Predictive Control for Offset-Free Motion of Industrial Articulated Robots

Květoslav Belda
Department of Adaptive Systems
Institute of Information Theory and Automation of the CAS
Pod Vodárenskou věží 4, 182 08 Prague 8, Czech Republic
E-mail: belda@utia.cas.cz

Václav Záda
Department of Electromechanical Systems and Robotics
Institute of Mechatronics and Engineering Informatics
FM, TUL, Studentská 2, 461 17 Liberec, Czech Republic
E-mail: vaclav.zada@tul.cz

***Abstract*—This paper deals with the design of model predictive control for the precise motion of industrial articulated robots. The solution is based on specific incremental formulations of equations of predictions. The proposed formulations enable the design to compensate and suppress undesirable positional offsets. The corresponding incremental predictive algorithms incorporating discrete integrators are introduced. The theoretical results are demonstrated by the set of simulation examples with the six-axis multipurpose ABB robot IRB 140 that belongs to the large class of industrial articulated robots.**

I. Introduction

A modern industrial production includes a huge number of manipulation operations performed by robots-manipulators. The quality and efficiency of that operations is highly dependent on the motion control algorithms implemented in used robots. The usual motion control employs a set of independent firmly set PID controllers related to actuated joints of given robot [1]. This configuration can perform required motion but without any optimization considering whole robot structure. Individual PID controllers evaluate simply only control errors between individual reference coordinates of reference motion and corresponding measured real coordinates.

A different way of motion control of the robots, recently under research and development, is model-based control design represented by predictive control [2], [3]. It lies in the formulation of control task as a specific energy-consumption optimization [4], which takes into account the appropriate mathematical model of the robot, reference trajectory and real measured robot motion. That task is formulated within a specific receding prediction horizon. Control actions are optimized in this horizon with respect to changeable robot model parameters along a reference trajectory of the required robot motion.

Relative to the real robot and its environment that cannot be precisely described by a mathematical model, the usual model-based control maintains usually the motion or positioning with relatively small but non-zero offset. It is crucial mainly for precise positional operations. Such problem of the offsets in predictive control design is continually investigated. There exist solutions based on experimentally tuned steady state working points or specific offset estimation via the estimation of true state augmented by a new state variable that represents unknown offset or disturbances [5]. Those solutions can be complicated or may not be efficient for multi-input multi-output systems like robots in general.

The paper aims at a specific solution based on incremental formulations of the equations of predictions suitably involved in the quadratic cost function [6]. The solution consists of generating increments of control actions in relation to predicted future system outputs or possibly expected control errors and increment accumulation to compensate steady-state control errors and suppress dynamical control errors relative to the specific required motion trajectory.

The proposed predictive control design copes with nonlinear character of a mathematical robot description by a specific decomposition that transforms the initial nonlinear model to a linear-like time-varying state-space model. The design (optimization) is repeated in each time instant considering appropriate updates of the model. In the paper, there are introduced two incremental algorithms differing in a number of discrete integrators. The number of integrators influences error compensation or suppression pace respecting shapes of required motion trajectories, e.g. one integrator is necessary for step signals and at least two integrators are required for ramp signals; the more integrators the more reactive response.

The paper is organized as follows. Section II describes synthesis of suitable mathematical model for control design including the advantageous decomposition leading to linear-like time-varying state-space model form. Section III deals with predictive control design in details for proposed two algorithms. Section IV demonstrates described theoretical results for the motion control of the six-axis multipurpose ABB robot IRB 140 considering variable load.

II. Mathematical Model of the Robot

The mathematical model intended for control design is represented by the dynamic model describing relations between control actions and robot dynamics or generally equations of motion of the robot. The model should be compact to be usable for control design. A suitable approach for the model composition is approach based on Lagrange equations. They can be written in the following form

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot{q}}\right)^T - \left(\frac{\partial E_k}{\partial q}\right)^T + \left(\frac{\partial E_p}{\partial q}\right)^T = \tau \qquad (1)$$

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 [1].

978-1-5386-2402-9/17/$31.00 ©2017 IEEE

The individual terms in (1) can be defined as follows

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot{q}}\right)^T = \dot{H}(q)\,\dot{q} \;+\; H(q)\,\ddot{q} \tag{2}$$

$$-\left(\frac{\partial E_k}{\partial q}\right)^T = S(q,\dot{q})\,\dot{q} \;-\; \frac{1}{2}\dot{H}(q)\,\dot{q} \tag{3}$$

$$\left(\frac{\partial E_p}{\partial q}\right)^T = g(q) \tag{4}$$

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

$$\ddot{q} = -H^{-1}\left(\frac{1}{2}\dot{H} + S\right)\dot{q} - H^{-1}g + H^{-1}\tau \tag{5}$$

The model (5) contains nonlinear terms. These terms can be encapsulated by the following rearrangements

$$f_0(q,\dot{q}) = -H^{-1}\left(\frac{1}{2}\dot{H} + S\right) \tag{6}$$

$$u = -H^{-1}g + H^{-1}\tau \tag{7}$$

where $f_0(q,\,\dot{q})$ in (6) represents a distribution of inertia effects and new system input u in (7) represents a distribution of gravitation effects and joint torques respectively. Considering these rearrangements, the continuous model (5) can be simply written as

$$\ddot{q} = f_0(q,\dot{q})\,\dot{q} + u \tag{8}$$

The model (8) stays nonlinear with respect to the term $f_0 = f_0(q,\dot{q})$. However, this term can be considered for individual time instants as a constant term multiplied by the isolate variable $\dot{q}$.

Such interpretation enables us to obtain usual linear-like, time-varying state-space model:

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & f_0 \end{bmatrix}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \tag{9}$$

$$y = \begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} \tag{10}$$

Equations (9) and (10) can be written in the following standard state-space form

$$\dot{x} = A(x)x + B\,u \tag{11}$$

$$y = C\,x \tag{12}$$

After discretization that affects both state and input matrices $A_k = A(x_k)$ and $B_k = B(x_k)$, the discrete prediction model used for control design is written as follows

$$\hat{x}_{k+1} = A_k\,x_k + B_k\,u_k \tag{13}$$

$$y_k = C\,x_k \tag{14}$$

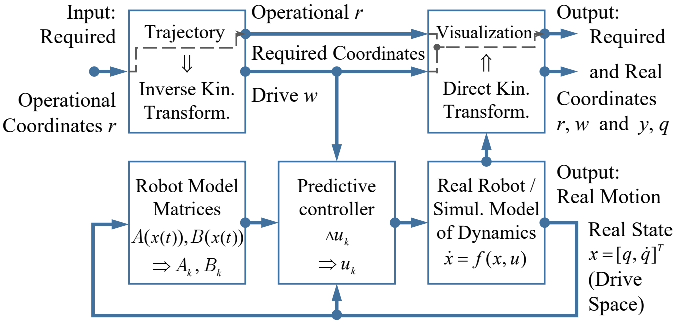


Fig. 1. Block diagram for the control of articulated robots.

Note, for completeness, that after computation of control actions in the current time instant, these control actions have to be recomputed to the real robot inputs (torques on individual robot joints) according to the following expression

$$\tau_k = H_k u_k + g_k \tag{15}$$

The expression (15) represents solution of (7) for vector of real values of torques τ including effects of gravitation.

III. Predictive Control

This section outlines essential elements and steps of the design. A used notation includes symbols:

$$\Delta x, \quad x, \quad \Delta y, \quad y, \quad \Delta u, \quad u, \quad e, \quad w\,.$$

These symbols represent increments and absolute values of the system state, system outputs, control actions, errors and references. Their definitions and utilization will be introduced in the subsequent sections that deal with a specific composition of the equations of predictions, used quadratic cost functions and their minimization. General block diagram of control circuit is shown in Fig. 1, [7].

A. Equations of Predictions

The equations of predictions express the functions of future system outputs in relation to unknown future control actions [8]. The composition of the equations is always initiated by a discretization of updated state-space model (11) considering current state x. For simplicity, let us consider the discrete state-space model (13)-(14) like this

$$\begin{aligned} \hat{x}_{k+1} &= A\,x_k + B\,u_k \\ y_k &= C\,x_k \end{aligned} \tag{16}$$

with $A = A_k = A_{k+i-1}$ and $B = B_k = B_{k+i-1}$ for $i = 1, \cdots, N$ within an optimization in the current time instant k along the current prediction horizon N. The next optimization in the instant $k+1$ begins by update of model (11) and its discretization again. In addition to the discrete model (16), a specific evolution model of the aggregated control error $\bar{e}_k$ is taken into account

$$e_k = w_k - y_k, \quad \bar{e}_k = \bar{e}_{k-1} + e_k \tag{17}$$

To achieve the integral property in the design, the model (16) can be written in increments as follows:

$$\begin{aligned} \hat{x}_{k+1} - x_k &= \quad A\,(x_k - x_{k-1}) + \quad B\,(u_k - u_{k-1}) \\ \hat{y}_{k+1} - y_k &= C\,A\,(x_k - x_{k-1}) + C\,B\,(u_k - u_{k-1}) \end{aligned} \tag{18}$$

or in a condensed incremental form as well

$$\begin{aligned} \vartriangle\hat{x}_{k+1} &= \quad A \vartriangle x_k + \quad B \vartriangle u_k \\ \vartriangle\hat{y}_{k+1} &= C\,A \vartriangle x_k + C\,B \vartriangle u_k \end{aligned} \tag{19}$$

The model (19) leads to an incremental feature of the equations of predictions in the proposed predictive control design. The sequence of their composition corresponds to the cost functions in Sections III-B and III-C.

The composition is based on recursive principle. It is involved in the initial equations (20), (21), (24) and (27) by the index $j = 1, \cdots, N$ that determines discrete time instants within the prediction horizon N.

The desired equations of predictions are defined as follows. • Equation (20) is stated for predictions of the system state increments $\vartriangle\hat{x}$

$$\vartriangle\hat{x}_{k+j} = A^j \vartriangle x_k + \sum_{i=1}^{j} A^{i-1} B \vartriangle u_{k+j-i} \tag{20}$$

• Equation (21) defines increments of system outputs $\vartriangle\hat{y}$

$$\vartriangle\hat{y}_{k+j} = CA^j \vartriangle x_k + \sum_{i=1}^{j} CA^{i-1} B \vartriangle u_{k+j-i} \tag{21}$$

It can be written in the condensed matrix form (22)

$$\vartriangle\hat{Y}_{k+1} = F_1 \vartriangle x_k + G_1 \vartriangle U_k \tag{22}$$

where individual elements $\vartriangle\hat{Y}$, $\vartriangle U$, F_1 and G_1 are:

$$\begin{aligned} \vartriangle\hat{Y}_{k+1} &= [\ \vartriangle\hat{y}^T_{k+1} \quad \cdots \quad \vartriangle\hat{y}^T_{k+N}\ ]^T \\ \vartriangle U_k &= [\ \vartriangle u^T_k \quad \cdots \quad \vartriangle u^T_{k+N-1}\ ]^T \\ F_1 &= \begin{bmatrix} CA \\ \vdots \\ CA^N \end{bmatrix}, \ G_1 = \begin{bmatrix} CB & \cdots & 0 \\ \vdots & \ddots & \vdots \\ CA^{N-1}B & \cdots & CB \end{bmatrix} \end{aligned} \tag{23}$$

• Equation (24) represents evolution of the full-value predictions of the system outputs $\hat{y}$

$$\hat{y}_{k+j} = y_k + \sum_{i=1}^{j} \vartriangle\hat{y}_{k+i} \tag{24}$$

The appropriate matrix notation of (24) is the following:

$$\hat{Y}_{k+1} = F_{\mathrm{I}}\ y_k + F_2 \vartriangle x_k + G_2 \vartriangle U_k \tag{25}$$

where subsequent individual elements $\hat{Y}_{k+1}$, F_{I}, F_2 and G_2 are defined as

$$\begin{gathered} \hat{Y}_{k+1} = [\ \hat{y}^T_{k+1} \quad \cdots \quad \hat{y}^T_{k+N}\ ]^T, \ \ F_{\mathrm{I}} = [\ I \quad \cdots \quad I\ ]^T \\ F_2 = \begin{bmatrix} CA \\ \vdots \\ \sum\limits_{i=1}^{N} CA^i \end{bmatrix}, \ G_2 = \begin{bmatrix} CB & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum\limits_{i=1}^{N} CA^{i-1}B & \cdots & CB \end{bmatrix} \end{gathered} \tag{26}$$

Finally, • equation (27) is for an aggregate control error $\hat{\bar{e}}$

$$\begin{aligned} \hat{\bar{e}}_{k+j-1} = \bar{e}_k &+ \sum_{i=1}^{j-1} \{w_{k+i}\} \ - (j-1)\, I\, y_k \\ &- \sum_{i=1}^{j-1} \{(j-i)\, C\, A^i\} \vartriangle x_k \\ &+ \sum_{l=1}^{j-1} \left\{ \sum_{i=l}^{j-1} \{(j-i)\, C\, A^{i-l} B\} \vartriangle u_{k+l-1} \right\} \end{aligned} \tag{27}$$

As well as previous equations, equation (27) can be expressed in the following matrix form:

$$\hat{E}_k = F_{\mathrm{I}}\ \bar{e}_k + W_s - F_{\mathrm{II}}\ y_k - F_3 \vartriangle x_k - G_3 \vartriangle U_k \tag{28}$$

where remaining elements $\hat{E}_k$, W_s, F_{II}, F_3 and G_3 are

$$\begin{aligned} \hat{E}_k &= [\ \bar{e}^T_k \quad \hat{\bar{e}}^T_{k+1} \quad \hat{\bar{e}}^T_{k+2} \quad \cdots \quad \hat{\bar{e}}^T_{k+N-1}\ ]^T \\ W_s &= \left[0^T \ \ w^T_{k+1} \ \ (w_{k+1} + w_{k+2})^T \ \ \cdots \left(\sum_{i=1}^{N-1} \{w_{k+i}\} \right)^T \ \right]^T \\ F_{\mathrm{II}} &= [\ 0 \quad I \quad 2I \quad \cdots \quad (N-1)\, I\ ]^T \\ F_3 &= \left[0^T \ \ (CA)^T \ \ (2CA + CA^2)^T \ \cdots \left(\sum_{i=1}^{N-1} (N-i) CA^i \right)^T \ \right]^T \\ G_3 &= \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 \\ CB & \cdots & 0 & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ \sum\limits_{i=1}^{N-1} (N-i) CA^{i-1} B & \cdots & 2CB + CAB & CB & 0 \end{bmatrix} \end{aligned} \tag{29}$$

B. Cost Function - First Algorithm

Let us consider the quadratic cost function in the form

$$\begin{aligned} J_k = \sum_{j=1}^{N} \Big\{ & \|Q_{yw}\,(\hat{y}_{k+j} - w_{k+j})\|_2^2 \\ & + \|Q_{\Delta y}\,\Delta\hat{y}_{k+j}\|_2^2 + \|Q_{\Delta u}\,\Delta u_{k+j-1}\|_2^2 \Big\} \\ = {} & (\hat{Y}_{k+1} - W_{k+1})^T Q_{YW}^T Q_{YW} (\hat{Y}_{k+1} - W_{k+1}) \\ & + \Delta\hat{Y}_{k+1}^T Q_{\Delta Y}^T Q_{\Delta Y} \Delta\hat{Y}_{k+1} + \Delta U_k^T Q_{\Delta U}^T Q_{\Delta U} \Delta U_k \end{aligned} \tag{30}$$

where reference values W_{k+1} and penalisation matrices $Q_{YW}^T Q_{YW}$, $Q_{\Delta Y}^T\, Q_{\Delta Y}$ and $Q_{\Delta U}^T\, Q_{\Delta U}$ are defined as

$$W_{k+1} = [\ w_{k+1}^T \quad w_{k+2}^T \quad \cdots \quad w_{k+N}^T\ ]^T$$

$$Q_\diamond^T Q_\diamond = \begin{bmatrix} Q_*^T Q_* & & 0 \\ & \ddots & \\ 0 & & Q_*^T Q_* \end{bmatrix} \left| \begin{array}{l} \text{subscripts } \diamond, *: \\ \diamond \in \{YW,\ \Delta Y,\ \Delta U\} \\ * \in \{yw,\ \Delta y,\ \Delta u\} \end{array} \right. \tag{31}$$

Then, the algorithm with integration of Δu and penalisation of output increments $Q_{\Delta y}^T Q_{\Delta y}$ will be composed as follows:

$$\begin{aligned} \Delta x_k &:= x_k - x_{k-1} \\ \Delta u_k &:= \textit{result of minimization} \text{ - see Section III-D} \\ u_k &:= u_{k-1} + \Delta u_k \end{aligned} \tag{32}$$

C. Cost Function - Second Algorithm

Let us consider another form of the quadratic cost function

$$\begin{aligned} J_k = \sum_{j=1}^{N} \Big\{ & \|Q_{yw}\,(\hat{y}_{k+j} - w_{k+j} - \hat{\bar{e}}_{k+j-1})\|_2^2 \\ & + \|Q_{\Delta y}\,\Delta\hat{y}_{k+j}\|_2^2 + \|Q_{\Delta u}\,\Delta u_{k+j-1}\|_2^2 \Big\} \\ = {} & (\hat{Y}_{k+1} - W_{k+1} - \hat{E}_k)^T Q_{YW}^T Q_{YW} (\hat{Y}_{k+1} - W_{k+1} - \hat{E}_k) \\ & + \Delta\hat{Y}_{k+1}^T Q_{\Delta Y}^T Q_{\Delta Y} \Delta\hat{Y}_{k+1} + \Delta U_k^T Q_{\Delta U_k}^T Q_{\Delta U_k} \Delta U_k \end{aligned} \tag{33}$$

Then, the algorithm with integration of Δu and $\bar{e}$ including penalisation of output increments $Q_{\Delta y}^T Q_{\Delta y}$ will be composed as follows:

$$\begin{aligned} e_k &:= w_k - y_k \\ \bar{e}_k &:= \bar{e}_{k-1} + e_k \\ \Delta x_k &:= x_k - x_{k-1} \\ \Delta u_k &:= \textit{result of minimization} \text{ - see Section III-D} \\ u_k &:= u_{k-1} + \Delta u_k \end{aligned} \tag{34}$$

D. Minimization Procedure

To minimize cost functions (30) and (33), let us consider the following expression

$$\min_{\Delta U_k} J_k \ = \ \min_{\Delta U_k} \mathbb{J}_k^T\, \mathbb{J}_k \ \rightarrow \ \min_{\Delta U_k} \mathbb{J}_k \tag{35}$$

indicating the minimisation of the square-root vector $\mathbb{J}_k$ instead of scalar J_k, which is more suitable for computation.

Thus, the square-root of the cost function (30) is

$$\min_{\Delta U_k} \mathbb{J}_k \ = \ \min_{\Delta U_k} \begin{bmatrix} Q_{YW} & 0 & 0 \\ 0 & Q_{\Delta Y} & 0 \\ 0 & 0 & Q_{\Delta U} \end{bmatrix} \begin{bmatrix} \hat{Y}_{k+1} - W_{k+1} \\ \Delta\hat{Y}_{k+1} \\ \Delta U_k \end{bmatrix} \tag{36}$$

which can be solved via a system of algebraic equations

$$\begin{bmatrix} Q_{YW}\, G_2 \\ Q_{\Delta Y}\, G_1 \\ Q_{\Delta U} \end{bmatrix} \Delta U_k = \begin{bmatrix} Q_{YW}(W_{k+1} - F_{\mathrm{I}} y_k - F_2 \Delta x_k) \\ Q_{\Delta Y}(-F_1 \Delta x_k) \\ 0 \end{bmatrix} \tag{37}$$

or similarly, the square-root of the cost function (33) is

$$\min_{\Delta U_k} \mathbb{J}_k = \min_{\Delta U_k} \begin{bmatrix} Q_{YW} & 0 & 0 \\ 0 & Q_{\Delta Y} & 0 \\ 0 & 0 & Q_{\Delta U} \end{bmatrix} \begin{bmatrix} \hat{Y}_{k+1} - W_{k+1} - \hat{E}_k \\ \Delta\hat{Y}_{k+1} \\ \Delta U_k \end{bmatrix} \tag{38}$$

which can be solved via a system of algebraic equations

$$\begin{aligned} & \begin{bmatrix} Q_{YW}\,(G_2 + G_3) \\ Q_{\Delta Y}\, G_1 \\ Q_{\Delta U} \end{bmatrix} \Delta U_k = \\ & \begin{bmatrix} Q_{YW}\{W_{k+1} + W_s + F_{\mathrm{I}} \bar{e}_k - (F_{\mathrm{I}} + F_{\mathrm{II}}) y_k - (F_2 + F_3) \Delta x_k\} \\ Q_{\Delta Y}(-F_1 \Delta x_k) \\ 0 \end{bmatrix} \end{aligned} \tag{39}$$

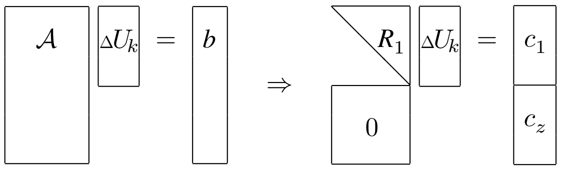
The systems (37) and (39), that are over-determined, can be written in one general form (40) according to [9] and solved for unknown ΔU_k

$$\mathcal{A}\,\Delta U_k \ = \ b \tag{40}$$

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

$$R_1\,\Delta U_k \ = \ c_1 \tag{41}$$

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



(42)

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 Δu_k are selected from computed vector ΔU_k. Then, the vector Δu_k is used in algorithm (32) or in (34) according to number of required integrators.

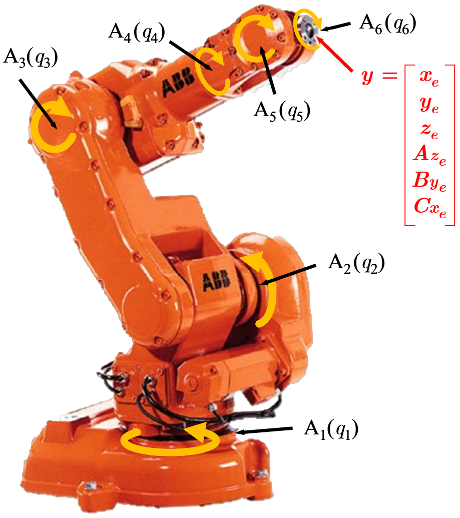

$A_3(q_3)$
$A_4(q_4)$
$A_6(q_6)$
$A_5(q_5)$
$y = \begin{bmatrix} x_e \\ y_e \\ z_e \\ Az_e \\ By_e \\ Cx_e \end{bmatrix}$
$A_2(q_2)$
$A_1(q_1)$

Fig. 2. Six-axis multipurpose ABB robot IRB 140.

IV. Simulation Examples

The examples demonstrate the behavior of the articulated ABB robot IRB 140 (Fig. 2) at the motion along selected testing trajectory (Fig. 3). 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 $y = q_{1:6}\,(rad)$ corresponding to the appropriate Cartesian coordinates $[\{x_e, y_e, z_e\,(m)\}, \{Az_e, By_e, Cx_e\,(rad)\,\}]^T$ and twelve state variables: $x = [q_{1:6}\,(rad), \dot{q}_{1:6}\,(rad\cdot s^{-1})]^T$. The testing trajectory (Fig. 3) consists of arc and abscissa segments. The segment specification is listed in the Table I.

TABLE I. 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			

The trajectory was time parameterized with 5^{th} order polynomial of acceleration (Fig. 4), [10]. Note that end-effector orientation was assumed to be parallel with axis x_e. Thus, the orientation angles are $Az_e = By_e = Cx_e = 0\,rad$.

Proposed algorithms (32) and (34) were tested with identical control parameters as follows:

- sampling period: $Ts = 0.01s$,
- horizon of prediction: $N = 10$,
- output penalisation: $Q_{yw} = I_{(6\times 6)}$,
- penalization of output increments: $Q_{\Delta y} = 5\cdot 10^{-1}\cdot I_{(6\times 6)}$,
- penalization of input increments: $Q_{\Delta u} = 1\cdot 10^{-5}\cdot I_{(6\times 6)}$, where I is the identity matrix.

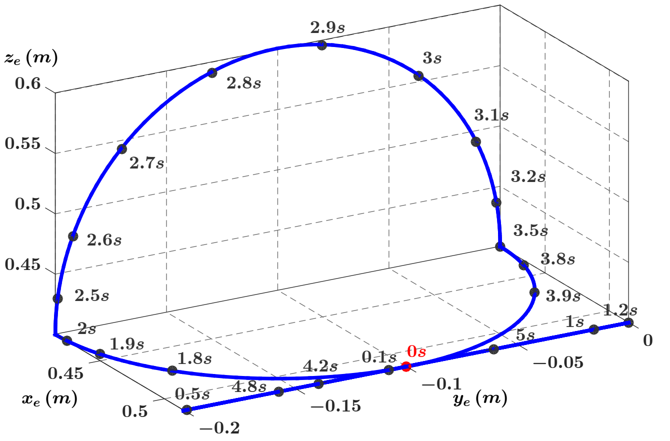


Fig. 3. Testing trajectory with specific time marks.

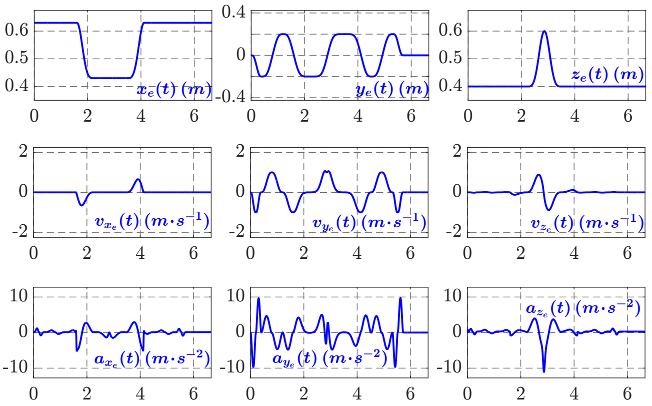


Fig. 4. Cartesian coordinates and derivatives (time is in (s)).

The algorithms were compared with usual predictive control ([3]) expressed as

$$u_k = M(G_1^T Q_{YW}^T Q_{YW} G_1 + Q_U^T Q_U)^{-1} \times G_1^T Q_{YW}^T Q_{YW}(W_{k+1} - F_1 x_k) \tag{43}$$

where matrix M selecting current control actions is defined as $M = [I_{(6\times 6)}, 0_{(6\times 6(N-1))}]$ for given robot with six degrees of freedom. Output and input penalisation matrices were selected as $Q_{yw} = I_{(6\times 6)}$ and $Q_u = 1{\cdot}10^{-4}{\cdot}I_{(6\times 6)}$, respectively.

All algorithms of predictive control were tested for zero load and for load $0.25\,kg$ considered in center of axis A$_6$, i.e. center of the flange of the robot end-effector. Note that the maximum load capacity of given robot (Fig. 2) is $6\,kg$. Fig. 5 shows time histories of generalized coordinates $q_{1:6}$ and appropriate torques (control actions) $\tau_{1:6}$. Note that torques represent inputs of drive control. The time histories of control actions for both cases of zero and nonzero load differ only negligibly on the order of units. However, big differences are in control errors that are shown in separate figures.

Fig. 6 shows situation with the precise mathematical model, i.e. zero load. Fig. 7 shows situation for nonzero load at axis A$_6$ that is not involved in the mathematical model. It represents a model mismatch with the robot. In the figures, the fastest response of the second predictive algorithm is evident: the smallest errors, the fastest trend towards zero errors.

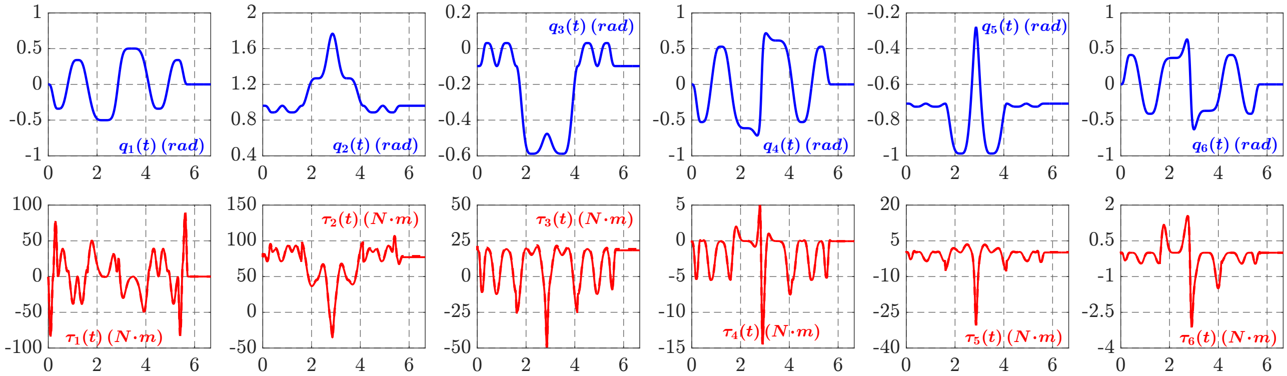


Fig. 5. Time histories (s) of generalized coordinates q_i and torques τ_i (control actions) in individual joints.

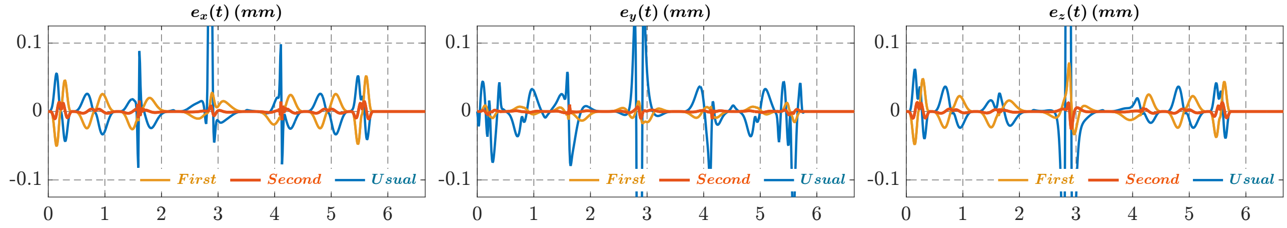


Fig. 6. Control errors of the First and Second algorithms compared with Usual predictive control (appropriate model).

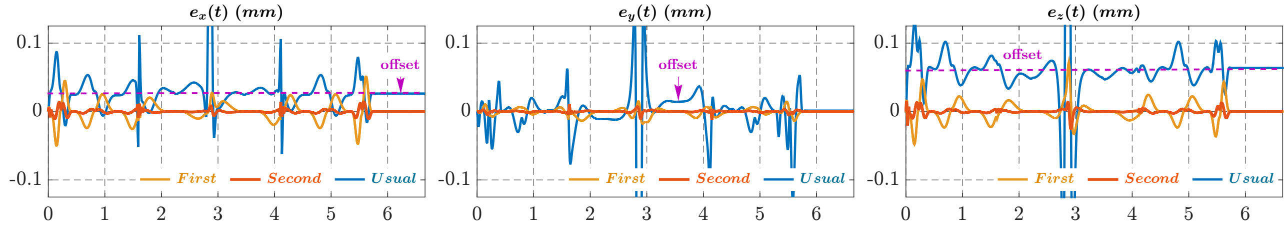


Fig. 7. Control errors of the First and Second algorithms compared with Usual predictive control (model mismatch).

Thus, the faster response of the second algorithm suitable for ramp (first-order) reference signals is noticeable in comparison with the first algorithm determined only for rectangular (zero-order) reference signals. It is caused by the number of involved integrators of individual algorithms. Thus, e.g. usual algorithm (43) has no integrative feature. It represents pure positional manner only. It is obvious especially in case of nonzero load, value of which is not involved in the model used for control design. The both incremental algorithms tend to zero in steady reference values. Other errors in the dynamical motion are caused by the difference of reference signals from either zero-order or first-order trends. Usual positional algorithm leads to steady nonzero offset as indicated in Fig. 7. This is reflected in the components (x_e, z_e) associated with the joints (q_2, q_3) that provide a movement in the vertical direction.

V. Conclusion

The paper introduces specific modification of initial nonlinear model leading to a linear-like time-varying state-space model (9) and design of two specific incremental predictive algorithms (32) and (34) differing in the number of involved integrators. The proposed algorithms lead to the suppression of undesirable offsets and they are intended for motion control at precise manipulation operations and robot positioning.

References

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics - Modelling, Planning and Control*. Springer, 2009.

[2] K. Belda, J. Böhm, and P. Píša, “Concepts of model-based control and trajectory planning for parallel robots,” in *Proc. of the 13th IASTED Int. Conf. on Robotics and Applications*, Germany, 2007, pp. 15–20.

[3] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, 2009.

[4] A. Othman, K. Belda, and P. Burget, “Physical modelling of energy consumption of industrial articulated robots,” in *Proc. of the 15th Int. Conf. on Control, Automation and Systems*, Korea, 2015, pp. 784–789.

[5] U. Maeder, F. Borrelli, and M. Morari, “Linear offset-free model predictive control,” *Automatica*, vol. 45, pp. 2214–2222, 2009.

[6] K. Belda and D. Vošmik, “Explicit generalized predictive control of speed and position of PMSM drives,” *IEEE Trns. Ind. Electron.*, vol. 63, no. 2, pp. 3889–3896, 2016.

[7] K. Belda and V. Záda, “Concepts of modeling and control of industrial articulated robots,” in *Proc. of the 3rd Int. Chemnitz Manufacturing Colloquium 2014*. Germany: Fraunhofer IWU, 2014, pp. 571–582.

[8] A. Ordis and D. Clarke, “A state-space description for GPC controllers,” *J. Systems SCI.*, vol. 24, no. 9, pp. 1727–1744, 1993.

[9] C. Lawson and R. Hanson, *Solving least squares problems*. Siam, 1995.

[10] K. Belda and P. Novotný, “Path simulator for machine tools and robots,” in *Proc. of MMAR IEEE Int. Conf.*, Poland, 2012, pp. 373–378.