Control and Cybernetics

vol. **49** (2020) No. 1

Closed-loop swing-up and stabilization of inverted pendulum by finite-horizon LQR applied in 2-DOF concept*

by

Tomas Docekal[1], Stepan Ozana[1,*], Abhaya Pal Singh[2] and Aleksandra Kawala-Sterniuk[3]

[1]Faculty of Electrical Engineering and Computer Science, VSB-TU Ostrava, 17. listopadu 15, 708 00 Ostrava, Czech Republic
ORCID 0000-0002-9485-0924; tomas.docekal@vsb.cz
∗corresponding author: ORCID 0000-0003-1102-8204 stepan.ozana@vsb.cz
[2]Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, India, ORCID 0000-0002-3046-9403
abhaya.aps@gmail.com
[3]Faculty of Electrical Engineering, Automatic Control and Informatics, Opole University of Technology, Opole, Poland, ORCID 0000-0001-7826-1292
kawala84@gmail.com

Abstract: The general framework of this paper is the control design of complex nonlinear systems. The proposed approach is demonstrated with the use of a case study regarding a typical mechatronic system - control design of inverted pendulum on the cart. The methodology used for the solution of this problem is based on two-degree of freedom control structure (2-DOF) with feed-forward and feedback terms. Feed-forward term represents a solution of trajectory generation problem and feedback term stands for a state controller. Both of these parts generally fall into the category of optimal control problems. The article focuses on the design of a finite-horizon linear quadratic controller and its application in 2-DOF structure with the use of customized LQR computation procedure, showing all necessary steps of the design, including source codes. It is proposed that the developed methodology is general and can be adopted for most of other nonlinear mechatronic systems, including unstable or non-minimum phase systems. This has been already tested successfully for models of both double and triple inverted pendulums. The functionality of the concept under real conditions can also be seen in Ozana (2018a) and Ozana (2018b) showing preliminary experiments with real apparatus.

Keywords: finite-horizon LQR; closed-loop swing-up; 2-DOF structure; inverted pendulum

*Submitted: December 2019; Accepted: April 2020

1. Introduction

The problem of inverted pendulum control is among the most popular in the field of technical cybernetics. It has been known for a long time and is still being used in laboratory conditions due to its complexity, especially in education of control theory. These properties predetermine it to be an appropriate subject for testing various new approaches and methods regarding modern control theory. Over the years, the inverted pendulum has become a benchmark in the field of control theory and robotics. At the same time, this problem is very attractive both for the lay public and experts.

Regarding the description of the respective laboratory setup, it is usually composed of a cart that can move along a certain form of linear guidance, and a pendulum arm that is connected to the cart via a free joint, so as to be able to rotate. The system is equipped with a single actuator capable of moving the cart. At the same time, this movement also influences the pendulum arm (its angle). Since the system has two degrees of freedom (linear movement of the cart and rotary movement of the pendulum arm), but only one actuator, it belongs to the category of the so called underactuated systems.

The problem of inverted pendulum control can be divided into two parts. The first and the easiest one is to stabilize the pendulum arm in the upright unstable position. In this situation, the system can be classified as a balance system. The task is to keep the pendulum arm in the upright position by the use of the cart moving to the sides. This is the equivalent of balancing the rod (or brush) on your finger or palm, but in one axis only in the case of the inverted pendulum.

This task can be handled in different ways, including the classic PD controller, the fuzzy or the state controller. Most of the solutions proposed and used all over the world refer to infinite-horizon LQR for this particular job.

The second part relates to transitions between the states, namely between the lower steady position, which is stable and the upper unstable position. In technical terms this is called a swing-up, in the opposite direction it would be a swing-down. These tasks are much more difficult to solve, especially in the case of complex nonlinear systems. There are more possibilities to solve this problem, as well. The most crucial factor is the decision regarding the concept of control (both open-loop and closed-loop approaches can be considered).

All control principles used for the control design of the inverted pendulum can also be used for a lot of other systems, appearing in practice. A typical example is the control of Segway balance vehicle, or its equivalent in the form

of a unicycle. Algorithms used to control such systems are also used, for example, for the so-called self-balancing robots, known as Acrobot, Pendubot, or a humanoid robot. In aerospace industry, this concept can be adopted for control design of a space rocket in which a tilting nozzle with a Cardan gimbal is used at the bottom of the rocket for stabilization purpose. Other aircraft (helicopters, aircraft, satellites) or vessels (boats, ships, submarines) are also included in the same category. An example of the medical technology is, for example, constituted by the iBOT wheelchair. In automotive industry, a back-up assistance system with a trailer is a typical application example.

2. Motivation

The basic motivation for the idea, which is proposed and described in this paper, was to find and verify a reliable and widely applicable method for control of nonlinear systems, represented by an inverted pendulum in this case. For these systems, more advanced methods are required for precise control to ensure the robustness and stability of the respective solution, as the classical methods usually appear insufficient.

A common problem that needs to be solved for complex systems is the transition between different states, often tracking a certain predefined trajectory. It may be a trajectory that meets some of the optimization criteria, where deviations lead, for example, to greatly increased costs or fuel consumption. Such a situation may in some cases be critical if we deal with a system featuring, in general terms, a limited amount of fuel or energy. For these and other reasons, it is necessary to ensure that these trajectories are maintained by using the feedback control that will provide a feedback along the trajectory.

3. Basic research concerning the categories of methods used for inverted pendulum

Due to the popularity of the inverted pendulum system, there have been a large number of papers that deal with its control. Not only conventional approaches, but also algorithms involving methods of modern control theory have been tried out, such as LQ control, model predictive control, adaptive control or H-∞. Last but not least, fuzzy control or neural networks play an important role in control of such complex systems as this is described in Ichtev (2008). The use of fuzzy controllers may not only involve stabilization in the upright position, for which the Mamdani-based controller is used, but it also makes it possible to solve the swing-up problem by introducing appropriate rules to make the pendulum swing, as this can be seen in Ichtev (2008). In a way, this is a technique for switching two control elements connected to one element, which is described

below. Regarding the neural network, it is necessary to let it learn how the inverted pendulum behaves and in essence to copy the attitude of a person who would manually move the pendulum to the upright position, see Hercus, Wong, Shee, and Ho (2013).

This paper focuses on approaches related to modern control theory. These approaches involve several general ways of providing solutions, which then differ in the implementation of individual parts.

Trajectory tracking

This is the approach described in this paper and in Tum, Gyeong, Park, and Lee (2014). First, there comes the computation of feed-forward trajectory that the system is supposed to follow and then the finite-horizon LQR controller provides the feedback control. In upright position, the steady-state of the time-varying finite-horizon LQR corresponds to the infinite-horizon LQR controller computed in this particular operation point.

Reference feed-forward control + switching stabilizing controller

This approach uses the design of the control signal to perform a swing-up, which is then applied to the system as open-loop (feed-forward) control. As soon as the pendulum reaches the proximity of upright position, the control structure is switched to a controller that provides stabilization in the upright position, designed for a single operation point. Various methods can be used for this, from PID controller to the LQR. Design of control signal can also be done in different ways, by solving dynamic optimization job (Ozana, Pies and Hajovsky, 2014) or, as in Kennedy and Conlon (2011), by experimental adjustment of the harmonic waveform.

Swaying controller + switching stabilizing controller

This is another approach that uses the switching mechanism between the two control parts. The distinction lies in the implementation of the swing-up. In this case, a certain control term is used to provide a swinging of the pendulum. The different methods are described in Durand, Castellanos, Marchand and Sanchez (2013) and Yang and Zheng (2018). It may be a P-controller connected in an unstable mode or simply a term which, when passing through the lower position, moves the pendulum cart to increase the pendulum's energy and increase the amplitude of its oscillation. More advanced approaches from this category

calculate the amount of the pendulum energy, which consists of a potential and kinetic component, based on the measured quantities. Once this value is equal to the potential energy corresponding to the upright position, it is not necessary to increase the pendulum's energy and it is simply sufficient to let it traverse to the inverse position.

4. Novelty

The search has shown that the overwhelming majority of solutions with a conventional or fuzzy controller use two separate control terms with switching between the corresponding two modes. The swing-up itself is solved either in an open loop with a predetermined control signal, or by gradual swinging of the pendulum. The here presented approach differs, in particular, by the fact that for both jobs regarding the inverted pendulum, i.e. the swing-up and stabilization in upright position, the only one common time-varying controller will be used, with the help of predefined trajectories to be tracked down during the swing-up and afterwards. This approach is much better suited to the general needs of control of non-linear systems. With its use it is also possible to assume a better repeatability of the entire swing-up process, as any disturbing influences that can vary will be compensated by the feedback controller, instead of changing the entire character of the control signal for the swing-up.

Also, it can be said that applicability of such a proposed solution is very high, because once this approach works for unstable and non-minimum phase nonlinear systems, considered as a worst-case, it will also work for other categories of cyberphysical systems.

The approach, presented in this paper, has been developed completely independently of Tum, Gyeong, Park and Lee (2014), who have used the same concept. One of the distinctions is that the proposed solution focuses on the design of time-varying controller in a deep level of details, showing all key parts of Matlab source codes as there is no embedded function for a finite-horizon LQR. Anyone interested in this problem can easily adopt the proposed solution for their own system. Moreover, controller design was improved by an additional optimization procedure. It is used for selecting optimal weights for finite-horizon LQR, which are significantly important for resulting time-varying controller.

Also, the work of Tum, Gyeong, Park and Lee (2014) uses a PD position controller to generate the required acceleration. In our paper, we use a speed controller, and so the acceleration does not have to be integrated twice but just once. It is a similar concept as in Yokoyama, Mihara, Suemitsu and Matsuo (2011), who use acceleration of the cart as the input of the pendulum. Note that most of the papers related to inverted pendulum consider the force as

constituting the input to the system. This is fine for simulation results, but the respective authors mostly do not mention how they actually achieve feedback law under real conditions, in which one has to apply a speed controller, or a position controller, or a torque controller.

5. Description of the proposed solution

The two-degrees of freedom structure

The here described problem falls into a trajectory tracking category of problems. It is assumed that the reference trajectories, to be tracked down, are already known and the tracking process is ensured by the feedback control term. A control structure with two degrees of freedom can be used to solve this kind of problem (Fig. 1).

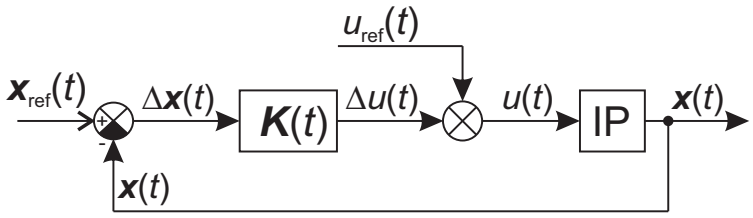

Figure 1. Two-degrees of freedom control structure schematic

As this is depicted in Fig. 1, the system contains two control parts – feed-forward and feedback. The feed-forward part is implemented by the reference (permissible) control signal $u_{ref}(t)$, which has been computed in order to provide a transition of the inverted pendulum system (IP) from the initial state to the final one. For the case study, considered in this paper, it is represented by the swing-up, where the initial state is the lower stable position and in the final state it is the upright unstable position. The second part is a feedback term, providing the tracking of the reference trajectories $x_{ref}(t)$, which are commonly obtained during the design of the control signal. These are basically the waveforms arising at the output of the system once reference control input $u_{ref}(t)$ is applied (to the ideal system with no disturbances, or for a perfect model). The deviation between the state variables of the real system $x(t)$ and the reference $x_{ref}(t)$ is used as an input to the time-varying controller $K(t)$. This generates an additional part of the control signal $\Delta u(t)$, which compensates for any possible deviations of the states $\Delta x(t)$ and forces the system back to the reference trajectory.

Linear Quadratic Control

For reasons of keeping the integrity of this paper, a very brief overview of finite-horizon LQR theory is provided in the following paragraph.

For a state-space description of linear system

$$x' = A \cdot x + B \cdot u \tag{1}$$

$$y = C \cdot x + D \cdot u \tag{2}$$

the finite-horizon LQR uses the following cost function:

$$J = \frac{1}{2} x^T(t_f) \cdot S \cdot x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} \left[x^T(t) \cdot Q \cdot x(t) + u^T(t) \cdot R \cdot u(t) \right] \mathrm{d}t \tag{3}$$

In order to solve an optimal control problem, Hamiltonian function is constructed and then solved to obtain the optimal control signal $u^*(t)$.

$$\begin{aligned} H = -\tfrac{1}{2} \left[x^T(t) \cdot Q \cdot x(t) + u^T(t) \cdot R \cdot u(t) \right] + \\ + \lambda^T(t) \cdot \left[A \cdot x(t) + B \cdot u(t) \right] \end{aligned} \tag{4}$$

Upon fulfilling the necessary condition for the optimal solution

$$\frac{\partial H}{\partial u} = 0 \tag{5}$$

we get

$$-R \cdot u(t) + B^T \cdot \lambda(t) = 0 \tag{6}$$

and thus

$$u^*(t) = R^{-1} \cdot B^T \cdot \lambda(t). \tag{7}$$

The Lagrangian multiplier can be written down in the form of

$$\lambda(t) = -P(t) \cdot x(t) \tag{8}$$

where $P(t)$ represents the solution to the differential Riccati equation

$$P'(t) = -P(t) \cdot A - A^T \cdot P(t) + P(t) \cdot B \cdot R^{-1} \cdot B^T \cdot P(t) - Q \tag{9}$$

and where

$$P(t_f) = S \tag{10}$$

represents its final value.

Note: S corresponds to the solution of the algebraic Riccati equation used to compute the infinite-horizon LQR (applied in upright position of the inverted pendulum).

Finally, optimal control signal $u^*(t)$ can be expressed as time-varying state feedback:

$$u^*(t) = -R^{-1} \cdot B^T \cdot P(t) \cdot x(t) = K(t) \cdot x(t). \tag{11}$$

Physical HW setup

The photo of the physical setup used for testing the proposed solution is shown in Fig. 2.

Figure 2. The photo of the real HW setup

The actuator of the system is physically represented by the DC motor, driven by the PWM (pulse width modulation) signal, provided by electronic unit. The value of the PWM signal is determined by the speed control loop, based on the original acceleration signal $u(t)$, generated by time-varying state controller $K(t)$. Notation of the signals used in this chapter corresponds to the state-space description given by (19)-(22). The feedback to the controller is represented by two incremental encoders providing information on pendulum arm angle x_1 and cart position x_3. The remaining unmeasured states, representing pendulum angular speed x_2 and cart speed x_4, are approximated by numeric derivatives.

The control scheme for the real hardware setup is shown in Fig. 3. The control system (C) generates acceleration $u(t)$, which is integrated in time and turned into the speed $v(t)$. This signal is brought to the input of the regulated plant (S), formally denoted as the speed setpoint $v_{sp}(t)$. The speed control loop, which is a part of the regulated plant (S), contains actuator (DC motor

and electronic unit), connected to the moving cart with pendulum arm via belt pulley, and speed controller (R_v). The speed control loop is approximated by a fast first-order system $G_{RV} = \frac{1}{\tau s+1}$. The time constant τ has been identified experimentally. It represents dynamical behavior of the entire speed control loop that uses a fast PI controller R_v applied for DC motor connected to the moving cart with pendulum arm via belt pulley. The information on the current output cart speed x_4, used for the feedback, is calculated via numerical approximation of the derivative based on cart position x_3.

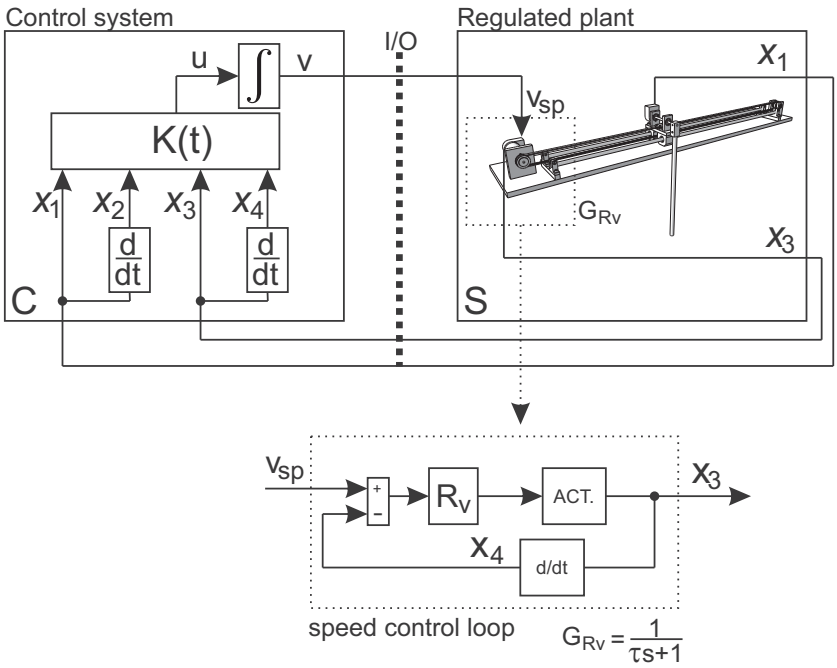

Figure 3. The control scheme for the real HW setup

For the simulation and control design purposes, the control scheme including the idea of approximation of the speed control loop is explained in Fig. 4.

The DC motor drives the cart using the speed controller, which induces appropriate acceleration $u_c(t)$. This signal then enters the input of the block IP, whose dynamics is described by (19)-(22). If the drive is strong enough, we approximately suppose that $u(t) \approx u_c(t)$. This assumption is used in this paper and therefore the speed control loop remains unmodelled for the purpose of trajectory planning. It is also possible to incorporate this unmodelled dynamics

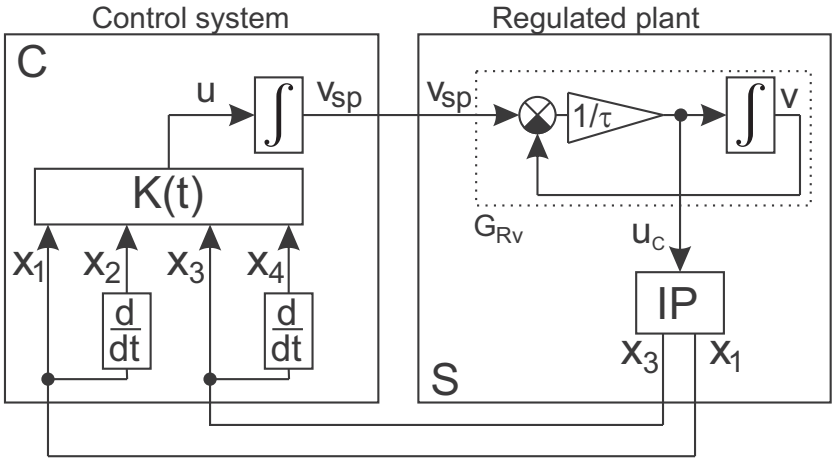

Figure 4. The control scheme for the simulation purposes

into trajectory planning problem, but it is not necessary. We repeatedly verified that this does not lead to improvement of the closed-loop control quality. Hence, all of the simulation results in this paper are associated with the scheme presented in Fig. 5.

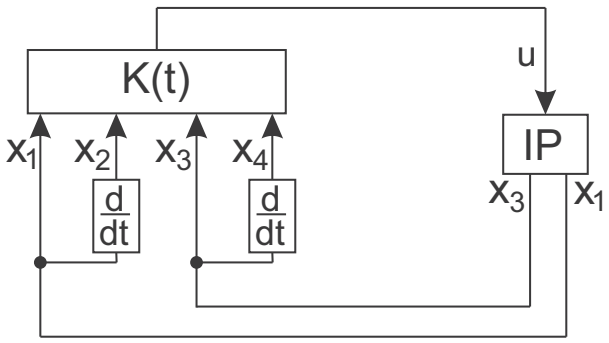

Figure 5. Simplified control scheme for the simulation purposes

Nonlinear model of inverted pendulum

The situation scheme used for identification of the system is introduced in Fig. 6. The differential equation, describing the movement of the inverted pendulum on the cart considering friction has been adopted from Yokoyama, Mihara,

Suemitsu and Matsuo (2011) and modified to the ultimate form of Eq. (12). The modification is formal: apart from using different symbols for physical variables, the second derivative of pendulum arm angle is multiplied by the moment of inertia, denoted "I".

$$I \cdot \varphi'' - m \cdot g \cdot l \cdot \sin\varphi + c \cdot \varphi' = m \cdot l \cdot u \cdot \cos\varphi. \tag{12}$$

The I term in this equation represents moment of inertia of pendulum with respect to the pivot P, by which the axis of rotation passes through. It can be computed according to Steiner's formula:

$$I = J + m \cdot l^2 \tag{13}$$

where J represents the moment of inertia of the pendulum arm with respect to the center of mass, and $l =$|MP| is the distance from the pivot P to the center of mass. For a homogeneous cylindrical rod of the length L, where $l = L/2$, the moment of inertia I can be computed as follows:

$$\begin{aligned} I &= \frac{1}{12} m \cdot L^2 + m \cdot l^2 = \frac{1}{12} m \cdot (2 \cdot l)^2 + m \cdot l^2 = \\ &= \frac{4}{12} m \cdot l^2 + m \cdot l^2 = \frac{16}{12} m \cdot l^2 = \frac{4}{3} m \cdot l^2. \end{aligned} \tag{14}$$

By substitution of I into Eq. (12) we get

$$\begin{aligned} \tfrac{4}{3} m \cdot l^2 \cdot \varphi'' - m \cdot g \cdot l \cdot \sin\varphi + c \cdot \varphi' = \\ = m \cdot l \cdot u \cdot \cos\varphi \end{aligned} \tag{15}$$

or, in another form,

$$\varphi'' - \frac{3}{4} \cdot \frac{g}{l} \cdot \sin\varphi + \frac{3}{4} \cdot \frac{c}{m \cdot l^2} \cdot \varphi' = \frac{3}{4} \cdot \frac{1}{l} \cdot u \cdot \cos\varphi \tag{16}$$

where φ is the pendulum arm angle with respect to the vertical axis, u is the acceleration of the cart and g is gravity acceleration. Due to the way of identifying the friction coefficient, we introduce a new term b (still representing a friction) as follows:

$$b = \frac{3}{4} \cdot \frac{c}{m \cdot l^2}. \tag{17}$$

The final form of the differential equation, to be used for modelling, identification and control design purposes and further computations, is as follows:

$$\varphi'' - \frac{3}{4} \cdot \frac{g}{l} \cdot \sin\varphi + b \cdot \varphi' = \frac{3}{4} \cdot \frac{1}{l} \cdot u \cdot \cos\varphi \tag{18}$$

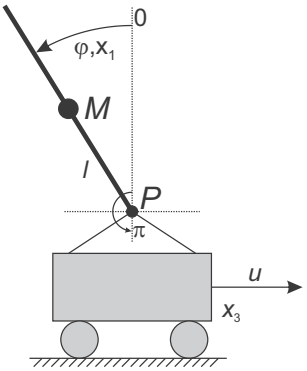

Figure 6. Situation scheme for analytical identification

This equation can be rewritten into a state-space description by (19)-(22), considering four state variables. Note that this state-space description corresponds to the block IP, referred to in Figs. 4 and 5.

$$x_1' = x_2 \tag{19}$$

$$x_2' = \frac{3}{4} \cdot \frac{g}{l} \cdot \sin x_1 + \frac{3}{4} \cdot \frac{1}{l} \cdot u \cdot \cos x_1 - b \cdot x_2 \tag{20}$$

$$x_3' = x_4 \tag{21}$$

$$x_4' = u \tag{22}$$

where

x_1[rad] pendulum arm angle,
x_2[rad/s] pendulum angular speed,
x_3[m] cart position,
x_4[m/s] cart speed.

Algorithm for computation of finite-horizon LQR

The first step in the overall solution is the calculation of the solution to the algebraic Riccati equation for the upright position, i.e. the unstable steady state. This is the procedure used when calculating the LQR controller at an infinite horizon for a linear system or a linearized system at one operating point. Matlab has the corresponding 'care' function as a part of Control System Toolbox:

[Plqr,L,G] = care(A0,B0,Q,R).

Parameters *A0* and *B0* are the matrices of state-space description of inverted pendulum linearized in the upright position. Weights Q and R are selected by the expert to reflect the importance of the individual state variables and the control signal. They contain non-zero elements on the main diagonal only. Thanks to these values it is possible to balance the different physical units of the state variables. For example, the deviation of the state x_1 (pendulum angle) by 0.1 rad (approximately 5.7) has a smaller weight than the equally large displacement of state x_3, i.e. change of position by 0.1 m. Based on these assumptions, the basic setting of weights Q and R was selected. After several iterations of slight modifications in the weights, the following setting was used for computations:

$$Q = [10 \quad 0 \quad 0 \quad 0; \quad 0 \quad 10 \quad 0 \quad 0; \quad 0 \quad 0 \quad 50 \quad 0; \quad 0 \quad 0 \quad 0 \quad 180];$$
$$R = 1.$$

The obtained solution of the algebraic Riccati equation serves as an initial condition for solving the differential Riccati equation. This will go back in time, i.e. it will come out of the final state, for which there is already the solution at disposal, and it will proceed towards the initial state (the lower position of the pendulum). Solution of the differential equation itself can be done via 'ode45'. available in Matlab.

```
S=reshape(Plqr,16,1)
opt = odeset('AbsTol',...
1.0e-08,'RelTol',1.0e-08)
[t,P] = ode45(@riccatiEquation,...
tf:-0.001:ti,S,opt)
t=flipud(t)
P=flipud(P)
```

As seen in the second parameter, the time vector is created from the end time towards the beginning with the step of 1 ms. Another parameter represents the initial conditions obtained in the previous step by solving the algebraic Riccati equation. The 'ode45' function requires returning of the results from the individual steps of the solution in the form of column vectors, not square matrices. Therefore, the initial conditions are rearranged from a 4×4 matrix to a 16-element column vector. As a result, the time vector t and the vector with the solutions at the given time P are obtained. Since both of them are oriented backwards in time (t goes from the end time to zero), they have to be reverted for further use.

The first parameter of 'ode45' is a reference to the "riccatiEquation" function in which the differential Riccati equation is formulated. It also contains matrices of the state description of the system to be controlled. These are obtained by linearization of the inverted pendulum description, valid for a certain general operating point. For the specified time t, the values of the state variables are obtained from the trajectory, along which the feedback control process has to be performed, and a linearization is computed in this particular operating point subsequently.

```
function dP = riccatiEquation(t,P)
[A,B]=getlinsys(t);
global Q R
Z=reshape(P,4,4); % rear. into 4 × 4 matrix
dZ=-(Z*A+A'*Z-Z*B*inv(R)*B'*Z+Q); % solut.
dP=reshape(dZ,16,1); % rear. to col. vect.
end
```

In the Riccati equation itself the solution P is obtained (from the previous step), which was modified in the initial conditions to a column vector. In order to solve the equation and to fit the matrix dimensions, it is necessary to rearrange it into the 4×4 square matrix and, after solving the equation, rearrange the solution back to the column vector.

At this point there is a solution of differential Riccati equation for the entire used trajectory with 1 ms time step. Another part of the procedure is to calculate the state controller for each point of the solution of the differential Riccati equation according to equation (11).

```
for k=1:length(t)
[A,B]=getlinsys(t(k));
Ptmp=reshape(P(k,:),4,4)
K(k,:)=-inv(R)*B'*Ptmp;
end
```

This process is performed in a cycle for all time points, at which the entire control process was divided up. The solution that is used to calculate the controller has again to be converted to a 4×4 matrix to maintain the correct dimensions. The procedure for obtaining a linearized description of the controlled system at the required operating point is the same as for the differential Riccati equations. The result of this operation is a matrix with four columns, in which the values for the four components of the time-varying controller are stored to provide the feedback control to the system along the reference trajectories.

```
function [A,B]=getlinsys(t)
global u_opt x1_opt x2_opt
uopt_current=interp1(u_opt.Time,...
u_opt.Data,t); %reference control (u)
x1opt_current=interp1(x1_opt.Time,...
x1_opt.Data,t); %reference trajectory (x1)
x2opt_current=interp1(x2_opt.Time,...
x2_opt.Data,t); %reference trajectory (x2)
l=0.15;b=0.07;g=10;
%A,B: Jacobi matrices
A=[0 1 0 0;
3/4*g/l*cos(x1opt_current)-...
3/4*1/l*uopt_current*sin(x1opt_current)
-b 0 0;
0 0 0 1;
0 0 0 0];
B=[0;3/4*1/l*cos(x1opt_current);0;1];
```

Optimization procedure for Q and R weight determination

One of the innovations of the present paper lies in the way, in which we choose the weight matrices Q and R. We designed an optimization procedure for this purpose. It acts as an extension to the described algorithm for finite-horizon LQR computation. The optimization changes the main diagonal elements of matrix Q and the elements of matrix R based on a cost function. In one iteration the time-varying controller $K(t)$ is calculated based on actual weight matrices Q and R. The controller is then used in MIL (model in the loop) simulation to control the system along the reference state trajectories. The cost function J, given in (23), is composed of sums of square differences between the simulated state values and the reference trajectories. The energy of the control signal $u(t)$ is also added to the cost function with small weight to improve the selection of the weight matrices:

$$J = \sum_{i=1}^{4} \sum_{k=0}^{N} \left(x_i \left[k \right] - x_{i-ref} \left[k \right] \right)^2 + \sum_{k=0}^{N} \left(u \left[k \right] \right)^2 \tag{23}$$

where N is number of signals from simulation samples.

We implement the optimization procedure in Matlab with the use of 'fmincon' function in GlobalSearch form. Lower bounds for the optimized parameters

were selected on the basis of the fact that weight matrices must be positive, upper bounds were selected as several times higher values than the initial setting of weight matrices, described before.

We tested the described solution in two cases. In the first case, all five parameters were optimized, while in the second case the value of R was fixed to 1 and four remaining parameters were optimized. The results are shown in Table 1 together with the initial setting and relevant cost function.

Table 1. Weight optimization procedure results

variant	q_{11}	q_{22}	q_{33}	q_{44}	R	J
Initial	10.0	10.0	50.0	180.0	1.0	67.50
Optimization	78.7	98.7	417.6	364.3	0.12	38.83
With fixed R	71.8	97.8	342.8	91.3	1.0	41.83

6. Example of application

To show the calculation of the controller for the solution of the above-mentioned problem, it is first necessary to have a reference trajectory and a reference control signal. We call this problem the trajectory generation problem. Generally, the framework for this problems is provided by the methods of solving the general optimal control problem, either with Mayer term only, or with both Mayer and Lagrange terms, according to what kind of cost function it is desired to minimize. It can be minimizing of the final state only, or adding a penalty on the trajectories along the path. In case of minimizing the final state only, this state-transition problem can be effectively solved as a TPBVP (two point boundary value problem) with free parameters. However, the analysis of this issue is beyond the scope of the article. The issue of the TPBVP, related to the inverted pendulum has been thoroughly analyzed in Ozana and Schlegel (2018). Apart from this, a lot of tools are available for the trajectory generation problem. In this article, the PyTrajectory tool was used to design the trajectories for transition between the states for non-linear systems. After formulation of the dynamics of the controlled system, containing constraints at the beginning and the end of the solution interval, the reference trajectories and the control signal shown in Figs. 7 and 8 were obtained.

The PyTrajectory tool provides solution with the time step of 10 ms. For the purpose of calculating the time-varying LQR controller, the obtained waveforms were interpolated to obtain signal values with 1 ms time step. They are used in the "getlinsys" function, which returns matrices A and B of the state descrip-

tion of a linearized system at a particular point of the trajectory, according to the time point specified as an input parameter.

Using the described algorithm, the time-varying controller with four components was designed for the reference trajectory so that $K(t) = [K_1(t), K_2(t), K_3(t), K_4(t)]$.

Verification of the proposed feedback control for the given reference trajectories and reference control signal was first carried out by the MIL simulation, performed in Simulink environment, while the simple inverted pendulum system has been modelled according to state-space description (19-22). The simulation contains two parts, both for open-loop and closed-loop control. Firstly, only reference signal enters the system. Theoretically, the same waveforms as the reference trajectories should be obtained from the state variables of this model. The second part of simulation experiment is connected to the 2-DOF control structure with the feedback term representing the calculated controller $K(t)$. After the swing-up, that is, for a time bigger than the horizon, for which the LQR controller was calculated, the last values (steady-state) of the gains apply in the feedback, corresponding to the infinite-horizon LQR controller. The whole simulation has been performed with the fixed 1 ms time step. Since the primary indicator of a success with the swing-up is the waveform of the pendulum position, Fig. 9 compares this value (x_1) from both parts of the simulation experiment.

It is obvious that open loop control does not meet the basic requirement for the transition between the two states. The pendulum deviates from the reference trajectory during the elevation despite the fact that the feed-forward control was calculated for the same model as that used in the simulation.

The problem is caused by several facts. Firstly, generation of reference trajectory with 10 ms time step may be insufficient in terms of accuracy. Between the respective particular values, the used values are interpolated because the simulation works with 1 ms increments. Secondly, simulation experiment uses some numerical method, so the found solution is always inaccurate compared to what would be obtained in case of analytical solution (if possible to compute). There is also some inaccuracy, given by the final resolution of the numerical values, with which the computer can work. These deviations gradually accumulate, and in case of highly nonlinear and unstable systems, like here, this results in significant violation of constraints given at the final time.

When looking at the waveform, representing the pendulum position from the closed-loop part of the model, it becomes clear at the first glance that the

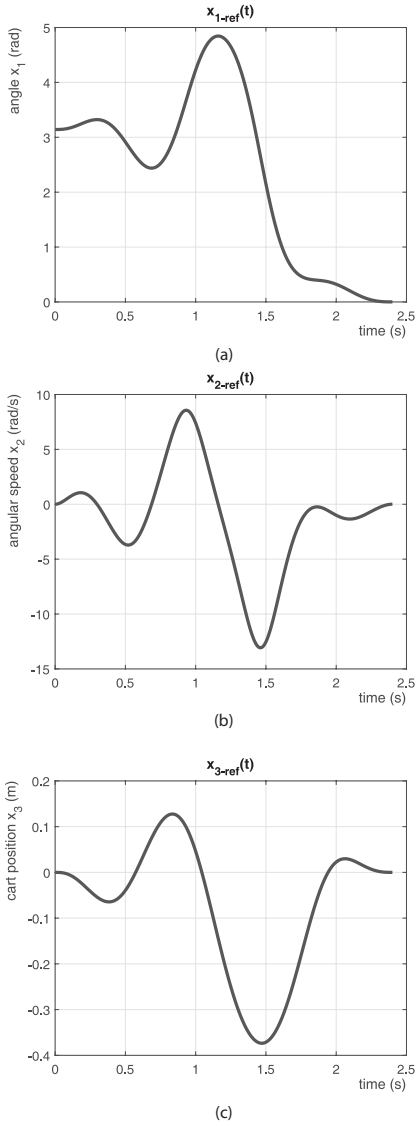

Figure 7. Reference trajectories and control signal: (a) pendulum angle, (b) pendulum angular speed, (c) cart position

transition condition between states is fulfilled. Only this closed-loop, containing time-varying LQR controller, will be further discussed. A more detailed

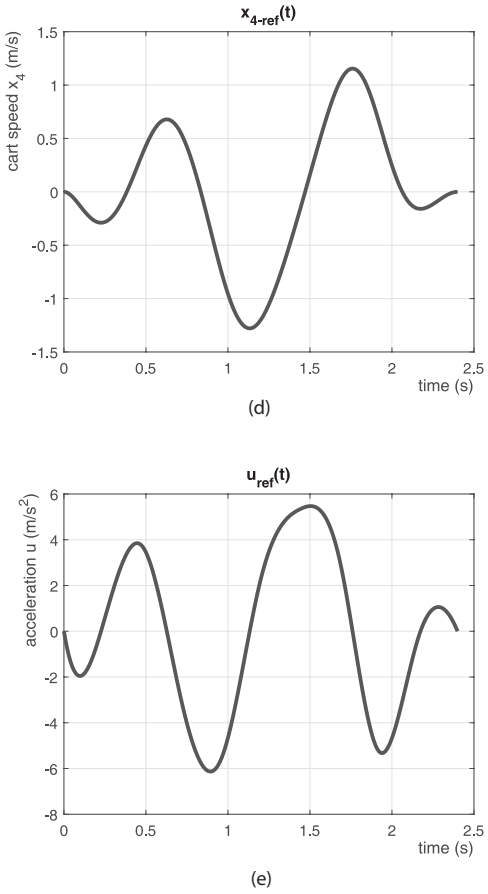

Figure 8. Reference trajectories and control signal: (d) cart speed, (e) cart acceleration

overview of the situation is given in Fig. 10(a), which compares the reference trajectory of the pendulum angle, i.e. the $x_{1ref}(t)$ signal, and the corresponding signal obtained from the simulated model $x_{1sim}(t)$ and real output $x_{1real}(t)$. Differences between these patterns are really small. In this case, the fact that the non-linear system has been replaced by a linear time-varying system (LTV), which keeps changing every 1 ms, has a certain influence on the calculation of the time-variable controller. However, at this time point the controlled system is considered as linear, this causing a deviation from the previously described non-linear model (equations (19-22)), which is cumulative again, too. Supposing an infinitely small period, it would be possible to eliminate this inaccuracy, but,

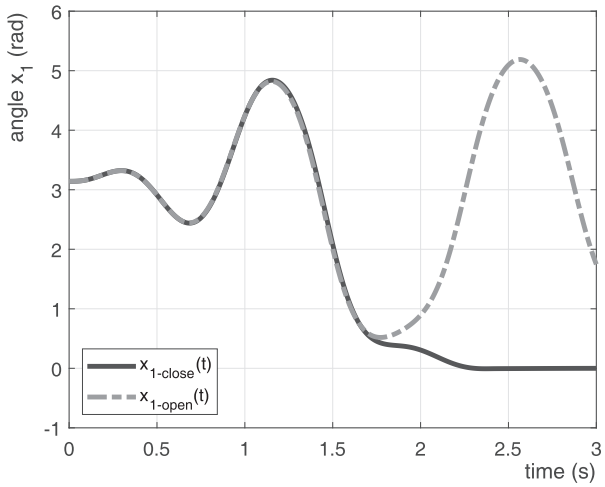

Figure 9. Comparison of the pendulum angle responses from the open and closed loop model

of course, such a situation is hypothetical. The used value of 1 ms time step is sufficiently small to demonstrate reliable functionality. The resulting deviations are then compensated by the feedback controller as shown in Fig. 10 (b). The difference is especially evident in about 1.5 seconds when the pendulum arm passes through the horizontal position (the angle is $\pi/2$ rad). This is a singularity, where it is not possible to affect the states of the system by any value of the control signal, and in its neighborhood it is possible, but it is very difficult.

7. Discussion

The real physical apparatus, representing a single pendulum on the cart is controlled by the cart speed (inducing its acceleration) instead of the acceleration signal itself, however, the control design and computation of the trajectories can be performed for the acceleration control input without being affected by this assumption. The relationship and the context regarding this issue are briefly introduced in Ozana and Docekal (2017) and explained in detail within our paper.

Both trajectory generation and LQR design fall into the framework of the optimal control problems, but each time these issues are handled differently, once as a feed-forward optimal control problem, for the second time as a feedback control problem.

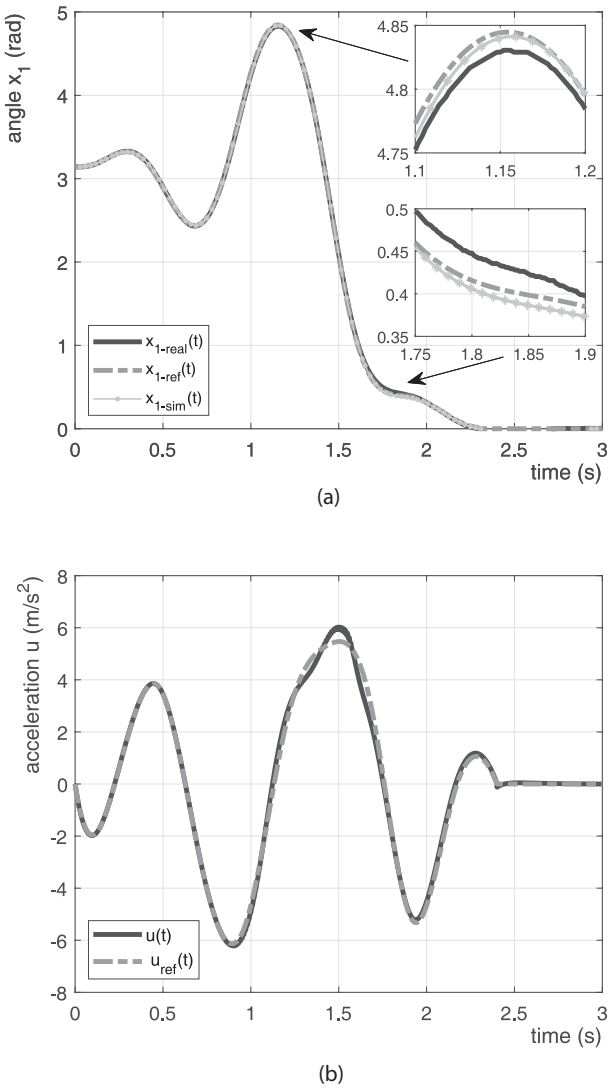

Figure 10. Comparison of signals: (a) real, reference, and simulated signal $x_1(t)$, (b) reference and simulated signal $u(t)$

Apart from Ozana and Schlegel (2018) and other similar proposals for solving the trajectory generation problem, there are many different libraries and toolboxes for different software environments and languages.

Thus, PyTrajectory (Kunze, 2016), Dynopt (Cizniar, Fikar and Latifi, 2017) and OptimTraj (Kelly, 2018) are three examples of very advanced tools with efficient and very promising results, mainly due to the capability of solving a general optimal control problems including constraints, and all of them have been tried for a single inverted pendulum. It is planned to use them for further work regarding double and triple inverted pendulums.

The proposed solution is based on Matlab, but it can be adopted to a different software environment – the crucial and most challenging part of the procedure would be the adoption of the *care* function.

For the trajectory planning in this paper, we chose PyTrajectory which does not allow the time step below 10ms. With this time periodicity, the computed reference trajectories can be considered as worst-case, yet the time-varying state controller is capable of dealing with the considered situation. Of course, it is always possible and recommended to reduce the period as much as possible with the use of a different trajectory generator. From the technical point of view, deviation between open-loop and closed-loop waveforms will always occur, no matter how precisely the numerical algorithms are set up, but this is efficiently handled by the feedback controller. With appropriate control design within the 2-DOF concept, feasibility of the solution is never broken.

Acknowledgements

This research was funded by the European Regional Development Fund in the Research Centre of Advanced Mechatronic Systems project, grant number CZ.02.1.01/0.0/0.0/16_019/0000867 within the Operational Programme Research, Development and Education.

This work was also supported by the project SP2020/42,"Development of algorithms and systems for control, measurement and safety applications VI" of Student Grant System, VSB-TU Ostrava.

References

Cizniar, M., Fikar, M. and Latifi, A. M. (2017) MATLAB Dynamic Optimisation Code DYNOPT. User's Guide, Technical Report, KIRP FCHPT STU, Bratislava, Slovak Republic, https://bitbucket.org/dynopt/dynopt_code_hg/ src/default/dynopt_guide.pdf

Durand, S., Castellanos, F. G., Marchand, N. and Sanchez, W. F. G. (2013) Event-based control of the inverted pendulum: Swing up and stabilization. *Journal of Control Engineering and Applied Informatics*, **15**(3), 96.

Hercus, R., Wong, K.-Y., Shee, S.-K. and Ho, K.-F. (2013) Control of an inverted pendulum using the neurabase network model. In: M. Lee, A. Hirose, Z.-G. Hou, & R. M. Kil, eds., *Neural Information Processing*. Springer, 605-615. https://doi.org/10.1007/978-3-642-42042-9_75

Ichtev, A. K. (2008) Fuzzy controller for swing-up and stabilization of inverted pendulum. *2008 4th International IEEE Conference Intelligent Systems*, 1, 2-57-2–62. https://doi.org/10.1109/IS.2008.4670408

Kelly, M. P. (2018) OptimTraj Users Guide. User's Guide, https://github.com/MatthewPeterKelly/OptimTraj/blob/master/docs /UsersGuide/ OptimTraj_UsersGuide.pdf

Kennedy, D. and Conlon, J. (2011) Inverted pendulum swing up controller. Conference Papers. *Mechanical Technologies and Structural Materials 2011*, Split, Croatia. https://arrow.tudublin.ie/engschmeccon/47

Kunze, A. (2016) Pytrajectory's documentation. User's Guide, https://pytrajectory.readthedocs.io/en/master/

Ozana, S. (2018a) *Swing-up and Control of Linear Simple Inverted Pendulum* [Video]. https://youtu.be/Sqhr8fYhMfg

Ozana, S. (2018b) *Swing-up and Control of Linear Triple Inverted Pendulum* [Video]. https://youtu.be/meMWfva-Jio

Ozana, S. and Docekal, T. (2017) The concept of virtual laboratory and PIL modeling with REX control system. *2017 21st International Conference on Process Control (PC)*, 98–103. https://doi.org/10.1109/PC.2017.7976196

Ozana, S., Pies, M. and Hajovsky, R. (2014) Computation of swing-up signal for inverted pendulum using dynamic optimization. In: K. Saeed & V. Snasel, eds., *Computer Information Systems and Industrial Management*. Springer, 301-314. https://doi.org/10.1007/978-3-662-45237-0_29

Ozana, S. and Schlegel, M. (2018) Computation of reference trajectories for inverted pendulum with the use of two-point bvp with free parameters. *IFAC-PapersOnLine,* **51**(6), 408–413. https://doi.org/10.1016/j. ifacol.2018.07.119

Tum, M., Gyeong, G., Park, J. H. and Lee, Y. S. (2014) Swing-up control of a single inverted pendulum on a cart with input and output constraints. *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO), 01*, 475–482. https://doi.org/10.5220/0005018604750482

Yang, X. and Zheng, X. (2018) Swing-up and stabilization control design for an underactuated rotary inverted pendulum system: Theory and experiments. *IEEE Transactions on Industrial Electronics,* **65**(9), 7229–7238. https://doi.org/10.1109/TIE.2018.2793214

Yokoyama, J., Mihara, K., Suemitsu, H. and Matsuo, T. (2011) Swing-up control of an inverted pendulum by two step control strategy. *2011*

IEEE/SICE International Symposium on System Integration (SII), 1061–1066. https://doi.org/10.1109/SII.2011.6147596