

# Computation of Reference Trajectories for Inverted Pendulum with the Use of Two-point BvP with Free Parameters

S. Ozana\* M. Schlegel\*\*

\* *Department of Cybernetics and Biomedical Engineering, Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, 17.listopadu 15, Ostrava, Czech Republic (e-mail: stepan.ozana@vsb.cz)*

\*\* *Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia in Pilsen, Technicka 8, Plzen, Czech Republic (e-mail: schlegel@kky.zcu.cz)*

**Abstract:** This paper deals with the computation of reference trajectories for inverted pendulum model to be used in two-degree of freedom control scheme. The proposed method uses a special customized candidate function for control signal  $u(t)$  involved in formulation of this case study as Two-point Boundary Value Problem with free parameters (TPBVP).

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Boundary value problem; automatic control; Trajectory planning; State trajectories; State-space model; Control system design; TPBVP

## 1. INTRODUCTION

As the problem of controlling inverted pendulum is an attractive, interesting and complex issue, it has been discussed much more frequently and intensively than any other common educational models. Despite this fact, new results in the area might still rise, as it can still be seen in the conferences around the world. There are many methods and approaches for control of the inverted pendulum in the upright position and for the swing-up, involving conventional and non-conventional controllers. The most common way to handle both swing-up and controlling in the upright position, is performing an open-loop control for the swing-up, then switching to the closed-loop control once the pendulum is close to the upright position, see Jadlovská and Sarnovsky (2013). However, this approach will very likely fail in case of double or triple pendulum due to the complexity and high sensitivity of such systems to various disturbance signals. The proposed computational technique described in this paper is suitable for the only one common closed-loop involving both swing-up and control in the upright position, using a time-varying LQR controller  $K(t)$  computed on a finite horizon. However, the design of LQR controller is beyond the scope of the paper, and it is focused on computation of the reference control signal and reference state trajectories used in so called two-degree of freedom control scheme. It is obvious that the reference control signal and reference state trajectories can be computed with the use of many approaches and principles. Basically, some cost function  $J(t)$  might be defined, or there is no cost function defined. The computational techniques for the first case consider some criteria for the control signal and reference trajectories, typically minimum-time or minimum-energy requirements, see Sha-

hab et al. (2017) or Bolotin and Kozlov (2015). It is then the optimal control problem. The latter case is discussed in this paper, using a boundary value problem (BvP) with free parameters. The solution lies in definition of a suitable so called candidate function and computation of BvP problem in Matlab.

## 2. MATHEMATICAL MODEL OF INVERTED PENDULUM

The situation scheme used for identification of the system is introduced in Fig. 1. Differential equation describing movement of the inverted pendulum on the cart without a friction can be described by (1).

$$I \cdot \ddot{\varphi} - m \cdot g \cdot l \cdot \sin \varphi = m \cdot l \cdot \ddot{x} \cdot \cos \varphi \quad (1)$$

where  $I = J + m \cdot l^2$ ,  $J$  represents moment of inertia with respect to the center of mass,  $l$  is the distance from the pivot P to the center of mass. This paper considers mass of the rod concentrated in the single mass point M, thus  $l = |MP|$ ,  $J = 0$  and  $I = m \cdot l^2$ . Other typical case might be homogeneous cylindrical rod of the length  $L$  ( $l = \frac{1}{2}L$ ) where  $J = \frac{1}{12}m \cdot L^2$  and  $I = \frac{1}{12}m \cdot L^2 + m \cdot l^2 = \frac{1}{3}m \cdot L^2$ . Gravity constant is marked by  $g$ . Considering dumping coefficient  $b$  reflecting a friction, the (1) turns into (2):

$$\ddot{\varphi} - \frac{g}{l} \cdot \sin \varphi + \frac{b}{l} \cdot \dot{\varphi} = \frac{1}{l} \cdot \ddot{x} \cdot \cos \varphi \quad (2)$$

Choice of state and input variables:

- $x_1(t)$ ...pendulum angle [rad]
- $x_2(t)$ ...pendulum angular speed [rad · s<sup>-1</sup>]
- $x_3(t)$ ...cart position [m]
- $x_4(t)$ ...cart speed [m · s<sup>-1</sup>]

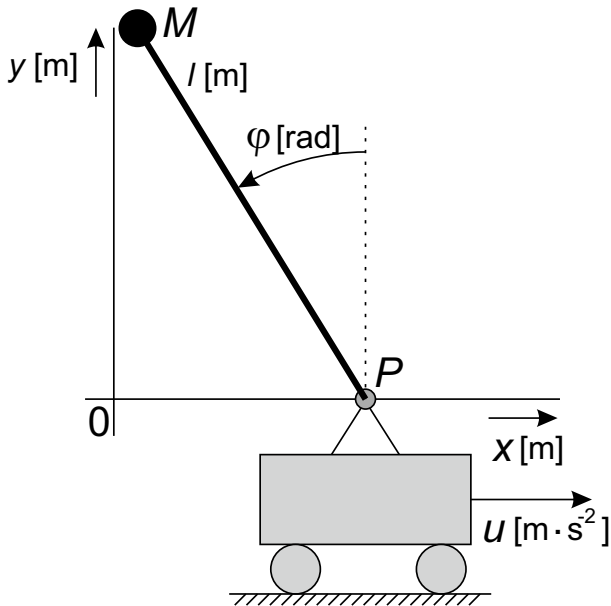


Fig. 1. Situation scheme for identification of the system  
 $u(t)$ ...cart acceleration [ $m \cdot s^{-2}$ ]

Note: The real model 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 is given in (OZANA and DOCEKAL (2017)).

Using the above mentioned notation, state-space model can be defined by (3), (4), (5), (6):

$$\dot{x}_1 = x_2 \quad (3)$$

$$\dot{x}_2 = \frac{g}{l} \cdot \sin x_1 - \frac{b}{l} \cdot x_2 + \frac{1}{l} \cdot u \cdot \cos x_1 \quad (4)$$

$$\dot{x}_3 = x_4 \quad (5)$$

$$\dot{x}_4 = u \quad (6)$$

### 3. TWO-DOF CLOSED-LOOP CONTROL SCHEME

A large class of control problems consist of planning and following a trajectory in the presence of noise and uncertainty, including inverted pendulum, see Boscaroli and Richiedei (2018). To control such systems, we make use of the notion of two degree of freedom controller design. This is a standard technique in linear control theory that separates a controller into a feed-forward compensator and a feedback compensator. The feed-forward compensator generates the nominal input (reference control signal) required to track given reference trajectories (outputs of TG-trajectory generator). The feedback compensator corrects for errors between the desired and actual trajectories, see MURRAY (2004). This is shown schematically in Fig. 2. The proposed algorithm focuses on computation of  $u^*(t)$  and  $x^*(t)$  based on BvP with free parameters. The crucial requirement for this pair is to follow the differential equation describing the dynamics of the system.

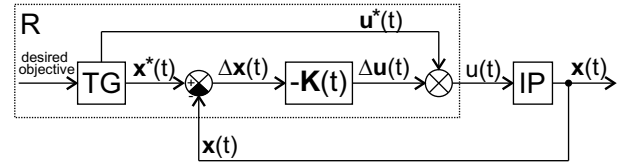


Fig. 2. Two degree of freedom control scheme

### 4. FORMULATION OF THE PROBLEM AS TWO-POINT BVP WITH FREE PARAMETERS

The formulation of the job in this section is based on state-space model of the inverted pendulum. Moreover, there are following constraints to be followed in initial state ( $t=0$ ) and final state ( $t=T$ ): position and speed of the rod, position and speed of the cart. Altogether there are eight constraints described by (7), (8), (9), (10):

$$x_1(0) = \pi, \quad x_1(T) = 0 \quad (7)$$

$$x_2(0) = 0, \quad x_2(T) = 0 \quad (8)$$

$$x_3(0) = 0, \quad x_3(T) = 0 \quad (9)$$

$$x_4(0) = 0, \quad x_4(T) = 0 \quad (10)$$

As the number of constrained is higher than number of states, the BvP is so called overdetermined, see Graichen and Zeitz (2008), and there should be four free parameters that makes it possible to find a solution of BvP problem. Below mentioned procedure explains why there are five parameters considered at the beginning, then reduced to four. At the beginning, so called candidate function must be defined. It can be a series of polynomials, splines, harmonic wave or other function, based on expert's choice. In this paper, the choice is based on simple physical assumptions reflecting some properties regarding cart acceleration  $u(t)$ , its speed  $v(t)$  and its position  $s(t)$ . It is highly reasonable to require zero initial and final values for all of these physical variables. Turned into physical representation speech, both at the beginning and at the end of experiment the cart is motionless, during the experiment it moves forth and back to the original position. A candidate function for  $u(t)$  (see (11) in the form of series of sinus waves is therefore a suitable candidate. Moreover, integration to obtain speed  $v(t)$  (see (12)) preserves the initial and final constraints. Constraints for position  $s(t)$  (see (13)) are secured separately.

$$u(t) = \sum_{k=1}^5 \lambda_k \cdot \sin(k\omega t) \quad (11)$$

$$v(t) = \int_0^t u(\tau) d\tau = \sum_{k=1}^5 \frac{\lambda_k}{k\omega} \cdot [1 - \cos(k\omega t)] \quad (12)$$

$$s(t) = \int_0^t v(\tau) d\tau = \sum_{k=1}^5 \frac{-1}{k^2\omega^2} \lambda_k \cdot [\sin(k\omega t) - k\omega t] \quad (13)$$

Using a time substitution within (11) and (12), it can be easily verified that boundary constraints for acceleration and speed are met for  $t = 0$  and  $t = T$ , together with position for  $t = 0$ :

$$u(0) = u(T) = v(0) = v(T) = s(0) = 0$$

Substitution  $t = T$  into (13) shows that  $s(T) \neq 0$ , see (14):

$$s(T) = \frac{T^2(7, 2\lambda_1 + 3, 6\lambda_2 + 2, 4\lambda_3 + 1, 8\lambda_4 + 1, 44\lambda_5)}{14, 4\pi} \quad (14)$$

Solution of (14) under condition  $s(T) = 0$  leads to a formula for  $\lambda_5$ , see (15).

$$\lambda_5 = -5\lambda_1 - \frac{5}{2}\lambda_2 - \frac{5}{3}\lambda_3 - \frac{5}{4}\lambda_4 \quad (15)$$

Introduction of the fifth parameter  $\lambda_5$  made it possible to assure zero boundary constraint for cart position at the final time  $t = T$ . Now all of the boundary constraints are met:

$$u(0) = u(T) = v(0) = v(T) = s(0) = s(T) = 0$$

Therefore, final form of the candidate function that depends on four free parameters is in the form of (16):

$$u(t) = \lambda_1 \sin(\omega t) + \lambda_2 \sin(2\omega t) + \lambda_3 \sin(3\omega t) + \lambda_4 \sin(4\omega t) + (-5\lambda_1 - \frac{5}{2}\lambda_2 - \frac{5}{3}\lambda_3 - \frac{5}{4}\lambda_4) \sin(5\omega t) \quad (16)$$

The problem of finding a reference control signal  $u(t)$  and reference state trajectories  $\mathbf{x}(t)$  is now transformed into a problem of finding coefficients  $\lambda_1, \lambda_2, \lambda_3, \lambda_4$  that, being substituted into candidate function, transfers the inverted pendulum from the bottom into the upright position, thus following the dynamics of the system and possibly other requirements for maximal magnitudes of particular state variables or other conditions placed on the trajectories.

## 5. IMPLEMENTATION OF COMPUTATIONAL ALGORITHM AND SIMULATIONS

This section focuses on solution of BvP with free parameters with the use of Matlab support, especially via `bvp4c` function. The section describes construction of Matlab script with some commentary to clear out all steps of computation. This is the source code used to compute key simulation results in this paper:

```
function sol=void()
close all,clear all
%A
lam1 = 0.1;lam2 = 0.075;lam3=0.1;lam4=0.25;
%B
% lam1 = 0.1;lam2 = 0.075;lam3=0.125;lam4=0.35;
lam=[lam1 lam2 lam3 lam4];
t_opt=linspace(0,4452*0.001,4452)';
solinit = bvpinit(t_opt,@xinit_fcn,lam);
sol = bvp4c(@system_odeset_fcn,@bc_function,solinit)
disp('lam_values:'),sol.parameters
figure,plot(sol.x,sol.y),grid on
legend('x1','x2','x3','x4'),title('x(t)')
u_star=candidate_fnc(sol.x,sol.parameters');
figure,plot(t_opt,u_star),grid on,title('u(t)')
% -----
function u=candidate_fnc(t,lam)
Tf=4.4520;w=2*pi/Tf;
u1=lam(1)*sin(w*t);u2=lam(2)*sin(2*w*t);
u3=lam(3)*sin(3*w*t);u4=lam(4)*sin(4*w*t);
lam5=-5*lam(1) -5/2*lam(2)-5/3*lam(3)-5/4*lam(4);
u5=lam5*sin(5*w*t);
u=u1+u2+u3+u4+u5;
% -----
```

```
function dxdt = system_odeset_fcn(t,x,lam)
g=10;b=0.0225;l=0.15;u_candidate_fnc=candidate_fnc(t,lam);
dxdt = [x(2)
g/l*sin(x(1))-b/l*x(2)+1/l*u_candidate_fnc*cos(x(1))
x(4)
u_candidate_fnc];
% -----
function res = bc_function(ya,yb,lambda)
res = [ya(1)-pi;ya(2);ya(3);ya(4);yb(1);yb(2);yb(3);yb(4)];
% -----
function xinit = xinit_fcn(t)
load xinit.dat; %xinit=[x1_opt;x2_opt;x3_opt;x4_opt];
```

The script consists of five main blocks represented by Matlab functions.

### 5.1 Main body of the script- `void`

This function starts with definition of the initial values of  $\lambda$  coefficients for two datasets marked as "A" and "B", see Tab. (1). Together with time vector and initial guess of  $\mathbf{x}(t)$  it uses  $\lambda$  vector to create initial solution stored into `solinit` variable. Then it is possible to call `bvp4c` function, having three inputs: previously defined initial solution, system dynamics and boundary constraints formulated in a special syntax. It is obvious that newly found state trajectories are stored in default `sol` variable representing a record with predefined inner variables  $x$  and  $y$ . Then `sol.x` is a time vector bound for solution, and `sol.y` represents vector of state trajectories themselves. Newly found  $\lambda$  coefficients are stored in `sol.parameters` and substituted into candidate function. All main graphs are plotted here.

### 5.2 Candidate function - `candidate_fnc`

This function defines candidate function for acceleration of the cart as the input to the system. It consists of five components and depends on four free parameters.

### 5.3 System dynamics function - `system_odeset_fnc`

Apart from parameters of the inverted pendulum system ( $g, b, l$ ), this function follows the dynamics of inverted pendulum described by a set of differential equations (3), (4), (5), (6). As it works with state vector, for particular state variables it uses right sides of state equations corresponding to  $x_1$  to  $x_4$ .

### 5.4 Boundary constraints function - `bc_fnc`

This function contains information about all boundary constraints in predefined syntax. The symbol  $ya$  represents the beginning of the time interval while  $yb$  represents the end of this interval. The constraints are written down in the orders according the meaning of particular state variables. For example, the boundary constraint  $x_1(0) = \pi$  written in correct format becomes `ya(1)-pi`. All of the other boundary constraints are zero, therefore `ya(2)` to `ya(4)` and `yb(1)` to `yb(4)` follow the rest of the lines.

### 5.5 Initial guess function - `xinit_fnc`

This function defines an initial guess of  $\mathbf{x}(t)$ . Its importance arises with level of nonlinear character of the system.

It is important for convergence. For different initial guesses different  $\lambda$  and thus different trajectories may be found.

5.6 Simulation results

Two different datasets for initial  $\lambda$  coefficients have been used to demonstrate the algorithm, see Table 1. Once the solution of BvP with free parameters converged to the solution, the script returned new resulting  $\lambda$  values. Fig. 3 demonstrates the reference control signals for A and B datasets. The reference control trajectories  $x_1, x_2, x_3, x_4$  are displayed on Fig. 4, Fig. 5, Fig. 6, Fig. 7. Both case studies included the same initial guess of  $x(t)$  and same system parameters.

Table 1. Simulation inputs and results

	A		B	
	Initial value	Value found	Initial value	Value found
$\lambda_1$	0,1	1,2583	0,1	1.2706
$\lambda_2$	0,075	-2,5412	0,075	-1.7642
$\lambda_3$	0,1	0,0363	0,125	1.2754
$\lambda_4$	0,25	0,2548	0,35	-6.0983

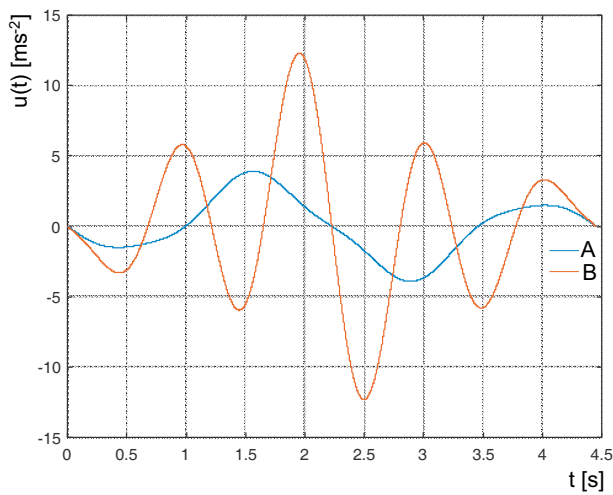


Fig. 3. Reference control signal  $u(t)$  - acceleration of the cart

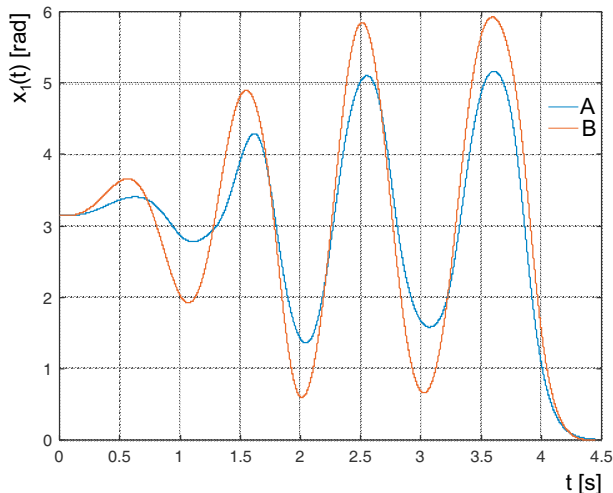


Fig. 4. Reference trajectory  $x_1(t)$  - position of the pendulum rod

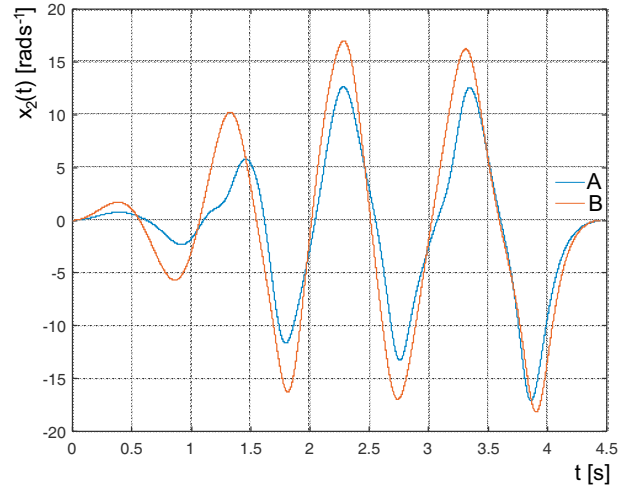


Fig. 5. Reference trajectory  $x_2(t)$  - angular speed of the pendulum rod

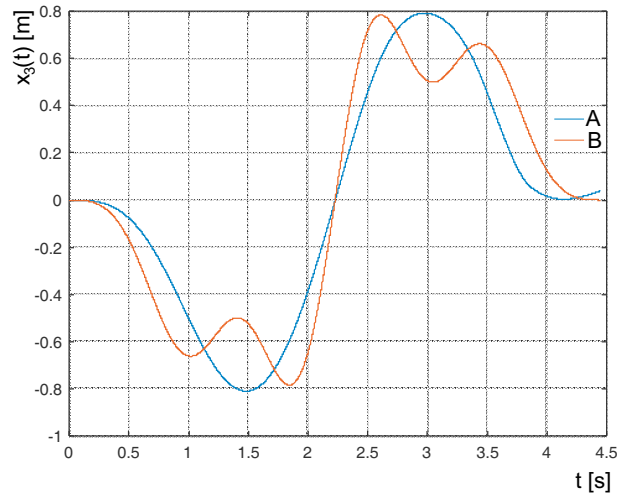


Fig. 6. Reference trajectory  $x_3(t)$  - position of the cart

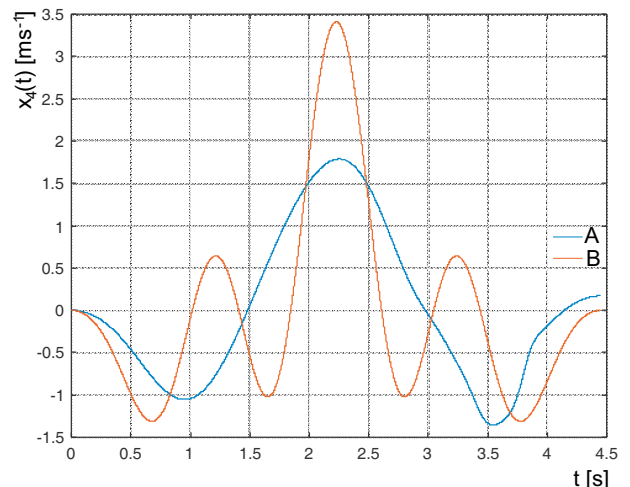


Fig. 7. Reference trajectory  $x_4(t)$  - speed of the cart

## 6. CONCLUSION

The paper demonstrates case study of trajectory planning for a simple inverted pendulum on the cart with the use of BvP with free parameters, see Graichen and Zeitz (2005) and Graichen and Zeitz (2008). As there was no cost function considered for computation of reference state trajectories, the found solution is not therefore optimal in any technical sense. An example of solution of trajectory planning problem formulated as optimal problem is described in Ozana et al. (2014), considering minimization of time needed for the swing-up. This solution and similar solutions (for example minimum-energy problem) assure minimal value of cost function, but there are no other consequences for any state variables, especially for minimal and maximal magnitudes of these variables, see Tum et al. (2014). These considerations are much more important in practice due to physical limits given by a particular hardware setup, actuators and other components of control systems. Main crucial issues that should be respected are as follows:

- range of  $x_1$  should fall into  $\langle 0, 2\pi \rangle$
- minimal and maximal value of  $x_3$  due to the limitation of cart railway length
- minimal and maximal value of  $x_4$  due to the limitation of actuator used for the motor

Cart railway which is at disposal under real conditions may be one of the most important construction parameters for inverted pendulum physical setup. Under certain circumstances there may be no trajectories found for a given range where cart is supposed to move. This value strongly depends on the length of the pendulum rod, and a certain minimal range for the cart position always exists.

These requirements are not taken into consideration in current computation algorithm, but it is planned to do so in its future versions. As the analytical formulas representing acceleration, speed and position of the cart are known, see (11), see (12), see (13), new requirements can be placed for the range of these signals.

Considering acceleration as the input to the system both for control design and computation of reference trajectories supposes ideal speed controller. In other words, limitation of actuator used for the DC motor is not considered, it is presumed to be fast enough. If this assumption is questionable, it is also possible to include the dynamics of inner speed loop into the control design and into the computation of state reference trajectories. In case of ideal speed controller  $G_{Rv}(s) = 1$  and  $u = u_c$ , see Fig. 8.

Two case studies using different datasets defining four free  $\lambda$  parameters have been presented. Both case studies used the same initial guess  $\mathbf{x}(t)$  as displayed in Fig. 9. It also shows corresponding control signal that was used to get this initial guess. It is obvious that initial control signal  $u(t)$  reaches higher magnitudes than the one found in "A" case study. As a consequence of this, newly found  $x_1$  shows more oscillatory character than the initial state trajectory. Moreover, maximal position of the pendulum rod, see signal  $x_1$  in Fig. 4 for "A" case study is quite close to  $2\pi$ . The resulting feasible control reference signal and reference state trajectories are based on a certain compromise between various contradictory requirements.

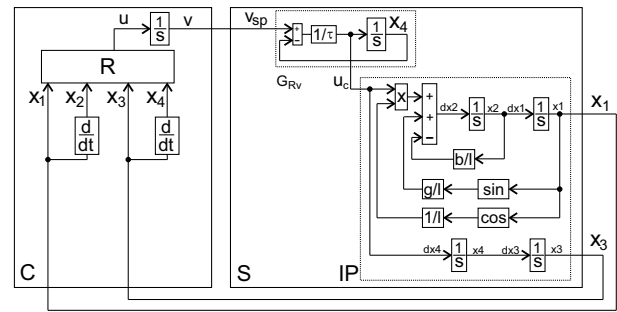


Fig. 8. Consideration of ideal speed control in control scheme

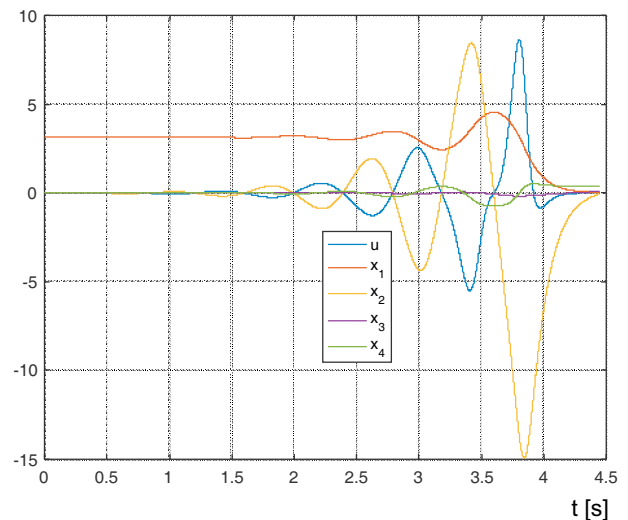


Fig. 9. Initial guess of  $\mathbf{x}(t)$  and corresponding reference signal  $u(t)$

The similar approach to this paper is described in Yu and Wang (2011), considering open-loop swing-up solved by BvP for rotary inverted pendulum model. The proposed approach can be adopted for double or triple pendulums, see Zhang et al. (2011) or Gluck et al. (2013), and also possible to use for other mechanical systems, see He and Geng (2008).

Also, there exists a professional powerful toolbox written in Python, called PyTrajectory, see Kunze (2005). It is capable of handling very complex nonlinear dynamic systems including boundary constraints and limits for particular state variables, using polynomial candidate functions.

The functionality of newly designed control reference signal  $u(t)$  and reference state trajectories  $\mathbf{x}(t)$  has been approved both in MIL and SIL simulations, using Matlab&Simulink and REX Control System, see Balda et al. (2005).

The same REX Control System will be used to implement the control algorithm using varying-time LQR controller and proposed reference control signal and reference state trajectories on a real setup. The implementation of this 2-DOF control scheme has been already previously successfully performed with the use of REX with Raspberry Pi and Arduino or STM32f4 board for reference control signal and trajectories designed by different approaches, including above mentioned PyTrajectory toolbox.

The more generalized goal of this paper is to point out that the proposed idea of trajectory planning can be extended for using for similar nonlinear systems, namely mechanic or mechatronic systems, underactuated and balancing systems, etc. One of the future plans in the area of trajectory planning is application of this and similar algorithms for a triple inverted pendulum and a quadcopter. The detail discussion analysis of particular  $\lambda_k$  parameters, chosen final time  $T$ , form of candidate function  $u(t)$  (resp. number of harmonic elements), is beyond the scope of this paper. All of these factors affect quality and features of reference control signal and reference state trajectories. A reasonable choice of  $\lambda_k$  and  $T$  achieved by more advanced analysis may determine range of magnitudes of  $u(t)$  and  $\mathbf{x}(t)$  without necessity of introduction extra conditions to limit the variables in the algorithm. This, together with a different form of candidate function, different way of computation of a fine initial guess and analysis of all free parameters, is the subject of further intensive follow-up research.

#### ACKNOWLEDGEMENTS

This work was supported by the project SP2018/160, "Development of algorithms and systems for control, measurement and safety applications IV" of Student Grant System, VSB-TU Ostrava.

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

#### REFERENCES

- P. Balda, M. Schlegel, and M. Stetina. Advanced control algorithms + simulink compatibility + real-time os = rex. volume 16, pages 121–126, 2005.
- S.V. Bolotin and V.V. Kozlov. Calculus of variations in the large, existence of trajectories in a domain with boundary, and whitney's inverted pendulum problem. *Izvestiya Mathematics*, 79(5):894–901, 2015. doi: 10.1070/IM2015v079n05ABEH002765.
- P. Boscaroli and D. Richiedei. Robust point-to-point trajectory planning for nonlinear underactuated systems: Theory and experimental assessment. *Robotics and Computer-Integrated Manufacturing*, 50:256–265, 2018. doi: 10.1016/j.rcim.2017.10.001.
- T. Gluck, A. Eder, and A. Kugi. Swing-up control of a triple pendulum on a cart with experimental validation. *Automatica*, 49(3):801–808, 2013. doi: 10.1016/j.automatica.2012.12.006.
- K. Graichen and M. Zeitz. Feedforward control design for nonlinear systems under input constraints. *Lecture Notes in Control and Information Sciences*, 322:235–252, 2005. doi: 10.1007/11529798\_15.
- K. Graichen and M. Zeitz. Feedforward control design for finite-time transition problems of non-linear mimo systems under input constraints. *International Journal of Control*, 81(3):417–427, 2008. doi: 10.1080/00207170701591465.
- G.-P. He and Z.-Y. Geng. Optimal motion planning for differentially flat underactuated mechanical systems. pages 1567–1572, 2008. doi: 10.1109/ICAL.2008.4636403.
- S. Jadlovska and J. Sarnovsky. A complex overview of modeling and control of the rotary single inverted pendulum system. *Advances in Electrical and Electronic Engineering*, 11(2):73–85, 2013. doi: 10.15598/aeec.v11i2.773.
- Andreas Kunze. Pytrajectory's documentation, 2005. URL <https://pytrajectory.readthedocs.io>.
- John HAUSER Ali JADBABAIE Mark B. MILAM Nicolas PETIT William B. DUNBAR a Ryan FRANZ MURRAY, Richard M. *Online Control Customization via Optimization-Based Control*. Hoboken, NJ, USA: John Wiley & Sons, 2004.
- S. OZANA and T. DOCEKAL. The concept of virtual laboratory and pil modeling with rex control system. pages 98–103, 2017. doi: 10.1109/PC.2017.7976196.
- S. Ozana, M. Pies, and R. Hajovsky. Computation of swing-up signal for inverted pendulum using dynamic optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8838:301–314, 2014.
- E. Shahab, A. Nikoobin, and A. Ghoddosian. Indirect-based approach for optimal swing up and stabilization of a single inverted pendulum with experimental validation. *Control Engineering and Applied Informatics*, 19(4):61–71, 2017.
- M. Tum, G. Gyeong, J.H. Park, and Y.S. Lee. Swing-up control of a single inverted pendulum on a cart with input and output constraints. volume 1, pages 475–482, 2014.
- Z.D. Yu and X.F. Wang. Swing-up control by bvp arithmetic for the rotational inverted pendulum. *Advanced Materials Research*, 211-212:515–519, 2011. doi: 10.4028/www.scientific.net/AMR.211-212.515.
- Y.-L. Zhang, H.-F. Cheng, and H.-X. Li. Automatic swinging-up of double inverted pendulum by inversion-based trajectory control. *Dalian Ligong Daxue Xuebao/Journal of Dalian University of Technology*, 51(6): 896–903, 2011.