Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

# PLC Implementation of a Nonlinear Model Predictive Controller

**B. Käpernick K. Graichen**

*Institute of Measurement, Control, and Microtechnology, University of Ulm, Germany (e-mail: {bartosz.kaepernick,knut.graichen}@uni-ulm.de)*

**Abstract:** This paper describes the implementation of an efficient model predictive control (MPC) approach on a standard programmable logic controller (PLC). PLCs are commonly used in industrial automation but are typically limited in view of computational performance and memory. The MPC scheme in this contribution is well suited for nonlinear input constrained systems with sampling times in the millisecond range and can also be applied to tracking control problems. Experimental results for a laboratory crane demonstrate the real-time applicability of the PLC implementation.

*Keywords:* nonlinear model predictive control, programmable logic controller, real-time feasibility

## 1. INTRODUCTION

The solution of an optimal control problem (OCP) along a moving horizon is the basis of model predictive control (MPC) (Mayne et al., 2000; Camacho and Bordons, 2003; Grüne and Pannek, 2011). This modern control method has become popular in the recent years due to its ability to handle nonlinear multiple input systems with constraints. However, the computational effort that is typically required to solve the underlying OCP usually limits the application of nonlinear MPC to sufficiently slow and/or low-dimensional systems. This drawback makes it also difficult to perform control tasks on hardware systems with limited performance and memory. A typical example are programmable logic controllers (PLCs) that are commonly used in industrial automation. In general, rather simple control strategies (such as PI control) are implemented on a PLC due to the limited resources.

There exist several approaches demonstrating the real-time applicability of nonlinear MPC. The algorithm developed in Ohtsuka (2004) employs a continuation method in combination with the generalized minimum residual (GMRES) method to solve an OCP in real-time. The problem formulation and the first-order optimality conditions are discretized and the solution of the resulting nonlinear algebraic equation is traced over the single MPC steps. The real-time iteration (RTI) scheme developed in Diehl et al. (2002) is based on the multiple shooting method and uses a Newton-type framework. Reduced algorithmic components of the RTI scheme are implemented within the ACADO Toolkit (Houska et al., 2011) with an automatic C code generation for real-time MPC. The MPC approach presented in Graichen and Käpernick (2012) exploits the structure of the optimality conditions and uses a gradient-based algorithm with a fixed number of iterations in order to obtain real-time applicability. Additionally, the previous iteration of the control trajectory is used for re-initialization in order to successively reduce the optimality error. Further MPC approaches aiming at real-time feasibility are presented in DeHaan and Guay (2007) and Zavala and Biegler (2009). The suboptimal MPC scheme (DeHaan and Guay, 2007) is based on a descent condition and accounts for constraints via an interior point formulation, whereas the advanced-step MPC method (Zavala and Biegler, 2009) predicts the future states in order to solve the corresponding future OCP in advance.

There are also approaches with regard to implementing and running a model predictive controller on a PLC. Valencia-Palomo and Rossiter (2011) discussed a suboptimal MPC approach based on parametric solutions with reduced complexity for PLC applications. Four examples of varying complexity were used to demonstrate the capabilities of the method. In Syaichu-Rohman and Sirius (2011) a PLC implementation of an MPC was applied for controlling the speed of a DC motor by means of a fast algorithm to solve quadratic programs (QP). Rauová et al. (2011) controlled a fan heater system on a PLC using an MPC scheme based on a parametric programming technique, which encodes the optimal control moves as a lookup table. The applicability of different online optimization approaches regarding the solution of a quadratic program by a model predictive controller on a PLC was investigated by Huyck et al. (2012). However, the mentioned approaches implemented linear model predictive controller on the PLC systems.

To the authors' knowledge, no results are published so far concerning the implementation of a nonlinear MPC scheme on a PLC and hence this subject is addressed in this paper. The considered model predictive control approach is based on a (projected) gradient algorithm (Graichen and Käpernick, 2012) and can be used for controlling nonlinear input constrained systems with sampling times in the millisecond range. It allows a time and memory efficient calculation of single MPC steps and hence is well suited for the implementation on low-performance systems such as PLCs. The control performance of the PLC implementation is demonstrated for a model predictively controlled laboratory crane with nonlinear dynamics.

## 2. MODEL PREDICTIVE CONTROL

This section introduces the MPC scheme and briefly describes the real-time MPC algorithm as the basis for the PLC implementation.

### 2.1 Problem formulation

In this paper, the following OCP formulation is considered for the MPC design:

$$\min_{u(\cdot)} \quad J(u,x_k) = V(x(t_k+t)) + \int_{t_k}^{t_k+T} L(x(t),u(t))\,\mathrm{d}t \tag{1a}$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t),u(t)),\;\; x(t_k)=x_k \tag{1b}$$

$$u(t) \in [u^-,u^+],\;\; t\in[t_k,t_k+T], \tag{1c}$$

where $T>0$ denotes the prediction horizon and $x\in\mathbb{R}^n$ and $u\in\mathbb{R}^m$ are the states and controls of the system, respectively. The terminal cost $V:\mathbb{R}^n\to\mathbb{R}^0_+$ and the integral cost $L:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^0_+$ both being positive definite functions as well as the vector field $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^n$ are all assumed to be continuously differentiable in their arguments. The initial condition $x(t_k)=x_k$ denotes the measured (or observed) state of the system at sampling instance $t_k=t_0+k\Delta t$ with sampling time $\Delta t$. Condition (1c) represents the input constraints.

For further considerations it is assumed that OCP (1) possesses an optimal solution that is denoted by

$$u_k^*(t) := u^*(t;x_k),\; x_k^*(t) := x^*(t;x_k),\; t\in[t_k,t_k+T], \tag{2}$$

where the subindex $k$ indicates the current sampling instance $t_k$. Typical MPC approaches aim at computing the optimal solution (2) in each sampling instance and inject the first part of the optimal control trajectory to the system, i.e.

$$u(t) = u_k^*(t),\;\; t\in[t_k,t_k+\Delta t). \tag{3}$$

In the next sampling instance $t_{k+1}=t_k+\Delta t$, OCP (1) is solved again with the new state $x(t_{k+1})$.

### 2.2 Real-time algorithm

The algorithm that is used in this paper for solving (1) relies on the optimality conditions. To this end, the Hamiltonian [1]

$$H(x,u,\lambda) = L(x,u) + \lambda^{\mathsf{T}} f(x,u) \tag{4}$$

with the costates $\lambda\in\mathbb{R}^n$ is introduced. Given an optimal solution $(x_k^*,u_k^*)$, Pontryagin's Maximum Principle (Kirk, 1970; Berkovitz, 1974) states that there exists a costate trajectory $\lambda_k^*$ satisfying the following conditions:

$$\dot{x}_k^* = f(x_k^*,u_k^*), \qquad x_k^*(t_k)=x_k \tag{5a}$$

$$\dot{\lambda}_k^* = -H_x(x_k^*,u_k^*,\lambda_k^*),\; \lambda_k^*(t_k+T)=V_x(x_k^*(t_k+T)) \tag{5b}$$

$$u_k^* = \operatorname*{arg\,min}_{u\in[u^-,u^+]} H(x_k^*,u,\lambda_k^*),\; t\in[t_k,t_k+T], \tag{5c}$$

where $H_x := \partial H/\partial x$ and $V_x := \partial V/\partial x$ denote the partial derivatives of the Hamiltonian $H$ and the terminal cost $V$ w.r.t. the states $x$, respectively. The separated boundary conditions in (5a), (5b) are due to the OCP formulation (1) without terminal conditions.

The optimality conditions (5) can be solved by means of the (projected) gradient method (Dunn, 1996; Graichen and Käpernick, 2012). In view of (5), the following gradient steps are performed for solving OCP (1):

- Initialization of input trajectory $u_k^{(0)}(t)$.
- Gradient iterations for $j=0,\dots,N$:
  1) Forward integration of (5a) to obtain $x_k^{(j)}(t)$.
  2) Backward integration of (5b) to obtain $\lambda_k^{(j)}(t)$.
  3) Control update $u_k^{(j+1)}(t) = \psi\Big(u_k^{(j)}(t) - \alpha_k^{(j)} s_k^{(j)}(t)\Big)$ with search direction
  $$s_k^{(j)}(t) = H_u\left(x_k^{(j)}(t),u_k^{(j)}(t),\lambda_k^{(j)}(t)\right) \tag{6}$$
  and projection function
  $$\psi(u(t)) = \begin{cases} u^- & \text{if } u(t)<u^- \\ u(t) & \text{if } u(t)\in[u^-,u^+] \\ u^+ & \text{if } u(t)>u^+. \end{cases} \tag{7}$$

The control update requires the computation of a suitable step size $\alpha_k^{(j)}$ in each gradient iteration. An efficient strategy to determine the step size provides the explicit line search approach originally presented in (Barzilai and Borwein, 1988) and adapted for the optimal control case in (Käpernick and Graichen, 2013). The main idea is to minimize the difference between two consecutive control updates for the unconstrained case and with the same step size. This results in the explicit formulation

$$\alpha_k^{(j)} = \frac{\displaystyle\int_{t_k}^{t_k+T} \left(\Delta u_k^{(j)}\right)^{\mathsf{T}} \Delta s_k^{(j)}\,\mathrm{d}t}{\displaystyle\int_{t_k}^{t_k+T} \left(\Delta s_k^{(j)}\right)^{\mathsf{T}} \Delta s_k^{(j)}\,\mathrm{d}t} =: \frac{\left\langle \Delta u^{(j)},\Delta s_k^{(j)}\right\rangle}{\left\langle \Delta s_k^{(j)},\Delta s_k^{(j)}\right\rangle} \tag{8}$$

with $\Delta u_k^{(j)} := u_k^{(j)} - u_k^{(j-1)}$ and $\Delta s_k^{(j)} := s_k^{(j)} - s_k^{(j-1)}$. The approach is well suited for the PLC implementation since the computation of the step size (8) only requires to store the previous iteration of the control trajectory as well as the gradient and to perform two separate integrations.

In order to maintain real-time feasibility of the overall MPC algorithm, a fixed number of gradient iterations $N$ is performed. Hence, in contrast to applying the optimal solution $u_k^*(t)$ (cf. (3)), the control that is injected to the system is given by

$$u(t) = u_k^{(N)}(t),\;\; t\in[t_k,t_k+\Delta t). \tag{9}$$

The last iteration is additionally used in the next sampling instance to re-initialize the controls. Convergence and stability analyses regarding the projected gradient method as well as the (prematurely stopped) MPC scheme can be found in (Dunn, 1996) and (Graichen and Kugi, 2010), respectively.

## 3. PLC IMPLEMENTATION AND EXPERIMENTAL RESULTS

The presented MPC scheme is implemented on a standard PLC and used for controlling a laboratory crane. To this end, the crane setup and implementation details are introduced prior to the experimental results.

### 3.1 Laboratory crane setup

Figure 1 shows the schematics of the laboratory crane. The two-dimensional configuration basically consists of a cart that moves along a rail and a mounted rope with a load of 0.5 kg which can also be altered in length.

The states $x\in\mathbb{R}^6$ of the system are the cart position $x_1=r$, the rope length $x_3=l$ and the angle $x_5=\vartheta$ to the vertical direction as well as the corresponding velocities $x_2=\dot{r}$, $x_4=\dot{l}$ and $x_6=\dot{\vartheta}$. The cart and rope accelerations $u_1=a_C$ and $u_2=a_R$ serve as control inputs. A nonlinear

[1] In the following lines, the time argument is omitted where it is convenient to maintain readability.

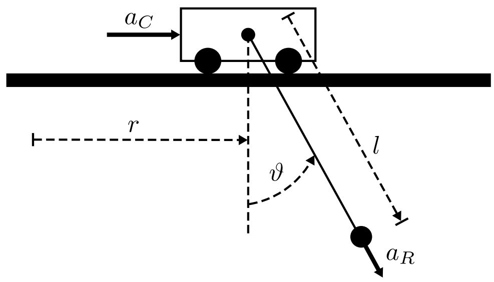


Fig. 1. Laboratory crane schematics.

model of the crane setup (Käpernick and Graichen, 2013) follows to

$$\dot{x} = f(x,u) = \begin{bmatrix} x_2 \\ u_1 \\ x_4 \\ u_2 \\ x_6 \\ -\frac{1}{x_3}\left(g \sin x_5 + u_1 \cos x_5 + 2x_4 x_6\right) \end{bmatrix}, \quad (10)$$

where $g$ denotes the gravitational constant. In addition, the dynamics (10) is subject to the input constraints

$$u_1 \in [-3, +3]\mathrm{m/s}^2, \quad u_2 \in [-3, +3]\mathrm{m/s}^2. \quad (11)$$

### *3.2 PLC and implementation details*

The real-time MPC algorithm described in Section 2.2 is implemented on a Festo PLC of type CECX-X-C1 that is equipped with a PowerPC CPU with 400 MHz and 64 MB memory. [2] Two I/O modules CECX-A-4A4E-V and CECX-C-2G2 are used for control and communication purposes. The PLC is programmed using a customized version of the development environment Codesys. [3]

The PLC implementation of the model predictive controller is performed by means of the Simulink PLC Coder. This toolbox allows generating separate function modules directly from Matlab/Simulink blocks. The code generation for the MPC module provides structured text that is subsequently integrated in the main PLC routine. A cyclic operation mode with $t_{\mathrm{cycle}} = 2\,\mathrm{ms}$ is chosen for the PLC corresponding to ensure an MPC sampling time of $\Delta t = 2\,\mathrm{ms}$. Even though such a low sampling time is not necessary in view of the crane dynamics (10) in combination with the intended transition times (cf. Section 3.3 and 4.3), it highlights the numerical performance of the gradient algorithm running on the PLC system. The entire implementation procedure is additionally illustrated in Figure 2.

### *3.3 Experimental results*

In the following experimental studies, the control task is to perform three successive setpoint changes starting from an initial state

$$x_0 = [0\,\mathrm{m}, 0\,\mathrm{m/s}, 0.3\,\mathrm{m}, 0\,\mathrm{m/s}, 0\,\mathrm{rad}, 0\,\mathrm{rad/s}]^\mathsf{T} \quad (12)$$

to the individual setpoints

$$x_{\mathrm{SP},1} = [0.5\,\mathrm{m}, 0\,\mathrm{m/s}, 0.8\,\mathrm{m}, 0\,\mathrm{m/s}, 0\,\mathrm{rad}, 0\,\mathrm{rad/s}]^\mathsf{T} \quad (13a)$$

$$x_{\mathrm{SP},2} = [-0.5\,\mathrm{m}, 0\,\mathrm{m/s}, 0.8\,\mathrm{m}, 0\,\mathrm{m/s}, 0\,\mathrm{rad}, 0\,\mathrm{rad/s}]^\mathsf{T} \quad (13b)$$

$$x_{\mathrm{SP},3} = [0\,\mathrm{m}, 0\,\mathrm{m/s}, 0.3\,\mathrm{m}, 0\,\mathrm{m/s}, 0\,\mathrm{rad}, 0\,\mathrm{rad/s}]^\mathsf{T}. \quad (13c)$$

This sequence corresponds to moving and lifting the load of the laboratory crane by 0.5 m, maneuvering the cart over the distance 1 m, and finally moving the crane back to its initial position. The terminal and integral part of the cost (1a) are chosen quadratically

$$V(x) = \|\Delta x\|_P^2, \quad L(x,u) = \|\Delta x\|_Q^2 + \|u\|_R^2 \quad (14)$$

with the weighted norm $\|\Delta x\|_P^2 := \Delta x^\mathsf{T} P \Delta x$ and $\Delta x := x - x_{\mathrm{SP},i}$ denoting the distance to the setpoints (13). The weighting matrices in (14) as well as the sampling time and prediction horizon are set to

$$P = \mathrm{diag}\,(100, 1, 100, 1, 100, 1) \quad (15a)$$

$$Q = \mathrm{diag}\,(100, 1, 100, 1, 100, 1) \quad (15b)$$

$$R = \mathrm{diag}\left(10^{-5}, 10^{-5}\right) \quad (15c)$$

and

$$\Delta t = 2\,\mathrm{ms}, \quad T = 1\,\mathrm{s}, \quad (16)$$

where the small input weights (15c) indicate an aggressive behavior of the controls. The integration of both the system dynamics (5a) and the adjoint dynamics (5b) is performed by means of the Euler method with 16 discretization points. The number of gradient iterations is $N = 2$.

Figure 3 shows the experimental results for the successive setpoint changes (12)-(13) of the laboratory crane. It can be seen that a good control performance with a transition time around 3 s - 4 s can be achieved. The second setpoint change $x_{\mathrm{SP},1} \to x_{\mathrm{SP},2}$, where the load is moved to the opposite direction, additionally reveals that the rope length is also altered by the controller to counteract the load swinging.

The computation time for a single MPC step with $N = 2$ gradient iterations on the PLC amounts approximately to 1 ms - 1.5 ms. Thus, the chosen sampling time $\Delta t = 2\,\mathrm{ms}$ is the lowest possible value for running the model predictive controller on the PLC.

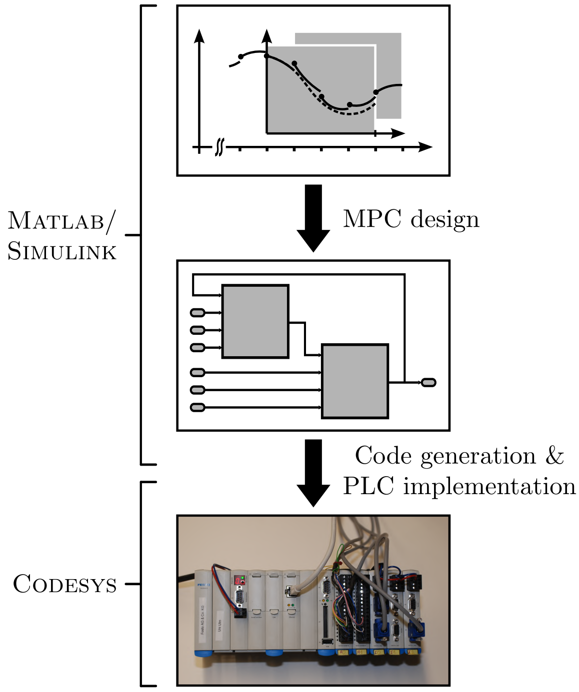


Fig. 2. PLC implementation procedure of the MPC.

## 4. MODEL PREDICTIVE TRACKING CONTROL

The experimental results in the last section revealed that a good control quality can already be achieved by

[2] See http://www.festo.com/cat/en-gb_gb/products_CECX for more information on the PLC (checked on 18 March 2014).

[3] See http://www.codesys.com/products/codesys-engineering.html for more information on Codesys (checked on 18 March 2014).

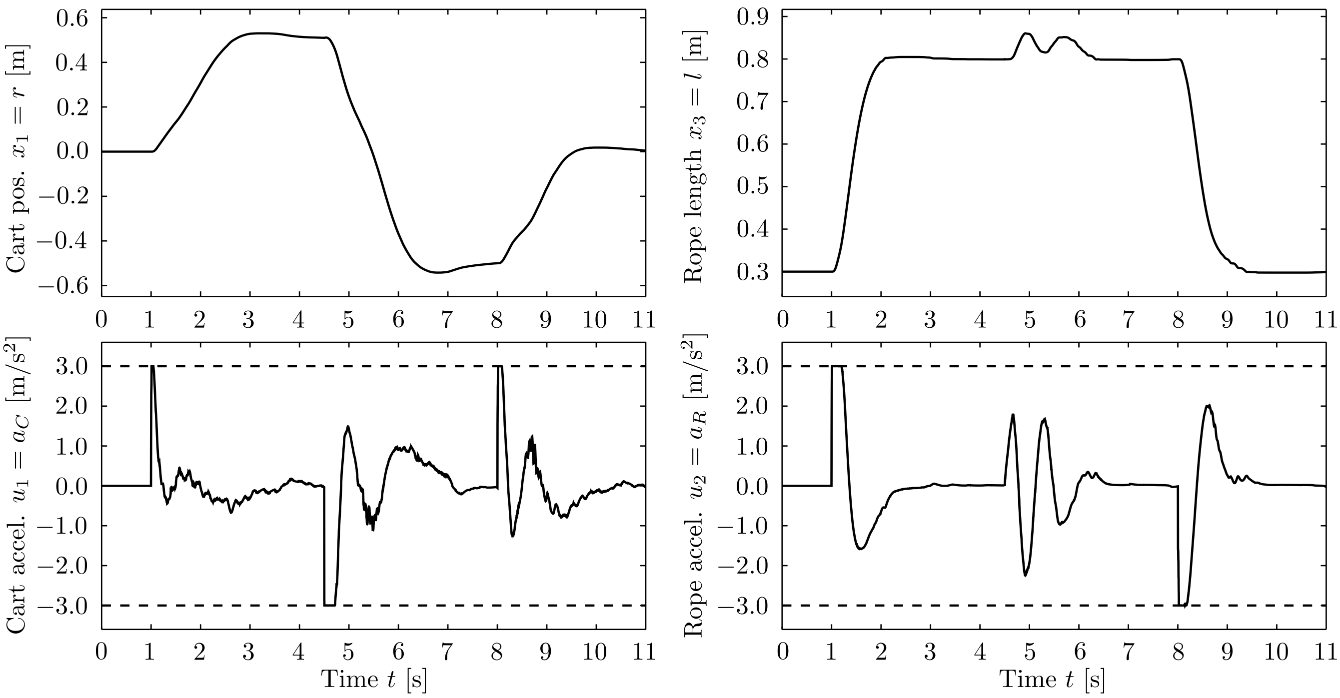

Fig. 3. Experimental results for the the model predictively controlled crane.

means of the classical MPC formulation (1). However, the performance can still be further increased using the MPC in a tracking control context within a two-degree-of-freedom (2DOF) control scheme as illustrated in Figure 4. The feedforward control in the 2DOF scheme is used to provide suitable reference trajectories $(x_{\mathrm{ref}}(t), u_{\mathrm{ref}}(t))$ for the states as well as the controls. Additionally, the feedback part stabilizes the system along the reference trajectories and compensates for model uncertainties and disturbances.

### *4.1 Flatness-based trajectory planning*

An elegant way to perform a trajectory planning is provided by the theory of differential flatness (Fliess et al., 1995; Lévine, 2009; Hagenmeyer and Zeitz, 2004). Given a nonlinear system $\dot{x} = f(x, u)$, an output $z \in \mathbb{R}^m$ of the form

$$z = \phi(x, u_1, \ldots, u_1^{(\gamma_1)}, \ldots, u_m, \ldots, u_m^{(\gamma_m)}) \tag{17}$$

is called a flat output if all states $x$ and controls $u$ can be parametrized by means of $z$ and its time derivatives, i.e.

$$x = \Psi_x\left(z_1, \ldots, z_1^{(\beta_1-1)}, \ldots, z_m, \ldots, z_m^{(\beta_m-1)}\right) \tag{18a}$$

$$u = \Psi_u\left(z_1, \ldots, z_1^{(\beta_1)}, \ldots, z_m, \ldots, z_m^{(\beta_m)}\right), \tag{18b}$$

where $\beta_1, \ldots, \beta_m$ denote the differentiation order. The flatness-based trajectory planning allows constructing a reference trajectory $z_{\mathrm{ref}}(t)$ within a finite time interval $t \in [0, \tau]$ and then deriving the related controls $u_{\mathrm{ref}}(t)$ and states $x_{\mathrm{ref}}(t)$ by means of the parametrization (18). In this regard, the computed trajectories can be used to perform a setpoint change with the finite transition time $\tau$. In addition, a suitable choice of the transition time indirectly facilitates to satisfy the constraints.

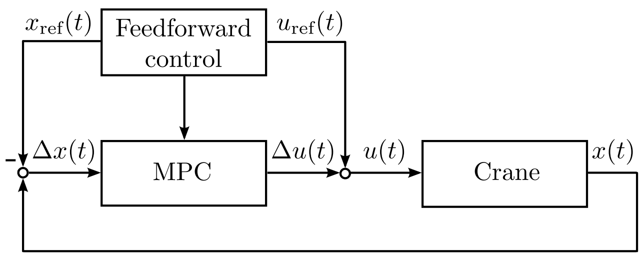


Fig. 4. Two-degrees-of-freedom control scheme.

In view of the crane dynamics (10), it can be shown that the position of the load (cf. Figure 1)

$$z_1 = r + l\sin\vartheta = x_1 + x_3 \sin x_5 \tag{19a}$$

$$z_2 = l\cos\vartheta \qquad = x_3 \cos x_5 \tag{19b}$$

is a flat output (Fliess et al., 1995). The parametrization of the states (18a) is given in the following way:

$$\begin{aligned}
x_1 &= z_1 + \frac{\ddot{z}_1 z_2}{g - \ddot{z}_2}\\
x_2 &= \frac{(g-\ddot{z}_2)\left(\ddot{z}_1\dot{z}_2 + z_1^{(3)} z_2 + \dot{z}_1(g-\ddot{z}_2)\right) + \ddot{z}_1 z_2 z_2^{(3)}}{(g-\ddot{z}_2)^2}\\
x_3 &= \begin{cases} -\arccos\left(\frac{g-\ddot{z}_2}{\sqrt{\ddot{z}_1^2+(g-\ddot{z}_2)^2}}\right) & \text{if } \ddot{z}_1 > 0\\ +\arccos\left(\frac{g-\ddot{z}_2}{\sqrt{\ddot{z}_1^2+(g-\ddot{z}_2)^2}}\right) & \text{if } \ddot{z}_1 \le 0\end{cases}\\
x_4 &= -\frac{\ddot{z}_1 z_2^{(3)} + z_1^{(3)}(g-\ddot{z}_2)}{\ddot{z}_1^2 + (g-\ddot{z}_2)^2}\\
x_5 &= \frac{z_2\sqrt{\ddot{z}_1^2 + (g-\ddot{z}_2)^2}}{g-\ddot{z}_2}\\
x_6 &= \frac{(g-\ddot{z}_2)\left(\dot{z}_2\left(\ddot{z}_1^2 + (g-\ddot{z}_2)^2\right) + \ddot{z}_1 z_1^{(3)} z_2\right) + \ddot{z}_1^2 z_2 z_2^{(3)}}{(g-\ddot{z}_2)^2\sqrt{\ddot{z}_1^2+(g-\ddot{z}_2)^2}}.
\end{aligned} \tag{20}$$

The control parametrization (18b) easily follows by differentiation of $x_2$ and $x_4$ and hence the differentiation order is $\beta_1 = \beta_2 = 4$. A reference trajectory $z_{\mathrm{ref}}(t)$ for the flat

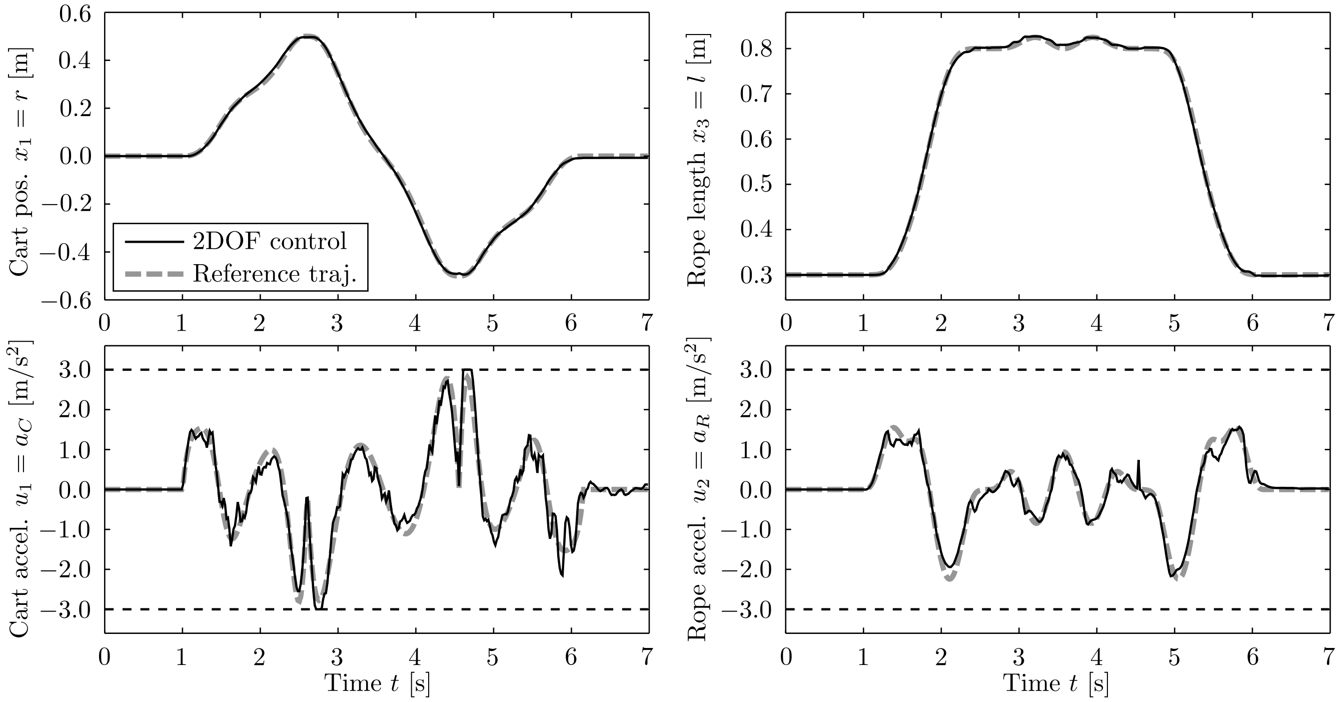


Fig. 5. Experimental results for the 2DOF control scheme with flatness-based trajectory planning.

output (19) is constructed by the polynomial

$$z_{\mathrm{ref}}(t) = z_0 + (z_\tau - z_0) \sum_{j=\sigma+1}^{2\sigma+1} c_j \left(\frac{t}{\tau}\right)^j, \quad t \in [0, \tau] \tag{21}$$

with the polynomial order $\sigma = 4$ (corresponding to the differentiation order of $z_{\mathrm{ref}}(t)$ at $t = \{0, \tau\}$) and the related coefficients

$$c_j = \frac{(-1)^{j-\sigma-1}(2\sigma+1)!}{j\sigma!(j-\sigma-1)!(2\sigma+1-j)!}\,. \tag{22}$$

Next, the reference trajectories $x_{\mathrm{ref}}(t)$ and $u_{\mathrm{ref}}(t)$ for $t \in [0, \tau]$ are determined using the parametrization (18) in combination with the constructed trajectory $z_{\mathrm{ref}}(t)$ and its time derivatives. The computed trajectories are then provided to both the MPC and the laboratory crane as illustrated in Figure 4.

### *4.2 Error dynamics and tracking formulation*

In view of the 2DOF control scheme in Figure 4, the MPC is used to counteract the deviation from the reference trajectory $x_{\mathrm{ref}}(t)$ by means of the additional control correction $\Delta u(t)$. To this end, the time-varying tracking error for the state

$$\Delta x(t) := x(t) - x_{\mathrm{ref}}(t) \tag{23}$$

is considered in the following. The resulting control that is injected to the system comprises the correction $\Delta u(t)$ and the feedforward control $u_{\mathrm{ref}}(t)$ (cf. Figure 4), i.e.

$$u(t) = u_{\mathrm{ref}}(t) + \Delta u(t). \tag{24}$$

The related error dynamics regarding the tracking error (23) and the control action (24) follows to

$$\begin{aligned}\Delta\dot{x}(t) &= \dot{x}(t) - \dot{x}_{\mathrm{ref}}(t)\\ &= f\left(\Delta x(t)+x_{\mathrm{ref}}(t), \Delta u(t)+u_{\mathrm{ref}}(t)\right)-\dot{x}_{\mathrm{ref}}(t)\\ &=: g(\Delta x(t), \Delta u(t)),\end{aligned} \tag{25}$$

where it is assumed that the state reference trajectory $x_{\mathrm{ref}}(t)$ is sufficiently smooth. The original MPC scheme (1) can then be reformulated to account for the error dynamics (25) in the following way:

$$\min_{\Delta u(\cdot)} \quad J_\Delta(\Delta u, \Delta x_k) = \frac{1}{2}\left\|\Delta x(t_k+T)\right\|^2_{P_\Delta} + \frac{1}{2}\int_{t_k}^{t_k+T} \left\|\Delta x(t)\right\|^2_{Q_\Delta} + \left\|\Delta u(t)\right\|^2_{R_\Delta}\,\mathrm{d}t \tag{26a}$$

$$\text{s.t.} \quad \Delta\dot{x}(t) = g(\Delta x(t), \Delta u(t)), \;\; \Delta x(t_k) = \Delta x_k \tag{26b}$$

$$\Delta u(t) \in \left[\Delta u^-(t), \Delta u^+(t)\right], \;\; t \in [t_k, t_k+T] \tag{26c}$$

with the initial tracking error $\Delta x_k = x_k - x_{\mathrm{ref}}(t_k)$ at sampling time $t_k$ and the time-varying input constraints $\Delta u^\pm(t) = u^\pm - u_{\mathrm{ref}}(t)$. The tracking based OCP (26) is repetitively solved at each sampling instance where the reference trajectories $x_{\mathrm{ref}}(t)$ and $u_{\mathrm{ref}}(t)$ must additionally be shifted by the sample time $\Delta t$ and provided to the MPC formulation (26).

Note that in the nominal case the crane exactly tracks the state reference $x_{\mathrm{ref}}(t)$ due to the feedforward control. Consequently, the tracking error $\Delta x(t)$ as well as the control correction $\Delta u(t)$ will be zero corresponding to a trivial optimal solution of (26) in the nominal case.

### *4.3 Experimental results for the 2DOF control scheme*

According to the successive setpoint changes (12)-(13), the initial and final values for the reference trajectory $z_{\mathrm{ref}}(t)$ of the flat output (cf. (21)) are

$$z_{0,1} = [\;\;0.0, 0.3]\mathrm{m},\; z_{\tau,1} = [\;\;0.5, 0.8]\mathrm{m},\; \tau_1 = 1.6\,\mathrm{s} \tag{27a}$$

$$z_{0,2} = [\;\;0.5, 0.8]\mathrm{m},\; z_{\tau,2} = [-0.5, 0.8]\mathrm{m},\; \tau_2 = 1.95\,\mathrm{s} \tag{27b}$$

$$z_{0,3} = [-0.5, 0.8]\mathrm{m},\; z_{\tau,3} = [\;\;0.0, 0.3]\mathrm{m},\; \tau_3 = 1.6\,\mathrm{s}, \tag{27c}$$

where the transition times $\tau_i$ are chosen to satisfy the input constraints (11). The weights in the tracking formulation (26) are adapted to

$$P_\Delta = \operatorname{diag}(10, 1, 10, 1, 10, 1) \tag{28a}$$
$$Q_\Delta = \operatorname{diag}(10, 1, 10, 1, 10, 1) \tag{28b}$$
$$R_\Delta = \operatorname{diag}\left(10^{-2}, 10^{-2}\right). \tag{28c}$$

Since the MPC has only corrective functionality in terms of the 2DOF control scheme (cf. Figure 4), the prediction horizon can be shortened to

$$T = 0.3\,\mathrm{s}. \tag{29}$$

The experimental results for the two-degrees-of-freedom control scheme with flatness-based trajectory planning are shown in Figure 5. A considerable improvement of the control performance can be observed resulting in an excellent control quality. The input constraints are also satisfied due to the choice of the transition times $\tau_i$ in combination with the time-varying input constraints (26c).

## 5. CONCLUSION

Programmable logic controllers (PLCs) are typically used in industrial automation for simple controller designs due to limited computational performance and memory. The MPC scheme discussed in this paper allows to deal with these limited hardware resources by means of an efficient gradient-based algorithm which was demonstrated by implementing and running a nonlinear model predictive controller on a standard PLC. The MPC scheme is well suited for handling nonlinear input constrained systems with sampling times in the millisecond range. The performance and applicability of the PLC implementation was illustrated for a laboratory crane setup with a sampling time of $\Delta t = 2\,\mathrm{ms}$. In addition to the classical MPC setup, a two-degrees-of-freedom control scheme with a flatness-based trajectory planning and a model predictive tracking controller was presented.

A video of the model predictively controlled laboratory crane with the two-degrees-of-freedom scheme can be found on `http://www.youtube.com/watch?v=-_yMXq4PLJo`. Moreover, the gradient-based MPC algorithm is implemented in the open source software `GRAMPC` (`GRA`dient based `MPC` – [græmp′si:]) that is available under `http://sourceforge.net/projects/grampc`. It also includes an interface to MATLAB/SIMULINK with an additional GUI in MATLAB in order to provide a convenient and interactive MPC design procedure.

Future work concerns the incorporation of state constraints. Moreover, the used MPC approach shall be more improved in terms of computational efficiency in order to allow the implementation and run on a wider range of hardware systems with limited performance and memory.

## ACKNOWLEDGEMENTS

This work was funded by the Austrian Science Fund (FWF): P21253N22. The authors also want to thank FESTO AG & Co. KG and in particular Dr. Alexander Hildebrandt and Martin Ehrle for support and providing the PLC that was used for the experiment with the laboratory crane.

## REFERENCES

Barzilai, J. and Borwein, J.M. (1988). Two-point step size gradient method. *IMA Journal of Numerical Analysis*, 8(1), 141–148.

Berkovitz, L.D. (1974). *Optimal Control Theory*. Springer, New York.

Camacho, E.F. and Bordons, C. (2003). *Model Predictive Control*. Springer, London.

DeHaan, D. and Guay, M. (2007). A real-time framework for model-predictive control of continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, 52(11), 2047–2057.

Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4), 577–585.

Dunn, J.C. (1996). On $L^2$ sufficient conditions and the gradient projection method for optimal control problems. *SIAM Journal on Control and Optimization*, 34(4), 1270–1290.

Fliess, M., Lévine, J., Martin, P., and Rouchon, P. (1995). Flatness and defect of non-linear systems: Introductory and examples. *International Journal of Control*, 61(6), 1327–1361.

Graichen, K. and Käpernick, B. (2012). A real-time gradient method for nonlinear model predictive control. In T. Zheng (ed.), *Frontiers of Model Predictive Control*, 9–28. InTech. `http://www.intechopen.com/books/frontiers-of-model-predictive-control`.

Graichen, K. and Kugi, A. (2010). Stability and incremental improvement of suboptimal MPC without terminal constraints. *IEEE Transactions on Automatic Control*, 55(11), 2576–2580.

Grüne, L. and Pannek, J. (2011). *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer, London.

Hagenmeyer, V. and Zeitz, M. (2004). Flatness-based design of linear and nonlinear feedforward controls. *at - Automatisierungstechnik*, 52(1), 3–12.

Houska, B., Ferreau, H.J., and Diehl, M. (2011). ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3), 298–312.

Huyck, B., Ferreau, H.J., Diehl, M., Brabanter, J.D., Impe, J.F.M.V., Moor, B.D., and Logist, F. (2012). Towards online model predictive control on a programmable logic controller: Practical considerations. *Mathematical Problems in Engineering*, 2012, 1–20.

Käpernick, B. and Graichen, K. (2013). Model predictive control of an overhead crane using constraint substitution. In *Proceedings of the 2013 American Control Conference*, 3979–3984.

Kirk, D.E. (1970). *Optimal Control Theory: An Introduction*. Dover Publications, Mineola.

Lévine, J. (2009). *Analysis and Control of Nonlinear Systems: A Flatness-based Approach*. Springer, Berlin.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Ohtsuka, T. (2004). A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4), 563–574.

Rauová, I., Valo, R., Kvasnica, M., and Fikar, M. (2011). Real-time model predictive control of a fan heater via PLC. In *Proccedings of the 18th International Conference on Process Control*, 388–393.

Syaichu-Rohman, A. and Sirius, R. (2011). Model predictive control implementation on a programmable logic controller for DC motor speed control. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*.

Valencia-Palomo, G. and Rossiter, J.A. (2011). Efficient suboptimal parametric solutions to predictive control for PLC applications. *Control Engineering Practice*, 19(7), 732–743.

Zavala, V.M. and Biegler, L. (2009). The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45(1), 86–93.