Copyright © 2013 IEEE

Reprinted from
Proceedings of the 2013 American Control Conference

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.

Model Predictive Control of an Overhead Crane Using Constraint Substitution

Bartosz Käpernick and Knut Graichen

Abstract—This paper describes a real-time model predictive control (MPC) scheme for an overhead crane subject to state and input constraints. The constraints are taken into account by means of a transformation technique which transforms the system dynamics with the corresponding constraints into a new unconstrained representation. This method allows to reformulate the underlying optimal control problem (OCP) of the MPC scheme into an unconstrained counterpart. The unconstrained OCP is then solved by means of a fast MPC algorithm that uses a finite number of iterations per MPC step in order to ensure real-time feasibility. Simulation as well as experimental results demonstrate the computational performance of the MPC scheme.

I. INTRODUCTION

Model predictive control (MPC) is a modern control method in which the current control update is obtained by solving an optimal control problem (OCP) online at each sampling instance. The current state is used as initial condition and the first part of the computed optimal solution is applied to the controlled system, see e.g. [1]–[3]. The main advantages of MPC are the ability to handle constraints and the consideration of general nonlinear multiple input systems.

However, a drawback of MPC is the considerable computational effort that is required to solve the constrained OCP. In particular, the consideration of state constraints increases the numerical complexity and consequently the computation time. This typically limits the use of MPC to sufficiently slow or low dimensional systems. On the other hand, input constraints can efficiently be considered, i.e. by means of a projection function used in a gradient projection method [4].

Several real-time or suboptimal MPC approaches in the literature cope with this problem from an algorithmic point of view. For instance, the procedure in [5] uses a continuation method to trace the solution of the optimality conditions over the single MPC steps based on a generalized minimum residual method (GMRES). The real-time method presented in [6] investigates a Newton-type iteration scheme that is used in [7] in combination with the ACADO toolkit. Another MPC approach in [8] uses a suboptimal MPC strategy based on a descent condition, where state and control constraints are accounted for by means of an interior point formulation.

This paper demonstrates a different approach to attack the problem of computational complexity in state constrained MPC from an analytic/algebraic perspective and illustrates this approach for an overhead crane. Following the ideas in [9], it is shown how the state and control constraints of the overhead crane can be incorporated into a new system

representation by means of saturation functions. This substitution can be seen as an analytic preprocessing step in order to reformulate the original OCP of the MPC formulation as an unconstrained OCP that can be solved with unconstrained optimization methods. The unconstrained OCP is then numerically solved by adapting the gradient based MPC algorithm in [4]. To further accelerate the computations, an alternative way is presented to compute the step size for the control update, which is based on results from static optimization [10] and is adapted for the optimal control case.

The performance and computational efficiency of the MPC controlled overhead crane is demonstrated in simulation and experimental studies, where a dSPACE DS1103 system is used for the implementation of the experimental setup. Moreover, computation times for the optimization are presented for different settings of the gradient based MPC scheme and are compared with the solution of the open-source software ACADO Toolkit [11].

II. OVERHEAD CRANE AND CONSTRAINT SUBSTITUTION

In the following, the model of the overhead crane is introduced to which the constraint transformation technique is applied. This approach leads to a new system representation as the basis for an unconstrained MPC formulation.

A. Crane dynamics and constraints

Figure 1 shows the two-dimensional configuration of an overhead crane. The states of the system are the cart position $x_1 = r$, the rope length $x_3 = l$, the angle $x_5 = \vartheta$ to the vertical direction and the corresponding velocities $x_2 = \dot{r}$, $x_4 = \dot{l}$ and $x_6 = \dot{\vartheta}$. The controls u_1 and u_2 are the acceleration of the cart a_C and the rope a_R , respectively. A nonlinear model of the crane in input affine representation is given by

$$\dot{x} = \underbrace{\begin{bmatrix} x_2 \\ 0 \\ x_4 \\ 0 \\ x_6 \\ \frac{-g \sin(x_5) + 2x_4x_6}{x_3} \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \frac{-\cos(x_5)}{x_3} \end{bmatrix}}_{g_1(x)} u_1 + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{g_2(x)} u_2 \quad (1)$$

with states $x = [x_1, \dots, x_6]^T$, controls $u = [u_1, u_2]^T$ and the gravitational constant g . In addition, the accelerations and velocities of the cart and the rope are limited to

$$x_2 \in [x_2^-, x_2^+], \quad x_4 \in [x_4^-, x_4^+], \quad (2a)$$

$$u_1 \in [u_1^-, u_1^+], \quad u_2 \in [u_2^-, u_2^+] \quad (2b)$$

This work was funded by the Austrian Science Fund (FWF): P21253N22
B. Käpernick and K. Graichen are with the Institute of Measurement, Control and Microtechnology, University of Ulm, Germany
{bartosz.kaepernick, knut.graichen}@uni-ulm.de

Fig. 1. Schematic of an overhead crane.

in order to account for control and security constraints. More details on the experimental crane setup and the constraint values are given in Section IV-A and IV-B.

B. Transformation of constraints

Typically, the consideration of constraints such as (2) increases the complexity of an MPC scheme considerably. An alternative way is to incorporate the constraints into new system dynamics following the ideas in [9].

The first step to derive a new system representation for the overhead crane is to replace the constrained states (2a) by saturation functions of the form

$$x_2 = \psi_1(\xi_1, \psi_1^\pm) \in (\psi_1^-, \psi_1^+) \quad (3a)$$

$$x_4 = \psi_2(\xi_2, \psi_2^\pm) \in (\psi_2^-, \psi_2^+) \quad (3b)$$

with new unconstrained variables $\xi_1, \xi_2 \in \mathbb{R}$ and the limits $\psi_1^\pm, \psi_2^\pm$. The saturation functions $\psi_i(\xi_i, \psi_i^\pm)$ are strictly monotonically increasing (i.e. $\psi'_i(\xi_i, \psi_i^\pm) := \partial \psi_i / \partial \xi_i > 0$) and asymptotic in the sense that $\psi_i(\xi_i, \psi_i^\pm) \rightarrow \psi_i^\pm$ for $\xi_i \rightarrow \pm\infty$, as illustrated in Figure 2. In view of the constraints (2a), the saturation limits (cf. (2)) are chosen as

$$\psi_1^\pm = x_2^\pm, \quad \psi_2^\pm = x_4^\pm. \quad (4)$$

The constrained states x_2 and x_4 are substituted by the new variables ξ_1 and ξ_2 via the saturation functions (3). To this end, the state substitutions (3) are differentiated with respect to time

$$\dot{x}_2 = u_1 = \psi'_1(\xi_1, \psi_1^\pm) \dot{\xi}_1 \quad (5a)$$

$$\dot{x}_4 = u_2 = \psi'_2(\xi_2, \psi_2^\pm) \dot{\xi}_2. \quad (5b)$$

The derivatives $\dot{\xi}_1$ and $\dot{\xi}_2$ now have to be chosen such that the input constraints (2b) are fulfilled. This can be achieved by introducing two more asymptotic saturation functions

$$\dot{\xi}_1 = \phi_1(v_1, \phi_1^\pm) \in (\phi_1^-, \phi_1^+) \quad (6a)$$

$$\dot{\xi}_2 = \phi_2(v_2, \phi_2^\pm) \in (\phi_2^-, \phi_2^+) \quad (6b)$$

with the new unconstrained inputs $v_1, v_2 \in \mathbb{R}$. Inserting (6) in (5) and dividing by the gradients $\psi'_i(\xi_i, \psi_i^\pm)$, $i = 1, 2$ allows one to reformulate $u_i^- \leq u_i \leq u_i^+$ in the following way

$$\frac{u_i^-}{\psi'_i(\xi_i, \psi_i^\pm)} \leq \phi_i(v_i, \phi_i^\pm) \leq \frac{u_i^+}{\psi'_i(\xi_i, \psi_i^\pm)}, \quad i = 1, 2. \quad (7)$$

Obviously, the input constraints (2b) can be satisfied by setting the saturation limits to

$$\phi_1^\pm(\xi_1) = \frac{u_1^\pm}{\psi'_1(\xi_1, \psi_1^\pm)}, \quad \phi_2^\pm(\xi_2) = \frac{u_2^\pm}{\psi'_2(\xi_2, \psi_2^\pm)}. \quad (8)$$

Note that the input saturation limits $\phi_i^\pm$ are state-dependent functions in contrast to the constant saturation limits (4) for the state constraints (2a).

Remark 1: The saturation limits (8) illustrate an interesting property of the input saturation functions $\phi_i(v_i, \phi_i^\pm)$. Due to the derivatives $\psi'_i(\xi_i, \psi_i^\pm)$ appearing in the denominator of (8), the saturation limits increase in magnitude when the state constraints are approached (since $\psi'_i \rightarrow 0$ when $\psi_i \rightarrow \psi_i^\pm$). Hence, with $u_i^- < 0$ and $u_i^+ > 0$, the limits $\phi_i^\pm(\xi_i) \rightarrow \pm\infty$ result in an “opening” of the saturation function $\phi_i(v_i, \phi_i^\pm)$ in the neighborhood of the state constraints and $\dot{\xi}_i = \phi_i(v_i, \phi_i^\pm)$ consequently becomes unconstrained [9], [12].

Remark 2: In the general case, the transformation approach in [9] requires the state constraints to possess a well defined vector relative degree [13], which reveals how many times the state constraints have to be differentiated until at least one control appears. For instance, the constrained variables x_2 and x_4 of the overhead crane have the vector relative degree $\{r_1, r_2\} = \{1, 1\}$.

C. Summary of transformation and new dynamics

The successive incorporation of the constraints by means of the saturation functions (3), (6) leads to a substitution of the constrained states x_2, x_4 and inputs u_1, u_2 by the unconstrained variables $\xi = [\xi_1, \xi_2]^T$ and $v = [v_1, v_2]^T$, whereas the unconstrained states x_1, x_3, x_5 remain unchanged. Hence, the overall transformation between the original variables and the new unconstrained ones is given by¹

$$\begin{aligned} x &\stackrel{(3)}{=} [x_1, \psi_1, x_3, \psi_2, x_5, x_6]^T =: h_x(\tilde{x}) \\ u &\stackrel{(5),(6)}{=} [\psi'_1 \phi_1, \psi'_2 \phi_2]^T =: h_u(\tilde{x}, v) \end{aligned} \quad (9)$$

with the new defined states $\tilde{x} := [x_1, \xi_1, x_3, \xi_2, x_5, x_6]^T$. Due to the monotonicity and asymptotic behavior of the saturation functions, the inverse relations

$$\tilde{x} = h_x^{-1}(x), \quad v = h_u^{-1}(h_x^{-1}(x), u) \quad (10)$$

exist on the open intervals of the constraints (2).

According to the transformation (9), the dynamics of the original states x_2 and x_4 are replaced by (6). Moreover, the control u and the states x are substituted by $h_u(\xi, v)$ and $h_x(\xi)$ in the remaining dynamics (1), which leads to the overall system representation with the new control v

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{\xi}_1 \\ \dot{x}_3 \\ \dot{\xi}_2 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix}}_{\dot{\tilde{x}}} = \begin{bmatrix} \psi_1(\xi_1, \psi_1^\pm) \\ \phi_1(v_1, \phi_1^\pm) \\ \psi_2(\xi_2, \psi_2^\pm) \\ \phi_2(v_2, \phi_2^\pm) \\ x_6 \\ -\frac{1}{x_3}(g \sin(x_5) + 2\psi_2(\xi_2, \psi_2^\pm)x_6 \dots \\ \dots + \psi'_1(\xi_1, \psi_1^\pm)\phi_1(v_1, \phi_1^\pm) \cos(x_5)) \end{bmatrix} =: \tilde{f}(\tilde{x}, v). \quad (11)$$

Note that due to the inverse relations (10), the new unconstrained system dynamics (11) is equivalent to the original dynamics (1) on the open intervals of the constraints (2).

¹In the following lines, the function arguments are omitted where it is convenient to maintain readability.

Fig. 2. Asymptotic saturation function $\psi_i(\xi_i, \psi_i^\pm)$ with limits $\psi_i^\pm$.

Remark 3: In the general case, the constraint transformation [9] leads to a new system in the normal form representation

$$\dot{\xi} = F_\xi(\xi, v) \quad (12a)$$

$$\dot{z} = F_z(\xi, z, v), \quad (12b)$$

where ξ and v denote the new unconstrained variables and z represents additional variables to complete the constraint transformation (9) and (10), respectively. The first part (12a) describes the constraint dynamics, whereas (12b) represents the “internal dynamics” with respect to the state constraints. For the overhead crane, the additional variables z correspond to the unconstrained states, i.e. $z = [x_1, x_3, x_5, x_6]^T$.

III. MODEL PREDICTIVE CONTROL OF THE OVERHEAD CRANE

The new unconstrained system form (11) can be used in the framework of MPC by constructing an optimal control problem (OCP) that can be solved with unconstrained optimization techniques, e.g. using the gradient method as done in this paper.

A. MPC formulation

After applying the constraint substitution from Section II, a suitable OCP formulation for the overhead crane based on the new unconstrained system dynamics (11) is

$$\min_{v(\cdot)} J(v; \tilde{x}_k) = V(\tilde{x}(t_k + T)) + \int_{t_k}^{t_k+T} l(\tilde{x}, v) dt \quad (13a)$$

$$\text{s.t. } \dot{\tilde{x}} = \tilde{f}(\tilde{x}, v), \quad \tilde{x}(t_k) = \tilde{x}_k = h_x^{-1}(x_k) \quad (13b)$$

where T denotes the prediction horizon. The positive definite integral and terminal cost function $l : \mathbb{R}^6 \times \mathbb{R}^2 \rightarrow \mathbb{R}_0^+$ and $V : \mathbb{R}^6 \rightarrow \mathbb{R}_0^+$ are directly formulated in the new variables $\tilde{x}$ and v . The initial state x_k in (13b) is the original measured state at sampling instant t_k that is transformed via the inverse relation (10).

For an optimal solution $\tilde{x}_k^*(t), v_k^*(t), t \in [t_k, t_k + T]$ of (13), the original control is computed from (9)

$$u(t) = h_u(\tilde{x}_k^*(t), v_k^*(t)), \quad t \in [t_k, t_k + \Delta t] \quad (14)$$

and is injected to the system (1) on the time interval $t \in [t_k, t_k + \Delta t]$ with the sampling time Δt . In the next sampling step $t_{k+1} = t_k + \Delta t$, OCP (13) has to be solved again with the new state measurement (or estimate) x_{k+1} .

Stability results for MPC formulations without terminal conditions are presented in the literature by several authors. Thereby, the different stability approaches rely on a sufficiently long prediction horizon T [14] or the construction of

a suitable Control Lyapunov Function (CLF) [15]. However, there also exist results where a CLF is not necessary [16].

Remark 4: The saturation function approach has some similarity to interior barrier functions [17], as it is usually used in static optimization [18]. However, special care has to be taken that iterates during the numerical solution of the OCP do not violate the constraints when a barrier function is directly included into the cost. This problem is circumvented by incorporating the constraints into the dynamics.

B. Gradient algorithm

The unconstrained OCP (13) can be treated in the calculus of variations. By defining the Hamiltonian

$$H(\tilde{x}, \lambda, v) = l(\tilde{x}, v) + \lambda^T \tilde{f}(\tilde{x}, v) \quad (15)$$

with the adjoint state vector $\lambda \in \mathbb{R}^6$, the first-order optimality conditions are

$$\dot{\tilde{x}} = \tilde{f}(\tilde{x}, v), \quad \tilde{x}(t_k) = \tilde{x}_k \quad (16a)$$

$$\dot{\lambda} = -H_{\tilde{x}}(\tilde{x}, \lambda, v), \quad \lambda(t_k + T) = \tilde{V}_{\tilde{x}}(\tilde{x}(t_k + T)) \quad (16b)$$

$$0 = H_v(\tilde{x}, \lambda, v), \quad (16c)$$

where $H_{\tilde{x}}, H_v$ and $V_{\tilde{x}}$ denote the partial derivatives of $H(\tilde{x}, \lambda, v)$ and $V(\tilde{x})$ with respect to $\tilde{x}$ and v . The separated boundary conditions in (16a), (16b) are due to the OCP formulation (13) without terminal constraints.

The optimality conditions (16) can e.g. be solved by means of the well-known gradient method in optimal control [4], [19], [20]. Based on an initial control trajectory $v_k^{(0)}(t), t \in [t_k, t_k + T]$, a gradient step j consists of the following computations:

- integrate (16a) in forward time to obtain $x_k^{(j)}(t)$
- integrate (16b) in backward time to obtain $\lambda_k^{(j)}(t)$
- compute the new control trajectory with step size α :

$$v_k^{(j+1)}(t) = v_k^{(j)}(t) - \alpha H_v(\tilde{x}_k^{(j)}(t), \lambda_k^{(j)}(t), v_k^{(j)}(t)). \quad (17)$$

In order to use the gradient algorithm within a real-time MPC implementation, the algorithm is stopped after N iterations and the last iterates substitute the optimal feedback (14)

$$u(t) = h_u(\tilde{x}_k^{(N)}(t), v_k^{(N)}(t)), \quad t \in [t_k, t_k + \Delta t]. \quad (18)$$

In the next MPC step $k+1$, the suboptimal control trajectory is used as initialization $v_{k+1}^{(0)}(t) = v_k^{(N)}(t)$ for the gradient algorithm.

The gradient algorithm with a fixed number of iteration steps N is a suboptimal approach. However, the results in [21] show for a MPC formulation without state constraints how exponential stability of the closed-loop system as well as exponential decay of the suboptimality can be guaranteed if the iteration number N satisfies a lower bound.

C. Step size calculation

One way to compute the step size α for each gradient iteration is an adaptive line search [4], where a polynomial approximation of the cost is used and an underlying search interval adaption is performed.

An alternative way to determine the step size adapts the ideas in [10]. To this end, the control update for the

j -th gradient iteration is considered, i.e. (cf. (17) without arguments)

$$v_k^{(j+1)} = v_k^{(j)} - \alpha H_v^{(j)} \quad (19)$$

with the gradient $H_v^{(j)} := H_v(\tilde{x}_k^{(j)}(t), \lambda_k^{(j)}(t), v_k^{(j)}(t))$. Using this rule, the difference between consecutive controls can be computed as

$$\underbrace{v_k^{(j+1)} - v_k^{(j)}}_{=:\Delta v_k^{(j+1)}} = \underbrace{v_k^{(j)} - v_k^{(j-1)}}_{=:\Delta v_k^{(j)}} - \underbrace{\alpha (H_v^{(j)} - H_v^{(j-1)})}_{=:\Delta H_v^{(j)}}. \quad (20)$$

The idea is now to minimize the norm $\|\Delta v_k^{(j+1)}\|_{L_2[0,T]}^2$ with respect to α , i.e.

$$\alpha = \arg \min_{\alpha > 0} \|\Delta v_k^{(j)} - \alpha \Delta H_v^{(j)}\|_{L_2[0,T]}^2 \quad (21)$$

with $\|z\|_{L_2[0,T]}^2 = \langle z, z \rangle := \int_0^T z^\top(t) z(t) dt$. The motivation for this choice is the secant equation in quasi-Newton methods, see [10] for more details. The solution of the minimization (21) yields the step size

$$\alpha = \frac{\langle \Delta v_k^{(j)}, \Delta H_v^{(j)} \rangle}{\langle \Delta H_v^{(j)}, \Delta H_v^{(j)} \rangle}. \quad (22)$$

Another way to compute an appropriate α for the control update (19) can be achieved by minimizing $\|\alpha \Delta v_k^{(j)} - \Delta H_v^{(j)}\|_{L_2[0,T]}^2$, which leads to the corresponding solution

$$\alpha = \frac{\langle \Delta v_k^{(j)}, \Delta v_k^{(j)} \rangle}{\langle \Delta v_k^{(j)}, \Delta H_v^{(j)} \rangle}. \quad (23)$$

Both (22) and (23) can be used for the computation of the step size α . However, simulations have shown better results by applying (23), which is the reason for using this rule in the remainder of the paper. Moreover, in order to reduce the implementation effort and assure that the step size is not too large, the integrations for computing (23) are performed in a simple Euler forward integration scheme and the step size is additionally bounded by an upper value, i.e. $\alpha \leq \alpha_{max}$.

IV. SIMULATION AND EXPERIMENTAL RESULTS FOR THE OVERHEAD CRANE

The MPC scheme based on the constraint substitution is used for controlling the overhead crane (1). The simulation as well as experimental results in this section demonstrate the performance and the computational efficiency of the presented approach.

A. Simulation results

The control task for the simulation studies is to stabilize a desired setpoint of the overhead crane

$$\begin{aligned} x_{SP} &= [0.5 \text{ m}, 0 \frac{\text{m}}{\text{s}}, 0.8 \text{ m}, 0 \frac{\text{m}}{\text{s}}, 0 \text{ rad}, 0 \frac{\text{rad}}{\text{s}}]^\top \\ u_{SP} &= [0 \frac{\text{m}}{\text{s}^2}, 0 \frac{\text{m}}{\text{s}^2}]^\top \end{aligned} \quad (24)$$

starting from the initial state

$$x_0 = [-0.5 \text{ m}, 0 \frac{\text{m}}{\text{s}}, 0.3 \text{ m}, 0 \frac{\text{m}}{\text{s}}, 0 \text{ rad}, 0 \frac{\text{rad}}{\text{s}}]^\top. \quad (25)$$

This maneuver corresponds to moving the load of the overhead crane by 1 m and to lift it by 0.5 m. The limits for the state and input constraints (2) are set to the values

$$x_2^\pm = \pm 0.3 \frac{\text{m}}{\text{s}}, \quad x_4^\pm = \pm 0.3 \frac{\text{m}}{\text{s}}, \quad (26a)$$

$$u_1^\pm = \pm 1.0 \frac{\text{m}}{\text{s}^2}, \quad u_2^\pm = \pm 1.0 \frac{\text{m}}{\text{s}^2}. \quad (26b)$$

The integral and terminal cost function in the MPC formulation (13) are chosen quadratically, i.e.

$$\begin{aligned} V(\tilde{x}) &= \frac{1}{2} \Delta \tilde{x}^\top P \Delta \tilde{x} \\ l(\tilde{x}, v) &= \frac{1}{2} (\Delta \tilde{x}^\top Q \Delta \tilde{x} + \Delta v^\top R \Delta v) \end{aligned} \quad (27)$$

with the positive definite matrices

$$P = \text{diag}(1, 1, 1, 1, 1, 1) \quad (28a)$$

$$Q = \text{diag}(0.01, 0.01, 10, 100, 1, 1) \quad (28b)$$

$$R = \text{diag}(0.01, 0.01). \quad (28c)$$

The variables

$$\Delta \tilde{x} = \tilde{x} - h_x^{-1}(x_{SP}), \quad \Delta v = v - h_u^{-1}(h_x^{-1}(x_{SP}), u_{SP})$$

denote the distance to the setpoint (24) via the inverse relations (10). The sampling time and the prediction horizon of the MPC scheme are set to

$$\Delta t = 1 \text{ ms}, \quad T = 1 \text{ s}. \quad (29)$$

The saturation functions that are used to construct the new system representation (11) are

$$\psi_i(\xi_i, \psi_i^\pm) = \psi_i^+ - \frac{\psi_i^+ - \psi_i^-}{1 + \exp\left(\frac{4\xi_i}{\psi_i^+ - \psi_i^-}\right)} \quad (30a)$$

$$\phi_i(v_i, \phi_i^\pm) = \phi_i^+ - \frac{\phi_i^+ - \phi_i^-}{1 + \exp\left(\frac{4v_i}{\phi_i^+ - \phi_i^-}\right)}, \quad i = 1, 2 \quad (30b)$$

with the normalized slope $\psi_i'(0, \psi_i^\pm) = 1$ and $\phi_i'(0, \phi_i^\pm) = 1$. Another option to construct the saturation functions are, for instance, tanh-functions.

The gradient method for the MPC scheme is a tailored C implementation following the description in [4]. The optimality conditions (16) are computed with MATHEMATICA and are exported as optimized C code. The numerical integrations of the canonical equations (16a) and (16b) are performed using an Euler forward integration scheme with 30 discretization points. For the computation of α , the adaptive step size [4] as well as the explicit step size (23) from Section III-C are both implemented for the sake of comparison.

Figure 3 shows the simulation results for $N = 2$ gradient iterations per MPC step. It can be seen that the state and input constraints (26) are satisfied and that already two gradient iterations lead to a good control performance with an exponentially decaying cost value. It can also be observed that the trajectories resulting from the MPC scheme with the explicit step size (23) provide nearly the same performance as the adaptive step size approach [4].

Table I additionally shows the average computation times that are required by the gradient method in each MPC step for the values of N as well as for both approaches of step size computation. The simulations were performed with MATLAB 2011a (64-bit) and Windows 7 (64-bit) on an Intel Core i5

Fig. 3. Simulation results for the overhead crane with $N = 2$ gradient iterations per MPC step and for the adaptive step size [4] as well as the explicit step size (23).

TABLE I
CPU TIMES FOR THE MPC (OVERHEAD CRANE).

Gradient iterations N per MPC step	adaptive step size [4]	explicit step size (23)
2	107	67
4	191	112
6	278	157
8	371	206
10	449	251

CPU with 2.67 GHz and 4 GB memory, where only one core of the CPU was used for the computations. In all cases, the CPU time is well below the sampling time $\Delta t = 1$ ms.

Finally, the results of the presented MPC approach were compared with the free open-source software ACADO-Toolkit [11]. The obtained solution with the implemented code generation module [7] is similar to the simulation results in Figure 3. Furthermore, the achieved computation time of ACADO was about $380\mu\text{s}$, which confirms that the MPC scheme presented in this paper is competitive and well suited for the control of nonlinear constrained systems with very short sampling intervals.

B. Experimental results

The MPC framework with the constraint substitution is verified for an experimental setup of an overhead crane. The cart and the change of the rope length are driven by two synchronous motors. Furthermore, cascaded velocity controllers for the cart and the rope length are used. The cart position r , the rope length l and the angle θ are measured by means of incremental encoders, whereas the corresponding velocities are determined by an extended Kalman filter.

The MPC algorithm and the underlying velocity controllers are implemented on a dSPACE DS1103 system with a PPC 750 GX CPU with 1 GHz and 96 MB memory. The number of gradient iterations per MPC step is set to $N = 2$. Moreover, the sampling time, the prediction horizon, the

constraints, the weighting matrices as well as the setpoint maneuver are taken from IV-A. Due to the short sampling time, the injected control is implemented in a sample-and-hold fashion.

The experimental results are shown in Figure 4. The trajectories demonstrate that the constraints (26) are satisfied and that two gradient iterations per MPC step are already sufficient to achieve excellent control performance. The computation time for the MPC update (including measurement and Kalman filtering) on the dSPACE platform is also given in Figure 4 and the average value amounts to $508\mu\text{s}$ with a maximum computation time of $565\mu\text{s}$, which is well below the sampling time $\Delta t = 1$ ms.

The comparison of Figure 3 and 4 shows good correspondence between the simulation and the experimental results. In the trajectories of the cart and rope velocities, a steady state error can be observed in the vicinity of active state constraints. This error arises due to the use of the cascaded velocity controllers, which consist of simple proportional controls with high gains.

Remark 5: In the case of noisy state measurements or disturbances, the determined states might exceed the saturation limits and thus result in undefined inverse relations (10). To this end, the determined states are internally monitored in order to maintain a well defined coordinate change (10). Alternatively, the MPC can be used as feedforward control to provide a nominal input trajectory where the predicted states are used to determine new initial conditions.

V. CONCLUSION

This paper described a real-time MPC scheme applied to an overhead crane that consists of a constraint substitution prior to the numerical solution. In this regard, the transformation into an unconstrained OCP represents an analytical preprocessing step that can be performed offline with computer algebra software such as MATHEMATICA.

The OCP formulation based on the unconstrained system dynamics of the crane can be solved by unconstrained

Fig. 4. Experimental results for the overhead crane with $N = 2$ gradient iterations per MPC step and for the explicit step size (23).

optimization methods. For instance, the gradient method used in the MPC algorithm is easy to implement and efficient in terms of memory consumption and computation time. An efficient way to compute the step size for the control update and a finite number of gradient iterations were used in each MPC step for a real-time implementation. It turned out that $N \geq 2$ iterations were sufficient for a good control performance of the overhead crane,

The performance and computational efficiency of the model predictive controller for the overhead crane was compared with the solution of the ACADO Toolkit and was demonstrated in simulation and experimental studies, respectively. The results showed that all input and state constraints were satisfied in combination with a good control performance. Moreover, very short computation times in the case of the overhead crane were achieved. It is obvious, that the computation time depends on the system dynamics as well as on the complexity of the constraints. The incorporation of constraints into the dynamics is a nonlinear transformation and might lead to more nonlinear relations. However, the resulting unconstrained problem can efficiently be solved via the presented gradient method which makes the approach suitable for controlling nonlinear constrained systems in the millisecond range.

An interactive Java applet of the model predictively controlled overhead crane can be downloaded from the institutes website <http://www.uni-ulm.de/en/in/mrm/research/control-and-optimization/model-predictive-control.html>.

REFERENCES

- [1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [2] E. Camacho and C. Bordons, *Model Predictive Control*. London: Springer, 2003.
- [3] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control – Theory and Algorithms*. London: Springer, 2011.
- [4] K. Graichen and B. Käpernick, "A real-time gradient method for nonlinear model predictive control," in *Frontiers of Model Predictive Control*, T. Zheng, Ed. InTech, 2012, pp. 9–28, <http://www.intechopen.com/articles/show/title/a-real-time-gradient-method-for-nonlinear-model-predictive-control>.
- [5] T. Ohtsuka, "A continuation/GMRES method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563 – 574, 2004.
- [6] M. Diehl, R. Findeisen, F. Allgöwer, H. Bock, and J. Schlöder, "Nominal stability of real-time iteration scheme for nonlinear model predictive control," *IEEE Proceedings - Control Theory and Applications*, vol. 152, no. 3, pp. 296 – 308, 2005.
- [7] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279 – 2285, 2011.
- [8] D. DeHaan and M. Guay, "A real-time framework for model-predictive control of continuous-time nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 11, pp. 2047–2057, 2007.
- [9] 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, 2009.
- [10] J. Barzilai and J. M. Borwein, "Two-point step size gradient method," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.
- [11] B. Houska, H. Ferreau, and M. Diehl, "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [12] K. Graichen, N. Petit, and A. Kugi, "Transformation of optimal control problems with a state constraint avoiding interior boundary conditions," in *CDC 2008 47th IEEE Conference on Decision and Control*, December 2008, pp. 913 – 920.
- [13] A. Isidori, *Nonlinear Control Systems*, 3rd ed. Berlin: Springer, 1995.
- [14] A. Jadbabaie, J. A. Primbs, and J. E. Hauser, "Unconstrained receding horizon control with no terminal cost," in *Proceedings of the American Control Conference*, 2001.
- [15] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.
- [16] M. Reble and F. Allgöwer, "Unconstrained model predictive control and suboptimality estimates for nonlinear continuous-time systems," *Automatica*, vol. 48, no. 8, pp. 1812–1817, 2012.
- [17] J. Bonnans and T. Guilbaud, "Using logarithmic penalties in the shooting algorithm for optimal control problems," *Optimal Control Applications and Methods*, vol. 24, no. 5, pp. 257–278, 2003.
- [18] A. Forsgren, P. E. Gill, and M. H. Wright, "Interior methods for nonlinear optimization," *SIAM Review*, vol. 44, no. 4, pp. 525–597, 2002.
- [19] J. Dunn, "On l^2 conditions and the gradient projection method for optimal control problems," *SIAM Journal on Control and Optimization*, vol. 34, no. 4, pp. 1270–1290, 1996.
- [20] M. Nikol'skii, "Convergence of the gradient projection method in optimal control problems," *Computational Mathematics and Modeling*, vol. 18, no. 2, pp. 148–156, 2007.
- [21] K. Graichen and A. Kugi, "Stability and incremental improvement of suboptimal mpc without terminal constraints," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2576–2580, 2010.