

Swing-up Problem of Inverted Pendulum Designed by DYNOPT Toolbox

Stepan Ozana, Martin Pies, Radovan Hajovsky, and Jana Nowakova

Abstract—The paper deals with design and implementation of swing-up algorithm for physical model of inverted pendulum. It presents detail derivation of the full nonlinear mathematical model, formulation of dynamic optimization task and its solution by use of DYNOPT toolbox that finds control signal which causes rising of the pendulum into upright position. The control scheme containing this control signal is implemented in PAC (programmable automation controller) WinPAC-8000 with the use of REX Control System.

Index Terms—MATLAB, regulators, educational products, control design.

I. INTRODUCTION

THE paper deals with design of swing-up control signal for physical model of inverted pendulum. The regulation of the pendulum itself in the upright position is quite complex model to control, therefore it is highly recommended to use algorithms based on so called modern control theory. So far, linear quadratic control has been successfully tested under real conditions. This work is inspired by other articles, for example the one referred in [1]. The previous work of this papers authors has been extended by adding swing-up problem so that the regulation of the model starts with special input signal that lifts up the pendulum. As soon as the angular position is within predefined interval, the regulation is switched to the automatic mode and the model becomes controlled by the state LQR controller designed according algorithms [2]. This is the most common way how to tackle the swing-up problem, yet there are more ways and algorithms how to compute the appropriate control signal responsible for lifting up. For the work described in this paper a numerical approach based on DYNOPT tool which is a special MATLAB third-party product designed to compute basic problems of dynamic optimization.

II. MATHEMATICAL MODEL

To be able to design and implement controller for inverted pendulum model, it is necessary to derive accurate mathematical model. This paper describes method of Lagrange equations of the second kind. Lagrange equations make up general form of Newton equations, because they make it possible to form movement equation even in the fields where Newton equations have no sense. The used method allows creating movement equations for a set of mass points (bodies) by introducing so called generalized coordinates. Other frequent methods use Newton movement rules themselves.

Manuscript received June 11th, 2013; revised July 26th, 2013. This work was supported by the project SP2013/168, named "Methods of collecting and transfer of the data in distributed systems" of Student Grant Agency (VSB - Technical University of Ostrava).

All authors are with the Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, 17. listopadu 2172/15, 70833 Ostrava, Czech Republic, Europe, e-mail: stepan.ozana@vsb.cz.

A. Lagrange equations

Derivation of Lagrange equations of the second kind is based on principle of virtual work, by which the plant is in a balance if virtual work $\delta\omega$ caused by all forces in the system is zero.

$$\delta\omega = \sum_{i=1}^n Q_i \delta q_i = 0$$

Q_i generalized force acting in the direction of
 i -th coordinate
 q_i i -th generalized coordinate

Forces acting in the systems can be divided into conservative and nonconservative. Conservative forces keep energy balance of the system, total sum of kinetic and potential energy is not affected by conservative forces. On the other hand, due to nonconservative forces the energy balance of the system change. These forces are for example dampening forces that depends on velocity.

The basic form of Lagrange equations is (1):

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i^* \text{ for } i = 1, 2, \dots, n \quad (1)$$

Lagrange function L , also referred to as kinetic potential, is defined as (2):

$$L = K - P [J] \quad (2)$$

K kinetic energy of the whole system
 P potential energy of the whole system

Movement equations for introduced generalized coordinates thus can be set up by use of scalar quantities only (kinetic and potential energy).

General procedure for application of Lagrange equations of the second kind is as follows:

- 1) Definition of independent generalized coordinates q_1, q_2, \dots, q_n .
- 2) Determination of kinetic energy K as a function of derivatives (velocities) of generalized coordinates $\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n$, and generalized coordinates q_1, q_2, \dots, q_n .
- 3) Determination of potential energy P as a function of generalized coordinates q_1, q_2, \dots, q_n . This function characterizes influence of all conservative forces.
- 4) Determination of Lagrange function.
- 5) Determination of generalized nonconservative forces $Q_1^*(t), Q_2^*(t), \dots, Q_n^*(t)$.
- 6) Performing derivation of movement equations.

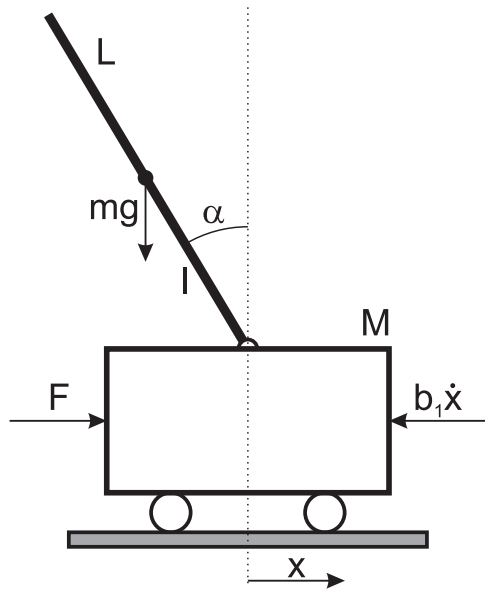


Fig. 1. Analysis of inverted pendulum model

B. Physical analysis of the model

The scheme of physical model of inverted pendulum and its basic variables and parameters is given in Fig. 1. Its physical realization can be seen from Fig. 2.

m	mass of the pendulum
g	gravity
M	mass of the cart
L	length of the pendulum
l	length between mass center and joint, $l = L/2$
F	force (manipulated value)
x	cart position
α	pendulum angle
b_1	friction of the cart
b_2	friction of the pendulum
J	inertia of the pendulum

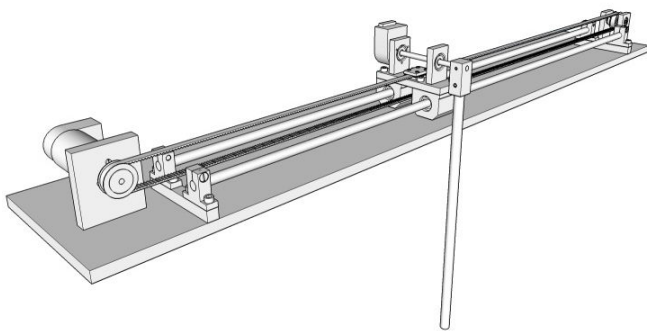


Fig. 2. Physical model of inverted pendulum

Having introduced parameters of the system it is possible to move on towards determination of movement equations. Overall kinetic energy of the system is given by (3):

$$K = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m\dot{x}^2 + \frac{1}{2}J\dot{\alpha}^2 + ml\dot{x}\dot{\alpha} \cos \alpha [J] \quad (3)$$

Where J represents inertia of the pendulum given by (4)

$$J = \frac{1}{3}mL^2 \quad (4)$$

or, by use of Steiner formula (5),

$$J = \frac{1}{12}mL^2 + ml^2 \quad (5)$$

The equations for the first generalized coordinate (x) are given by (6)–(10):

$$\frac{\partial K}{\partial \dot{x}} = M\dot{x} + m\dot{x} + ml\dot{\alpha} \cos \alpha \quad (6)$$

$$\frac{\partial K}{\partial x} = 0 \quad (7)$$

$$Q_x = F - b_1\dot{x} \quad (8)$$

$$\frac{d}{dx} \left(\frac{\partial K}{\partial \dot{x}} \right) = (M + m)\ddot{x} + ml\ddot{\alpha} \cos \alpha - ml\dot{\alpha}^2 \sin \alpha \quad (9)$$

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{x}} \right) - \frac{\partial K}{\partial x} = Q_x \quad (10)$$

The first output movement equation is then (11):

$$(M + m)\ddot{x} + ml\ddot{\alpha} \cos \alpha - ml\dot{\alpha}^2 \sin \alpha = F - b_1\dot{x} \quad (11)$$

The equations for the second generalized coordinate (α) are given by (12)–(16):

$$\frac{\partial K}{\partial \dot{\alpha}} = J\dot{\alpha} + ml\dot{x} \cos \alpha \quad (12)$$

$$\frac{\partial K}{\partial \alpha} = -ml\dot{x}\dot{\alpha} \sin \alpha \quad (13)$$

$$Q_\alpha = mgl \sin \alpha - b_2\dot{\alpha} \quad (14)$$

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\alpha}} \right) = J\ddot{\alpha} + ml\ddot{x} \cos \alpha - ml\dot{x}\dot{\alpha} \sin \alpha \quad (15)$$

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\alpha}} \right) - \frac{\partial K}{\partial \alpha} = Q_\alpha \quad (16)$$

The second output movement equation is then (17):

$$J\ddot{\alpha} + ml\ddot{x} \cos \alpha - ml\dot{x}\dot{\alpha} \sin \alpha = mgl \sin \alpha - b_2\dot{\alpha} \quad (17)$$

Equations (11) and (17) make up the full nonlinear model of the system.

C. State nonlinear model

Firstly, we obtain the formulas (18) and (19) that represent evaluation of second derivatives \ddot{x} and $\ddot{\alpha}$. This is easily carried out by use of equations (11) and (17):

$$\ddot{x} = \frac{F - b_1\dot{x} + ml\dot{\alpha}^2 \sin \alpha - ml\ddot{\alpha} \cos \alpha}{M + m} \quad (18)$$

$$\ddot{\alpha} = \frac{mgl \sin \alpha - b_2\dot{\alpha} - ml\ddot{x} \cos \alpha}{J} \quad (19)$$

Introducing the state variables, input variables and output variables, we get

III. DYNAMIC OPTIMIZATION

A. Introduction

The mathematical theory of dynamic programming used for solution of dynamic optimization problems dates to the early contributions of Bellman [3] and Bertsekas [4]. Dynamic programming was systematized by Richard E. Bellman. He began the systematic study of dynamic programming in 1955. The word "programming," both here and in linear programming, refers to the use of a tabular solution method and not to writing computer code.

As the analytical solutions are generally very difficult, chosen software tools are used widely. These software packages are often third-party products bound for standard simulation software tools on the market. As typical examples of such tools, TOMLAB and DYNOPT could be effectively applied for solution of problems of dynamic programming. We can classify the dynamic programming tasks concerning the type of final time (free/fixed) and final point (free/fixed), thus we can distinguish 4 combinations: problem with free time and free end point, problem with free time and fixed end point, problem with fixed time and free end point, problem with fixed time and fixed end point.

B. DYNOPT

DYNOPT is a set of MATLAB functions for determination of optimal control trajectory by given description of the process, the cost to be minimized, subject to equality and inequality constraints, using orthogonal collocation on finite elements method.

The actual optimal control problem is solved by complete parameterization both the control and the state profile vector. That is, the original continuous control and state profiles are approximated by a sequence of linear combinations of some basis functions. It is assumed that the basis functions are known and optimized are the coefficients of their linear combinations. In addition, each segment of the control sequence is defined on a time interval whose length itself may also be subject to optimization. Finally, a set of time independent parameters may influence the process model and can also be optimized.

IV. SOLUTION OF THE TASK IN DYNOPT

A. Adjusting of the model for DYNOPT

It is the problem with free time and fixed end point, because we don't know the time when the pendulum reaches the vertical position. The objective function is defined as (29):

$$J = \int_0^t dt \quad (29)$$

and it has to be minimized, finding the unknown final time t_f . For that reason, the current model of the system will be added by one more state variable $x_5 = t$. Overall system is then described by basic equations (25)–(28), plus (30).

$$\dot{x}_5 = 1 \quad (30)$$

Then the objective function becomes as described by (31).

$$\begin{aligned} x_1 &= x \\ x_2 &= \dot{x}_1 = \dot{x} \\ x_3 &= \alpha \\ x_4 &= \dot{x}_3 = \dot{\alpha} \\ u &= F \\ y_1 &= x_1 \\ y_2 &= x_3 \end{aligned} \quad (20)$$

With the following physical meanings

x_1	position of the cart	[m]
x_2	velocity of the cart	[m · s ⁻¹]
x_3	angular position of the pendulum	[rad]
x_4	angular velocity of the pendulum	[rad · s ⁻¹]
u	force (manipulated value)	[N]
y_1	position of the cart	[m]
y_2	velocity of the cart	[m · s ⁻¹]

by substitution into (18) and (19) we get (21) and (22)

$$\dot{x}_2 = \frac{u - b_1 x_2 + mlx_4^2 \sin x_3 - ml\dot{x}_4 \cos x_3}{M + m} \quad (21)$$

$$\dot{x}_4 = \frac{mgl \sin x_3 - b_2 x_4 - ml\dot{x}_2 \cos x_3}{J} \quad (22)$$

Having substituted \dot{x}_4 described in (21) by (22) we get (23)

$$\begin{aligned} \dot{x}_2 &= \frac{u - b_1 x_2 + mlx_4^2 \sin x_3}{M + m} - \\ &- \frac{ml \cos x_3 (mgl \sin x_3 - b_2 x_4 - ml\dot{x}_2 \cos x_3)}{J(M + m)} \end{aligned} \quad (23)$$

Similarly, having substituted \dot{x}_2 described in (22) by (21) we get (24)

$$\begin{aligned} \dot{x}_4 &= \frac{mgl \sin x_3 - b_2 x_4}{J} - \\ &- \frac{ml \cos x_3 (u - b_1 x_2)}{J(M + m)} - \\ &- \frac{ml \cos x_3 (mlx_4^2 \sin x_3 - ml\dot{x}_4 \cos x_3)}{J(M + m)} \end{aligned} \quad (24)$$

Making \dot{x}_2 and \dot{x}_4 single from (23), (24), and adding \dot{x}_1 , \dot{x}_3 (that stayed the same during adjustments), we get the full final state description of the model, described by (25)–(28).

$$\dot{x}_1 = x_2 \quad (25)$$

$$\begin{aligned} \dot{x}_2 &= \frac{gl^2 m^2 \cos x_3 \sin x_3 - Jlmx_4^2 \sin x_3}{l^2 m^2 \cos^2 x_3 - J(M + m)} \\ &+ \frac{Jb_1 b_2 - Ju - b_2 lmx_4 \cos x_3}{l^2 m^2 \cos^2 x_3 - J(M + m)} \end{aligned} \quad (26)$$

$$\dot{x}_3 = x_4 \quad (27)$$

$$\begin{aligned} \dot{x}_4 &= \frac{(M + m)(b_2 x_4 - glm \sin x_3)}{l^2 m^2 \cos^2 x_3 - J(M + m)} \\ &+ \frac{lm \cos x_3 (lmx_4^2 \sin x_3 + u - b_1 x_2)}{l^2 m^2 \cos^2 x_3 - J(M + m)} \end{aligned} \quad (28)$$

$$J = x_5(t_f) = \int_0^{t_f} \dot{x}_5 dt \quad (31)$$

This is required by Dynopt toolbox as the assignment has to be set up in so called Mayer form.

B. Solution on DYNOPT

The solution of the problem in DYNOPT lies in setup of needed scripts `confun.m`, `graph.m`, `objfun.m`, `process.m` according DYNOPT guide [5]. The core of the computation is defining the system itself as $F = f(x, u, p, t)$ represented by equations (25)–(28) plus (30), then the derivatives $\frac{\partial F}{\partial x}$ and $\frac{\partial F}{\partial u}$. The numerical computation consists of iterations which leads to the final control signal as shown in Fig. 3.

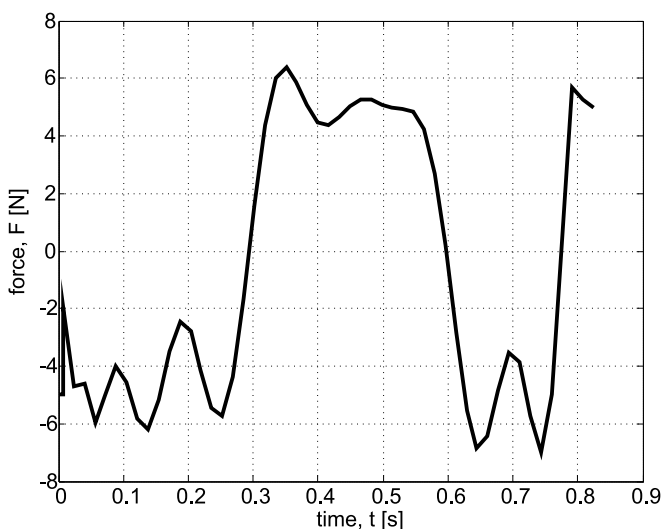


Fig. 3. Result of DYNOPT numerical computation finding the swing-up control signal

V. IMPLEMENTATION

A. Block Scheme of solution

The scheme of control circuit given in Fig. 5 is a combination of control scheme and electronic components, altogether representing the idea how to control the inverted pendulum model. It uses analogue and digital input and output modules (AI, AO, DO) of the programmable automation controller. It also shows electronic elements SG3524N and LM18200T that represents hardware current (torque) controller, according functional diagram, see reference [6], page 2. Connection diagram with the bridge LM18200T can be seen from reference [7], page 11. The middle part of the scheme represents system observer designed based on LQG technique, using Kalmann filter. This is used to generate approximations of two state variables which are not measured (velocities of the cart and pendulum \dot{x} , $\dot{\alpha}$). These approximated state variables together with two other measured variables (cart and pendulum position x , α) are then used as input to state (LQR) controller represented by matrix K .

The switch referred to as “T” represents switching to the automatic mode which is triggered once the angular position

of the pendulum is close to the vertical position 0 rad, this is predefined as an interval of angles between -0.5 to $+0.5$ rad.

B. REX Control System + WinPAC

REX control system is an advanced tool for design and implementation of complex control systems for automatic control. Basically it consists of two parts: the development tools and the runtime system, see Fig. 4. The control algorithms are composed from individual function blocks, which are available in the extensive function block library called RexLib. This library covers all common areas of automation and robotics. Moreover, several unique advanced function blocks are contained [11].

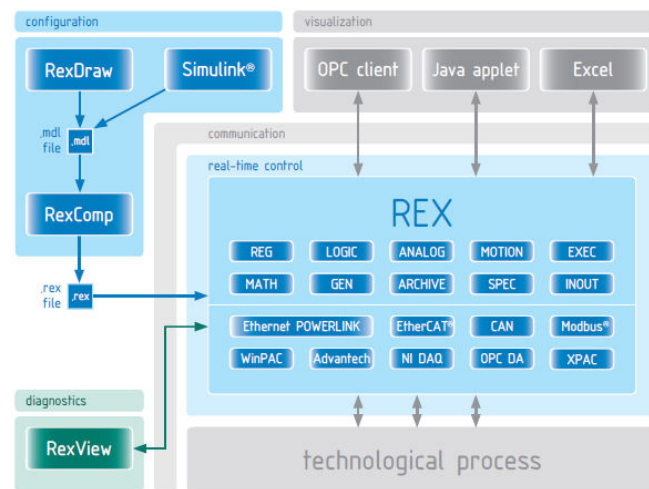


Fig. 4. Basic architecture of REX control system [11]

The algorithms are composed of individual function blocks, which are available in extensive function block libraries. These libraries cover not only all common fields of automation and regulation but offer also a variety of elements for high-level control algorithms. Runtime version of the REX control system is available for industrial PLC/PAC WinPAC and ViewPAC or their predecessor WinCon of the ICPDAS company.

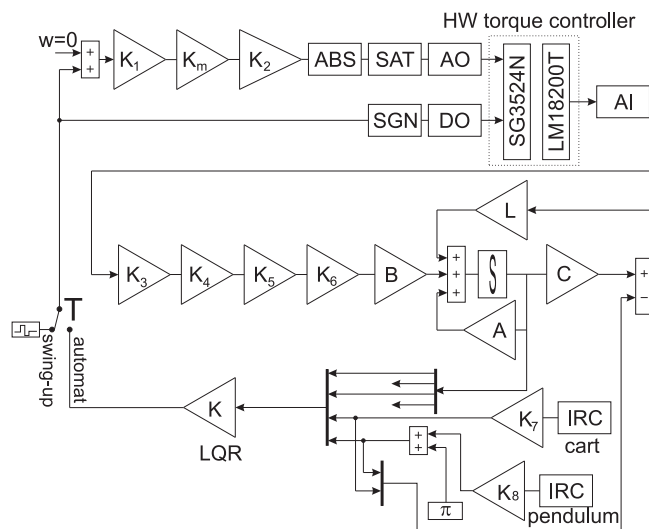


Fig. 5. Block control scheme of the software solution

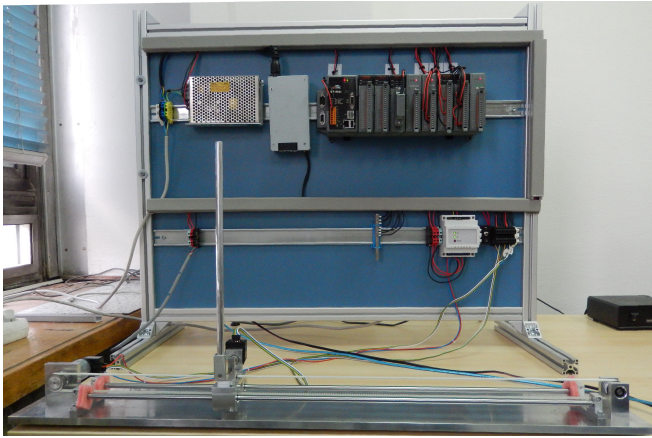


Fig. 6. Laboratory model of inverted pendulum at the Department of cybernetics and biomedical engineering

The block scheme in Fig. 5 has been implemented on programmable automation controller PAC WinPAC-8000, see [8]. The creation of control algorithm for this PAC is performed at two steps.

Firstly, it is creation of executive task that defines target platform and the main tick (time period) of the process. The executive can handle up to 5 tasks, particularly one fast QuickTask, and 4 slower tasks referred to as Level0 – Level3. Each of slower tasks has predefined its own tick based on the main tick and factor (priority). Currently the inverted pendulum model is connected to a QTask with 4 ms sampling period. Processor scheduling is controlled by REX core executive running on a target platform (WinCE), [9], [10].

The second step of implementation of control algorithm is creation of control scheme containing blocks for reading/writing from/to IO modules of the automation controller. The control scheme is similar and compatible with Simulink environment. This scheme together with executive scheme is stored as *.mdl file so as it can be open and even edited in Simulink provided REXLib library is installed on the computer, [11]. The advantage of compatibility between REX and Simulink is the possibility of tuning and verification of the algorithms without loading the program to the real hardware.

Besides described hardware setup (WinCE, WinPAC-8000, REX) the proposed approach using REX Control System allows implementation on many other modern and common platforms, such as Embedded PC/Single-Board PC + Linux/Xenomai + B&R I/O modules or usage of Raspberry PI or Arduino boards.

VI. CONCLUSION

The paper demonstrates use of DYNOPT Toolbox to design and implement swing-up control signal for physical model of inverted pendulum. The swing-up impulse for inverted pendulum designed and calculated by DYNOPT

toolbox was approximated by a sequence of two pulses with inverse orientation. Based on long-term experience, the swing-up problem is sensitive to constructional features of the system. As it can be seen in Fig. 2, not all the swing-up attempts are successful when the cart moves through the railways, as even slight friction changes may cause a failure. Further work will require adding a signal of cart absolute position that can represent a flag indicating successful or unsuccessful swing-up.

Based on this absolute position, the swing-up impulse may be adjusted in case of need. For example, it can handle some situations and conditions under which the swing-up is impossible at all. Typically, the cart needs approximately 2/3 of the length to be able to erect so if the cart is in the middle at the beginning than the swing-up is impossible due to insufficient space for swaying the rod.

The model is currently used for educational purposes at the Department of cybernetics and biomedical engineering for analysis and synthesis of the systems, representing a nonlinear very complex control system but attractive at the same time, see Fig. 6. Both swing-up and consequent LQR control have been successfully implemented and tested. The results implemented on a real system are documented in the form of YouTube video accessible through reference [12].

REFERENCES

- [1] H. Przemyslaw, "Stabilization of the cart-pendulum system using normalized quasi-velocities," in *Proceedings of the 17th Mediterranean Conference on Control & Automation*, 2009, pp. 827–830.
- [2] A. Tewari, *Modern Control Design With MATLAB and SIMULINK*. Chichester: Wiley, 2002.
- [3] R. Bellman, *Dynamic Programming*. Princeton, N. J.: Princeton University Press, 1998.
- [4] D. Bertsekas, *Dynamic Programming and Stochastic Control*. New York: Academic Press, 1976.
- [5] M. Cizniar, M. Fikar, and M. A. Latifi, "Matlab DYNAMIC OPTimisation code DYNOPT," Institute of Information Engineering, Automation, and Mathematics, Department of Information Engineering and Process Control, Bratislava, Slovak Republic, Tech. Rep., 2013, [Accessed on 3rd Jun 2013]. [Online]. Available: <http://www.kirp.chtf.stuba.sk/moodle/mod/resource/view.php?id=5464>
- [6] Texas instruments, "Regulating pulse width modulators SG3524 datasheet," 2009, [Accessed on 3rd Jun 2013]. [Online]. Available: <http://www.ti.com/lit/ds/symlink/sg2524.pdf>
- [7] National Semiconductor, "3A 55V H-bridge LMD18200 datasheet," 2012, [Accessed on 6th Jun 2013]. [Online]. Available: <http://www.ti.com/lit/ds/symlink/lmd18200.pdf>
- [8] ICP DAS Co. Ltd., "Winpac-8441/8841 WinCE Based Programmable Automation Controller," 2013, [Accessed on 6th Jun 2013]. [Online]. Available: http://www.icpdas-usa.com/documentation/Quickstarts/wp-8441_wp-8841.pdf
- [9] P. Balda, M. Schlegel, and M. Stetina, "Advanced Control Algorithms + Simulink Compatibility + Real-time OS = REX," in *16th Triennial World Congress of International Federation of Automatic Control*, vol. 16, Prague, Czech Republic, 2005, pp. 121–126.
- [10] M. Kocanek and P. Balda, "General sequential function charts editor," in *12th International Carpathian Control Conference*, Velke Karlovice; Czech Republic, 2011, pp. 191–194.
- [11] REX Controls, "REX controls," ZCU Plzen, Czech Republic, 2009, [Accessed on 10th Jun 2013]. [Online]. Available: <http://www.rexcontrols.com>
- [12] S. Ozana, "Inverted pendulum with swing-up," 2013, [Accessed on 6th Jun 2013]. [Online]. Available: <http://www.youtube.com/watch?v=rOt1FiJNVjA>