time behavior of the arm and pendulum angles.

motor in two alternative forms depending on the type of system (classical / rotary). For rotary IPS, the DC motor is modeled as the following voltage-to-torque conversion relationship, derived in [1]:

$$M(t) = \frac{k_m k_g}{R_a} V_a(t) - \frac{k_m^2 k_g^2}{R_a} \dot{\theta}_0(t) \cdot \qquad (3)$$

where $V_a(t)$ is the input voltage applied to the motor, k_m is the motor torque constant, equal in value to the back EMF constant, k_g is the gear ratio, and R_a is the armature resistance.

If the DC motor model is appended to an inverted pendulum system (i.e. (3) is substituted into (2)), a *voltage model* of the system is obtained (Fig. 5). In Fig. 6, the response of the rotary inverted pendulum voltage model to an impulse signal of 1V lasting 1s is depicted. The numeric parameters of the DC motor simulation model were borrowed from the motor featured in the series of popular laboratory models of inverted pendula systems issued by *Quanser Academic* [9].

III. Stabilization of the Rotary Single Inverted Pendulum via State-Feedback Control Techniques

The principal control objective was defined as the *stabilization of the system in the unstable equilibrium, i.e. in the vertical upright (inverted) position of the pendulum*. In other words, the pendulum needs to be balanced around the upright position after an initial deflection (nonzero initial conditions) and every time-constrained (impulse) or permanent (step) disturbance input signal has to be compensated. Meanwhile, the arm must track a desired reference trajectory. *Linear state-feedback control* was emphasized as the principal control technique because control of several degrees of freedom at once can only be ensured if they are all taken into consideration. Linear approximation is a necessary prerequisite for this approach to control.

The additional problem of swinging the pendulum up from the pendant to the upright position leads to a hybrid control setup with two additional components: a swing-up controller and a transition (switching) mechanism, which intercepts the pendulum when it reaches the upright position and switches to balancing control. A control strategy which ensures successful swing-up and stabilization of both the classical and rotary single inverted pendulum system was presented in [10].

A. *Automatic Linear Approximation of the Rotary Inverted Pendulum System*

After the rearrangement of the derived motion equations into the minimal ODE form (1), it is possible to express the rotary single inverted pendulum model using the canonic form of the standard nonlinear state-space description:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}\left(\boldsymbol{x}(t), u(t), t\right) \\ \boldsymbol{y}(t) &= \boldsymbol{g}\left(\boldsymbol{x}(t), u(t), t\right) \end{aligned}, \qquad (4)$$

by defining the state vector as $\boldsymbol{x}(t) = \left(\boldsymbol{\theta}(t) \;\; \dot{\boldsymbol{\theta}}(t)\right)$ and isolating the second derivative $\ddot{\boldsymbol{\theta}}(t)$ from (1).

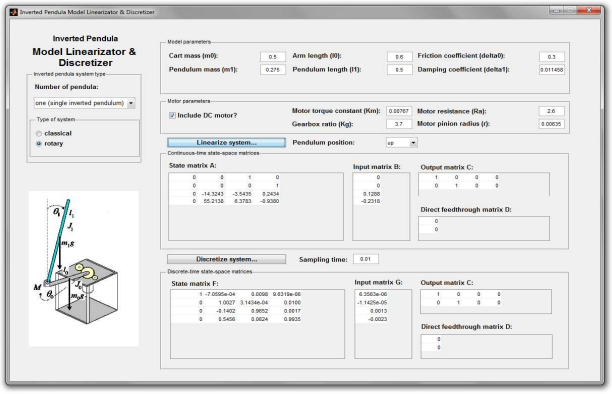

Fig. 7 Obtaining the state-space matrices of the rotary single inverted pendulum via the *Inverted Pendula Model Linearizator & Discretizer*.

The general procedure implemented by the *Inverted Pendula Model Equation Derivator* uses an assumption which defines the "all-upright" equilibrium as $\boldsymbol{x}(t) = \boldsymbol{x}_S = \boldsymbol{0}^T$. If the input $u(t) = u_S = 0$, then the continuous-time state-space description of the linearized inverted pendulum system has the form:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + du(t) \end{aligned}. \qquad (5)$$

In this paper, the $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{C}$, d matrices which make up the linearized state-space model (5) were obtained from the *Inverted Pendula Model Linearizator & Discretizer*, a MATLAB GUI application which generates the matrices in (5) by expanding (4) into the Taylor series around a given equilibrium point with the higher-order terms neglected (Fig. 7), and also returns the matrices of the discretized state-space model

$$\begin{aligned} \boldsymbol{x}(i+1) &= \boldsymbol{F}\boldsymbol{x}(i) + \boldsymbol{g}u(i) \\ \boldsymbol{y}(i) &= \boldsymbol{C}\boldsymbol{x}(i) + du(i) \end{aligned}. \qquad (6)$$

if the sampling period constant has been provided.

Using the numeric parameters from the open-loop simulations in section II, the following continuous-time state-space matrices of the linearized rotary single inverted pendulum voltage model were generated:

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -14{,}3243 & -3{,}5435 & 0{,}2434 \\ 0 & 55{,}2138 & 6{,}3783 & -0{,}938 \end{pmatrix} \quad \boldsymbol{b} = \begin{pmatrix} 0 \\ 0 \\ 0{,}1288 \\ -0{,}2318 \end{pmatrix}$$

The system eigenvalues were computed to be $\begin{pmatrix} 0 & 6{,}404 & -9{,}096 & -1{,}79 \end{pmatrix}$, which implies that the linear approximation of rotary single inverted pendulum in the upper equilibrium is an unstable system with first-degree astatism. The system was next discretized with the sampling period of $T_s = 0{,}01s$ and the following discrete-time state-space matrices were obtained:

$$\boldsymbol{F} = \begin{pmatrix} 1 & -0{,}0007 & 0{,}0098 & 0 \\ 0 & 1{,}0027 & 0{,}0004 & 0{,}01 \\ 0 & -0{,}1402 & 0{,}9652 & 0{,}0017 \\ 0 & 0{,}5456 & 0{,}0624 & 0{,}9935 \end{pmatrix} \quad \boldsymbol{g} = \begin{pmatrix} 0{,}000006 \\ -0{,}000011 \\ 0{,}0013 \\ -0{,}0023 \end{pmatrix}$$

B. *State-Feedback Control with a State Estimator*

The *Inverted Pendula Control* sublibrary of the *IPMaC* provides complex software support for the *linear state-feedback controller design*. It contains several dynamic-masked function blocks which were described in [2] in terms of their structure and user interface (Fig. 8).

The *State-Feedback Controller with Feedforward Gain* (*SFCFG*) library block implements the standard state-feedback control law which is calculated either from the continuous-time state-space description:

$$u(t) = u_R(t) + u_{ff}(t) + d_u(t) = -\boldsymbol{k}\boldsymbol{x}(t) + k_{ff}\,w(t) + d_u(t) \qquad (7)$$

or from the discrete-time linear state-space description:

$$u(i) = u_R(i) + u_{ff}(i) + d_u(i) = -\boldsymbol{k}_D x(i) + k_{ffD} w(i) + d_u(i) \qquad (8)$$

where $\boldsymbol{k}$ / $\boldsymbol{k}_D$ is the *feedback gain* which brings the state vector $\boldsymbol{x}(t)$ / $\boldsymbol{x}(i)$ into the origin of the state space, k_{ff} / k_{ffD} is the *feedforward (setpoint) gain* which makes the output track the reference command and $d_u(t)$ / $d_u(i)$ is the unmeasured disturbance input [1] [2][3] [4]. To match an additional control objective (initial deflection of the pendula, compensation of disturbance signal, tracking a reference position of the arm or a combination of the three), the block's appearance may be adjusted by optional enabling or disabling of the nonzero setpoint input $w(t)$ / $w(i)$ and the disturbance input $d_u(t)$ / $d_u(i)$.

The method to determine the feedback gain $\boldsymbol{k}$ / $\boldsymbol{k}_D$ can be selected from between the pole-placement algorithm and the linear quadratic (*LQ*) optimal control method [1][11].

Due to measurement limitations, it is often impossible to retrieve the whole state-space vector at every time instant. The *Luenberger Estimator* (*LE*) block provides the controller block with a complete, reconstructed state vector by evaluating a model of the original discrete-time system in the structure:

$$\hat{\boldsymbol{x}}(i+1) = \boldsymbol{F}\hat{\boldsymbol{x}}(i) + \boldsymbol{g}u(i) + \boldsymbol{L}\left(\boldsymbol{y}(i) - \boldsymbol{C}\boldsymbol{x}(i)\right) \qquad (9)$$

where $\boldsymbol{L}$ is the estimator gain matrix, $\hat{\boldsymbol{x}}(i)$ is the reconstructed state vector and the estimation error $\tilde{\boldsymbol{x}}(i) = \boldsymbol{x}(i) - \hat{\boldsymbol{x}}(i)$ is being minimized.

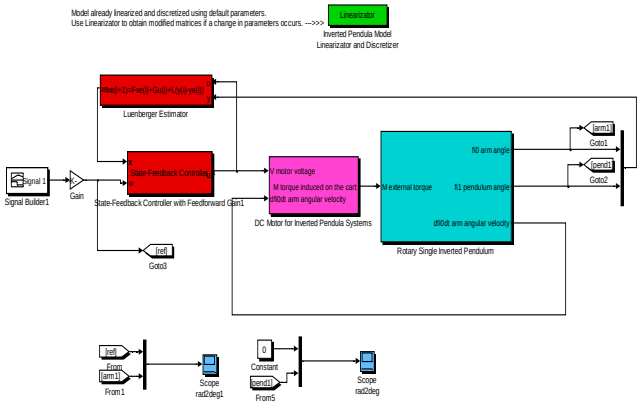

Fig. 8 State-feedback control of the rotary single inverted pendulum (voltage model) – general simulation setup.

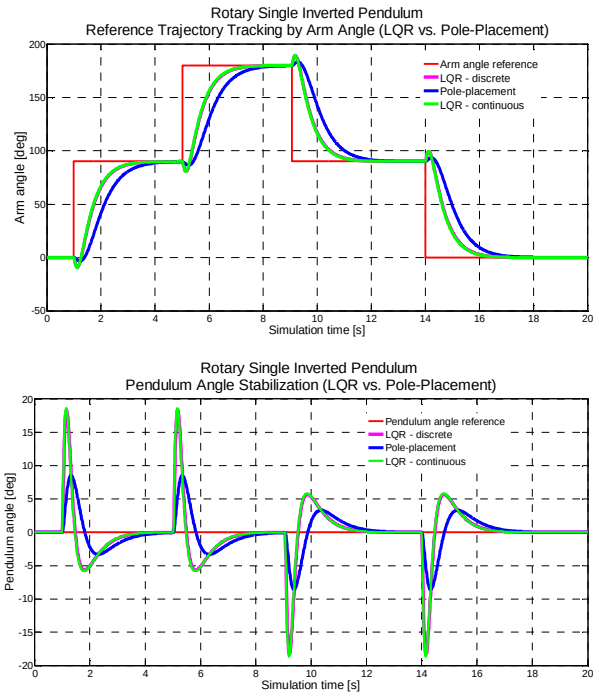

Fig. 9 Rotary single inverted pendulum – simulation results for LQR control with feedforward gain, compared to pole-placement.

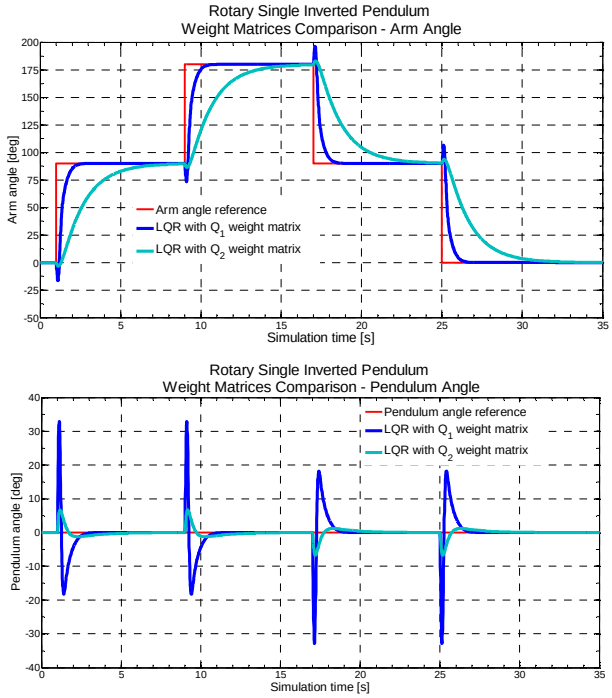

Fig. 10 Rotary single inverted pendulum - evaluation of the influence of the Q weight matrix on the performance of the system.

Fig. 9 illustrates the results of applying continuous-time and discrete-time LQR state-feedback to the rotary single inverted pendulum voltage model; *RSIP*, *SFCFG* and *LE* blocks were employed in the corresponding simulation scheme. The control objective was to make the arm rotate for a total of half a circle and stop every quarter-turn to stabilize before returning to its initial position; the pendulum had to be kept upright all the time. The weight matrices of both the standard continuous-time LQ functional:

$$J_{LQR}(t)=\frac{1}{2}\int_0^{\infty}\left(\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t)+u_R^T(t)ru_R(t)\right)dt \qquad (10)$$

and the discrete-time LQ functional:

$$J_{LQR}(i)=\sum_{i=0}^{N-1}\boldsymbol{x}^T(i)\boldsymbol{Q}\boldsymbol{x}(i)+u_R^T(i)ru_R(i) \qquad (11)$$

were set to $\boldsymbol{Q}=diag(100 \quad 20 \quad 20 \quad 0)$, $r=1$.

Moreover, it was assumed that the arm and pendulum angles would be directly measurable while the velocities would need to be estimated; the vector of estimator poles was set to $(0.1 \quad 0.2+0.1i \quad 0.2-0.1i \quad 0.3)$.

It is obvious that the response of the discrete LQR algorithm closely follows that of the continuous algorithm, and there is no steady-state error in either case. The simulation results were compared with those of continuous pole-placement control: the vector of desired closed-loop poles was set to $(-2 \quad -3+i \quad -3-i \quad -10)$.

To enhance the performance of the LQR controller, optimal weight matrices were sought. After setting $r=1$, the positive real diagonal elements of the $\boldsymbol{Q}$ weight matrix were modified and the influence of each candidate matrix on the overall performance of the system was verified in the discrete-time LQR simulation setup for the rotary inverted pendulum voltage model.

The results of two extreme cases of selecting $\boldsymbol{Q}$: $\boldsymbol{Q}_1=diag(500 \quad 0 \quad 20 \quad 0)$, $\boldsymbol{Q}_2=diag(10 \quad 120 \quad 20 \quad 0)$ are depicted in Fig. 10. For $\boldsymbol{Q}_1$, the arm is fast to reach the reference position but the pendulum overshoot is far from desired. If $\boldsymbol{Q}_2$ is selected, the pendulum swings are well within the limits, however the arm's rise time is unacceptably slow. It has therefore been concluded that the optimal time behavior of the arm and the pendulum are two conflicting requirements which cannot be satisfied at once. Any successful tuning of weight matrices must therefore result in a reasonable compromise between the quick rise time of the base and low overshoot of the pendulum.

C. State-Feedback Control with Permanent Disturbance Compensation

Applying state-feedback control with feedforward gain on a system is insufficient if the steady-state effect of a permanent disturbance input needs to be compensated. The structure of the *State-Feedback Controller with Summator (SFCS)* block implements a summator term $v(i)$ which sums up all past error values [12]. This ensures that the system output will track the changes in the reference command and eliminate the influence of permanent disturbances. The evaluated control law is specified as

$$u(i)=u_R(i)+u_{ff}(i)+u_S(i)=-\boldsymbol{k}_1\boldsymbol{x}(i)+k_2w(i)+v(i) \qquad (12)$$

where $w(i)$ is the reference command and $\boldsymbol{k}_1$, k_2 are gain matrices which are computed from a matrix structure derived in and implemented into the *SFCS* block.

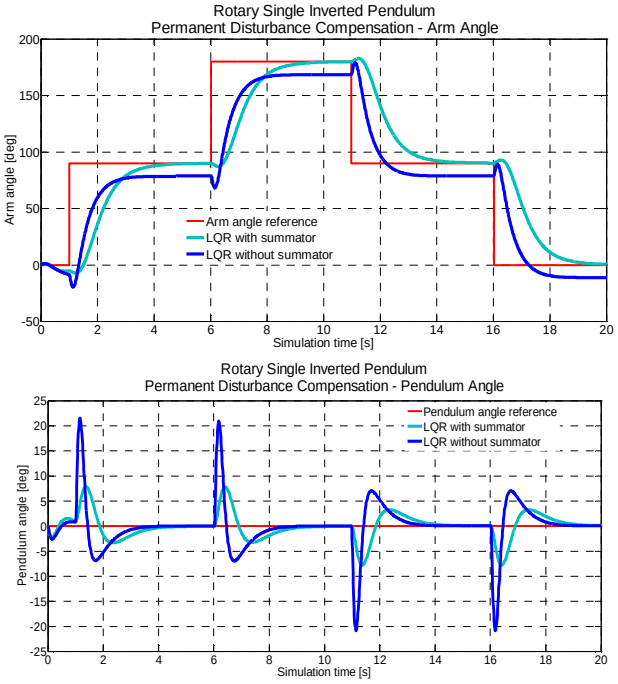

Fig. 11 Rotary single inverted pendulum – simulation results for LQR control with a summator.

The control law (12) was verified for the rotary single inverted pendulum voltage model and the weight matrices of the standard discrete-time LQ functional (11) were set to $\boldsymbol{Q} = diag(500 \quad 0 \quad 20 \quad 0)$, $r = 1$. The results were compared to those of a conventional state-feedback controller using the same values of weight matrices. A constant disturbance input (5V for the first controller, 1V for the second) was present in both simulations. To compensate for measurement limitations, the *LE* block was included in both schemes, with the estimator poles set to the same values as in section III/B.

As it can be seen in Fig. 11, the conventional LQR controller fails to track the reference trajectory without producing steady-state error, but the permanent disturbances are successfully compensated by a LQR algorithm with a summator included in the control structure.

IV. Conclusion

The purpose of this paper was to present a comprehensive approach to the modeling and control of the rotary single inverted pendulum system. *Inverted Pendula Modeling and Control*, a custom-designed *Simulink* block library developed by the authors of the paper, was used as a software framework for all covered issues which included model derivation and open-loop analysis, linearization and state-feedback control algorithm design.

The library provided suitable function blocks to support every step of the process (e.g. a pre-prepared simulation model of the rotary single inverted pendulum system or a detailed state-feedback controller block) as well as several innovative applications with graphical user interface. One of these was used to derive the mathematical model for the selected inverted pendulum system in form of symbolic equations of motion, and the other performed the automatic linear transformation of the system in a selected equilibrium point. In these applications, great practical potential of the symbolic mathematical software was demonstrated.

The *IPMaC* block library enhances the capabilities of the *MATLAB/Simulink* program environment by providing means for modeling and control of an important class of nonlinear mechanical systems. It can therefore be considered as a meaningful contribution to modeling and control education at technical universities.

Acknowledgment

This work has been supported by the Scientific Grant Agency of Slovak Republic under project Vega No.1/0286/11 Dynamic Hybrid Architectures of the Multiagent Network Control Systems.

References

[1] S. Jadlovská, *Modeling and Optimal Control of Inverted Pendula Systems*, Diploma (Master) Thesis. Supervisor: prof. Ing. Ján Sarnovský, CSc.. Košice: Technical University of Košice, Faculty of Electrical Engineering a Informatics, 2011.

[2] S. Jadlovská, J. Sarnovský, "An extended Simulink library for modeling and control of inverted pendula systems," *Proc. of the Int. Conf. Technical Computing Prague 2011*, November 8, 2011, Prague, Czech Republic, ISBN 978-80-7080-794-1

[3] S. Jadlovská, A. Jadlovská, "Inverted pendula simulation and modeling–a generalized approach," *Proc. of the 9th Int. Scientific–Technical Conf. on Process Control*, June 7-10, 2010, University of Pardubice, Czech Republic, ISBN 978-80-7399-951-3

[4] S. Jadlovská, J. Sarnovský, "Classical double inverted pendulum – a complex overview of a system," *Proc. of the IEEE 10th Int. Symposium on Applied Machine Intelligence and Informatics – SAMI 2012*, January 26-28, 2012, Herľany, Slovakia, ISBN 978-1-4577-0195-5

[5] K. Furuta, M. Yamakita, S. Kobayashi: "Swing Up Control of Inverted Pendulum", *Proc. of the Int. Conf. on Industrial Electronics, Control and Instrumentation (IECON'91)*, Oct 28-Nov 1, 1991, Kobe, Japan

[6] P. Ernest, P. Horáček, "Algorithms for control of a rotating pendulum", *Proc. of the 19th IEEE Mediterranean Conf. on Control and Automation (MED '11)*, Corfu, Greece, 2011

[7] Kats, C.J.A. (2004): Nonlinear control of a Furuta rotary inverted pendulum, DCT Report, No. 2004:69

[8] REX Controls s r.o.: *Rotary Inverted Pendulum FPM-210/211. User Manual* [Rotační inverzní kyvadlo FPM-210/211. Užívatelská příručka] 2010

[9] Quanser Academic, Rotary Motion Servo Plant: SRV02, User Manual, No. 700, Rev. 2.3

[10] S. Jadlovská, J. Sarnovský, "Swing-up and stabilizing control of classical and rotary inverted pendulum systems," *Proc. of the 12th Scientific Conference of Young Researchers (SCYR 2012)*, in press.

[11] J. Sarnovský, A. Jadlovská, P. Kica, *Optimal and Adaptive Systems Theory* [Teória optimálnych a adaptívnych systémov], Košice: ELFA s r.o., 2005, ISBN 80-8086-020-3.

[12] O. Modrlák, *Automatic Control Theory, vol. 1 – Basic State-Space Analysis and Synthesis (Study Material)* [Teorie automatického řízení I. - Základy analýzy a syntézy ve stavovém prostoru (Studijní materiály)], Liberec: Technical University of Liberec, 2002.