Nonlinear Dyn (2023) 111:8383–8395
https://doi.org/10.1007/s11071-023-08295-z

ORIGINAL PAPER

Predictive tracking of an object by a pan–tilt camera of a robot

Robert Nebeluk · Krzysztof Zarzycki · Dawid Seredyński · Patryk Chaber · Maksym Figat · Paweł D. Domański · Cezary Zieliński

Received: 5 May 2022 / Accepted: 26 January 2023 / Published online: 16 February 2023

Abstract Moving-object tracking using a pan–tilt camera setup is quite a well-known task in robotics. However, the presented research addresses specific properties of the tracked object and introduces novel features to the pan–tilt camera control strategy. Pan–tilt camera control does not operate in an isolated environment. It is a part of the visual servoing system with specific goals. The system has to fulfill certain purposes, which affect its configuration and functionality. The pan–tilt system aims at keeping the visually tracked object within the middle of the image. At the same time, the overall visual servoing efficiently recognizes and tracks the object enabling its grasping by the robot arm. It uses a predictive strategy utilizing specific second-order linear models for pan and tilt joints. Model predictive control (MPC) introduces into the system the ability to predict camera operation over the specific horizon according to the predefined tracking goals. As the system anticipates future positions over the horizon of operation, the setpoint prediction of the future tracked system positions is required. Visual object recognition and tracking system use particular strategies for preparing online tracked object extrapolation over MPC horizon. Therefore, the pan–tilt camera system is intrinsically coupled to camera-based recognition and tracking. Predictive pan–tilt positioning keeps the tracked system in the middle of the image, while the visual system extrapolation improves the tracking performance. The proposed approach is thoroughly tested in the dedicated Gazebo-based robot simulator. Finally, the system is implemented and validated on the Velma robot. The results and their comparison with other control strategies confirm the initial assumptions, allowing further visual servoing system development.

Keywords Visual servoing · Pan–tilt camera · Predictive tracking · Model predictive control · Setpoint extrapolation

1 Introduction

Pan–tilt camera systems are frequently utilized in 3D visual tracking. For instance, they are used by

R. Nebeluk (✉) · K. Zarzycki · D. Seredyński · P. Chaber · M. Figat · P. D. Domański · C. Zieliński
Warsaw University of Technology, Institute of Control and Computation Engineering, ul. Nowowiejska 15/19, 00–665 Warsaw, Poland
e-mail: Robert.Nebeluk@pw.edu.pl

K. Zarzycki
e-mail: Krzysztof.Zarzycki@pw.edu.pl

D. Seredyński
e-mail: Dawid.Seredynski@pw.edu.pl

P. Chaber
e-mail: Patryk.Chaber@pw.edu.pl

M. Figat
e-mail: Maksym.Figat@pw.edu.pl

P. D. Domański
e-mail: pawel.domanski@pw.edu.pl

C. Zieliński
e-mail: Cezary.Zielinski@pw.edu.pl

autonomous vehicles or robots that have a vision head [2,5,20,27]. Such vision systems have to solve two problems: real-time processing of visual data and inducing the correct tracking motion of the camera. Solving the problem of efficient visual information processing is a significant challenge and constitutes the majority of the research. Data acquired by an RGB-D camera need to be processed in real time [3], and thus the image must be segmented, target features must be identified and extracted, useful information must be structured into control data, and finally, these data must be delivered to the system component that requires them. Those calculations and data handling actions are time-consuming. The majority of the research focuses on their speed up using time-efficient algorithms, usually through parallelization of calculations [24] or faster hardware. The task of servopositioning of the camera supporting neck is considered separately. Control issues and very high control accuracy do not lie in the mainstream of research. The main goal is to keep the object within the image.

Practical experience shows that tracking and recognition are coupled. Image recognition performance depends on proper camera tracking and vice versa. Once we can keep the tracked object within the same sector of the image frame, we may limit the search and analysis space of the whole frame, what speeds up calculations. On the contrary, if we fail to achieve image processing within the system sampling period, the information about the target will be lost for one or more samples, what may cause the servomechanism tracking degradation or even induce losing the tracked object out of sight. Therefore, camera image processing and servo tracking tasks are closely coupled, and the correlation works in both directions. Better camera tracking operation speeds up visual data processing and prevents losing the object from sight. Fast visual data evaluation prevents sampling losses and tracking problems. As the tracked object can be located quite far away from the head and thus the cameras, even a small improvement in tracking accuracy may help.

One may find different control strategies applied in pan–tilt servoing. Classical PID-based feedback control is the primary choice. When the servomotor is overrated and fast, just simple P (proportional) version might be satisfactory, especially when the steady-state error is allowed [44]. However, most frequently full PID version is used [22,41], sometimes modified by factors similar to those used in process control, such as derivative filtering [44] or lead-lag compensation [40]. Feedback control is improved by feedforward decoupling [27] in situations when disturbance measurement is available. A similar task can be solved alternately by active disturbance rejection control [5]. If accurate control is required, feedback control is replaced by more advanced control strategies [4]. Once system nonlinearities affect control performance, the multi-regional PID strategies can be applied, such as fuzzy PID [28].

Interesting, although quite different, idea uses synergetic control [18] and the analytical design of aggregated regulators (ADAR) methodology [38]. The use of linear quadratic regulation (LQR) [14,33], improved augmented LQR (IALQR) [29] or linear quadratic Gaussian (LQG) control [31] forms the first steps towards the use of the model predictive control (MPC) techniques. Although MPC strategy is quite frequently, especially recently, used in robotic applications [10,11,16,17,21,26], its application to a pan–tilt tracking system is rare [6,34,43].

Visual trackers can use various algorithms to evaluate and extrapolate tracked object trajectories. Kalman filters are the most obvious choices [30,39]. However, the subject literature is much richer [36]. Interestingly, deep coupling between both tasks is not popular, although quite obvious. MPC strategy needs to have a setpoint trajectory over the prediction horizon. And yet the tracking algorithm allows us to evaluate it. The relationship works in the opposite direction as well. Accurate tracking and keeping the tracked target in the same position of the camera image enables improving image recognition algorithm minimizing tracking evaluation time, and minimizing the possible effect of losing a sample due to the prolonged calculations. It is important to stress that if the tracked object is at all times in the center of the image frame, the whole procedure gives the best results.

Following the above observations, the proposed research fills the observed gap. It focuses on efficient, predictive pan–tilt camera servomechanism (neck-MPC). The aim of the neck-MPC is high target-object tracking accuracy within a camera image frame. The application of predictive control algorithms, such as DMC and GPC in the pan–tilt tracking system enables practical realization of coupling between the servomechanism and image recognition. The DMC and GPC control rules represent a relatively simple predictive approach. However, their superiority lies not only in an embedded model or optimization, but in the repetitive operation over a horizon. It is especially important how the setpoint is formulated. The PID-type controller uses only the current setpoint and current control error in its rule, even if the future setpoint trajectory is known. Predictive control enables the incorporation of the knowledge about the future setpoint trajectory in the evaluation of the current manipulated variable. We acknowledge the fact that the more sophisticated is the control algorithm, the better is its performance. However, its robustness, sustainability and the number of engineers that can maintain it generally diminishes. Therefore, a simple approach, such as DMC or GPC, is taken into account in the present research. High tracking performance improves visual data processing, generating a beneficial synergy effect.

The paper is organized as follows. Section 2 presents the theoretical aspects of this research, i.e., the system kinematics (Sect. 2.1), reference PID control (Sect. 2.2), and the servomotor predictive control strategy (Sect. 2.3). The developed neck-MPC is validated with Matlab, the Gazebo simulator and the real Velma robot, what is described in Sect. 3. Section 4 summarizes the results of the work, draws conclusions, and presents the identified open issues for further research.

2 Pan–tilt control system

This section describes two main elements of the proposed contribution, i.e., the controlled pan–tilt system and the applied control strategy.

2.1 Neck kinematics

The relevant part of the kinematic structure of the Velma robot [35] equipped with a pan-tilt (neck) mechanism is presented in Fig. 1. Table 1 contains the respective D-H parameters [7].

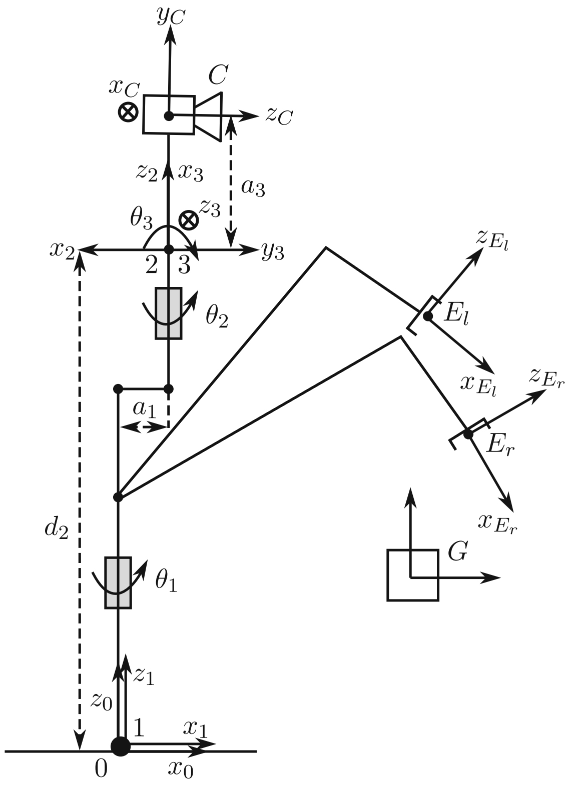

Fig. 1 Coordinate frames assigned according to Denavit–Hartenberg convention to the links of the Velma robot between the base coordinate frame 0 and the camera frame C; the left and right arms are marked symbolically and only their end-effector frames E_l and E_r are presented

Table 1 Denavit–Hartenberg parameters of the kinematic structure between the base and the neck of the Velma robot

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	a_1	0	d_2	θ_2
3	0	$\pi/2$	0	θ_3
4	a_3	$-\pi/2$	0	$-\pi/2$

The following frames are relevant to the considerations: 0 – base coordinate frame, 1 – robot torso frame, 2 – pan–tilt first link frame, 3 – pan-tilt second link frame, C – camera frame, G – goal frame (the tracked object). Homogeneous matrices represent those frames, e.g., ${}^{0}_{C}\mathcal{T}$ – camera with respect to the base. A homogeneous matrix contains the orientation submatrix, e.g., ${}^{0}_{C}\mathcal{R}$, and the position vector, e.g., ${}^{0}_{C}\mathcal{P}$.

The task of the visual servo is to track the object of interest G in such a way that the location of its image will be approximately at the center of the camera imaging plane. For that to happen, the versor of the camera optical axis ${}^{C}_{C}z$ and the vector from the camera to the center of the object ${}^{C}_{G}\mathcal{P}$ should coincide. In such a situation, the vector cross-product of those two vectors should be 0. If this is not the case, the vector cross-product defines the axis of rotation about which the camera should be rotated so that the image of the object will become centered in the camera image plane. If ${}^{C}_{G}\mathcal{P}$ is normalized the length of the resulting vector is equal to $\left\| {}^{C}_{C}z \times \frac{{}^{C}_{G}\boldsymbol{\mathcal{P}}}{\|{}^{C}_{G}\boldsymbol{\mathcal{P}}\|} \right\| = \sin\phi$, where ϕ is the required angle of rotation. If this value is nonzero, the versor of the axis of rotation κ has to be derived

$$\boldsymbol{\kappa} = \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_z \end{bmatrix} = \begin{bmatrix} -\frac{{}^{C}_{G}\boldsymbol{\mathcal{P}}_y}{\sqrt{{}^{C}_{G}\boldsymbol{\mathcal{P}}_x^2 + {}^{C}_{G}\boldsymbol{\mathcal{P}}_y^2}} \\ \frac{{}^{C}_{G}\boldsymbol{\mathcal{P}}_x}{\sqrt{{}^{C}_{G}\boldsymbol{\mathcal{P}}_x^2 + {}^{C}_{G}\boldsymbol{\mathcal{P}}_y^2}} \\ 0 \end{bmatrix}. \tag{1}$$

If $\sin\phi = 0$, no camera motion is required, thus versor κ is irrelevant. Hence, if the motion is required, the orientation of the camera has to change by rotating about κ by ϕ, i.e., the desired rotation is [7]:

$$\boldsymbol{\mathcal{R}}_{\kappa,\phi} = \left[\begin{array}{c|c|c} \kappa_x\kappa_x(1-\mathrm{c}\phi)+\mathrm{c}\phi & \kappa_y\kappa_x(1-\mathrm{c}\phi)-\kappa_z\mathrm{s}\phi & \kappa_z\kappa_x(1-\mathrm{c}\phi)+\kappa_y\mathrm{s}\phi \\ \kappa_x\kappa_y(1-\mathrm{c}\phi)+\kappa_z\mathrm{s}\phi & \kappa_y\kappa_y(1-\mathrm{c}\phi)+\mathrm{c}\phi & \kappa_z\kappa_y(1-\mathrm{c}\phi)-\kappa_x\mathrm{s}\phi \\ \kappa_x\kappa_z(1-\mathrm{c}\phi)-\kappa_y\mathrm{s}\phi & \kappa_y\kappa_z(1-\mathrm{c}\phi)+\kappa_x\mathrm{s}\phi & \kappa_z\kappa_z(1-\mathrm{c}\phi)+\mathrm{c}\phi \end{array}\right], \tag{2}$$

where here and further s stands for sin and c for cos. The desired orientation of the camera with respect to the torso is: ${}^{1}_{C}\boldsymbol{\mathcal{R}}(\varphi_c, \psi_c)\boldsymbol{\mathcal{R}}_{\kappa,\phi}$. This has 3-DOF, while the neck has only 2-DOF; thus an artificial angle γ is introduced:

$${}^{1}_{C}\boldsymbol{\mathcal{R}}(\theta_2, \theta_3)\boldsymbol{\mathcal{R}}_{\kappa,\phi} = {}^{1}_{C}\boldsymbol{\mathcal{R}}(\theta_2^{\mathrm{SP}}, \theta_3^{\mathrm{SP}})\boldsymbol{\mathcal{R}}_{Cz,\gamma} \tag{3}$$

where θ_2 and θ_3 are the current value of the neck joint angles, while superscript SP represents the desired value (set-point) of the angles.

The value of ${}^{1}_{C}\boldsymbol{\mathcal{R}}(\theta_2, \theta_3)$ is produced by solving the direct kinematics problem:

$${}^{1}_{C}\boldsymbol{\mathcal{R}}(\theta_2, \theta_3) = \begin{bmatrix} \mathrm{s}\theta_2 & \mathrm{c}\theta_2\mathrm{c}\theta_3 & -\mathrm{c}\theta_2\mathrm{s}\theta_3 \\ -\mathrm{c}\theta_2 & \mathrm{s}\theta_2\mathrm{c}\theta_3 & -\mathrm{s}\theta_2\mathrm{s}\theta_3 \\ 0 & \mathrm{s}\theta_3 & \mathrm{c}\theta_3 \end{bmatrix} \tag{4}$$

using the current encoder readings. The value of $\boldsymbol{\mathcal{R}}_{\kappa,\phi}$ is computed based on (2). Hence

$${}^{1}_{C}\boldsymbol{\mathcal{R}}(\theta_2, \theta_3)\boldsymbol{\mathcal{R}}_{\kappa,\phi} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{5}$$

is a known matrix. Thus the desired orientation is:

$$\begin{aligned} {}^{1}_{C}\boldsymbol{\mathcal{R}}(\theta_2^{\mathrm{SP}}, \theta_3^{\mathrm{SP}}) &= {}^{1}_{C}\boldsymbol{\mathcal{R}}(\theta_2, \theta_3)\boldsymbol{\mathcal{R}}_{\kappa,\phi}\boldsymbol{\mathcal{R}}^{-1}_{Cz,\gamma} \\ &= \begin{bmatrix} r_{11}\mathrm{c}\gamma - r_{12}\mathrm{s}\gamma & r_{11}\mathrm{s}\gamma + r_{12}\mathrm{c}\gamma & r_{13} \\ r_{21}\mathrm{c}\gamma - r_{22}\mathrm{s}\gamma & r_{21}\mathrm{s}\gamma + r_{22}\mathrm{c}\gamma & r_{23} \\ r_{31}\mathrm{c}\gamma - r_{32}\mathrm{s}\gamma & r_{31}\mathrm{s}\gamma + r_{32}\mathrm{c}\gamma & r_{33} \end{bmatrix}. \end{aligned} \tag{6}$$

Equating (6) and (4) and using their last columns, as they do not depend on γ, the inverse kinematics problem for the neck is solved:

$$\begin{aligned} \theta_3^{\mathrm{SP}} &= \theta_{\mathrm{tilt}}^{\mathrm{SP}} = \mathrm{Atan2}(\pm\sqrt{r_{13}^2 + r_{23}^2}, r_{33}), \\ \theta_2^{\mathrm{SP}} &= \theta_{\mathrm{pan}}^{\mathrm{SP}} = \mathrm{Atan2}(r_{23}, r_{13}) \text{ or} \\ \theta_2^{\mathrm{SP}} &= \theta_{\mathrm{pan}}^{\mathrm{SP}} = \mathrm{Atan2}(-r_{23}, -r_{13}). \end{aligned} \tag{7}$$

It is important to note that equations (7) assume that $sin(\theta_3) \neq 0$. Fortunately, the angle θ_3 never equals to $0 + k\pi$, $k = 0, 1, 2, \ldots$, due to the kinematics limitations. In other words – it is physically impossible for the camera to be pointed vertically upwards.

2.2 PID – benchmark control strategy

A PID algorithm has also been formulated to check whether an MPC algorithm such as DMC would give better results than this standard controller. In continuous time this algorithm generates new values of the control signal according to the following formula:

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(0) + K_{\mathrm{G}}\left(\boldsymbol{e}(t) + \frac{1}{T_{\mathrm{I}}}\int_0^t \boldsymbol{e}(\tau)\mathrm{d}\tau + T_{\mathrm{D}}\frac{\mathrm{d}\boldsymbol{e}(t)}{\mathrm{d}t}\right) \tag{8}$$

where $\hat{\boldsymbol{\theta}}(0)$ is the initial value of control for the integral part, $\boldsymbol{e}$ is the control error, K_{G} is the gain, T_{I} is the time constant of the the integral part and T_{D} is the time constant of the derivative part. Vectors in the formulation (8) have the following form: $\hat{\boldsymbol{\theta}}(t) = [\hat{\theta}_{\mathrm{pan}}(t)\ \hat{\theta}_{\mathrm{tilt}}(t)]^{\mathrm{T}}$, $\hat{\boldsymbol{\theta}}(0) = [\hat{\theta}_{\mathrm{pan}}(0)\ \hat{\theta}_{\mathrm{tilt}}(0)]^{\mathrm{T}}$, $\boldsymbol{e} = [\boldsymbol{e}_{\mathrm{pan}}\ \boldsymbol{e}_{\mathrm{tilt}}]^{\mathrm{T}}$. The control error is calculated as follows:

$$\boldsymbol{e}(t) = \boldsymbol{\theta}^{\mathrm{SP}}(t) - \boldsymbol{\theta}(t) \tag{9}$$

Formula (8) can be rewritten into discrete time (k) form, with the sampling period T_{s}:

$$\hat{\boldsymbol{\theta}}(k) = a_2\boldsymbol{e}(k-2) + a_1\boldsymbol{e}(k-1) + a_0\boldsymbol{e}(k) + \hat{\boldsymbol{\theta}}(k-1) \tag{10}$$

where

$$a_0 = K_{\mathrm{G}}\left(1 + \frac{T_{\mathrm{s}}}{2T_{\mathrm{I}}} + \frac{T_{\mathrm{D}}}{T_{\mathrm{s}}}\right) \tag{11}$$

$$a_1 = K_{\mathrm{G}}\left(\frac{T_{\mathrm{s}}}{2T_{\mathrm{I}}} - 2\frac{T_{\mathrm{D}}}{T_{\mathrm{s}}} - 1\right) \tag{12}$$

$$a_2 = K_{\mathrm{G}} \frac{T_{\mathrm{D}}}{T_{\mathrm{s}}} \tag{13}$$

Vectors in the discrete-time notation presented in (10) have analogous form to the respective vectors in continuous time notation.

2.3 Predictive pan–tilt servoing

Model predictive control (MPC) constitutes the focal element of the system. The considered predictive pan–tilt camera system control is realized in a hierarchical control structure with the MPC delivering joint angle setpoints for the low level servomotors controlling the joint motors. Figure 2 presents the pan–tilt control system neck-MPC($\boldsymbol{\theta}$) tracking the target in the pan $\theta_{\mathrm{pan}}\ (k)$ and tilt $\theta_{\mathrm{tilt}}\ (k)$ joint angle spaces.

The neck-MPC controller operates in a direct mode with a high sampling frequency of 500 Hz. It directly controls the pan and tilt joint angles. Its setpoint is generated by the vision system, which acquires the image of the target. Visual object pose estimation operates with a longer sampling rate of 20 Hz. This setpoint value over the 50-ms interval is interpolated into the 2 ms sampling of the pan–tilt servomechanism. Therefore, the dynamic pan–tilt controller has enough time to realize each setpoint signal. Moreover, some time is left for setpoint signal shaping and smoothing during interpolation.

MPC controllers work with the sampling frequency of 500 Hz (denoted by time instant index k). They obtain their setpoint with 20 Hz (denoted by time instant index τ), which is further interpolated from 20 Hz to 500 Hz. The employed MPC algorithm and the proposed modifications, i.e., the internal model, cost function and constraints, are described below. The schematic diagram of the implementation procedures of the proposed approach is sketched in Fig. 3.

2.3.1 *Model predictive control—general formulation*

The process input, i.e., manipulated variable (MV), is denoted by u and the output, controlled variable (CV), is denoted by y. The vector of decision variables calculated at each discrete sampling moment ($k = 0, 1, 2, \ldots$) by the MPC [37] is

$$\Delta \boldsymbol{u}(k) = [\Delta u(k|k)\ \Delta u(k+1|k) \ldots \\ \Delta u(k + N_{\mathrm{u}} - 1|k)]^{\mathrm{T}}, \tag{14}$$

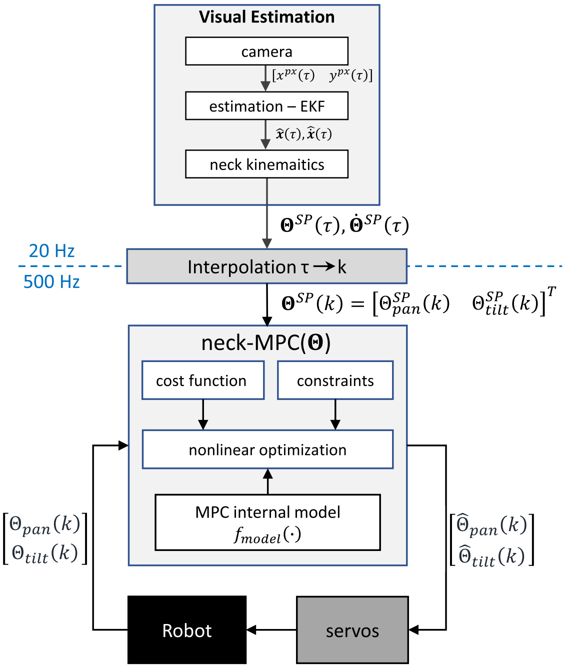
Visual Estimation
camera
$[x^{px}(\tau) \quad y^{px}(\tau)]$
estimation – EKF
$\hat{\boldsymbol{x}}(\tau), \hat{\dot{\boldsymbol{x}}}(\tau)$
neck kinemaitics
$\boldsymbol{\Theta}^{SP}(\tau), \dot{\boldsymbol{\Theta}}^{SP}(\tau)$
20 Hz
500 Hz
Interpolation τ→k
$\boldsymbol{\Theta}^{SP}(k) = [\Theta^{SP}_{pan}(k) \quad \Theta^{SP}_{tilt}(k)]^T$
neck-MPC($\boldsymbol{\Theta}$)
cost function
constraints
nonlinear optimization
MPC internal model
$f_{model}(\cdot)$
$\begin{bmatrix}\Theta_{pan}(k)\\ \Theta_{tilt}(k)\end{bmatrix}$
$\begin{bmatrix}\hat{\Theta}_{pan}(k)\\ \hat{\Theta}_{tilt}(k)\end{bmatrix}$
Robot
servos

Fig. 2 Pan–tilt neck-MPC($\boldsymbol{\theta}$) control in the joint space

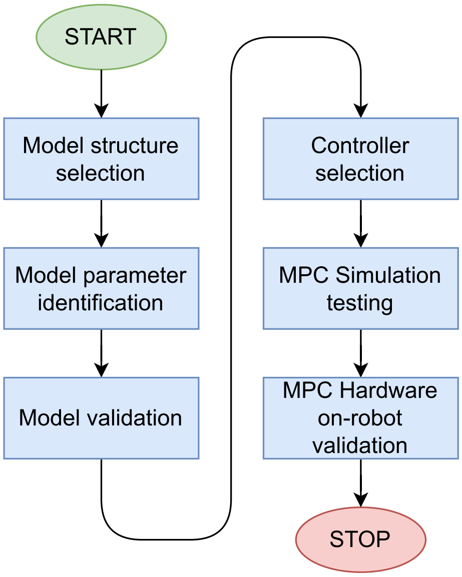
START
Model structure selection
Model parameter identification
Model validation
Controller selection
MPC Simulation testing
MPC Hardware on-robot validation
STOP

Fig. 3 Diagram of the procedures necessary to implement the MPC approach

where N_{u} is the control horizon, i.e., the number of calculated future control increments defined as backward differences, i.e., $\Delta u(k|k) = u(k|k) - u(k-1)$ and $\Delta u(k+p|k) = u(k+p|k) - u(k+p-1|k)$ for $p = 1, \ldots, N_{\mathrm{u}} - 1$. For $p \geq N_{\mathrm{u}}$ it is assumed that the manipulated variable is constant, i.e., $u(k+p|k) = u(k+N_{\mathrm{u}}-1|k)$. The decision variables of MPC (14) are calculated by solving the optimization problem:

$$\min_{\Delta\boldsymbol{u}(k)} \left\{ \sum_{p=1}^{N} \psi_p (y^{\mathrm{SP}}(k+p|k) - \bar{y}(k+p|k))^2 + \right.$$
$$\left. + \sum_{p=0}^{N_{\mathrm{u}}-1} \lambda_p (\Delta u(k+p|k))^2 \right\} \tag{15}$$

subject to

$$\begin{aligned}
&u^{\min} \leq u(k+p|k) \leq u^{\max},\ \ p = 0, \ldots, N_{\mathrm{u}}-1,\\
&\quad -\Delta u^{\max} \leq \Delta u(k+p|k) \leq \Delta u^{\max},\\
&\quad p = 0, \ldots, N_{\mathrm{u}}-1,\\
&\quad y^{\min} \leq \bar{y}(k+p|k) \leq y^{\max},\ \ p = 1, \ldots, N,\\
&\quad \bar{y}(k+p|k) = f_{\mathrm{model}}(\cdot),\ \ p = 1, \ldots, N.
\end{aligned}$$

The role of the first part of the MPC cost function is to minimize the predicted control errors over the prediction horizon N. The setpoint (CV demand) and predicted values of the CV for the future sampling instant $k+p$ known or calculated at the current moment k are denoted by $y^{\mathrm{SP}}(k+p|k)$ and $\bar{y}(k+p|k)$. The predictions $\bar{y}(k+p|k)$ are calculated on-line using the model of the process, described by the general function $f_{\mathrm{model}}(\cdot)$.

The role of the second part of the cost function is to eliminate excessive MV changes. In general, the constraints may be imposed on

- future excessive values of the manipulated variable (over the control horizon), limited by the minimal and maximal allowed values $u^{\min}$ and $u^{\max}$,
- future MV changes with its maximal value denoted as $\Delta u^{\max}$ and
- predicted values of the controlled variable (over the prediction horizon), limited by the minimal and maximal allowed values $y^{\min}$ and $y^{\max}$.

Although the whole sequence of decision variables (14) is calculated at each sampling, only its first component is applied to the process. Measurement of the process output is updated during the next sampling period $k+1$, and the procedure is repeated. Appropriate stability proofs exist and can be found in the literature [1,23]. The considered MPC optimization task (15) is common in industrial practice [9].

2.3.2 Applied MPC formulation

The optimization problem (15) can be rewritten for the neck into the following form:

$$\min_{\Delta\hat{\boldsymbol{\Theta}}(k)} \left\{ J(k) = \left\| \boldsymbol{\Theta}^{\mathrm{SP}}(k) - \bar{\boldsymbol{\Theta}}(k) \right\|^2_{\boldsymbol{\Psi}} + \left\| \Delta\hat{\boldsymbol{\Theta}}(k) \right\|^2_{\boldsymbol{\Lambda}} \right\}$$

subject to (16)

$$\begin{aligned}
&-\hat{\boldsymbol{\Theta}}^{\max} \leq \hat{\boldsymbol{\Theta}}(k) \leq \hat{\boldsymbol{\Theta}}^{\max}\\
&-\Delta\hat{\boldsymbol{\Theta}}^{\max} \leq \Delta\hat{\boldsymbol{\Theta}}(k) \leq \Delta\hat{\boldsymbol{\Theta}}^{\max}
\end{aligned} \tag{17}$$

where

$$\boldsymbol{\Theta}^{\mathrm{SP}}(k) = \begin{bmatrix} \boldsymbol{\theta}^{\mathrm{SP}}(k+1|k) \\ \vdots \\ \boldsymbol{\theta}^{\mathrm{SP}}(k+N|k) \end{bmatrix}, \bar{\boldsymbol{\Theta}}(k) = \begin{bmatrix} \bar{\boldsymbol{\theta}}(k+1|k) \\ \vdots \\ \bar{\boldsymbol{\theta}}(k+N|k) \end{bmatrix},$$

$$\Delta\hat{\boldsymbol{\Theta}}^{\max} = \begin{bmatrix} \Delta\hat{\boldsymbol{\theta}}^{\max} \\ \vdots \\ \Delta\hat{\boldsymbol{\theta}}^{\max} \end{bmatrix}, \Delta\hat{\boldsymbol{\Theta}}(k) = \begin{bmatrix} \Delta\hat{\boldsymbol{\theta}}(k|k) \\ \vdots \\ \Delta\hat{\boldsymbol{\theta}}(k+N_{\mathrm{u}}-1|k) \end{bmatrix},$$

$$\hat{\boldsymbol{\Theta}}^{\max} = \begin{bmatrix} \hat{\boldsymbol{\theta}}^{\max} \\ \vdots \\ \hat{\boldsymbol{\theta}}^{\max} \end{bmatrix}, \hat{\boldsymbol{\Theta}}(k) = \begin{bmatrix} \hat{\boldsymbol{\theta}}(k|k) \\ \vdots \\ \hat{\boldsymbol{\theta}}(k+N_{\mathrm{u}}-1|k) \end{bmatrix}.$$

The optimization task (16) minimizes the control error for the estimated target pan and tilt positions as well as excessive changes of their next positions. It is subject to a single constraint. The joints can be positioned between ±90°, i.e., ±1.5708 radians.

The first element of the predicted target estimated positions for both joints $\bar{\boldsymbol{\Theta}}(k)$ can be expanded into $\bar{\boldsymbol{\theta}}(k+1|k) = [\bar{\theta}_{\mathrm{pan}}(k+1|k) \quad \bar{\theta}_{\mathrm{tilt}}(k+1|k)]^{\mathrm{T}}$ and the last element into $\bar{\boldsymbol{\theta}}(k+N|k) = [\bar{\theta}_{\mathrm{pan}}(k+N|k) \quad \bar{\theta}_{\mathrm{tilt}}(k+N|k)]^{\mathrm{T}}$. Similarly, all elements of $\boldsymbol{\theta}^{\mathrm{SP}}(k)$, $\hat{\boldsymbol{\theta}}(k)$, $\Delta\hat{\boldsymbol{\theta}}(k)$, $\hat{\boldsymbol{\theta}}^{\max}$ and $\Delta\hat{\boldsymbol{\theta}}^{\max}$ are described, where $\hat{\boldsymbol{\theta}}^{\max} = [1.5708 \quad 1.5708]^{\mathrm{T}}$ and $\Delta\hat{\boldsymbol{\theta}}^{\max} = [0.002 \quad 0.002]^{\mathrm{T}}$. The matrix weighting the errors $\boldsymbol{\Psi} = \mathrm{diag}(\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N)$, where $\boldsymbol{\psi}_i = \left[\psi_{\mathrm{pan}} \ \ \psi_{\mathrm{tilt}}\right]^{\mathrm{T}}$, while the matrix weighting the excessive changes of control $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda}_0, \ldots, \boldsymbol{\lambda}_{N_{\mathrm{u}}-1})$, where $\boldsymbol{\lambda}_i = \left[\lambda_{\mathrm{pan}} \ \ \lambda_{\mathrm{tilt}}\right]^{\mathrm{T}}$. By manipulating the values of parameters inside the matrix $\boldsymbol{\Psi}$ an improvement of control performance can be obtained [25]. The result may be significantly better, when the parameters of matrix $\boldsymbol{\Lambda}$ are changed. However, this approach is less effective, if the setpoint trajectory constantly changes, as in our case.

For the considered robot neck two MPC algorithms are used: DMC and GPC.

DMC (*Dynamic Matrix Control*) algorithm uses a step response model to calculate the prediction of the controlled variables. First, the step response coefficients are gathered from a dynamic model of the process. Next, the step responses are used to create the following matrix

$$\boldsymbol{m}_p = \begin{bmatrix} m_p^{\text{pan,pan}} & m_p^{\text{pan,tilt}} \\ m_p^{\text{tilt,pan}} & m_p^{\text{tilt,tilt}} \end{bmatrix} \tag{18}$$

for $p = 1, \ldots, D$, where D denotes the horizon of sampling intervals needed for the output to stabilize. The step response matrix $\boldsymbol{m}_p$, in (18), is then used to formulate the following dynamic matrix

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_1 & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{m}_N & \boldsymbol{m}_{N-1} & \ldots & \boldsymbol{m}_{N-N_\text{u}+1} \end{bmatrix} \tag{19}$$

and also the past dynamic matrix

$$\boldsymbol{M}^\text{p} = \begin{bmatrix} \boldsymbol{m}_2 - \boldsymbol{m}_1 & \ldots & \boldsymbol{m}_D - \boldsymbol{m}_{D-1} \\ \vdots & \ddots & \vdots \\ \boldsymbol{m}_{N+1} - \boldsymbol{m}_1 & \ldots & \boldsymbol{m}_{N+D-1} - \boldsymbol{m}_{D-1} \end{bmatrix} \tag{20}$$

which are of dimensionality $2N \times 2N_\text{u}$ and $2N \times 2(D-1)$, respectively. Finally, the vector of the predicted controlled variables is calculated as follows

$$\hat{\hat{\boldsymbol{\Theta}}}(k) = \boldsymbol{M}\triangle\hat{\boldsymbol{\Theta}}(k) + \boldsymbol{\Theta}(k) + \boldsymbol{M}^\text{p}(k)\triangle\hat{\boldsymbol{\Theta}}^\text{p}(k) \tag{21}$$

where the vectors

$$\boldsymbol{\Theta}(k) = \begin{bmatrix} \boldsymbol{\theta}(k) \\ \vdots \\ \boldsymbol{\theta}(k) \end{bmatrix}, \ \triangle\hat{\boldsymbol{\Theta}}^\text{p}(k) = \begin{bmatrix} \triangle\hat{\boldsymbol{\theta}}(k-1) \\ \vdots \\ \triangle\hat{\boldsymbol{\theta}}(k-(D-1)) \end{bmatrix} \tag{22}$$

are of length $2N$ and $2(D-1)$, respectively. The minimized MPC cost function (16), with the use of the prediction equation (21), can be rewritten as

$$J_\text{DMC}(k) = \Big\| \boldsymbol{\Theta}^\text{SP}(k) - \boldsymbol{M}\triangle\hat{\boldsymbol{\Theta}}(k) - \boldsymbol{\Theta}(k) - \boldsymbol{M}^\text{p}(k)\triangle\hat{\boldsymbol{\Theta}}^\text{p}(k) \Big\|^2_{\boldsymbol{\Psi}} + \Big\| \triangle\hat{\boldsymbol{\Theta}}(k) \Big\|^2_{\boldsymbol{\Lambda}} \tag{23}$$

The cost function in (23) is quadratic with respect to $\triangle\hat{\boldsymbol{\Theta}}$. After differentiating it, we obtain

$$\frac{\mathrm{d}J_\text{DMC}(k)}{\mathrm{d}\triangle\hat{\boldsymbol{\Theta}}(k)} = -2\boldsymbol{M}^T\boldsymbol{\Psi}\Big(\boldsymbol{\Theta}^\text{SP}(k) - \boldsymbol{M}\triangle\hat{\boldsymbol{\Theta}}(k) - \boldsymbol{\Theta}(k) - \boldsymbol{M}^\text{p}\triangle\hat{\boldsymbol{\Theta}}^\text{p}(k)\Big) + 2\boldsymbol{\Lambda}\triangle\hat{\boldsymbol{\Theta}}(k) \tag{24}$$

The optimal solution vector is found by making the gradient vector calculated by equation (24) equal to zero and as a result the following formula is obtained

$$\triangle\hat{\boldsymbol{\Theta}}(k) = \boldsymbol{K}(\boldsymbol{\Theta}^\text{SP}(k) - \boldsymbol{\Theta}(k) - \boldsymbol{M}^\text{p}\triangle\hat{\boldsymbol{\Theta}}^\text{p}(k)) \tag{25}$$

where

$$\boldsymbol{K} = (\boldsymbol{M}^\text{T}\boldsymbol{\Psi}\boldsymbol{M} + \boldsymbol{\Lambda})^{-1}\boldsymbol{M}^\text{T}\boldsymbol{\Psi} \tag{26}$$

is a matrix of dimensionality $2N_\text{u} \times 2N$.

GPC (*Generalized Predictive Control*) algorithm is the foundation of the second approach. In this control approach, the vector of predicted controlled variables is found using the following formula

$$\bar{\boldsymbol{\Theta}}(k) = \boldsymbol{M}\triangle\hat{\boldsymbol{\Theta}}(k) + \boldsymbol{\Theta}^0(k) \tag{27}$$

where the matrix $\boldsymbol{M}$ is the same as in the DMC algorithm (19). The free trajectory vector, denoted as $\boldsymbol{\Theta}^0(k)$, is calculated using the dynamic model of the process

$$\boldsymbol{\Theta}^0(k) = \begin{bmatrix} \boldsymbol{\theta}^0(k+1|k) \\ \vdots \\ \boldsymbol{\theta}^0(k+N|k) \end{bmatrix} \tag{28}$$

where the first entry $\boldsymbol{\theta}^0(k+1|k)$ can be expanded into $\boldsymbol{\theta}^0(k+1|k) = [\theta^0_\text{pan}(k+1|k)\ \theta^0_\text{tilt}(k+1|k)]^\text{T}$ and the last element into $\boldsymbol{\theta}^0(k+N|k) = [\theta^0_\text{pan}(k+N|k)\ \ \theta^0_\text{tilt}(k+N|k)]^\text{T}$. The considered model, in this algorithm, is as follows

$$\boldsymbol{A}(q^{-1})\boldsymbol{\Theta}(k) = \boldsymbol{B}(q^{-1})\hat{\boldsymbol{\Theta}}(k) \tag{29}$$

where the entries of the matrices

$$\boldsymbol{A}(q^{-1}) = \begin{bmatrix} A_\text{pan}(q^{-1}) & 0 \\ 0 & A_\text{tilt}(q^{-1}) \end{bmatrix} \tag{30}$$

$$\boldsymbol{B}(q^{-1}) = \begin{bmatrix} B_\text{pan,pan}(q^{-1}) & B_\text{pan,tilt}(q^{-1}) \\ B_\text{tilt,pan}(q^{-1}) & B_\text{tilt,tilt}(q^{-1}) \end{bmatrix} \tag{31}$$

are the polynomials in the backward shift operator q^{-1}

$$\begin{aligned} A_\text{pan}(q^{-1}) &= 1 + a_{1,\text{pan}}q^{-1} + \ldots + a_{n_\text{A},\text{pan}}q^{-n_\text{A}} \\ A_\text{tilt}(q^{-1}) &= 1 + a_{1,\text{tilt}}q^{-1} + \ldots + a_{n_\text{A},\text{tilt}}q^{-n_\text{A}} \end{aligned} \tag{32}$$

$$
\begin{aligned}
&B_{\mathrm{pan,pan}}(q^{-1}) = b_{1,\mathrm{pan}}q^{-1} + \ldots + b_{n_\mathrm{B},\mathrm{pan}}q^{-n_\mathrm{B}}\\
&B_{\mathrm{pan,tilt}}(q^{-1}) = 0\\
&B_{\mathrm{tilt,tilt}}(q^{-1}) = b_{1,\mathrm{tilt}}q^{-1} + \ldots + b_{n_\mathrm{B},\mathrm{tilt}}q^{-n_\mathrm{B}}\\
&B_{\mathrm{tilt,pan}}(q^{-1}) = 0
\end{aligned}
\qquad (33)
$$

In GPC, the cost function (23) can be rewritten as

$$
\begin{aligned}
J_{\mathrm{GPC}}(k) = &\left\| \boldsymbol{\Theta}^{\mathrm{SP}}(k) - \boldsymbol{M}\triangle\hat{\boldsymbol{\Theta}}(k) - \boldsymbol{\Theta}^{0}(k) \right\|^2_{\boldsymbol{\Psi}}\\
&+ \left\| \triangle\hat{\boldsymbol{\Theta}}(k) \right\|^2_{\boldsymbol{\Lambda}}.
\end{aligned}
\qquad (34)
$$

Similarly to DMC, as the cost function (34) is quadratic in terms of the decision variables, the following solution applies

$$
\triangle\hat{\boldsymbol{\Theta}}(k) = \boldsymbol{K}(\boldsymbol{\Theta}^{\mathrm{SP}}(k) - \boldsymbol{\Theta}^{0}(k)). \qquad (35)
$$

It is to be noted that the only difference between DMC and GPC is in the way the free trajectory is calculated. Respective stability analyzes can be found for DMC in [12] and for GPC in [13,42]. Robust stability considerations are addressed in [8,15].

2.3.3 Simplified pan–tilt camera model

A continuous-time model of the considered robot neck is proposed using second-order transfer functions for pan and tilt joints, respectively.

$$
G_{\mathrm{pan}}(s) = \frac{K_{\mathrm{pan}}(T_{1,\mathrm{pan}}s + 1)}{(T_{2,\mathrm{pan}}s + 1)(T_{3,\mathrm{pan}}s + 1)}, \qquad (36)
$$

$$
G_{\mathrm{tilt}}(s) = \frac{K_{\mathrm{tilt}}(T_{1,\mathrm{tilt}}s + 1)}{(T_{2,\mathrm{tilt}}s + 1)(T_{3,\mathrm{tilt}}s + 1)}, \qquad (37)
$$

where K_{pan} and K_{tilt} are static gains and $T_{1,\mathrm{pan}}$, $T_{2,\mathrm{pan}}$, $T_{3,\mathrm{pan}}$ and $T_{1,\mathrm{tilt}}$, $T_{2,\mathrm{tilt}}$, $T_{3,\mathrm{tilt}}$ are respective time constants. Pan and tilt joint positions, denoted as $\hat{\theta}$, measured at the current instant k of simulation form the model inputs. Target positions estimated for the pan and tilt, denoted as θ^{SP}, are the model outputs. Transfer functions are discretized and rewritten in the form of the following difference equations applied to the MPC framework:

$$
\begin{aligned}
\theta_{\mathrm{pan}}(k) = &- a_{1,\mathrm{pan}} \cdot \theta_{\mathrm{pan}}(k-1) - a_{2,\mathrm{pan}} \cdot \theta_{\mathrm{pan}}(k-2)\\
&+ b_{1,\mathrm{pan}} \cdot \hat{\theta}_{\mathrm{pan}}(k-1), \qquad (38)\\
\theta_{\mathrm{tilt}}(k) = &- a_{1,\mathrm{tilt}} \cdot \theta_{\mathrm{tilt}}(k-1) - a_{2,\mathrm{tilt}} \cdot \theta_{\mathrm{tilt}}(k-2)\\
&+ b_{1,\mathrm{tilt}} \cdot \hat{\theta}_{\mathrm{tilt}}(k-1), \qquad (39)
\end{aligned}
$$

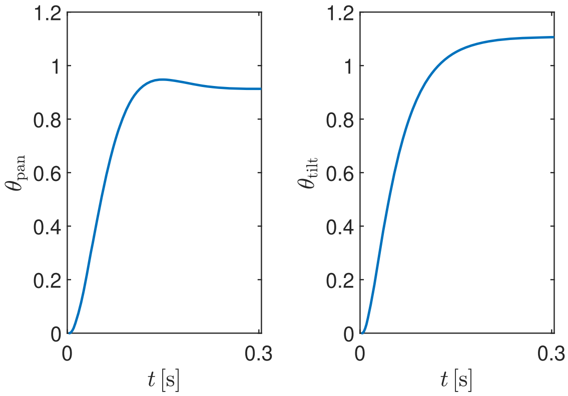

Fig. 4 Step responses of the neck simplified dynamics model; output of the pan joint – left plot, output of the tilt joint – right plot

where k is the time instant and a_1, a_2, b_1 are constant coefficients for pan and tilt joints, respectively. The following parameters of the model have been chosen through numerous simulations: $a_{1,\mathrm{pan}} = -1.91 \times 10^0$, $a_{2,\mathrm{pan}} = 9.14 \times 10^{-1}$, $b_{1,\mathrm{pan}} = 3.38 \times 10^{-3}$, $a_{1,\mathrm{tilt}} = -1.83 \times 10^0$, $a_{2,\mathrm{tilt}} = 8.39 \times 10^{-1}$ and $b_{1,\mathrm{tilt}} = 6.09 \times 10^{-3}$. These models approximate very well real robot neck joints. Their step responses are shown in Fig. 4.

Two performance indices are used to compare control strategies: integral absolute error (IAE), denoted as E_{IAE} (40), and mean squared error (MSE), denoted as E_{MSE} (41). The IAE index assigns equal weights to errors regardless of their absolute values. This index is robust with respect to outliers [32]. It is invaluable in checking the overall excessive changes of joint positions. If these changes are too large harmful vibrations of the neck of the real robot occur. This phenomenon can lead to potential damage of the mechanical parts of the robot. The excessive changes are minimal, so the IAE delivers more proper measure than a MSE index calculating a quadratic function of the control error.

$$
E_{\mathrm{IAE}} = \frac{1}{N}\sum_{l=1}^{k^{\max}} |\triangle\hat{\theta}(l)|, \qquad (40)
$$

where N is the total number of samples and $k^{\max}$ is the last sampling instant. MSE is used to take into account large control errors. They are magnified by the quadratic function.

$$
E_{\mathrm{MSE}} = \frac{1}{N}\sum_{l=1}^{k^{\max}} (\triangle\hat{\theta}(l))^2. \qquad (41)
$$

3 Results

The proposed control strategy is validated in two ways. First, the control scheme is implemented in the Gazebo simulator and tested with the Gazebo model of the robot. Secondly, the MPC controller is tested with a real Velma robot. The components of the system are implemented using the FABRIC framework (Framework for Agent-Based Robot Control Systems) [35].

3.1 Validation using the Gazebo simulator

The proposed control strategy is validated using the Gazebo simulator. All the simulations are performed in a discrete-time form, as the predictive control formulation is in a discrete-time form and the ultimate controller implementation is done on a real robot in the discrete-time form. The camera tracks sample recorded object trajectory representing a circular motion in two axes. Setpoint trajectory: $\theta_{\text{pan}}^{\text{SP}}$ for the pan joint and $\theta_{\text{tilt}}^{\text{SP}}$ for the tilt joint is provided by the visual estimation system. Three controllers are compared: PID, DMC and GPC. The PID uses the following parameters: $K_{\text{G}} = 0.1$, $T_{\text{I}} = 0.01\text{s}$ $T_{\text{D}} = 0.01\text{s}$. The DMC is set up with: $N = 50$, $N_{\text{u}} = 25$, $\lambda_{\text{pan}} = 100$, $\lambda_{\text{tilt}} = 100$, $\psi_{\text{pan}} = 1$ and $\psi_{\text{tilt}} = 1$, while the GPC uses the same prediction and control horizons as well as the penalty imposed on the control error as the DMC, but differs in the penalty imposed on the control increments $\lambda_{\text{pan}} = 100$, $\lambda_{\text{tilt}} = 10000$.

The setpoint trajectory is circular. As the pan–tilt predictive control is implemented in the KUKA LWR 4+ robot, which is envisaged to be used as a companion robot, it is not expected that the system will have to track fast moving objects that rapidly change the direction of their motion. Thus, the parameters of the trajectory comply with the assumed robot operating conditions.

Figure 5 shows the comparison of the performance of PID and DMC algorithms, while Fig. 6 depicts the operation of the two MPC algorithms: DMC and GPC. Table 2 presents respective performance indices for both joints. The IAE obtained for PID control has low enough values for both joints to prevent harmful effects on the robot. DMC control provides similar results in this regard. However, comparing the MSE, we notice that the DMC control gives much better results. The MSE values are much lower for both joints than in the PID control scenario. The DMC algorithm uses setpoint trajectory prediction over the prediction horizon, unavailable for PID. GPC algorithm, similarly to DMC, can utilize future changes in the setpoint trajectory. Still, the inclusion of past values of measured signals in the calculations tends to generate high control increments to quickly compensate for measurement noise, disturbances, or modeling inaccuracies. Therefore, a higher value of the penalty imposed on the control signal increments is often required to achieve stability of the controller, as it is in this case. Unfortunately, this causes a tradeoff with the control quality. As it is visible in Fig. 6 and the Table 2, even though the smoothness of the control signal is comparable to the PID and DMC algorithms, the quality is lower.

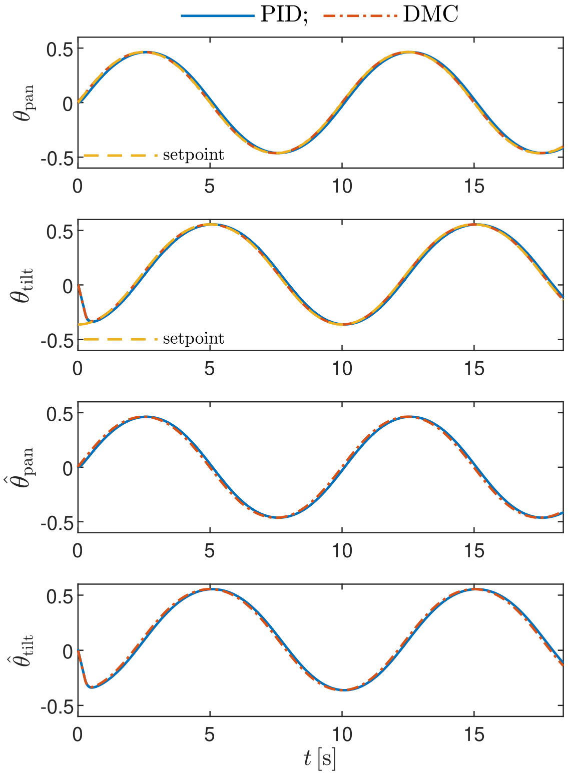

Fig. 5 Simulation results: trajectories of the control system with the PID controller vs. trajectories of the same system, but with the DMC controller; controlled variables of both joints – first two top plots, manipulated variables of both joints – two bottom plots

The MSE and IAE performance measures suggest the following interpretation. IAE index shows quite similar values, but MSE shows the superiority of the DMC control. It is due to the fact that each of these

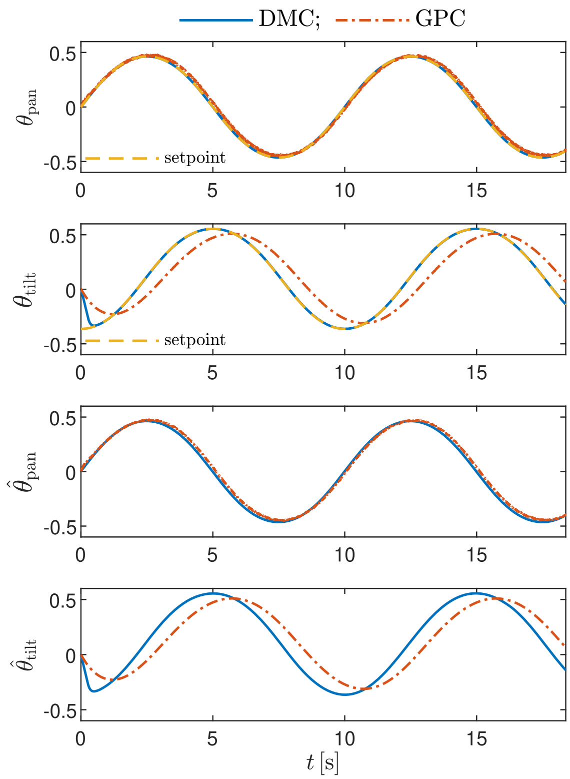

Fig. 6 Simulation results: trajectories of the control system with the DMC controller vs. trajectories of the same system, but with the GPC controller; controlled variables of both joints – first two top plots, manipulated variables of both joints – two bottom plots

indices considers different performance aspects. IAE compares normal operation, while MSE focuses on the operation in the case of rapid changes and transient periods, due to the squared error. Comparison of both measures justified the interpretation of the results. Hence the DMC algorithm was implemented and used in the real Velma robot.

3.2 Validation on the Velma robot

The DMC algorithm with the same parameters has been implemented on the Velma robot. A similar target movement has been used for validation purposes. Figure 7 presents the results for this case. The following values of the performance indices have been obtained: $E_{\text{IAE}}^{\text{pan}} = 2.05 \times 10^{-4}$, $E_{\text{IAE}}^{\text{tilt}} = 2.86 \times 10^{-4}$, $E_{\text{MSE}}^{\text{pan}} = 1.70 \times 10^{-4}$ and $E_{\text{MSE}}^{\text{tilt}} = 6.19 \times 10^{-3}$. The IAE values for both joints are much smaller than expected from the simulation results, what is even better. MSE index values are also not large. The values of these indices are close to the values obtained using the Gazebo simulator. The setpoint tracking is very accurate, also in the scenario with the use of the Velma robot. These results confirm that the assumed predictive control is appropriate.

We see that the obtained responses look very similar and the level of the accuracy improvement is quite low. However, we have to keep in mind the fact that pan–tilt control is not used just to move the robot neck. We use it to control the camera that tracks the object and the object should be kept in the middle of the image. When the tracked object is in the middle of the image pattern recognition is improved, especially as the time needed for object recognition becomes lower. Moreover, the tracked object can be located quite far away from the head and therefore even small inaccuracies in the angles start to matter. Thus, the pan–tilt head operation performance has to be kept in the visual object tracking context, not just as simply movement.

4 Conclusions and further research

This paper describes the results of research focused on evaluating predictive control of a pan–tilt camera system. The issue considered in this paper is part of a larger project related to the launch of a whole visual servoing system, comprising recognition and tracking of objects with an RGB-D camera and predictive control of a KUKA LWR 4+ manipulator.

The algorithm evaluation and validation consist of three steps. Matlab simulations are followed with the simulation using the Velma robot Gazebo simulator.

Table 2 Simulation results in terms of performance indices

Algorithm	$E_{\text{IAE}}^{\text{pan}}$	$E_{\text{IAE}}^{\text{tilt}}$	$E_{\text{MSE}}^{\text{pan}}$	$E_{\text{MSE}}^{\text{tilt}}$
PID	3.58×10^{-4}	4.08×10^{-4}	4.18×10^{-4}	1.38×10^{-3}
DMC	3.60×10^{-4}	4.08×10^{-4}	6.95×10^{-7}	9.22×10^{-4}
GPC	1.24×10^{-3}	3.32×10^{-4}	6.56×10^{-4}	2.12×10^{-2}

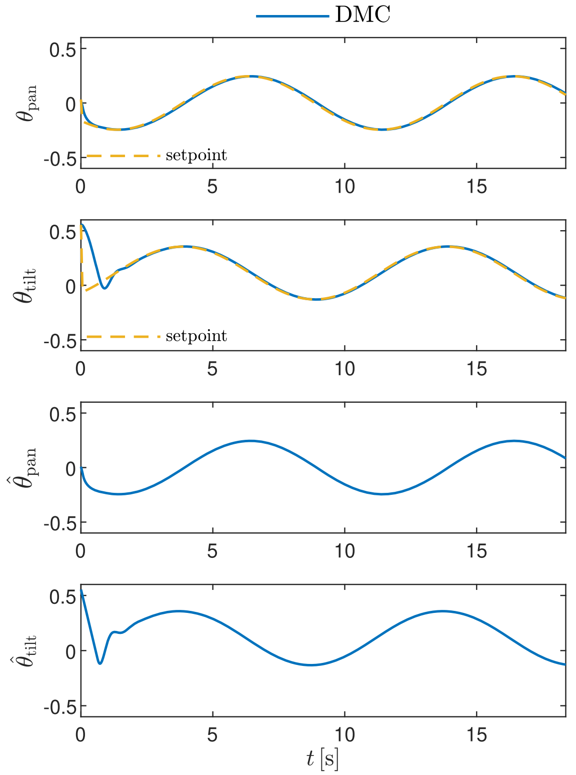

Fig. 7 Verification of the proposed control scheme with the Velma robot: trajectories of the control system using the DMC controller; controlled variables of both joints – first two top plots, manipulated variables of both joints – two bottom plots

Finally, the selected predictive controller is implemented in the real Velma robot. Initially, three control strategies were taken into account: PID algorithm and two predictive methods: DMC and GPC. Detailed simulations point out the superiority of the DMC predictive controller, which is eventually implemented on the Velma robot. It is interesting that it matters which predictive strategy is used. They are not equivalent. One has to remember that both predictive algorithms are not equivalent. There is a single difference in the type of the model used. One cannot use one or the other with complete freedom. In addition to the different forms of the model, the control law is implemented differently, what translates into their specificity and the resulting control performance. In our case, the properties of the DMC control are more predisposed to this application.

Real-time pan–tilt object tracking confirms initial assumptions about the DMC predictive strategy and enables further investigation of the entire visual servoing Velma robot project. It is shown that even the simple predictive strategy, such as the DMC, improves tracking control. It is mostly due the fact that the MPC approach allows to incorporate into the control rule definition the knowledge about the future behavior of the setpoint. It is the main advantage of any predictive control over PID-type controllers. It is shown that we do not have to use complex predictive approaches. Simple DMC algorithm, with a simple dynamical model, improves performance. Moreover, we have to be aware that practice prefers simple approaches. Simplicity probably slightly lowers the performance, but it increases the robustness, control system sustainability, facilitates human supervision (simpler knowledge) and lowers maintenance requirements. DMC introduces improvement while being simple.

One issue that requires further analysis is the possible impact of disturbances. Actually, they virtually have not not been considered here. Although a possible movement of the robot column can be treated as a disturbance. During the realization of this project, we consciously decided not to analyze this issue. We are aware that such disturbance can be easily included into the MPC control. But then, its comparison with the PID control would be unfair from the PID perspective. Thus, the PID control layout should be modified with the feedforward disturbance decoupling. Nonetheless, this subject is really worth to be considered and as such is planned in the future research. Uncertainties are harder to be considered, especially in the simulation, as we cannot evaluate the risk they introduce. We are fully aware that they exist and impact the project, as for instance the unknown and unmodeled dynamics, which certainly occurs.

Unmodeled dynamics and the impact of unknown disturbances constitute two separate issues. The effect of unmodeled dynamics has been already taken into account. The proposed linear model, which is used as the internal model of the predictive strategy is considered as a simplification of the real pan–tilt system. It is shown that the controller that uses such a simplified model achieves good control performance. Therefore, the unmodeled dynamics is considered in the research. The other problem is the inaccurate knowledge of kinematic parameters of the considered system, e.g., Denavit–Hartenberg parameters of the kinematic model. However, our system has been calibrated, and thus those inaccuracies do not influence the system performance to a noticeable extent.

The aspect of unknown disturbances constitutes a different issue. Those disturbances can be produced by quantization effects introduced by the camera. The position of the tracked object is determined on the basis of its position in the image; hence the resolution is of significance. However, this problem is solved either by applying a camera with sufficient resolution or by using a second low-resolution camera mounted on the end effector, as described in [19]. In that case the resolution of the eye-in-hand camera does not influence significantly, the precision of locating the object, because as the gripper gets nearer to the object the image of the object occupies more space in the image, thus can be located more precisely.

Further operational validation with the robot should be performed to observe them and to investigate their impact. As the algorithm robustness is a significant feature of an embedded control strategy, the research on pan–tilt robust control offers an interesting alternative for future research.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by RN, KZ, DS, PC and MF. The first draft of the manuscript was written by PDD and CZ and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The project was funded by POB Research Centre for Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative Program – Research University (ID-UB).

Data availability The datasets generated during and/or analyzed during the current study are not publicly available due to complexity of the experiments, but are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no relevant financial or nonfinancial interests to disclose.

References

1. Bemporad, A., Morari, M.: Robust model predictive control: a survey. In: Garulli, A., Tesi, A. (eds.) Robustness in Identification and Control, pp. 207–226. Springer, London, London (1999)
2. Chaumette, F.: Visual servoing. In: Ang, M.H., Khatib, O., Siciliano, B. (eds.) Encyclopedia of Robotics, pp. 1–9. Springer, Berlin, Heidelberg (2020)
3. Chaumette, F., Hutchinson, S.: Visual servo control. i. basic approaches. IEEE Robot. Autom. Mag. **13**(4), 82–90 (2006)
4. Chaumette, F., Hutchinson, S.: Visual servo control. ii. advanced approaches [tutorial]. IEEE Robot. Autom. Mag. **14**(1), 109–118 (2007)
5. Chen, H., Zhao, X., Tan, M.: A novel pan-tilt camera control approach for visual tracking. In: Proceeding of the 11th World Congress on Intelligent Control and Automation, pp. 2860–2865 (2014)
6. Chroust, S., Vincze, M.: Comparison of prediction methods for vision-based control of motion. In: IFAC Proceedings vol. **33**(27), pp. 207–212 (2000). 6th IFAC Symposium on Robot Control (SYROCO 2000), Vienna, Austria, 21–3 Sep 2000
7. Craig, J.J.: Introduction to Robotics, Mechanics & Control. Addison-Wesley (1986)
8. Dai, L., Cheng, Z.: Robust stability conditions for siso dynamic matrix control. Dev. Chem. Eng. Miner. Process. **9**(1–2), 49–56 (2001)
9. Domański, P.D.: Performance assessment of predictive control - a survey. Algorithms **13**(4), 97 (2020)
10. Faulwasser, T., Weber, T., Zometa, P., Findeisen, R.: Implementation of nonlinear model predictive path-following control for an industrial robot. IEEE Trans. Control Syst. Technol. **25**(4), 1505–1511 (2017)
11. Fehr, J., Schmid, P., Schneider, G., Eberhard, P.: Modeling, simulation, and vision-/mpc-based control of a powercube serial robot. Appl. Sci. **10**(20) (2020)
12. Garcia, C.E., Morshedi, A.: Quadratic programming solution of dynamic matrix control (QDMC). Chem. Eng. Commun. **46**(1–3), 73–87 (1986)
13. Gomma, H., Owens, D.: Stability analysis for generalized predictive control (gpc) and time varying weighting generalized predictive control (tgpc). In: Proceedings of the 2004 IEEE International Conference on Control Applications, 2004., vol. 2, pp. 1662–1667 Vol.2 (2004)
14. Han, S.E., Unel, M.: Stabilization of pan-tilt systems using acceleration based lmi-lqr controller. In: 2017 11th Asian Control Conference (ASCC), pp. 2552–2557 (2017)
15. Hongbin, D., Pingjing, Y.: Closed loop robust stability of mimo dynamic matrix control with input saturation. In: Chen, B., Westerberg A.W. (eds.) Process Systems Engineering 2003, 8th International Symposium on Process Systems Engineering, Computer Aided Chemical Engineering, vol. 15, pp. 790–795. Elsevier (2003)

16. Killpack, M.D., Kemp, C.C.: Fast reaching in clutter while regulating forces using model predictive control. In: 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 146–153 (2013)
17. Kleff, S., Meduri, A., Budhiraja, R., Nicolas Mansard, L.R.: High frequency nonlinear model predictive control of a manipulator. In: IEEE International Conference on Robotics and Automation (ICRA 2021). Xi'an, China (2021)
18. Kolesnikov, A.A.: Introduction of synergetic control. In: 2014 American Control Conference, pp. 3013–3016 (2014)
19. Kornuta, T., Zieliński, C.: Robot control system design exemplified by multi-camera visual servoing. J. Intell. Robot. Syst. **77**, 499–523 (2015)
20. Kumar, P., Dick, A., Sheng, T.S.: Real time target tracking with pan tilt zoom camera. In: 2009 Digital Image Computing: Techniques and Applications, pp. 492–497 (2009)
21. Li, J., Sun, J., Liu, L., Xu, J.: Model predictive control for the tracking of autonomous mobile robot combined with a local path planning. Meas. Control **0**(0), 00202940211043,070 (2021)
22. Maaß, F.L.: An analysis on object detection and tracking with a tilt-pan camera on an embedded device. Hamburg University of Applied Sciences, Tech. rep. (2020)
23. Mayne, D., Rawlings, J., Rao, C., Scokaert, P.: Constrained model predictive control: stability and optimality. Automatica **36**(6), 789–814 (2000)
24. Meribout, M., Nakanishi, M., Ogura, T.: A parallel algorithm for real-time object recognition. Pattern Recognit. **35**, 1917–1931 (2002)
25. Nebeluk, R., Marusak, P.: Efficient mpc algorithms with variable trajectories of parameters weighting predicted control errors. Arch. Control Sci. **30**(2), 325–363 (2020)
26. Nubert, J., Köhler, J., Berenz, V., Allgöwer, F., Trimpe, S.: Safe and fast tracking on a robot manipulator: robust mpc and neural network control. IEEE Robot. Autom. Lett. **5**, 3050–3057 (2020)
27. Park, J., Hwang, W., Bahn, W., hun Lee, C., il Kim, T., Shaikh, M.M., soo Kim, K., il "Dan" Cho, D.: Pan/tilt camera control for vision tracking system based on the robot motion and vision information. In: IFAC Proceedings Vol. **44**(1), pp. 3165–3170 (2011). 18th IFAC World Congress
28. Pebrianti, D., Peh, O., Samad, R., Mustafa, M., Abdullah, N.R.H., Bayuaji, L.: Intelligent control for visual servoing system. Indones. J. Electr. Eng. Comput. Sci. **6**, 72 (2017)
29. Qu, Y., Yang, B., Yu, Z., Song, Y., Sun, Y.: Active visual servo pan/tilt control design based on improved augmented lqr. In: 2020 Chinese Control And Decision Conference (CCDC), pp. 3702–3707 (2020)
30. Raimondo, D.M., Gasparella, S., Sturzenegger, D., Lygeros, J., Morari, M.: A tracking algorithm for ptz cameras. In: IFAC Proceedings vol. **43**(19), pp. 61–66 (2010). 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems
31. Ramelan, A., Saputro, J.S., Apribowo, C.H.B., Ibrahim, M.H., Pramono, S.: Design and simulation linear quadratic gaussian (lqg) for pan-tilt face tracking camera servos. AIP Conf. Proce. **2217**(1), 030,073 (2020)
32. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. John Wiley & Sons Inc, New York, NY, USA (1987)
33. Ruangpayoongsak, N.: Mobile robot positioning by using low-cost visual tracking system. MATEC Web Conf. **95**, 08,006 (2017)
34. Saragih, C.F.D., Kinasih, F.M.T.R., Machbub, C., Rusmin, P.H., Rohman, A.S.: Visual servo application using model predictive control (mpc) method on pan-tilt camera platform. In: 2019 6th International Conference on Instrumentation, Control, and Automation. ICA, pp. 1–7 (2019)
35. Seredyński, D., Winiarski, T., Zieliński, C.: FABRIC: framework for agent-based robot control systems. In: Kozłowski K. (ed.) 12th International Workshop on Robot Motion and Control (RoMoCo), pp. 215–222 (2019). https://doi.org/10.1109/RoMoCo.2019.8787370
36. Tang, Y., Bilodeau, G.A.: Evaluation of trackers for pan-tilt-zoom scenarios. ArXiv:1711.04260, Cornell Univesity Library (2017)
37. Tatjewski, P.: Advanced Control of Industrial Processes, Structures and Algorithms. Springer, London (2007)
38. Tran, P., Nguyen, H., Nguyen, A., Le, T., Phan, H., Nguyen1, C.: Design control system for pan-tilt camera for visual tracking based on adar method taking into account energy output. In: E3S Web Conferences - III International Conference "Energy Efficiency and Energy Saving in Technical Systems, vol. 279, p. 02007 (2021)
39. Tsai, C.Y., Song, K.T., Dutoit, X., Brussel, H., Nuttin, M.: Robust mobile robot visual tracking control system using self-tuning kalman filter. In: Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA 2007, pp. 161–166 (2007)
40. Yosafat, S.R., Machbub, C., Hidayat, E.M.I.: Design and implementation of pan-tilt control for face tracking. In: 2017 7th IEEE International Conference on System Engineering and Technology (ICSET), pp. 217–222 (2017)
41. Zhang, B., Huang, J., Lin, J.: A novel control algorithm for object tracking by controlling pan/tilt automatically. In: 2010 2nd International Conference on Education Technology and Computer, vol. 1, pp. V1–596–V1–602 (2010)
42. Zhang, J., Xi, Y.: Some gpc stability results. Int. J. Control **70**, 831–840 (1998)
43. Zhao, Y.H., Chai, L., Jin, L.Z.: The application of pid-dmc algorithm with feedforward compensation in the ptz platform. IOP Conf. Ser. Mater. Sci. Eng. (2018)
44. Żabiński, T., Grygiel, T., Kwolek, B.: Design and implementation of visual feedback for an active tracking. Mach. Graph. Vis. **15**(3–4), 681–690 (2006)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.