ACHIEVING AN EQUILIBRIUM POSITION OF PENDUBOT VIA SWING–UP AND STABILIZING MODEL PREDICTIVE CONTROL

Martin Gulan — Michal Salaj — Boris Rohaľ-Ilkiv [*]

In this paper, a complex control strategy to swing-up and balance the under-actuated pendubot system in one of its unstable equilibrium positions is presented. The initial swing-up maneuver is performed using an energy-based approach. For the purposes of stabilizing the system in its mid unstable equilibrium position a model predictive controller based on optimal control law with integral action is proposed. Satisfaction of constraints is ensured by introducing perturbations in the LQ control law acting as corrections when input amplitude/rate bounds are to be exceeded. The stability issue is addressed via additional constraints imposed on the terminal set. The emphasis of the paper is on the experimental realization of the pendubot swing-up followed by its balancing, which reveals the accuracy of the proposed control scheme.

K e y w o r d s: pendubot, swing-up, stabilization, model predictive control, constrained optimization

1 INTRODUCTION

The inverted pendulum presents a standard problem in the field of control system. It is often used to demonstrate various concepts in linear and nonlinear control such as stabilization or swinging up of unstable systems [1]. This system belongs to the class of non-minimum systems because of its internal dynamics. With regards to significant nonlinearities and input/state constraints, predictive control can be seeked as a systematic methodology to handle these challenging control problems [2].

In this paper a predictive control approach is discussed for the problem of stabilization of a nonlinear mechatronic system with fast dynamics, namely the pendubot, which together with other mechanical systems such as double pendulum on cart or acrobot belongs to the class of under-actuated double inverted pendulums [3].

The pendubot is an interesting example of inverted pendulum; it is an under-actuated system since the number of its control inputs is less than the number of its degrees of freedom, which makes it difficult to control. However, controlling such systems is challenging due to nonlinear dynamics, nonholonomic behaviour, and lack of linearizability exhibited by these systems [4]. The pendubot itself is a two-link planar robot with a single actuator at the base (shoulder) of the first link, and the (elbow) joint between the two links is unactuated and allowed to swing freely [5].

As outlined before, some of the principal control problems for the pendubot are swing-up, stabilization (balancing) and trajectory tracking. The problem of swing-up and stabilization has been addressed in several works, e.g. [6] used partial feedback linearization techniques for swing-up control, and linear quadratic regulator (LQR) for stabilization around the desired equilibrium position. In [7] the authors propose a swing-up strategy based on

an energy approach and passivity properties of the system where the second link moves and reaches the desired position according to a homoclinic orbit while some conditions on the initial state of pendubot have to be met to avoid a singularity. Another works investigating the balancing of pendubot involve strategies such as robust control, fuzzy control, quotient method or sliding mode control [8].

This work focuses mainly on design of balancing controller via model predictive control (MPC) with guaranteed stability while for the swing-up an approach adopted from [9] based on difference of actual energy and energy of pendulum needed to reach the desired position is used. Here, the main objective is to "take" the pendubot from the stable downward position to balance it at its mid upright unstable equilibrium position.

The structure of this paper is as follows. In Section 2, the pendubot system is described and its dynamic model is given. The state-space model is subsequently linearized around the desired equilibrium position and discretized. The swing-up strategy is presented in Section 3. Section 4 is dedicated to design of stabilizing model predictive controller with stability guarantees. In Section 4, the pendubot laboratory setup is described and experimental results are presented. Finally concluding remarks are drawn in Section 6.

2 SYSTEM MODEL

The mathematical model of an under-actuated planar robot can be in general derived by means of the Lagrange formalism

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial}{\partial \dot{q}_k}\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \frac{\partial}{\partial q_k}\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \frac{\partial}{\partial \dot{q}_k}\mathcal{R}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau}, \ k = 1, \ldots, n \quad (1)$$

[*] Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, Nám. slobody 17, 812 31 Bratislava, Slovakia, martin.gulan@stuba.sk
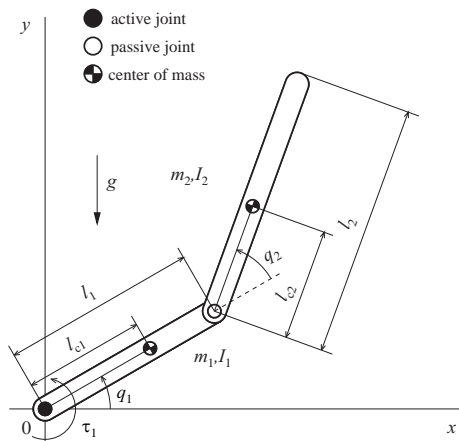
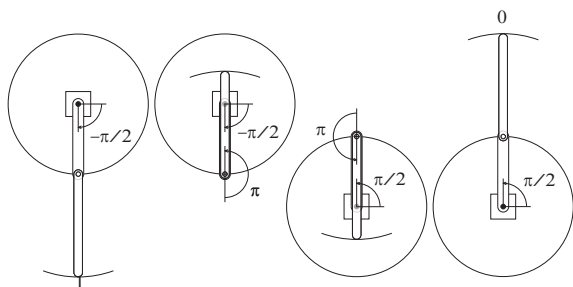**Fig. 1.** Schematic of the pendubot in a relative coordinate system

Fig. 2. Main equilibrium positions of the pendubot. From left to right: down-down $(-\frac{\pi}{2}, 0, 0, 0)$, down-up $(-\frac{\pi}{2}, 0, \pi, 0)$, up-down $(\frac{\pi}{2}, 0, \pi, 0)$, up-up $(\frac{\pi}{2}, 0, 0, 0)$

Table 1. Basic Mechanical Parameters of Pendubot Model

	Parameter	Symbol	Value*
Arm Link 1	mass	m_1	0.265 kg
	length	l_1	0.206 m
	distance of the center of mass of Link 1	l_{c1}	0.107 m
	moment of inertia about the centroid	I_1	0.0025 kg m^2
	friction coefficient	b_1	0.08 kg m^{2}s^{-1}
Pendulum Link 2	mass	m_2	0.226 kg
	length	l_2	0.298 m
	distance of the center of mass of Link 1	l_{c2}	0.133 m
	moment of inertia about the centroid	I_2	0.0011 kg m^2
	friction coefficient	b_2	0.00001 kg m^{2}s^{-1}
	torque (control input)	τ	(Nm)

* valid for the actual experimental setup described in Section 5

with the Lagrangian

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{V}(\boldsymbol{q}) \qquad (2)$$

defined as the difference between the kinetic energy $\mathcal{T}$ and potential energy $\mathcal{V}$ of the system, and the Rayleigh

dissipation function $\mathcal{R}$ accounting for friction. $\boldsymbol{q} = [q_1, \ldots, q_n]^\top \in \mathbb{R}^n$ is the vector of the angles of all joints in generalized coordinates, and $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_n]^\top \in \mathbb{R}^n$ denotes the external control force vector. By applying Eq. (1), the resulting equation of motion can be described in standard vector/matrix form

$$\mathbf{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{F}(\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \qquad (3)$$

where $\mathbf{D}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is a symmetric positive definite inertia matrix, $\mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n \times n}$ contains the Coriolis and centrifugal terms, $\mathbf{F}(\dot{\boldsymbol{q}}) \in \mathbb{R}^n$ is the vector of viscous frictional terms, and $\boldsymbol{F}(\boldsymbol{q}) \in \mathbb{R}^n$ denotes the vector of gravitational terms. For the pendubot system, schematically illustrated in Fig. 1, the following quantities can be obtained:

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \ \mathbf{D}(\boldsymbol{q}) = \begin{bmatrix} \theta_1 + \theta_2 + 2\theta_3 \cos q_2 & \theta_2 + \theta_3 \cos q_2 \\ \theta_2 + \theta_3 \cos q_2 & \theta_2 \end{bmatrix},$$

$$\mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} -\theta_3 \sin(q_2)\dot{q}_2 & -\theta_3 \sin(q_2)\dot{q}_2 - \theta_3 \sin(q_2)\dot{q}_1 \\ \theta_3 \sin(q_2)\dot{q}_1 & 0 \end{bmatrix},$$

$$\boldsymbol{F}(\boldsymbol{q}) = \begin{bmatrix} \theta_4 g \cos q_1 + \theta_5 g \cos(q_1 + q_2) \\ \theta_5 g \cos(q_1 + q_2) \end{bmatrix}, \text{ and } \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix},$$

where

$$\theta_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1, \quad \theta_4 = m_1 l_{c1} + m_2 l_1,$$
$$\theta_2 = m_2 l_{c2}^2 + I_2, \qquad \theta_5 = m_2 l_{c2}$$
$$\theta_3 = m_2 l_1 l_{c2},$$

are the parameter equations needed for the subsequent control design. Taking into account friction in both joints the dynamic system (3) can be rewritten as follows

$$\ddot{q}_1 = \frac{1}{\theta_1 \theta_2 - \theta_3^2 \cos^2 q_2} \big[\theta_2 \theta_3 \sin q_2 (\dot{q}_1 + \dot{q}_2)^2$$
$$+ \theta_3^2 \cos q_2 \sin(q_2)\dot{q}_1^2 - \theta_2 \theta_4 g \cos q_1 + \theta_3 \theta_5 g \cos q_2 \cos(q_1 + q_2)$$
$$+ \theta_2 \tau_1 - \theta_2 b_1 \dot{q}_1 + (\theta_2 + \theta_3 \cos q_2)b_2 \dot{q}_2 \big], \quad (4)$$

$$\ddot{q}_2 = \frac{1}{\theta_1 \theta_2 - \theta_3^2 \cos^2 q_2} \big[-\theta_3(\theta_2 + \theta_3 \cos q_2) \sin q_2 (\dot{q}_1 + \dot{q}_2)^2$$
$$- (\theta_1 + \theta_3 \cos q_2)\theta_3 \sin(q_2)\dot{q}_1^2 + (\theta_2 + \theta_3 \cos q_2)(\theta_4 g \cos q_1 - \tau_1)$$
$$- (\theta_1 + \theta_3 \cos q_2)\theta_5 g \cos(q_1 + q_2) + (\theta_2 + \theta_3 \cos q_2)b_1 \dot{q}_1$$
$$- (\theta_1 + \theta_2 + 2\theta_3 \cos q_2)b_2 \dot{q}_2 \big], \quad (5)$$

yielding pendubot's two non-linear equations of motion, which are to be used for design of a balancing controller.

The main mechanical parameters of the pendubot along with their values valid for the laboratory model are listed in Table 1.

By introducing the state vector $\mathbf{x}(t) = [x_1, \dot{x}_1, x_2, \dot{x}_2]^\top = [q_1, \dot{q}_1, q_2, \dot{q}_2]^\top$ the equations of motion of the pendubot may be written in the input-affine system representation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{f}(\mathbf{x})u, \qquad \mathbf{x}(0) = \mathbf{x}_0. \qquad (6)$$

For the purpose of subsequent controller design, the system (6) is linearized along the desired reference trajectory, in our case considered to be time-constant, *ie* $\mathbf{x}^* = [q_1^*, \dot{q}_1^*, q_2^*, \dot{q}_2^*]^\top$ corresponding to an input u^*, with initial condition $\mathbf{x}^*(0) = \mathbf{x}_0^*$. For sufficiently small deviations from the desired setpoint, $\mathbf{x}(t) = \mathbf{x}^* + \check{\mathbf{x}}(t)$, $\mathbf{y}(t) = \mathbf{y}^* + \check{\mathbf{y}}(t)$ and $u(t) = u^* + \check{u}(t)$, the system (6) can be described by the continuous, linear, time-invariant state-space formulation

$$\dot{\check{\mathbf{x}}} = \mathbf{A}\check{\mathbf{x}} + \mathbf{B}\check{u}, \qquad \check{\mathbf{x}}(t_0) = \check{\mathbf{x}}_0\,, \\ \check{\mathbf{y}} = \mathbf{C}\check{\mathbf{x}} \tag{7}$$

with $\check{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}^*$ and system matrix and input vector

$$\mathbf{A} = \frac{\partial}{\partial \mathbf{x}}(\mathbf{f}(\mathbf{x}) + \mathbf{F}(\mathbf{x})u)\Big|_{\mathbf{x}=\mathbf{x}^*, u=u^*}, \quad \mathbf{B} = \mathbf{F}(\mathbf{x}^*)\,.$$

In our one-input two-output case $u(t) = \tau_1$ is the input at time (t); $\mathbf{A} \in \mathbb{R}^{4\times4}$, $\mathbf{B} \in \mathbb{R}^{4\times1}$ and $\mathbf{C} \in \mathbb{R}^{2\times4}$ is the output matrix. Output of the system are the angular positions of both links $\mathbf{y}(t) = [q_1(t), q_2(t)]^\top$ measured in pendubot's joints.

Note that for $u = \tau_1 = 0$ the system (6) exhibits in total four main equilibrium positions, see Fig. 2, the first being a stable and the other three are unstable equilibrium positions. Challenging from the control point of view are the down-up (*mid*) and up-up (*top*) pendubot's equilibrium positions. In this work the control objective is to stabilize the system around its *mid* unstable equilibrium position, *ie* $\mathbf{x}^* = (-\pi/2, 0, \pi, 0)$ with respective input $u^* = 0$.

Next, a simple numerical integration by means of the Euler method, see *eg* [10], is utilized to calculate the time-discrete system representation for the sampling time T in the form

$$\check{\mathbf{x}}_{k+1} = \mathbf{A}_d\check{\mathbf{x}}_k + \mathbf{B}_d\check{u}(k)\,, \\ \check{\mathbf{y}}_k = \mathbf{C}_d\check{\mathbf{x}}_k \tag{8}$$

where $\mathbf{A}_d = e^{\mathbf{A}T}$, $\mathbf{B}_d = \left(\int_0^T e^{\mathbf{A}\tau}\mathrm{d}\tau\right)\mathbf{B} = \mathbf{A}^{-1}(\mathbf{A}_d - \mathbf{I})\mathbf{B}$, $\mathbf{C}_d = \mathbf{C}$.

3 SWING–UP CONTROL

As outlined in Section 1 there exists a variety of approaches to inverted pendulum swing-up. Predominantly, they are based on adding energy to the pendubot system until it equals the potential energy in the desired equilibrium position.

Our approach is based on the same but simplified idea. Here, the sequential adding of energy is achieved by swinging the arm (link 1) around its downward position ($\theta_1 = -\pi/2$). The pendulum (link 2) then approaches the upright position where it can be caught with an appropriate stabilizing strategy.

For the calculation of required torque the following control law [8] is used (subscript $(\cdot)_a$ denotes that variable refers to absolute coordinate system)

$$u = \mathrm{sat}_n\big(k_s(E - E_0)\big)\mathrm{sign}\big(\dot{\theta}_{2a}\cos\theta_{2a}\big) \tag{9}$$

where E, E_0 are the actual and desired energy of the pendulum, respectively. The gain k_s is a design parameter which adjusts the speed of swinging-up. The arm swings are limited to help smoothening the transition to balancing control. The correct polarity of control input applied to the pivot is determined by the signum term. The normalized energy E [8] can be determined as

$$E = m_2 g l_2\left(\frac{1}{2}\Big(\frac{\dot{\theta}_{2a}}{\omega_0}\Big)^2 + \cos\theta_{2a} - 1\right) \tag{10}$$

$$\omega_0 = \sqrt{\frac{m_2 g l_2}{I_2}} \tag{11}$$

is the frequency of pendulum's small oscillations around its downward position. A convenient choice is to set the desired upright position of pendulum as the base with zero energy, *ie* $E_0 = 0$. Hence, according to (10) the energy in the downward position is $-2m_2 g l_2$.

This approach is clearly a bang-bang strategy for large errors and proportional control for small errors [8]. Notice, however, that the described strategy does not consider the entire energy of the system, and therefore can not guarantee that both links will enter the basin of attraction of the balancing controller at the same time. Nevertheless, it has been proven true by a series of simulation and experimental runs; and after all the fact that the arm can be found in the vicinity of the downward position when the transition happens is implied by close-to-zero values of torque (see Fig. 6) as directly results from (9). This is however valid only for the *mid* equilibrium position, which is in the scope of this work.

4 BALANCING CONTROL

The balancing controller proposed in this Section uses the mathematical model, (4) and (5), which is further linearized for the *mid* unstable equilibrium position. To obtain better numerical conditioning of predictions' calculation, the technique of stabilization of the nominal model is employed. First, the linear quadratic controller is described, which is commonly used to balance a pendulum system and here also serves as a basis for the further derived predictive controller.
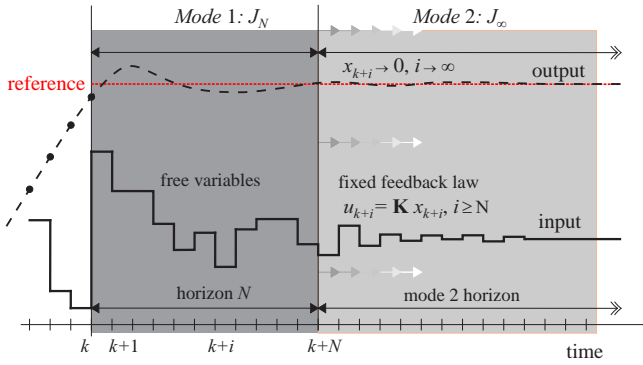
Fig. 3. Dual mode predictions and receding horizon control idea

4.1 Discrete Linear Quadratic Controller

For a discrete time-invariant state-space system, the LQ controller is defined as the fixed matrix feedback gain $\mathbf{K}$, which in case of our system minimizes the following infinite horizon cost function

$$J_k = \sum_{k=0}^{\infty}\big(\check{\mathbf{x}}_k^{\top}\mathbf{Q}\check{\mathbf{x}}_k + ru(k)^2\big) \tag{12}$$

where $\mathbf{Q} \in \mathbb{R}_+^{4\times 4}$ is a positive semi-definite state weighting matrix and $r \in \mathbb{R}_{++}$ is a positive input weighting scalar. The output torque at motor shaft applied at time instant (k) is then

$$u(k) = -\mathbf{K}\check{\mathbf{x}}_k \tag{13}$$

where the LQ feedback gain may be calculated from

$$\mathbf{K} = (\mathbf{B}_d^{\top}\mathbf{P}\mathbf{B}_d + r)^{-1}\mathbf{B}_d^{\top}\mathbf{P}\mathbf{A}_d \tag{14}$$

where $\mathbf{P}$ is the positive definite solution of trhe discrete time algebraic Ricatti equation (DARE) defined by

$$\mathbf{P} = \mathbf{Q} + \mathbf{A}_d^{\top}\big(\mathbf{P} - \mathbf{P}\mathbf{B}_d(\mathbf{B}_d^{\top}\mathbf{P}\mathbf{B}_d + r)^{-1}\mathbf{B}_d^{\top}\mathbf{P}\big)\mathbf{A}_d\,. \tag{15}$$

4.2 Stabilizing Model Predictive Controller

The main idea of predictive control is to compute an optimal input sequence by minimizing the predicted cost on a certain horizon and applying its first element to the plant repeatedly at each sampling instant, while respecting specified constraints. Hence, it contains the basic components of prediction, optimization and receding horizon implementation. The length of the prediction horizon has to meet both stability and computational requirements. Here, the so-called *dual mode* control was used, see Fig. 3, combining both finite and infinite horizon control. The first mode contains free optimization variables, while the second mode associated with the last state will be steered

into equilibrium by a stabilizing fixed feedback law, usually a LQ control law, *ie*

$$u(k+i\,|\,k) = \begin{cases} optimization \\ variables, & i = 0, 1, \ldots, N-1\,(\text{mode 1}), \\ \mathbf{K}\mathbf{x}_{k+i|k} & i = N, N+1, \ldots\,(\text{mode 2}) \end{cases} \tag{16}$$

with the assosiated cost in following form:

$$J_k = \sum_{i=1}^{N-1}\big(\mathbf{x}_{k+i}^{\top}\mathbf{Q}\mathbf{x}_{k+i} + \boldsymbol{u}_{k+i}^{\top}\mathbf{R}\boldsymbol{u}_{k+i}\big) + \mathbf{x}_{k+N}^{\top}\mathbf{P}_f\mathbf{x}_{k+N} \tag{17}$$

where the terminal weighting matrix $\mathbf{P}_f$ can calculated as the solution of the Lyapunov equation [11].

From the numerical robustness point of view it is convenient to express predictions as perturbations on the stabilizing linear feedback law according to the approach of plant pre-stabilization prior to prediction phase proposed by [12]. The control law is then

$$u(k + i\,|\,k) = \mathbf{K}\mathbf{x}_{k+i|k} + c(i\,|\,k) \tag{18}$$

where K is the optimal LQ gain and

$$c(i\,|\,k) = \begin{cases} optimization \\ variables, & i = 0, 1, \ldots, N-1\,,;(\text{mode 1}) \\ 0\,, & i \geq N\,, & (\text{mode 2}) \end{cases}$$

are the input perturbations that implicitly ensure satisfaction of constraints along the finite horizon N.

The prediction problem can be now reformulated using the so-called autonomous augmented state-space description [13] yielding an augmented cost function in the following form

$$J_k = \mathbf{x}_{\to k}^{\top}\mathbf{P}\,\mathbf{x}_{\to k} + \boldsymbol{f}_{\to k}^{\top}\mathbf{S}\,\boldsymbol{f}_{\to k} \tag{19}$$

where $\mathbf{S} = \mathrm{diag}\big(\mathbf{B}_d^{\top}\mathbf{P}\mathbf{B}_d + r\big)$, $\mathbf{S} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{f}_{\to k} \in \mathbb{R}^N$ is a vector of predicted input perturbations (More details on this topic can be found *eg* in [16]).

Since the cost (19) is to be minimized with respect to input perturbations, the first term that depends only on the actual state can be further omitted. Hence, it is sufficient to minimize

$$J_k = \boldsymbol{f}_{\to k}^{\top}\mathbf{S}\,\boldsymbol{f}_{\to k} + \text{const.} \tag{20}$$

In this work, for minimization of (20) the following constraining condition were considered.

– control action limitation (amplitude and rate constraints on control input),
– achieving a desired invariant and feasible target set at the end of prediction horizon.
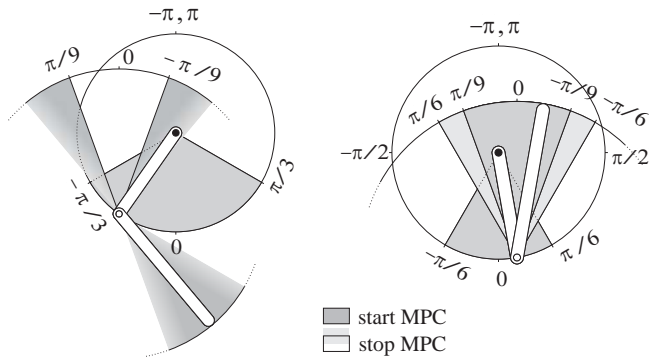
Fig. 4. Illustration of position ranges where particular control regimes of the pendubot are active. The angular values represent deflections with respect to the vertical axis (Not to be confused with the control errors q_1^s and q_2^s, respectively).

For a successful implementation of input constraints it is essential to derive prediction equations of inputs on horizon N. These can be obtained as follows.

$$u(k) = \mathbf{K}\mathbf{x}_k + c(k),$$
$$u(k+1) = \mathbf{K}\mathbf{x}_{k+1} + c(k+1)$$
$$= \mathbf{K}\Phi\mathbf{x}_k + \mathbf{K}\mathbf{B}_d c(k) + c(k+1),$$

$$\vdots = \vdots$$

$$u(k+N-1) = \mathbf{K}\Phi^{N-1}\mathbf{x}_k + c(k+N-1)$$
$$+ \sum_{i=0}^{N-2} \mathbf{K}\Phi^{N-i-2}\mathbf{B}_d c(k+i)$$

or, in a compact vector/matrix form,

$$\underset{\to k}{\boldsymbol{u}} = \mathbf{M}\mathbf{x}_k + \mathbf{N}\underset{\to k}{\boldsymbol{f}}. \tag{21}$$

where $\underset{\to k}{\boldsymbol{u}} = [u(k), u(k+1), \dots, u(k+N-1)]^\top$, $\mathbf{M}$ and $\mathbf{N}$ are appropriate input prediction matrices, and $\Phi = \mathbf{A}_d + \mathbf{B}_d\mathbf{K}$, $\Phi \in \mathbb{R}^{n \times n}$ is a closed-loop system state matrix, eigenvalues of which lie within a unit circle. Notice that $\mathbf{x}_{k+1} = \Phi\mathbf{x}_k$.

Prediction equations for input increments can be derived in a similar way yielding

$$\Delta\underset{\to k}{\boldsymbol{u}} = \mathbf{M}_\Delta\mathbf{x}_k + \mathbf{N}_\Delta\underset{\to k}{\boldsymbol{f}} + \boldsymbol{u}_0 \tag{22}$$

where $\boldsymbol{u}_0 = [-u(k-1), 0, \dots, 0]^\top \in \mathbb{R}^N$.

This paper further utilizes following low-complexity polytopic invariant and feasible target set as the terminal set [14]:

$$\mathcal{X}_N = \mathcal{P}_0(\mathbf{W}, \underline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\alpha}}) = \{\mathbf{x} \in \mathbb{R}^n : \underline{\boldsymbol{\alpha}} \le \mathbf{W}\mathbf{x} \le \overline{\boldsymbol{\alpha}}\} \tag{23}$$

where the polytope definition matrix $\mathbf{W} \in \mathbb{R}^n$ contains left eigenvectors of Φ; and $\underline{\boldsymbol{\alpha}}$, $\overline{\boldsymbol{\alpha}}$ denote its defining hyperplanes.

The terminal state $\mathbf{x}_{k+N}$, which must belong to the target set, *ie* $\mathbf{x}_{k+N} \in \mathcal{X}_N \subseteq \mathbb{R}^n$, can be expressed as

$$\mathbf{x}_{k+N} = \mathbf{G}_{\Phi N}\mathbf{x}_k + \mathbf{H}_{\Phi N}\underset{\to k}{\boldsymbol{f}} \tag{24}$$

where matrices $\mathbf{G}_{\Phi N}$ and $\mathbf{H}_{\Phi N}$ can be derived as

$$\mathbf{G}_{\Phi N} = \Phi^N, \qquad \mathbf{H}_{\Phi N} = \begin{bmatrix} \Phi^{N-1}\mathbf{B} & \Phi^{N-2}\mathbf{B} & \dots & \mathbf{B} \end{bmatrix}.$$

Parameters $\underline{\boldsymbol{\alpha}}$ and $\overline{\boldsymbol{\alpha}}$ of the target set $\mathcal{P}_0$ may be pre-calculated using an effective linear programming based (LP) algorithm, as shown *eg* in [16].

Notice also that since the control objective is to steer the system to a non-zero reference, the following shifted variables are introduced (In this case the target set $\mathcal{P}_0$ needs to be re-calculated for shifted input bounds $\underline{u} - u_{\text{ref}}$ and $\overline{u} - u_{\text{ref}}$ as well).

$$u^s(k) = u(k) - u_{\text{ref}},$$
$$\mathbf{x}_k^s = \mathbf{x}_k - \mathbf{x}_{\text{ref}} \tag{25}$$

which satisfy the state equation

$$\mathbf{x}_{k+1}^s = \mathbf{A}_d\mathbf{x}_k^s + \mathbf{B}_d u^s(k). \tag{26}$$

Referring to Section 2 it is clear that $\mathbf{x}^s = \check{\mathbf{x}}$, $\mathbf{x}_{\text{ref}} = \mathbf{x}^*$ and $u^s = \check{u}$, $u_{\text{ref}} = u^*(=0$ in our case).

The above described approach, however, requires an exact knowledge of the controlled system's dynamics and in case of any inaccuracies may lead to a residual steady-state error. This can be eliminated by introducing of *integral action* into the control law. To achieve that one can consider incremental form of control inputs in the cost function, however, more convenient in terms of invariant and feasible sets' computation is to augment the state vector $\mathbf{x}_k$ with the state of integrator ε_k, evolution of which is described by $\varepsilon_{k+1} = \varepsilon_k + (\mathbf{y}_k - \mathbf{y}_{\text{ref}})$. Let us therefore rewrite the shifted system (26) in the following fashion.

$$\tilde{\mathbf{x}}_{k+1}^s = \tilde{\mathbf{A}}_d\tilde{\mathbf{x}}_k^s + \tilde{\mathbf{B}}_d u^s(k),$$
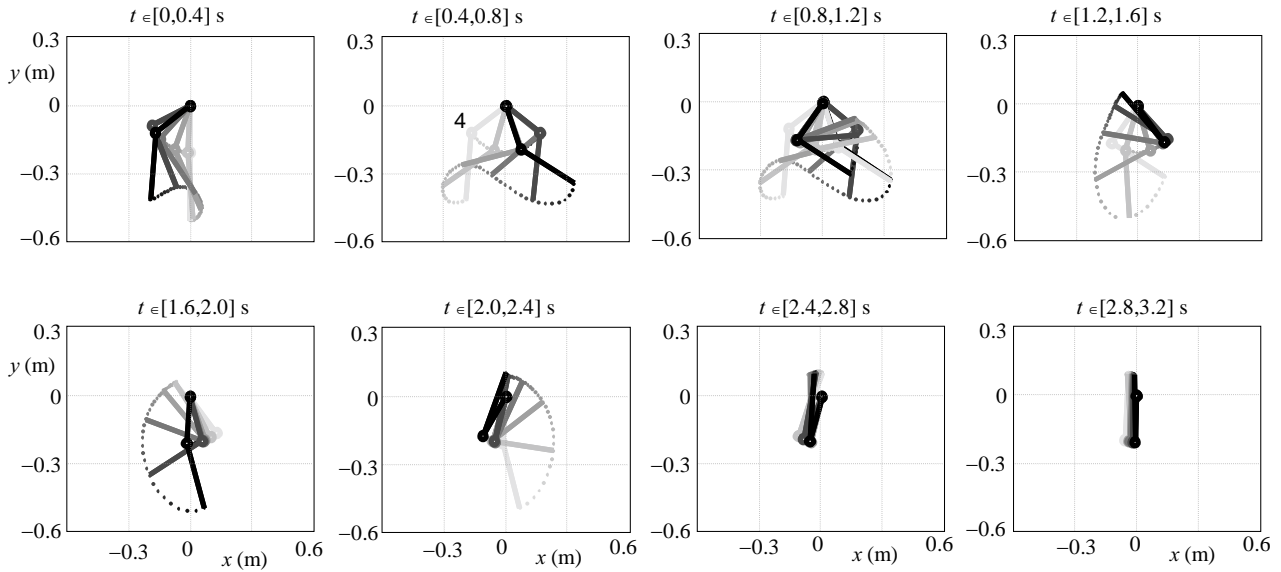$$\mathbf{y}_k^s = \tilde{\mathbf{C}}_d\tilde{\mathbf{x}}_k^s \tag{27}$$

with the augmented vectors and matrices

$$\tilde{\mathbf{x}}_k^s = \begin{bmatrix} \mathbf{x}_k^s \\ \varepsilon_k^s \end{bmatrix}, \; \tilde{\mathbf{A}}_d = \begin{bmatrix} \mathbf{A}_d & \mathbf{0} \\ \mathbf{C}_d & \mathbf{I} \end{bmatrix}, \; \tilde{\mathbf{B}}_d = \begin{bmatrix} \mathbf{B}_d \\ \mathbf{0} \end{bmatrix}, \; \tilde{\mathbf{C}}_d = \begin{bmatrix} \mathbf{C}_d & \mathbf{0} \end{bmatrix}.$$

Applying the LQR problem in order to obtain the controller with integral action $\tilde{\mathbf{K}} = [\mathbf{K_x} \; \mathbf{K}_\varepsilon]$ for system (27), the control signal is given by

$$u^s(k) = \tilde{\mathbf{K}}\tilde{\mathbf{x}}_k^s = \mathbf{K_x}\mathbf{x}_k^s + \mathbf{K}_\varepsilon\varepsilon_k^s \tag{28}$$

where $\tilde{\mathbf{K}}$ is calculated similarly to (14) and (15).

Fig. 5. Snapshots of the swing-up maneuver depicted in eight consecutive sequences with increasing darkness of snapshots as time increases during the respective sequence. Notice that the actual swing-up phase ends approximately at $t = 2.4$ s when it is switched to the balancing controller

Finally, to obtain the optimizing input sequence an optimization procedure has to be performed at each sampling instant (k), repeatedly solving a minimization problem of the form

$$\min_{\underset{\to k}{\boldsymbol{f}}} \left\{ \underset{\to k}{\boldsymbol{f}}^{\top} \tilde{\mathbf{S}} \underset{\to k}{\boldsymbol{f}} + \text{const} \right\}. \tag{29}$$

subjected to input amplitude and rate constraints imposed on control inputs as well as boundaries of the invariant target set

$$\mathbf{1}\underline{u} \leq \underset{\to k}{\boldsymbol{u}} \leq \mathbf{1}\overline{u},$$

$$\mathbf{1}\underline{\Delta u} \leq \Delta \underset{\to k}{\boldsymbol{u}} \leq \mathbf{1}\overline{\Delta u} \tag{30}$$

$$-\underline{\boldsymbol{\alpha}} \leq \tilde{\mathbf{W}}\mathbf{x}_{k+N} \leq \overline{\boldsymbol{\alpha}}$$

where $\mathbf{1} \in \mathbb{R}^N$ is a vector of ones.

Assume we have pre-calculated target set $\mathcal{P}_0(\tilde{\mathbf{W}}, \underline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\alpha}})$ and matrices $\tilde{\mathbf{S}}$, $\tilde{\mathbf{M}}$, $\tilde{\mathbf{N}}$, $\tilde{\mathbf{M}}_\Delta$, $\tilde{\mathbf{N}}_\Delta$, $\tilde{\mathbf{G}}_{\Phi N}$ and $\tilde{\mathbf{H}}_{\Phi N}$. Then the control is given by Algorithm 1.

Algorithm 1. MPC with guaranteed stability

- Observe (State estimation was not considered here since all the states can be directly obtained or derived) or measure actual system state at sample (k), $\mathbf{x}_k^s$, and set element $u(k-1)$ in vector $\boldsymbol{u}_0$.
- Solve the following quadratic programming problem

$$\underset{\underset{\to k}{\boldsymbol{f}}}{\text{minimize}} \qquad \underset{\to k}{\boldsymbol{f}}^{\top} \tilde{\mathbf{S}} \underset{\to k}{\boldsymbol{f}}$$

subject to
$$\mathbf{1}\underline{u} \leq \tilde{\mathbf{M}}\tilde{\mathbf{x}}_k^s + \tilde{\mathbf{N}} \underset{\to k}{\boldsymbol{f}} + u_{\text{ref}}\mathbf{1} \leq \mathbf{1}\overline{u},$$

$$\mathbf{1}\underline{\Delta u} \leq \tilde{\mathbf{M}}_\Delta \tilde{\mathbf{x}}_k^s + \tilde{\mathbf{N}}_\Delta \underset{\to k}{\boldsymbol{f}} + \boldsymbol{u}_0 \leq \mathbf{1}\overline{\Delta u},$$

$$-\underline{\boldsymbol{\alpha}} \leq \tilde{\mathbf{W}}\big(\tilde{\mathbf{G}}_{\Phi N}\tilde{\mathbf{x}}_k^s + \tilde{\mathbf{H}}_{\Phi N} \underset{\to k}{\boldsymbol{f}}\big) \leq \overline{\boldsymbol{\alpha}}.$$

- According to the receding horizon strategy apply only the first element of the sequence of optimal control moves

$$u^*(k) = \tilde{\mathbf{K}}\tilde{\mathbf{x}}_k + \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \underset{\to k}{\boldsymbol{f}^*} = \tilde{\mathbf{K}}\tilde{\mathbf{x}}_k + c^*(k)$$

to the controlled system, and restart the procedure.

By minimization of cost (29) with respect to constraints an optimal sequence of control input perturbations, $\underset{\to k}{\boldsymbol{f}^*}$, along the prediction horizon can be obtained, of which the first element, $c^*(k)$, is used as a correction to the optimal LQR control law according to (18). If the steady-state values u_{ref}, $\mathbf{x}_{\text{ref}}$, ε_{ref} and $\mathbf{y}_{\text{ref}}$ are determined properly, Algorithm 1 ensures that the shifted state converges to zero $(\mathbf{x}_k^s \to 0)$, and hence the actual state, integrator state and input converge to their steady-state values $(\mathbf{x}_k \to \mathbf{x}_{\text{ref}}, \varepsilon_k \to \varepsilon_{\text{ref}}, u(k) \to u_{\text{ref}})$.

To evaluate the constrained minimization quadratic problem (QP) described by Algorithm 1 the qpOASES on-line active set solver developed by Ferreau *et al* [17] was utilized.

4.3 Controller Switching

To implement the idea of a complex pendubot controller it must switch from swing-up to balancing controller whenever it enters its basin of attraction. To accomplish this, a switching strategy was devised and implemented into Simulink model. Its idea is schematically illustrated in Fig. 4.

5 REAL–TIME EXPERIMENTAL RESULTS

The experimental setup developed for testing efficient MPC in pendulum control applications consists of two main parts:
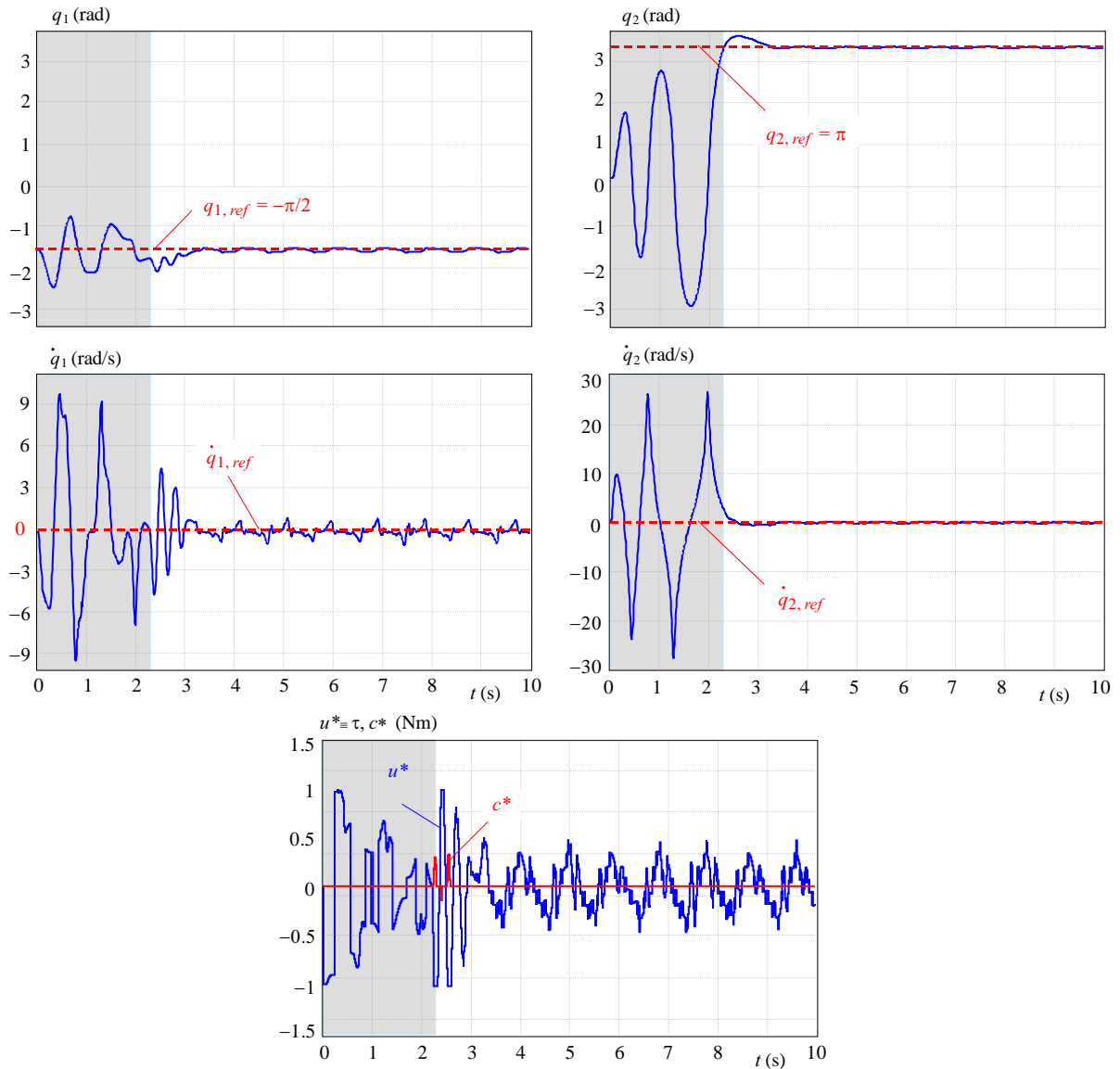
Fig. 6. Measurement results for swing-up and stabilization of the pendubot around the *mid* unstable equilibrium position. The first four graphs depict the evolution of system's state variables while the last one shows the respective control actions. The swing-up phase is distinguished by grey colour

Mechanical part of the system

The pendubot body consists of an adjustable frame, which allows the pendubot to be reassembled into a rotary, so-called Furuta pendulum or vice versa, and a double inverted pendulum. The joint between its two links is unactuated, whereas the joint between the first link and the frame is actuated by a motor. The physical parameters of the system (Tab. 1) can be modified by replacing one or both links with ones of desired length, as well as equipping them with additional weights.

Control and data acquisition electronics

The actuator of the system is a Mitsubishi HC-KFS43 servo motor with a WITTENSTEIN alpha CP060 gearbox. It uses a Mitsubishi MR-J2S-40A control unit working in torque control mode. It is also equipped with an absolute encoder allowing the real-time measurement of

the arm angular position. In order to measure the angle that the pendulum makes with the arm, an incremental rotary encoder OMRON E6B2-C is used.

The whole system is running under xPC rapid prototyping system (a target PC used for real-time control and a host PC used for controller development). Data acquisition is carried out utilizing a multifunction Humusoft MF-624 I/O card.

The experimental results for the swing-up and stabilization of the pendubot model around the *mid* position are presented in Fig. 6. It shows the development of system's states during the selected period of 10 seconds from start of the real-time experiment. The swing-up algorithm was initialized manually - by touching the pendulum gently. This is then swung up from the downward equilibrium position to the pre-determined vicinity of the upright *mid* equilibrium position (see Fig. 4), where the control is

taken over by the balancing predictive controller. After the initial transient the pendubot successfully balances about the desired equilibrium position. Notice that during the critical phase of entering the basin of attraction of the balancing MPC controller the appropriate input was ensured by perturbations that adjusted the overall torque when the constraints were to be violated (see Fig. 6). The swing-up maneuver itself is depicted in more detail in Fig. 5.

The following values of design parameters were used: $T = 0.01$ s, $N = 100$ steps, $k_s = 1.4$, $\mathbf{Q} = \mathrm{diag}(10, 0, 8, 0)$ and $r = 1$. Bounds used in minimization of cost function are $\underline{u} = -1$, $\overline{u} = 1$, $\underline{\Delta u} = -0.5$ and $\overline{\Delta u} = 0.5$ Nm.

6 CONCLUSION AND FUTURE WORK

In this paper a methodology and successful implementation of a control strategy to swing-up and balance the underactuated pendubot system has been presented. Here, an energy-based approach was utilized to perform the initial swing-up maneuver. For the purposes of balancing the pendubot in the up-right *mid* position a constrained model-based predictive controller with guaranteed stability was used.

The overall performance and efficiency of the proposed control algorithm were experimentally demonstrated on a laboratory test bench. It was shown to perform very satisfactorily and may be thus applied in a variety of not only under-actuated but mechatronic systems with fast dynamics in general.

Our future work will be focused on achieving and possibly traversing also the other equilibria of pendubot by means of a moving horizon state and parameter estimation-based nonlinear model predictive control exploiting the full nonlinear model of the system.

Acknowledgment

REFERENCES

[1] TOUATI, N.—CHEMORI, A. : Predictive Control for the Stabilization of a Fast Mechatronic System: from Simulation to Real-Time Experiments, The 6th IFAC Symposium on Mechatronic Systems, Hangzhou, China, 2013.

[2] MAYNE, D. Q.—RAWLINGS, J. B.—RAO, C. V.—SCOKAERT, P. O. M. : Constrained Model Predictive Control: Stability and Optimality, Automatica **36** No. 6 (2000), 789–814.

[3] FANTONI, I.—LOZANO, R. : Non-Linear Control for Under-actuated Mechanical Systems, The Institution of Electrical Engineers, London, 2001.

[4] XIN, X.—TANAKA, S.—SHE, J.—YAMASAKI, T : New Analytical Results of Energy-Based Swing-Up Control for the Pendubot, International Journal of Non-Linear Mechanics **52** (2013), 110–118.

[5] SPONG, M. W.—BLOCK, D. J. : The Pendubot: a Mechatronic System for Control Research and Education, Proceedings of the 34th IEEE Conference on Decision & Control, 1995, vol. 1, Dec 1995, pp. 65–73.

[6] BLOCK, D. J. : Mechanical Design and Control of the Pendubot, Master's thesis, University of Illinois, 1996.

[7] FANTONI, I.—LOZANO, R.—SPONG, M. W. : Energy Based Control of the Pendubot, IEEE Transactions on Automatic Control **45** No. 4 (2000), 725–729.

[8] LEE, K.—COVERSTONE-CARROLL, V. : Control Algorithms for Stabilizing Underactuated Robots, Journal of Robotic Systems **15** No. 12 (1998), 681–697.

[9] ÅSTRÖM, K. J.—FURUTA, K. : Swinging up a Pendulum by Energy Control, Automatica **36** No. 2 (2000), 287–295.

[10] ANTSAKLIS, P. J.—MICHEL, A. N. : Linear Systems, Birkhäuser, Boston, 2006.

[11] ROSSITER, J. A. : Model-Based Predictive Control: A Practical Approach, CRC Press, Boca Raton, 2003.

[12] ROSSITER, J. A.—KOUVARITAKIS, B.—RICE, M. J. : A Numerically Robust State-Space Approach to Stable-Predictive Control Strategies, Automatica **34** No. 1 (1998), 65–73.

[13] KOUVARITAKIS, B.—LEE, Y. I.—TORTORA, G.—CANNON, M. : MPC Stability Constraints and their Implementations, Preprints from the IFAC Conference Control Systems Design, Bratislava, Slovakia, 2000, pp. 205–212.

[14] BLANCHINI, F. : Set Invariance in Control, Automatica **35** No. 11 (1999), 1747–1767.

[15] KARAS, A.—ROHAĽ-ILKIV, B.—BELAVÝ, C : Praktické Aspekty Prediktívneho Riadenia, Slovak University of Technology / Slovenská e-akadémia, Bratislava, 2006. (In Slovak)

[16] FERREAU, H. J.—BOCK, H. G.—DIEHL, M. : An online active set strategy to overcome the limitations of explicit MPC, International Journal of Robust and Nonlinear Control **18** No. 8 (2008), 816–830.

Martin Gulan (Ing) is PhD student with the Institute of Automation, Measurement and Applied Informatics of Mechanical Engineering Faculty, STU in Bratislava. His research interests include physics-based modeling, nonlinear and optimal control, state and parameter estimation.

Michal Salaj (Ing) is PhD student with the Institute of Automation, Measurement and Applied Informatics of Mechanical Engineering Faculty, STU in Bratislava. His research interests include physics-based modeling, nonlinear and optimal control, state and parameter estimation.

Boris Rohaľ-Ilkiv (Prof, Ing, CSc) is with the Institute of Automation, Measurement and Applied Informatics of Mechanical Engineering Faculty, STU in Bratislava. His research is mainly focused on identification, modeling, adaptive and predictive control of industrial processes and mechatronic systems, including combustion engines, underactuated mechanical systems and active vibration systems.