Nonlinear Model Predictive Control Algorithms 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

Abstract. This paper deals with a novel nonlinear design of the discrete model predictive control represented by two algorithms based on the features of linear methods for the numerical solution of ordinary differential equations. The design algorithms allow more accurate motion control of robotic or mechatronic systems that are usually modelled by nonlinear differential equations up to the second order. The proposed ways apply nonlinear models directly to the construction of equations of predictions employed in predictive control design. These equations are composed using principles of explicit linear multi-step methods leading to straightforward and unambiguous construction of the predictions. Examples of the noticeably improved behaviour of proposed ways in comparison with conventional linear predictive control are demonstrated by comparative simulations with the nonlinear model of six-axis articulated robot.

Keywords: Discrete model predictive control · Nonlinear design · Lagrange equations · Articulated robots

1 Introduction

The integration of industrial robot applications in production increases rapidly. Such continuous trend proceeds with the advent of Industry 4.0 and it will continue in future as well [10]. The robots in industrial production perform huge number of operations. Their efficiency depends on used motion control that can exploit available information from used robot and measured data [9].

Nowadays, necessary information, data and computing power are broadly available, however, the designers are not able to use them effectively. In industrial production, there exist a lot of elaborated strategies that follow from long-term, empirical experiences [24]. Unfortunately, such strategies are usually not general enough. They are not scalable or transferable for different or modified systems.

From mathematical point of view, the robots, manipulators, such as a mechanical structure of articulated robot class shown in Fig. 1, represent dynamic systems that are usually described by systems of ordinary differential

O. Gusikhin and K. Madani (Eds.): ICINCO 2018, LNEE 613, pp. 230–251, 2020.
https://doi.org/10.1007/978-3-030-31993-9_11

equations (ODEs) as suitable models of dynamics [23]. These models can be considered as a proper substitution of the real physical robot mechanism for computer simulations and motion control design as well.

The aforementioned ODE systems reflect various relations among individual elements of the robot constructions. These relations are usually nonlinear. It is given by nonlinear operations on descriptive variables and related derivatives [1]. Thus, usual linear control theory [25] cannot be directly applied without some modifications.

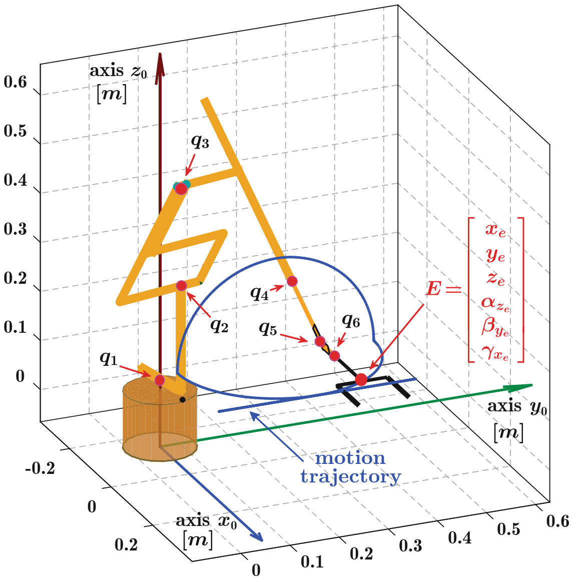


Fig. 1. Wire-frame model of the articulated robot class [2].

In practice, several common solutions are considered. They employ local linearization [21] by means of Taylor series, partial derivative models or switching local linear models and discretization to obtain discrete linear-like model, often in state-space form. Then, usual linear design of discrete model predictive control can be applied [3,5,6,18,20,25]. Further approaches consider linear models with stochastic uncertainties as substitution of initial nonlinear models [16]. Different solutions can be realized by neural network [19] or by the direct nonlinear optimization such as in [11,14,15,26,28]. However, mentioned ways are not immediately applicable as straightforward multi-step design in control process.

This paper deals with a novel way of the design of the discrete model predictive control (single-pass algorithm) that however employs a continuous-time nonlinear ODE model of the robot dynamics. This way was initially introduced in [2] and this paper extends its idea for a novel general way (two-cycle algorithm) using nonlinear prediction. The new algorithm serves as a specific etalon

belda@utia.cas.cz

or ideal for demonstrated fast single-pass algorithm. It is given by computation demands. The two-cycle algorithm is more demanding, slower but gives results near the ideal in comparison with the single-pass algorithm that is fast and fully comparable in computation demands with the conventional linear approach [25]. However, the conventional linear approach cannot achieve comparable results.

The model in both algorithms is employed directly by means of specifically adapted explicit linear multi-step numerical methods. These methods are used for the construction of equations of predictions or using of these methods substitutes conventional equations fully. Considered equations of predictions can be applied in usual way to the common quadratic cost function and optimization criterion. The explanation is introduced with Adams-Bashforth method as a representative of the aforementioned explicit methods [8]. Features of the proposed solutions are discussed and compared with usual linear design of model predictive control [20,25] that considers conventional repeated linearization and discretization along motion trajectories of the robot.

2 Nonlinear Robot Model

The model of the given class "articulated robots" is generally represented by a nonlinear function expressing 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) expresses equations of motions [22,23] that reflect 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 corresponding to generalized coordinates [23].

The individual elements of (2) are defined as follows

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

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

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

belda@utia.cas.cz

where, with specific 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.

Then, the model (the equations of motion of articulated robots) can be written as follows

$$
\begin{aligned}
\ddot{q} &= \overbrace{\underbrace{\underbrace{-H^{-1}\left(\frac{1}{2}\tilde{H} + S\right)\dot{q}}_{f(q,\dot{q})}\ \underbrace{-H^{-1}g}_{f_g(q)}}_{f_c(q,\dot{q})}\ \underbrace{+H^{-1}\tau}_{g_\tau(q)\,\tau}}^{\mathtt{f}(q,\dot{q},\tau)} \\
&= f(q,\, \dot{q}) \ + \ u
\end{aligned}
\tag{6}
$$

where $f(q,\, \dot{q}) = -H^{-1}\left(\frac{1}{2}\tilde{H} + S\right)\dot{q}$ and $u = H^{-1}(-g + \tau)$.
Thus, Eq. (6) can be considered as a particular form of the model (1):

$$
\begin{aligned}
\ddot{q} &= f_c(q, \dot{q}) \ + \ g_\tau(q)\,\tau = f(q, \dot{q}) + f_g(q) \ + \ g_\tau(q)\,\tau \\
&= \ f(q, \dot{q}) \ + \ g(q)\,u
\end{aligned}
\tag{7}
$$

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

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

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

i.e. backward recomputation of designed control actions to real torques including a compensation of effects of gravity.

3 Integration Concept

To design nonlinear predictive control in discrete form, let us consider individual time instants of the initial continuous nonlinear function in the model (1) as follows (i.e. only time sampling, but 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 appropriately selected sampling period. Let the same be applied to the terms from (7)

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

belda@utia.cas.cz

The propositions (9) or (10) will be apt for a specific construction of the predictions towards unknown control actions $u(t)$ within in an ordered finite set of discrete time instants $t \in \{k\,T_s,\,(k+1)\,T_s,\,\cdots,(k+N-1)\,T_s\}$, where N is a prediction horizon. The predictions still take into account the continuous nonlinear model, but only the indicated discrete time samples that will be applied to discrete design of predictive control. This concept can be realised by means of numerical methods [8] used for the numerical approximation of the solution of the first-order ordinary differential equations (ODEs):

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

Thus, these methods are used to find a numerical approximation of the exact integral over a particular 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 given time interval, $\hat{y}_{k+1}$ is an approximation of the exact solution $y_{k+1} = y|_{t=(k+1)\,T_s}$, $\delta(t,y)$ means in general 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 a large number of the methods, let us take into account linear multi-step methods that are convenient for the predictions in predictive design. Generally, linear multi-step methods are expressed as follows

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

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

For the design, the explicit methods are useful. One their representative is 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

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

Note, for completeness, that 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 [8] will be used explicitly in an explanation of the single-pass algorithm in the following section.

4 Nonlinear Design of Model 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 [11,28]. Alternatively, common additive forms of the criterion employed in linear control theory can be considered as well. The operation of integration is just moved from the criterion towards predictions composed by means of the numerical methods for ODEs involving the continuous-time model (7). This idea will be introduced and employed in the proposed design.

Let us start just from the continuous model (7) considered in the discrete time samples (time instants) as was indicated in (10). Moreover, let us take more usual, universal notation into account: instead of q use for outputs symbol y, 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 given spatial articulated robot class as well as specially for vertical planar robot configurations. Note, for completeness, $f_{g\,k}|_{\,y \in R} = 0$ applies to horizontal planar robot configurations. The knowledge of the property of f_k is useful for maintaining the stability in the control design.

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

4.1 Criterion and Cost Function

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

$$\min_{U_k} \; J_k\;(\,\hat{Y}_{k+1}, W_{k+1}, U_k\,) \tag{16}$$

$$\text{subject to: } \hat{Y}_{k+1} = f(y_k,\, \dot{y}_k,\, \delta(t, y, \dot{y}))$$

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

where the function $\delta(t, y, \dot{y})$ arises from the second order model $\ddot{y}_k = f_k + g_k\, u_k$, and $f_k = f(y, \dot{y})$ as was shown in (10). Vectors $\hat{Y}_{k+1}$, W_{k+1} and U_k represent the sequences of the robot output predictions, references and control actions from given current time sample up to the horizon of prediction N, respectively

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

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

$$U_k = [u_k^T, \cdots, u_{k+N-1}^T]^T \tag{19}$$

belda@utia.cas.cz

Thereafter, the cost function J_k is chosen in usual quadratic form as follows

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

where $Q_{YW}^T\, Q_{YW}$ and $Q_U^T\, Q_U$ represent penalizations defined by the following matrix form

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

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

However, the cost function can be selected differently according to user requirements, e.g. considering incremental terms that can slightly moderate and smooth the robot motion [18] or suppress steady-state control error, as shown in [7,25]. It means that cost function (20) may also be selected in some specific incremental form from control action point of view, for instance, as follows

$$J_k = (\hat{Y}_{k+1} - W_{k+1})^T Q_{YW}^T\, Q_{YW}\,(\hat{Y}_{k+1} - W_{k+1}) + \triangle U_k^T\, Q_{\triangle U}^T\, Q_{\triangle U}\, \triangle U_k \tag{22}$$

However, a minimization of the cost functions may be provided by similar optimization procedure. Such procedure will be introduced at the end of this section.

4.2 Equations of Predictions – Algorithm with Nonlinear Prediction

The algorithm with nonlinear prediction (two-cycle algorithm) is developed to improve accuracy of predicted outputs $\hat{y}$ by specific approximation of nonlinearities of the robot model by specific equations of predictions involving specific nonlinear prediction. The algorithm consists of two cycles as follows from Fig. 2.

The main cycle represents usual control cycle extended by internal simulation cycle. Internal cycle composes the essential part of the control $u_{sim\,k}$ and predicts future values of the system state $-x_{sim\,k:k+N}$. For these operations, the general predictive control algorithm and numerical integration of nonlinear model (i.e. function $f(x) + g(x)\,u$ $(= \mathtt{f}(x,\, u))$ are used. Obtained values are stored for computations in main cycle. In it, at first, the models determined by states $x_{sim\,k:k+N}$ are calculated. Then, they are used for composition of equations of predictions serving for generating of increments of control actions. Finally, the increments are added to actions from internal cycle $u_{sim\,k}$. Final control actions u_k are applied to controlled systems. Now, this generally defined algorithm can be mathematically formulated.

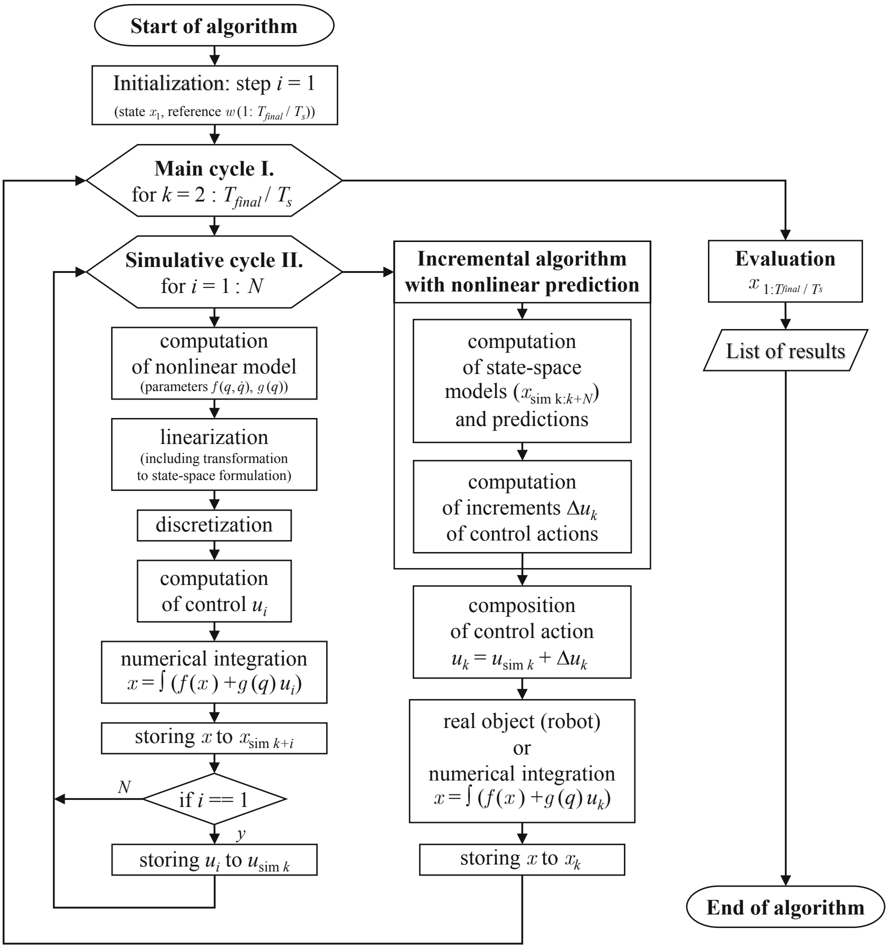


Fig. 2. Flowchart of the algorithm with nonlinear prediction.

In detail, the specific equations of predictions are constructed as follows

$$
\begin{aligned}
x_{k+1} &= \underbrace{A_k x_k + B_k u_{sim\,k}} + B_k \triangle u_k \\
\hat{x}_{k+1} &= \quad x_{sim\,k+1} \quad + B_k \triangle u_k \\
x_{k+2} &= A_{k+1} x_{k+1} + B_{k+1} u_{sim\,k+1} + B_{k+1} \triangle u_k \\
\hat{x}_{k+2} &= \underbrace{A_{k+1} x_{sim\,k+1} + B_{k+1} u_{sim\,k+1}} + A_{k+1} B_k \triangle u_k + B_{k+1} \triangle u_{k+1} \\
\hat{x}_{k+2} &= \qquad x_{sim\,k+2} \qquad + A_{k+1} B_k \triangle u_k + B_{k+1} \triangle u_{k+1} \\
\vdots \quad & \quad \vdots \\
\hat{x}_{k+N} &= x_{sim\,k+N} + A_{k+N-1} \cdots A_{k+1} B_k \triangle u_k \\
&\quad + \cdots + A_{k+N-1} B_{k+N-2} \triangle u_{k+N-2} + B_{k+N-1} \triangle u_{k+N-1}
\end{aligned}
\tag{23}
$$

In (23), $x_{sim\,k+1}$, $u_{sim\,k}$ (etc. $x_{sim\,k+i}$, $u_{sim\,k+i-1}, i = 1, \cdots, N$) are vectors of state and control actions given from pre-simulation in considered receding horizon of incremental control algorithm. Furthermore, $A_k = A(x_k)$, $B_k = B(x_k), A_{k+i} = A(x_{sim\,k+i}), B_{k+i} = B(x_{sim\,k+i}), i = 1, \cdots, N-1$ are matrices of state-space model given by model linearization and discretization repeating in every time instant k. C is an output matrix. All terms follow from state-space form:

$$\underbrace{\begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & I \\ 0 & f(\dot{y}, y) \end{bmatrix}}_{A(t)} \underbrace{\begin{bmatrix} y \\ \dot{y} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_{B} u \tag{24}$$

$$y = \underbrace{\begin{bmatrix} I & 0 \end{bmatrix}}_{C} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \tag{25}$$

that is discretized: $A(t),\ B\,|_{T_s} \ \Rightarrow\ A_k,\ B_k$ by first-order-hold method [12] and arranged as

$$\hat{y}_{k+1} = C A_k x_k + C\, B_k\ u_k \tag{26}$$

The derivation of the Eq. (23) follows from usual construction of equations of predictions, i.e. suitably repetitive substitution for predicted future states and outputs, respectively. However, the structure of the equations of predictions is more complex due to variable state-space matrices within optimisation interval given be horizon of prediction N.

$$\begin{aligned} \begin{bmatrix} \hat{y}_{k+1} \\ \hat{y}_{k+2} \\ \vdots \\ \hat{y}_{k+N} \end{bmatrix} &= \begin{bmatrix} C x_{sim\,k+1} \\ C x_{sim\,k+2} \\ \vdots \\ C x_{sim\,k+N} \end{bmatrix} \\ &+ \begin{bmatrix} C B_k & \cdots & 0 \\ C A_{k+1} B_k & & \vdots \\ \vdots & \ddots & 0 \\ C A_{k+N-1} \cdots A_{k+1} B_k & \cdots & C B_{k+N-1} \end{bmatrix} \\ &\times \begin{bmatrix} \triangle\, u_k \\ \triangle\, u_{k+1} \\ \vdots \\ \triangle\, u_{k+N-1} \end{bmatrix} \end{aligned} \tag{27}$$

A corresponding condensed matrix notation is as follows

$$\begin{aligned}\hat{Y}_{k+1} &= \hat{Y}_{sim\,k+1} \\ &+ \bar{G}_k \\ &\times \triangle U_k \left| \begin{aligned} \hat{Y}_{k+1} &= [\hat{y}_{k+1}, \hat{y}_{k+2}, \cdots, \hat{y}_{k+N}]^T \\ \triangle U &= [\triangle u_k, \triangle u_{k+1}, \cdots, \triangle u_{k+N-1}]^T \end{aligned} \right.\end{aligned} \tag{28}$$

$$\hat{Y}_{k+1} = \bar{F}_k + \bar{G}_k \ \triangle U_k \tag{29}$$

where vector $\bar{F}_k = \hat{Y}_{sim\,k+1}$ and matrix $\bar{G}_k$ follow from the structure (27).

4.3 Equations of Predictions – Single-Pass Algorithm

As was already mentioned, the equations of predictions can also 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). It can be ensured by the exemplarily selected linear multi-step Adams-Bashforth method of fourth order (14) for the solution of the first-order ODEs.

However, the considered nonlinear model of the robot (15) represents a system of the second-order ODEs. To apply the chosen suitable numerical method, but without loss of information about included nonlinear relations, the model (15) has to be specifically transformed into ODEs of the first order. For such necessary rearrangement, the following backward Euler formula can be applied

$$\hat{\dot{y}}_{k+1} = \frac{\hat{y}_{k+1} - y_k}{h} \tag{30}$$

This specific formula ensures the coupling (propagation) of nonlinear relations from the model (15) into positional estimates, since the numerical methods represent only linear combinations within rows of the ODEs. Hence, usual rearrangement via addition of further ODEs decreasing the order of initial ODE system would not be useful because of the loss of information.

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

$$\hat{y}_{k+1} = y_k + h\,\hat{\dot{y}}_{k+1} \tag{31}$$

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

$$\hat{\dot{y}}_{k+1} = \dot{y}_k + h\,\delta(t,\, y,\, \dot{y}) \tag{32}$$

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 [13,27], this step is not needed.

belda@utia.cas.cz

Now, the real equations of predictions can be composed considering the initial nonlinear model (15), together with a specific numerical method (32) (here, the Adams-Bashworth method (14)) and the transformation to the first-order ODE set as indicated by (31).

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

$$\hat{\dot{Y}}_{k+1} = \dot{y}_k\, F_I + F_k + G_k\, U_k \tag{33}$$

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

$$\hat{Y}_{k+1} = y_k\, F_I + L_k + M_k\, U_k \tag{34}$$

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 defined using the following sequences for velocities, where, for clear arrangement, particular time instants are separated by horizontal lines:

$$
\begin{aligned}
\hat{\dot{y}}_{k+1} &= \dot{y}_k + F_{1,k} \\
&\qquad + \beta_0\, g_k\, u_k \\
F_{1,k} &= \beta_3\, \ddot{y}_{k-3} + \beta_2\, \ddot{y}_{k-2} + \beta_1\, \ddot{y}_{k-1} + \beta_0\, f_k \\
\hline
\hat{\dot{y}}_{k+2} &= \dot{y}_k + F_{2,k} \\
&\qquad + (\beta_1 + \beta_0)\, g_k\, u_k + \beta_0\, \hat{g}_{k+1}\, u_{k+1} \\
F_{2,k} &= F_{1,k} + \beta_3\, \ddot{y}_{k-2} + \beta_2\, \ddot{y}_{k-1} + \beta_1\, f_k + \beta_0\, \hat{f}_{k+1} \\
\hline
&\ \vdots \\
\hline
\hat{\dot{y}}_{k+N} &= \dot{y}_k + F_{N,k} + \{\sum_{j=0}^{3} \beta_j\}\, g_k u_k \\
&\qquad + \cdots + \beta_0\, \hat{g}_{k+N-1}\, u_{k+N-1} \\
F_{N,k} &= F_{N-1,k} + \sum_{j=1}^{4} \beta_{j-1} \hat{f}_{k+N-j}
\end{aligned} \tag{35}
$$

Note that in step k, the topical value $\dot{y}_k$ as well as its appropriate past values $\ddot{y}_{k-1}$, $\ddot{y}_{k-2}$ and $\ddot{y}_{k-3}$ are known from measurements or possibly from some suitable running state estimation. The coefficients β_j appearing in (35) are identical to the coefficients β_j that were introduced in the Eq. (13).

Consequently, vector F_k and matrix G_k from (33) can be defined as follows:

$$F_k = \begin{bmatrix} F_{1,\,k} \\ F_{2,\,k} \\ \vdots \\ F_{N,\,k} \end{bmatrix}, \quad G_k = \begin{bmatrix} \beta_0\, g_k & 0 & \cdots & 0 \\ (\beta_1 + \beta_0)\, g_k & \beta_0\, \hat{g}_{k+1} & & \vdots \\ \vdots & & \ddots & \vdots \\ \{\sum_{j=0}^{3} \beta_j\}\, g_k & \cdots & \cdots & \beta_0\, \hat{g}_{k+N-1} \end{bmatrix} \tag{36}$$

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

Similarly in the construction of the Eqs. (34), $\hat{Y}_{k+1}$, L_k and M_k can be defined by analogical sequences for positions as follows (again for clear arrangement, particular time instants are separated by horizontal lines):

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

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

$$\vdots$$

$$\begin{aligned} \hat{y}_{k+N} &= y_k + L_{N,\,k} \\ &\qquad + h\, \{\sum_{j=0}^{3} (N-j)\beta_j\}\, g_k\, u_k \\ &\qquad + \cdots + h\beta_0\, \hat{g}_{k+N-1}\, u_{k+N-1} \\ L_{N,\,k} &= L_{N-1,\,k} + h\, F_{N,\,k} \end{aligned} \tag{37}$$

Then, vector L_k and matrix M_k from (34) are:

$$L_k = \begin{bmatrix} L_{1,\,k} \\ L_{2,\,k} \\ \vdots \\ L_{N,\,k} \end{bmatrix}, M_k = \begin{bmatrix} h\, \beta_0\, g_k & 0 & \cdots & 0 \\ h\,(\beta_1 + 2\,\beta_0)\, g_k & h\, \beta_0\, \hat{g}_{k+1} & & \vdots \\ \vdots & & \ddots & \vdots \\ h\{\sum_{j=0}^{3} (N-j)\beta_j\} g_k & \cdots & \cdots & h\, \beta_0\, \hat{g}_{k+N-1} \end{bmatrix} \tag{38}$$

Note that the two-step equations of predictions (33) and (34) are used here with respect to the second order of ODEs.

4.4 Square-Root Minimization

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

$$\min_{U_k} J_k = \min_{U_k} \mathbb{J}_k^T \mathbb{J}_k \;\Rightarrow\; \min_{U_k} \mathbb{J}_k \tag{39}$$

that indicates the square-root minimization of the vector $\mathbb{J}_k$ instead of minimization of the scalar J_k. The reason is that minimizing the square-root is more suitable in terms of calculation. Thus, the square-root of the criterion (16) with the cost function (20) can be expressed as follows

$$\min_{U_k} \mathbb{J}_k = \min_{U_k} \begin{bmatrix} Q_{YW} & 0 \\ 0 & Q_U \end{bmatrix} \begin{bmatrix} \hat{Y}_{k+1} - W_{k+1} \\ U_k \end{bmatrix} \tag{40}$$

The indicated minimization (40) can be solved as a specific least-squares problem by the following system of algebraic equations [17] that involves equations of predictions (34) for $\hat{Y}_{k+1}$

$$\begin{bmatrix} Q_{YW} M_k \\ Q_U \end{bmatrix} U_k = \begin{bmatrix} Q_{YW}\,(W_{k+1} - y_k\,F_I - L_k) \\ 0 \end{bmatrix} \tag{41}$$

A similar least-squares problem can also be written specifically for the cost function (22) used in the two-cycle algorithm with nonlinear prediction (29):

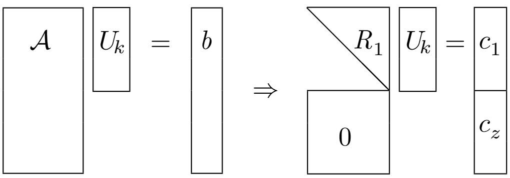
$$\begin{bmatrix} Q_{YW} \bar{G}_k \\ Q_U \end{bmatrix} \triangle U_k = \begin{bmatrix} Q_{YW}\,(W_{k+1} - \bar{F}_k) \\ 0 \end{bmatrix} \tag{42}$$

The system (41) or (42), that is over-determined, can be written in condensed general form (43). It can be transformed to another form (44) by orthogonal-triangular decomposition [17] and solved for unknown U_k (or $\triangle U_k$)

$$\mathcal{A} U_k = b \tag{43}$$

$$\begin{aligned} Q^T \mathcal{A} U_k &= Q^T b \text{ assuming that } \mathcal{A} = Q\,R \\ R_1 U_k &= c_1 \end{aligned} \tag{44}$$

where Q^T is an orthogonal matrix that transforms matrix $\mathcal{A}$ into upper triangle R_1 as it is indicated by the following equation diagram



(45)

Vector c_z represents a loss vector, Euclidean norm $||c_z||$ of which equals to the square-root of the optimal cost function minimum, i.e. scalar value $\sqrt{J}$, where $J = c_z^T c_z$. Only the first elements corresponding to u_k are used from computed vector U_k . Note that for $\triangle\, U_k$ it is necessary to add the control action value $u_{sim\,k}$ from internal cycle as follows: $u_k = u_{sim\,k} + \triangle\, u_k$, where $\triangle\, u_k$ is the appropriate first vector element from $\triangle U_k$ obtained from the main cycle.

5 Simulation Examples

The examples demonstrate the behaviour of the articulated robot along a selected testing trajectory. The corresponding wire-frame model of the given robot including trajectory is shown in Fig. 1. Trajectory in detail is in Figs. 3 and 4. The depicted trajectory was time parameterized with acceleration polynomial of fifth-order [4,23]. The specification of individual trajectory segments is listed in the Table 1, where G19, G01 and G02 are G-codes of plane selection for full definiteness of circles, motion with linear and circular interpolation, respectively.

The model of given robot dynamics (7) and (8) from (2)–(6) took into account the parameters of ABB robot IRB 140 (Fig. 5). 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 adequate Cartesian coordinates:

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

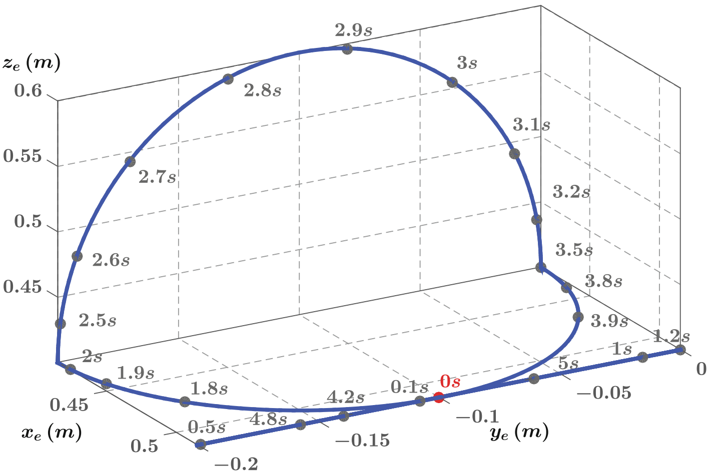


Fig. 3. Testing trajectory with specific time marks [2].

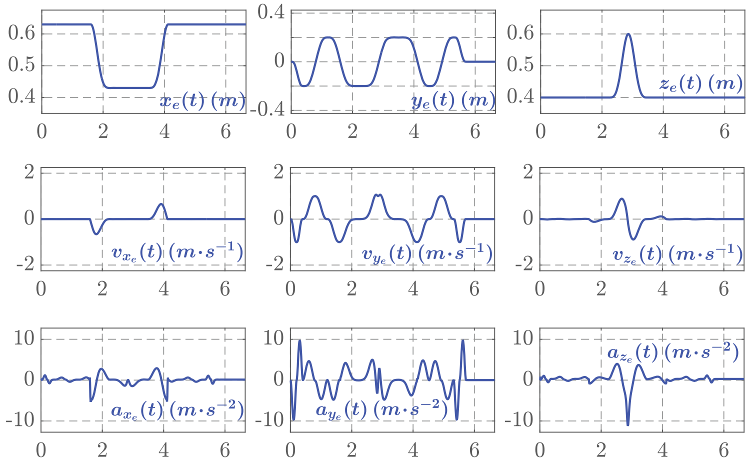


Fig. 4. Cartesian coordinates and derivatives (time in (s)) [2].

Table 1. Testing trajectory in G code (mm).

001 :	N010	G19						
002 :	N020	G01	X630	Y-200	Z400			
003 :	N030	G01	X630	Y200	Z400			
004 :	N040	G01	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	G01	X630	Y-200	Z400			
009 :	N090	G01	X630	Y200	Z400			
010 :	N010	G01	X630	Y0	Z400			

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 respective 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 time-varying according to appropriate inverse kinematic transformations [23], specific for the considered robot.

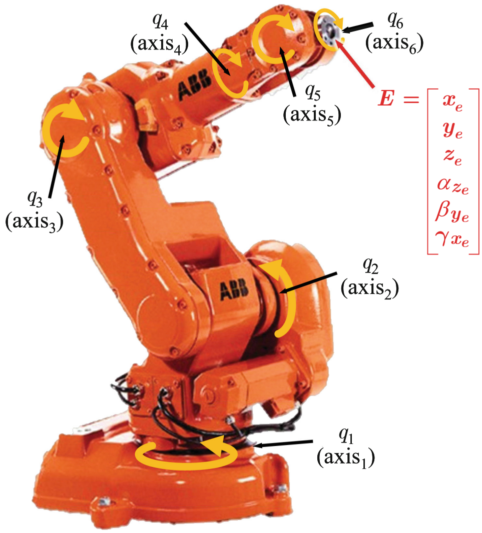


Fig. 5. Six-axis multipurpose ABB robot IRB 140 [2].

5.1 Simulation Setup

Introduced nonlinear design of model predictive control, i.e. two-cycle algorithm (PreSim MPC) and the fast single-pass algorithm (NonLin MPC), was tested with the following parameters:

- sampling period: $Ts = 0.01\,\mathrm{s}$
- horizon of prediction: $N = 10$
- output penalization: $Q_{yw} = I_{(6\times 6)}$
- input penalization: $Q_u = 2 \cdot 10^{-4}\, I_{(6\times 6)}$

where I is the identity matrix.

The both algorithms were compared with normal model predictive control (Normal MPC, MPC) [25] having identical setting involved in its equation:

$$U_k = (\tilde{G}_k^T Q_{YW}^T\, Q_{YW} \tilde{G}_k + Q_U^T\, Q_U)^{-1} \times \tilde{G}_k^T Q_{YW}^T\, Q_{YW} (W_{k+1} - \tilde{F}_k\, x_k) \qquad (46)$$

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

$$\tilde{F}_k = \begin{bmatrix} CA_k \\ \vdots \\ CA_k^N \end{bmatrix}, \qquad \tilde{G}_k = \begin{bmatrix} CB_k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ CA_k^{N-1} B_k & \cdots & CB_k \end{bmatrix} \qquad (47)$$

The details on used normal MPC can be found e.g. in [20, 25].

5.2 Summary of the Results

The comparative simulations were performed with the aforementioned setting and with artificially added mismatch between the model used for control design and the model for the simulation. The mismatch consisted in the four-times increased weight of the last, the sixth robot link in the simulation model against model for the design, i.e. $m_6 = 0.25\,\mathrm{kg}$ and $\tilde{m}_6 = 4 \times m_6$.

The Figs. 6, 7 and 8 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 determined from 'measured' values of joint coordinates by appropriate direct kinematic transformations [23].

The corresponding errors in the joint space for the most exposed joints q_2 and q_3 to load are shown in Figs. 9 and 10. The exposition is caused by the robot motion itself or its character across the symmetry plane $x_0z_0|_{y_0=0}$. The aforementioned figures show the lower tracking errors for the both proposed algorithms. Considering moving-mass distribution in the robot: $m_1 = 35\,\mathrm{kg}$, $m_2 = 16\,\mathrm{kg}$, $m_3 = 14\,\mathrm{kg}$, $m_4 = 6\,\mathrm{kg}$, $m_5 = 0.75\,\mathrm{kg}$ and $m_6 = 0.25\,\mathrm{kg}$, then the highest vertical load is on the second link between joints q_2 and q_3, which is actuated in joint q_2.

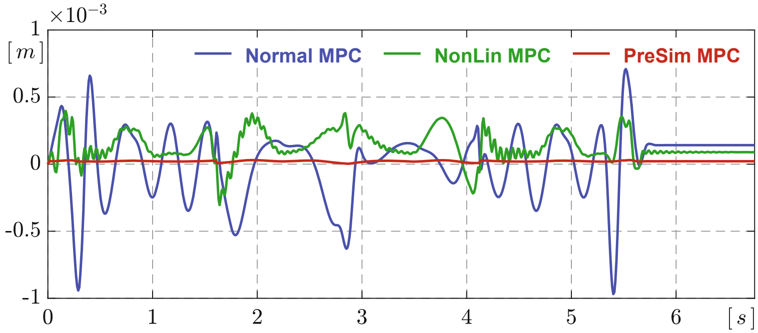


Fig. 6. Time histories of errors in the axis x.

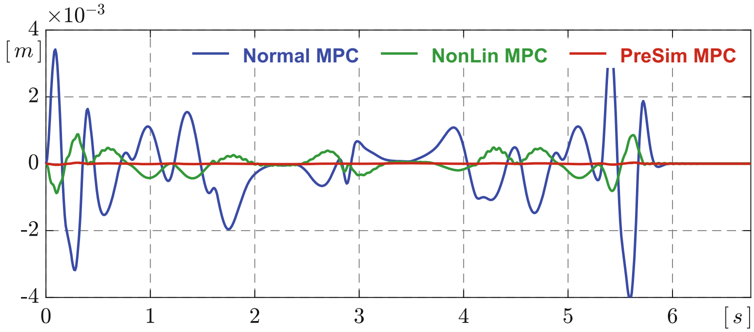


Fig. 7. Time histories of errors in the axis y.

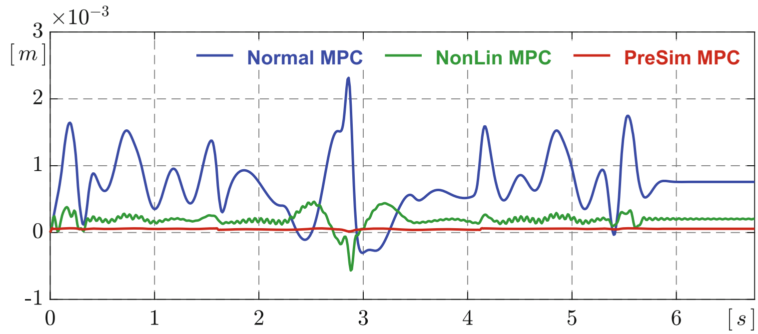


Fig. 8. Time histories of errors in the axis z.

For the chosen 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 the motion in the direction of axis y_0, especially if the motion trajectory leads to a specific robot orientation through the mentioned vertical symmetry plane $x_0z_0|_{y_0=0}$.

Since the both proposed algorithms as well as normal MPC design have positional character, then specific steady-state errors are perceptible. This is especially obvious for the vertical axis z (Fig. 8) and a bit less for the horizontal axis x (Fig. 6), which is coupled with the joints serving predominantly for the motion in the direction of axis z, i.e. joints q_2 and q_3.

The Fig. 11 shows joint coordinates q_i corresponding to Cartesian coordinates in Fig. 4. 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 varies rapidly. Figure 12 shows corresponding situation in designed control actions τ_i generated during control process.

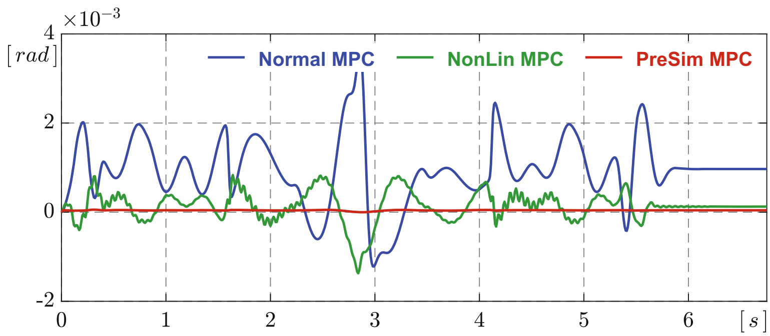


Fig. 9. Time histories of control error for joint q_2.

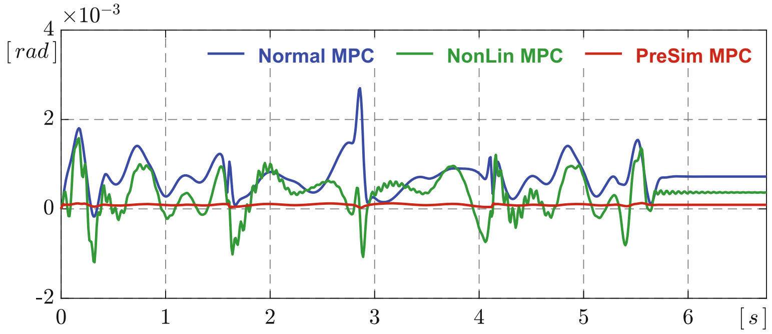


Fig. 10. Time histories of control error for joint q_3.

Such rapid turn or change cause variations in the model parameters, which cannot be expressed with one linearized model (24) fixed within respective moving time intervals at standard design (46) instead of flexible varying nonlinear model along the same time intervals at proposed algorithms based on equations of predictions (29) or (33) and (34), respectively.

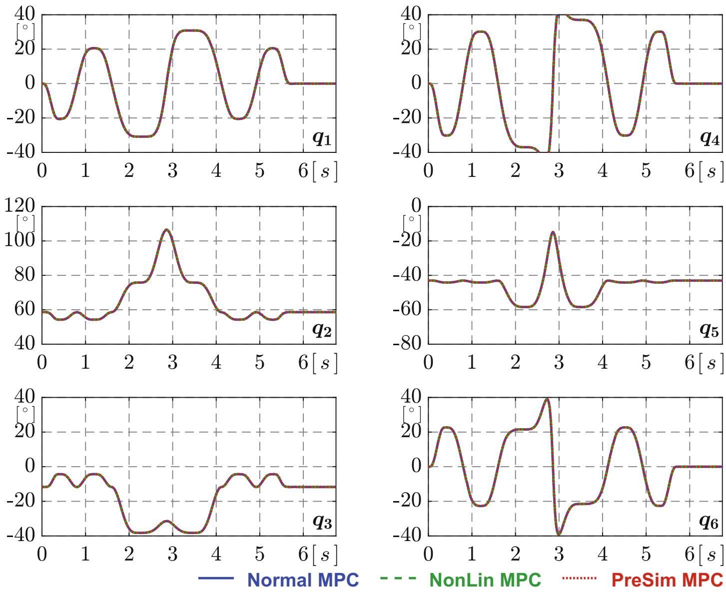


Fig. 11. Time histories of joint coordinates: q_i, $i = 1, \cdots, 6$.

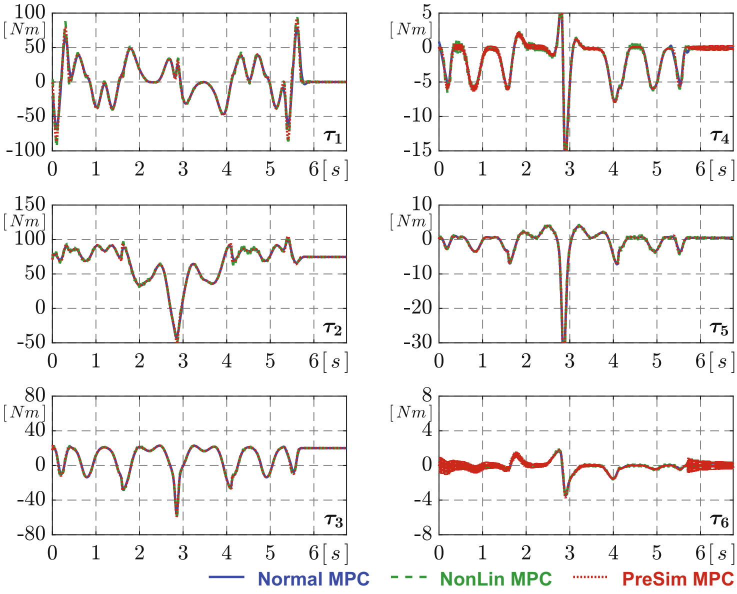


Fig. 12. Time histories of control actions, joint torques: τ_i, $i = 1, \cdots, 6$.

6 Conclusion

The proposed nonlinear design algorithms are characterized by a specific straightforward use of initial nonlinear continuous model in the construction of predictions. For the fast single-pass algorithm, the design is fully without any linearization and conventional model discretization. The introduced algorithms can consider reasonable prediction horizon as normal MPC. However, they can offer more accurate tracking the desired motion trajectories in comparison with the use of conventional linear control approaches containing linearization of used nonlinear models.

The emphasis in further research will be placed on the selection analysis of adequate numerical method, incremental prediction forms for offset-free motion 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

1. Arimoto S (1996) Control theory of non-linear mechanical systems. Oxford Press, New York

2. Belda K (2018) 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, vol 2, pp 71–80
3. Belda K, Böhm J, Píša P (2007) Concepts of model-based control and trajectory planning for parallel robots. In: Proceedings of the 13th IASTED international conference on robotics and applications, pp 15–20
4. Belda K, Novotný P (2012) Path simulator for machine tools and robots. In: Proceedings of MMAR IEEE international conference, Poland, pp 373–378
5. Belda K, 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 international conference on process control, pp 339–344
6. Belda K, Vošmik D (2016) Explicit generalized predictive control of speed and position of PMSM drives. IEEE Trns Ind Electron 63(2):3889–3896
7. Belda K, Záda V (2017) Predictive control for offset-free motion of industrial articulated robots. In: Proceedings of MMAR IEEE international conference, Poland, pp 688–693
8. Butcher J (2016) Numerical methods for ordinary differential equations. Wiley, New York
9. Chung W, Fu L, Kröger T (2016) Motion control. Springer, pp 163–194
10. Colombo AW, Karnouskos S, Shi Y, Yin S, Kaynak O (2016) Industrial cyber-physical systems - scanning the issue. Proc IEEE 104(5):899–903
11. Faulwasser T, Weber T, Zometa P, Findeisen R (2017) Implementation of nonlinear model predictive path-following control for an industrial robot. IEEE Tran Control Syst Technol 25(4):1505–1511
12. Franklin G, Powell J, Workman M (1998) Digital control of dynamic systems, 3rd edn. Addison Wesley, Menlo Park
13. Groothuis S, Stramigioli S, Carloni R (2017) Modeling robotic manipulators powered by variable stiffness actuators: a graph-theoretic and port-hamiltonian formalism. IEEE Trans Rob 33(4):807–818
14. Houska B, Ferreau H, Diehl M (2011) ACADO toolkit - an open-source framework for automatic control and dynamic optimization. Optimal Control Appl Methods 32(3):298–312
15. Kirches C (2011) Fast numerical methods for mixed-integer nonlinear model-predictive control. Springer
16. Kothare MV, Balakrishnan V, Morari M (1996) Robust constrained model predictive control using linear matrix inequalities. Automatica 32(10):1361–1379
17. Lawson C, Hanson R (1995) Solving least squares problems. Siam
18. Maciejowski J (2002) Predictive control: with constraints. Prentice Hall, London
19. Negri GH, Cavalca MSM, Celeberto AL (2017) Neural nonlinear model-based predictive control with faul tolerance characteristics applied to a robot manipulator. Int J Innovative Comput Inf Control 13(6):1981–1992
20. Ordis A, Clarke D (1993) A state-space description for GPC controllers. J Syst SCI 24(9):1727–1744
21. Ostafew CJ, Schoellig AP, Barfoot TD, Collier J (2016) Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. J Field Rob 33(1):133–152
22. Othman A, Belda K, Burget P (2015) Physical modelling of energy consumption of industrial articulated robots. In: Proceedings of the 15th international conference on control, automation and systems. ICROS, Korea, pp 784–789
23. Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) Robotics - modelling, planning and control. Springer, London

24. SPS IPC Drives (2018) Internationale fachmesse für elektrische automatisierung, systeme und komponenten, November 2018. https://www.mesago.de/en/SPS/
25. Wang L (2009) Model predictive control system design and implementation using MATLAB®. Springer, London
26. Wilson J, Charest M, Dubay R (2016) Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator. Rob Comput Integr Manuf 41:23–30
27. Záda V, Belda K (2017) Application of hamiltonian mechanics to control design for industrial robotic manipulators. In: Proceedings of MMAR IEEE international conference, Poland, pp 390–395
28. Zanelli A, Domahidi A, Jerez J, Morari M (2017) FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. Int J Control, pp 1–17