

Experimental Validation of Nonlinear MPC on an Overhead Crane using Automatic Code Generation

Milan Vukov Wannes Van Loock Boris Houska Hans Joachim Ferreau Jan Swevers Moritz Diehl

Abstract—Recent advances in improving the efficiency of nonlinear model predictive control (MPC) algorithms have made them suited for challenging mechatronic applications that require high sampling rates. We demonstrate this fact by applying a highly efficient nonlinear MPC algorithm to a laboratory-scale overhead crane setup, featuring a fast moving cart and a winch mechanism. The aim is to perform optimized point-to-point motions with varying line length while respecting actuator limits. In order to solve the resulting optimization problems in less than one millisecond, an automatically generated Gauss-Newton real-time iteration algorithm is employed. We show experimental results illustrating the control performance of the closed-loop system as well as the efficiency of the nonlinear MPC algorithm.

I. INTRODUCTION

During the last decades, the concept of model predictive control (MPC) has received widespread acceptance in both academia and industry [1], [2], [3]. As convex quadratic programming solvers became increasingly faster [4], [5], [6], especially *linear* MPC became applicable for fast dynamic systems such as mechatronic devices [7], [8]. In contrast, *nonlinear* MPC has been mostly applied to systems exhibiting slower dynamics so far. This is mainly due to the fact that nonlinear MPC requires more computation power than linear formulations, but also ensuring convergence of iterative methods for solving non-convex optimization problems is more challenging. Several approaches have been proposed for nonlinear MPC applications with high sampling rates. Among them are the continuation/GMRES method by [9], the advanced step NMPC controller by [10] and the real-time iteration scheme by [11], [12]. An overview of existing algorithms for fast nonlinear MPC algorithms can be found in [13].

In order to yield highly efficient implementations of algorithms for solving MPC problems online, the use of code generation has been recently proposed by several authors. For example, the tool `AutoGenU` [14] generates source code implementing the continuation/GMRES method. The software package `CVXGEN` [15] allows the user to generate customized interior-point solvers for small-scale LP and QP problems as arising in linear MPC problems. The `ACADO Code Generation` tool as presented in [16] generates customized and self-contained C-code implementations of the real-time iteration (RTI) scheme. Based on a symbolic

Milan Vukov, Boris Houska, Hans Joachim Ferreau, and Moritz Diehl are with the Department of Electrical Engineering, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium.

Wannes Van Loock and Jan Swevers are with the Department of Mechanical Engineering, KU Leuven, Celestijnenlaan 300 B, 3001 Leuven, Belgium.

Contact: milan.vukov@esat.kuleuven.be

problem formulation, the auto-generated code is tailored to the specific problem structure and optimized for fast execution.

These developments seem to make nonlinear MPC suited for controlling systems with fast dynamics at high sampling rates. This paper investigates the efficiency of today's nonlinear MPC algorithms by applying them to a challenging overhead crane setup. The experimental setup consists of a cart moving in one dimension and a varying length pendulum attached to it. In contrast to [17], [18], where similar crane setups have been controlled by means of nonlinear MPC and linear time-optimal MPC with constant cable length, line length variations which greatly increase the nonlinearities of the system, are controlled directly by the nonlinear MPC.

A. Nonlinear MPC Problem Formulation

Within this paper, we consider algorithms for solving nonlinear MPC problems of the following form:

$$\begin{aligned}
 \min_{x(\cdot), u(\cdot)} \quad & \int_t^{t+T} (\|x(\tau) - x_{\text{ref}}(\tau)\|_Q^2 + \|u(\tau) - u_{\text{ref}}(\tau)\|_R^2) d\tau \\
 & + \|x(t+T) - x_{\text{ref}}(t+T)\|_P^2 \\
 \text{s.t.} \quad & x(t) = x_t \\
 & \dot{x}(\tau) = f(x(\tau), u(\tau)) \\
 & \underline{u} \leq u(\tau) \leq \bar{u} \\
 & \underline{x} \leq x(\tau) \leq \bar{x} \quad \text{for all } \tau \in [t, t+T]
 \end{aligned} \tag{1}$$

Here, t denotes the current time, x the differential state, u the control input, and T the fixed length of the prediction horizon. Variables x_{ref} and u_{ref} denote state and control references, respectively. As usual in tracking MPC formulations, the norms in the objective function can be weighted with positive-definite matrices Q , R and P . Additionally, upper and lower bounds \underline{u} , \bar{u} and \underline{x} , \bar{x} on the controls and states, respectively, are to be respected by the controller. Note, that the problem (1) can be regarded as a parametric nonlinear optimal control problem (OCP), where the parameter x_t represents the state estimate at the current time instant t .

B. Contributions and Overview

This paper demonstrates that nonlinear MPC can be applied in practice to mechatronic systems with fast dynamics. To this end, an auto-generated real-time iteration algorithm based on an MPC formulation comprising 8 states, 2 control inputs and 12 prediction intervals is applied to an overhead crane setup. It is shown that the worst-case execution time of the MPC controller is significantly faster than the necessary

sampling time of 10 milliseconds. Also the resulting closed-loop performance when performing point-to-point motions or rejecting external disturbances is discussed.

Section II briefly introduces the real-time iteration scheme as previously proposed for fast nonlinear MPC applications and summarizes the main features of the ACADO Code Generation tool. The overhead crane setup and its dynamic model are presented in detail in Section III. Section IV discusses experimental results as obtained by using nonlinear MPC for controlling the overhead crane. Finally, Section V concludes the paper.

II. REAL-TIME ITERATIONS AND CODE GENERATION FOR FAST NONLINEAR MPC

This section briefly describes the real-time iteration scheme and introduces the ACADO Code Generation tool as used for performing the experiments as described in Section III.

A. Gauss-Newton Real-Time Iterations

In this paper we employ a nonlinear real-time iteration algorithm, which has been originally developed in [11]. The basic strategy is to discretize the optimal control problem (1) with a multiple shooting discretization [19] using numerical integration and a piecewise constant control parameterization. This leads to a structured nonlinear programming problem that can be solved with a sequential quadratic programming (SQP) method. As the objective consists of a least squares tracking term, it is reasonable to employ a Gauss-Newton method to approximate the Hessian matrices [20]. Thus, the algorithm requires only first-order sensitivities of the state trajectory with respect to the inputs.

The main idea of the real-time iterations is to perform only one SQP step per sampling time, i.e., the optimal control problem (1) is only approximately solved in each real-time iteration. Due to the fact that the state estimate x_t enters the optimization problem only via the affine constraint $x(t) = x_t$, we can integrate the dynamic system as well as the corresponding sensitivities without knowing the actual value of the system state. This part is called the preparation phase. As soon as an estimate of the system state is available we only need to solve a single quadratic programming (QP) problem. Afterwards, the QP solution corresponding to the control inputs of the first time-interval can be immediately sent to the real process. This phase is called the feedback phase and is typically much shorter than the overall SQP iteration. For the details of this approach, we refer to [11]. Moreover, a mathematical foundation of the nonlinear real-time iteration scheme as well as stability results can be found in [21], [22], [10].

B. Automatic C-Code Generation

In order to obtain a fast implementation of the outlined nonlinear MPC algorithm, we use the latest version of the ACADO Code Generation tool, which is part of the open-source software ACADO Toolkit [23], [16]. The idea behind this tool is to optimize the number of required

operations in the evaluation of the nonlinear right-hand side of the differential equation as well as in the associated derivatives. In order to accomplish this, the tool makes use of a symbolic representation of the optimal control problem. The derivatives are symbolically simplified employing automatic differentiation tools and taking into account zero-entries in the Jacobian. Similarly, a tailored Runge-Kutta method with fixed step-sizes for integrating the model equations is generated in form of optimized C-code. The Runge-Kutta routine also integrates the associated variational differential equations to obtain the first-order derivative information as needed within the Gauss-Newton real-time iteration algorithm.

Note that the exported code is a direct optimal control method, as it is based on the multiple shooting concept [20]. Also the associated linear algebra operations for eliminating the states from the QP sub-problem are optimized. In the implementation used in this paper, the resulting dense QP sub-problems are solved by an embedded variant of the online QP solver qpOASES [24], [5] using fixed dimensions and static memory. For more details on the implementation of the ACADO Code Generation tool we refer to [16].

III. EXPERIMENTAL SETUP AND DYNAMIC MODEL

This section briefly describes the experimental setup as well as the dynamic model which is used in the nonlinear model predictive controller.

A. Experimental Setup

A schematic description of the considered overhead crane is given in Fig. 1. The cart is actuated by a Servotube 1108 linear motor from Copley Controls. The motor has an integrated incremental encoder which measures the position of the cart x_C with a resolution of $5\mu\text{m}$. Its maximum stroke is 0.6m. The pendulum consists of a cylindrical load with mass $m = 1.3\text{kg}$ hanging on two parallel cables. One end of each cable is mounted to a fixed point on the cart, Fig. 1, while the other two ends are connected to a winch mechanism. The winch mechanism consists of a pulley and a coupled DC motor (A-max 32 from Maxon Motors) with a gearbox reduction ratio 18. An incremental encoder with a resolution of 500 pulses per revolution is attached to

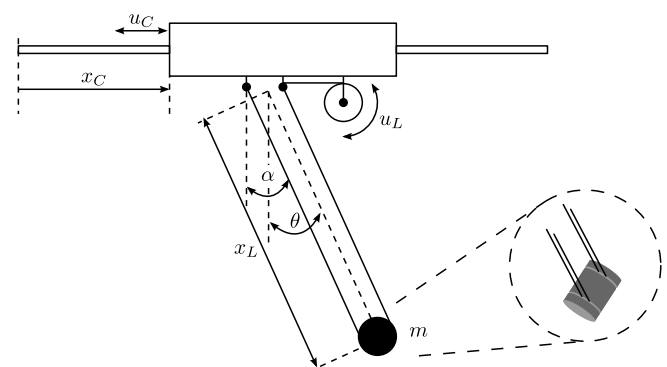


Fig. 1. Schematic of the experimental overhead crane.

the winch motor yielding a resolution of the cable length measurement x_L of $2.15\mu\text{m}$. The maximum cable length is 0.95m , while the minimum cable length is bounded to 0.5m for safety reasons. The angular deflection α of the left cable is measured with rotary incremental encoder BFH 1P.05A40000-B2-5 from Baumer Electric with a resolution of 40000 pulses per revolution. The relation between the angular deflection of the pendulum θ and of the left cable α is detailed in the appendix.

The inputs to this system are the voltages u_C and u_L , which represent setpoints for the associated velocity controllers. These voltages are internally limited to $\pm 10\text{V}$. The control software is implemented using the OROCOS Toolchain [25] and runs on a PC with an Intel Xeon 2.53GHz quad core processor, 12GB RAM memory, and a preemptive Linux kernel as operating system. Control commands are sent via EtherCAT to IO-boxes while measurements are received in a similar way. The sampling frequency was fixed to 100Hz .

B. Dynamic Model

The equation of motion of the variable cable length overhead crane is given by [26]:

$$\ddot{x}_C \cos(\theta) = -x_L \ddot{\theta} - 2\dot{x}_L \dot{\theta} - g \sin(\theta), \quad (2)$$

where g denotes the gravitational constant. The damping of the pendulum is neglected.

The closed-loop transfer function, $G_C(s)$, from the velocity controller setpoint u_C to the cart position x_C can be modeled as a pure integrator due to the fast dynamics of the velocity controller compared to the overall system dynamics. Moreover, a higher order model would require smaller discretization steps resulting in a longer integration time of the model. On the other hand, the winch mechanism dynamics, $G_L(s)$, from the input of the winch velocity controller u_L to the cable length x_L is modeled as a second order system. Thus the transfer functions are given by

$$G_C(s) = \frac{A_C}{s} \quad \text{and} \quad G_L(s) = \frac{A_L}{s(\tau_L s + 1)}. \quad (3)$$

Non-parametric estimates of the frequency response functions (FRFs) of these systems were obtained from random phase multisine excitations [27]. The parameters A_C , A_L and τ_L were subsequently identified by a weighted nonlinear least squares frequency domain identification method [27]. The numerical values for the parameters of the model are given in Table I.

By combining (2) and (3) and introducing voltage rate variables u_{CR} and u_{LR} , which allow us to account for actuator torque constraints indirectly, we find the following nonlinear

TABLE I
ESTIMATE OF THE MODEL PARAMETERS.

Parameter	Description	Value
A_C	gain of $G_C(s)$	0.0474m/s/V
A_L	gain of $G_L(s)$	0.0341m/s/V
τ_L	time constant of $G_L(s)$	0.0247s
g	gravitational constant	9.81m/s^2

ordinary differential equation:

$$\begin{aligned} \dot{x}_C &= v_C, \\ \dot{v}_C &= A_C u_{CR}, \\ \dot{x}_L &= v_L, \\ \dot{v}_L &= -\frac{1}{\tau_L} v_L + \frac{A_L}{\tau_L} u_L, \\ \dot{\theta} &= \omega, \\ \dot{\omega} &= -\frac{1}{x_L} (A_C u_{CR} \cos(\theta) + g \sin(\theta) + 2v_L \omega), \\ \dot{u}_C &= u_{CR}, \\ \dot{u}_L &= u_{LR}. \end{aligned} \quad (4)$$

These equation will in the following be summarized as $\dot{x} = f(x, u)$ where

$$x := (x_C, v_C, x_L, v_L, \theta, \omega, u_C, u_L)^T \quad \text{and} \quad u := (u_{CR}, u_{LR})^T \quad (5)$$

denote the states and controls, respectively.

IV. EXPERIMENTAL RESULTS

In this section we present experimental results illustrating the real-time performance of the closed-loop system. First, the nonlinear optimal control problem is described in more detail. Afterwards the results of two experiments are discussed: i) a point-to-point motion in the vertical plane and ii) the rejection of an external disturbance applied to the load.

A. The Nonlinear Model Predictive Controller

At each sampling time the nonlinear OCP (1) is solved with the following bounds on controls and states

$$\begin{aligned} -10\text{V} &\leq u_C(\tau) \leq 10\text{V}, \\ -10\text{V} &\leq u_L(\tau) \leq 10\text{V}, \\ -100\text{V/s} &\leq u_{CR}(\tau) \leq 100\text{V/s}, \\ -100\text{V/s} &\leq u_{LR}(\tau) \leq 100\text{V/s}. \end{aligned}$$

The system inputs u_C and u_L are constrained by internal limitations of the corresponding velocity controller. Constraints for the control rates u_{CR} and u_{LR} are tuned such that saturations in the cart and winch actuators are avoided. The state and control vectors $x(\cdot)$ and $u(\cdot)$ and the system description are defined as in (5) and (4), respectively. The control horizon is chosen to be $T = 1\text{s}$, which was found to be suitable by empirical testing.

The current MPC formulation (1) can incorporate references for full state and control vectors. We keep the references constant over the control horizon. Furthermore, only the references for the cart position and cable length are

changed online, while all other references are always set to zero:

$$x_{\text{ref}} := (x_{\text{C,ref}}, 0, x_{\text{L,ref}}, 0, 0, 0, 0, 0)^T \quad \text{and} \quad u_{\text{ref}} := (0, 0)^T.$$

Moreover, we choose weighting matrices Q and R defined as:

$$Q := \text{diag} (56 \text{m}^{-2}, 6 \text{s}^2/\text{m}^2, 115 \text{m}^{-2}, 0.01 \text{s}^2/\text{m}^2, \\ 10, 10 \text{s}^2, 10^{-8} \text{V}^{-2}, 10^{-8} \text{V}^{-2}), \\ R := \text{diag} (10^{-5} \text{s}^2/\text{V}^2, 10^{-5} \text{s}^2/\text{V}^2).$$

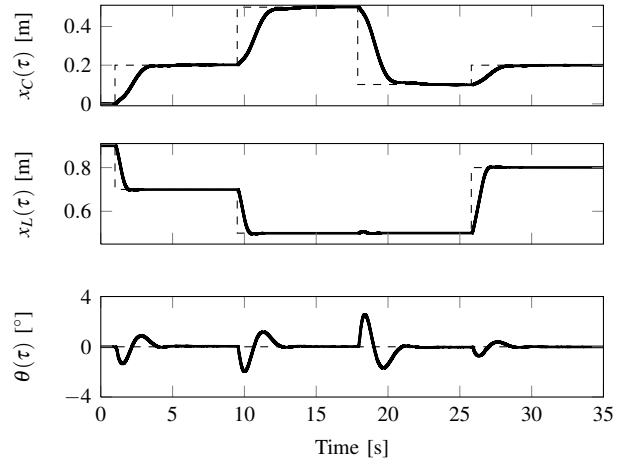
Since our intention is to perform accurate point-to-point motions, the weights corresponding to the tracking errors of the cart position and the cable length are chosen to be relatively large in comparison to the weights which penalize the tracking error of the velocities. Similarly, the weights for the control cost are quite low. The aim of this tuning is to enable fast but well damped closed-loop behaviour. The Mayer term in the objective formulation (1), $\|x(t+T) - x_{\text{ref}}(t+T)\|_p^2$, is omitted during the experiments.

Note that the nonlinear MPC controller itself requires estimates for all the eight states as an input. However, only the positions are directly measured by the encoders. Consequently, we have to estimate the corresponding velocities $v_{\text{C}}, v_{\text{L}}$ and ω , in our case by using finite differences. For the cart velocity we use $v_{\text{C}}[k] = (x_{\text{C}}[k] - x_{\text{C}}[k-1])/T_{\text{s}}$, $T_{\text{s}} = 10\text{ms}$. To reduce effects of high-frequency noise, we filter all velocities using a first order low-pass digital filter with a cut-off frequency of $f_{\text{c}} = 10\text{Hz}$. This choice is justified by the slow pendulum dynamics and the assumption that the NMPC does not trigger fast modes of the cart mechanism. An auto-generated nonlinear moving horizon estimator (MHE) is under development and will be used in future implementations.

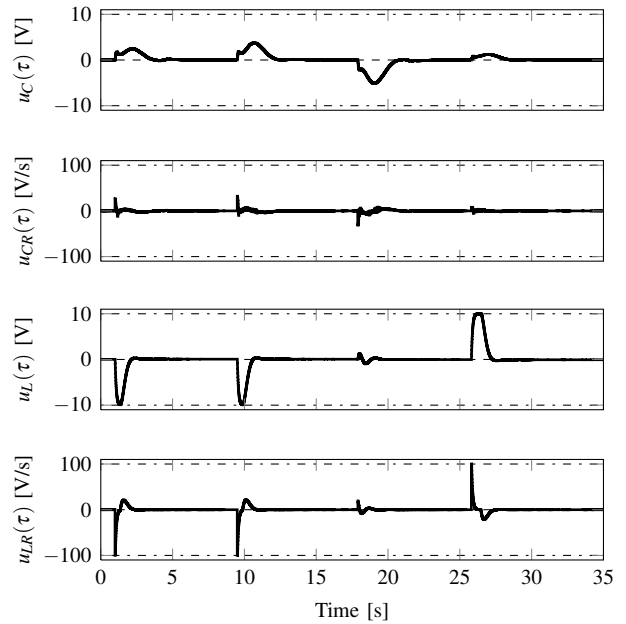
B. Results

A first experiment illustrates the controller performance for point-to-point motions, see Fig. 2. Multiple reference changes are applied to the controller (Fig. 2(a)). The steady-state error on x_{C} is less than 2mm and less than 1mm on x_{L} . The control inputs and corresponding control rates are shown in Fig. 2(b). Note that the winch control and its slew-rate, u_{L} and u_{LR} , hit their limits due to the fact that the corresponding weights in the Q and R matrix are set quite aggressively. Weights in the Q and R which correspond to motion of the cart and to pendulum swinging are set to quite conservative values resulting in rather slow yet accurate motion. Also, it is interesting to note that the third setpoint change, a change in x_{C} only, results in a control input for the winching motor as well.

Fig. 3(a) validates the low execution time of the auto-generated C-code. The average execution time is less than 1ms. The peak execution time measures less than 1.1ms. In the worst case, the time spent in the feedback phase is 17% of the overall execution time, see Table II. The peak times in the feedback phase are directly proportional to the number of active-set changes for solving the QP.



(a) Cart position x_{C} , cable length x_{L} , angle deflection θ ; solid lines: measurements, dashed lines: references.



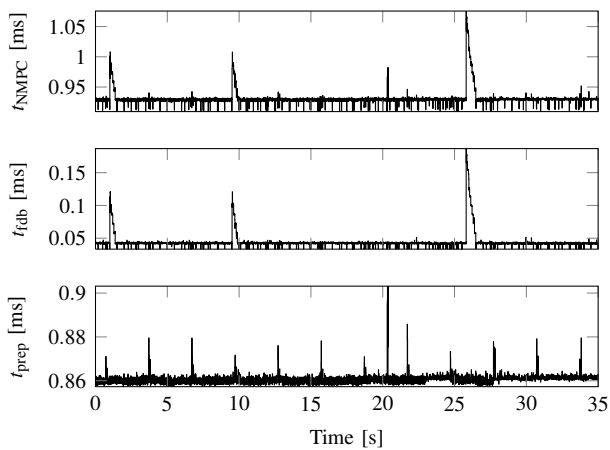
(b) Controls u_{C} , u_{L} and control rates u_{CR} , u_{LR} ; solid lines: measurements, dashed lines: references, dash-dotted lines: constraints.

Fig. 2. Point-to-point motions in the vertical plane: states and controls

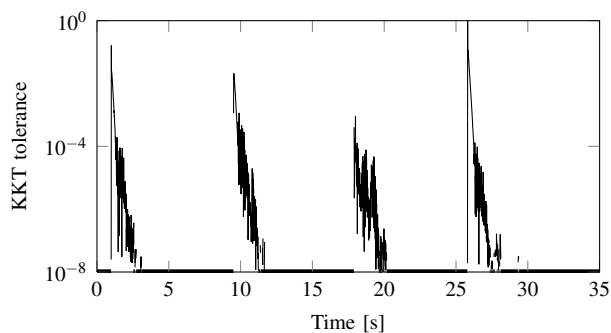
TABLE II
WORST CASE EXECUTION TIME OF THE AUTO-GENERATED NMPC.

	CPU time	Percentage
Preparation phase	0.90ms	83%
Feedback phase	0.19ms	17%
Overall execution time	1.09ms	100%

The ratio between the average time spent in the feedback and preparation phase is almost equal to results obtained in synthetic tests in [16]. Execution times are measured with OROCOS timer services. Those services internally use the Linux function `clock_gettime()`, which provides resolution in the nanosecond range.



(a) $t_{\text{NMPC}} = t_{\text{fdb}} + t_{\text{prep}}$: overall NMPC execution time, t_{fdb} : duration of the feedback phase, t_{prep} : duration of the preparation phase.

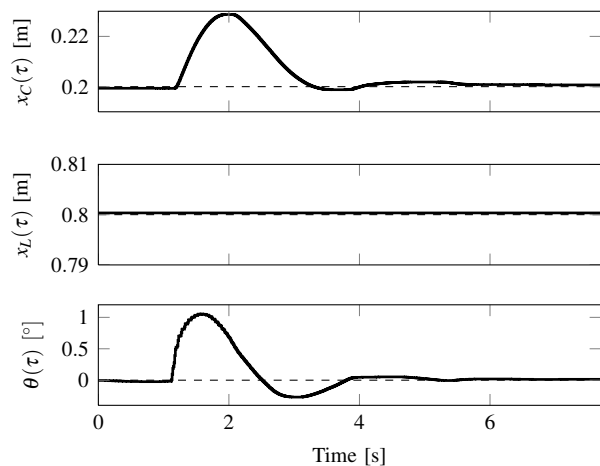


(b) KKT tolerance; the bold bottom line indicates the segments where KKT tolerance is below 10^{-8} .

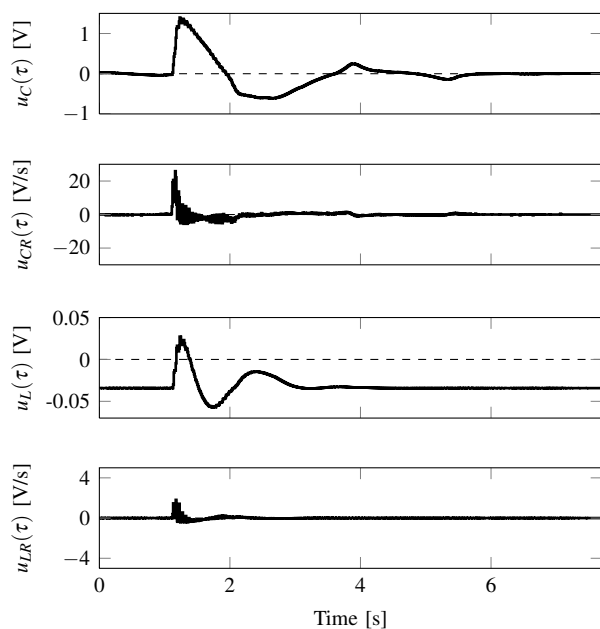
Fig. 3. Point-to-point motions in the vertical plane: performance of the nonlinear MPC.

Performing only one SQP iteration at each sampling time results in suboptimal control. Therefore it is good to know how well the controller performs in terms of optimality. The controller performance is measured using the Karush-Kuhn-Tucker (KKT) tolerance which in this case is calculated as the absolute value of the gradient of the Lagrangian function. The measurements presented in Fig. 3(b) show a good convergence rate of the proposed MPC controller. Jumps in the KKT tolerance occur at big reference changes since the solution obtained from one SQP iteration is only a rough approximation to the optimal solution in these situations. However, due to the good contraction properties of the Gauss-Newton real-time iteration scheme, the KKT tolerance decreases quickly in subsequent sampling instants. Note that the KKT tolerance would be identically to zero for a linear system because in each RTI a QP is solved exactly; thus, the non-zero values of the KKT tolerance are an indication of the nonlinearity of the optimal control problem.

In a second experiment an external disturbance acts on the load by pushing it from its equilibrium. As seen in Fig. 4(a), the controller successfully rejects the disturbance. Note that the cable length does not change despite a non-zero control signal (Fig. 4(b)), due to a low voltage signal which



(a) Cart position x_C , cable length x_L , angle deflection θ ; solid lines: measurements, dashed lines: references.



(b) Controls u_C , u_L and control rates u_{CR} , u_{LR} ; solid lines: measurements, dashed lines: references

Fig. 4. Rejection of an external disturbance force applied to the hanging mass.

cannot overpower the static friction in the winch mechanism. Furthermore, since there is no integral action on the states x_C and x_L in the model (4), the NMPC is outputting a small non-zero values for the voltages u_C and u_L in the steady state.

V. CONCLUSIONS

In this paper we have experimentally validated a nonlinear MPC using automatic code generation applied to an overhead crane. The controller shows a good tracking accuracy. The worst case steady-state error in positioning of the load is less than 2mm. Furthermore, we have illustrated the low execution time of the proposed control strategy, as well as a fast convergence of the nonlinear optimal control problem.

ACKNOWLEDGEMENTS

The research was supported by the Research Council KUL via GOA/11/05 Ambiorics, GOA/10/09 MaNet, GOA/10/11 "Global real-time optimal control of autonomous robots and mechatronic systems", CoE EF/05/006 Optimization in Engineering (OPTEC) en PFV/10/002 (OPTEC), IOF-SCORES4CHEM and PhD/postdoc/fellow grants, the Flemish Government via FWO (PhD/postdoc grants, projects G0226.06, G0321.06, G.0302.07, G.0320.08, G.0558.08, G.0557.08, G.0588.09, G.0377.09, research communities ICCoS, ANMMM, MLDM) and via IWT (PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare), the Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011), the IBBT, the EU (ERNSI; FP7-HD-MPC (INFISO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC HIGH-WIND (259 166)), the Contract Research (AMINAL), the Helmholtz Gemeinschaft via viCERP and the ACCM. The fourth author holds a PhD fellowship of the Research Foundation – Flanders (FWO).

REFERENCES

- [1] M. Morari and J. Lee, "Model predictive control: past, present and future," *Computers and Chemical Engineering*, vol. 23, pp. 667–682, 1999.
- [2] F. Allgöwer and A. Zheng, *Nonlinear Predictive Control*, ser. Progress in Systems Theory. Basel Boston Berlin: Birkhäuser, 2000, vol. 26.
- [3] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill, 2009.
- [4] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002.
- [5] 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*, vol. 18, no. 8, pp. 816–830, 2008.
- [6] J. Mattingley, Y. Wang, and S. Boyd, "Code generation for receding horizon control," in *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, Yokohama, Japan, 2010.
- [7] A. Wills, D. Bates, A. Fleming, B. Ninness, and S. Moheimani, "Application of MPC to an active structure using sampling rates up to 25kHz." 44th IEEE Conference on Decision and Control and European Control Conference ECC'05, Seville, 2005.
- [8] L. Van den Broeck, J. Swevers, and M. Diehl, "Experimental validation of Time Optimal MPC on a linear drive system," in *Proceedings of the 11th International Workshop on Advanced Motion Control*, Nagaoka, Japan, 2010, pp. 355–360.
- [9] T. Ohtsuka, "A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control," *Automatica*, vol. 40, no. 4, pp. 563–574, 2004.
- [10] V. M. Zavala and L. Biegler, "The Advanced Step NMPC Controller: Optimality, Stability and Robustness," *Automatica*, vol. 45, pp. 86–93, 2009.
- [11] M. Diehl, "Real-Time Optimization for Large Scale Nonlinear Processes," Ph.D. dissertation, Universität Heidelberg, 2001, <http://www.ub.uni-heidelberg.de/archiv/1659/>.
- [12] M. Diehl, H. Bock, J. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations," *J. Proc. Contr.*, vol. 12, no. 4, pp. 577–585, 2002.
- [13] M. Diehl, H. J. Ferreau, and N. Haverbeke, *Nonlinear model predictive control*, ser. Lecture Notes in Control and Information Sciences. Springer, 2009, vol. 384, ch. Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation, pp. 391–417.
- [14] H. Seguchi and T. Ohtsuka, "Nonlinear Receding Horizon Control of an Underactuated Hovercraft," *International Journal of Robust and Nonlinear Control*, vol. 13, no. 3–4, pp. 381–398, 2003.
- [15] J. Mattingley and S. Boyd, *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2009, ch. Automatic Code Generation for Real-Time Convex Optimization.
- [16] B. Houska, H. 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.
- [17] D. Schindele and H. Aschemann, "Fast nonlinear MPC for an overhead travelling crane," in *Proceedings of the IFAC World Congress*, 2011.
- [18] L. Van den Broeck, M. Diehl, and J. Swevers, "Experimental validation of Time Optimal MPC on a flexible motion system," in *Proceedings of the 2011 American Control Conference*, San Francisco, USA, 2011, pp. 4749–4754.

- [19] H. Bock and K. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proceedings 9th IFAC World Congress Budapest*. Pergamon Press, 1984, pp. 243–247. [Online]. Available: <http://www.iwr.uni-heidelberg.de/groups/agbock/FILES/Bock1984.pdf>
- [20] H. Bock, "Recent advances in parameter identification techniques for ODE," in *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, P. Deuffhard and E. Hairer, Eds. Boston: Birkhäuser, 1983.
- [21] F. Allgöwer, T. Badgwell, J. Qin, J. Rawlings, and S. Wright, "Nonlinear Predictive Control and Moving Horizon Estimation – An Introductory Overview," in *Advances in Control, Highlights of ECC'99*, P. M. Frank, Ed. Springer, 1999, pp. 391–449.
- [22] M. Diehl, R. Findeisen, and F. Allgöwer, "A Stabilizing Real-time Implementation of Nonlinear Model Predictive Control," in *Real-Time and Online PDE-Constrained Optimization*, L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, Eds. SIAM, 2007, pp. 23–52.
- [23] 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.
- [24] "qpOASES Homepage," <http://www.qpOASES.org>, 2007–2011.
- [25] Orocos. (2011, September) Orocos - Open Robot Control Software project. [Online]. Available: <http://www.orocos.org/>
- [26] M. Fliess, J. Lévine, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, pp. 1327–1361, 1995.
- [27] J. Schoukens and R. Pintelon, *Identification of Linear Systems: A Practical Guide to Accurate Modeling*. Pergamon Press, 1991.

APPENDIX

MEASUREMENT OF THE ANGULAR DEFLECTION

A detailed situation is depicted in Fig. 5. Noting that triangles ABF and DEF are similar, the following equation can be obtained:

$$\frac{L_3}{L_2} = \frac{L_3 + L_1 + x_L}{d} = \tan(\beta), \quad (6)$$

with:

$$\begin{aligned} \beta &= \pi/2 + \alpha - \theta, \quad L_1 = a \sin(\theta), \quad L_2 = a \cos(\theta), \\ a &= 9.9 \text{ mm}, \quad d = 14.5 \text{ mm}. \end{aligned} \quad (7)$$

Solving (6) gives the angle deflection θ in function of α .

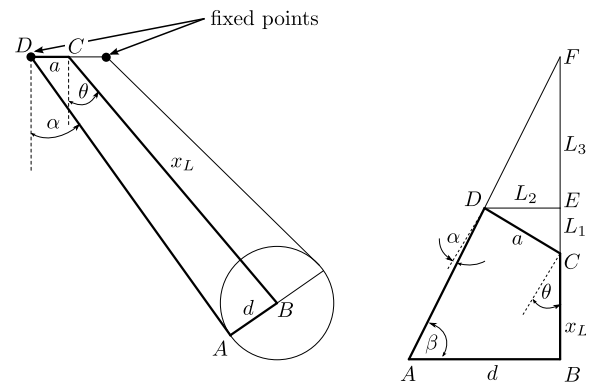


Fig. 5. Relation between measured angle α and pendulum angle deflection θ