Interpolating Control Based Trajectory Tracking*

Zdeněk Bouček and Miroslav Flídr

Abstract— **The paper is dealing with a modification of Interpolating Control (IC) for the employment in trajectory tracking problem with inherent constraints. First, the Optimal Control Problem (OCP) with reference trajectory tracking is described. The OCP is hard to solve analytically, thus, two feasible approaches will be presented. The standard method for the trajectory tracking called Model Predictive Control (MPC), and afterward, a more computationally efficient alternative, the IC will be described. Further, the modification of IC for the tracking of the reference trajectory is presented. Finally, the IC and MPC are compared on a simple example using OCP cost function and computational time.**

I. INTRODUCTION

In many control applications, tracking of reference trajectory is required. To ensure the control strategy delivers the highest quality, it is advantageous to consider not only the current reference point of trajectory but also its future development. Additionally, limitations of systems must be considered during designing the control strategy. The demands on the quality of control are usually described in the form of a cost function. To solve this kind of problem, it can be described as the constrained Optimal Control Problem (OCP) [1]. Nevertheless, it can be complicated to solve OCP in real-time, especially for systems with fast dynamics. Fortunately, some suboptimal control methodology can be employed to make the problem more tractable.

The most widely used example of such a methodology is the Model Predictive Control (MPC) [2], [3], [4], [5], [6]. The MPC provides a solution to the problem on the shorter receding horizon. Thanks to the direct incorporation of the prediction in the control strategy acquisition, the MPC is capable to consider the future development of the reference trajectory. However, the consideration of a significant part of the future trajectory can result in a major increase in complexity and therefore in much higher computational time demands.

Another methodology for solving the constrained control problems is the Interpolating Control (IC) [7], [8], [9], [10]. It has already been proven to be a decent alternative to the MPC. Its main advantage is much lower complexity [7] which results in reducing computational demands and simple implementation, with comparable control quality [9].

Currently, only the modification of IC for the control to the setpoint was presented [10], where the future development of the reference trajectory was not considered.

The main goal of this article is to present an adjusted IC which can reflect the whole reference trajectory in the acquisition of the control law. First, the adjustment of the IC will be discussed. Second, the performance of the modified method will be studied.

The paper is structured as follows. In the next section, the general constrained trajectory-tracking OCP for a discrete linear time-invariant system with linear constraints is described. Afterward, the trajectory-tracking MPC is presented. Next, the paper focuses on the main topic, i.e. inclusion of reference trajectory in the acquisition of the IC control law. Finally, both described methodologies are compared using several standard solvers for better independence of results.

II. TRAJECTORY TRACKING PROBLEM

In this section, a general constrained trajectory-tracking OCP will be presented, and further, the special case of this OCP will be described.

The controlled system is considered as a discrete-time linear time-invariant (LTI) system with linear constraints. The optimization problem for trajectory tracking is formulated as

$$J\left(\boldsymbol{x}_0, \boldsymbol{u}_0^M\right) = \left(\boldsymbol{x}_M - \boldsymbol{r}_M\right)^\top \boldsymbol{Q}\left(\boldsymbol{x}_M - \boldsymbol{r}_M\right) + \sum_{k=0}^{M-1} \left(\boldsymbol{x}_k - \boldsymbol{r}_k\right)^\top \boldsymbol{Q}\left(\boldsymbol{x}_k - \boldsymbol{r}_k\right) + \boldsymbol{u}_k^\top \boldsymbol{R} \boldsymbol{u}_k, \tag{1}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k,\ k = 0, 1, 2, \ldots, M, \tag{2}$$

$$\boldsymbol{x}_k \in \mathcal{X}, \mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{F}^x \boldsymbol{x} \leq \boldsymbol{g}^x\}, \tag{3}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \mathcal{U} = \{\boldsymbol{u} \in \mathbb{R}^m : \boldsymbol{F}^u \boldsymbol{u} \leq \boldsymbol{g}^u\}, \tag{4}$$

where a long control horizon $M \gg 0$ is considered. The system is controlled along the given reference trajectory $\boldsymbol{r}_0^M$. The weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ of the quadratic cost function (1) are known symmetric positive semidefinite and positive definite, respectively. The quantities $\boldsymbol{x}_k \in \mathbb{R}^n$ and $\boldsymbol{u}_k \in \mathbb{R}^m$ are a state and control vector at time instant k, respectively.

The optimization constraints (2)-(4) represent the linear system dynamics and the inequalities with given matrices $\boldsymbol{F}^x$, $\boldsymbol{F}^u$ and vectors $\boldsymbol{g}^x$, $\boldsymbol{g}^u$ constraining the state space and control actions.

A solution to the OCP is a control strategy that minimizes the optimality criterion and steers the system along the given reference trajectory respecting the constraints at the same

This work was partially funded by ECSEL JU program and by the Czech Ministry of Education, Youth and Sports as a part of the Comp4Drones project with grant No. 826610 and 8A19004, respectivelly.

The authors are with the European Centre of Excellence: New Technologies for the Information Society (NTIS) & Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia, Czechia. Email: zboucek@ntis.zcu.cz, flidr@ntis.zcu.cz

time. Unfortunately, the closed-form solution is hard to obtain mainly due to the very long control horizon. The computational and storage demands are also prohibitive. This necessitates the employment of some suitable approximation of the OCP that makes the problem more tractable.

Many feasible solutions to the OCP are based on the employment of the standard Linear Quadratic Regulator (LQR) law [1]. This control law is optimal for the OCP given only by relations (1)-(2) without consideration of constraints. The LQR law is determined by solving the Bellman optimization recursion that leads in the time-invariant case to algebraic Ricatti equation

$$\boldsymbol{P} = \boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{A} + \boldsymbol{Q} - \left(\boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{B}\right)\left(\boldsymbol{B}^\top \boldsymbol{P} \boldsymbol{B} + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{B}\right)^\top, \tag{5}$$

where $\boldsymbol{P} \in \Re^{n \times n}$ is a symmetric positive semidefinite matrix. The LQR law is given in the form of the state-feedback controller

$$\boldsymbol{u}_k(\boldsymbol{x}_k) = \boldsymbol{K}\boldsymbol{x}_k \tag{6}$$

with state feedback gain $\boldsymbol{K} \in \Re^{n \times m}$

$$\boldsymbol{K} = -\left(\boldsymbol{B}^\top \boldsymbol{P} \boldsymbol{B} + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{B}\right)^\top. \tag{7}$$

For the control following the reference trajectory $\boldsymbol{r}_k^M$, the LQR law is according to [1] given by

$$\boldsymbol{u}_k(\boldsymbol{x}_k, \boldsymbol{r}_k^{k+M}) = \boldsymbol{K}\boldsymbol{x}_k + \boldsymbol{L}_k, \tag{8}$$

where the compensation for the trajectory tracking $\boldsymbol{L}_k \in \Re^m$ is described as

$$\boldsymbol{L}_k = -\left(\boldsymbol{B}^\top \boldsymbol{P} \boldsymbol{B} + \boldsymbol{R}\right)^{-1} \boldsymbol{B}^\top \boldsymbol{b}_{k+1}, \tag{9}$$

where $\boldsymbol{P}$ is the same as for the stabilization to the origin and

$$\boldsymbol{b}_k = \left(\boldsymbol{A}^\top + \boldsymbol{K}\boldsymbol{B}^\top\right)\boldsymbol{b}_{k+1} - \boldsymbol{Q}\boldsymbol{r}_k, \;\; \boldsymbol{b}_M = 0. \tag{10}$$

In the case of the LTI system, it is possible to calculate the compensation $\boldsymbol{L}$ in advance according to recursive relation (9). In this case, the control law is given as follows

$$\boldsymbol{u}_k(\boldsymbol{x}_k, \boldsymbol{r}_k^{k+M}) = \boldsymbol{K}\boldsymbol{x}_k + \bar{\boldsymbol{L}}\boldsymbol{r}_k^{k+M}, \tag{11}$$

$$\bar{\boldsymbol{L}}\boldsymbol{r}_k^{k+M} = \sum_{i=0}^{M} \bar{\boldsymbol{L}}_i \cdot \boldsymbol{r}_{k+i}, \tag{12}$$

$$\begin{aligned} \bar{\boldsymbol{L}}_i = &-\left(\boldsymbol{B}^\top \boldsymbol{P} \boldsymbol{B} + \boldsymbol{R}\right)^{-1} \boldsymbol{B}^\top \\ &\cdot \left(\left(\boldsymbol{A}^\top + \boldsymbol{K}\boldsymbol{B}^\top\right)^{(i-1)} - \boldsymbol{Q}\right). \end{aligned} \tag{13}$$

III. MODEL PREDICTIVE CONTROL

The MPC [2], [3], [4] reduces the complexity of the constrained OCP by solving the OCP on a much shorter control horizon and employs a receding horizon policy, which means that at each time instant only the control $\boldsymbol{u}_k$, that is given as a solution to a particular OCP at the time instant k, is applied. As the name implies the MPC is a model-based control methodology. The MPC is the state-of-the-art methodology for the trajectory tracking problem because it inherently uses prediction for the acquisition of control strategy and at the same time it can consider given constraints. Therefore, it can reflect the future reference trajectory and control the system smoothly.

In the past, it used to be employed only in the systems with slow dynamics due to the necessity of solving the problem on-line at each time instant. Over the years, it has been investigated in various perspectives by both academics and industry, and it has become a standard in the constrained control. With current computational power, the MPC is applicable even on such fast processes as an Unmanned Aerial Vehicle [11], [12].

In this paper, the linear MPC is considered, where a deterministic discrete-time LTI system and a quadratic criterion are incorporated. The description of the MPC is similar to the OCP (1)-(4) with a reformulated criterion, that is at each time instant given as

$$\begin{aligned} J\left(\boldsymbol{x}_k, \boldsymbol{u}_k^{k+N}, N\right) = &\left(\boldsymbol{x}_N - \boldsymbol{r}_N\right)^\top \boldsymbol{Q}\left(\boldsymbol{x}_N - \boldsymbol{r}_N\right) \\ &+ \sum_{l=k}^{k+N-1} \left(\boldsymbol{x}_l - \boldsymbol{r}_l\right)^\top \boldsymbol{Q}\left(\boldsymbol{x}_l - \boldsymbol{r}_l\right) + \boldsymbol{u}_l^\top \boldsymbol{R}\boldsymbol{u}_l, \end{aligned} \tag{14}$$

for the N-step control horizon, where N is the length of the receding horizon. The MPC is by nature a dynamic optimization problem. A QP solver is commonly used to solve the MPC, nevertheless, the QP is unlike the MPC a static optimization problem. Thus, it is necessary to transform the MPC into the form of a static problem. There are two main forms used for this transformation, the sparse and dense [6], which can be acquired either analytically or using software such as CVXGEN [13] or YALMIP [14].

In addition to an implicit solution to the MPC, which consists of a repeated solution to the problem at each time instant, it is possible to obtain an explicit solution to the MPC [2], [5] for the whole state-space (or for the desired subset). The solution can be acquired using multi-parametric programming. However, there are drawbacks in computing demands (nearly impossible in higher dimensions), in memory demands because of storing the whole solution in a control device and in a time-consuming searching within the solution for its use in control. Moreover, it is important to mention that there is a significant increase of multi-parametric problem dimensionality with free trajectory tracking (the solution depends not only on the state $\boldsymbol{x}_k$ but rather on the whole trajectory $\boldsymbol{r}_k^{k+N}$).

IV. INTERPOLATING CONTROL BASED TRAJECTORY TRACKING

Another promising methodology for trajectory tracking is the IC [7], [8], [9] which is based on the interpolation between a couple or several state-feedback gain control laws designed without consideration of the inherent constraints. Using the invariant set theory, it is ensured that the constraints are not violated. Thanks to the fusion of several control laws with known positively invariant sets, the IC applies to the same type of problems as the MPC.

Knowing the positively invariant set for each control law, the IC searches only an optimal coefficient for the interpolation between control laws as a solution to a simple LP. More-

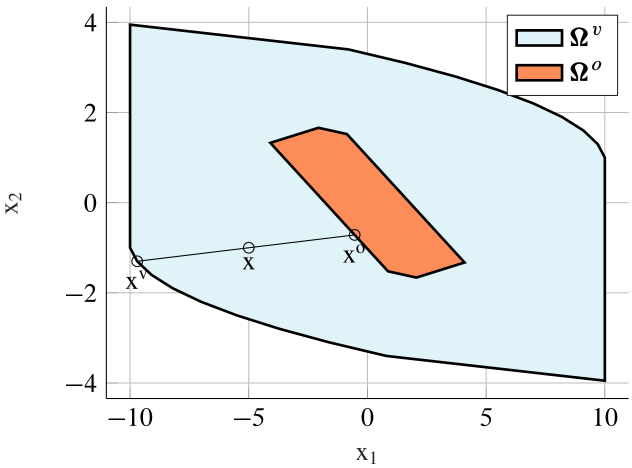

Fig. 1. Example of state decomposition

over, its explicit solution for the stabilization to the origin has low complexity. The IC is fully comparable with the MPC considering the control quality [9] and its modified version for the setpoint control was also studied in [10]. In this paper, the control laws are considered in the form of trajectory-tracking LQR which was presented in Section II. Thanks to the employment of LQR, the tracking ability can also be incorporated into IC.

For the interpolation, the principle of state decomposition (see Fig. 1) is employed, which can be denoted as

$$\boldsymbol{x} = c\boldsymbol{x}^v + (1-c)\,\boldsymbol{x}^o, \tag{15}$$

where $\boldsymbol{x}$ is the state vector, c is the interpolating coefficient, $c \in \langle 0, 1\rangle$, and $\boldsymbol{x}^o$ and $\boldsymbol{x}^v$ is the state vector for the high-gain and the low-gain controller, respectively. It is possible to reflect the decomposed state (15) in the interpolating control law as follows

$$\boldsymbol{u}(\boldsymbol{x}) = c\boldsymbol{u}^v(\boldsymbol{x}^v) + (1-c)\,\boldsymbol{u}^o(\boldsymbol{x}^o), \tag{16}$$

where $\boldsymbol{u}^0(\boldsymbol{x}^o)$ is the high-gain control law for $\boldsymbol{x}^o$ and $\boldsymbol{u}^v(\boldsymbol{x}^v)$ is the low-gain control law for $\boldsymbol{x}^v$.

The unconstrained LQR operates within a limited region given by constraints (depicted in Fig. 1 with the orange area) which can be described using the positively invariant set $\mathbf{\Omega}^o$ or $\mathbf{\Omega}^v$. If the state of the system reaches the positively invariant set, it will remain inside [7]. The acquisition of set is identical for each LQR; therefore, it will be described below for the high-gain controller with the positively invariant set $\mathbf{\Omega}^o$ by a polytope with the half-space representation as

$$\mathbf{\Omega}^o = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{F}^o\boldsymbol{x} \le \boldsymbol{g}^o\}. \tag{17}$$

If $\boldsymbol{F}^o$ and $\boldsymbol{g}^o$ are found such that $\mathbf{\Omega}^o$ is maximal, then it is called the maximal positively invariant set $\mathbf{\Omega}^o_{\max}$. The set is searched using the closed-loop dynamics denoted with the matrix of dynamics as

$$\boldsymbol{A}_{c_k} = \boldsymbol{A}_k + \boldsymbol{B}_k\boldsymbol{K}^o, \tag{18}$$

where $\boldsymbol{K}^o$ is the gain of the LQR. Algorithm for the acquisition of $\mathbf{\Omega}^o$ and $\mathbf{\Omega}^o_{\max}$ is provided in App. VIII. Within the minimization of the interpolating coefficient c, the property $\boldsymbol{x}^o \in \mathbf{\Omega}^o$ must be satisfied.

For the successful tracking respecting the constraints, the whole reference trajectory must be located in the invariant set $\mathbf{\Omega}^v$ including its boundary, $\boldsymbol{r}_k^{k+N} \in \mathbf{\Omega}^v$.

As has been said, the IC depends on finding the optimal interpolation coefficient c^*, which determines the decomposition of the state (15) and subsequently the resulting control (16). The optimal ratio c^* can be acquired by minimizing the criterion $J(\boldsymbol{x}, c)$ in the following NLP

$$\begin{aligned} J(\boldsymbol{x}^v, c) &= c, && (19)\\ \text{s.t.}\ \boldsymbol{F}^v\boldsymbol{x}^v &\le \boldsymbol{g}^v, && (20)\\ \boldsymbol{F}^o(\boldsymbol{x}^o - \boldsymbol{r}) &\le \boldsymbol{g}^o, && (21)\\ c\boldsymbol{x}^v + (1-c)\,\boldsymbol{x}^o &= \boldsymbol{x}, && (22)\\ 0 \le c &\le 1, && (23) \end{aligned}$$

where $\boldsymbol{r} = \boldsymbol{r}_k$, $\boldsymbol{x} = \boldsymbol{x}_k$. Using an auxiliary variable $\boldsymbol{z}^v = c \cdot \boldsymbol{x}^v$ the problem becomes linear

$$\begin{aligned} J(\boldsymbol{z}^v, c) &= c, && (24)\\ \text{s.t.}\ \boldsymbol{F}^v\boldsymbol{z}^v &\le c\boldsymbol{g}^v, && (25)\\ \boldsymbol{F}^o(\boldsymbol{x} - \boldsymbol{z}^v) &\le (1-c)\,(\boldsymbol{g}^o + \boldsymbol{F}^o\boldsymbol{r}), && (26)\\ 0 \le c &\le 1. && (27) \end{aligned}$$

where c^* is found by minimizing the criterion $J(\boldsymbol{x}^v, c)$ or $J(\boldsymbol{z}^v, c)$. In Eq. (21), it can be seen that $\mathbf{\Omega}^o$ is shifted by the coordinates of $\boldsymbol{r}_k$. To keep the interpolating coefficient optimal and to ensure that the constraints are not violated, the LP must be solved at each time instant k.

After obtaining a solution of the LP, the decomposed control laws are calculated as follows

$$\begin{aligned} \boldsymbol{u}^v &= \boldsymbol{K}^v\boldsymbol{z}^v + c\bar{\boldsymbol{L}}^v\boldsymbol{r}_k^{k+N}, && (28)\\ \boldsymbol{u}^o &= \boldsymbol{K}^o\boldsymbol{z}^o + (1-c)\bar{\boldsymbol{L}}^o\boldsymbol{r}_k^{k+N}, && (29) \end{aligned}$$

where $\boldsymbol{z}^o = x - \boldsymbol{z}^v$.

In case the set $\mathbf{\Omega}^v \setminus \mathbf{\Omega}^o$ is large, the performance of the IC can be problematic. Fortunately, the performance can be improved by adding an intermediate set

$$\begin{aligned} \mathbf{\Omega}^s &= \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{F}^s\boldsymbol{x} \le \boldsymbol{g}^s\}, && (30)\\ \mathbf{\Omega}^v &\subset \mathbf{\Omega}^s \subset \mathbf{\Omega}^o, && (31) \end{aligned}$$

where another LQR is defined. This state decomposition is presented in Fig. 2. This controller can be obtained for example with an increase in the weight $\boldsymbol{R}$. The set $\mathbf{\Omega}^s$ is calculated in the same way as $\mathbf{\Omega}^o$ with Algorithm 1 and as in case of $\mathbf{\Omega}^o$, it is also shifted by $\boldsymbol{r}_k$. This version of the IC will be further denoted as eIC. In the eIC, there are two different LPs. If $\boldsymbol{x} \in \mathbf{\Omega}^v \setminus \mathbf{\Omega}^o$, the interpolation is done between $\boldsymbol{u}^v$ and $\boldsymbol{u}^s$. If the $\boldsymbol{x} \in \mathbf{\Omega}^s$, the interpolation is performed for both $\boldsymbol{u}^s$ and $\boldsymbol{u}^o$. For an easier analysis of the IC performance, there are two different interpolating coefficients c_1 and c_2.

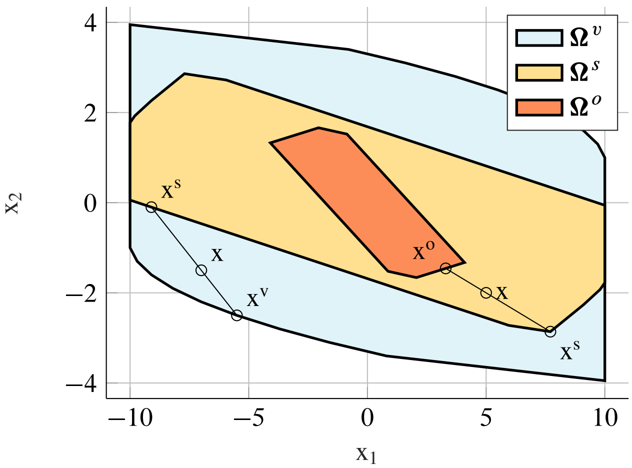

Fig. 2. Example of state decomposition with three invariant sets

In case $x \in \boldsymbol{\Omega}^v \setminus \boldsymbol{\Omega}^s$ the LP is analogical to previously denoted LP for IC with substituted $\boldsymbol{\Omega}^o$ for $\boldsymbol{\Omega}^s$

$$J\left(\boldsymbol{z}^v, c_1\right) = c_1, \tag{32}$$
$$\text{s.t.} \quad \boldsymbol{F}^v \boldsymbol{z}^v \leq c \boldsymbol{g}^v, \tag{33}$$
$$\boldsymbol{F}^s\left(\boldsymbol{x} - \boldsymbol{z}^v\right) \leq \left(1 - c_1\right)\left(\boldsymbol{g}^s + \boldsymbol{F}^s \boldsymbol{r}\right), \tag{34}$$
$$0 \leq c_1 \leq 1, \tag{35}$$

and $c_2 = 0$. The control law is given as follows

$$\boldsymbol{u}^v = \boldsymbol{K}^v \boldsymbol{z}^v + c_1 \bar{\boldsymbol{L}}^v \boldsymbol{r}_k^{k+N}, \tag{36}$$
$$\boldsymbol{u}^s = \boldsymbol{K}^s \boldsymbol{z}^s + (1 - c_1) \bar{\boldsymbol{L}}^s \boldsymbol{r}_k^{k+N}, \tag{37}$$

where $\boldsymbol{z}^s = x - \boldsymbol{z}^v$.

If $\boldsymbol{x} \in \boldsymbol{\Omega}^s$, the form of LP is different because both sets $\boldsymbol{\Omega}^o$ and $\boldsymbol{\Omega}^s$ are shifted to the $\boldsymbol{r}_k$ and the LP is described as follows

$$J\left(\boldsymbol{z}^s, c_2\right) = c_2, \tag{38}$$
$$\text{s.t.} \quad \boldsymbol{F}^s \boldsymbol{z}^s \leq c_2 \boldsymbol{g}^s, \tag{39}$$
$$\boldsymbol{F}^o\left(\boldsymbol{x} - \boldsymbol{r} - \boldsymbol{z}^s\right) \leq \left(1 - c_2\right) \boldsymbol{g}^o, \tag{40}$$
$$0 \leq c_2 \leq 1, \tag{41}$$

where $\boldsymbol{z}^s = c_2 \cdot \boldsymbol{x}^s$ is new auxiliary variable and $c_1 = 0$ and the control law is calculated as

$$\boldsymbol{u}^s = \boldsymbol{K}^s \boldsymbol{z}^s + c_2 \bar{\boldsymbol{L}}^s \boldsymbol{r}_k^{k+N}, \tag{42}$$
$$\boldsymbol{u}^o = \boldsymbol{K}^o \boldsymbol{z}^o + (1 - c_2) \bar{\boldsymbol{L}}^o \boldsymbol{r}_k^{k+N}, \tag{43}$$

where $\boldsymbol{z}^o = x - \boldsymbol{z}^s$.

V. EXAMPLE SETUP

The basic versions of IC and MPC were compared in [7] and it has been proven that the IC reaches much lower computational demands and complexity than the standard MPC for LTI and linear time-varying (LTV) system.

In [10], the IC for control to the constant setpoint was investigated. It has been presented that IC is advantageous because of much lower complexity even in the explicit solution and in some scenarios it delivered even slightly better performance than MPC.

In this section, a comparison between the IC and the MPC will be presented on the problem of trajectory tracking for the 2nd order LTI system. The IC is implemented according to the description in Section IV with the employment of several LQRs described in Section II. The IC is intended in the implicit form where the simple LP is solved at each time instant.

Both methods were implemented in MATLAB 2018b using YALMIP [14], which is a toolbox for modeling and solving optimization problems such as semidefinite, quadratic, or linear programming. In YALMIP all well-known solvers are interfaced, which was utilized in this paper to achieve the high independence of results. The QP and LP were solved using SEDUMI, GUROBI, and CPLEX solvers with identical parameters for the precision of the solution. In addition to YALMIP another toolbox called Multi-Parametric Toolbox 3 (MPT3) [5], which is a MATLAB toolbox for multi-parametric optimization, computational geometry, and MPC, was employed for calculation of invariant sets and detection of active sets in IC.

For the tests, the following deterministic discrete-time LTI system with constrained state and control vectors

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} 1 \\ 0.3 \end{bmatrix} \boldsymbol{u}_k, \tag{44}$$
$$\begin{bmatrix} -10 \\ -5 \end{bmatrix} \leq \boldsymbol{x}_k \leq \begin{bmatrix} 10 \\ 5 \end{bmatrix}, \; -1 \leq \boldsymbol{u}_k \leq 1 \tag{45}$$

is considered. The quadratic criterion is defined with the constant weights $\boldsymbol{Q} = \boldsymbol{I}_2$, $\boldsymbol{R} = 1$. Both, MPC and IC, were designed for the N-steps where $N = 150$ is equal to the length of the simulation. For the design of the IC and included LQRs, the weight matrices were chosen as follows

$$\boldsymbol{Q}^o = \boldsymbol{Q}, \boldsymbol{R}^o = \boldsymbol{R}, \boldsymbol{Q}^v = \begin{bmatrix} 10^{-2} & 0 \\ 0 & 5^{-2} \end{bmatrix}, \boldsymbol{R}^v = 10 \cdot \boldsymbol{R}, \tag{46}$$

where the LQR for the invariant set $\boldsymbol{\Omega}^v$ was designed to cover as large volume as possible with the increase of weight for control action penalty $\boldsymbol{R}^v$ and decrease of weight for state penalty $\boldsymbol{Q}^v$ according to state constraints (45).

The eIC includes the additional set $\boldsymbol{\Omega}^s$ with LQR. A suitable $\boldsymbol{\Omega}^s$ was acquired with an increase of the weight $\boldsymbol{R}^s = 10 \cdot \boldsymbol{R}^o = 10$ opposed to $\boldsymbol{\Omega}^o$. Other LQRs in the eIC remained unchanged.

The controllers were tested in several scenarios with different reference trajectories. The initial conditions were given by the origin of state-space as zero coordinates. The evaluation was performed according to the quadratic criterion (1) and it was carried out using MATLAB 2018b in discrete-time simulations using a standard desktop PC with Intel Core i7-4790 and 32GB DDR3 RAM. For the test with computational time, all simulations were performed for 100 times.

The tracking ability of controllers was tested in simulation with three different reference trajectories, where the reference for the second state was always equal to zero. The first reference trajectory for the first state is composed of a sine

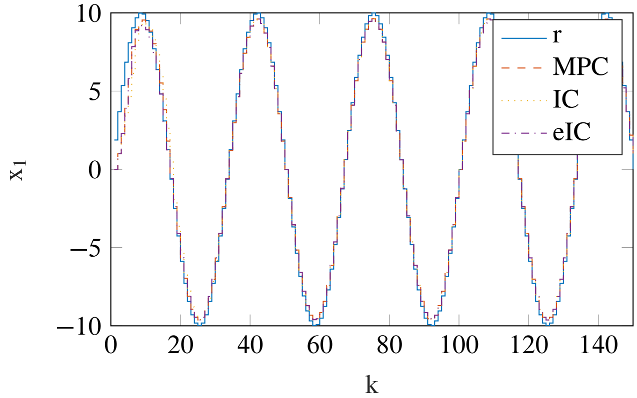

Fig. 3. The first state of the controlled system tracking the sine wave reference signal

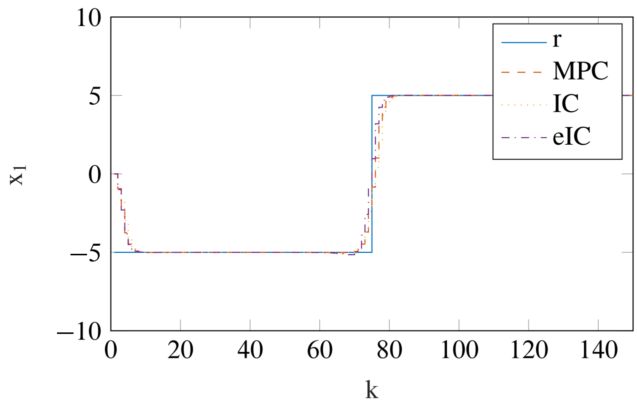

Fig. 4. The first state of the controlled system tracking the step reference signal

wave signal depicted in Fig. 3 and it is described as follows

$$\boldsymbol{r}_k = \left[10 \cdot \sin\left(\frac{9\pi k}{N}\right), 0 \right]^{\mathsf{T}}, \tag{47}$$

where N is the length of simulation.

For the second case, the reference was a step function which requested the system to move from -5 to 5 described as

$$\boldsymbol{r}_k = \begin{cases} [-5, 0]^{\mathsf{T}}, & k < \frac{N}{2}, \\ [5, 0]^{\mathsf{T}}, & k \geq \frac{N}{2} \end{cases} \tag{48}$$

and it is shown in Fig. 4.

In the last simulation, the triangular function was followed, which is denoted as

$$\boldsymbol{r}_k = \left[10 - 20 \left| \frac{2k}{N} - 1 \right|, 0 \right]^{\mathsf{T}} \tag{49}$$

and it is presented in Fig. 5.

VI. Results and Discussion

Since, the original problem (1)-(4) was described as the OCP, for which both MPC and IC are suboptimal solutions, the comparison of both control methodologies is performed with respect to the OCP criterion (1). Computational demands, which are essential for real-time applications, serve as an additional indicator.

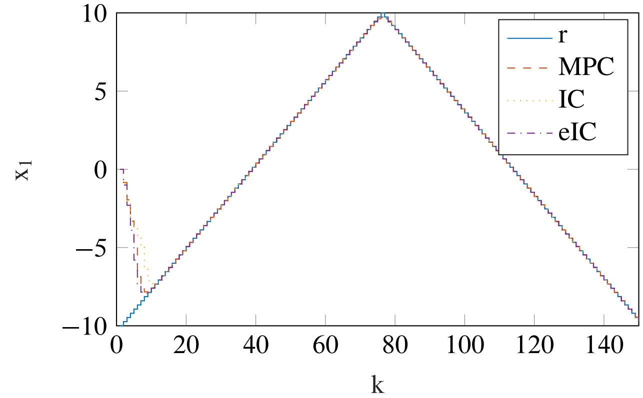

Fig. 5. The first state of the controlled system tracking the triangular reference signal

TABLE I
Evaluation and comparison of criterion for MPC, IC, and eIC

reference	measure	MPC	IC	eIC
sine	J	$3.13 \cdot 10^2$	$3.13 \cdot 10^2$	$3.42 \cdot 10^2$
	%	-	+1%	+9%
step	J	$1.18 \cdot 10^2$	$1.37 \cdot 10^2$	$1.50 \cdot 10^2$
	%	-	+16%	+27%
triangle	J	$2.91 \cdot 10^2$	$3.13 \cdot 10^2$	$3.47 \cdot 10^2$
	%	-	+8%	+19%

The results from every solver were identical, therefore, results acquired using GUROBI solver will be presented and compared by the quality of control. The computational time will be the measure for comparing the solvers. In Table I, the comparison of OCP criterion values for each controller in every simulation with different reference trajectories are denoted. The results imply, that the MPC has better performance than the IC. Important to say, given the length of the control horizon, the MPC should deliver the same performance as the strategy given by OCP. It is observable that the IC achieves better results in cases the reference trajectory is smooth without large changes. The results imply that the eIC has not delivered expected improvement of performance compared to the IC (the criterion value is $\sim 10\%$ higher) and thus, it is not recommended for employment in the trajectory tracking problem.

The computational time for every reference signal was similar, hence, only the case with sine wave reference signal will be denoted. The computational time demands are compared in Table II. The calculation of invariant sets, LQRs, and initialization in YALMIP are not included in the results. In Table II, the total time of computing in [s], longest period in [ms] and percentage reduction compared with MPC is denoted. The results imply, that the most suitable solver for investigated example is GUROBI, which performed better with solving QP for MPC and even LP for IC and eIC. Regardless of the type of solver, interpolation controllers have achieved significant computational time savings.

TABLE II

THE TIME DEMANDS FOR MPC, IC AND eIC FOR THE TRACKING OF SINE WAVE REFERENCE TRAJECTORY

SEDUMI	t [s]	%	$t_{\max}$ [ms]	%
MPC	854	-	102	-
IC	110	-87%	17	-83%
eIC	105	-88%	16	-84%
GUROBI	t [s]	%	$t_{\max}$ [ms]	%
MPC	126	-	17	-
IC	31	-75%	5	-71%
eIC	104	-18%	15	-6%
CPLEX	t [s]	%	$t_{\max}$ [ms]	%
MPC	218	-	30	-
IC	84	-62%	15	-50%
eIC	104	-52%	18	-40%

In case $x \in \mathbf{\Omega}^o$, it is not necessary to solve the LP for IC and eIC. The LQR for $\mathbf{\Omega}^o$ can be used directly. This setup was also investigated and it resulted in another dramatic decrease in computational time ($\sim 10 - 20\%$), however, the longest period remained the same.

VII. CONCLUSION

The paper dealt with the modification of IC to the trajectory tracking problem with constraints. First, the general constrained trajectory-tracking OCP for the discrete LTI system with linear constraints and quadratic criterion was presented. Further, the MPC, which is the state-of-the-art methodology for trajectory tracking, was described. Afterward, the IC was presented as the computationally efficient alternative to MPC and its adjustment for the trajectory tracking was discussed. Finally, IC and MPC were compared for several reference trajectories considering the quality of control and demands for computational time using several well-known solvers.

The results imply that the MPC delivers better control quality. The IC performance was slightly worse than MPC. However, considering the length of the MPC control horizon, the MPC should deliver the same performance as the strategy given by OCP. Moreover, the IC is incomparably more efficient in terms of computational time regardless of the selected solver. Considering the simplicity of IC with much lower computational demands it can be used as a decent alternative to the MPC especially with low-computational power systems. On the contrary, the eIC performed worse than IC and therefore, it is not suitable for trajectory tracking.

VIII. APPENDIX

In this appendix, the algorithms for computation of the set that is employed in the IC will be described. Algorithm 1 is based on the procedure in [7] and it computes the set $\mathbf{\Omega}^o$ that denotes the space where the LQR can be used without violation of constraints.

Algorithm 1 Computation of positive invariant set $\mathbf{\Omega}^o$

Input: Matrix $\boldsymbol{A}_c$ and $\boldsymbol{K}^o$, sets $\mathcal{X}$ and $\mathcal{U}$.
Output: positive invariant set $\mathbf{\Omega}^o$.

$\boldsymbol{F}^o = \begin{bmatrix} \boldsymbol{F}_x \\ \boldsymbol{F}_u \boldsymbol{K}^o \end{bmatrix}$, $\boldsymbol{g}^o = \begin{bmatrix} \boldsymbol{g}_x \\ \boldsymbol{g}_u \end{bmatrix}$
$\mathcal{X}^o = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{F}^o \boldsymbol{x} \leq \boldsymbol{g}^o\}$
loop
    $\mathcal{P} = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \begin{bmatrix} \boldsymbol{F}^o \\ \boldsymbol{F}^o \boldsymbol{A}_{c_k} \end{bmatrix} \boldsymbol{x} \leq \begin{bmatrix} \boldsymbol{g}^o \\ \boldsymbol{g}^o \end{bmatrix} \right\}$
    $\mathcal{P} =$ MINIMALREPRESENTATION($\mathcal{P}$) ▷ Erase redundant inequalities
    if $\mathcal{P} == \mathcal{X}^o$ **then**
        break
    end if
    $\mathcal{X}^o = \mathcal{P}$
end loop
$\mathbf{\Omega}^o = \mathcal{X}^o$

REFERENCES

[1] B. Anderson and J. Moore, *Optimal Control: Linear Quadratic Methods*, ser. Dover Books on Engineering. Dover Publications, 2007.

[2] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[3] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System," in *Studies in Computational Intelligence*, ser. Studies in Computational Intelligence, A. Koubaa, Ed. Cham: Springer International Publishing, 2017, no. January, pp. 3–39.

[4] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[5] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/~mpt.

[6] M. Klaučo and M. Kvasnica, *MPC-Based Reference Governors*, ser. Advances in Industrial Control. Cham: Springer International Publishing, 2019.

[7] H. Nguyen, *Constrained Control of Uncertain, Time-Varying, Discrete-Time Systems: An Interpolation-Based Approach*, ser. Lecture Notes in Control and Information Sciences. Springer International Publishing, 2014.

[8] H. N. Nguyen, P. O. Gutman, S. Olaru, and M. Hovd, "Explicit constraint control based on interpolation techniques for time-varying and uncertain linear discrete-time systems," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 44, no. PART 1, pp. 5741–5746, 2011.

[9] Z. Bouček and M. Flídr, "Explicit Interpolating Control of Unmaned Aerial Vehicle," in *Proc. of the 24th International Conference on Methods and Models in Automation and Robotics*, Międzyzdroje, Poland, August 26–29 2019.

[10] Z. Bouček and M. Flídr, "Modification of Explicit Interpolating Controller for Control Problem with Constant Setpoint," in *Proc. of the 15th European Workshop on Advanced Control and Diagnosis, ACD 2019*, Bologna, Italy, November 21–22 2019.

[11] M. Kamel, M. Burri, and R. Siegwart, "Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463–3469, 2017.

[12] T. Baca, P. Stepan, V. Spurny, D. Hert, R. Penicka, M. Saska, J. Thomas, G. Loianno, and V. Kumar, "Autonomous landing on a moving vehicle with an unmanned aerial vehicle," *Journal of Field Robotics*, no. January, 2019.

[13] J. Mattingley and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, mar 2012.

[14] J. Lofberg, "Yalmip : a toolbox for modeling and optimization in matlab," in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, pp. 284–289.