Nonlinear Design of Model Predictive Control Adapted for Industrial Articulated Robots

Květoslav Belda
Department of Adaptive Systems
The Czech Academy of Sciences, Institute of Information Theory and Automation
Pod Vodárenskou věží 4, 182 08 Prague 8, Czech Republic
belda@utia.cas.cz

Keywords: Discrete Model Predictive Control, Nonlinear Design, Lagrange Equations, Articulated Robots.

Abstract: This paper introduces a specific nonlinear design of the discrete model predictive control based on the features of linear methods used for the numerical solution of ordinary differential equations. The design is intended for motion control of robotic or mechatronic systems that are usually described by nonlinear differential equations up to the second order. For the control design, the explicit linear multi-step methods are considered. The proposed way enables the design to apply nonlinear model to the construction of equations of predictions used in predictive control. An example of behavior of proposed versus linear predictive control is demonstrated by a comparative simulation with nonlinear mathematical model of six-axis articulated robot.

1 INTRODUCTION

The number of industrial robots increases steadily. Such trend proceeds with the advent of Industry 4.0 also in the immediate future (Colombo et al., 2016). The robots in industrial production perform thousands of operations. Their efficiency depends highly on the adequate motion control that can employ available pieces of information from user and real data (Chung et al., 2016).

Nowadays, necessary information, data and computing power are broadly available, however, the designers have to combine them effectively. In industrial production, there exist a lot of elaborated strategies that follow from long-term, empirical experiences (SPS IPC Drives, 2017). Unfortunately, such strategies are usually not universal enough. They are not scalable or simply transferable for general use in different or modified systems.

From mathematical point of view, the robots, manipulators, such as a mechanical structure of articulated robot class depicted in Fig. 1, represent dynamic systems that are usually described by systems of ordinary differential equations (ODEs) forming appropriate dynamic models (Siciliano et al., 2009). Such models can be considered as a proper substitution of the real physical robot mechanism for computer simulations and motion control design as well.

The mentioned systems of ODEs, include various relations among individual elements of the robot constructions. These relations tend to be nonlinear due to the presence of nonlinear operations on descriptive variables and their appropriate derivatives such as multiplication, division, goniometric functions etc. (Arimoto, 1996). Thus, conventional linear control theory (Wang, 2009) cannot be directly exploited without modifications.

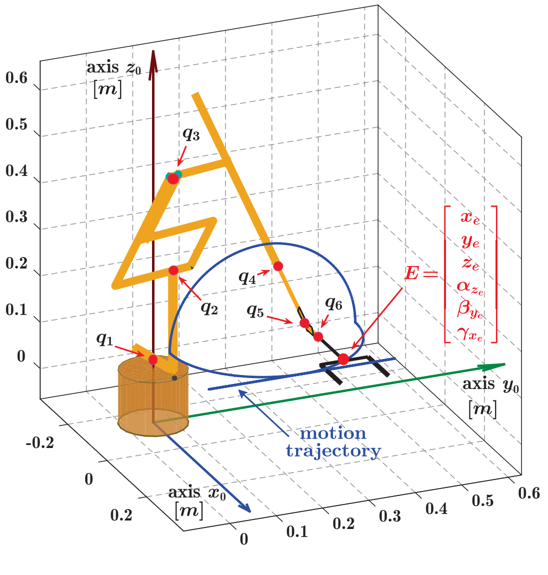


Figure 1: Wire-frame model of the articulated robot class.

Belda, K.
Nonlinear Design of Model Predictive Control Adapted for Industrial Articulated Robots.
In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018) - Volume 2*, pages 71-80
ISBN: 978-989-758-321-6

In engineering practice, there exist several usual solutions. They are based on local linearization (Ostafew et al., 2016) by means of Taylor series, partial derivative models or switching local linear models with an inclusion of discretization to obtain discrete linear-like model, often in state-space form. Then, usual linear design of discrete model predictive control can be applied (Ordis and Clarke, 1993; Maciejowski, 2002; Wang, 2009; Belda et al., 2007; Belda and Vošmik, 2016; Belda and Rovný, 2017). Succeeding approaches use linear models with stochastic uncertainties substituting locally initial nonlinear model (Kothare et al., 1996). Other solutions can be realized by neural network (Negri et al., 2017) or by the direct nonlinear optimization such as in (Kirches, 2011; Houska et al., 2011; Wilson et al., 2016; Faulwasser et al., 2017; Zanelli et al., 2017). However, they do not represent immediately applicable, straightforward multi-step design in each time-instant of the control process.

This paper introduces a novel way of the design of the discrete model predictive control that however employs a continuous-time nonlinear ODE model of the robot dynamics. The model is employed directly by means of specifically adapted explicit linear multi-step numerical methods without any linearizaton and conventional model discretization. The numerical methods are used for the construction of equations of predictions. These equations can be involved in usual way to the regular quadratic cost function and optimization criterion. The explanation is introduced with Adams-Bashforth method as a representative of the aforementioned explicit methods (Butcher, 2016). A summary of the features of the proposed solution is involved including its comparison with usual linear design of model predictive control (Ordis and Clarke, 1993; Wang, 2009) considering conventional repeated linearization and discretization in compliance with the robot motion.

2 NONLINEAR ROBOT MODEL

The model of the considered class “articulated robots” is generally represented by a nonlinear function describing relations between control actions (robot inputs, joint torques $\tau = \tau(t)$) and descriptive variables (robot outputs, joint angles and their derivatives $q = q(t)$, $\dot q = \dot q(t)$ and $\ddot q = \ddot q(t)$):

$$\ddot q = \mathtt{f}(q, \dot q, \tau) \tag{1}$$

The model (1) represents equations of motions (Siciliano et al., 2009; Othman et al., 2015) that describe robot dynamics.

Such a model is mostly composed by Lagrange equations, e.g. in the following form

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot q}\right)^T - \left(\frac{\partial E_k}{\partial q}\right)^T + \left(\frac{\partial E_p}{\partial q}\right)^T = \tau \tag{2}$$

where q, $\dot q$, E_k, E_p and τ are generalized coordinates and their appropriate derivatives, total kinetic and potential energy and vector of generalized force effects associated with generalized coordinates (Siciliano et al., 2009).

The individual terms in the equation system (2) can be defined as follows

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot q}\right)^T = \tilde H(q,\dot q)\,\dot q + H(q)\,\ddot q \tag{3}$$

$$-\left(\frac{\partial E_k}{\partial q}\right)^T = S(q,\dot q)\,\dot q - \frac{1}{2}\tilde H(q,\dot q)\,\dot q \tag{4}$$

$$\left(\frac{\partial E_p}{\partial q}\right)^T = g(q) \tag{5}$$

where, considering simplified notation, the matrices $H = H(q)$, $S = S(q,\dot q)$ and $\tilde H = \tilde H(q,\dot q) = \frac{d}{dt}\,(H(q))$ relate to inertia effects and vector $g = g(q)$ corresponds to effects of gravity. Thereafter, the model (equations of motion of articulated robots) can be defined as follows

$$\ddot q = \overbrace{\underbrace{\underbrace{-H^{-1}\left(\frac{1}{2}\tilde H + S\right)\dot q}_{f(q,\dot q)}\;\underbrace{-H^{-1}g}_{f_g(q)}}_{f_c(q,\dot q)}\;\underbrace{+H^{-1}\tau}_{g_\tau(q)\,\tau}}^{\mathtt{f}(q,\dot q,\tau)}$$

$$= f(q,\dot q) + u \tag{6}$$

where the terms $f(q,\dot q) = -H^{-1}\left(\frac{1}{2}\tilde H + S\right)\dot q$ and $u = H^{-1}(-g + \tau)$.

Thus, equation (6) can be introduced as a particular expression of the model (1) as follows

$$\ddot q = f_c(q,\dot q) + g_\tau(q)\,\tau = f(q,\dot q) + f_g(q) + g_\tau(q)\,\tau$$

$$= f(q,\dot q) + g(q)\,u \tag{7}$$

where $g(q)$ is added just for the generality; it is identity matrix here. In (7), the effects of gravity $f_g(q)$ are incorporated into robot inputs as outer forces considering their static and fixed character that cannot be reduced or suppressed in the control design.

Note that final torques required on drives (reference torques for drive control) are given by

$$\tau = H\,u + g \tag{8}$$

3 INTEGRATION CONCEPT

To obtain discrete form of nonlinear predictive control, let us consider individual time instants of the initial continuous nonlinear function in the model (1) as follows (i.e. sampling, no model discretization)

$$\mathtt{f}_k = \mathtt{f}(q, \dot{q}, \tau)|_{\,t\,=\,kT_s}\,, \quad k = 0, 1, \cdots \tag{9}$$

where T_s is suitably selected sampling period. Let us apply the same to the terms from (7)

$$f_k = f(q, \dot{q}) \;\; g_k = g(q)\,|_{\,t\,=\,kT_s} \tag{10}$$

The statements (9) or (10) will be suitable for a specific construction of the predictions relative to unknown control actions $u(t)$ within finite discrete time set $t \in \{k\,T_s,\,(k{+}1)\,T_s, \cdots, (k{+}N{-}1)\,T_s\}$, where N is a prediction horizon.

The predictions still take into account the continuous nonlinear model, but only in the indicated discrete time samples for discrete design of predictive control. Such a concept can be realised by means of numerical methods (Butcher, 2016) used for the numerical approximation of the solution of the first-order ordinary differential equations (ODEs):

$$\dot{y} = \mathtt{f}(t, y) \;\text{ with initial condition }\; y_0 = y|_{\,t\,=\,0} \tag{11}$$

Specifically, these methods are used to find a numerical approximation of the exact integral over a specific time interval, e.g. $t \in \langle k\,T_s, (k{+}1)\,T_s \rangle$

$$\begin{aligned} y_{k+1} &= y_k + \int_{k\,T_s}^{(k+1)\,T_s} \dot{y}\, dt = y_k + \int_{k\,T_s}^{(k+1)\,T_s} \mathtt{f}(t, y)\, dt \\ \hat{y}_{k+1} &= y_k + h\,\delta(t,\, y) \end{aligned} \tag{12}$$

where $y_k = y|_{t=k\,T_s}$ is an initial condition of the considered time interval, $\hat{y}_{k+1}$ is an approximation of the exact solution $y_{k+1} = y|_{t=(k+1)\,T_s}$, $\delta(t, y)$ represents generally the function approximating $\dot{y}$ so that $\hat{y}_{k+1}$ would be the adequate approximation of y_{k+1} and h is a step of integration method, which is selected as $h = T_s$.

From wide range of the methods, let us focus on linear multi-step methods suitable for the predictions in predictive design. Generally, linear multi-step methods are defined as

$$\hat{y}_{k+1} = \sum_{i=0}^{r} \alpha_i\, y_{k-i} + h \sum_{j=-1}^{s} \beta_j\, \mathtt{f}_{k-j} \tag{13}$$

where $\hat{y}_{k+1}$ is a result of the numerical integration based on the previous y_{k-i} .

For the design, the explicit methods are useful such as explicit Adams-Bashforth method of fourth order with $r = 0$, $\alpha_0 = 1$, $s = 3$ and $\beta_j = \frac{\gamma_j}{24}$, $j \in \{-1, 0, 1, 2, 3\}$, where $\gamma_{-1} = 0$, $\gamma_0 = 55$, $\gamma_1 = -59$, $\gamma_2 = 37$ and $\gamma_3 = -9$, as follows

$$\begin{aligned} \hat{y}_{k+1} = y_k + h\,(&- \tfrac{9}{24}\,\mathtt{f}_{k-3} + \tfrac{37}{24}\,\mathtt{f}_{k-2} \\ &- \tfrac{59}{24}\,\mathtt{f}_{k-1} + \tfrac{55}{24}\,\mathtt{f}_k) \end{aligned} \tag{14}$$

Thus, for completeness, the function approximating $\dot{y}$ from (12) is as follows:

$$\delta(t, y) = -\tfrac{9}{24}\,\mathtt{f}_{k-3} + \tfrac{37}{24}\,\mathtt{f}_{k-2} - \tfrac{59}{24}\,\mathtt{f}_{k-1} + \tfrac{55}{24}\,\mathtt{f}_k .$$

The aforementioned Adams-Bashforth linear method (Butcher, 2016) will be used in the next explanation of the following section.

4 NONLINEAR DESIGN OF PREDICTIVE CONTROL

The nonlinear model predictive design focusses recently on the solution of nonlinear optimal control problem with integral criterion of optimality. It represents general solution using sophisticated nonlinear optimization algorithms. But it leads to sequential quadratic programming (QP) representing one-ahead spreading-in-time-optimization process, thus iteratively approximating the nonlinear problem with QP (Faulwasser et al., 2017; Zanelli et al., 2017). Alternatively, the usual summation form of the criterion used in linear control theory can be also taken into account. The operation of integration is just shifted from the criterion towards predictions via continuous-time model represented by ODEs. This idea will be used and introduced in the proposed design.

Let us start from the continuous model (7) considered in the discrete time samples (time instants) as was designated in (10). Moreover, let us use more common, universal notation for outputs y instead of q, i.e. $y_k = q_k$ as well as for appropriate derivatives $\dot{y}_k = \dot{q}_k$ and $\ddot{y}_k = \ddot{q}_k$

$$\ddot{y}_k = f_k + g_k\, u_k \tag{15}$$

From (6), the function f_k holds $f_k|_{[\dot{y}\,=\,0,\; y\,\in R)]} = 0$ whereas term $f_{g\,k}|_{\,y\,\in\,R} \neq 0$ in (7) for the considered spatial articulated robot class as well as for vertical planar robot configurations. For completeness, $f_{g\,k}|_{\,y\,\in\,R} = 0$ applies to horizontal planar robot configurations. The property of f_k is helpful for the stability of the control design.

Using the model (15), the specific design of the nonlinear model predictive control can be now explained within the following three sections.

4.1 Criterion and Cost Function

The criterion for predictive control design can generally be written as follows

$$\begin{aligned} &\min_{U_k} \; J_k \; (\hat{Y}_{k+1}, W_{k+1}, U_k\,) \\ &\text{subject to: } \hat{Y}_{k+1} = f(y_k,\, \dot{y}_k,\, \delta(t, y, \dot{y})) \\ &\qquad\qquad\;\; \ddot{y}_k \;\; = f_k + g_k\, u_k \end{aligned} \tag{16}$$

where $\delta(t, y, \dot{y})$ follows from the second order model $\ddot{y}_k = f_k + g_k u_k$, where $f_k = f(y, \dot{y})$ as in (10).

$\hat{Y}_{k+1}$, W_{k+1} and U_k stand for the sequences of the robot output predictions, references and control actions, respectively

$$\hat{Y}_{k+1} = [\hat{y}^T_{k+1}, \cdots, \hat{y}^T_{k+N}]^T \tag{17}$$

$$W_{k+1} = [w^T_{k+1}, \cdots, w^T_{k+N}]^T \tag{18}$$

$$U_k = [u^T_k, \cdots, u^T_{k+N-1}]^T \tag{19}$$

Then the cost function J_k is selected in common quadratic form as follows

$$\begin{aligned} J_k &= \sum_{i=1}^{N} \{ ||Q_{yw}\,(\hat{y}_{k+i} - w_{k+i})||_2^2 + ||Q_u\, u_{k+i-1}||_2^2 \} \\ &= (\hat{Y}_{k+1} - W_{k+1})^T\, Q^T_{YW}\, Q_{YW}\, (\hat{Y}_{k+1} - W_{k+1}) \\ &\quad + U^T_k\, Q^T_U\, Q_U\, U_k \end{aligned} \tag{20}$$

where $Q^T_{YW}\, Q_{YW}$ and $Q^T_U\, Q_U$ are penalizations defined as

$$Q^T_\diamond Q_\diamond = \begin{bmatrix} Q^T_* Q_* & & 0 \\ & \ddots & \\ 0 & & Q^T_* Q_* \end{bmatrix} \tag{21}$$

where the symbolic subscripts $\diamond$, $*$ have the following interpretation: $\diamond \in \{YW,\, U\}$ and $* \in \{yw,\, u\}$.

However, the function can be selected variously according to user requirements, e.g. considering incremental terms that can slightly moderate and smooth the robot motion (Maciejowski, 2002) or suppress steady-state control error (Wang, 2009; Belda and Záda, 2017).

4.2 Equations of Predictions

As was already mentioned, the equations of predictions will be composed with the nonlinear continuous model (15) and by means of the idea of the approximation of the exact integral as indicated in (12) by the exemplarily selected linear multi-step Adams-Bashforth method of fourth order (14) for the solution of the first-order ODEs.

However, the nonlinear model of the robot (15) represents a system of the second-order ODEs. To apply the selected suitable numerical method, but without lost of information about included nonlinear relations, the model (15) has to be specifically rearranged into ODEs of the first order. For the necessary rearrangement, the following backward Euler formula can be selected for simplicity

$$\hat{\dot{y}}_{k+1} = \frac{\hat{y}_{k+1} - y_k}{h} \tag{22}$$

The used formula ensures coupling (propagation) of nonlinear relations from (15) into positional estimates, since the numerical methods represent only linear combinations within rows of the ODEs. Thus, usual rearrangement via addition of further ODEs decreasing the order of initial ODE system would not be helpful.

Taking the aforementioned features into account, then the positional estimate $\hat{y}_{k+1}$ can be evaluated by the velocity estimate $\hat{\dot{y}}_{k+1}$, which includes fully the initial nonlinear model (15), as follows

$$\hat{y}_{k+1} = y_k + h\, \hat{\dot{y}}_{k+1} \tag{23}$$

where the estimate $\hat{\dot{y}}_{k+1}$ is given generally by the numerical integration of the ODE set as follows

$$\hat{\dot{y}}_{k+1} = \dot{y}_k + h\, \delta(t,\, y,\, \dot{y}) \tag{24}$$

Note, for completeness, if the appropriate robot model would be set of first-order ODEs only, i.e.

$$\tilde{\dot{y}}_k = \tilde{f}_k + \tilde{g}_k\, u_k$$

as e.g. in (Groothuis et al., 2017; Záda and Belda, 2017), this step is omitted.

Now, the pure equations of predictions can be composed considering the model (15), a specific numerical method (24) (here, the Adams-Bashworth (14) is selected) and the rearrangement to the first-order ODE set as indicated by (23).

The equations of predictions in a matrix form are defined for the velocity vector $\hat{\dot{Y}}_k$ as follows

$$\hat{\dot{Y}}_{k+1} = \dot{y}_k\, F_I + F_k + G_k\, U_k \tag{25}$$

and as well as for the position vector $\hat{Y}_k$ as

$$\hat{Y}_{k+1} = y_k\, F_I + L_k + M_k\, U_k \tag{26}$$

where the individual terms represent multiple identity matrix: $F_I = [\, I \cdots I\,]^T$, specific free responses: " $\dot{y}_k\, F_I + F_k$ ", " $y_k\, F_I + L_k$ " and forced responses: " $G_k\, U_k$ ", " $M_k\, U_k$ ", respectively.

F_k and G_k can be derived from the following sequences for velocities, where, for clarity, individual time steps are separated by horizontal lines:

$$\begin{aligned} \hat{\dot{y}}_{k+1} &= \dot{y}_k + F_{1,k} \\ &\quad + \beta_0\, g_k\, u_k \\ F_{1,k} &= \beta_3\, \ddot{y}_{k-3} + \beta_2\, \ddot{y}_{k-2} + \beta_1\, \ddot{y}_{k-1} + \beta_0\, f_k \end{aligned}$$

$$\begin{aligned} \hat{\dot{y}}_{k+2} &= \dot{y}_k + F_{2,k} \\ &\quad + (\beta_1 + \beta_0)\, g_k\, u_k + \beta_0\, \hat{g}_{k+1}\, u_{k+1} \\ F_{2,k} &= F_{1,k} + \beta_3\, \ddot{y}_{k-2} + \beta_2\, \ddot{y}_{k-1} + \beta_1\, f_k + \beta_0\, \hat{f}_{k+1} \end{aligned}$$

$$\vdots$$

$$\begin{aligned} \hat{\dot{y}}_{k+N} &= \dot{y}_k + F_{N,k} + \{\sum_{j=0}^{3} \beta_j\}\, g_k u_k \\ &\quad + \cdots + \beta_0\, \hat{g}_{k+N-1}\, u_{k+N-1} \\ F_{N,k} &= F_{N-1,k} + \sum_{j=1}^{4} \beta_{j-1} \hat{f}_{k+N-j} \end{aligned} \tag{27}$$

Note that in step k, the topical value $\dot{y}_k$ as well as its appropriate past values $\dot{y}_{k-1}$, $\dot{y}_{k-2}$ and $\dot{y}_{k-3}$ are known from measurements or some state estimation.

Consequently, vector F_k from (25) can be defined in the following way:

$$F_k = \begin{bmatrix} F_{1,k} \\ F_{2,k} \\ \vdots \\ F_{N,k} \end{bmatrix} \tag{28}$$

Similarly, the matrix G_k from (25) is defined as

$$G_k = \begin{bmatrix} \beta_0\, g_k & 0 & \cdots & 0 \\ (\beta_1 + \beta_0)\, g_k & \beta_0\, \hat{g}_{k+1} & & \vdots \\ \vdots & & \ddots & \vdots \\ \{\sum\limits_{j=0}^{3} \beta_j\}\, g_k & \cdots & \cdots & \beta_0\, \hat{g}_{k+N-1} \end{bmatrix} \tag{29}$$

where $\hat{f}_{k+i}$ and $\hat{g}_{k+i}$ can be substituted by future reference values as $\hat{f}_{k+i} = f_{k+i}(w_{k+i}, \dot{w}_{k+i})$ and $\hat{g}_{k+i} = g_{k+i}(w_{k+i})$.

Similarly in the construction of the equations (26), $\hat{Y}_{k+1}$, L_k and M_k can be defined by analogical sequences for positions as follows (again for clarity, individual time steps are separated by horizontal lines):

$$\begin{aligned} \hat{y}_{k+1} &= y_k + L_{1,k} + h\, \beta_0\, g_k\, u_k \\ L_{1,k} &= h\, \dot{y}_k + h\, F_{1,k} \end{aligned}$$

$$\begin{aligned} \hat{y}_{k+2} &= y_k + L_{2,k} \\ &\quad + h\, (\beta_1 + 2\, \beta_0)\, g_k\, u_k + h\, \beta_0\, \hat{g}_{k+1}\, u_{k+1} \\ L_{2,k} &= L_{1,k} + h\, F_{2,k} \end{aligned}$$

$$\vdots$$

$$\begin{aligned} \hat{y}_{k+N} &= y_k + L_{N,k} \\ &\quad + h\, \{\sum_{j=0}^{3} (N-j)\beta_j\}\, g_k\, u_k \\ &\quad + \cdots + h\beta_0\, \hat{g}_{k+N-1}\, u_{k+N-1} \\ L_{N,k} &= L_{N-1,k} + h\, F_{N,k} \end{aligned} \tag{30}$$

Then, vector L_k and matrix M_k from (26) are:

$$L_k = \begin{bmatrix} L_{1,k} \\ L_{2,k} \\ \vdots \\ L_{N,k} \end{bmatrix} \tag{31}$$

$$M_k = \begin{bmatrix} h\, \beta_0\, g_k & 0 & \cdots & 0 \\ h\, (\beta_1 + 2\, \beta_0)\, g_k & h\, \beta_0\, \hat{g}_{k+1} & & \vdots \\ \vdots & & \ddots & \vdots \\ h\{\sum\limits_{j=0}^{3} (N-j)\beta_j\} g_k & \cdots & \cdots & h\, \beta_0\, \hat{g}_{k+N-1} \end{bmatrix} \tag{32}$$

Note that due to second order ODEs, the two-step equations of predictions (25) and (26) are used here.

4.3 Square-root Minimization

To minimize cost function (20), let us consider the following expression

$$\min_{U_k} J_k = \min_{U_k} \mathbb{J}_k^T \mathbb{J}_k \;\Rightarrow\; \min_{U_k} \mathbb{J}_k \tag{33}$$

indicating the square-root minimization of the vector $\mathbb{J}_k$ instead of minimization of the scalar J_k, where the square-root minimization is more suitable from computation point of view. Thus, the square-root of the criterion (16) with the cost function (20) is as follows

$$\min_{U_k} \mathbb{J}_k = \min_{U_k} \begin{bmatrix} Q_{YW} & 0 \\ 0 & Q_U \end{bmatrix} \begin{bmatrix} \hat{Y}_{k+1} - W_{k+1} \\ U_k \end{bmatrix} \tag{34}$$

which can be solved as a specific least-square problem by the following system of algebraic equations (Lawson and Hanson, 1995) that involves equations of predictions (26) for $\hat{Y}_{k+1}$

$$\begin{bmatrix} Q_{YW} M_k \\ Q_U \end{bmatrix} U_k = \begin{bmatrix} Q_{YW}(W_{k+1} - y_k F_I - L_k) \\ 0 \end{bmatrix} \tag{35}$$

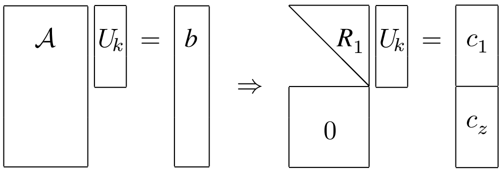
The system (35), that is over-determined, can be written in condensed general form (36). This form can be transformed to another form (37) by orthogonal-triangular decomposition (Lawson and Hanson, 1995) and solved for unknown U_k

$$\mathcal{A} U_k = b \tag{36}$$

$$Q^T \mathcal{A} U_k = Q^T b \quad \text{assuming that} \quad \mathcal{A} = QR$$

$$R_1 U_k = c_1 \tag{37}$$

where Q^T is an orthogonal matrix that transforms matrix $\mathcal{A}$ into upper triangle R_1 as indicated



(38)

Vector c_z represents a loss vector, Euclidean norm $||c_z||$ of which equals to the square-root of the optimal cost function minimum, scalar value $\sqrt{J}$ i.e. $J = c_z^T c_z$. Only the first elements corresponding to u_k are selected from computed vector U_k .

5 SIMULATION EXAMPLE

The example demonstrates the behavior of the articulated robot along selected testing trajectory. The corresponding wire-frame model including trajectory is shown in Fig. 1. Trajectory in detail is in Fig. 2 and Fig. 3.

The depicted trajectory in Fig. 1 was time parameterized with acceleration polynomial of fifth-order (Siciliano et al., 2009; Belda and Novotný, 2012). The specification of individual trajectory segments is listed in the Table 1

Table 1: Testing trajectory in G code (mm)

001:	N010	G19						
002:	N020	G00	X630	Y-200	Z400			
003:	N030	G00	X630	Y200	Z400			
004:	N040	G00	X630	Y0	Z400			
005:	N050	G02	X430	Y-200	Z400	I-200	J0	K0
006:	N060	G02	X430	Y200	Z400	I0	J200	K0
007:	N070	G02	X630	Y0	Z400	I0	J-200	K0
008:	N080	G00	X630	Y-200	Z400			
009:	N090	G00	X630	Y200	Z400			
010:	N010	G00	X630	Y0	Z400			

where G19, G00 and G02 are commands for plane selection, linear and circular interpolation, respectively.

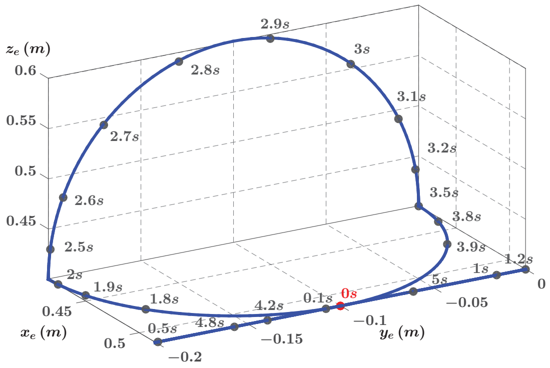


Figure 2: Testing trajectory with specific time marks.

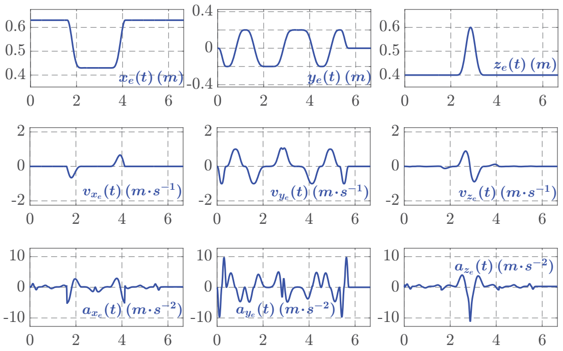


Figure 3: Cartesian coordinates and derivatives (time in (s)).

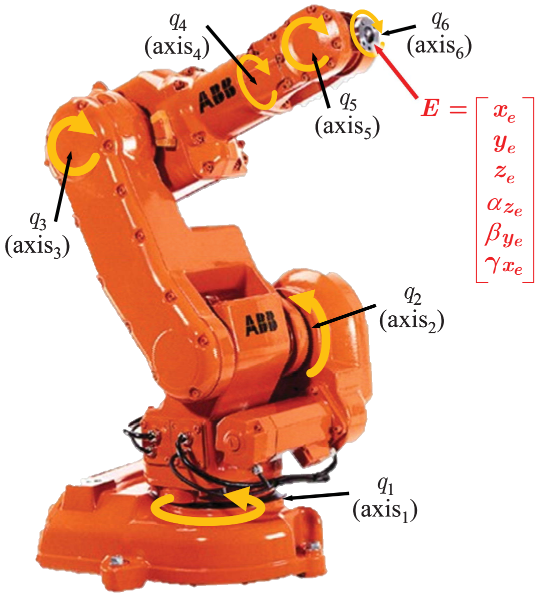

q_4 (axis$_4$)
q_6 (axis$_6$)
q_5 (axis$_5$)
$E = \begin{bmatrix} x_e \\ y_e \\ z_e \\ \alpha_{z_e} \\ \beta_{y_e} \\ \gamma_{x_e} \end{bmatrix}$
q_3 (axis$_3$)
q_2 (axis$_2$)
q_1 (axis$_1$)

Figure 4: Six-axis multipurpose ABB robot IRB 140.

The used dynamic model (7) and (8) from (2)-(6) had parameters of ABB robot IRB 140 (Fig. 4).

The number of actuated (driven) axes of the robot is six as well as a number of degrees of freedom of the robot. Six degrees of freedom correspond to six inputs: torques $\tau_{1:6}\,(N{\cdot}m)$, six outputs: joint coordinates $y = q_{1:6}\,(rad)$ relating to the appropriate Cartesian coordinates:

$$E \;=\; [\,\{x_e,\; y_e,\; z_e\,(m)\},\; \{\alpha_{z_e},\; \beta_{y_e},\; \gamma_{x_e}\,(rad)\,\}\,]^T$$

and twelve state variables:

$$x = [\,q_{\,1:6}^{\,T}\,(rad),\; \dot{q}_{\,1:6}^{\,T}\,(rad\cdot s^{-1})\,]^{\,T}$$

corresponding to joint coordinates and their appropriate derivatives.

Note that end-effector was oriented to be parallel with axis x_0. Thus, the orientation angles are considered to be constant:

$$\alpha_{z_e} \;=\; \beta_{y_e} \;=\; \gamma_{x_e} \;=\; 0\,rad$$

However, corresponding reference values in joint space $w_{1:6,\forall k}$, $k = 1, 2, \cdots$, are changing according to appropriate kinematic transformation specific for considered robot.

5.1 Simulation Setup

Proposed nonlinear design of model predictive control (NdMPC) was tested with the following parameters:

- sampling period: $Ts = 0.01s$
- horizon of prediction: $N = 10$
- output penalization: $Q_{yw} = I_{(6\times6)}$
- input penalization: $Q_u = 2\cdot 10^{-4}\, I_{(6\times6)}$ where I is the identity matrix.

The proposed algorithm was compared with regular model predictive control (MPC) (Wang, 2009) having identical setting. The MPC, used for the comparison, was as follows

$$\begin{aligned} U_k = {} & (\tilde{G}_k^T Q_{YW}^T\, Q_{YW} \tilde{G}_k + Q_U^T\, Q_U)^{-1} \\ & \times \tilde{G}_k^T Q_{YW}^T\, Q_{YW} (W_{k+1} - \tilde{F}_k\, x_k) \end{aligned} \tag{39}$$

where matrices $\tilde{F}$ and $\tilde{G}$ are derived from the initial nonlinear model (7) as follows

$$\tilde{F}_k = \begin{bmatrix} CA_k \\ \vdots \\ CA_k^N \end{bmatrix}$$

$$\tilde{G}_k = \begin{bmatrix} CB_k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ CA_k^{N-1}B_k & \cdots & CB_k \end{bmatrix} \tag{40}$$

Linear-like matrices A_k and B_k, involved in (40), are obtained conventionally by model linearization and discretization repeating in every time instant k. C is an output matrix. All terms follow from state-space form:

$$\underbrace{\begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & I \\ 0 & f(\dot{y}, y) \end{bmatrix}}_{A(t)} \underbrace{\begin{bmatrix} y \\ \dot{y} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_{B} u \tag{41}$$

$$y \;= \underbrace{\begin{bmatrix} I & \quad 0 \end{bmatrix}}_{C} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \tag{42}$$

that is discretized: $A(t), B\,|_{T_s} \;\Rightarrow\; A_k, B_k$ by first-order-hold method (Franklin et al., 1998) and arranged as

$$y_{k+1} = C\,A_k\,x_k \;+\; C\,B_k\,u_k \tag{43}$$

The details on used MPC can be found e.g. in (Ordis and Clarke, 1993; Wang, 2009).

5.2 Summary of the results

The simulation was performed with the mentioned setting and with artificially added mismatch between the model used for the control design and the model for simulation. The mismatch consisted in the four-times increased weight of the last, 6^{th} robot link in the simulation model against model for the design, i.e. $m_6 = 0.25\,\text{kg}$ and $\tilde{m}_6 = 4 \times m_6$.

The Fig.5 - Fig.7 show control errors at the robot motion along the testing trajectory. They represent the errors in Cartesian coordinate system. The values of the appropriate Cartesian coordinates were recomputed from ‘measured’ values of joint coordinates.

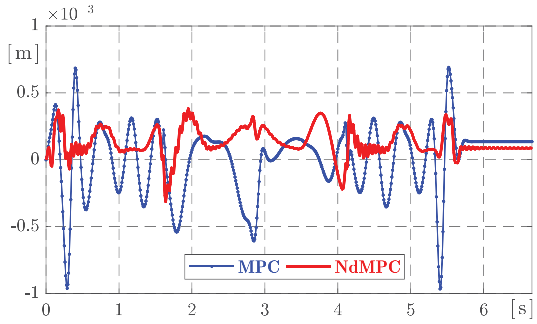


Figure 5: Time histories of errors in the axis x.

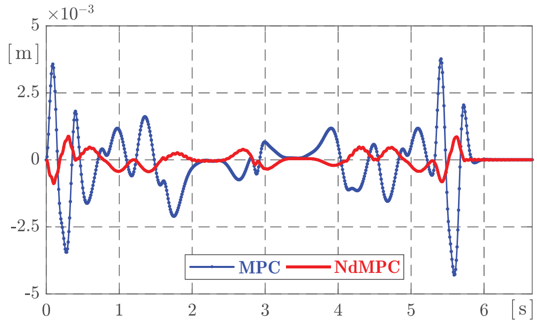


Figure 6: Time histories of errors in the axis y.

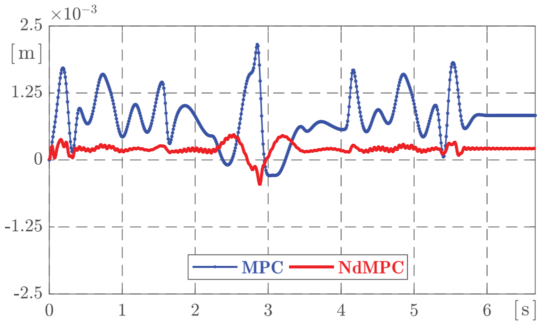


Figure 7: Time histories of errors in the axis z.

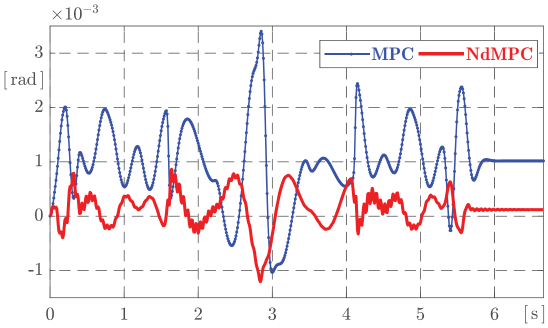


Figure 8: Time histories of control error for joint q_2.

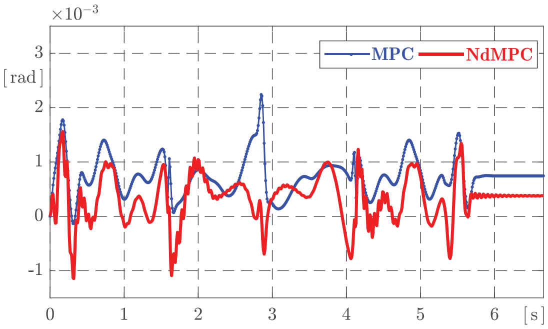


Figure 9: Time histories of control error for joint q_3.

The corresponding errors in the joint space for the most exposed joints q_2 and q_3 to load are shown in Fig. 8 and Fig. 9. The exposition is caused by the robot motion itself, thus its character across the symmetry plane $x_0 z_0|_{y_0=0}$. The mentioned figures show the lower tracking errors for the proposed method. Considering moving-mass distribution in the robot: $m_1 = 35\,\text{kg}$, $m_2 = 16\,\text{kg}$, $m_3 = 14\,\text{kg}$, $m_4 = 6\,\text{kg}$, $m_5 = 0.75\,\text{kg}$ and $m_6 = 0.25\,\text{kg}$, then the highest vertical load is on 2^{nd} link between joints q_2 and q_3, which is actuated in joint q_2.

For the selected trajectory or its orientation, the joint q_2 together with joint q_3 influence dominantly the motion in the direction parallel to axis x_0 and axis z_0 whereas joint q_1 (rotation around axis z_0) influences motion in the direction of axis y_0, especially if the motion trajectory leads in specific robot orientation through the mentioned vertical symmetry plane $x_0 z_0|_{y_0=0}$.

Since the both proposed NdMPC and regular MPC design have positional character, then specific steady-state errors are noticed. This is especially obvious for the vertical axis z (Fig. 7) and a bit less for the horizontal axis x (Fig. 5), which is coupled with the joints serving predominantly for the axis z, i.e. joints q_2 and q_3.

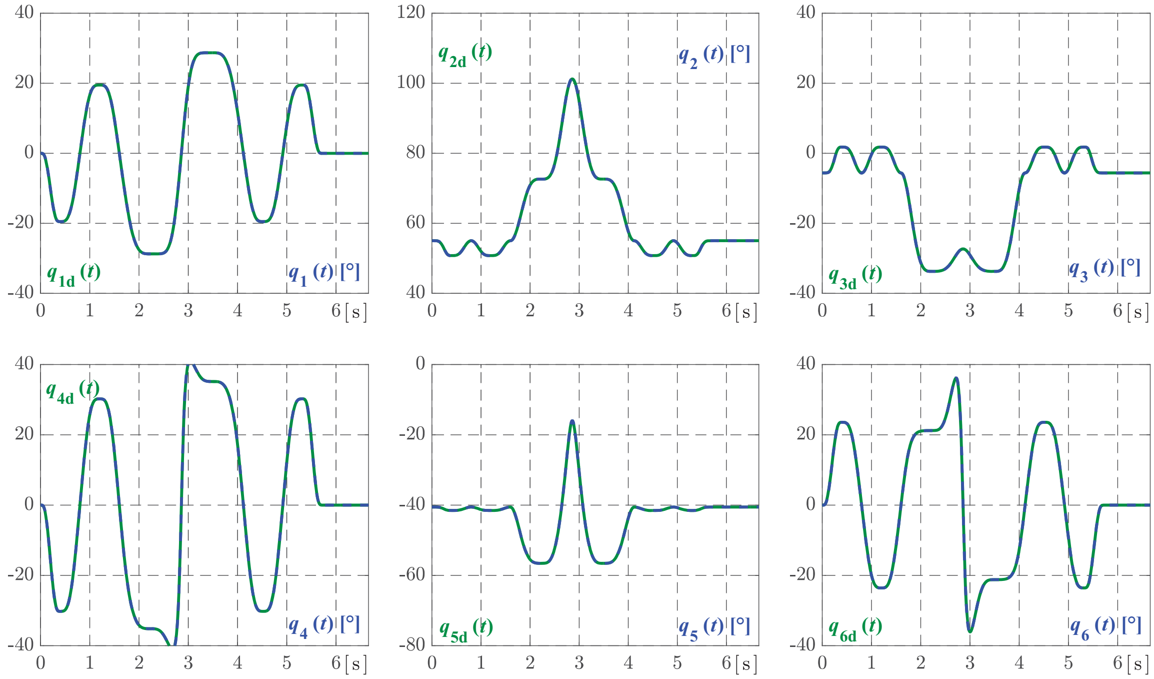


Figure 10: Time histories of joint coordinates q_i and their references q_{id}: generalized coordinates $q_{i(\cdot)}$, $i = 1, \cdots, 6$.

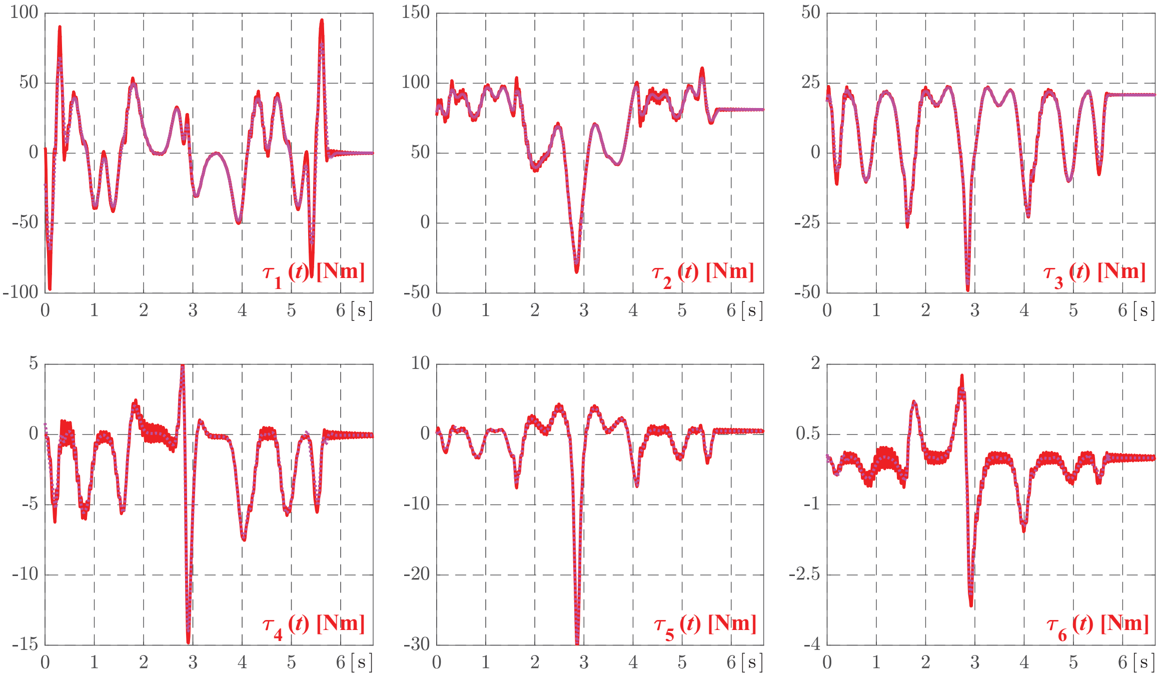


Figure 11: Time histories of control actions: joint torques τ_i, $i = 1, \cdots, 6$.

The Fig. 10 shows joint coordinates q_i corresponding to cartesian coordinates in Fig. 3. It is evident that the most difficult motion phase is around 2.9 s, because the robot arms and end-effector are in full motion speed and the trajectory decomposed into the individual joint angles changes rapidly.

Fig. 11 shows corresponding situation in designed control actions τ_i generated during control process. Such rapid turn or change cause changes in the model parameters, which cannot be expressed with one linearized model (41) fixed within respective moving time intervals at standard design (39) instead of varying nonlinear model along the same time intervals at proposed nonlinear design based on equations of predictions (25) and (26).

6 CONCLUSION

The proposed nonlinear design introduces specific direct use of initial nonlinear continuous model without any linearizaton and conventional model discretization. The introduced NdMPC can consider reasonable prediction horizon as regular MPC. Further emphasis will be placed on the selection analysis of adequate numerical method, incremental offset-free form and on a general point-to-point motion in unconstrained and constrained robot workspace.

ACKNOWLEDGEMENT

This paper was supported by The Czech Academy of Sciences, Institute of Information Theory and Automation.

REFERENCES

Arimoto, S. (1996). *Control Theory of Non-linear Mechanical Systems*. Oxford Press.

Belda, K., Böhm, J., and Píša, P. (2007). Concepts of model-based control and trajectory planning for parallel robots. In *Proc. of the 13th IASTED Int. Conf. on Robotics and Applications*, pages 15–20.

Belda, K. and Novotný, P. (2012). Path simulator for machine tools and robots. In *Proc. of MMAR IEEE Int. Conf.*, pages 373–378, Poland.

Belda, K. and Rovný, O. (2017). Predictive control of 5 DOF robot arm of autonomous mobile robotic system motion control employing mathematical model of the robot arm dynamics. In *21st Int.Conf. on Process Control*, pages 339–344.

Belda, K. and Vošmik, D. (2016). Explicit generalized predictive control of speed and position of PMSM drives. *IEEE Trns. Ind. Electron.*, 63(2):3889–3896.

Belda, K. and Záda, V. (2017). Predictive control for offset-free motion of industrial articulated robots. In *Proc. of MMAR IEEE Int. Conf.*, pages 688–693, Poland.

Butcher, J. (2016). *Numerical Methods for Ordinary Differential Equations*. Wiley.

Chung, W., Fu, L., and Kröger, T. (2016). *Motion Control*, pages 163–194. Springer.

Colombo, A. W., Karnouskos, S., Shi, Y., Yin, S., and Kaynak, O. (2016). Industrial cyber-physical systems scanning the issue. *Proc. of the IEEE*, 104(5):899–903.

Faulwasser, T., Weber, T., Zometa, P., and Findeisen, R. (2017). Implementation of nonlinear model predictive path-following control for an industrial robot. *IEEE Tran. Control Syst. Technology*, 25(4):1505–1511.

Franklin, G., Powell, J., and Workman, M. (1998). *Digital Control of Dynamic Systems (3rd ed.)*. Addison Wesley.

Groothuis, S., Stramigioli, S., and Carloni, R. (2017). Modeling robotic manipulators powered by variable stiffness actuators: A graph-theoretic and port-hamiltonian formalism. *IEEE Trans. on Robotics*, 33(4):807–818.

Houska, B., Ferreau, H., and Diehl, M. (2011). ACADO toolkit – An open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312.

Kirches, C. (2011). *Fast numerical methods for mixed-integer nonlinear model-predictive control*. Springer.

Kothare, M. V., Balakrishnan, V., and Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):13611379.

Lawson, C. and Hanson, R. (1995). *Solving least squares problems*. Siam.

Maciejowski, J. (2002). *Predictive Control: With Constraints*. Prentice Hall.

Negri, G. H., Cavalca, M. S. M., and Celeberto, A. L. (2017). Neural nonlinear model-based predictive control with faul tolerance characteristics applied to a robot manipulator. *Int. J. Innovative Computing Infomration and Control*, 13(6):1981–1992.

Ordis, A. and Clarke, D. (1993). A state-space description for GPC controllers. *J. Systems SCI.*, 24(9):1727–1744.

Ostafew, C. J., Schoellig, A. P., Barfoot, T. D., and Collier, J. (2016). Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *J. Field Robotics*, 33(1):133–152.

Othman, A., Belda, K., and Burget, P. (2015). Physical modelling of energy consumption of industrial articulated robots. In *Proc. of the 15th Int. Conf. on Control, Automation and Systems*, pages 784–789, Korea. ICROS.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics – Modelling, Planning and Control*. Springer.

SPS IPC Drives (2017). Internationale fachmesse für elektrische automatisierung, systeme und komponenten. Online: < https://www.mesago.de/en/SPS/ >.

Wang, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB*. Springer.

Wilson, J., Charest, M., and Dubay, R. (2016). Nonlinear model predictive control schemes with application on a 2 link vertical robot manipulator. *Robotics and Computer-Integrated Manufacturing*, 41:23–30.

Záda, V. and Belda, K. (2017). Application of hamiltonian mechanics to control design for industrial robotic manipulators. In *Proc. of MMAR IEEE Int. Conf.*, pages 390–395, Poland.

Zanelli, A., Domahidi, A., Jerez, J., and Morari, M. (2017). FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *Int. J. Control*, pages 1–17.