

This document contains a post-print version of the paper

## Real-time Nonlinear Model Predictive Path-Following Control of a Laboratory Tower Crane

authored by **Martin Böck and Andreas Kugi**

and published in *IEEE Transactions on Control Systems Technology*.

---

The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing. Please, scroll down for the article.

---

### Cite this article as:

M. Böck and A. Kugi, "Real-time nonlinear model predictive path-following control of a laboratory tower crane", *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1461–1473, 2014. DOI: [10.1109/TCST.2013.2280464](https://doi.org/10.1109/TCST.2013.2280464)

---

### BibTex entry:

```
@Article{acinpaper,  
  Title = {Real-time Nonlinear Model Predictive Path-Following Control of a Laboratory Tower Crane},  
  Author = {Martin B{"o"}ck and Andreas Kugi},  
  Journal = {{IEEE} Transactions on Control Systems Technology},  
  Year = {2014},  
  Number = {4},  
  Pages = {1461--1473},  
  Volume = {22},  
  Doi = {10.1109/TCST.2013.2280464}  
}
```

---

### Link to original paper:

<http://dx.doi.org/10.1109/TCST.2013.2280464>

---

### Read more ACIN papers or get this document:

<http://www.acin.tuwien.ac.at/literature>

---

### Contact:

Automation and Control Institute (ACIN)  
Vienna University of Technology  
Gusshausstrasse 27-29/E376  
1040 Vienna, Austria

Internet: [www.acin.tuwien.ac.at](http://www.acin.tuwien.ac.at)  
E-mail: [office@acin.tuwien.ac.at](mailto:office@acin.tuwien.ac.at)  
Phone: +43 1 58801 37601  
Fax: +43 1 58801 37699

---

### Copyright notice:

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Real-time Nonlinear Model Predictive Path-Following Control of a Laboratory Tower Crane

Martin Böck, Andreas Kugi, *Member, IEEE*

**Abstract**—A path-following controller is developed and applied to a laboratory tower crane. The control task is to move a load along a predefined geometric path. The time evolution along the path is not fixed but left as a degree of freedom to be determined by the controller. In order to be able to account for system constraints, a model predictive control scheme is adopted with special focus on real-time feasibility with small sampling times. The resulting controller is applied to a laboratory-scale tower crane and validated by means of simulation studies and measurement results.

**Index Terms**—Model predictive control, optimal control, path-following, real-time control, state constraint, tower crane.

## I. INTRODUCTION

THE goal of tracking control is to ensure that the output of a dynamical system follows (at least asymptotically) a given reference. If the reference signal is a predefined function of time, this is usually referred to as *trajectory* tracking control. On the contrary, if the time evolution along the reference signal is left as a degree of freedom for the controller, the resulting control scheme is called *path*-following control, see, e.g., [1].

Model predictive control (MPC), sometimes also named receding horizon control, relies on solving an optimal control problem (OCP) at each sampling instant and applying the first part of the optimal control input to the system. Subsequently, the optimization horizon is shifted forwards and the OCP is solved again with the actual (measured) system states as initial conditions, cf. [2]. MPC has the advantages of being able to deal with multiple input multiple output systems, to systematically account for constraints, and to achieve an optimal (in an appropriate sense) closed-loop behaviour. Current research is amongst others devoted to the application of MPC to complex nonlinear dynamical systems and to systematically derive stability and convergence conditions. However, usually the computational demand of the solution of the underlying OCP is rather high. Therefore, real-time capable formulations of MPC have been developed, which are able to deal with the control of complex dynamical systems subject to small sampling times. Very often the idea is to refrain from computing the optimal solution of the underlying OCP at each sampling instant. Rather a suboptimal but feasible solution is searched

for, which is improved from one sampling instant to the next, cf. [3]–[8].

Suboptimal MPC schemes can be classified with respect to the numerical computation of the solution estimates of the underlying OCP. Numerical methods for solving OCPs are frequently subdivided into direct and indirect methods [9]. Suboptimal MPC schemes belonging to the first kind, i.e., which rely on a suitable parameterization of the state and/or input variables to obtain a finite-dimensional nonlinear programming problem, are investigated, e.g., in [10]–[12]. Indirect methods aim at solving the necessary conditions of Pontryagin's maximum principle (cf. [13]). Suboptimal MPC methods adopting the latter principle are presented, e.g., in [14], [15].

Several approaches to path-following control exist in the literature. From a geometric point of view, the problem can be tackled by transforming the system into new coordinates of a transverse normal form and designing controllers for the dynamics transverse and tangential to the zero path error manifold, cf. [16], [17] and references given therein. Other approaches investigate Lyapunov and backstepping techniques [18], [19] and hybrid control strategies [20].

Recent works [21]–[25] study the combination of MPC with path-following. The benefit of this approach is the possibility to systematically account for input and state constraints. Based on different MPC formulations, with an explicit terminal region constraint in [22], without a terminal region constraint but utilizing a certain controllability criterion in [24], or based on contractive MPC in [21], the stability of the proposed predictive path-following controllers is proven. However, very often the problem of an efficient real-time implementation to cope with very small sampling times is not explicitly addressed. To overcome this issue, the authors in [21] propose a linear time-varying formulation based on the discrete-time system equations. This yields a convex quadratic program which can be solved efficiently at each sampling instant. The application of the resulting controller to an X-Y table is presented in [26].

The contribution of this paper is to combine a suboptimal MPC scheme based on indirect methods with path-following. The proposed framework is motivated by the application to a laboratory experiment of a tower crane which is shown in Fig. 1. The goal is to achieve a fast model predictive path-following control under consideration of input constraints of the system. This paper focuses on the derivation of the proposed control framework and its implementation on a real

Manuscript received November 6, 2012; revised July 19, 2013; accepted August 22, 2013.

M. Böck and A. Kugi are with the Automation and Control Institute, Vienna University of Technology, 1040 Vienna, Austria (e-mail: {boeck, kugi}@acin.tuwien.ac.at).



Fig. 1. Tower crane system.

hardware for obtaining measurement results whereas stability and convergence are not rigorously investigated. The tower crane is an underactuated mechanical system which exhibits a nonlinear dynamics. It can be represented by a mathematical model with a state space dimension of 10 and three control inputs. The controller operates with a sampling time of 2 ms. These facts demonstrate the performance of the proposed path-following MPC. Nevertheless, the presented control framework is also applicable to predictive path-following for other dynamical systems.

Compared to the approach described in [21], [26] the proposed method does not rely on a linearization of the system equations, constraints or terms related to the cost functional. Furthermore, in contrast to this paper the method in [21], [26] utilizes discretization techniques to obtain a quadratic program at each sampling instant, i.e. it belongs to the class of direct methods. The quadratic program is solved to optimality at each sampling instant by means of a static optimization solver.

This work is organized as follows. Section II introduces the considered tower crane system and the corresponding mathematical model. The objective of path-following control together with a suitable parameterization of the path and the MPC approach are presented in Section III. The real-time implementation is devised in Section IV. In Section V, the proposed control framework is evaluated by means of simulation and experimental results. Conclusions are drawn in Section VI.

### Notation

In the context of an OCP, the optimal quantities are indicated with the superscript  $*$ . The total derivatives of a function  $x(t)$  with respect to time are denoted by  $\dot{x}$ ,  $\ddot{x}$ ,  $x^{(3)}$ , and so forth. Given a vector  $y \in \mathbb{R}^n$ ,  $\left(y\right)_Q$  represents the quadratic form  $y^T Q y$  with the positive (semi-)definite matrix  $Q \in \mathbb{R}^{n \times n}$ . The index  $i$  refers to the  $i$ th component of the respective quantity. A diagonal matrix  $D$  with  $D_{i,i} = y_i$  is denoted as  $\text{diag}(y)$ . The orthogonal basis vectors of the Cartesian coordinate system are denoted as  $e_x$ ,  $e_y$ , and  $e_z$ .

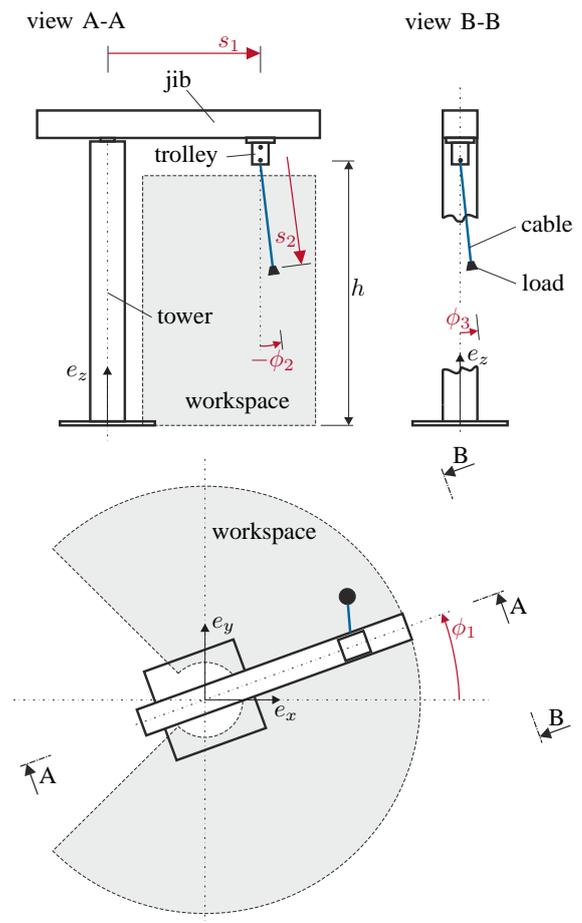


Fig. 2. Overview of the tower crane system.

## II. TOWER CRANE SYSTEM

A tower crane is a representative benchmark for path-following control with real-world applicability. The considered laboratory experiment has already been investigated in different contexts in [27] and [28]. For the sake of self-containedness and clearness of this presentation, the description of the system together with the overall control concept, the corresponding mathematical model, and the experimental setup including sensors and actuators are shortly recapitulated.

Two mathematical models are derived. Firstly, a comprehensive mathematical model is assembled in Section II-B which is used for simulation purposes. Secondly, a specific control structure is proposed in Section II-C which allows to set up a compact model serving as basis for the path-following controller.

### A. Description of the Tower Crane Experimental Setup

Figure 2 gives an overview of the considered laboratory experiment. The tower crane has five degrees of freedom (DOF), namely the position  $s_1$  of the trolley along the jib, the length  $s_2$  of the cable from the trolley to the load, the angular displacement  $\phi_1$  of the jib, and the angular displacements  $\phi_2$

and  $\phi_3$  of the cable with respect to the trolley. The cable is guided on the trolley by means of a sleeve mounted on a gimbal, which allows the cable to sway in any direction. The gimbal has a vertical distance of  $h = 0.92$  m to the ground of the workspace. The laboratory tower crane is an underactuated mechanical system. Three DOF are actuated ( $s_1$ ,  $s_2$ , and  $\phi_1$ ) by means of DC motors with armature voltages  $v_i$ ,  $i = 1, 2, 3$ , and currents  $I_i$ ,  $i = 1, 2, 3$ , while  $\phi_2$  and  $\phi_3$  are unactuated.

All five DOF  $s_1$ ,  $s_2$ ,  $\phi_1$ ,  $\phi_2$ , and  $\phi_3$  are directly measurable by means of incremental encoders. The corresponding velocities  $\dot{s}_1$ ,  $\dot{s}_2$ ,  $\dot{\phi}_1$ ,  $\dot{\phi}_2$ , and  $\dot{\phi}_3$  are obtained by approximate differentiation. The angular displacements  $\phi_2$  and  $\phi_3$  of the cable are measured by the angular displacements of the sleeve. However, to reduce friction, the guidance of the cable in the sleeve is not completely tight. This loose guidance inherently introduces measurement errors, which will become apparent in Section V.

### B. Comprehensive Mathematical Model

The mathematical model of the laboratory-scale tower crane is calculated based on rigorous physical considerations by means of the Euler-Lagrange equations of motion, see, e.g., [29]. To this end, it is assumed that the cable and the load can be modelled as a mathematical spherical pendulum. This particularly entails that the friction torques in the gimbal (corresponding to the DOF  $\phi_2$  and  $\phi_3$ ) are neglected which is justified as the pendulum motion of the load is nearly undamped. The laboratory experiment suffers from considerable friction effects in the actuated axes, which, for the sake of simplicity, are modelled as viscous friction.

Each of the DC motors at the laboratory tower crane is equipped with a current controller which sets the corresponding armature voltage, cf. Fig. 3. The dynamics of these subordinate current control loops are much faster than the typical mechanical time constants of the crane. Due to this fact it is justified, by arguing along the lines of the singular perturbation theory [30], to utilize the actual currents of the DC motors as inputs to the system.

Based on these assumptions the resulting mathematical model of the tower crane partitioned into the actuated and unactuated part reads as

$$\underbrace{\begin{bmatrix} D_A(q) & D_{AU}(q) \\ D_{AU}^T(q) & D_U(q) \end{bmatrix}}_{D(q)} \ddot{q} + \underbrace{\begin{bmatrix} C_A(q, \dot{q}) & C_{AU}(q, \dot{q}) \\ C_{UA}(q, \dot{q}) & C_U(q, \dot{q}) \end{bmatrix}}_{C(q, \dot{q})} \dot{q} + \begin{bmatrix} g_A(q) \\ g_U(q) \end{bmatrix} + \begin{bmatrix} \tau_F(\dot{q}) \\ 0 \end{bmatrix} = \begin{bmatrix} B_\tau \\ 0 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} \quad (1)$$

with the generalized coordinates  $q_A = [s_1 \ s_2 \ \phi_1]^T$ ,  $q_U = [\phi_2 \ \phi_3]^T$ , and  $q = [q_A^T \ q_U^T]^T$ . The equations of motion (1) consist of the gravitational and friction terms

$$g_A(q) = [0 \ -m_L g \cos(\phi_2) \cos(\phi_3) \ 0]^T \quad (2a)$$

$$g_U(q) = \begin{bmatrix} m_L g \sin(\phi_2) \cos(\phi_3) s_2 \\ m_L g \cos(\phi_2) \sin(\phi_3) s_2 \end{bmatrix} \quad (2b)$$

$$\tau_F(\dot{q}) = [d_1 \dot{s}_1 \ d_2 \dot{s}_2 \ d_3 \dot{\phi}_1]^T \quad (2c)$$

and the actuator input matrix

$$B_\tau = \text{diag} \left( \begin{bmatrix} \frac{R_1 k_1}{r_1} & \frac{R_2 k_2}{r_2} & R_3 k_3 \end{bmatrix} \right) \quad (2d)$$

with  $m_L, g, d_i, R_i, k_i$  denoting the load mass, gravitational acceleration, viscous friction coefficients, transmission ratios, and torque constants,  $i = 1, 2, 3$ , respectively. The inertia matrix  $D(q)$  and the vector of centripetal and Coriolis terms  $C(q, \dot{q}) \dot{q}$  are omitted for brevity. The trolley is driven by means of a toothed belt with  $r_1$  denoting the radius of the pulley. The constant radius of the cable drum is denoted as  $r_2$ .

### C. Control Concept and Mathematical Model

In principle, the comprehensive mathematical model (1) of the laboratory tower crane with the idealized current controllers could be used as the basis for a path-following controller. However, to compensate for the friction in the actuated DOF and to be independent of the actual value of the load mass it is advantageous to additionally use velocity controllers for each of the actuated DOF in another level of the cascaded control structure. It turns out in practice that the velocity controllers perform well and the closed-loop dynamics can be tuned considerably fast, cf. Section V-B. This reasoning allows to set up an overall mathematical model of all components where the dynamics of the velocity control loops is neglected. Thereby, the accelerations  $\ddot{s}_1$ ,  $\ddot{s}_2$ , and  $\ddot{\phi}_1$  of the three actuated DOF are utilized as new control inputs  $u = [u_1 \ u_2 \ u_3]^T$ . Thus, in consideration of (1), the overall mathematical model reads as

$$\ddot{s}_1 = u_1 \quad (3a)$$

$$\ddot{s}_2 = u_2 \quad (3b)$$

$$\ddot{\phi}_1 = u_3 \quad (3c)$$

$$\ddot{q}_U = -D_U^{-1}(q) (D_{AU}^T(q) u + C_{UA}(q, \dot{q}) \dot{q}_A + C_U(q, \dot{q}) \dot{q}_U + g_U(q)) \quad (3d)$$

in which  $\ddot{q}_U = [\ddot{\phi}_2 \ \ddot{\phi}_3]^T$  are explicitly given by

$$\ddot{\phi}_2 = \frac{1}{s_2 \cos(\phi_3)} \left( \cos(\phi_2) u_1 - s_2 \cos(\phi_2) \sin(\phi_3) u_3 + 2s_2 \dot{\phi}_2 \dot{\phi}_3 \sin(\phi_3) + s_2 \dot{\phi}_1^2 \cos(\phi_2) \cos(\phi_3) \sin(\phi_2) - 2\dot{s}_2 \dot{\phi}_2 \cos(\phi_3) - g \sin(\phi_2) - 2\dot{s}_2 \dot{\phi}_1 \cos(\phi_2) \sin(\phi_3) - 2s_2 \dot{\phi}_1 \dot{\phi}_3 \cos(\phi_2) \cos(\phi_3) - s_1 \dot{\phi}_1^2 \cos(\phi_2) \right) \quad (3e)$$

$$\ddot{\phi}_3 = -\frac{1}{s_2} \left( \sin(\phi_2) \sin(\phi_3) u_1 + s_1 \cos(\phi_3) u_3 - s_2 \sin(\phi_2) u_3 + 2\dot{s}_1 \dot{\phi}_1 \cos(\phi_3) - 2\dot{s}_2 \dot{\phi}_1 \sin(\phi_2) + 2\dot{s}_2 \dot{\phi}_3 - s_2 \dot{\phi}_1^2 \cos(\phi_3) (\cos(\phi_2))^2 \sin(\phi_3) - s_1 \dot{\phi}_1^2 \sin(\phi_2) \sin(\phi_3) - 2s_2 \dot{\phi}_1 \dot{\phi}_2 \cos(\phi_2) (\cos(\phi_3))^2 + s_2 \dot{\phi}_2^2 \cos(\phi_3) \sin(\phi_3) + g \cos(\phi_2) \sin(\phi_3) \right). \quad (3f)$$

The model (3) solely contains explicit differential equations and its complexity is much lower than that of (1) which permits fast online calculations. It can be rewritten as a first

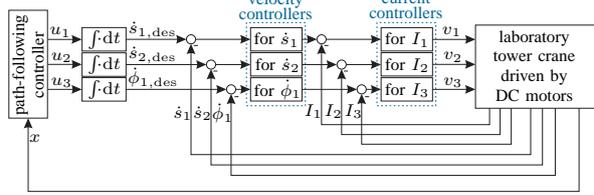


Fig. 3. Overall control structure.

order system of differential equations in state space form completed with an output  $y \in \mathbb{R}^m$

$$\dot{x} = f(x, u) \quad (4a)$$

$$y = h(x) \quad (4b)$$

with  $x = [q^T \ \dot{q}^T]^T$ . The set of feasible inputs is given by

$$U = \{u \in \mathbb{R}^3 \mid -2 \text{ m s}^{-2} \leq u_1 \leq 2 \text{ m s}^{-2}, \\ -2 \text{ m s}^{-2} \leq u_2 \leq 2 \text{ m s}^{-2}, \\ -2 \text{ rad s}^{-2} \leq u_3 \leq 2 \text{ rad s}^{-2}\}. \quad (5)$$

Figure 3 shows the proposed control structure with subordinate current and velocity controllers which is implemented in this way at the laboratory experiment. This structure together with the simplifications stated so far yields the compact mathematical model (4) which is not only able to accurately describe the dynamic behaviour of the laboratory tower crane together with the current and velocity controllers but is also computationally very efficient. The model (4), which is independent of friction effects and the actual value of the load mass, serves as a basis for the path-following controller. In the following, an MPC is adopted as path-following controller. The reference inputs  $\dot{s}_{1,\text{des}}$ ,  $\dot{s}_{2,\text{des}}$ , and  $\dot{\phi}_{1,\text{des}}$  for the velocity controllers are generated by simply integrating the control inputs  $u_i$ ,  $i = 1, 2, 3$ .

### III. PATH-FOLLOWING CONTROL

This section aims at precisely specifying the problem of path-following control. The required mathematical preliminaries are developed and the control approach using MPC is formulated. For the sake of conciseness, most of the results in this section are given for the considered tower crane system.

#### A. Problem Statement

In the following, a map

$$\theta \rightarrow p(\theta) \in \mathbb{R}^q, \theta \in [\theta_0, \theta_1] \subset \mathbb{R} \quad (6)$$

is considered, which defines a geometric path

$$\mathcal{P} = \{\bar{p} \in \mathbb{R}^q \mid \bar{p} = p(\theta), \theta \in [\theta_0, \theta_1]\} \quad (7)$$

in a vector space  $\mathbb{R}^q$ .  $\theta$  is the so-called path parameter which is supposed to take values in the interval  $[\theta_0, \theta_1] \subset \mathbb{R}$ . It is assumed that the start and the end point of the path correspond to  $\theta = \theta_0$  and  $\theta = \theta_1$ , respectively. Path-following control for a dynamical system aims at tracking a geometric path as given in (7). The difference to trajectory tracking lies in the fact that

the time evolution of the path parameter is not predetermined but left as a degree of freedom for the controller. Intuitively the path (7) can be defined in the output space of the dynamical system. The purpose of the path-following controller is to determine inputs such that the output  $y = h(x)$  of the system (4) follows the path (7) with  $q = m$ . This is usually referred to as *output path-following*, cf. [23]. To simplify the presentation, we restrict our considerations to output paths defined in a flat output space of (4), see [31] for details on differential flatness. In contrast to the general case, this allows to choose the components (assuming  $m \geq 2$ ) of  $p(\theta(t))$  independently.

Alternatively the geometric reference to be followed can be defined in the state space  $x \in \mathbb{R}^n$ . In general this entails that the components of the geometric reference cannot be chosen independently from each other. Furthermore, apart from special cases (see, e.g., [22]) it is not possible to state a curve in the state space defined by

$$\theta \rightarrow \tilde{p}_x(\theta) \in \mathbb{R}^n, \theta \in [\theta_0, \theta_1] \subset \mathbb{R} \quad (8)$$

which the system is supposed to follow. Instead, one needs to consider a manifold in the state space which is subsequently referred to as the *zero path error manifold* [32]. In certain cases it is possible to deduce an explicit parameterization of the zero path error manifold in the form

$$p_x(\theta, \dot{\theta}, \ddot{\theta}, \dots) \subset \mathbb{R}^n. \quad (9)$$

One natural approach is to derive (9) from a geometric path defined in the output space of the system. In particular, based on an output path defined in a flat output space of (4) the parameterization (9) can be systematically calculated. This fact is also utilized for the considered tower crane system. Therefore, the subsequent considerations rely on (9).

For the tower crane, the goal of path-following control is that the position of the load follows a predefined geometric path. From this point of view it would be natural to formulate the control task as an output path-following problem. However, we will reformulate the problem as a path-following problem in the state space. There are two main reasons for this: First, for the design of the controller, it is desirable to use as much information about the system as possible. As will be described in Section III-B, the load position of the tower crane constitutes a flat output of the system. This has the advantage that all system variables (states and inputs) can be algebraically parameterized by means of the flat output and its time derivatives, cf. [31]. The second reason is that existing stability results on suboptimal MPC usually require to penalize the whole state in the cost functional, see, e.g., [8].

Based on the preceding reasoning, the path-following problem for the tower crane can be formulated as follows. Consider the zero path error manifold (9) in the state space with  $\theta \in [\theta_0, \theta_1]$  and the path-following error

$$e(t) = x(t) - p_x(\theta(t), \dot{\theta}(t), \ddot{\theta}(t), \dots). \quad (10)$$

Suppose that the path-following problem starts at the initial time  $t_0$ . The goals of the path-following controller are to achieve

- (A) a time evolution of the path parameter  $\theta$  such that  $\theta(t) \in [\theta_0, \theta_1] \forall t \in [t_0, \infty)$  and  $\lim_{t \rightarrow \infty} \theta(t) = \theta_1$ ,
- (B) asymptotic path convergence  $\lim_{t \rightarrow \infty} \|e(t)\| = 0$ , and
- (C) satisfaction of the system constraints, i.e.  $u(t) \in U \forall t \in [t_0, \infty)$ .

One remark concerning the first goal (A) is necessary. Some authors additionally require that the path parameter satisfies  $\dot{\theta}(t) > 0 \forall \theta(t) \in [\theta_0, \theta_1]$  [19], [22] or  $\dot{\theta}(t) \geq 0 \forall t \in [t_0, \infty)$  [24] which corresponds to strict monotonous and monotonous forward motion along the path, respectively. These requirements are not used here. Hence, the MPC formulation has to ensure an adequate forward motion towards the end of the path, e.g., by appropriately choosing the cost functional of the underlying OCP.

In principle it is possible to not only account for input but also state constraints of the system, i.e.  $x(t) \in X$  with  $X$  describing the set of feasible states. Although this can be done with the same methods as presented in the following, it does not lie within the scope of this paper.

### B. Parameterization of the Zero Path Error Manifold

Based on the preceding problem specification the first task is to find the parameterization of the zero path error manifold (9). As outlined, this is in general a non-trivial task. However, based on the fact that the load position in Cartesian coordinates, given by

$$y_L = \begin{bmatrix} -s_2 (\sin(\phi_1) \sin(\phi_3) + \cos(\phi_1) \sin(\phi_2) \cos(\phi_3)) \\ s_2 (\cos(\phi_1) \sin(\phi_3) - \sin(\phi_1) \sin(\phi_2) \cos(\phi_3)) \\ h - s_2 \cos(\phi_2) \cos(\phi_3) \end{bmatrix} + \begin{bmatrix} s_1 \cos(\phi_1) \\ s_1 \sin(\phi_1) \\ 0 \end{bmatrix}, \quad (11)$$

constitutes a flat output of the system (cf. [28]), this problem is greatly simplified. The flatness-based parameterization of the state and the input reads as

$$x = \psi_x \left( y_L, \dot{y}_L, \ddot{y}_L, y_L^{(3)} \right) \quad (12a)$$

$$u = \psi_u \left( y_L, \dot{y}_L, \ddot{y}_L, y_L^{(3)}, y_L^{(4)} \right), \quad (12b)$$

which can be utilized to obtain the desired parameterization  $p_x(\theta, \dot{\theta}, \ddot{\theta}, \dots)$  from the output path defined by

$$\theta \rightarrow p_{y_L}(\theta) \in \mathbb{R}^3, \theta \in [\theta_0, \theta_1]. \quad (13)$$

To this end, the map (13) is successively differentiated with respect to time yielding (cf. [33])

$$\frac{d}{dt} (p_{y_L}(\theta)) = \frac{\partial p_{y_L}}{\partial \theta}(\theta) \dot{\theta} \quad (14a)$$

$$\frac{d^2}{dt^2} (p_{y_L}(\theta)) = \frac{\partial^2 p_{y_L}}{\partial \theta^2}(\theta) \dot{\theta}^2 + \frac{\partial p_{y_L}}{\partial \theta}(\theta) \ddot{\theta} \quad (14b)$$

$$\frac{d^3}{dt^3} (p_{y_L}(\theta)) = \varphi_1(\theta, \dot{\theta}, \ddot{\theta}, \theta^{(3)}) \quad (14c)$$

$$\frac{d^4}{dt^4} (p_{y_L}(\theta)) = \varphi_2(\theta, \dot{\theta}, \ddot{\theta}, \theta^{(3)}, \theta^{(4)}). \quad (14d)$$

By inserting the expressions (13) and (14) into (12a) and combining the path parameter together with its derivatives

to  $\zeta = [\theta \ \dot{\theta} \ \ddot{\theta} \ \theta^{(3)} \ \theta^{(4)}]^T$ , the parameterized zero path error manifold

$$\zeta \rightarrow p_x(\zeta), \zeta_1 \in [\theta_0, \theta_1] \quad (15)$$

is obtained, see also [34]. Furthermore, the same procedure and equation (12b) yield the system inputs

$$\zeta \rightarrow p_u(\zeta), \zeta_1 \in [\theta_0, \theta_1], \quad (16)$$

which in the nominal, undisturbed case with consistent initial conditions guarantee exact path-following, provided that the constraints (5) are not violated. The expression  $p_u(\zeta)$  will not be utilized for the proposed path-following MPC but it is useful for comparison purposes. Therefore,  $\theta^{(4)}$  is also contained in  $\zeta$  to allow the calculation of  $p_x(\zeta)$  and  $p_u(\zeta)$  from a given trajectory  $\zeta(t)$ .

In principle the presented control scheme can also be applied to other dynamical systems. Note that a major step in the calculations is to find the parameterization of the zero path error manifold (15). As shown this task can be carried out in a straightforward way if  $p_x(\zeta)$  is derived from an output path defined in a flat output space. In the non-flat case, this might be very difficult.

### C. Model Predictive Control Approach

To accomplish the goals for path-following formulated in Section III-A, an MPC approach is utilized. This requires the formulation of an appropriate OCP based on the current system state  $x_k$ , which is solved at every sampling instant  $t_k = kT_s$ , with the sampling time  $T_s$ . As mentioned before, the MPC also has to take care of a suitable time evolution  $\theta(t)$  along the path. To this end, a linear, time-invariant auxiliary system (timing law) is defined as (cf. [24])

$$\dot{\zeta} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_A \zeta + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_B w, \quad (17)$$

with a new unconstrained virtual control input  $w$ . With these ingredients, the OCP of the MPC at time instant  $t_k$  reads as

$$\min_{\bar{v}(\cdot)} J(t_k, \bar{v}(\cdot)) \quad (18a)$$

subject to

$$\dot{\bar{x}} = f(\bar{x}, \bar{u}), \quad \bar{x}(t_k) = x_k \quad (18b)$$

$$\dot{\bar{\zeta}} = A\bar{\zeta} + B\bar{w}, \quad \bar{\zeta}(t_k) = \bar{\zeta}_{k-1}^*(t_{k-1} + T_s) \quad (18c)$$

$$\bar{u}(t) \in U, \quad \forall t \in [t_k, t_k + T] \quad (18d)$$

$$\bar{\zeta}_1(t) \in [\theta_0, \theta_1], \quad \forall t \in [t_k, t_k + T] \quad (18e)$$

with  $\bar{v} = [\bar{u}^T \ \bar{w}^T]^T$  and the cost functional

$$J(t_k, \bar{v}(\cdot)) = \int_{t_k}^{t_k+T} l(\bar{x}, \bar{\zeta}, \bar{u}, \bar{w}) dt + V(\bar{x}(t_k + T), \bar{\zeta}(t_k + T)). \quad (19)$$

Here  $T$  is the prediction horizon and  $x_k$  the measured system state at sampling instant  $t_k$ . The initial condition for the auxiliary system is set to  $\bar{\zeta}_{k-1}^*(t_{k-1} + T_s)$ , which is the optimal solution found at the previous sampling instant  $t_{k-1}$  and evaluated at  $t_k = t_{k-1} + T_s$ . For  $t_0$ ,  $\bar{\zeta}(t_0) = [\theta_0 \ 0 \ 0 \ 0 \ 0]^T$  is used. The cost functions in (19) are chosen in the form

$$l(\bar{x}, \bar{\zeta}, \bar{u}, \bar{w}) = \frac{1}{2} \left[ \left( \bar{x} - p_x(\bar{\zeta}) \right)_{Q_e} + \left( \bar{\zeta} - \zeta_e \right)_{Q_\zeta} + \left( \bar{u} \right)_{R_u} + \left( \bar{w} \right)_{R_w} \right] \quad (20a)$$

and

$$V(\bar{x}, \bar{\zeta}) = \frac{1}{2} \left[ \left( \bar{x} - p_x(\bar{\zeta}) \right)_{S_e} + \left( \bar{\zeta} - \zeta_e \right)_{S_\zeta} \right], \quad (20b)$$

respectively. The bar refers to internal variables of the algorithm. Having in mind the goals of the path-following MPC, the cost functional weights the deviation of  $\bar{x}$  from the desired zero path error manifold and the deviation of  $\bar{\zeta}$  from the desired end point  $\zeta_e = [\theta_1 \ 0 \ 0 \ 0 \ 0]^T$ . The focus lies on the MPC formulation without explicit terminal constraints, which on the one hand improves the computational efficiency when solving (18) and on the other hand gives the possibility for the application of efficient suboptimal MPC schemes, cf. [8], [14].

Several authors investigate stability of MPC without explicit terminal constraints, see, e.g., [8], [35], [36]. However, an in-depth stability and convergence analysis for the proposed control approach does not lie within the scope of this paper.

Let  $\bar{v}(t, t_k)$  denote the control input to (18) at sampling instant  $t_k$  with  $t \in [t_k, t_k + T]$ . In the "ideal" MPC-case, (18) is solved at every sampling instant  $t_k$  yielding the global optimal solution  $\bar{v}^*(t, t_k)$  together with the optimal value of the cost functional  $J^*(t_k) := J(t_k, \bar{v}^*(t, t_k))$ . The optimal input  $\bar{v}^*(t, t_k)$  is applied to the system during the time interval  $[t_k, t_k + T_s]$ . At  $t_k + T_s$  the optimization horizon is shifted forwards, the system state is measured, and (18) is solved again. However, the requirement to calculate the optimal solution of (18) within the sampling time  $T_s = 2$  ms is computationally intractable. Therefore, a suboptimal strategy is employed which does not require the calculation of the optimal solution of (18) but basically relies on the improvement of the value of the cost functional from one sampling instant to the next, cf. [8].

#### IV. REAL-TIME IMPLEMENTATION

##### A. Fast Model Predictive Control

Fast MPC schemes usually relax the requirement of finding the optimal solution at each sampling instant. Instead, and particularly with regard to stability arguments, they rely on an improvement of the current value of the cost functional from one sampling instant to the next.

For solving the path-following MPC in a suboptimal and therefore real-time capable fashion, an MPC scheme based on indirect methods using the gradient projection method is utilized, subsequently referred to as GPMPC, see, e.g., [14]. It

has already been applied to the considered tower crane in [27] for set-point stabilization and in [28] for trajectory tracking. However, path-following MPC has not been investigated so far, which is especially challenging due to the increased number of states and inputs and the state constraint (18e). The reasons for choosing GPMPC are the memory and time efficient computations and the low implementation effort.

The gradient projection method aims at solving the necessary optimality conditions of Pontryagin's maximum principle (cf. [13]) for the OCP (18a)-(18d), i.e. without the state constraint (18e),

$$\dot{\bar{x}}^* = f(\bar{x}^*, \bar{u}^*), \quad \bar{x}^*(t_k) = x_k \quad (21a)$$

$$\dot{\bar{\zeta}}^* = A\bar{\zeta}^* + B\bar{w}^*, \quad \bar{\zeta}^*(t_k) = \bar{\zeta}_{k-1}^*(t_{k-1} + T_s) \quad (21b)$$

$$\dot{\lambda}_x^* = - \left( \frac{\partial H}{\partial \bar{x}} \right)^T (\bar{x}^*, \bar{\zeta}^*, \bar{v}^*, \lambda_x^*, \lambda_\zeta^*) \quad (21c)$$

$$\dot{\lambda}_\zeta^* = - \left( \frac{\partial H}{\partial \bar{\zeta}} \right)^T (\bar{x}^*, \bar{\zeta}^*, \bar{v}^*, \lambda_x^*, \lambda_\zeta^*) \quad (21d)$$

$$H(\bar{x}^*, \bar{\zeta}^*, \bar{v}^*, \lambda_x^*, \lambda_\zeta^*) \leq H(\bar{x}^*, \bar{\zeta}^*, \bar{v}, \lambda_x^*, \lambda_\zeta^*), \quad \forall t \in [t_k, t_k + T], \quad \forall \bar{v} \in \left\{ [\bar{u}^T \ \bar{w}^T] \mid \bar{u} \in U, \bar{w} \in \mathbb{R} \right\} \quad (21e)$$

$$\lambda_x^*(t_k + T) = \left( \frac{\partial V}{\partial \bar{x}} \right)^T (\bar{x}^*(t_k + T), \bar{\zeta}^*(t_k + T)) \quad (21f)$$

$$\lambda_\zeta^*(t_k + T) = \left( \frac{\partial V}{\partial \bar{\zeta}} \right)^T (\bar{x}^*(t_k + T), \bar{\zeta}^*(t_k + T)), \quad (21g)$$

where the Hamiltonian reads as  $H(\bar{x}, \bar{\zeta}, \bar{v}, \lambda_x, \lambda_\zeta) = l(\bar{x}, \bar{\zeta}, \bar{u}, \bar{w}) + \lambda_x^T f(\bar{x}, \bar{u}) + \lambda_\zeta^T (A\bar{\zeta} + B\bar{w})$ . The gradient projection method is especially well suited for OCPs without terminal constraints because in this case the adjoint states  $\lambda^T = [\lambda_x^T \ \lambda_\zeta^T]$  are fully determined by (21f) and (21g). Therefore, a simple backward integration of (21c) and (21d) is possible. Let  $\bar{v}_{(j)}(t, t_k)$  denote the input in iteration  $j$  of the gradient projection method. Starting with an initial guess  $\bar{v}_{(0)}(t, t_k)$ , the method can briefly be described as follows (for further details see [14]).

##### Algorithm 1 (Gradient Projection Method)

- 1) Integrate the system differential equations (21a), (21b) in forward direction by using  $\bar{v}_{(j)}(t, t_k)$  yielding  $\bar{x}_{(j)}$  and  $\bar{\zeta}_{(j)}$ .
- 2) Integrate the adjoint differential equations (21c), (21d) in backward direction by utilizing the terminal conditions (21f) and (21g) yielding  $\lambda_{x,(j)}$  and  $\lambda_{\zeta,(j)}$ .
- 3) Calculate the search direction as the negative gradient of the Hamiltonian  $d_{(j)}(t) = - \left( \frac{\partial H}{\partial \bar{v}} \right)^T (\bar{x}_{(j)}, \bar{\zeta}_{(j)}, \bar{v}_{(j)}(t, t_k), \lambda_{x,(j)}, \lambda_{\zeta,(j)})$ .
- 4) Solve the line search problem  $\alpha^* = \arg \min_{\alpha > 0} J(t_k, \Pi(\bar{v}_{(j)}(t, t_k) + \alpha d_{(j)}(t)))$ .
- 5) Calculate  $\bar{v}_{(j+1)}(t, t_k) = \Pi(\bar{v}_{(j)}(t, t_k) + \alpha^* d_{(j)}(t))$ .
- 6) Check if some termination condition is satisfied, if not set  $j \leftarrow j + 1$  and return to step 1).

$\Pi(\cdot)$  denotes the projection operator onto the feasible set  $\left\{ [u^T \ w]^T \mid u \in U, w \in \mathbb{R} \right\}$ . The line search in step 4) of the

algorithm is usually carried out approximately, for example by using polynomial interpolation. When using Algorithm 1 in the context of suboptimal GPMPC it is terminated after a fixed number of iterations  $N$ . Subsequently  $\bar{v}_{(N)}(t, t_k)$  is applied to the system during the time interval  $[t_k, t_k + T_s)$ . At the next sampling instant,  $\bar{v}_{(N)}(t, t_k)$  is used to construct an initial input  $\bar{v}_{(0)}(t, t_{k+1})$  (cf. [14]) and the gradient projection method again performs  $N$  iterations.

Algorithm 1 reveals the advantages of the gradient projection method which allows an efficient MPC implementation. In each MPC step,  $N$  numerical integrations of the system and adjoint differential equations have to be performed, which is computationally inexpensive and results in a predictable behaviour. Furthermore, a scalar optimization problem has to be solved for  $\alpha^*$ , which, however, can be done efficiently in an approximate fashion. Provided that a sufficiently small sampling time  $T_s$  is used, a small number of iterations (e.g.  $N = 2$ ) suffices to achieve satisfactory control results.

Considering the problem at hand, the main disadvantage of GPMPC in the presented formulation is that state constraints like (18e) cannot be handled in a straightforward way.

### B. Application to Path-Following MPC

To allow an application of the presented GPMPC to the OCP (18), the state constraint (18e) has to be implicitly taken into account and must not appear explicitly in the OCP. Here only the upper bound

$$\zeta_1 \leq \theta_1 \quad (22)$$

is considered whereas the lower bound  $\zeta_1 \geq \theta_0$  is supposed to be automatically fulfilled. This assumption is reasonable since  $\zeta_1(t_0) = \theta_0$  and the cost functional weights the deviation of  $\zeta_1$  from  $\theta_1$ . Therefore, by an appropriate choice of the weights for  $\zeta_1$ , the MPC will aim at steering  $\zeta_1$  to  $\theta_1$  with  $\zeta_2 \geq 0$ , i.e.  $\zeta_1$  increases, which means that  $\zeta_1 \geq \theta_0$  is not violated.

The state constraint (22) is considered by combining the ideas presented in [37] and [38]. The concept is to transform the states of the auxiliary system (17) into unconstrained coordinates via an invertible transformation. By introducing a slack variable  $\eta_1(t)$ , the inequality constraint (22) is formulated as an equality constraint

$$\zeta_1 - \theta_1 + \frac{1}{2}\eta_1^2 = 0, \quad (23)$$

which guarantees satisfaction of (22) for all  $\eta_1(t) \in \mathbb{R}$ . Successive differentiation of (23) and substitution of the system equations (17) until the input  $w$  appears explicitly yields

$$\zeta_2 + \eta_1\eta_2 = 0 \quad (24a)$$

$$\zeta_3 + \eta_2^2 + \eta_1\eta_3 = 0 \quad (24b)$$

$$\zeta_4 + 3\eta_2\eta_3 + \eta_1\eta_4 = 0 \quad (24c)$$

$$\zeta_5 + 3\eta_3^2 + 4\eta_2\eta_4 + \eta_1\eta_5 = 0 \quad (24d)$$

$$w + 10\eta_3\eta_4 + 5\eta_2\eta_5 + \eta_1w_t = 0 \quad (24e)$$

with the integrator chain  $\dot{\eta}_i = \eta_{i+1}$ ,  $i = 1, \dots, 4$  and the new control input  $w_t = \dot{\eta}_5$ . From (23) and (24a)-(24d), a nonlinear invertible coordinate transformation

$$\zeta = \gamma_1(\eta) \quad (25)$$

with  $\eta = [\eta_1 \ \eta_2 \ \eta_3 \ \eta_4 \ \eta_5]^T$  can be easily deduced. Notice that the inverse transformation  $\eta = \gamma_1^{-1}(\zeta)$  exists only on the subspace  $\{\zeta \in \mathbb{R}^5 | \zeta_1 \leq \theta_1\}$ . From (24e) the original input

$$w = \gamma_2(\eta, w_t) \quad (26)$$

can be calculated. Inserting (25) and (26) into the OCP (18) (apart from  $\zeta_1 \geq \theta_0$ ) yields the new OCP without state constraints

$$\min_{\bar{v}_t(\cdot)} J_t(t_k, \bar{v}_t(\cdot)) \quad (27a)$$

subject to

$$\dot{\bar{x}} = f(\bar{x}, \bar{u}), \quad \bar{x}(t_k) = x_k \quad (27b)$$

$$\dot{\bar{\eta}} = A\bar{\eta} + B\bar{w}_t, \quad \bar{\eta}(t_k) = \bar{\eta}_{k-1}^*(t_{k-1} + T_s) \quad (27c)$$

$$\bar{u}(t) \in U, \quad \forall t \in [t_k, t_k + T], \quad (27d)$$

with  $\bar{v}_t = [\bar{u}^T \ \bar{w}_t^T]^T$  and the cost functional

$$J_t(t_k, \bar{v}_t(\cdot)) = \int_{t_k}^{t_k+T} l(\bar{x}, \gamma_1(\bar{\eta}), \bar{u}, \gamma_2(\bar{\eta}, \bar{w}_t)) dt + V(\bar{x}(t_k + T), \gamma_1(\bar{\eta}(t_k + T))). \quad (28)$$

The term  $\bar{\eta}_{k-1}^*(t_{k-1} + T_s)$  denotes the optimal solution of the previous sampling instant, evaluated at  $t_{k-1} + T_s$ , where for  $t_0$ ,  $\bar{\eta}(t_0) = \gamma_1^{-1}(\bar{\zeta}(t_0))$  is used. Utilizing (25) and (26), one can easily calculate the optimal quantities  $\bar{\zeta}^*$  and  $\bar{w}^*$  from the optimal solution of (27)  $\bar{\eta}^*$  and  $\bar{w}_t^*$ . The solution of (27) automatically satisfies the constraint (22) and  $J(t_k, \bar{v}^*(t, t_k)) = J_t(t_k, \bar{v}_t^*(t, t_k))$ .

Another rationale for omitting an explicit implementation of  $\zeta_1 \geq \theta_0$  is that the solution of the transformed OCP (27) exhibits singular arcs (cf. [39]) whenever the original constraint (22) is fulfilled with equality (cf. [38]). At the initial point, where  $\zeta_1(t_0) = \theta_0$ , this would lead to numerical difficulties in the solution procedure of the gradient projection method.

The OCP (27) can be solved with GPMPC in a receding horizon fashion, which results in a real-time capable path-following MPC.

## V. SIMULATION AND EXPERIMENTAL RESULTS

This section contains performance evaluations of the proposed MPC strategy and results from the laboratory tower crane system introduced in Section II. In addition, robustness investigations and a comparison with a trajectory tracking MPC are carried out.

The output path defined by (13) is represented by means of a four times continuously differentiable B-spline curve with the path parameter values at the beginning and at the end of the curve satisfying  $\theta_0 = 0$  and  $\theta_1 = 1$ , respectively. Henceforth two different geometric paths  $p_{yL,1}$  and  $p_{yL,2}$  will be considered. Both output paths start at  $p_{yL,i}(\theta_0) = [0.35 \text{ m} \ 0.5 \text{ m} \ 0.15 \text{ m}]^T$ ,  $i = 1, 2$ . The initial condition of the system is either chosen in conformity with this starting point (nominal initial condition) or with a slight deviation according to  $y_L = [0.4 \text{ m} \ 0.5 \text{ m} \ 0.2 \text{ m}]^T$ . Figure 4 shows

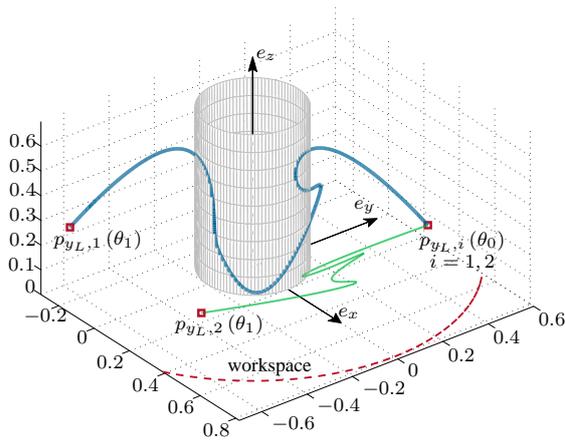


Fig. 4. Desired output paths  $p_{yL,1}(\theta)$  and  $p_{yL,2}(\theta)$  represented by four times continuously differentiable B-spline curves (all quantities in m).

TABLE I  
COMPUTATION TIME OF THE GRADIENT PROJECTION METHOD FOR DIFFERENT NUMBERS OF ITERATIONS  $N$ .

$N$	Computation time in ms	
	mean	standard deviation
2	0.85	0.065
3	1.28	0.073
5	2.13	0.097
10	4.27	0.17

the desired output paths in the workspace of the crane. The cylinder and the dashed arc represent the inner and outer constraints for  $s_1$  with values 0.2 m and 0.8 m, respectively. The complicated shapes of the desired output paths are chosen for demonstration purposes to illustrate the performance of the proposed control scheme.

The weighting matrices of the path-following MPC are chosen according to

$$Q_e = \text{diag} \left( \begin{bmatrix} 1000 & 1000 & 1000 & 200 & 200 \\ & & & 50 & 50 & 50 & 5 & 5 \end{bmatrix} \right) \quad (29a)$$

$$Q_\zeta = \text{diag} \left( \begin{bmatrix} 5 & 2 & 1 & 1 & 1 \end{bmatrix} \right) \quad (29b)$$

$$R_u = \text{diag} \left( \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \right) \quad (29c)$$

$$R_w = 0.05 \quad (29d)$$

$$S_\zeta = \text{diag} \left( \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} \right) \quad (29e)$$

and  $S_e$  equal to zero. The control algorithm is executed on the real-time operating system MATLAB xPC Target on an Intel Core-Duo CPU E6700 with 2.7 GHz. Forward and backward integration of the system and the adjoint differential equations within the gradient projection method is carried out by means of Heun's method with 21 equally spaced discretization points on an optimization horizon of  $T = 2$  s. To illustrate the real-time capability, the execution time of the gradient projection method according to Algorithm 1 is determined for different numbers of iterations  $N$ . The respective number of iterations is carried out for one fixed initial condition 1000 times in order to calculate the mean and standard deviation of the execution times which are shown in Table I. The results confirm the

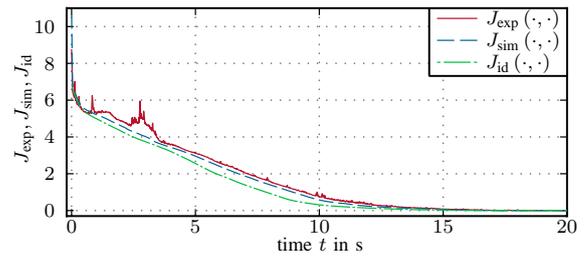


Fig. 5. Values of the cost functional over time for deviating initial conditions.

linear run-time complexity in the number of iterations  $N$ . Considering the sampling time  $T_s = 2$  ms,  $N = 2$  is chosen, which results in a satisfactory control performance and leaves enough time for other tasks to be fulfilled by the CPU. Due to the small sampling time, it suffices to apply the constant value  $\bar{v}_{(N)}(t_k, t_k)$  during the sampling interval  $[t_k, t_k + T_s)$  to the system.

At the laboratory experiment the control structure according to Fig. 3 is utilized. All subsequent simulation studies are carried out with the comprehensive model (1) of the laboratory tower crane. In particular the same velocity controllers as for the laboratory experiment are used in the simulation.

#### A. Evaluation of the Proposed Path-Following Controller

In the following, the indices  $_{exp}$  and  $_{sim}$  refer to the experimental and simulation results obtained from the application of the proposed path-following MPC to the laboratory tower crane. Furthermore, quantities labelled with the index  $_{id}$  denote the simulated results of the "ideal" MPC case, where the optimal solution of (18) (up to numerical accuracy) is calculated in each sampling instant and applied to the system. If confusion is excluded, the indices are omitted. In all cases the desired geometric path is given by  $p_{yL,1}$ . The following Subsections V-A1 and V-A2 present results of the proposed path-following control concept for the scenarios of a deviating initial condition and an external disturbance force acting on the load. A video of these test cases can be found on [http://www.acin.tuwien.ac.at/fileadmin/cds/videos/crane\\_pathFollowingNMPC.wmv](http://www.acin.tuwien.ac.at/fileadmin/cds/videos/crane_pathFollowingNMPC.wmv).

1) *Deviation from the Nominal Initial Condition:* The time evolutions of the values of the cost functional in each sampling instant are shown in Fig. 5. The requirement of monotonically decreasing values of the cost functional is fulfilled, apart from small time intervals in the experiment, which result from measurement inaccuracies and the approximate differentiation scheme to estimate the velocities.

Figure 6 shows  $\zeta_{1,exp}$  together with the virtual input  $w_{exp}$  compared to  $\zeta_{1,id}$  from the ideal MPC-case. The quantities  $\zeta_{1,exp}$  and  $w_{exp}$  are calculated by means of (25) and (26). The constraint  $\zeta_1 \leq \theta_1$  is fulfilled and the assumption  $\zeta_1 \geq \theta_0$  is also justified. At the laboratory experiment, the motion of the load along the path needs slightly more time compared to the ideal MPC-case. This results from the suboptimal MPC implementation and the measurement and modelling inaccuracies at the laboratory experiment.

Figure 7 shows the inputs  $u$  from the laboratory experiment in comparison with  $\hat{u}$  which is calculated by inserting  $\zeta(t)$

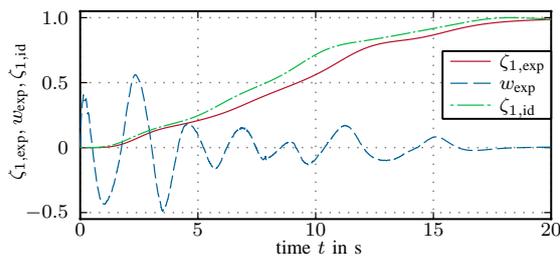


Fig. 6. Path parameter evolution and virtual input  $w$  of the auxiliary system for deviating initial conditions.

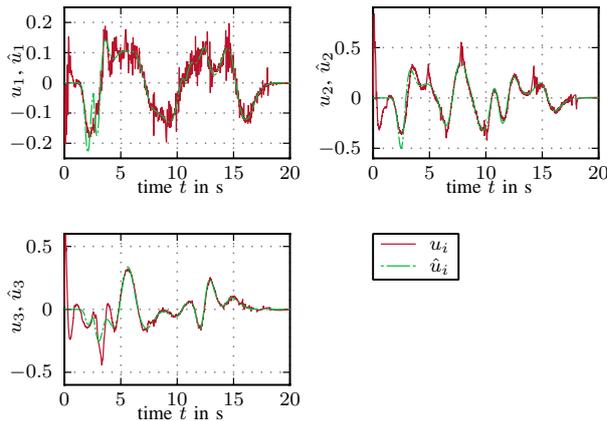


Fig. 7. Control input  $u$  for deviating initial conditions (all quantities in  $\text{m s}^{-2}$  and  $\text{rad s}^{-2}$ , respectively).

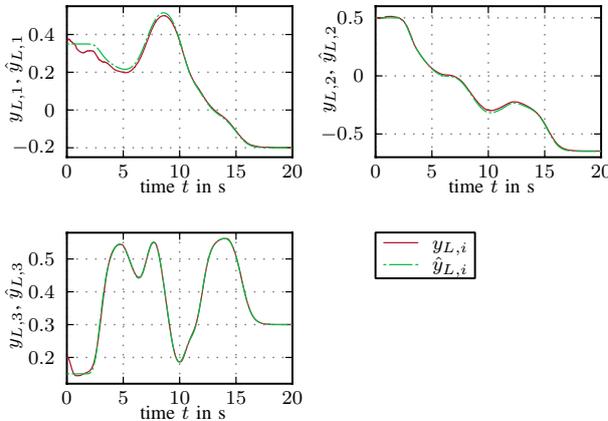


Fig. 8. Comparison of  $y_L$  with  $\hat{y}_L$  for deviating initial conditions (all quantities in m).

into (16). The good agreement confirms the quality of the underlying mathematical model.

The performance of the path-following MPC at the laboratory experiment can be assessed by means of Figs. 8 and 9. Figure 8 shows the actual position of the load  $y_L$  in comparison with the desired geometric path  $\hat{y}_L$ . Due to a lack of appropriate sensors,  $y_L(t)$  is not directly measured but instead calculated by inserting the measurement  $x(t)$  into (11). The quantity  $\hat{y}_L(t)$  is obtained from (13) with  $\theta = \zeta_1(t)$ . Note that the controller not only aims at tracking the path in an appropri-

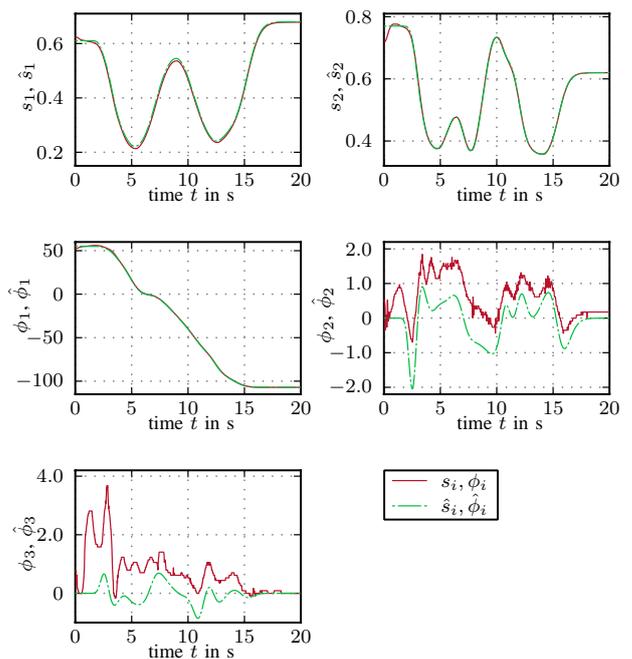


Fig. 9. Trajectories of the states  $s_1, s_2, \phi_1, \phi_2$ , and  $\phi_3$  in comparison with the desired trajectories for deviating initial conditions (all quantities in m and  $^\circ$ , respectively).

ate manner but also compensates for the deviation in the initial condition of the system. The end point of the desired geometric path lies at  $p_{y_{L,1}}(\theta_1) = [-0.2 \text{ m} \ -0.65 \text{ m} \ 0.3 \text{ m}]^T$ . Figure 6 shows a small deviation of  $\zeta_{1,\text{exp}}$  from  $\theta_1$  at  $t = 20 \text{ s}$ . Despite this fact, the load is virtually at the end point of the path with  $y_L(20) = [-0.199 \text{ m} \ -0.647 \text{ m} \ 0.3002 \text{ m}]^T$ . Furthermore, the quality of the proposed controller can be evaluated by comparing the measurements of the states  $x(t)$  with the desired zero path error manifold according to (15). This study is carried out for the states  $s_1, s_2, \phi_1, \phi_2$ , and  $\phi_3$  in Fig. 9. The desired trajectory in the state space is calculated by inserting  $\zeta(t)$  into (15). The corresponding quantities are marked with  $\hat{\cdot}$ . Again, the compensation for the initial deviation is clearly visible. The differences in  $\phi_2$  and  $\phi_3$  stem from the loose guidance of the cable in the sleeve attached to the gimbal (cf. Section II-A) and from sticktion effects.

2) *Deviation from the Nominal Initial Condition and External Force on the Load:* In this subsection, not only the deviation in the initial condition of the system is considered but additionally a disturbance force acting on the load is included. The following results solely refer to the measurements obtained from the laboratory experiment. As can be seen in Figs. 11 and 12, impulse-like disturbance forces are applied to the load at  $t \approx 6 \text{ s}$  and  $t \approx 24 \text{ s}$ , respectively. These times are marked with vertical dashed lines. One disturbance occurs while the path is traversed and the other one at the final position after the path traverse is completed. All quantities are calculated as explained in the previous Subsection V-A1.

Figure 10 shows the evolution of the path parameter and

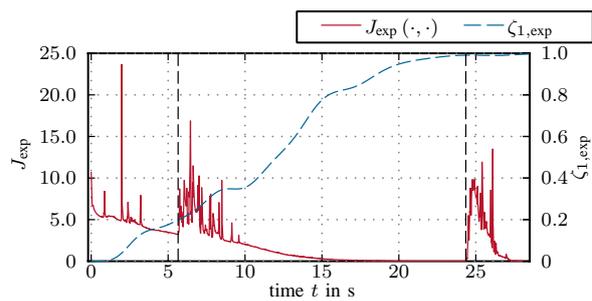


Fig. 10. Path parameter evolution and values of the cost functional for a disturbance force acting on the load.

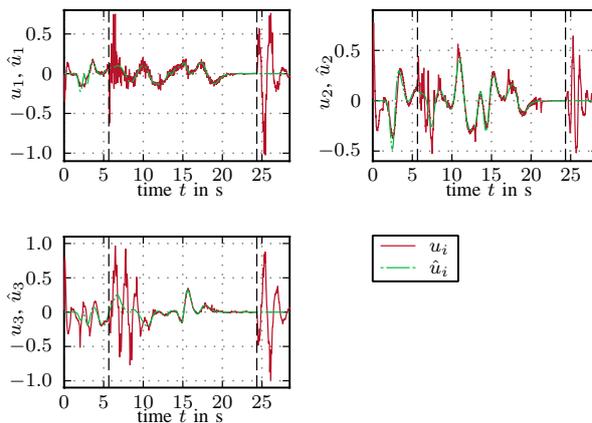


Fig. 11. Control input  $u$  for a disturbance force acting on the load (all quantities in  $\text{m s}^{-2}$  and  $\text{rad s}^{-2}$ , respectively).

the values of the cost functional in each sampling instant. The disturbance at  $t \approx 6$  s causes the path parameter evolution to slow down resulting in a complete stop at  $t \approx 9$  s. As a result, the overall travel along the path lasts a few seconds longer than in the undisturbed case, cf. Figs. 6 and 10. As soon as the value of the cost functional returns to the value it had without disturbance, the path traverse is continued with the cost functional values converging to zero in a monotonically decreasing manner. The disturbance at  $t \approx 24$  s has a significant influence on the position of the load (cf. Fig. 12) with a displacement in  $e_x$ - and  $e_y$ -direction of approximately 0.2 m and 0.15 m, respectively. Thus, large values of the cost functional are the result, which, however, are quickly forced to zero. The spikes in the values of  $J_{\text{exp}}$  result from measurement inaccuracies, the approximate differentiation scheme to estimate the velocities, and from friction effects.

The time evolutions of the control inputs  $u_i$  together with their flatness-based parameterization  $\hat{u}_i$ ,  $i = 1, 2, 3$ , are depicted in Fig. 11. The effect of the disturbances can be clearly inferred from the difference between  $u_i$  and  $\hat{u}_i$ . However, after compensating for the disturbances, the corresponding values match again quite well.

Figure 12 shows that the deviation of the load position  $y_L$  from the nominal initial condition as well as the disturbances resulting from the external forces are well compensated.

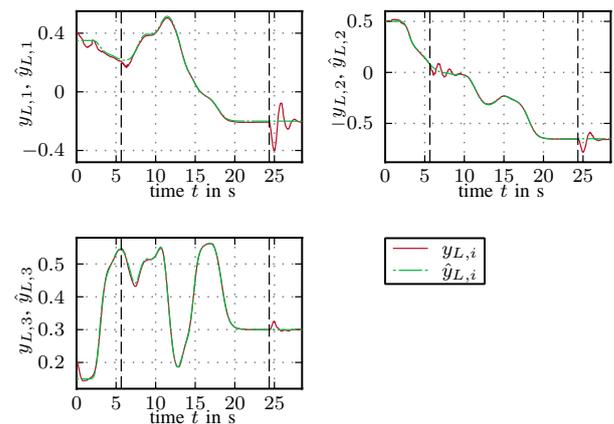


Fig. 12. Comparison of  $y_L$  with  $\hat{y}_L$  for a disturbance force acting on the load (all quantities in m).

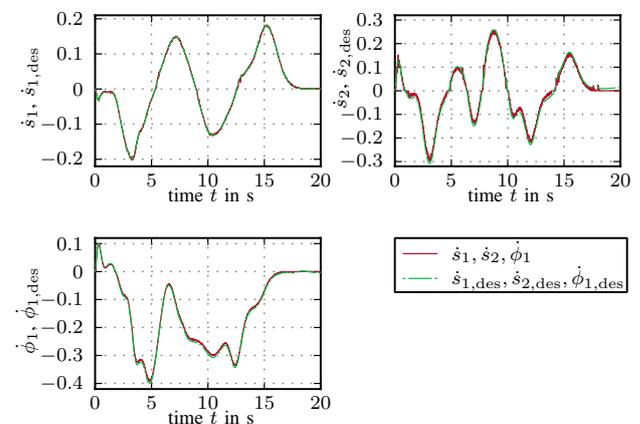


Fig. 13. Comparison of the desired velocities  $\hat{s}_{1,\text{des}}$ ,  $\hat{s}_{2,\text{des}}$ ,  $\hat{\phi}_{1,\text{des}}$  with the actual velocities  $s_1$ ,  $s_2$ ,  $\phi_1$  at the laboratory experiment for deviating initial conditions (all quantities in  $\text{m s}^{-1}$  and  $\text{rad s}^{-1}$ , respectively).

### B. Robustness Investigations

According to Section II-C the path-following MPC assumes perfectly working velocity controllers. In this regard, Fig. 13 shows the comparison of the desired with the actual velocities of the actuated axes at the laboratory experiment for the test case of deviating initial conditions (cf. Subsection V-A1). The desired velocities  $\hat{s}_{1,\text{des}}$ ,  $\hat{s}_{2,\text{des}}$ , and  $\hat{\phi}_{1,\text{des}}$  are calculated by integration of the control inputs  $u_i$ ,  $i = 1, 2, 3$ , according to Fig. 7. As can be inferred from Fig. 13, the velocity controllers work quite well at the laboratory experiment which also applies to the simulations. However, the assumption of perfectly working velocity controllers is never fulfilled exactly. Therefore, it is of interest to investigate if the MPC still achieves satisfactory results even if the velocity controllers are far from being ideal.

In practice, a bad performance of the velocity controllers may be caused, e.g., by large variations of the load mass or unknown friction effects. To investigate possible consequences of these effects simulation studies are carried out using velocity controllers which are badly tuned on purpose. Again, the scenario of deviating initial conditions is considered. Figure 14

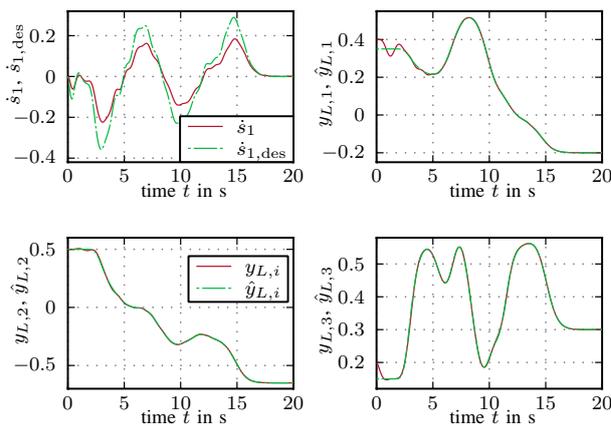


Fig. 14. Comparison of the desired velocity  $\dot{s}_1, \dot{s}_{1,des}$  with the actual velocity  $\dot{s}_1$  and comparison of  $y_L$  with  $\hat{y}_L$  for deviating initial conditions and on purpose badly tuned velocity controllers (all quantities in  $\text{m s}^{-1}$  and  $\text{m}$ , respectively).

shows the actual position of the load  $y_L$  in comparison with the desired geometric path  $\hat{y}_L$ . Additionally,  $\dot{s}_1$  is compared to  $\dot{s}_{1,des}$ . Despite the large deviations in the velocity (which occur similarly for  $\dot{s}_2$  and  $\dot{\phi}_1$ ), the path-following MPC is still able to follow the reference very well which can be seen in Fig. 14 by comparing  $y_L$  with  $\hat{y}_L$ . Therefore, the much smaller deviations at the laboratory experiment according to Fig. 13 are not critical for the path-following MPC. Furthermore, the assumption of perfectly working velocity controllers for the mathematical model (3) is not crucial.

### C. Comparison with a Trajectory Tracking Controller

In this subsection, the proposed path-following control concept (PFC) is compared to a model predictive trajectory tracking controller (TTC). The TTC also utilizes GPMPC as described in Section IV-A and is based on the OCP

$$\min_{\bar{u}(\cdot)} J_{\text{TTC}}(t_k, \bar{u}(\cdot)) \quad (30a)$$

subject to

$$\dot{\bar{x}} = f(\bar{x}, \bar{u}), \quad \bar{x}(t_k) = x_k \quad (30b)$$

$$\bar{u}(t) \in U, \quad \forall t \in [t_k, t_k + T] \quad (30c)$$

with the cost functional

$$J_{\text{TTC}}(t_k, \bar{u}(\cdot)) = \frac{1}{2} \left[ \bar{x}(t_k + T) - x_d(t_k + T) \right]_{S_e} + \frac{1}{2} \int_{t_k}^{t_k + T} \left[ \left[ \bar{x}(t) - x_d(t) \right]_{Q_e} + \left[ \bar{u}(t) \right]_{R_u} \right] dt \quad (31)$$

and the desired trajectory of the states  $x_d(t)$  which is generated based on the parameterized zero path error manifold (15). To this end a monotonically increasing four times continuously differentiable function

$$[0, T_t] \ni t \rightarrow \theta_t(t) \in [\theta_0, \theta_1] \quad (32)$$

with  $\theta_t(0) = \theta_0$  and  $\theta_t(T_t) = \theta_1$  is defined where  $T_t$  denotes the end time of the trajectory. The desired trajectory of the states follows as  $x_d(t) = p_x(\zeta_t(t))$  with  $\zeta_t^T(t) =$

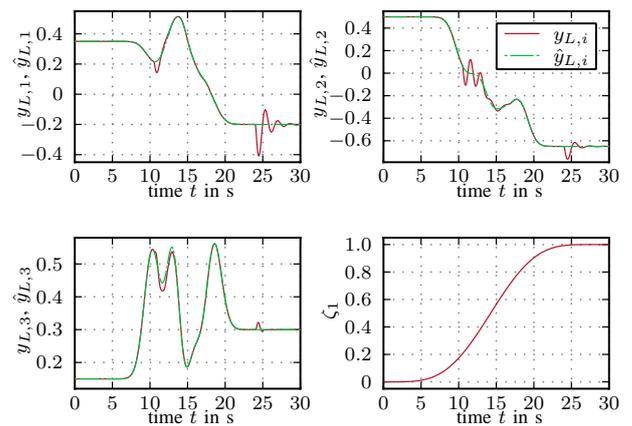


Fig. 15. Comparison of  $y_L$  with  $\hat{y}_L$  and path parameter evolution for the trajectory tracking controller and the path  $p_{y_L,1}$  of Fig. 4 ( $y_L$  and  $\hat{y}_L$  in  $\text{m}$ ).

$\left[ \theta_t(t) \quad \dot{\theta}_t(t) \quad \ddot{\theta}_t(t) \quad \theta_t^{(3)}(t) \quad \theta_t^{(4)}(t) \right]$ . The weighting matrices in the cost functional (31) are the same as for the PFC.

In order to hit the input constraints, in the following the set of feasible inputs is changed to

$$U = \left\{ u \in \mathbb{R}^3 \mid \begin{aligned} -0.8 \text{ m s}^{-2} &\leq u_1 \leq 0.8 \text{ m s}^{-2}, \\ -0.8 \text{ m s}^{-2} &\leq u_2 \leq 0.8 \text{ m s}^{-2}, \\ -0.8 \text{ rad s}^{-2} &\leq u_3 \leq 0.8 \text{ rad s}^{-2} \end{aligned} \right\}. \quad (33)$$

The reason for this step is that the physically motivated constraints (5) are quite generous. Thus, unreasonable measures, e.g., huge disturbances, would have to be taken to come close to one of the original input constraints (5).

Two scenarios for comparing the PFC and the TTC are considered with the desired geometric paths given by  $p_{y_L,1}$  and  $p_{y_L,2}$ , respectively (cf. Fig. 4). In both test cases the system starts at the nominal initial condition. The first scenario comprises two impulse-like disturbance forces on the load similar to Subsection V-A2 whereas in the second scenario no disturbances are considered. For reasons of reproducibility and to be able to apply the same disturbance to the system for both controllers the investigations are carried out in the simulation and not on the laboratory experiment (to this end the model (1) is appropriately extended). A further benefit of this strategy is that the essential results are not distorted by, e.g., measurement inaccuracies.

Concerning the first scenario, Fig. 15 shows the actual position  $y_L$  of the load in comparison with the desired trajectory  $\hat{y}_L$  for the TTC together with  $\zeta_1(t)$  given by (32). In Fig. 16 the inputs and  $\zeta_1(t)$  for the PFC are depicted. The corresponding results for  $y_L$  and  $\hat{y}_L$  are similar to Fig. 12 and are omitted for brevity. The first disturbance causes the path parameter evolution of the PFC to stop with a short period of time where  $\dot{\theta}(t) < 0$ . Additionally the input constraints for  $u_2$  and  $u_3$  according to (33) get active. The deviations resulting from the second disturbance are well compensated by both controllers in a very similar manner.

Figure 17 shows  $y_L$  in comparison with  $\hat{y}_L$  for the PFC and the TTC in the workspace of the crane. The corresponding results are reduced to the region of the path parameter where

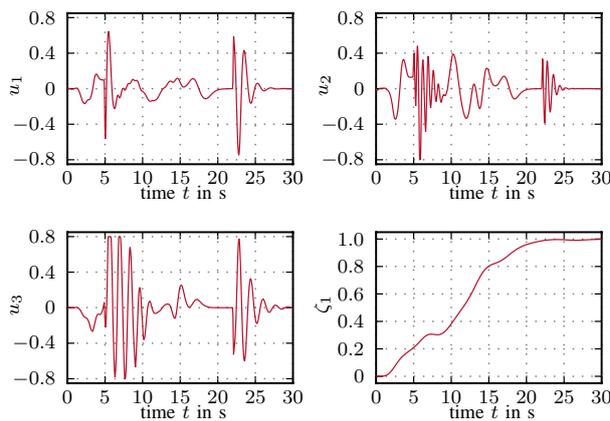


Fig. 16. Control input  $u$  and path parameter evolution for the path-following controller and the path  $p_{y_L,1}$  of Fig. 4 ( $u_i$  in  $\text{m s}^{-2}$  and  $\text{rad s}^{-2}$ , respectively).

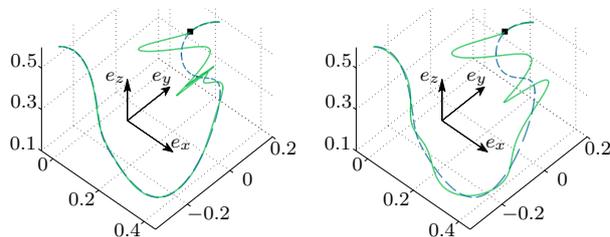


Fig. 17. Comparison of  $y_L$  with  $\hat{y}_L$  (dashed) for the path-following (left) and the trajectory tracking controller (right) and the path  $p_{y_L,1}$  of Fig. 4 (all quantities in m).

the effects of the first disturbance force are visible (the small squares mark the points where the impulse-like force is acting on the load). The superiority of the PFC in terms of tracking the geometric reference is clearly highlighted.

The second scenario is concerned with following the path  $p_{y_L,2}$  (cf. Fig. 4), which comprises several sharp turns. Figures 18 and 19 show the inputs  $u$  from the PFC and the TTC, respectively. Additionally, the time evolutions of  $\zeta_1(t)$  are depicted. The time needed for the overall travel along the path as well as the performance in tracking the geometric reference is virtually the same for both controllers. However, the input  $u_3$  from the TTC touches the upper constraint during a short period of time which precludes exact tracking. This emphasizes the fact that it is in general a non-trivial task to choose an appropriate end time  $T_t$  and  $\theta_t(t)$  such that system constraints are respected. The PFC slows down the evolution of the path parameter during sharp turns to avoid touching the constraints. Generally this behaviour results in smaller values of the control inputs.

## VI. CONCLUSION

A path-following MPC scheme for a laboratory tower crane was presented. Starting from the geometric path of the load, the corresponding zero path error manifold in the state space was obtained using the theory of differential flatness. The proposed controller aims at tracking the zero path error manifold while additionally determining the time evolution

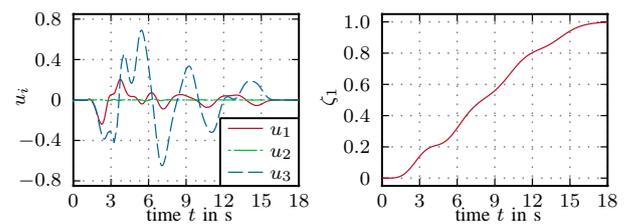


Fig. 18. Control input  $u$  and path parameter evolution for the path-following controller and the path  $p_{y_L,2}$  of Fig. 4 ( $u_i$  in  $\text{m s}^{-2}$  and  $\text{rad s}^{-2}$ , respectively).

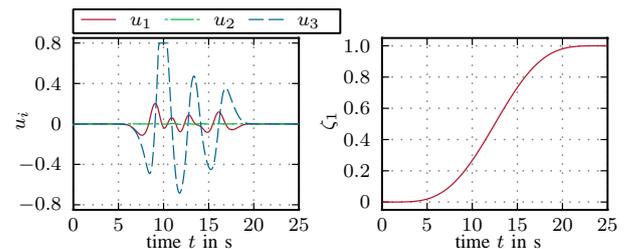


Fig. 19. Control input  $u$  and path parameter evolution for the trajectory tracking controller and the path  $p_{y_L,2}$  of Fig. 4 ( $u_i$  in  $\text{m s}^{-2}$  and  $\text{rad s}^{-2}$ , respectively).

along the path. To this end, an auxiliary system was defined which describes the time evolution of the path parameter and its derivatives. The control input of the auxiliary system was determined by MPC in such a way that the deviation from the zero path error manifold is kept as small as possible. For the design of the control scheme, existing ideas on path-following MPC were adopted but formulated without terminal constraints. This not only allows for the consideration of system constraints but also permits the application of a fast MPC scheme based on the gradient projection method. In each sampling instant, the gradient projection method is terminated after a predefined finite number of iterations which guarantees real-time feasibility even with very small sampling times.

Results from simulation and experimental studies on a laboratory tower crane show promising results and emphasize the usefulness of the proposed approach. Robustness studies underline the validity of the proposed control concept. The comparison with a trajectory tracking controller reveals the advantages of the path-following controller. Current work is dedicated to the investigation of stability and convergence issues of the proposed path-following MPC scheme.

## REFERENCES

- [1] A. P. Aguiar, D. B. Dačić, J. P. Hespanha, and P. Kokotović, "Path-following or reference-tracking? An answer relaxing the limits to performance," in *Proc. IAV2004 - 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisboa, Portugal, Jul. 2004.
- [2] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright, "Nonlinear predictive control and moving horizon estimation - An introductory overview," in *Advances in Control: Highlights of ECC '99*, P. M. Frank, Ed. London: Springer, 1999, pp. 391–449.
- [3] A. Jadbabaie, J. Yu, and J. Hauser, "Unconstrained receding-horizon control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 46, no. 5, pp. 776–783, May 2001.
- [4] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *IEEE Trans. Autom. Control*, vol. 44, no. 3, pp. 648–654, Mar. 1999.

- [5] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 35, no. 7, pp. 814–824, Jul. 1990.
- [6] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Trans. Autom. Control*, vol. 38, no. 11, pp. 1623–1633, Nov. 1993.
- [7] M. Diehl, R. Findeisen, F. Allgöwer, H. G. Bock, and J. P. Schlöder, "Nominal stability of real-time iteration scheme for nonlinear model predictive control," *IEE Proceedings Control Theory and Applications*, vol. 152, no. 3, pp. 296–308, May 2005.
- [8] K. Graichen and A. Kugi, "Stability and incremental improvement of suboptimal MPC without terminal constraints," *IEEE Trans. Autom. Control*, vol. 55, no. 11, pp. 2576–2580, Nov. 2010.
- [9] J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*, ser. Advances in Design and Control. Philadelphia, USA: SIAM - Society for Industrial and Applied Mathematics, 2001.
- [10] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, Jun. 2002.
- [11] C. Kirches, L. Wirsching, H. G. Bock, and J. P. Schlöder, "Efficient direct multiple shooting for nonlinear model predictive control on long horizons," *Journal of Process Control*, vol. 22, no. 3, pp. 540–550, Mar. 2012.
- [12] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions in Biomechanics and Robotics*, ser. Lecture Notes in Control and Information Sciences, M. Diehl and K. Mombaur, Eds. Berlin, Heidelberg: Springer, 2006, no. 340, pp. 65–93.
- [13] L. S. Pontryagin, V. G. Boltyansky, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*. Oxford: Pergamon Press, 1964.
- [14] K. Graichen and B. Käpfernick, "A real-time gradient method for nonlinear model predictive control," in *Frontiers of Model Predictive Control*, T. Zheng, Ed. Rijeka, Croatia: InTech, Feb. 2012, pp. 9–28.
- [15] K. Graichen, "A fixed-point iteration scheme for real-time model predictive control," *Automatica*, vol. 48, no. 7, pp. 1300–1305, Jul. 2012.
- [16] A. Banaszuk and J. Hauser, "Feedback linearization of transverse dynamics for periodic orbits," *Systems & Control Letters*, vol. 26, no. 2, pp. 95–105, Sep. 1995.
- [17] C. Nielsen and M. Maggiore, "Maneuver regulation via transverse feedback linearization: Theory and examples," in *Proc. 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, Stuttgart, Germany, Sep. 2004, pp. 59–66.
- [18] P. Encarnação and A. Pascoal, "Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft," in *Proc. 40th IEEE Conference on Decision and Control*, Orlando, Florida, USA, Dec. 2001, pp. 964–969.
- [19] R. Skjetne, T. I. Fossen, and P. V. Kokotović, "Robust output maneuvering for a class of nonlinear systems," *Automatica*, vol. 40, no. 3, pp. 373–383, Mar. 2004.
- [20] D. B. Dačić, D. Nešić, and P. V. Kokotović, "Path-following for nonlinear systems with unstable zero dynamics," *IEEE Trans. Autom. Control*, vol. 52, no. 3, pp. 481–487, Mar. 2007.
- [21] D. Lam, C. Manzie, and M. Good, "Model predictive contouring control," in *Proc. 49th IEEE Conference on Decision and Control*, Atlanta, GA, USA, Dec. 2010, pp. 6137–6142.
- [22] T. Faulwasser, B. Kern, and R. Findeisen, "Model predictive path-following for constrained nonlinear systems," in *Proc. 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference CDC/CCC*, Shanghai, P.R. China, Dec. 2009, pp. 8642–8647.
- [23] T. Faulwasser and R. Findeisen, "Constrained output path-following for nonlinear systems using predictive control," in *Proc. 8th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, Bologna, Italy, Sep. 2010, pp. 753–758.
- [24] —, "Predictive path following without terminal constraints," in *Proc. 20th Int. Symposium on Mathematical Theory of Networks and Systems*, Melbourne, Australia, Jul. 2012.
- [25] S. Yu, X. Li, H. Chen, and F. Allgöwer, "Nonlinear model predictive control for path following problems," in *Proc. 4th IFAC Nonlinear Model Predictive Control Conference*, Noordwijkerhout, Netherlands, Aug. 2012, pp. 145–150.
- [26] D. Lam, C. Manzie, and M. Good, "Application of model predictive contouring control to an x-y table," in *Proc. 18th IFAC World Congress*, Milano, Italy, Aug./Sep. 2011, pp. 10 325–10 330.
- [27] K. Graichen, M. Egretzberger, and A. Kugi, "Suboptimal model predictive control of a laboratory crane," in *Proc. 8th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, Bologna, Italy, Sep. 2010, pp. 397–402.
- [28] M. Egretzberger, K. Graichen, and A. Kugi, "Flatness-based MPC and global path planning towards cognition-supported pick-and-place tasks of tower cranes," in *Advanced Dynamics and Model-Based Control of Structures and Machines*, H. Irschik, M. Krommer, and A. K. Belyaev, Eds. Vienna, Austria: Springer, 2012, pp. 63–72.
- [29] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: John Wiley & Sons, 1989.
- [30] H. K. Khalil, *Nonlinear Systems*, 2nd ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 1996.
- [31] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.
- [32] T. Faulwasser, *Optimization-based Solutions to Constrained Trajectory-tracking and Path-following Problems*, ser. Contributions in Systems Theory and Automatic Control. Aachen, Germany: Shaker, Jan. 2013, no. 3.
- [33] C. Raczky and G. Jacob, "Fast and smooth controls for a trolley crane," *Journal of Decision Systems*, vol. 8, no. 3, pp. 367–388, 1999.
- [34] T. Faulwasser, V. Hagenmeyer, and R. Findeisen, "Optimal exact path-following for constrained differentially flat systems," in *Proc. 18th IFAC World Congress*, Milano, Italy, Aug./Sep. 2011, pp. 9875–9880.
- [35] A. Jadbabaie and J. Hauser, "On the stability of receding horizon control with a general terminal cost," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 674–678, May 2005.
- [36] M. Reble and F. Allgöwer, "Unconstrained nonlinear model predictive control and suboptimality estimates for continuous-time systems," in *Proc. 18th IFAC World Congress*, Milano, Italy, Aug./Sep. 2011, pp. 6733–6738.
- [37] K. Graichen and N. Petit, "Incorporating a class of constraints into the dynamics of optimal control problems," *Optimal Control, Applications and Methods*, vol. 30, no. 6, pp. 537–561, Nov./Dec. 2009.
- [38] D. H. Jacobson and M. M. Lele, "A transformation technique for optimal control problems with a state variable inequality constraint," *IEEE Trans. Autom. Control*, vol. 14, no. 5, pp. 457–464, Oct. 1969.
- [39] H. J. Kelley, R. E. Kopp, and H. G. Moyer, "Singular extremals," in *Topics in Optimization*, G. Leitmann, Ed. New York, USA: Academic Press, 1967, vol. 31, ch. 3, pp. 63–101.



**Martin Böck** received the Dipl.-Ing. degree in mechatronics from Johannes Kepler University, Linz, Austria, in 2010.

Since 2010, he works as a university assistant at the Automation and Control Institute at Vienna University of Technology. His research interests include distributed optimization, optimization based control methods, model predictive control and differential geometric methods for nonlinear control.



**Andreas Kugi** (M'94) received the Dipl.-Ing. degree in electrical engineering from Graz University of Technology, Austria, and the Ph.D. (Dr. techn.) degree in control engineering from Johannes Kepler University (JKU), Linz, Austria, in 1992 and 1995, respectively. He received his "Habilitation" degree in the field of automatic control and control theory from JKU in 2000.

From 1995 to 2000 he worked as an assistant professor and from 2000 to 2002 as an associate professor at JKU. In 2002, he was appointed full professor at Saarland University, Saarbrücken, Germany, where he held the Chair of System Theory and Automatic Control until May 2007. Since June 2007 he is a full professor for complex dynamical systems and head of the Automation and Control Institute at Vienna University of Technology, Austria. Since 2010, he is corresponding member of the Austrian Academy of Sciences. His research interests include the physics-based modeling and control of (nonlinear) mechatronic systems, differential geometric and algebraic methods for nonlinear control, model predictive control and the controller design for infinite-dimensional systems. He is Editor-in-Chief of the *Control Engineering Practice*.