

AN EXTENDED SIMULINK LIBRARY FOR MODELING AND CONTROL OF INVERTED PENDULA SYSTEMS

Slávka Jadlovská, Ján Sarnovský

Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovakia

Abstract

The aim of this paper is to present an improved and extended version of the *Inverted Pendula Modeling and Control*, a structured thematic Simulink block library which enhances the capabilities of the MATLAB/Simulink program environment by providing means for modeling and control of classical and rotary inverted pendula systems in form of function blocks, demo simulation schemes and applications with graphical user interface.

1 Introduction

Inverted pendula systems represent a significant class of highly nonlinear mechanical systems used in classical control education. Although underactuated and extremely unstable by nature, the systems are controllable and have a number of practical applications: problems such as balancing a broomstick on a handpalm, stabilization of a walking human or robot, control of a launching rocket or the vertical movement of a human shoulder or arm can all be simulated by some type of an inverted pendula system [1][2][3][4][5]. The diversity of *modeled systems* is reflected in the wide variety of available *inverted pendula models*. These may be classified according to several criteria:

- *the type of base which actuates the pendula* – the system base may either be moving in a linear manner (*classical* or *linear* inverted pendulum system), or in the rotary manner in a horizontal plane (*rotary* inverted pendulum system);
- *the number of pendulum links attached to the mechanism* – every extra pendulum link increases the number of the system's degrees of freedom, making control in turn more challenging. For inverted pendula systems, controllability has been shown for up to 4 links attached to the base [6].
- *the distribution of mass along the pendulum rod* – the pendulum links can either be homogenous rods with the mass concentrated in their center of gravity, or the rod may be considered to be massless, with the mass concentrated in the load at the end.

Fig. 1 shows the schematic depictions of three characteristic representatives of inverted pendula systems which are most frequently used as testbed systems for nonlinear control design [1][2]: the classical single (one-link), classical double (two-link) and rotary single inverted pendulum system.

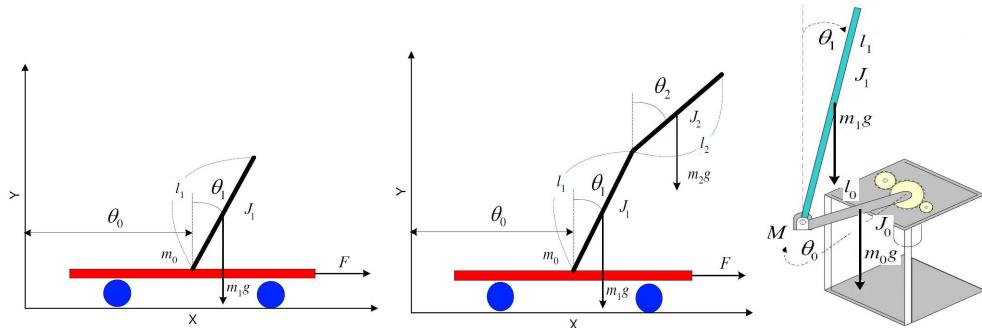


Fig. 1 Typical representatives of inverted pendula systems: a) classical single inverted pendulum system, b) classical double inverted pendulum system, c) rotary single inverted pendulum system

When the prominent position that inverted pendula systems occupy in nonlinear control theory was considered by the authors of this paper, the idea of providing integrated program support for the whole class of systems immediately arose. As a result, a structured thematic Simulink block library, *Inverted Pendula Modeling and Control (IPMaC)*, has been developed since 2009 as a comprehensive software framework for the problems of analysis and control of inverted pendula systems. While the

first presented version of the block library [3] was limited to classical inverted pendula systems, this paper describes an improved and extended 2011 version of the library which deals with two principal categories of inverted pendula systems (classical and rotary). The importance of the mutual analogy of the systems is emphasized: if a system of inverted pendula is approached as an instance of a generalized (n -link) system, crucial procedures related to system modeling and control algorithm design can be processed into an algorithm and implemented via symbolic software, represented in *MATLAB* by the *Symbolic Math Toolbox*. These considerations can be further extended to control algorithm design.

2 The Installation and Structure of the *IPMaC* Block Library

The *IPMaC* block library was designed in *MATLAB/Simulink* and it is to be used exclusively within this program environment. The core of the block library was developed under the configuration of OS Windows Vista™, MATLAB 7.6 (Release R2008a). After compatibility testing, extensions to the library were made under OS Windows 7™ Enterprise, MATLAB 7.9 (Release R2009b) as well as MATLAB 7.12 (R2011a).

The installation process of the *IPMaC* block library consists of unzipping the provided *InvPend.zip* package into a desired directory and calling the included installation script *slblocks.m*. Once installed, the library becomes an integral part of the *Simulink Library Browser*, and the *IPMaC* function blocks are fully compatible with the blocks from the rest of *Simulink* built-in libraries.

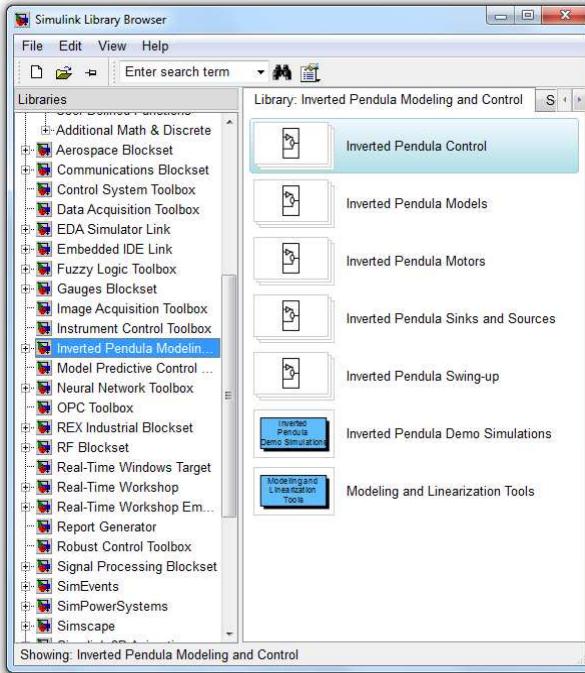


Fig. 2 The *IPMaC*, installed and active in the *Simulink Library Browser*

Like in any other *Simulink* blockset, the *IPMaC* function blocks are thematically arranged into sublibraries which open when the corresponding item of the *IPMaC* tree structure in the *Simulink Library Browser* is singleclicked. The current version of the *IPMaC* has kept the core of the sublibrary structure from the original 2009 version: the **Inverted Pendula Models** sublibrary contains simulation models of the selected representatives of inverted pendula systems and the **Inverted Pendula Control** sublibrary is made up of function blocks that implement control algorithms. Additional sublibraries have been designed to incorporate the actuating mechanism block; useful sink and source blocks; or a heuristic swing-up controller. The sublibraries that the *IPMaC* is currently structured into are:

- **Inverted Pendula Control** sublibrary – contains dynamic-masked function blocks which encapsulate state-feedback control algorithms
- **Inverted Pendula Models** sublibrary – contains dynamic-masked simulation models of selected classical and rotary inverted pendula systems, pre-prepared for use in open-loop analysis as well as in state-feedback control design

- **Inverted Pendula Motors** sublibrary – contains a block which implements the analytical model of the most frequently employed actuating mechanism (brushed direct-current motor)
- **Inverted Pendula Sinks and Sources** sublibrary – concentrates all sink and source blocks which may find their use in pendula modeling and control, such as an input signal constrained in time and amplitude or a sink block that displays a signal in degrees rather than radians
- **Inverted Pendula Swing-up** sublibrary – contains a heuristic controller which swings the pendulum from the downward to the upright equilibrium and switches to stabilizing control in the moment the pendulum is sufficiently close to the upright position.

In addition to sublibraries, the root directory of the *IPMaC* includes two blocks of a special type which open up into a collection of links that lead to separate *MATLAB/Simulink* tools:

- **Demo Simulations** - opens up into a tree structure of links to simulation schemes which illustrate the ways of interconnecting the *IPMaC* blocks to solve a variety of problems
- **Modeling and Linearization Tools** – opens up into a set of links to a pair of applications which provide a comfortable, user-friendly graphical interface to modeling and linearization.

3 Automatic Approach to Inverted Pendula Mathematical Modeling

The automatic approach to motion equation derivation was promoted for a number of reasons: it yields a particularly precise approximation of the real system's dynamics and eliminates any factual or numeric errors which could arise during manual mathematical modeling. To enable the automatic derivation of motion equations for inverted pendula systems, a general procedure has to be available which will output the equations for any given number of pendula [1][3][4]. Analysis of generalized classical and rotary inverted pendula systems therefore needs to be dealt with.

The generalized *system of classical inverted pendula* is defined as a set of $n > 1$ rigid, homogenous rods (*pendulum links*) which are interconnected in joints and attached to a cart (i.e. a stable mechanism which allows for movement alongside a single axis). Analogically, the generalized *system of rotary inverted pendula* is composed of a set of $n > 1$ interconnected homogenous rods mounted on a rigid arm which rotates in a horizontal plane, perpendicular to the pendula. Every system of n inverted pendula is a multi-body composed of the base and the individual pendula, and thus has $n + 1$ degrees of freedom. It is a typical example of an underactuated system since the number of actuators is lower than the number of system links: the only input (the force $F(t)$ acting upon the cart or the torque $M(t)$ applied on the rotary arm) is used to control the $n + 1$ outputs of the system: cart position [m] or arm angle [rad], and pendula angles [rad].

3.1 Derivation of Motion Equations

The *Inverted Pendula Model Equation Derivator* is a MATLAB GUI application which generates the motion equations for a user-chosen type of inverted pendula system (classical/rotary, single/double). Fig. 3 shows an example preview of the *Derivator* window which contains the generated model equations for the rotary single inverted pendulum system. The whole derivation process can be tracked in the command window (see the preview in [3]).

The core of the *Derivator* tool is represented by MATLAB functions that use the *Symbolic Math Toolbox* to implement general procedures that derive the motion equations for a classical or rotary inverted pendula system. If we view the system's degrees of freedom as the following vector of generalized coordinates [2][7]:

$$\theta(t) = (\theta_0(t) \quad \theta_1(t) \quad \dots \quad \theta_n(t))^T \quad (1)$$

then the system can be mathematically described by the *Euler-Lagrange equations of second kind* (one for each generalized coordinate). The vector form of the equations is:

$$\frac{d}{dt} \left(\frac{\partial L(t)}{\partial \dot{\theta}(t)} \right) - \frac{\partial L(t)}{\partial \theta(t)} + \frac{\partial D(t)}{\partial \dot{\theta}(t)} = Q^*(t) \quad (2)$$

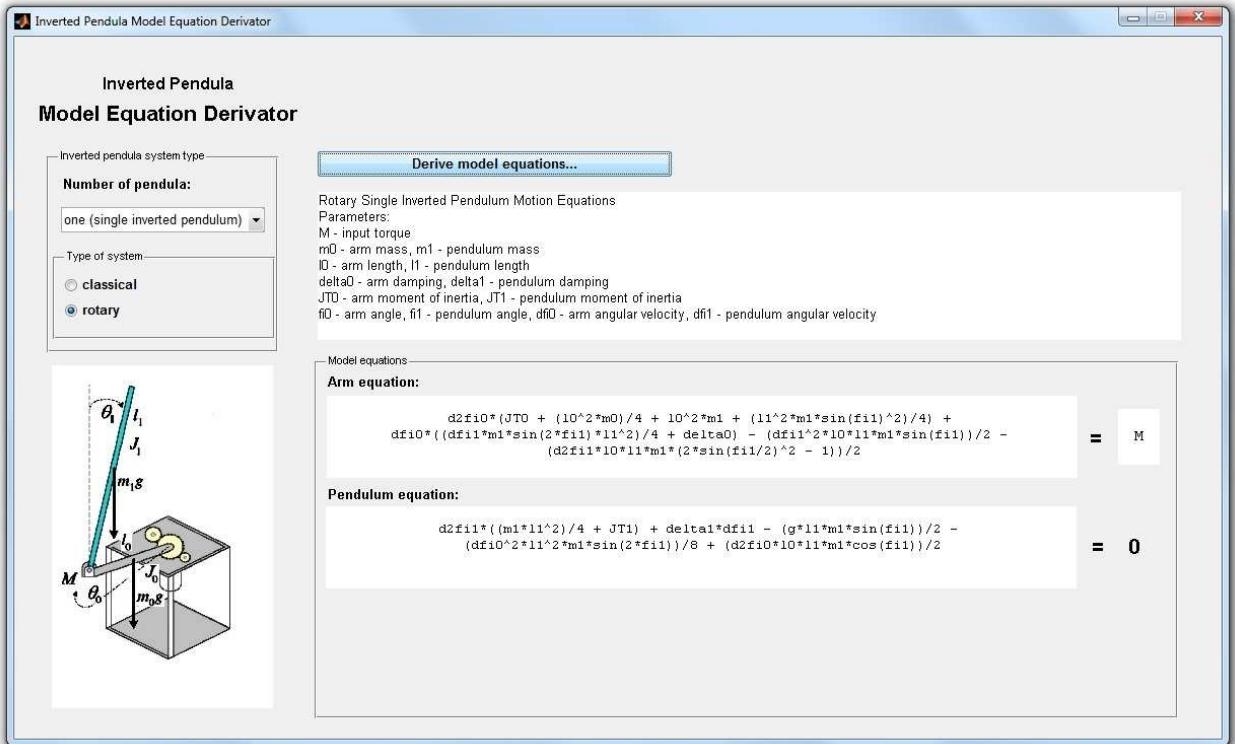


Fig. 3 Inverted Pendula Model Equation Derivator with the derived rotary single inverted pendulum model equations

where $L(t)$ (*Lagrange function*) is defined as the difference between the system's kinetic and potential energy, $D(t)$ (*Rayleigh, dissipation function*) describes the viscous (friction) forces and $\mathbf{Q}^*(t)$ is the vector of *generalized external forces* acting upon the system. The process of derivation of the motion equations to describe any kind of inverted pendula system has thus transformed into the determination of kinetic, potential and dissipation energies related to the base and all pendula:

$${}^nE_K(t) = \sum_{i=0}^n E_{Ki}(t), \quad {}^nE_P(t) = \sum_{i=0}^n E_{Pi}(t), \quad D(t) = \sum_{i=0}^n D_i(t) \quad (3)$$

Using well-known physical formulae and the necessary theoretical assumptions, general relations that describe the energetic balances of the base and i -th pendulum in a system of n inverted pendula were derived and can be found in [1] for both classical and rotary inverted pendula systems. Procedures which generate the equations of motion for any given system type and number of pendula were subsequently constructed and implemented as symbolic MATLAB functions (*invpenderiv.m* for classical, *rotinvpenderiv.m* for rotary) which produce the equations in the simplified and rearranged form, equivalent to the most likely form obtained by manual derivation.

The mathematical models generated by the *Derivator* will hereafter be referred to as *force/torque models*, to distinguish them from *voltage models*, which will be derived in the following section.

3.2 Simulation Models of Classical and Rotary Inverted Pendulum Systems

The *Inverted Pendula Models* sublibrary of the *IPMaC* block library contains library blocks which represent the simulation models of a classical single, classical double and rotary single inverted pendulum system: the *Classical Single Inverted Pendulum*, the *Classical Double Inverted Pendulum* and the *Rotary Single Inverted Pendulum* block. The structure of all blocks is composed of logically designed subsystem blocks which are interconnected with respect to their mutual physical relations (Fig. 4); each subsystem block implements a nonlinear differential equation that is part of the system's Lagrange mathematical model, obtained from the *Inverted Pendula Model Equation Derivator* application. Every model is composed of $n+1$ (2 for single, 3 for double) nonlinear second-order differential equations that describe the base and each pendulum. The motion equations that make up the respective mathematical models are specified in [1][2][3].

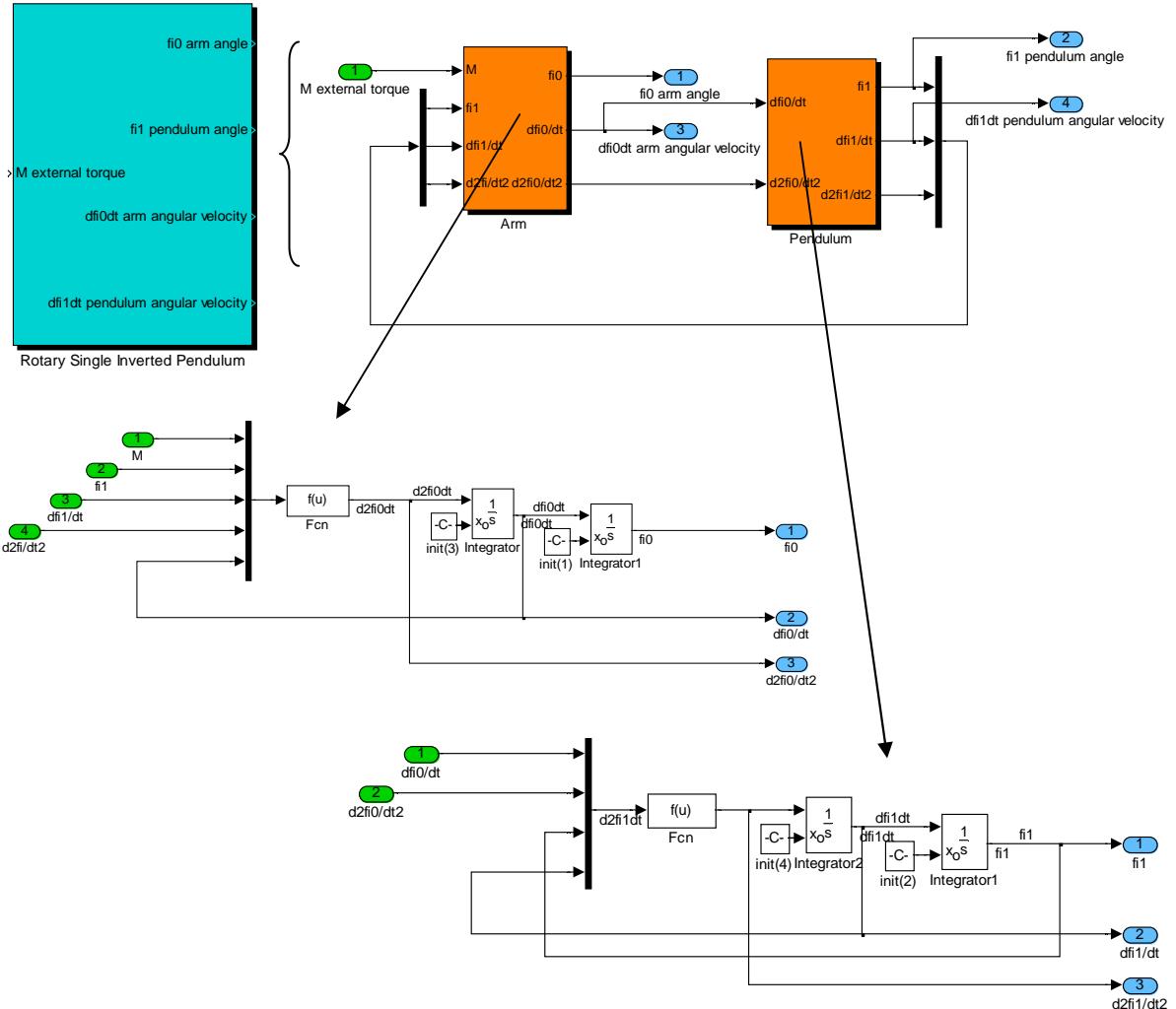


Fig. 4 The structure of the Rotary Single Inverted Pendulum library block

Each included system is equipped with a dynamic block mask (*Edit – Mask Subsystem*) which enables the user to edit the physical parameters and initial conditions, to enable or disable the input (force/torque) port and to flexibly adjust the number of the block's output ports, which is equivalent to fitting a real model with sensors. This was implemented by creating scripts which add the *Input/Output Port* block into the scheme whenever the input or one of the outputs is marked as required; and replace it by the *Ground/Terminator* block if this is not the case.

3.3 Simulation Models of Direct-Current Motors

The practical use of *force/torque models* generated by the *Derivator* and included in the *Inverted Pendula Models* sublibrary is limited purely to the simulational environment: we are unable to manually generate a force or torque which, if applied to the base, would stabilize the pendula in a chosen position. As a result, *electric motors* need to be coupled with inverted pendula systems to act as mechanisms which actually produce the force or torque which actuates both the base and the attached pendula.

Although there has recently been a rise in the percentage of brushless (EC, BLDC) direct-current motors, three-phase synchronous motors and asynchronous (induction) motors [1][7][10], the great majority of authors dealing with modeling and control of inverted pendula systems still use the brushed direct-current (DC) motor as a high-performance drive to actuate their system. Hence, it was decided that the *IPMaC/Inverted Pendula Motors* sublibrary would presently contain a single block: the **DC Motor for Inverted Pendula Systems**, which implements the mathematical model of a brushed direct-current (DC) motor in form of a voltage-to-force or a voltage-to-torque conversion relationship, derived in [1]. Appending the DC motor model to an inverted pendula system yields a *voltage model* of the system.

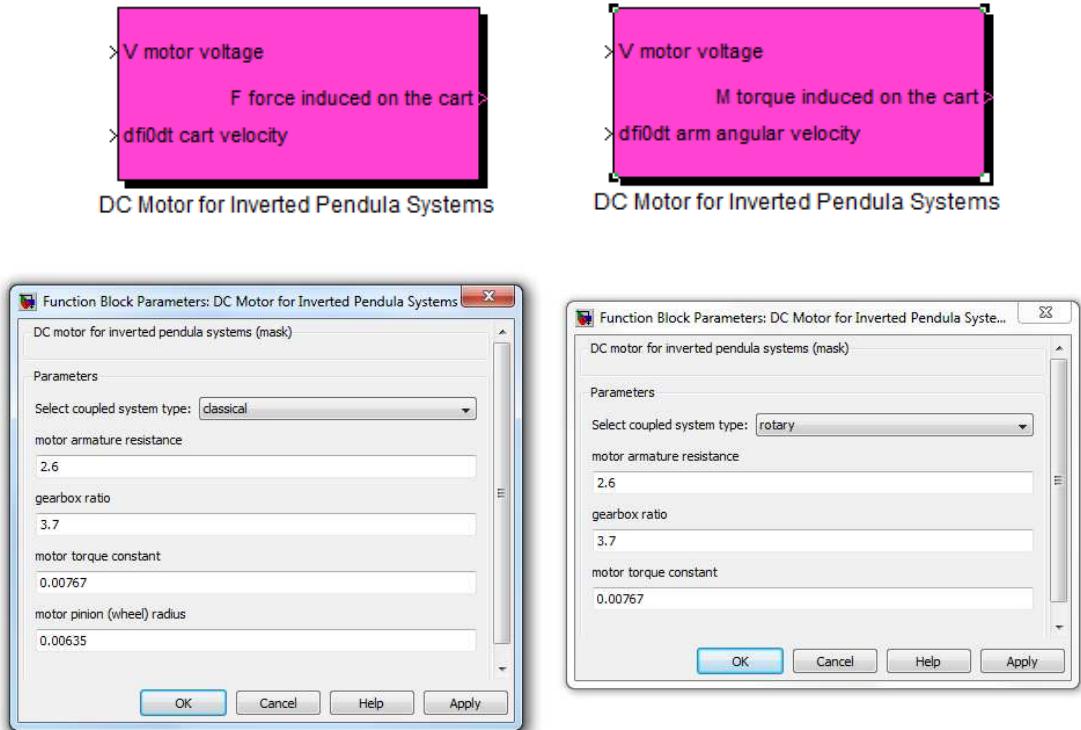


Fig. 5 The dynamically changing dialog box of the DC Motor for Inverted Pendula Systems block

The appearance of the block and the dynamic block mask can be adjusted to correspond with the type of inverted pendula system (classical/rotary) it is associated with (Fig. 5).

4 Implementation of State-Feedback Control Algorithms

It is well-known from experience that, once released and independently of the initial state, an inverted pendula system will always stabilize with all pendula in their natural hanging position, i.e. in their stable equilibrium. The principal control objective for all considered inverted pendula systems was therefore *to stabilize all pendulum links in the vertical upright (inverted) position*, which represents the unstable equilibrium of the system. Several additional problems (subobjectives) were approached and solved:

- stabilizing the system following the pendulum's initial deflection (nonzero initial conditions)
- compensation of a time-constrained (impulse) or permanent (step) disturbance input signal,
- tracking a desired position (reference trajectory) of the base.

Linear state-feedback control was emphasized as the principal technique for the control of the inverted pendula systems since control of several degrees of freedom at once can only be ensured if the whole state vector is taken into consideration.

4.1 Linear Approximation of Inverted Pendula Systems

To be able to employ linear control techniques, the linear approximation of the nonlinear inverted pendulum system needs to be obtained. The motion equations generated by the *Derivator* are first rewritten into the standard (minimal ODE – ordinary differential equation) form [8]:

$$M(\theta(t))\ddot{\theta}(t) + N(\theta(t), \dot{\theta}(t))\dot{\theta}(t) + P(\theta(t)) = V(t) \quad (4)$$

which provides the only way to express this kind of system in the nonlinear state-space form of

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= g(x(t), u(t), t) \end{aligned} \quad (5)$$

by defining the state vector as $x(t) = (\theta(t) \quad \dot{\theta}(t))^T$ and isolating the second derivative $\ddot{\theta}(t)$ from (4).

All inverted pendula systems included in the *IPMaC* were modeled in a way which defines the “all upright” equilibrium as $\mathbf{x}(t) = \mathbf{x}_s = \mathbf{0}^T$. If the input $u(t) = u_s = 0$, then the state-space description of the continuous linearized system is given as

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{d}u(t)\end{aligned}\quad (6)$$

and is obtained by expanding (5) into the Taylor series around a given equilibrium point and neglecting the terms of the order greater than 1.

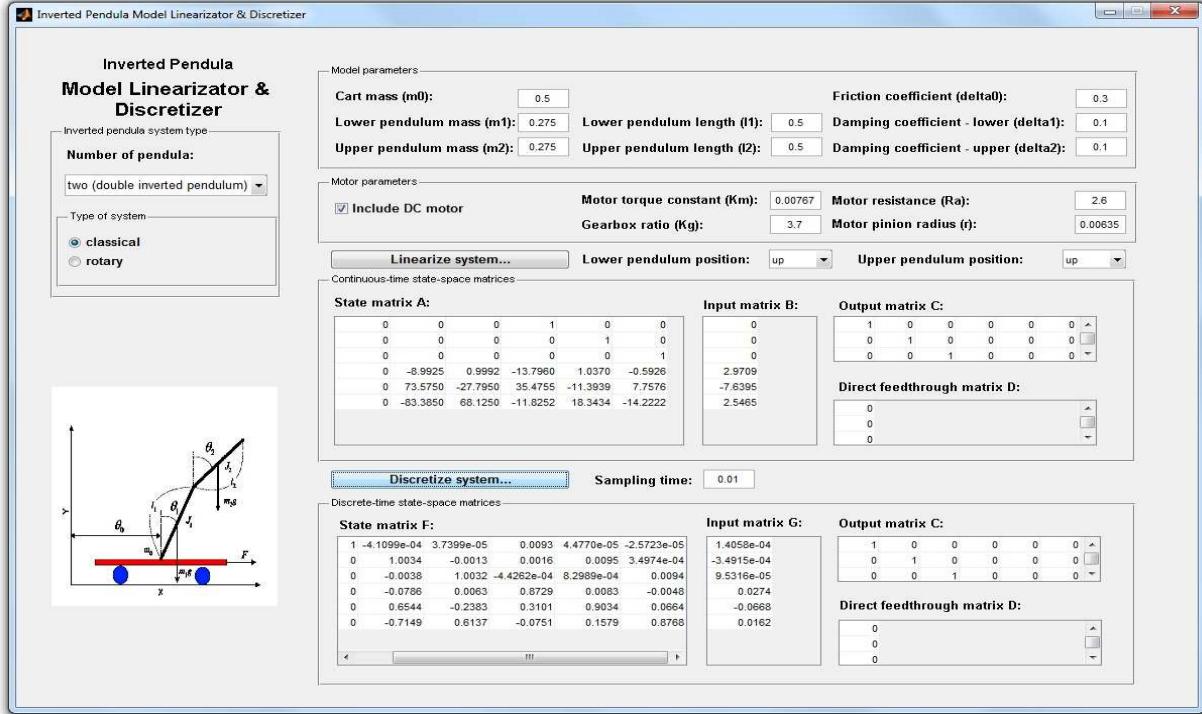


Fig. 6 Inverted Pendula Model Linearizer & Discretizer with the linearized and discretized state-space matrices of the classical double inverted pendulum system

The process of transformation of the Lagrange mathematical model of an inverted pendulum system into a linearized state-space matrix form can be considerably sped up with help of another GUI application designed as part of the *IPMaC*: the *Inverted Pendula Model Linearizer & Discretizer* (Fig. 6). In case the type of inverted pendula system (classical/rotary), model parameters and operating (equilibrium) point have been provided by the user, the application generates the numeric form of the \mathbf{A} , \mathbf{b} , \mathbf{C} , \mathbf{d} continuous state-space matrices (see (6)) of the system in a selected operating point. In case a discrete-time model is required, the *Linearizer & Discretizer* returns the discretized state-space matrices \mathbf{F} , \mathbf{g} , \mathbf{C} , \mathbf{d} of the chosen inverted pendula system provided the sampling period constant T_{vz} was entered and the continuous-time matrices have already been successfully computed.

4.2 Inverted Pendula Control

The expanded *Inverted Pendula Control* sublibrary of the *IPMaC* chiefly provides software support for the state-feedback methods of controller design for inverted pendula systems. Most importantly, the **State-Feedback Controller with Feedforward Gain** block implements the standard state-feedback control law calculated either from the continuous-time state-space description

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

or from the discrete-time linear state-space description

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

where \mathbf{k} (\mathbf{k}_D) is the *feedback gain* which brings the state vector $\mathbf{x}(t)$ ($\mathbf{x}(i)$) into the origin of the state space, k_{ff} is the *feedforward (setpoint) gain* which makes the output track the reference command and $d_u(t)$ ($d_u(i)$) is the unmeasured disturbance [1][2][3][9][10]. The preferred state-space description is

selected in the dynamic block mask (Fig. 7) which also allows the user to choose the method to determine the feedback gain \mathbf{k} (\mathbf{k}_D) from between the pole-placement algorithm and the linear quadratic regulation (*LQR*) optimal control method. To match a particular control subobjective, the block's appearance may be adjusted by optional enabling or disabling of the reference command input $w(t)$ ($w(i)$) and/or the disturbance input $d_u(t)$ ($d_u(i)$).

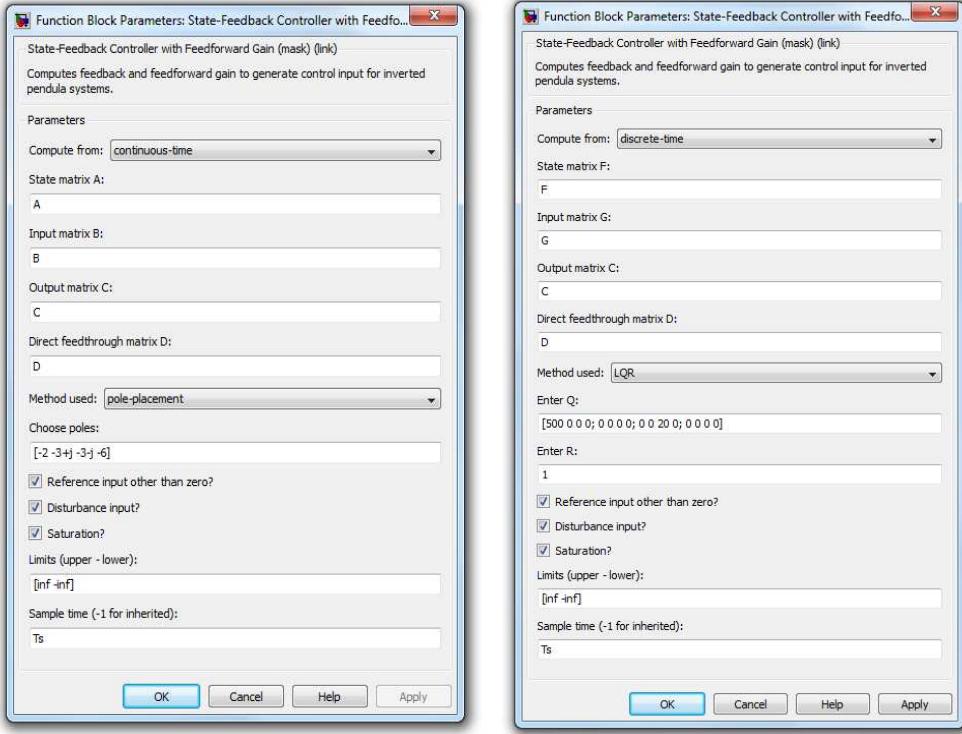


Fig. 7 The dialog box of the *State-Feedback Controller with Feedforward Gain* block, dynamically changing according to the chosen method

The structure of the *State-Feedback Controller with Summator* block implements a summator $v(i)$ of all past error values [11], which ensures that the system output will track the changes in the reference command and eliminate the influence of permanent disturbances. The control law is given as

$$u(i) = u_R(i) + u_{ff}(i) + u_S(i) = -k_1 \mathbf{x}(i) + k_2 w(i) + v(i), \quad (9)$$

The layout of the block mask once again allows the user to adjust the number of block input ports to match a control subobjective and to compute the state-feedback vector \mathbf{k}_1 using a preferred method and state-space description.

The *Luenberger Estimator* block provides a complete, reconstructed state vector by evaluating a model of the original discrete-time system in the structure:

$$\hat{\mathbf{x}}(i+1) = \mathbf{F}\hat{\mathbf{x}}(i) + \mathbf{g}u(i) + \mathbf{L}(\mathbf{y}(i) - \mathbf{C}\mathbf{x}(i)) \quad (10)$$

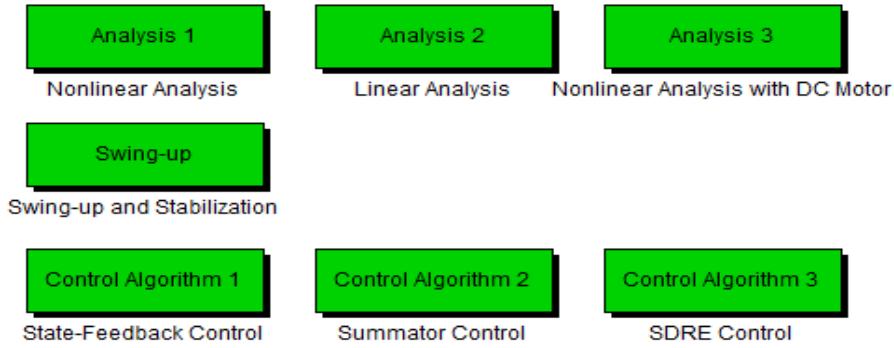
where \mathbf{L} is the estimator gain matrix and $\hat{\mathbf{x}}(i)$ is the reconstructed state vector.

5 Inverted Pendula Demo Simulations Section – Overview

The *Inverted Pendula Demo Simulations* section, accessible through the root directory of the *IPMaC*, displays a structured set of links to simulation schemes which demonstrate the functionality of the designed library blocks. Each demo scheme can be accessed by doubleclicking a block that briefly describes the issue it solves. Making a classic masked block to act as a link to another file involves creating an *OpenFcn* callback function within the *Block Properties*.

Compared to the *Demo Simulations* section of the 2009 version of the library, the collection of links has now been notably expanded to include a wider variety of addressed problems. The most significant changes to the *Demo Simulations* section of the *IPMaC* were the following:

Inverted Pendula Demo Simulations



Copyright 2009-2011 Ba. Slavka Jadlovska

Fig. 8 Structure of the root directory of the Demo Simulations section

- the rotary single inverted pendulum model was introduced into the library
- a DC motor block was implemented into a separate sublibrary; both *force/torque models* and *voltage models* of inverted pendula were used in simulations
- the control section was expanded to include more state-feedback controller blocks and control schemes; control algorithms were verified using every implemented simulation model
- the tree structure of the *Demo Simulations* section was completely revised (Fig. 8).

5.1 Open-loop Dynamic Analysis

Evaluatory open-loop responses to an impulse signal were performed for all analyzed nonlinear simulation models both without the motor (*force/torque model*) and with the motor (*voltage model*). The respective simulation schemes can be located in the *Nonlinear Analysis* and *Nonlinear Analysis with DC Motor* sections. Examples are shown in Fig. 9 (force model of a classical double inverted pendulum) and Fig. 10 (voltage model of a classical single inverted pendulum).

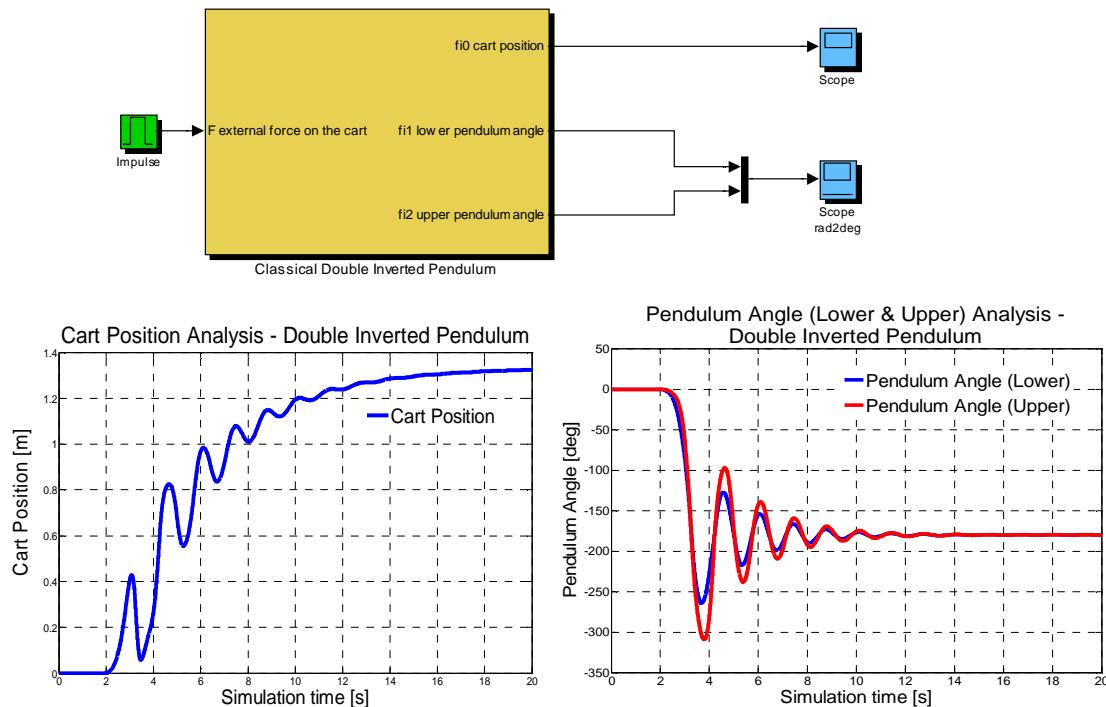


Fig. 9 Classical double inverted pendulum system – simulation scheme and time behavior results (cart position and pendula angles)

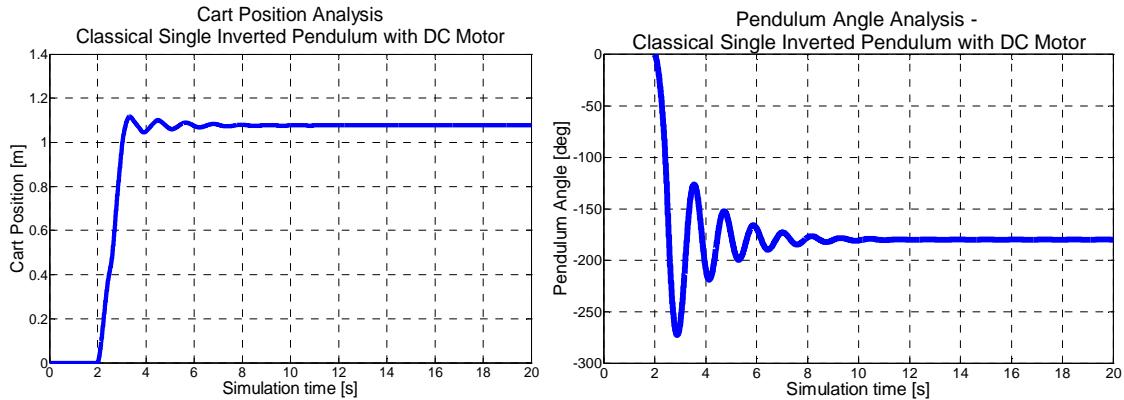
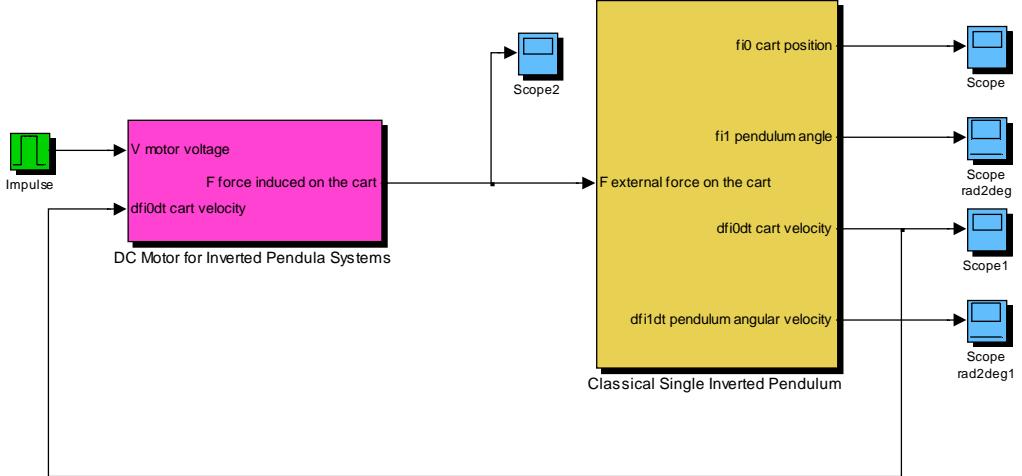


Fig. 10 Classical single inverted pendulum system with a DC motor – simulation scheme and time behavior results (cart position and pendulum angle)

Reasonable behavior of the open-loop responses of the simulation models (damped oscillatory transient state, system reaching the stable equilibrium point with all pendula pointing downward, visible backward impact of the pendulum/pendula on the base, motor-driven classical inverted pendulum response being much more rigid than that of its rotary equivalent) means the models can be considered accurate enough to serve as a reliable testbed for control algorithms.

5.2 Verifications of Inverted Pendula State-Feedback Control

The schemes in the *Demo Simulations* section document a number of simulation experiments which were performed to prove that the implemented controllers are able to meet the required control objectives for all available simulation models of inverted pendula systems.

Each demo simulation scheme is designed as an independent unit. All computations required for the simulation to run (model parameter definition, state-space matrices of the linear approximation necessary for control algorithm design) are programmed to occur during the initialization of the scheme, which eliminates the need for additional m-files. Moreover, if the parameters of the simulation model are changed by the user in an open scheme, the *Linearizator & Discretizer* tool can be called from a provided link to obtain modified state-space matrices (Fig. 11, Fig. 12).

The example schemes below were picked from the *State-Feedback Control* and *Summator Control* sections to illustrate the way of interconnecting the blocks to control the voltage model of a rotary single inverted pendulum system. Fig. 11 depicts the simulation results given that the control objective was to maintain the desired arm angle value while keeping the pendulum upright. An estimator block was included to simulate measurement limitations.

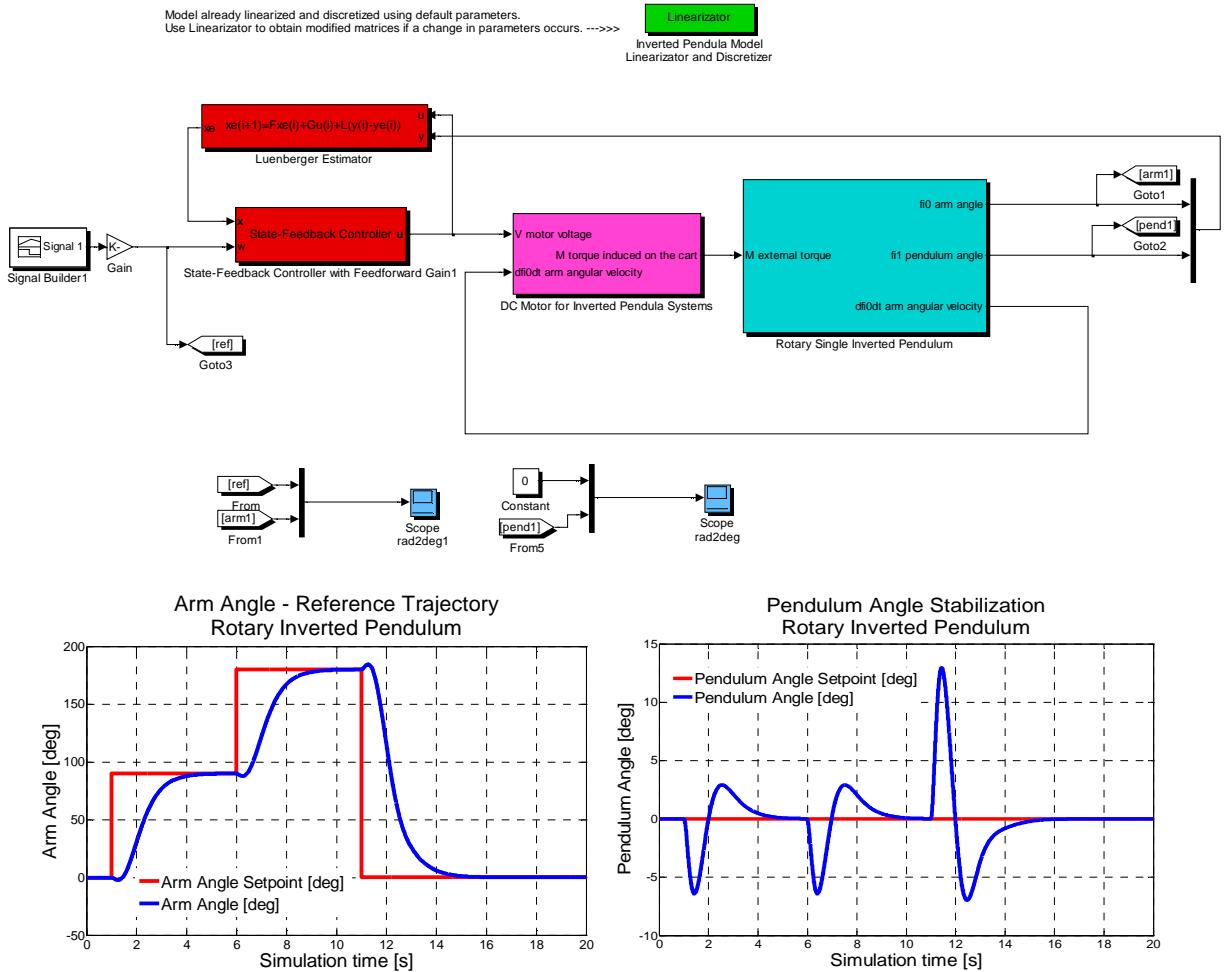
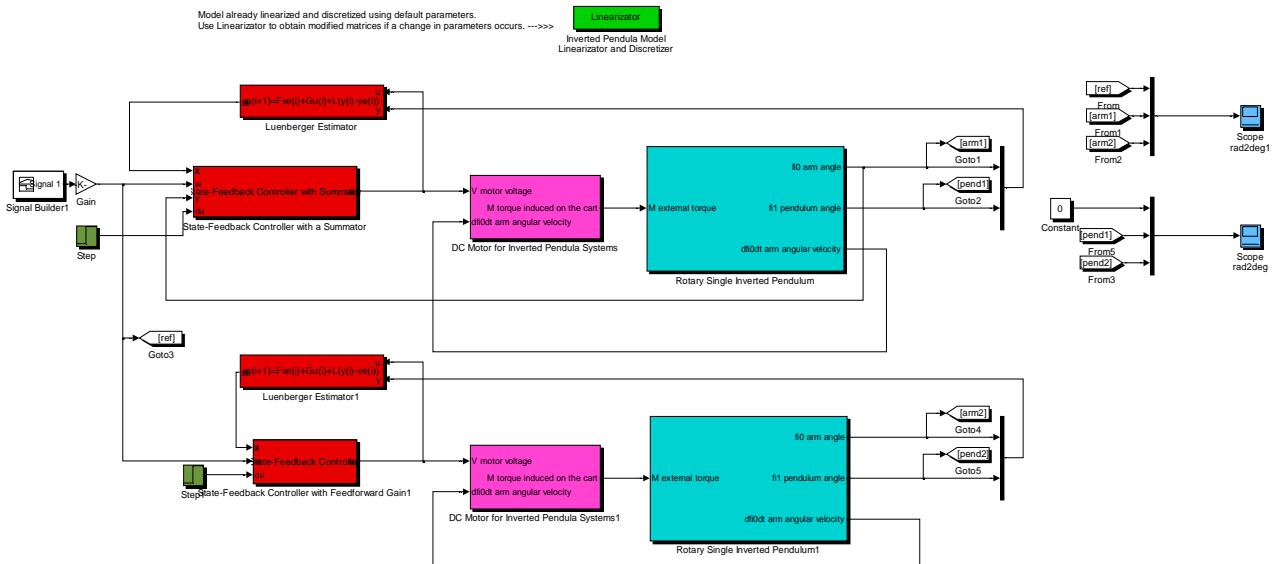


Fig. 11 Rotary single inverted pendulum: simulation results for LQR control with estimator included; note that the arm is supposed to rotate for exactly a half-circle before returning to its initial position

Finally, it is shown in Fig. 12 that applying state-feedback control with feedforward gain on a system is unsufficient if the steady-state effect of a permanent disturbance input needs to be eliminated. The compensation of permanent disturbances is successfully executed by a LQR algorithm with a summator included in the control structure.



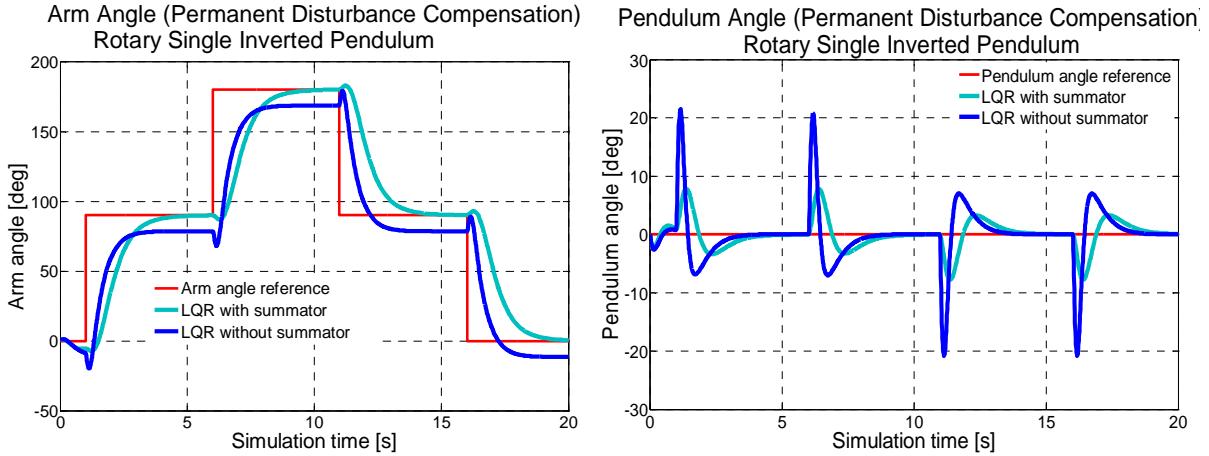


Fig. 12 Rotary single inverted pendulum – simulation results for LQR control with a summator (integral action)

5.3 Verifications of Inverted Pendula Swing-up

To illustrate the problem of pendulum swing-up, the *Swing-up Controller* block was included in the *IPMaC/Inverted Pendula Swing-up* sublibrary. The block implements three heuristic, energy-based control laws (cosine value controller, zero speed controller and absolute value controller) that swing the pendulum up from the downward to the upright equilibrium [12]. A transition mechanism switches between swing-up and stabilizing control provided the pendulum is sufficiently close to the upright position. The block mask allows the user to select a swing-up method, the input voltage magnitude and the size of the balancing region.

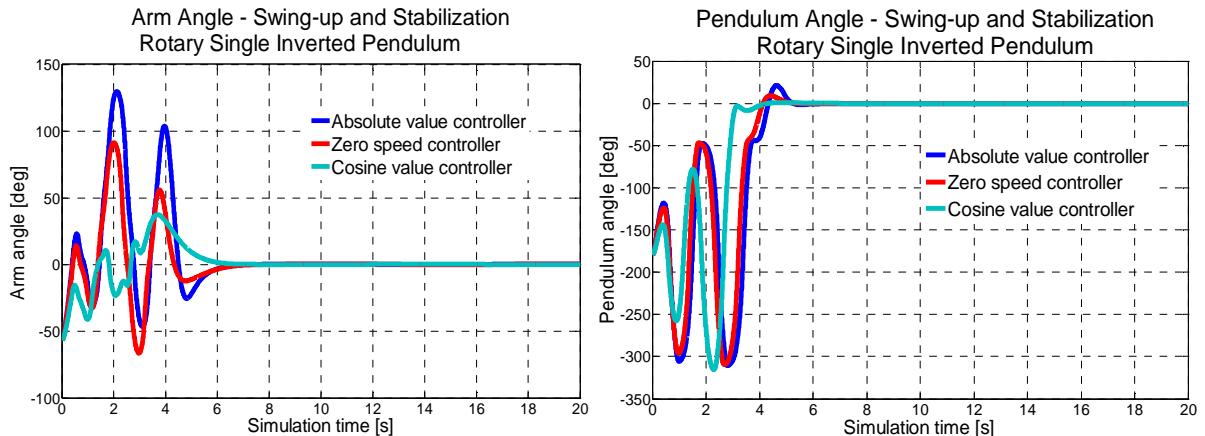


Fig. 13 Swing-up and stabilization of the rotary single inverted pendulum system – comparison of methods

The effectiveness of the three swing-up methods was compared in a simulation experiment involving the rotary single inverted pendulum model (Fig. 13).

6 Conclusion

The purpose of this paper was to present a comprehensive and unifying approach to the problem of modeling and control of classical and rotary inverted pendula dynamical systems. A structured, thematic *Simulink* block library, *Inverted Pendula Modeling and Control (IPMaC)*, was developed to serve as a software framework for all solved issues. The library is composed of a set of custom function blocks which centers around the simulation models of the classical single, classical double and rotary single inverted pendulum system. An integral part of the library is the *Demo Simulations* section, which is basically a collection of links to simulation schemes which illustrate the ways of interconnecting the blocks to solve various problems related to analysis and control of inverted pendula systems.

Practical importance of symbolic mathematical software, represented in MATLAB by *Symbolic Math Toolbox*, was demonstrated: general symbolic procedures were implemented that either yield the motion equations of a mathematical model for a user-chosen classical or rotary inverted pendulum system, or perform the symbolic linear transformation of a specified system in a chosen equilibrium point. Applications with graphical user interface were developed to provide a comfortable and user-friendly access to both procedures.

The presented *Simulink* block library offers solid program support for both modeling and control of inverted pendula systems, based on the modern object-oriented approach to problem solving. Since the library was designed as an open system with the classical and rotary inverted pendula simulation models at its core, it will be possible to add a wider variety of controller blocks and control schemes to the already included state-feedback and optimal control algorithms. Model predictive control algorithms and exact linearization are considered among other candidate structures.

Acknowledgements

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 (50%). This work is also the result of the project implementation Development of the Center of Information and Communication Technologies for Knowledge Systems (project number: 26220120030) supported by the Research & Development Operational Program funded by the ERDF (50%).

References

- [1] S. Jadlovská. *Modeling and Optimal Control of Inverted Pendula Systems*. Diploma (Master) Thesis. Supervisor: prof. Ing. Ján Sarnovský, PhD., Technical University of Košice, Faculty of Electrical Engineering and Informatics, Košice, 2011.
- [2] S. Jadlovská, A. Jadlovská. *A Simulink Library for Inverted Pendula Modeling and Simulation*. 17th Annual Conference Proceedings of the International Scientific Conference - Technical Computing Prague 2009, Congress Center of Czech Technical University in Prague, November 19, 2009, ISBN 978-80-7080-733-0.
- [3] S. Jadlovská, A. Jadlovská. *Inverted Pendula Simulation and Modeling – a Generalized Approach*. Proceedings of the 9th International Scientific – Technical Conference on Process Control '10, Říp 2010, June 7-10, University of Pardubice, Czech Republic, ISBN 978-80-7399-951-3.
- [4] S. Jadlovská, J. Sarnovský. *Matlab-Based Tools for Analysis and Control of Inverted Pendula Systems*. Electrical Engineering and Informatics – Proceeding of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, 2011, ISBN 978-80-553-0460-1
- [5] K. Sultan. *Inverted Pendulum, Analysis, Design and Implementation*. From: <http://www.mathworks.com/matlabcentral/fileexchange/3790>, accessed on 1-8-2009
- [6] H. X. Li, J. Y. Wang, Y. D. Gu, Y. B. Feng. *Hardware Implementation of the Quadruple Inverted Pendulum with Single Motor*. Progress in Natural Science, Vol. 14, Issue 9, 2002.
- [7] M. Schlegel, J. Mešťánek. *Limitations on the Inverted Pendula Stabilizability According to Sensor Placement*. 16th International Conference Process Control : Proceedings 2007 (June 11-14, 2007, Štrbské Pleso, Slovakia).
- [8] A. Bogdanov. *Optimal Control of a Double Inverted Pendulum on a Cart*. Technical Report CSE-04-006, OGI School of Science and Engineering, OHSU, 2004

- [9] M. Demirci. *Design of Feedback Controllers for a Linear System with Applications to Control of a Double Inverted Pendulum*. International Journal of Computational Cognition, Vol. 2, No. 1, p. 65-84, 2004
 - [10] J. Lam. *Control of an Inverted Pendulum*. From:
<https://research.cc.gatech.edu/humanoids/sites/edu.humanoids/files/Control%20of%20inverted%20pendulum.pdf>, accessed on 10-5-2011
 - [11] O. Modrlák. *Teorie automatického řízení I. - Základy analýzy a syntézy ve stavovém prostoru (Studijní materiály)*. Technical University Liberec, 2002.
 - [12] M. Korda. *Rotary and Linear Pendulum Control*. Bachelor Thesis. Faculty of Electrical Engineering, Prague, 2009.
-

Ing. Slávka Jadlovská
Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovak Republic
e-mail: slavka.jadlovska@tuke.sk

prof. Ing. Ján Sarnovský, CSc.
Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovak Republic
e-mail: jan.sarnovsky@tuke.sk