RESEARCH PAPERS
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

2019 Volume 27, Number 45

STABILIZATION OF FURUTA PENDULUM USING NONLINEAR MPC

Matúš FURKA[1], Martin KLAUČO[1], Michal KVASNICA[1]

[1]SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY
INSTITUTE OF INFORMATION ENGINEERING, AUTOMATION, AND MATHEMATICS
RADLINSKÉHO 9, 812 37 BRATISLAVA, SLOVAK REPUBLIC
e-mail: matus.furka@stuba.sk, martin.klauco@stuba.sk, michal.kvasnica@stuba.sk
Received 19 August 2019, accepted 24 September 2019, published 29 November 2019

Abstract

This paper is devoted to design of a non-linear model predictive controller (NMPC), which will swing-up and stabilize an inverse rotary pendulum known as the Furuta Pendulum. This paper presents a simulation validation of the NMPC strategy using a full-fidelity non-linear mathematical model of the Furuta pendulum obtained from the Euler-Lagrange motion equations. The NMPC strategy was implemented in MATLAB using the MATMPC toolbox.

Key words

Furuta Pendulum, NMPC, MATMPC

INTRODUCTION

Nowadays, underactuated mechanical systems are generating interest among researchers of modern control theories. This interest is based on the fact that these systems face similar problems to those found in industrial applications, such as external shocks and nonlinear behavior in some conditions of operation. The inverted rotary pendulum is a clear example of a mechanical underactuated system. It was invented in 1992 in Tokyo, the Institute of Technology by the team of K. Furuta [1]. It is a system with two degrees of freedom (DOF) with two rotational joints. It is an example of a complex nonlinear oscillator. The dynamical mathematical model of the rotary pendulum consists of several heavy nonlinearities that arise mainly from Coriolis and centripetal forces.

The objective of this paper is to formulate a non-linear model predictive control strategy [2], that will swing-up and stabilize the arm of the rotary pendulum in the upright position. To obtain the NMPC controller, we first formulate the dynamical mathematical model using Euler-Lagrange motion equations. Subsequently, we use these equations as a design model for the NMPC strategy and for the simulation validation [3]. The NMPC strategy is formulated using a MATMPC toolbox [4].

FURUTA PENDULUM

The Furuta pendulum, also known as the rotary inverted pendulum, is primarily composed of three elements: motor and two bars called arm and pendulum. The motor-driven arm rotates in a horizontal plane and a pendulum attached to the free end of the arm, which can freely rotate in a vertical plane. A schematic picture of the pendulum is shown in Figure 1.

The symbols in the Figure represent the following parameters:

- $\boldsymbol{g}$ – gravitational force
- $\boldsymbol{m_0}$ – mass of the arm
- $\boldsymbol{m_1}$ – mass of the pendulum
- $\boldsymbol{L_0}$ – length of the arm
- $\boldsymbol{L_1}$ – length of the pendulum
- $\boldsymbol{l_1}$ – location of the center of mass of the pendulum
- $\boldsymbol{I_0}$ – moment of inertia of the arm
- $\boldsymbol{I_1}$ – moment of inertia of the pendulum
- $\boldsymbol{\theta_0}$ – arm angle
- $\boldsymbol{\theta_1}$ – pendulum angle
- $\boldsymbol{\tau}$ – motor torque.

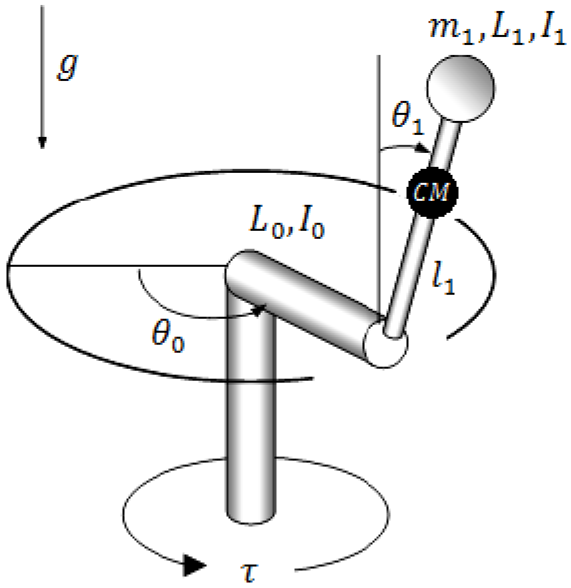

Figure 1 *Schematic picture of Furuta pendulum*

DYNAMICAL MATHEMATICAL MODEL

Dynamic equations of any mechanical system can be obtained from the known classical Newtonian mechanics. The drawback of this formalism is the use of the variables in vector form, complicating the analysis considerably when increasing the joints or there are rotations present in the system. In these cases, it is favourable to employ the Lagrange equations, which exhibit formalism of scale, facilitating the analysis for any mechanical system [5].

The Lagrange function *L* is defined as:

$$L = K - P, \tag{1}$$

where symbol *K* represents the kinetic energy of the system and *P* is potential energy. These two types of energy are represented by following equations:

$$K = f\big(q(t), \dot{q}(t)\big), \tag{2}$$

$$P = f\big(q(t)\big). \tag{3}$$

Kinetic energy is the function of the position component$q(t)$, velocity $\dot{q}(t)$, while potential energy is only a function of position.

For *n* degrees of freedom system, the Euler-Lagrange equations are:

$$\frac{d}{dt}\left(\frac{\partial L(q,\dot{q})}{\partial \dot{q}_i}\right) - \left(\frac{\partial L(q,\dot{q})}{\partial q_i}\right) = \tau_i\,, \tag{4}$$

where τ_i represents the torque of each component.

By solving the Euler-Lagrange equations, we obtained equations of motion for each component of the Furuta pendulum, namely the arm and the pendulum, in the following form:

$$\alpha\ddot{\theta}_0 + 2l_1^2 m_1 \dot{\theta}_0 \dot{\theta}_1 \sin\theta_1 \cos\theta_1 + L_0 l_1 m_1 \left(\ddot{\theta}_1 \cos\theta_1 - \dot{\theta}_1^2 \sin\theta_1\right) = \tau, \qquad (5)$$
$$\alpha = I_0 + L_0^2 m_1 + l_1^2 m_1 \sin^2\theta_{1,}$$

$$I_1\ddot{\theta}_1 + l_1^2 m_1 \ddot{\theta}_1 + L_0 l_1 m_1 \ddot{\theta}_0 \cos\theta_1 - l_1^2 m_1 \dot{\theta}_0^2 \sin\theta_1 \cos\theta_1 - m_1 g l_1 \sin\theta_1 = 0. \qquad (6)$$

The Eq. (5) represents the motion of the arm and (6) is the equation of motion of the pendulum.

To transform the dynamic model into state space representation, we defined the state variables as follows:

$$[x_1 \quad x_2 \quad x_3 \quad x_4]^T = \left[\theta_0 \quad \dot{\theta}_0 \quad \theta_1 \quad \dot{\theta}_1\right]^T, \qquad (7)$$
$$\tau = u.$$

Replacing these variables in equations (5) and (6), the following non-linear equations were obtained:

$$\dot{x}_1 = x_2, \qquad (8a)$$
$$\dot{x}_2 = \frac{\beta(\delta x_2^2 + \rho)\cos x_3 - \gamma(u + \beta x_4^2 \sin x_3 - 2\delta x_2 x_4)}{(\beta\cos x_3)^2 - (\alpha + l_1^2 m_1 \sin^2 x_3)\gamma}, \qquad (8b)$$
$$\dot{x}_3 = x_4, \qquad (8c)$$
$$\dot{x}_2 = \frac{\gamma(u + \beta x_4^2 \sin x_3 - 2\delta x_2 x_4) - (\alpha + l_1^2 m_1 \sin^2 x_3)(\delta x_2^2 + \rho)}{(\beta\cos x_3)^2 - (\alpha + l_1^2 m_1 \sin^2 x_3)\gamma}, \qquad (8d)$$

where

$$\alpha = I_0 + L_0^2 m_1, \qquad (9a)$$
$$\beta = L_0 m_1 l_1, \qquad (9b)$$
$$\gamma = I_1 + l_1^2 m_1, \qquad (9c)$$
$$\delta = l_1^2 m_1 \sin x_3 \cos x_3, \qquad (9d)$$
$$\rho = m_1 g l_1 \sin x_3. \qquad (9e)$$

NMPC DESIGN

We consider the control of nonlinear systems in the discrete-time domain represented by

$$x(t+\Delta) = f\big(x(t), u(t)\big), \qquad (10)$$

where $x(t) \in \mathbb{R}^n$ is the state vector at time t, $u(t)$ is the vector of control inputs, $x(t+\Delta)$ is the successor state at time $t+\Delta$ with Δ denoting the sampling time, and $f\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the (possibly nonlinear) state update function.

The objective of nonlinear model predictive control (NMPC) is to determine the sequence $\{u_0^*, \dots, u_{N-1}^*\}$ of optimal control inputs over a fixed prediction horizon $N \in \mathbb{N}$ that bring the system (10) from any admissible initial state $x(t)$ to a desired final state while minimizing a given performance index, and making sure that the control inputs, as well as the system states remain bounded by $x \in \mathcal{X}$, $u \in \mathcal{U}$ where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^n$. Specifically, the optimal control sequence can be obtained by solving the following NMPC problem:

$$\min_{u_0,\dots,u_{N-1}} \; l_N(x_N) + \textstyle\sum_{j=0}^{N-1} l(x_j, u_j)\,, \qquad (11a)$$
$$\text{s.t.} \quad {}_{j+1} = f\big(x_j, u_j\big), \;\; j = 0, \dots, N-1, \qquad (11b)$$
$$x_j \in \mathcal{X}, \;\; j = 0, \dots, N-1, \qquad (11c)$$
$$u_j \in \mathcal{U}, \;\; j = 0, \dots, N-1, \qquad (11d)$$

$$x_N \in \mathcal{T}, \quad (11e)$$

$$x_0 = x(t). \quad (11f)$$

Here, x_j and u_j represent, respectively, the state and input predictions at a time step j of the prediction window, whose length is N (the prediction horizon). The predictions are initialized from the currently known measurement of the state $x(t)$, or its estimate. The performance index in (11a) involves two penalty functions: l_N as the final penalty, and l as the stage cost. A particular form of these two functions depends on the control objective. If, for instance, one aims at regulating the nonlinear system in (10) to a desired steady state, represented by the tuple (x_S, u_S), one can choose

$$l_N(x_N) = \|x_N - x_S\|_P^2 \quad (12)$$

and

$$l(x_j, u_j) = \|x_j - x_S\|_Q^2 + \|u_j - u_S\|_R^2, \quad (13)$$

where x_S is the vector of desired steady states, u_S represents the control action when the system is in steady state x_S and $\|z\|_W^2 = z^T W z$ is the weighted squared two-norm of a vector.

In this paper, we propose a solution to the NMPC problem (11) using the MATMPC toolbox. It is an open source software build in Matlab for non-linear model predictive control. MATMPC has a number of algorithmic modules, including automatic differentiation; direct multiple shooting, considering linear quadratic program (QP) solver and globalization.

EXPERIMENTAL

The aim of this paper is to stabilize the rotary inverted pendulum in upright position using NMPC controller designed by MATMPC toolbox. First, we assumed the following physical parameters of the pendulum:

Parameter	Value	Unit
g	9.810	ms^{-2}
m_0	0.600	kg
m_1	0.198	kg
L_0	0.510	m
L_1	0.230	m
l_1	0.210	m
I_0	0.031	kgm^2
I_1	0.002	kgm^2

where we made an approximation in the calculation of the moments of inertia for each, arm and pendulum. The formulas take the following form:

$$I_0 = \frac{1}{3} m_0 L_0^2 \quad (14)$$

$$I_1 = \frac{1}{12} m_1 L_1^2. \quad (15)$$

Since the aim is to stabilize the pendulum in upright position, meaning that $\theta_1 = 0$, the reference point for non-linear model predictive control is as follows:

$$x_{S\_UP} = [0 \quad 0 \quad 0 \quad 0]^T$$
$$u_{S\_UP} = 0,$$

where x_{S_UP} is the desired steady state in (12) and (13), and u_{S_UP} is steady state control action in (13).

Having all the necessary information about the pendulum, the work continued with the simulation setup. First, we defined two weighting matrices. One for the *N-1* states, defined as

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}. \tag{16}$$

Matrix W has to be a symmetric positive definite matrix of dimension $W \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$, where n_x is number of states and n_u is number of inputs.

The second matrix was for the last *N* stage. It was defined as

$$W_N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}, \tag{17}$$

where W_N has to be a symmetric positive definite matrix of dimension $W_N \in \mathbb{R}^{n_x \times n_x}$.

The only constrains we need are the ones on the control action

$$-6 \leq u \leq 6. \tag{18}$$

The simulation starts, when the pendulum is hanging downwards, meaning that the initial condition is

$$x_0 = [0 \quad 0 \quad \pi \quad 0]^T.$$

To design the NMPC controller, we first discretized the nonlinear dynamics in (8) using the forward Euler discretization with a sampling time of Δ = 0.02 seconds. Subsequently, we employed the MATMPC toolbox to compile the mathematical control formulation in (11) into an executable code. Using such a designed controller, we obtained the results represented in Figure 2. Coming out from this control behavior, it is obvious, that the NMPC controller was able to swing up the pendulum and stabilize it in the upright unstable position within 4 seconds. The last graph in Figure 2 represents the control action, which remained between the specified boundaries.

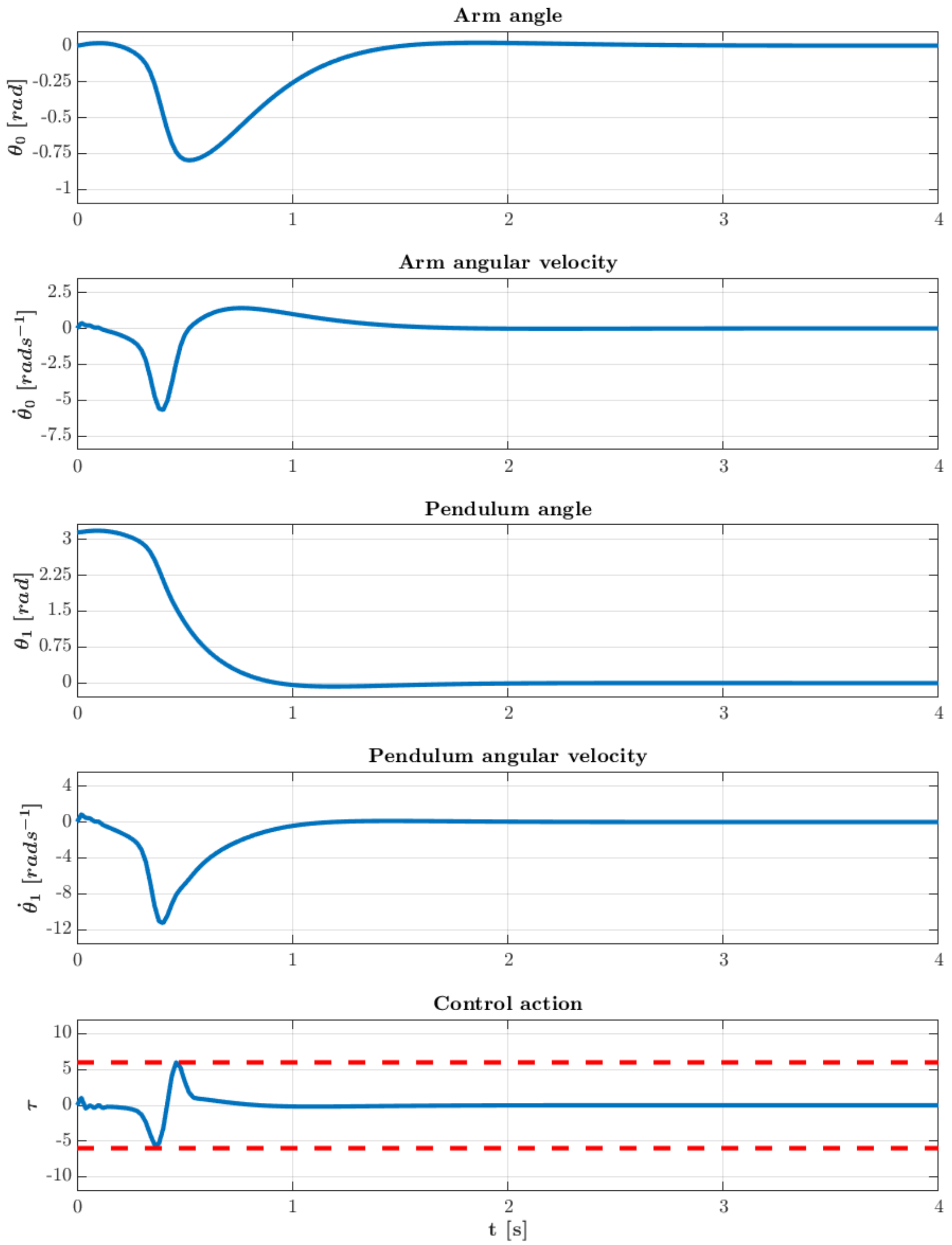

Figure 2 *Swing-up and upright control by non-linear MPC designed using MATMPC toolbox*

CONCLUSION

In this paper, we show how to control a Furuta pendulum using model predictive control. To facilitate a fast computation of the control action in the range of milliseconds, we employed the MATMPC library that allows for compilation of the mathematical control problem into an executable form. A further advantage of the approach is that it can cope directly with the nonlinear dynamics of the controlled plant without having to resort to linearizations. Simulations results clearly show that the proposed approach can achieve the goal of swinging up the pendulum and stabilizing it in the upwards unstable equilibrium while respecting physical constraints of the controlled plant.

Acknowledgement

This paper was supported by the funding of Slovak Ministry of Education, Science, Research and Sport under the project of STU as the Leader of Digital Coalition 002STU-2-1/2018. The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0585/19, the contribution of the Slovak Research and Development Agency under the APVV 15-0007 Project.

References

[1] FURUTA, K., YAMAKITA, M., and KOBAYASHI, S. 1992. Swing-up control of inverted pendulum using pseudo-state feedback. *Journal of Systems and Control Engineering*, **206**(6), pp. 263–269.
[2] MAGNI, L. RAIMONDO, D. M., & ALLGÖWER, F. 2009. Nonlinear model predictive control. Lecture Notes in Control and Information Sciences, (384).
[3] HOUSKA, B., FERREAU, H. J., & DIEHL, M. 2011. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, **47**(10), pp. 2279-2285.
[4] Y. CHEN, M. BRUSCHETTA, E. PICOTTI, and A. BEGHI. MATMPC – A MATLAB Based Toolbox for Real-time Nonlinear Model Predictive Control, 2018.
[5] DUART, J.L., MONTERO, B., OSPINA, P.A., GONZÁLES, E. 2017. Dynamic Modelling and Simulation of a Rotational Inverted Pendulum, *Journal of Physics*.

ORCID

Martin Klaučo 0000-0003-0098-2625
Michal Kvasnica 0000-0001-8699-551X
Matúš Furka 0000-0003-1945-6772