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

Experimental Validation of Combined Nonlinear Optimal Control and Estimation of an Overhead Crane ⋆

Frederik Debrouwere * **Milan Vukov** ** **Rien Quirynen** **
Moritz Diehl **,*** **Jan Swevers** *

* *Division PMA, Department of Mechanical Engineering, KU Leuven, Belgium, e-mail:* <firstname>.<lastname>@mech.kuleuven.be
** *Electrical Engineering Department (ESAT-STADIUS), KU Leuven, Belgium, e-mail:* <firstname>.<lastname>@esat.kuleuven.be
*** *University of Freiburg (IMTEK), Freiburg, Germany*

Abstract: This paper validates the combination of nonlinear model predictive control and moving horizon estimation to optimally control an overhead crane. Real-time implementation of this combined optimal control and estimation approach with execution times far below the sampling time was realized through the use of automatic code generation. Besides experiments that reflect good point-to-point performance, the approach showed to be good in disturbance rejection as well as in servo-tracking.

1. INTRODUCTION

Recently developed fast solvers for convex as well as non-convex optimization enabled application of nonlinear model predictive control (NMPC) and moving horizon estimation (NMHE) to mechatronic systems. Some of the existing codes for convex optimization include an active-set method based on aquadratic programming (QP) solver qpOASES [Ferreau et al., 2008], auto-generated interior point codes FORCES and CVXGEN [Domahidi et al., 2012][Mattingley and Boyd, 2009], a dual Newton strategy solver qpDUNES [Frasch et al., 2013], and an auto-generated fast-gradient solver Fiordos [Richter et al., 2011]. In the area of NMPC, existing tools for fast NMPC and NMHE include AutoGenU [Seguchi and Ohtsuka, 2003] and the ACADO code-generation tool (CGT) [Houska et al., 2011b]. The latest extenstions to the ACADO CGT support optimal control problems (OCPs) that can include nonlinear measurement/reference function as well as nonlinear path and point constraints. Moreover, the tool support export of tailored solvers for NMHE.

In the previous work by the authors [Vukov et al., 2012] the validation of the NMPC on an overhead crane has been presented. At the time, constrained by the feature set of the ACADO CGT, a rather simple OCP formulation was possible that used linear reference functions in the least-square objective. Employing explicit integrators was the only option, leading to simplification of the stiff model in order to make the whole control scheme real-time feasible and to avoid expensive integration resulting from a huge number of integration steps. Furthermore, there was no module for estimation, thus finite differences were used to estimate certain states such as cart, cable and swinging velocities. In this paper, we present a combined NMPC and NMHE approach to control the overhead crane using the latest features from the ACADO code-generaion tool enabling us to consider the full non-linear crane model for estimation and control (dynamical model, measurement model and control output model). One recent application of the gain-scheduled NMPC for an overhead crane has been reported in [Schindele and Aschemann, 2011].

1.1 Nonlinear MPC problem formulation

For the purpose of this paper, we use the following optimal control problem (OCP) formulation:

$$\min_{\substack{x_0,\ldots,x_N \\ u_0,\ldots,u_{N-1}}} \quad \sum_{k=0}^{N_c-1} ||h_r(x_k,u_k)-\tilde{r}_k||_R^2 + |||h_{r,N}(x_N)-\tilde{r}_N||_{R_N}^2 \tag{1a}$$

$$\text{s.t.} \quad x_0 = \hat{x}_0 \tag{1b}$$

$$x_{k+1} = F(x_k,u_k), \text{ for } k=0,\ldots,N-1, \tag{1c}$$

$$x_k^{lo} \le x_k \le x_k^{up}, \text{ for } k=0,\ldots,N, \tag{1d}$$

$$u_k^{lo} \le u_k \le u_k^{up}, \text{ for } k=0,\ldots,N-1, \tag{1e}$$

where $x \in \mathbb{R}^{n_x}$ denotes the vector of differential states, $u \in \mathbb{R}^{n_u}$ the vector of control inputs. Discretized system dynamics is represented by the function F. The current state estimate provided by an estimator is denoted $\hat{x}_0 \in \mathbb{R}^{n_x}$. The controlled system output is captured with reference functions in (1a): $h_r \in \mathbb{R}^{n_r}$ and $h_{r,N} \in \mathbb{R}^{n_{r,N}}$,

⋆ This research was supported by Research Council KUL: PFV/10/002 Optimization in Engineering Center OPTEC, GOA/10/09 MaNet and GOA/10/11 Global real- time optimal control of autonomous robots and mechatronic systems. Flemish Government: IOF/KP/SCORES4CHEM, FWO: PhD/postdoc grants and projects: G.0320.08 (convex MPC), G.0377.09 (Mechatronics MPC); IWT: PhD Grants, projects: SBO LeCoPro; Belgian Federal Science Policy Office: IUAP P7 (DYSCO, Dynamical systems, control and optimization, 2012-2017); EU: FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), Eurostars SMART, ACCM.

and the corresponding weighting matrices are $R \in \mathbb{R}^{n_r \times n_r}$ and $R_N \in \mathbb{R}^{n_{r,N} \times n_{r,N}}$. Variables $\tilde{r}_k \in \mathbb{R}^{n_r}$ and $\tilde{r}_N \in \mathbb{R}^{n_{r,N}}$ denote time-varying references. Within this formulation we use only simple box constraint on the controls and on the states: $u^{\text{lo}} \leq u^{\text{up}} \in \mathbb{R}^{n_u}$ and $x^{\text{lo}} \leq x^{\text{up}} \in \mathbb{R}^{n_x}$ in (1e) and (1d) respectively. Finally, the number of control intervals is denoted as N_{c}.

1.2 Nonlinear MHE problem formulation

In a similar fashion, the moving horizon estimation (MHE) problem reads:

$$\min_{\substack{x_{-N_{\text{e}}},\ldots,x_0 \\ u_{-N_{\text{e}}},\ldots,u_1}} \quad \sum_{k=-N_{\text{e}}}^{1} ||h_y(x_k, u_k) - \tilde{y}_k||_S^2 + |||h_{y,0}(x_0) - \tilde{y}_0||_{S_0}^2 \tag{2a}$$

$$\text{s.t.} \qquad x_{k+1} = F(x_k, u_k), \text{ for } k = -N_{\text{e}}, \ldots, 1, \tag{2b}$$

where the number of estimation intervals is N_{e}. Measurement functions are denoted with $h_y \in \mathbb{R}^{n_y}$ and $h_{y,0} \in \mathbb{R}^{n_{y,0}}$ and the corresponding weighting matrices are $S \in \mathbb{R}^{n_y \times n_y}$ and $S_0 \in \mathbb{R}^{n_{y,0} \times n_{y,0}}$.

We choose to have the same dynamic model for both the controller (1) and the estimator (2). The main motivation for this approach is that we want the estimator to capture the same dynamics we want to control using NMPC.

1.3 Contributions and Overview

The main contribution of the paper is the validation of the combined NMHE and NMPC approach to control an overhead crane. We show that the proposed approach is real-time feasible and that the computation times are far below the chosen sampling time. Control performance is validated through a set of experiments: point-to-point motion, disturbance rejection and servo-tracking.

The paper is organized as follows. Section 2 present the so called real-time iterations approach to solve an OCP in real-time. In Section 3 we briefly present a tool for exporting the customized solvers for optimal control and it's properties. The experimental setup, model equations as well as OCP formulations are detailed in Section 4. Experimental results are presented in Section 5. Section 6 concludes the paper.

2. REAL-TIME ITERATION SCHEME

The main idea behind the real-time iteration (RTI) scheme [Diehl, 2001] is to employ multiple shooting discretization [Bock and Plitt, 1984] together with numerical integration and a piece-wise constant control parameterization. The structured nonlinear program that comes as a result of this procedure can be efficiently treated with a sequential quadratic programming (SQP) methods. As the objective consists of a least squares tracking term, it is reasonable to employ a Gauss-Newton method to approximate the Hessian matrices [Bock, 1983]. In order to ensure real-time feasibility of the algorithm, only one SQP iteration is performed per sapling time step. Consequently, OCPs (1) and (2) are solved only approximately. The RTI scheme [Diehl et al., 2002] uses the control and state variables of the previous optimization run, possibly after a shift, as new linearization point, and performs only one Newton-type iteration per sampling instant. Due to the fact that the initial value enters the problem linearly, it can be shown to deliver a generalized tangential predictor to perturbations, and nominal convergence of the resulting NMPC loop can be proven. We refer to [Diehl et al., 2009] and the references therein for a detailed survey on the RTI and other algorithms for nonlinear MPC.

Due to the fact that the state estimate $\hat{x}_0$ enters the optimization problem (1) only via the affine constraint (1b), the model can be integrated and sensitivities can be generated without knowing the current state estimate. This part of the algorithm is called *the preparation phase*. Once the feedback signal is obtained, we only need to solve the underlying quadratic problem (QP) – *the feedback phase*. After the QP is solved, we take the solution u_0^* and send it to the process. In the context of the MHE, cf. (2), we can do model integration and generate sensitivities without knowing the current measurement $\tilde{y}_0$. As in the previous case, as soon as y_0 is available we have to solve (2) in order to obtain the current state estimate $\hat{x}_0^*$ which is sent to the controller.

From the implementation point of view, preparation phases of the NMPC and NMHE can be executed in parallel. Having in mind that preparation phase(s) typically consume more than 80% of the execution time of an RTI step, one can reduce the sampling time – if needed.

3. AUTOMATIC CODE GENERATION

In order to make the RTI scheme real-time feasible, we utilize the automatic code-generation approach. The `ACADO Code Generation Tool` is part of the open-source software package `ACADO Toolkit` [Houska et al., 2011a] for automatic control and dynamic optimization [Houska et al., 2011b]. The tool generates self-contained ANSI-C compliant code for NMPC and MHE [Ferreau et al., 2012], which can be deployed on any platform supporting the standard C library. The tool implements the algorithmic ideas based on the RTI scheme. The user interface allows one to specify nonlinear dynamic model equations as well as objective and constraint functions. Before a solver gets exporter, problem structure and dimensions are exploited together with sparsity patterns to remove all unnecessary computations and remove any need for dynamic memory allocation.

In order to prevent prohibitively long execution times while integrating stiff systems and to tackle more challenging nonlinear dynamical systems represented by differential algebraic equations DAE, support for implicit integrators for ODEs and (DAEs) has been added [Quirynen et al., 2012]. One of the latest extension of the tool makes it possible to solve long horizon OCPs employing an interior point QP solver [Vukov et al., 2013].

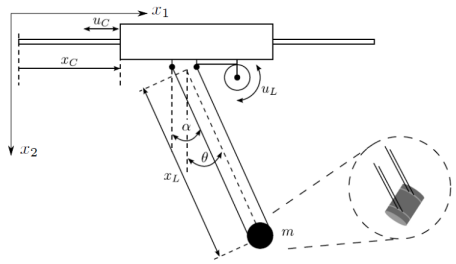

Fig. 1. Schematic of the overhead crane.

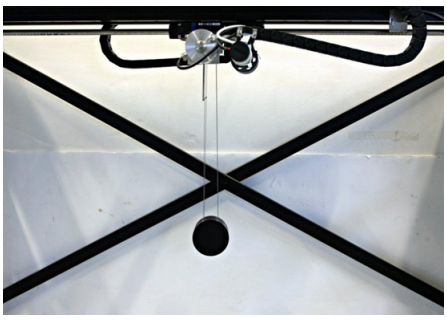

Fig. 2. Picture of the actual overhead crane used in the experiments.

4. EXPERIMENTAL SETUP AND DYNAMIC MODEL

4.1 Experimental Setup

A schematic representation of the considered overhead crane is given in Fig. 1, while Fig. 2 shows a picture of the actual crane used in the experiments. The pendulum consists of a cylindrical load hanging on two parallel cables. A cart can position the load in the x_1 direction while a winching mechanism can position the load in the x_2 direction. Reliable sensors (encoders) are available to measure the cart position x_C, cable length x_L and cable angle α (see Figure 1). A more detailed description of the system can be found in [Vukov et al., 2012]. The cart and hoisting motor are controlled using an internal velocity controller. The system inputs are the voltages u_C and u_L representing setpoints for the respective internal velocity controller. The control software is implemented using the OROCOS Toolchain [Bruyninckx et al., 2003] and runs on a PC with an Intel Xeon 2.53 GHz quad core processor, 12 GB RAM memory, and a preemptive Linux kernel as operating system. The sampling frequency for the state estimation and feedback controller is fixed to 100 Hz.

4.2 Dynamic Model

This Section briefly reviews the dynamic model of the set-up. Detailed information on the dynamic modelling can be found in [Vukov et al., 2012]. Second order models have been identified in the frequency domain for the input-output relation for both the cart mechanism (u_C to x_C) and the winching mechanism (u_L to x_L).

$$G_\mathrm{C} = \frac{A_\mathrm{C}}{s(\tau_\mathrm{C} s + 1)} \text{ and } G_\mathrm{L} = \frac{A_\mathrm{L}}{s(\tau_\mathrm{L} s + 1)}$$

where $\tau_\mathrm{C} = 0.01279, A_\mathrm{C} = 0.04742, \tau_\mathrm{L} = 0.02470$ and $A_\mathrm{L} = 0.03409$. These models and the equation of motion of the variable cable length overhead crane [Vukov et al., 2012] result in the following nonlinear ordinary differential equations:

$$\begin{aligned}
\dot{x}_\mathrm{C} = v_\mathrm{C}, \;\; \dot{v}_\mathrm{C} &= -\tau_\mathrm{C}^{-1}\left(v_\mathrm{C} - A_\mathrm{C} u_\mathrm{C}\right) \\
\dot{x}_\mathrm{L} = v_\mathrm{L}, \;\; \dot{v}_\mathrm{L} &= -\tau_\mathrm{L}^{-1}\left(v_\mathrm{L} - A_\mathrm{L} u_\mathrm{L}\right) \\
\dot{\theta} = \omega, \;\; \dot{\omega} &= -x_\mathrm{L}^{-1}\left(A_\mathrm{C} u_\mathrm{CR}\cos(\theta) + g\sin(\theta) + 2 v_\mathrm{L}\omega\right) \\
\dot{u}_\mathrm{C} = u_\mathrm{CR}, \;\; \dot{u}_\mathrm{L} &= u_\mathrm{LR}
\end{aligned}$$

These equations are summarized as $F(x,u)$ where the state $x = (x_\mathrm{C}, v_\mathrm{C}, x_\mathrm{L}, v_\mathrm{L}, \theta, \omega, u_\mathrm{C}, u_\mathrm{L})^T$ and the control input $u = (u_\mathrm{CR}, u_\mathrm{LR})^T$.

4.3 Measurement model for MHE

The measured states of the system are $y = (x_\mathrm{C}, x_\mathrm{L}, \alpha, u_\mathrm{C}, u_\mathrm{L}, u_\mathrm{CR}, u_\mathrm{LR})^T$ and we have $h_y(x,u) = (x_\mathrm{C}, x_\mathrm{L}, f(\theta), u_\mathrm{C}, u_\mathrm{L}, u_\mathrm{CR}, u_\mathrm{LR})^T$. Here α is the measured cable angle, which is different from the real load angle θ due to imperfect alignment of the cables (see [Vukov et al., 2012] for more details on the $\alpha - \theta$ relation).

Optimizing the control allows one to account for disturbances and unmodeled dynamics. This approach is motivated by the fact that the control hardware and the actuators are sometimes connected via analog lines. Those analog lines collect noise and disturb the control signal that is sent to the process.

Inclusion of the state noise in the MHE formulation is possible, but probably prohibitively expensive given that the solver used in this paper has cubic complexity in number of control inputs.

4.4 Output model for MPC

Our aim is to control the position and swinging of the load in the x_1-x_2 plane. For this purpose, the selected controlled system outputs are the load position x_1 and x_2 and the swinging velocity ω.

$$\tilde{r} = \begin{pmatrix} x_{1,\mathrm{ref}} \\ x_{2,\mathrm{ref}} \\ \omega \\ u_\mathrm{CR} \\ u_\mathrm{LR} \end{pmatrix}, h_r(x,u) = \begin{pmatrix} x_\mathrm{C} + x_\mathrm{L}\sin(\theta) \\ x_\mathrm{L}\cos(\theta) \\ \omega \\ u_\mathrm{CR} \\ u_\mathrm{LR} \end{pmatrix} \tag{3}$$

In [Vukov et al., 2012], the load position and swinging was controlled more indirectly by controlling the cart position x_C, cable length x_L and angle θ. Since there is a coupling between these variables tuning of the MPC R matrix was involved in order to obtain acceptable behaviour in the $x_1 - x_2$ plane. By considering (3) as a controlled output this coupling is implicitly included and the control weights can directly be imposed on the actual output (x_1, x_2).

5. EXPERIMENTAL RESULTS

In this section we illustrate the real-time performance of the closed loop system. At each sampling time the nonlinear MHE and MPC are solved sequentially. First the MHE estimats the current state by using the sensor measurements as an input. This estimate serves as in input for the MPC control problem. This control loop is closed at a sampling rate of 100 Hz. First the non linear estimation and control problems are described in more detail. Then, experiments and results are discussed.

5.1 The nonlinear moving horizon estimator

The MHE diagonal weighting matrices we used are given by:

$$S = \text{diag}(16.5\ \text{m}^{-2}, 25.1\ \text{m}^{-2}, 119.4, 1.2\ \text{V}^{-2}, 0.4\ \text{V}^{-2}, 0.01\ \text{s}^2/\text{V}^2, 0.01\ \text{s}^2/\text{V}^2)$$

These are obtained from experimentally tuning an initial guess. This initial guess for the weights was calculated using the variance of measured data for an open loop experiment (weight = $\text{var}^{-1/2}$). The experimental results given in the remainder of this Section show that these weights result in reliable state estimates. The estimation horizon is chosen to be 0.2 s with 20 shooting intervals, which was found to be suitable by empirical testing.

5.2 The nonlinear model predictive controller

We consider following bounds on the control and control rate

$$-10\ \text{V} \leq u_{\text{C}}(\tau), u_{\text{L}}(\tau) \leq 10\ \text{V}$$
$$-100\ \text{V/s} \leq u_{\text{CR}}(\tau), u_{\text{CR}}(\tau) \leq\ 100\ \text{V/s}$$

The considered state vector x, control vector u and reference vector $\tilde{r}$ are defined as in Section 4.4 . The control horizon is chosen to be 1 s with 10 shooting intervals, which was found to be suitable by empirical testing. The MPC formulation incorporates a reference vector $\tilde{r}$, which is a nonlinear combination of the state vector x. Point to point motions are executed by giving step reference changes, hence the references are constant over the whole control horizon. We consider the following reference for $\tilde{r}$ and u:

$$\tilde{r}_k = (x_{1,\text{ref}}, x_{2,\text{ref}}, 0, 0, 0)^T$$

Here $x_{1,\text{ref}}, x_{2,\text{ref}}$ are the desired load x_1 and x_2 position, while the desired angular velocity is zero to prevent swinging. The control reference is set to zero to minimize the total control effort. After some experimental tuning we obtain the following weighting matrices

$$R = \text{diag}(100\ \text{m}^{-2}, 100\ \text{m}^{-2}, 1\ \text{s}^2, 10^{-5}\ \text{s}^2/\text{V}^2, 10^{-5}\ \text{s}^2/\text{V}^2)$$

Since we want accurate point to point motions, the weight on the position x_1 and x_2 is relatively large with respect to the weight on the control inputs. Furthermore, since we also want fast point to point motions, the weight on the swinging velocity ω is smaller than the weight on the position x_1 and x_2.

5.3 Point to point motions

A first experiment illustrates both estimator and controller performance for several point to point motions.

Controller performance Multiple step references for the x_1 and x_2 position of the load are applied to the controller. The response of the controller can be seen in Figure 3. The steady state error for the x_1 and x_2 position is about 1 mm and 0.5 mm respectively. The settling time is about 3.5 s. It can be seen that the angle is still oscillating around zero with small amplitude (about 0.04 °). This small residual oscillation on θ is however not due to swinging in the

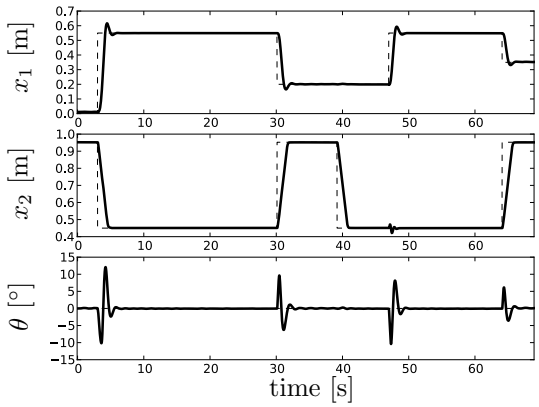

Fig. 3. Position of the load x_1, x_2 and angle deflection θ; solid lines: measurements, dashed lines: references

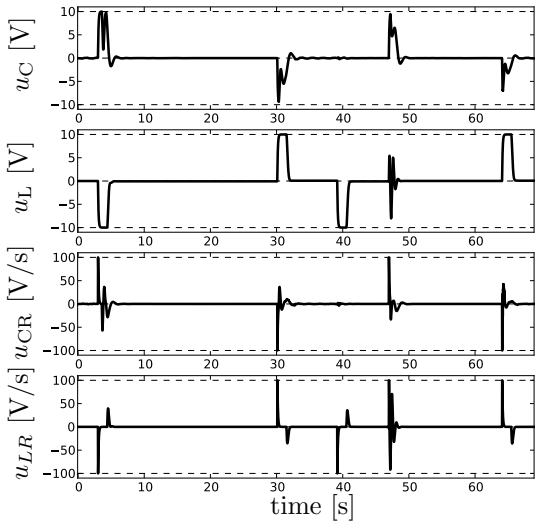

Fig. 4. Control of cart u_{C} and cable u_{L} and control rates u_{CR} and u_{LR}

$x_1 - x_2$ plane but due to the yawing of the load around the x_2 axis. This yawing motion is induced by the motion of the crane and is a consequence of the front and back of the load not hanging perfectly horizontal, due to a misalignment in the front and back cables of the load. This yaw results in front and back cable motions and hence results in oscillating θ measurements. Since the set-up cannot control the yawing of the load, this residual cannot be controlled to zero. The control reaction to this residual is negligible.

The control inputs and control rates are given in Figure 4. Here one can see that the cart control u_{C} and control rate u_{CR} and cable control u_{L} and control rate u_{LR} hit their limits. This rather aggressive performance is due to the small weights in the R matrix on the control effort, and large weights on the tracking error $(h_r - \tilde{r})$. The large weights on the tracking error will drive the controller toward the system limits to minimize the set point error as fast as possible.

It is interesting to note that a set point change in x_1 without change in x_2 results in a control input for the winching motor (u_{L}) as well (see Figure 3 at $\tau = 47$ s). This is due to the problem formulation which considers

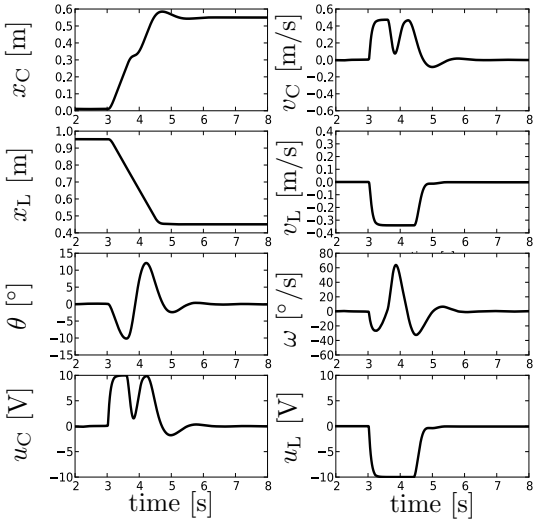

Fig. 5. MHE estimates of all states (Cart position ad velocity x_C and v_C, cable length and velocity x_L and v_L, angle and angular velocity θ and ω and control of cart and cable u_C and u_L) for point to point motion with MPC

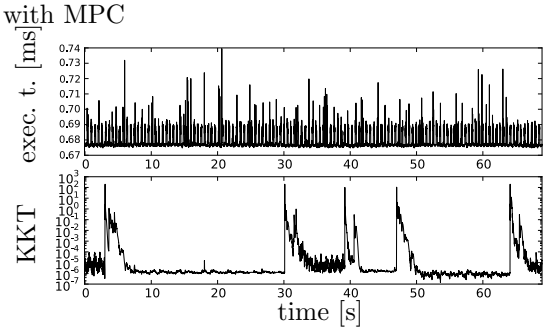

Fig. 6. (a) Overall NMHE execution time, (b) KKT tolerance.

the minimization of the 2-norm of the tracking error both on x_1 and x_2.

Estimator performance The state estimates the controller uses are given in Figure 5 for the first point to point motion. In order to validate these estimates of the state they are compared to the sensor measurements (for x_C, x_L, θ) and time derivative of the sensor measurements computed by finite differences (for v_C, v_L, ω). The estimates of the position measurements x_C, x_L, θ are reliable since rms values are small (0.48 mm, 0.543 mm, 0.01104 °) for the point to point motions in the previous experiment. The rms values (4.2 mm/s, 6.08 mm/s, 0.9857 °/s) of the velocities v_C, v_L, ω are a lot larger which is due to the finite difference calculation of the velocity measurements, which are very noisy (not shown). Figure 5 shows that the estimates for these velocities are smooth, while the finite difference signals are very noisy (not shown). Filtering this finite difference signal with a low pass filter could resolve this problem, however this will distort the signal and introduce a lag. Hence, using the estimates as an input for the controller is much more reliable than using the finite difference data. Since the rms values are small, and by visually inspecting the estimator signals, we conclude that the estimates of the states, given by the nonlinear moving horizon estimator are robust and reliable. Furthermore, the controller performance, which uses these estimates as in input, validates the performance of the estimator.

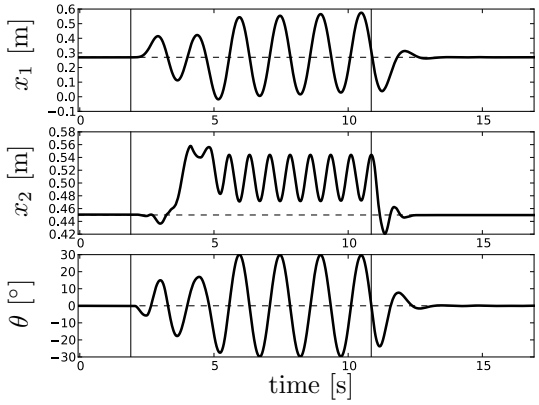

Fig. 7. Position of the load x_1, x_2 and angle deflection θ; solid lines: measurements, dashed lines: references.

Computation performance and optimality Figure 6 validates the low execution time of the estimators autogenerated C-code. The average execution time for this experiment is 0.68 ms while the peak execution time is 0.74 ms. The controllers autogenerated C-code execution times are similar with an average execution time of 0.34 ms and a peak execution time of 0.43 ms. The average execution time of the estimator-controller is then 1.02 ms while the peak execution time is 1.17 ms. The total computation time is hence significantly lower than the sampling time (1.17 ms << 10 ms) (which could even be increased). Execution times are measured using the Linux function `clock_gettime()`, which provides resolution in the nanosecond range.

Solving only one SQP iterations at each sampling instance results in a suboptimal estimate and control action. A good measure for the suboptimality is the Karush-Kuhn-Tucker (KKT) tolerance which in this case is calculated as the absolute value of the gradient of the Lagrangian function. This tolerance is given in Figure 6 for the MHE. Jumps in the KKT tolerance occur at big reference changes since the solution from one SQP iteration is only a rough approximation of the optimal solution. However, due to the good contraction properties of the Gauss-Newton real-time iteration scheme, the KKT tolerance decreases quickly in subsequent sampling instants. Similar results are obtained with the MPC.

5.4 Disturbance rejection

In a second experiment, an external disturbance is applied to the load position x_1, x_2 by making is swing with a very large amplitude (30 °). As can be seen in Figure 7. At $\tau = 0$ s, the controller has controlled the system to the given reference. At $\tau = 1.9$ s the controller is turned of and a disturbance, driving the systems away from the reference, is applied. The controller is turned on again at $\tau = 10.86$ s and fully rejects the large disturbance on the the position x_1, x_2 and the angle θ. While compensating for the disturbance, the control inputs again saturate.

Figure 8 shows the online estimate of the angle and the difference with the measurement. It can be seen that the estimate of the angle corresponds very well to the real

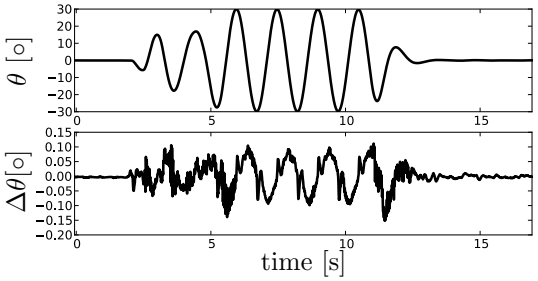

Fig. 8. Top: online estimate of the angle θ, Bottom, difference bewteen the estimate and the measured value of the angle θ.

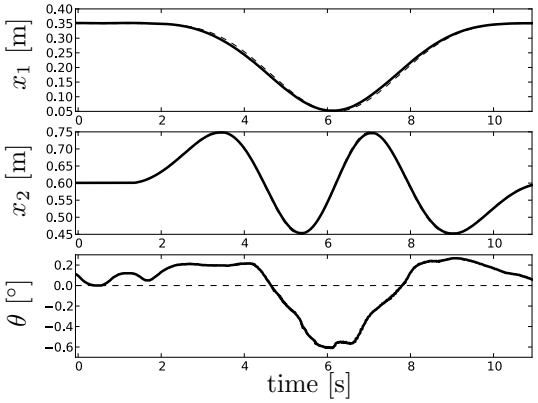

Fig. 9. Position of the load x_1, x_2, cart x_{C} and cable x_{L} and angle deflection θ; solid lines: measurements, dashed lines: references

angle, even for very large angles (here 30 °), while this makes the state equations highly nonlinear.

5.5 Servo-tracking

In this last experiment we show how the controller performs while following a given reference trajectory in x_1 and x_2 position as a function of time. The reference for the nonlinear controller over the time horizon is no longer constant but time varying for x_1 and x_2.

$$\tilde{r}_k = (x_{1,\mathrm{ref},k}, x_{2,\mathrm{ref},k}, 0, 0, 0)^T, \text{ for all } k = 0, \ldots, N-1$$

while the control reference is again set to zero over the whole control interval. As a reference trajectory for x_1 and x_2 we choose a Lissajous figure:

$$x_{1,\mathrm{ref}}(\tau_{\mathrm{ref}}) = 0.2 + 0.15\sin(2\pi s(\tau_{\mathrm{ref}}) + \pi/2)$$
$$x_{2,ref}(\tau_{\mathrm{ref}}) = 0.6 + 0.15\sin(4\pi s(\tau_{\mathrm{ref}}))$$

where $s(\tau_{\mathrm{ref}}) = -2\tau_{\mathrm{ref}}^3 + 3\tau_{\mathrm{ref}}^2$ for $\tau_{\mathrm{ref}} = 0..10$ s ensures that the trajectories start and stop smoothly in the velocity. Figure 9 shows the result of this trajectory following. We observe good tracking performance with maximal tracking error 10 mm and 5 mm for x_1 and x_2 respectively and RMS tracking error 5.7 mm and 2.7 mm for x_1 and x_2 respectively.

6. CONCLUSIONS

We can conclude that real-time implementation of the NMHE-NMPC control structure (1)-(2), obtained using the latest code-gen tools of ACADO shows low computation time (1.17 ms) and high performance (reliable state estimates and good point-to-point, tracking and disturbance rejection control performance) on an experimental overhead crane setup.

REFERENCES

H.G. Bock. Recent advances in parameter identification techniques for ODE. In P. Deuflhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*. Birkhäuser, Boston, 1983.

H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings 9th IFAC World Congress Budapest*, pages 242–247. Pergamon Press, 1984.

Herman Bruyninckx, Peter Soetens, and Bob Koninckx. The real-time motion control core of the Orocos project. In *IEEE International Conference on Robotics and Automation*, pages 2766–2771, 2003.

M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, Universität Heidelberg, 2001. URL `http://www.ub.uni-heidelberg.de/archiv/1659/`.

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*, 12(4):577–585, 2002.

M. Diehl, H. J. Ferreau, and N. Haverbeke. *Nonlinear model predictive control*, volume 384 of *Lecture Notes in Control and Information Sciences*, chapter Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation, pages 391–417. Springer, 2009.

A. Domahidi, A. Zgraggen, M.N. Zeilinger, M. Morari, and C.N. Jones. Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control. In *IEEE Conference on Decision and Control (CDC)*, pages 668 – 674, Maui, HI, USA, December 2012.

H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008. doi: 10.1002/rnc.1251. URL `http://onlinelibrary.wiley.com/doi/10.1002/rnc.1251/abstract`.

H.J. Ferreau, T. Kraus, M. Vukov, W. Saeys, and M. Diehl. High-speed moving horizon estimation based on automatic code generation. In *Proceedings of the 51th IEEE Conference on Decision and Control (CDC 2012)*, 2012.

J. V. Frasch, M. Vukov, H.J. Ferreau, and M. Diehl. A dual Newton strategy for the efficient solution of sparse quadratic programs arising in SQP-based nonlinear MPC, 2013. URL `http://www.optimization-online.org/DB_HTML/2013/07/3972.html`. Optimization Online 3972.

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

B. Houska, H.J. Ferreau, and M. Diehl. An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range. *Automatica*, 47(10):2279–2285, 2011b. doi: 10.1016/j.automatica.2011.08.020.

J. Mattingley and S. Boyd. *Convex Optimization in Signal Processing and Communications*, chapter Automatic Code Generation for Real-Time Convex Optimization. Cambridge University Press, 2009.

R. Quirynen, M. Vukov, and M. Diehl. Auto Generation of Implicit Integrators for Embedded NMPC with Microsecond Sampling Times. In Mircea Lazar and Frank Allgöwer, editors, *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference*, 2012.

S. Richter, M. Morari, and C.N. Jones. Towards computational complexity certification for constrained MPC based on Lagrange Relaxation and the fast gradient method. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5223 –5229, dec. 2011. doi: 10.1109/CDC.2011.6160931.

D Schindele and D Aschemann. Fast nonlinear mpc for an overhead travelling crane. In *In 18th IFAC World Congress 2011, Milano, Italy*, 2011.

H. Seguchi and T. Ohtsuka. Nonlinear Receding Horizon Control of an Underactuated Hovercraft. *International Journal of Robust and Nonlinear Control*, 13(3–4):381–398, 2003.

M. Vukov, W. Van Loock, B. Houska, H.J. Ferreau, J. Swevers, and M. Diehl. Experimental Validation of Nonlinear MPC on an Overhead Crane using Automatic Code Generation. In *The 2012 American Control Conference, Montreal, Canada.*, 2012.

M. Vukov, A. Domahidi, H. J. Ferreau, M. Morari, and M. Diehl. Auto-generated Algorithms for Nonlinear Model Predicitive Control on Long and on Short Horizons. In *Proceedings of the 52nd Conference on Decision and Control (CDC)*, 2013. (accepted).