Fast Nonlinear MPC for an Overhead Travelling Crane

Dominik Schindele ∗ Harald Aschemann ∗

∗ Chair of Mechatronics, University of Rostock, Germany, (email: {dominik.schindele,harald.aschemann}@uni-rostock.de)

Abstract: This paper presents a nonlinear model predictive control scheme for the two main axes of an overhead travelling crane, which guarantees both tracking of desired trajectories for the crane load and an active damping of crane load oscillations. The main idea of the used NMPC algorithm consists in a minimization of the tracking error at the end of the prediction horizon. That way the computation load can be kept relatively small. The varying length of the rope is considered by gain-scheduling techniques. The position of the crane load is measured by a CMOS camera using the spatial filtering principle. Desired trajectories for the crane load position in the three-dimensional workspace can be tracked independently with high accuracy. Experimental results from an implementation on a test rig show a high control performance.

Keywords: nonlinear control, predictive control, underactuated mechanical systems

1. INTRODUCTION

In the last decade, numerous model-based trajectory control schemes for overhead travelling cranes have been proposed by different authors. Beside nonlinear control approaches exploiting differential flatness, as in Fliess et al. (1991), Boustany and d’Andrea Novel (1992), energy based control, see Fang et al. (2003), and extended linearisation techniques have proved efficient, see Aschemann (2002), Nguyen (1994). Aiming at an increased handling frequency and a fully automated crane operation, the focus has to be on the motion of the crane load. Feedback control provides for accurate tracking of desired trajectories for the crane load in the three-dimensional workspace. In practical implementations, however, the additional inclusion of appropriate control action to counteract disturbances, especially nonlinear friction acting on the drives, is crucial. Furthermore, a robust or adaptive control approach is required as regards varying system parameters like rope length or load mass during crane operation (Nguyen (1994)). By this, the capabilities of an automated overhead crane can be extended in order to use it as a robot manipulator for the handling of heavy loads in a large cartesian workspace.

In this paper, the first principle modelling of the two main translational crane axes is addressed first. Aiming at a decentralized control structure, for each axis a separate design model is derived in symbolic form by exploiting Lagrange’s equations. Then a nonlinear model predictive control (NMPC) design is presented allowing for fast trajectory control. In general, at model predictive control the optimal input vector is mostly calculated by minimizing a quadratic cost function. A problem with most of the MPC algorithms is that they are not suitable for systems with fast dynamics because they cannot be computed within very small sample times. One approach to deal with this problem is given in Graichen et al. (2010), where a suboptimal optimization problem is solved. In contrast, the NMPC approach considered here aims at reducing future state errors, see Jung and Wen (2004), and allows for relatively small computational effort as required in a real-time implementation. A further attractive characteristic of this MPC approach is its applicability to linear as well as nonlinear systems. The varying height of the crane load is taken into account using gain-scheduling with the rope length as scheduling parameter. For the practical realization at a test rig (Fig. 1) the crane load position is measured by a CMOS camera, exploiting the spatial filtering principle. Desired trajectories for the crane load position in the $xyz$-workspace can be tracked independently with high accuracy. Experimental results of the closed-loop system show both excellent tracking performance and steady-state accuracy.
2. CONTROL-ORIENTED MODELLING

As a decentralized control structure is envisaged, each crane axis is modeled separately for controller design. Since the control-oriented modelling for the $x$- and $y$-axis is similar, the modelling shall be presented only for the $y$-axis. A two-body model for this part of the crane consisting of trolley and load is shown in Fig. 2. The origin of the $y$-axis, $y_k = 0$, is located on the left side of the crane bridge. With a bridge length $l_{br} = 0.9\, \text{m}$, the available workspace in $y$-direction is characterized by $y_k \in (0 \, \text{m}, 0.9\, \text{m})$. The trolley is modeled by a mass $m_k$, whereas the crane load is represented by a lumped mass $m_l$. The trolley is electrically driven by a motor force $F_k$. The main disturbances, nonlinear friction and damping, are taken into account by the resulting disturbance force $F_y$. The rope suspension is considered as massless connection, where rope deflections and small external damping are neglected. The two degrees of freedom for the mechanical model of the $y$-axis are chosen as the trolley position $q_1 = y_k$ and as the rope angle $q_2 = \varphi_y$. Then, the vector of generalized coordinates becomes $q = [y_k \, \varphi_y]^T$. By exploiting Lagrange’s equations, the equations of motion of the crane axis can be calculated and stated in the following matrix notation

$$
\begin{equation}
\begin{pmatrix}
\ddot{q}_1 + m_l \dot{q}_1 \cos \varphi_y \\
+ m_l \dot{q}_1 \sin \varphi_y \\
\end{pmatrix} =
\begin{pmatrix}
F_k - F_y \\
0
\end{pmatrix},
\end{equation}
$$

As the velocity of the trolley $\dot{y}_k(t)$ is controlled by an underlying control loop, the motion of the trolley can be described by a $PT_1$-system. Consequently, the first differential equation in (1) is replaced by

$$
T_1 \ddot{y}_k + \dot{y}_k = v_{ky},
$$

The variable $v_{ky}$ denotes the desired trolley velocity, which serve as control input for the simplified system. The variable $T_1$ represents the time constant of the underlying $PT_1$-system. Thus the disturbances lumped in $F_y$ are counteracted by the underlying velocity controller and do not need to be considered at the further control design. With the generalized coordinates as well as their time derivatives as state variables $x_y = [q^T \, \dot{q}^T]^T$, the crane load position $y_l$ as output variable, and the desired trolley velocity input variable $v_y = v_{ky}$ the corresponding state space representation becomes

$$
x_y = \begin{bmatrix}
\dot{y}_k \\
\dot{\varphi}_y \\
\frac{\ddot{y}_k}{T_1} - \frac{y_k}{T_1} \cos \varphi_y - g \sin \varphi_y \\
\frac{1}{T_1} \dot{y}_l \\
\end{bmatrix},
y_l = y_k + l_y \sin \varphi_y.
$$

3. NONLINEAR MODEL PREDICTIVE CONTROL

Predictive control represents a class of algorithms that are based on the prediction of the system states $x$ over a time span denoted as prediction horizon $T_P$. Model predictive control implies that a process model is used for the prediction of the dynamic behaviour. An outline of NMPC is given by Findeisen and Allgöwer (2002). Based on the state space model and the measured state vector at time $t_0$, the sequence of input variables according to a chosen cost function is calculated. After applying the first element of the input vector to the process, the optimization procedure is repeated at the following time instant with the prediction horizon moving forward: the moving horizon approach, see Diehl (2002). The main idea of the control approach consists in a minimization of a future tracking error in terms of the predicted state vector based on the actual state as well as the desired state vector resulting from trajectory planning, see Lizarraldea et al. (1999), Jung and Wen (2004). The minimization is achieved by repeated approximate numerical optimization in each time step, in the given case using the Newton-Raphson technique. The optimization is initialised in each time step with the optimization result of the preceding time step in form of the input vector. The NMPC-algorithm is based on the following nonlinear time-discrete state space representation

$$
x_{k+1} = f(x_k, u_k), \quad y_k = h(x_k),
$$

with the state vector $x_k \in \mathbb{R}^n$, the control input $u_k \in \mathbb{R}$, the output vector $y_k \in \mathbb{R}$ and the initial vector $x_0 \in \mathbb{R}^n$. The constant $M$ specifies the prediction horizon $T_P$ as a multiple of the sampling time $t_s$.

$$
T_P = M \cdot t_s.
$$

The predicted input vector at time $k$ becomes

$$
u_{k,M} = [u_k, u_{k+1}, \ldots, u_{k+M-1}]^T,
$$

with $u_{k,M} \in \mathbb{R}^M$. The predicted state vector at the end of the prediction horizon $\phi_M(x_k, u_{k,M})$ is obtained by repeated substitution of $k$ by $k+1$ in the time-discrete state equation (4)

$$
x_{k+2} = f(x_{k+1}, u_{k+1}) = f(f(x_k, u_k), u_{k+1})
$$

$$
: \quad x_{k+M} = f(\ldots f(x_k, u_k), \ldots, u_{k+M-1}) = \phi_M(x_k, u_{k,M}).
$$

The difference of $\phi_M(x_k, u_{k,M})$ and the desired state vector $x_d$ yields the final control error

$$
e_{M,k} = \phi_M(x_k, u_{k,M}) - x_d,
$$
i.e., the control error at the end of the prediction horizon. The cost function to be minimized follows as
\[ J_{\text{MPC}} = \frac{1}{2} e_{M,k}^T e_{M,k} , \]
and, hence, the necessary condition for an extremum can be stated as
\[ \frac{\partial J_{\text{MPC}}}{\partial e_{M,k}} = e_{M,k} = 0 . \]
A Taylor-series expansion of (10) at \( \mathbf{u}_{k,M} \) in the neighbourhood of the optimal solution leads to the following system of equations
\[ 0 = e_{M,k} + \frac{\partial \phi_M}{\partial \mathbf{u}_{k,M}} \Delta \mathbf{u}_{k,M} + T.h.O . , \]
with \( \Delta \mathbf{u}_{k,M} \) denoting the difference which has to be added to the input vector \( \mathbf{u}_{k,M} \) to obtain the optimal solution. The \( n \) equations (11) represent an under-determined set of equations with \( M \) unknowns having an infinite number of solutions. An unique solution for \( \Delta \mathbf{u}_{k,M} \) can be determined by solving a \( L_2 \)-optimization problem with (11) as side condition, which leads to the Moore-Penrose pseudo inverse of \( \frac{\partial \phi_M}{\partial \mathbf{u}_{k,M}} \) (Aschemann and Schindele (2007))
\[ \Delta \mathbf{u}_{k,M} = \left( \frac{\partial \phi_M}{\partial \mathbf{u}_{k,M}} \right)^+ e_{M,k} . \]
One major advantage of predictive control is the possibility to easily account for input constraints, which are present in almost all control applications. To this end, the cost function can be extended with a corresponding penalty term \( h(u) \), see Aschemann and Schindele (2007).
The overall NMPC-algorithm can be described as follows: Choice of the initial input vector \( \mathbf{u}_{0,M} \) at time \( k = 0 \), e.g. \( \mathbf{u}_{0,M} = 0 \), and repetition of steps a) - c) at each sampling time \( k \geq 0 \):

a) Calculation of an improved input vector \( \mathbf{v}_{k,M} \) according to
\[ \mathbf{v}_{k,M} = \mathbf{u}_{k,M} - \eta_k \left( \frac{\partial \phi_M}{\partial \mathbf{u}_{k,M}} \right)^+ e_{M,k} . \]
The step width \( \eta_k \) can be determined with, e.g., the Armijo-rule.

b) For the calculation of \( \mathbf{u}_{k+1,M} \) the elements of the vector \( \mathbf{v}_{k,M} \) have to be shifted by one element and the steady-state input \( u_d \) corresponding to the final state has to be inserted at the end
\[ \mathbf{u}_{k+1,M} = \begin{bmatrix} 0_{(M-1) \times 1} \mathbf{I}_{M-1} \\ 0_{(1 \times (M-1))} \end{bmatrix} \mathbf{v}_{k,M} + \begin{bmatrix} 0_{(M-1) \times 1} \\ 1 \end{bmatrix} u_d . \]
In general, the steady-state control input \( u_d \) follows from
\[ x_d = f(x_d, u_d) . \]
Alternatively, the desired input vector \( u_d \) can be calculated by an inverse system model. If the system is differentially flat (Fliess et al. (1995)), as in the given case, the desired input \( u_d \) is determined exactly using the flat system output and a finite number of its time derivatives.
c) The first element of the improved input vector \( \mathbf{v}_{k,M} \) is applied as control input at time \( k \)
\[ u_k = \begin{bmatrix} 1 \ 0_{(1 \times (M-1))} \end{bmatrix} \mathbf{v}_{k,M} . \]
In the proposed algorithm only one iteration is performed per time step. A similar approach using several iteration steps is described in Weidemann et al. (2004). An improvement of the trajectory tracking behaviour can be achieved if an input vector resulting from an inverse system model is used as initial vector for the subsequent optimization step instead of the last input vector. The slightly modified algorithm can be stated as follows:

a) Calculation of the ideal input vector \( \mathbf{u}_{k,M}^{(d)} \) by evaluating an inverse system model with the specified reference trajectory as well as a certain number \( \beta \in \mathbb{N} \) of its time derivatives
\[ \mathbf{u}_{k,M}^{(d)} = \mathbf{u}_{k,M}^{(d)} \left( y_d, y_d, \ldots, y_d^{(\beta)} \right) . \]
b) Calculation of the improved input vector \( \mathbf{v}_{k,M} \) based on the equation
\[ \mathbf{v}_{k,M} = \mathbf{u}_{k,M}^{(d)} - \eta_k \left( \frac{\partial \phi_M}{\partial \mathbf{u}_{k,M}} \right) e_{M,k} . \]
c) Application of the first element of \( \mathbf{v}_{k,M} \) to the process
\[ u_k = \begin{bmatrix} 1 \ 0_{(1 \times (M-1))} \end{bmatrix} \mathbf{v}_{k,M} . \]
If the reference trajectory is known in advance, the according reference input vector \( \mathbf{u}_{k,M}^{(d)} \) can be computed offline. Consequently, the online computational time remains unaffected.

3.1 Numerical Calculations

The analytical computation of the Jacobian \( \frac{\partial \phi_M}{\partial \mathbf{u}_{k,M}} \) becomes increasingly complex for larger values of \( M \). Therefore, a numerical approach is preferred taking advantage of the chain rule, \( i = 0, \ldots, M - 1 \)
\[ \frac{\partial \phi_M}{\partial \mathbf{u}_{k+1}^{(i)}} = \frac{\partial \phi_M}{\partial \mathbf{x}_{k+1}^{(i)}} \cdot \frac{\partial \mathbf{x}_{k+1}^{(i)}}{\partial \mathbf{x}_{k+1}^{(i)}} \cdot \frac{\partial \mathbf{x}_{k+1}^{(i)}}{\partial \mathbf{x}_{k+1}^{(i)}} \cdot \frac{\partial \mathbf{x}_{k+1}^{(i)}}{\partial \mathbf{u}_{k+1}^{(i)}} . \]
Introducing the abbreviations
\[ A_i := \frac{\partial \mathbf{x}_{k+1}^{(i)}}{\partial \mathbf{x}_{k+1}^{(i)}} = \frac{\partial f}{\partial \mathbf{x}} (x_{k+1}^{(i)}, u_{k+1}^{(i)}) , \]
\[ b_i := \frac{\partial \mathbf{x}_{k+1}^{(i)}}{\partial \mathbf{u}_{k+1}^{(i)}} = \frac{\partial f}{\partial \mathbf{u}} (x_{k+1}^{(i)}, u_{k+1}^{(i)}) , \]
the Jacobian can be computed as follows
\[ \frac{\partial \phi_M}{\partial \mathbf{u}_{k,M}} = \begin{bmatrix} A_{M-1} A_{M-2} \cdots A_1 b_1, A_{M-1} b_2 \cdots A_1 b_1, \ldots, A_{M-1} b_1 \end{bmatrix} . \]
For the inversion of the symmetric and positive definite matrix \( \mathbf{S} (\phi_M, \mathbf{u}_{k,M}) = \left( \frac{\partial \phi_M}{\partial \mathbf{u}_{k,M}} \left( \frac{\partial \phi_M}{\partial \mathbf{u}_{k,M}} \right)^T \right) \) the Cholesky-decomposition has proved advantageous in terms of computational effort.

3.2 Choice of the NMPC Design Parameters

The most important NMPC design parameter is the prediction horizon \( T_P \), which is given as product of the sampling time \( t_s \) and the constant value \( M \) as described by equation (5), see also fig. 3. Large values of \( T_P \) lead to
a slow and smooth transient behaviour and result in a robust and stable control loop. For fast trajectory tracking, however, a smaller value $T_P$ is desirable concerning a small tracking error. The choice of the sampling time $t_s$ is crucial as well: a small sampling time is necessary regarding discretization error and stability; however, the NMPC-algorithm has to be evaluated in real-time within the sampling interval. Furthermore, the smaller $t_s$, the larger becomes $M$ for a given prediction horizon, which in turn increases the computational complexity of the optimization step. Consequently, a system-specific trade-off has to be made for the choice of $M$ and $t_s$. This paper follows the moving horizon approach with a constant prediction horizon and, hence, a constant dimension $M$ of the corresponding optimization problem in contrast to the shrinking horizon approach, see Weidemann et al. (2004), Weidemann et al. (2005).

3.3 NMPC of the crane load position

The state space representation for the cage position control in $y$-direction design is given by (3). The discrete-time representation of the continuous-time system is obtained by Heun discretization

$$x_{y,k+1} = x_{y,k} + \frac{t_s}{2} \mathbf{f}(x_{y,k}, u_{y,k}) + \frac{t_s}{2} \mathbf{f}(x_{y,k} + t_s \mathbf{f}(x_{y,k}, u_{y,k}, u_{y,k}), u_{y,k}).$$

This discretization method is more precise than the Euler method, which is employed in Schindele and Aschemann (2008), and the computational effort for the NMPC-algorithm can be kept acceptable for the given system. The rope length $l$ is controlled by an underlying proportional feedback controller. The rope length in $y$-direction $l_y$ is obtained from the measured values as follows

$$l_y = \sqrt{l^2 - (y - y_k)^2},$$

and is adapted in the control algorithm for the load position at each time step.

Taking advantage of the differential flatness of the system, the ideal input $u_{y,d}(t)$ can be stated as function of the flat output variable $y$ and its first four time derivatives

$$u_{y,d} = u_{ky,d}(\dot{y}_{1,d}, \ddot{y}_{1,d}, \dddot{y}_{1,d}, \dddot{y}_{1,d}).$$

For this purpose, the desired trajectory $y_{l,d}(t)$ as well as the first four time derivatives are available from a trajectory planning module. Moreover, exploiting the differential flatness the desired system states can be obtained as function of the flat output and its first three time derivatives

$$y_{k,d} = y_{l,d} + \frac{l_y}{g} \dot{y}_{l,d}, \quad \ddot{y}_{k,d} = \ddot{y}_{l,d} + \frac{l_y}{g} \dddot{y}_{l,d}, \quad \dddot{y}_{k,d} = \dddot{y}_{l,d} + \frac{l_y}{g} \dddot{y}_{l,d}, \quad \dddot{y}_{k,d} = \dddot{y}_{l,d} + \frac{l_y}{g} \dddot{y}_{l,d}.$$ 

4. EXPERIMENTAL RESULTS

Tracking performance as well as steady-state accuracy w.r.t. the crane load position have been investigated by experiments at the crane test rig at the Chair of Mechatronics, University of Rostock. It is equipped with a separate servo-motor for each axis, see Fig. 1. In analogy to the control design for the $y$-axis, a corresponding NMPC control structure has been derived for the $x$-axis as well. The crane load position in $x$- and $y$-direction is measured by a CMOS camera using the spatial filtering principle (Schindele et al. (2009)). The velocity of the trolley is limited to $v_{kx,max} = 0.2 \frac{m}{s}$ and the velocity of the bridge is constrained to $v_{ky,max} = 0.25 \frac{m}{s}$. For the measurements, the sampling time $t_s$ was set to 10 ms, the factor $M$ in the NMPC algorithm has been chosen as $M = 140$. This results in a prediction horizon of $T_P = 1.4$ s. The active damping of load oscillations is demonstrated in Fig. 4. Here the load mass is disturbed by an impact. The resulting load positions and angles in both directions are shown together with the corresponding control inputs. Furthermore, a sequence of diagonal movements of the crane load has been specified in the $xyz$-plane, see Fig. 5. The desired trajectories for the crane load position and their corresponding time derivatives are obtained from a trajectory planning module that provides synchronous time optimal trajectories. Here the desired $x$- and $y$-positions vary in an interval from 0.1 m to 0.8 m, whereas the desired $z$-position varies between 0.2 m and 0.4 m. The resulting tracking performance is shown in Fig. 5. The maximum errors of the crane load position in $x$-$y$-direction, $e_x/e_y$ are approximately 6 mm. The maximum error in $z$-direction is below 1 mm. The benefit of including the input constraints by an exterior penalty function in the NMPC algorithm is depicted in Fig. 6. In the left part of this figure, the step response corresponding to a step in the desired crane load position from $y_{l, d} = 0.4$ m to $y_{l, d} = 0.1$ m is shown. A consideration of the given input constraints leads to a faster convergence to the desired load position and an improved damping of the load oscillations. In the right part of Fig. 6, the tracking of a desired trajectory that intentionally violates the given input constraints is illustrated. It becomes obvious that by an explicit consideration of the input constraints the tracking error can be significantly reduced.

5. CONCLUSION

This paper presents a gain-scheduled fast nonlinear model predictive control strategy for the translational axes of an overhead travelling crane. The suggested control algorithm aims at reducing the future tracking error at the end...
Fig. 4. Control behaviour after an external disturbance: crane load positions \((x_l, y_l)\) (upper part), rope angles \((\varphi_x, \varphi_y)\) (middle part) and control inputs \((v_{kx}, v_{ky})\) (lower part).

Fig. 5. Synchronized movement in the \(xyz\)-workspace: desired trajectories for the crane load position (left part) and corresponding tracking errors (right part).
Fig. 6. Comparison of the control behaviour with and without consideration of input constraints in the NMPC algorithm. Left part: step response, right part: trajectory control.

of the prediction horizon. This NMPC approach can be computed within very small sample times and is well suited for systems with fast dynamics. A further advantage of the presented control algorithm is the possibility to account for input constraints. The efficiency of the proposed control is shown by experimental results realized at a test rig at the University of Rostock, involving tracking of desired trajectories within the $xyz$-plane. The maximum tracking errors of approximately 6 mm emphasize the excellent closed-loop performance.

REFERENCES

Aschemann, H. (2002). *Optimale Trajektorienplanung sowie modellgestützte Steuerung und Regelung für einen Brückenkran (in German).* Fortschrittsberichte VDI, Reihe 8, Nr. 929. VDI Verlag, Düsseldorf, Germany.


