Mechatronics 21 (2011) 1203–1212

Contents lists available at SciVerse ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

A model predictive control approach for time optimal point-to-point motion control

Lieboud Van den Broeck [a,*], M. Diehl [b], J. Swevers [a]

[a] Department of Mechanical Engineering, Celestijnenlaan 300 B, B-3001 Leuven, Belgium
[b] Department of Electrical Engineering, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

ARTICLE INFO

Article history:
Received 9 November 2010
Accepted 24 July 2011
Available online 3 September 2011

Keywords:
Model-based control
Time-optimal control
Mechanical systems

ABSTRACT

This paper presents a new model predictive control method for time-optimal point-to-point motion control of mechatronic systems. The formulation of time-optimal behavior within the model predictive control framework and the structure of the underlying optimization problem are discussed and modifications are presented in order to decrease the computational load of the numerical solution method such that sampling rates in the millisecond range and long prediction horizons for large point-to-point motions are feasible. An extensive experimental validation on a linear motor drive and an overhead crane setup demonstrates the advantages of the developed time-optimal model predictive control approach in comparison with traditional model predictive control.

1. Introduction

Most mechatronic systems are controlled using linear feedback controllers, e.g. traditional PID controllers [1] or more advanced model-based controllers, like e.g. H_∞ robust controllers [2] or internal model controllers [3]. Their main advantage is their simplicity and for numerous applications, linear controllers are perfectly suited and can be well-tuned. Their main disadvantage however, is their inability to account for constraints on inputs, outputs and states. Hence linear controllers can not cope well with applications where time optimality is required within stringent input constraints, except if they are combined with reference trajectories that take into account these constraints. Two different time-optimal applications can be considered: time-optimal trajectory tracking like e.g. [4] and time-optimal point-to-point or setpoint control. This paper considers the latter application, which means that a desired endpoint is defined without specifying an intermediate trajectory. Hence, in order to achieve time-optimal point-to-point motion with a linear controller, a point-to-point reference trajectory has to be designed first. Demeulenaere et al. [5] present a polynomial spline based reference trajectory optimization approach. The method can be applied to any linear time invariant system, input, output and state constraints can be accounted for, and time-optimality is achieved by solving a sequence of feasibility problems. Also, Henrion and Lasserre [6] present a method to compute a polynomial reference trajectory which can take into account constraints on inputs and outputs. Although the formulated trajectory optimization problems are either linear, quadratic or LMI problems [7], and hence can be solved typically within one second, the method is an off-line method, meaning that either all reference trajectories have to be optimized beforehand, or references are generated during motion by interpolating between a limited set of optimized trajectories. In the latter case, time-optimality and constraint satisfaction cannot be guaranteed. Besides this off-line approach, several on-line approaches exist, however none of them can guarantee time-optimality and constraints satisfaction for all possible point-to-point motions. Input shapers [8–12] are linear filters that generate reference trajectories aiming at minimal residual vibrations. These filters can be designed to yield time-optimal behavior for one particular reference step. However, if they are applied to smaller or larger reference steps, the resulting reference trajectories are either conservative or yield input constraint violation. Alternatively, to obtain near time optimality over a wider range of step references, these prefilters which compensate for higher order vibrations modes can be combined with an optimized rigid body reference trajectory [13]. [14,15] present strategies that calculate reference trajectories which satisfy constraints on velocity, acceleration and eventual higher derivatives. Hence, these methods cannot take input constraints into account directly. In addition, time-optimality can only be guaranteed for specific point-to-point motions that e.g. include a constant velocity part and for systems of which the order is limited to four. None of these on-line methods can cope well with the situation where a new reference step is requested while still executing the previous step. Model predictive control (MPC) is more appropriate for these applications since it can take system constraints explicitly into account. MPC algorithms calculate future control actions by solving at each sampling time an optimization problem specified over a certain prediction horizon for a given system model, a given estimate of the current system state and reference signal, and taking into account constraints on inputs, outputs and states. The main

* Corresponding author. Tel.: +32 16 322833.
E-mail address: lieboud.vandenbroeck@mech.kuleuven.be (L. Van den Broeck).

0957-4158/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2011.07.008

drawback of MPC is the high computational load that stems from this real-time optimization which introduces computational delays and therefore a higher response time than linear controllers.

MPC became very popular in the eighties [16,17], mostly for process control applications which have less stringent real-time requirements due to large sampling periods in the order of seconds or minutes [18]. Since the end of the nineties, research on the development of fast MPC solution methods (fast MPC in short) has intensified to extend its application area to faster systems such as mechatronic or motion systems. One fast MPC approach is called explicit MPC [19], which stores exact precomputed optimal control solutions, depending on the current state. This approach is applied to numerous control problems, however the system dimensions and number of control horizon time steps are typically limited to 5 each, due to memory and search space limitations. Another fast MPC approach is using fast numerical solution methods such as qpOASES [20] to solve the optimal control problem in real-time. This approach has been applied successfully to a diesel engine and a cantilever beam with up to 10–20 state variables and prediction horizons of 10–12 time-steps [21,22]. This paper discusses the application of fast MPC based on qpOASES for time-optimal point-to-point motion control of mechatronic systems.

In order to control the transient behavior between two setpoints, MPC optimizes a least squares objective function weighting the input cost against the response speed. However, for most mechatronic applications, the input cost is negligible and time optimality is the main goal. Hence, for these applications an MPC formulation that aims at time optimal system control is required, i.e. achieving a minimal settling time while respecting the constraints on inputs and states. This time optimal behavior has to be combined with local quadratic cost function behavior, typical for traditional MPC, in order to reduce the sensitivity to measurement noise close to the setpoint. This paper proposes a scheme addressing both aims, called Time Optimal MPC (TOMPC). Zhao et al. [23] develops a controller for robots with similar requirements; however, they worked in continuous time and used a rescaling of time with discretized controls for their numerical solution, making convergence proofs impossible. It should also be noted that even for continuous time approaches, time optimality can only be approximated from above. Grieder and Morari [24] discusses an explicit MPC approach which drives the controlled system to an invariant feasible set in a minimal number of steps. They defined for their application the invariant feasible set as a set where a linear controller respects all system and input constraints and keeps the controlled system state inside this set. However, inside this set no time optimality is imposed. Van den Broeck et al. [25] develop an open-loop controller that minimizes the settling time while respecting the system constraints. However, time optimality is not directly imposed as in this paper, but only approximated by imposing an exponential weighting on the absolute value of the positioning error in order to increase numerical efficiency. Moreover, because no feedback is incorporated, this controller can not account for external disturbances and hence can not eliminate steady state positioning errors caused by model-plant mismatch and static friction.

This paper is organized as follows. Section 2 presents the idea of TOMPC. Section 3 discusses how the structure of the formulated optimization problem can be exploited to achieve real-time implementation. Section 4 validates TOMPC experimentally on two representative mechatronic systems, a linear motor drive and an overhead crane, and shows the merits of the designed controller. This section also compares TOMPC with traditional MPC. Section 5 concludes the paper.

The TOMPC is developed in discrete time, and hence all system models are discrete. Moreover, $\bar{x}_l$ and $\bar{u}_l$ denote the measured or estimated system state and input at time l, while x_k and u_k denote the predicted system state and input within the prediction horizon at step k.

The main contributions of this paper are the formulation of time optimality within the MPC framework, the efficient real-time implementation of this closed loop controller and the experimental validation on two representative mechatronic setups.

2. Time optimal MPC

This section presents the TOMPC approach and proves that the resulting control law is asymptotically converging.

2.1. Model predictive control

MPC is an advanced control technique which determines the control action by solving on-line, at every sampling time l, an open-loop optimal control problem, based on the current state of the system $\bar{x}_l$. The optimization generates an input sequence for a specified time horizon N. However, only the first input $\bar{u}_l = u_0$ is applied to the system. In its simplest setting, the optimization problem to be solved at each time step l is:

$$V_A^{\star}(\bar{x}_l, N) = \min_{\substack{x_0,\ldots,x_N \\ u_0,\ldots,u_{N-1}}} \sum_{k=0}^{N-1} \|u_k - u_{\mathrm{ref}}\|_R^2 + \|x_k - x_{\mathrm{ref}}\|_Q^2, \tag{1a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_l, \tag{1b}$$

$$x_{k+1} = f(x_k, u_k), \tag{1c}$$

$$g(x_k, u_k) \geqslant 0 \quad k \in [0, N-1], \tag{1d}$$

$$x_N = x_{\mathrm{ref}}, \tag{1e}$$

where $(x_0, \ldots, x_N, u_0, \ldots, u_{N-1})$ are the optimization variables over time horizon N and $\bar{x}_l$ is the system state at time l (1b). The optimization is based on a system model (1c) and acknowledges bounds on inputs, outputs and internal states (1d). Constraint (1e) is a moving endpoint constraint and requires the system to be at rest at the desired setpoint at time N. This constraint hence guarantees stability, assuming semi-definite positivity of the weights Q and R [26]. Vectors x_{ref} and u_{ref} express together a feasible equilibrium point; i.e. they satisfy $g(x_{\mathrm{ref}}, u_{\mathrm{ref}}) \geqslant 0$ and $x_{\mathrm{ref}} = f(x_{\mathrm{ref}}, u_{\mathrm{ref}})$, which is together with an output function also the defining function of these values, see e.g. [27]. The input is optimized considering the Euclidean norm of the system input and state, weighted with R and Q respectively. Fig. 1 summarizes the idea of MPC. Often, in contrast to the setting here, the control and prediction horizon do not have the same length to keep this problem computationally feasible, see e.g. [28]. Note that many other MPC schemes with guaranteed stability exist, without using terminal equality constraint (1e), see [29]. Also, like many linear controllers MPC has inherent robustness

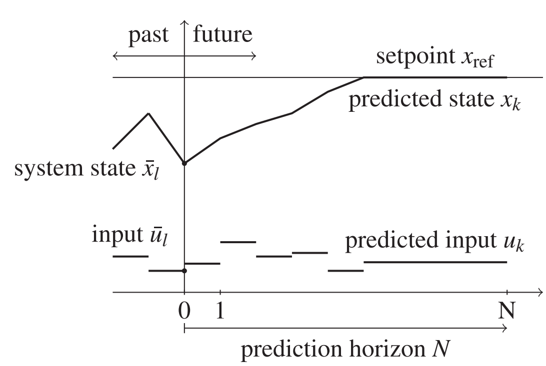

Fig. 1. General idea of MPC. At each time step l, the state $\bar{x}_l$ of the system is measured. Then, the optimal input is computed such that an objective function is optimized over the prediction horizon N.

properties [30] and by formulating the problem in differential inputs Δu instead of the inputs u inherent integral action can be introduced [31].

2.2. Time optimal MPC

In traditional MPC, there is an assessment between input energy and tracking error through the matrices R and Q. By choosing appropriate values of these matrices, a trade-off between input energy and tracking error can be made, resulting in different behaviors. However, in many mechatronic applications, time optimal behavior is usually more important than energy efficiency. Time optimal behavior can not be attained through an Euclidean norm based objective function. It requires a different MPC formulation, presented in this paper and called Time Optimal MPC. TOMPC is formulated as a two layer optimization problem. At every time sample the settling time is minimized, where settling time is defined as the time required for the system to be at rest at the desired setpoint.[1] This minimization is constrained by a specified minimal value (denoted below by $N_{\min}$). If this minimal settling time is obtained, a regulating objective function similar to traditional MPC (1a) is minimized. Hence, at each time step l the following two layer optimization problem is defined:

'Problem A': First, (1) is defined as 'Problem A', and denoted by $P_A(\bar{x}_l, N)$, to stress its dependence on the initial value $\bar{x}_l$ and the horizon length N. Objective function $V_A^{\star}$ is extended to $V_A^{\star} = \infty$ if $P_A(\bar{x}_l, N)$ is infeasible. The physical meaning of infeasibility of P_A is that the system can not settle at the reference point x_{ref} in N time steps while respecting all constraints. Therefore, an admissible set $\mathbb{X}(N)$ is defined:

$$\mathbb{X}(N) = \{\bar{x}_l | P_A(\bar{x}_l, N) \text{ is feasible}\} = \{\bar{x}_l | V_A^{\star}(\bar{x}_l, N) \text{ is finite}\}. \tag{2}$$

$\mathbb{X}(N)$ is the set of system states from which the setpoint can be reached within N time steps, while respecting all system constraints (1d).

'Problem B': Second, 'Problem B', which is denoted by $P_B(\bar{x}_l)$, is defined as follows:

$$V_B^{\star}(\bar{x}_l) = \min_{N \in \mathbb{N}} \quad N, \tag{3a}$$
$$\text{s.t.} \quad N \geqslant N_{\min}, \tag{3b}$$
$$N \leqslant N_{\max}, \tag{3c}$$
$$\bar{x}_l \in \mathbb{X}(N), \tag{3d}$$

where N is the required settling time, $N_{\max}$ is the maximal optimization horizon and $N_{\min}$ is a minimal bound on N which allows to reduce the sensitiveness with respect to measurement noise, which is discussed in Section 2.4. Note that the level sets of $V_B^{\star}$ in $\bar{x}_l$ space are plateaus of height N:

$$V_B^{\star}(\bar{x}_l) = \begin{cases} \infty \text{ if } V_A^{\star}(\bar{x}_l, N_{\max}) = \infty \\ N \text{ if } V_A^{\star}(\bar{x}_l, N) < \infty \;\&\; V_A^{\star}(\bar{x}_l, N-1) = \infty \\ \qquad \& N \geqslant N_{\min} \& N \leqslant N_{\max} \\ N_{\min} \text{ if } V_A^{\star}(\bar{x}_l, N_{\min}) < \infty \end{cases} \tag{4}$$

Fig. 2 illustrates the level sets for a system with 2 states.

At each time step, problem P_B which minimizes the settling time N is optimized. As problem P_A is underlying P_B, also a traditional MPC problem with endpoint constraints is optimized if optimization freedom is left, i.e. if $N = N_{\min}$ or if multiple time optimal solutions exist. This two-layer optimization problem therefore produces the desired functionality: the system approaches the setpoint as fast as possible while respecting the constraints on inputs and states, and close to the setpoint the system reacts as traditional MPC on measurement noise and disturbances. The choice of the weights Q and R in (1) can therefore be tuned to obtain the desired regulating behavior in the neighborhood of the setpoint.

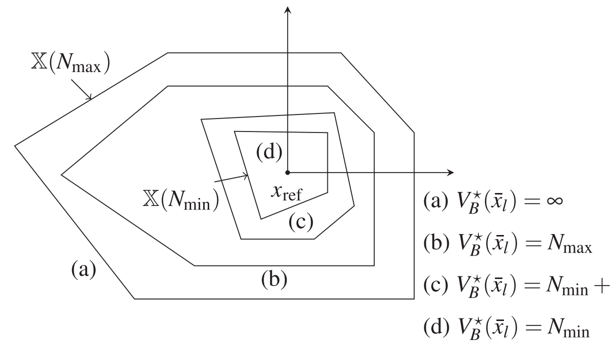

Fig. 2. Visualization of level sets of $V_B^{\star}$ in $\bar{x}_l$ space.

2.3. Proof of asymptotic convergence

Lemma 1. *If $(\tilde{x}^l, \tilde{u}^l) = (x_0, x_1, \ldots, x_N, u_0, u_1, \ldots, u_{N-1})$ solves $P_A(\bar{x}_l, N)$, with $x_0 = \bar{x}_l$, then $(x_1, \ldots, x_N, u_1, \ldots, u_{N-1})$ solves $P_A(x_1, N-1)$*

Proof. This lemma is a corollary of Bellman's principle of optimality of subarcs [32]. □

Lemma 2. *For an unperturbed system without model-plant mismatch and assuming uncorrupted plant measurements, feasibility at time $l = 0$ guarantees feasibility at each time $l > 0$.*

Proof. By assumption, a feasible solution of $P_A(\bar{x}_0, N)$ exists at $l = 0$, which is denoted by $S_0 = (\tilde{x}^0, \tilde{u}^0)$. The closed loop control action at the first time step is hence the first element of $\tilde{u}_0$, indicated as $\tilde{u}_0^0$. At time $l = 1$, a new TOMPC optimization problem is formulated with initial state $\bar{x}_1$, where $\bar{x}_1 = \tilde{x}_1^0$ because of the assumption of no model-plant mismatch and no disturbances. Hence, if $N > N_{\min}$, a feasible solution of $P_A(\bar{x}_1, N-1)$ exists because of Lemma 1. If $N = N_{\min}$, then because of endpoint constraint (1e), $\left([\tilde{x}^{0T}_{(1\ldots N)}, x^T_{\text{ref}}]^T, \left[\tilde{u}^{0T}_{(1\ldots N-1)}, u^T_{\text{ref}}\right]^T\right)$ is a feasible point of $P_A(\bar{x}_1, N_{\min})$. The lemma follows by induction. □

Theorem 1. *For each $\bar{x}_0 \in \mathbb{X}(N_{\max})$, TOMPC generates a closed-loop response for an undisturbed system without model-plant mismatch, asymptotically attracted by x_{ref}, with $N_{min} \geqslant 1$.*

Proof. First, consider $\bar{x}_l \notin \mathbb{X}(N_{\min} + 1)$. It is clear from Lemma 2, that if a feasible solution with $V_B^{\star}(\bar{x}_l) = N$ exists at time step l, there exists a feasible solution at time step $l + 1$ with $V_B^{\star}(\bar{x}_l) = N - 1$. Hence, every time step, the system will be driven closer to $\mathbb{X}(N_{\min})$. Second, consider $\bar{x}_l \in \mathbb{X}(N_{\min})$. The TOMPC reduces then to a traditional MPC with endpoint constraints. It can easily be proven that the system controlled by this local controller is then asymptotically attracted to x_{ref}, see e.g. [26]. □

2.4. Choice of N_{min}

The formulation of the TOMPC optimization problem requires the choice of $N_{\min}$. To guarantee unconstrained solvability, N

[1] This is not the standard definition of settling time in linear control literature and is only meaningful because TOMPC is a nonlinear controller.

should always be bigger than n/n_u with n the number of states and n_u the number of inputs [33]. In order to make the controller less sensitive to measurement noise and to avoid too aggressive behavior when the setpoint is reached, a higher value of $N_{\min}$ is recommended, e.g. two to three times larger. However, $N_{\min}$ should not be chosen too high, as this would destroy the time optimal behavior. Fig. 3 illustrates the degradation of settling time as a function of $N_{\min}$ for a second order system. The noise insensitivity is clearly illustrated in Section 4 on the linear motor drive.

It should be noted that time optimal behavior can be approximated by tuning the weights Q and R very aggressively, i.e. Q much larger than R. However, even by putting $R = 0$, the same behavior can not be obtained. Moreover, this controller would also react very aggressively on noise. By introducing a two-layer optimization problem with endpoint constraints, these problems can be avoided, which is shown in Section 4.

3. Real-time implementation

This section discusses the real-time implementation of the two level TOMPC optimization problem defined in Section 2.2. First, the TOMPC optimization problem is further analyzed, followed by a brief description of the active set strategy [34] used to solve the underlying quadratic problems (QPs). Then, it is explained how the structure of the problem is exploited in order to yield a TOMPC implementation that is sufficiently efficient for some mechatronic applications. Finally, it is discussed how the feasibility region of the controller can be extended, without compromising on the computational speed.

3.1. Problem formulation

In general, optimization problem (1) is a non-convex problem. However, if the system dynamics (1c) and all constraints (1d) are linear, this problem reduces to a convex QP:

$$V_A^{\star}(\bar{x}_l, N) = \min_{\substack{x_0,\dots,x_{N_{\max}} \\ u_0,\dots,u_{N_{\max}-1}}} \sum_{k=0}^{N_{\max}-1} \|u_k - u_{\text{ref}}\|_R^2 + \|x_k - x_{\text{ref}}\|_Q^2, \tag{5a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_l, \tag{5b}$$

$$x_{k+1} = Ax_k + Bu_k, \tag{5c}$$

$$Hx_k + Gu_k \geqslant e \quad k \in [0, N_{\max} - 1], \tag{5d}$$

$$x_k = x_{\text{ref}} \quad k = N, \tag{5e}$$

$$u_k = u_{\text{ref}} \quad k \in [N, N_{\max} - 1]. \tag{5f}$$

Note that in this formulation, extra variables $(u_N, \dots, u_{N_{\max}-1})$ and $(x_{N+1}, \dots, x_{N_{\max}})$ are added which will be beneficial for the numerical solution as shown in Paragraph Section 3.2. The paper will henceforth implement, exploit the problem structure and validate this type of controller for a linear system. It should be noted however, that this kind of controller can be applied for non-linear systems as well and the asymptotic convergence proof will still be valid. The real-time implementation is however even more challenging for the non-linear case. For this linear system, the definition of the reference state is:

$$x_{\text{ref}} = Ax_{\text{ref}} + Bu_{\text{ref}}. \tag{6}$$

Depending on the value of $\bar{x}_l$, two different optimization problems are to be solved. If $\bar{x}_l \in \mathbb{X}(N_{\min})$, the problem reduces to a regular MPC problem with endpoint constraints, which is under the above assumptions a QP. This QP can efficiently be solved using an on-line Active Set Method [20], see Paragraph Section 3.2. If $\bar{x}_l \notin \mathbb{X}(N_{\min})$ the resulting optimization problem $P_B(\bar{x}_l)$ is a mixed integer problem in one integer variable N. This problem can be solved by a series of feasibility problems $P_A(\bar{x}_l, N)$, i.e. a series of QPs. Algorithm 1 summarizes this procedure.

Algorithm 1. Optimization procedure

input: $\bar{x}_l$
output: $u^{\star}$ or error 'infeasible problem'
start with initial guess for N
solve QP problem $P_A(\bar{x}_l, N)$ (5)
if $P_A(\bar{x}_l, N)$ feasible **then**
  while $P_A(\bar{x}_l, N)$ was feasible **do**
    store $u^{\star} = u_0(\bar{x}, N)$
    $N = N - 1$
    if $N \geqslant N_{\min}$ **then**
      solve QP-problem $P_A(\bar{x}_l, N)$
    else
      break
    end if
  end while
else
  while $P_A(\bar{x}_l, N)$ was infeasible **do**
    $N = N + 1$
    if $N \leqslant N_{\max}$ **then**
      solve QP-problem $P_A(\bar{x}_l, N)$
    else
      error 'infeasible problem'
    break
    end if
  end while
  store $u^{\star} = u_0(\bar{x}, N)$
end if

3.2. Online active set methods

In the real-time implementation of the TOMPC, the above mentioned QP's are solved using an online active set method [20]. Therefore, this paragraph gives first a short introduction to active set methods (see also e.g. [35,36]). They are general methods for solving convex quadratic optimization problems:

$$\min_w \quad w^T H w + g^T w \tag{7a}$$

$$\text{s.t.} \quad Aw \leqslant b, \tag{7b}$$

with w the vector of optimization variables, (7a) the objective function, (7b) the constraints under which the objective function is optimized and H a positive definite matrix. Introducing dual variables λ, the Karush–Kuhn–Tucker (KKT) conditions for optimality can be formulated as stated in [34]. An active set method makes a guess of the active set in the optimum, i.e. the set of all inequality con-

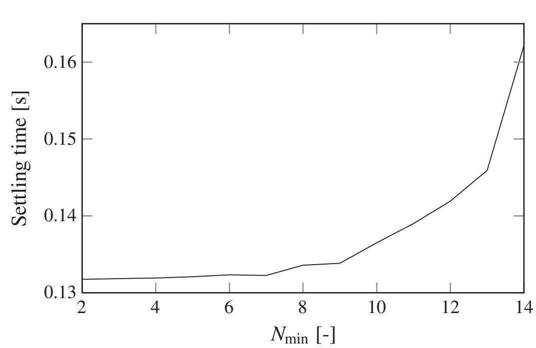

Fig. 3. The 1% settling time for a second order system as a function of the lower bound on the settling time $N_{\min}$.

straints which become equality constraints in the optimum. Subsequently, the corresponding equality constrained QP is solved, resulting in a solution $(x^{\star}, \lambda^{\star})$. The method then verifies whether the KKT conditions are satisfied. If they are not, constraints are added to or removed from the active set and the procedure is repeated until the optimal solution is found. As already shown in Eq. (5), problem P_A is reformulated into an equivalent QP in the variables $(x_0, \ldots, x_{N_{\max}})$ and $(u_0, \ldots, u_{N_{\max}-1})$ which have hence a constant size allowing to use the online active set strategy as implemented in qpOASES [20]. This online strategy works most efficiently if a condensed version of (5) is used, i.e. if all state variables are eliminated using equalities (5b) and (5c). The strategy exploits the fact that all QP matrices stay constant and only gradient and constraint vectors change. It keeps all matrix factorizations from one problem to the next and performs a linear homotopy, resulting in a low number of active set changes compared to conventional ways of hotstarting [20].

The following two paragraphs discuss how to speed up the solution method. First, by hotstarting the active set method on a given time step based on the solution from the previous time step, which is an application of the technique developed in [20]. Second, by reformulating endpoint constraints (5e) and (5f) to optimize N with a minimum number of active set changes.

3.3. Transition over time using an online active set strategy

When the controller propagates from time step l to $l+1$, two different scenarios are possible:

- The setpoint does not change: Because the system state does not change much during one time step, neither will the optimal solution S. Hence, the number of active set changes compared with this previous solution is limited. By hotstarting the Active Set Method from this previous solution S_l, the new solution S_{l+1} can be found efficiently.
- The setpoint changes: Because the desired setpoint changes, the number of active set changes compared with the previous solution S_l can be high. Hence, the optimization will not reuse the previous solution. However, this problem can be hot started by starting from a static controller, which has properties similar to the desired controller [12], or by making an educated guess of the optimal value of N based on simulations which are performed beforehand. These optimal guesses of N for a set of reference step lengths are stored in a table which is available for the TOMPC algorithm. When a new reference step is applied, the closest upper approximation is selected from the table as an initial guess for N.

3.4. Optimization of N with a minimum number of active set changes

Once $N = N_{\min}$, the QP-solver only has to solve one regular QP (5). However, if $N > N_{\min}$, a series of optimization problems has to be solved within one sampling time step, until the optimal value $N^{\star}$ is found. In these optimization problems, N is increased or decreased by one as a good guess of N is available (see Section 3.3) and because by changing the value of N by only one the optimization problem structure can be maximally exploited, which is illustrated further in this section. This results each time in at least $2n + n_u$ active set changes as the deactivation of constraint (5e) for $k = N$ requires n active set changes, the subsequent activation of (5e) for $k = N \pm 1$ requires another n active set changes, and the deactivation or activation of constraint (5f) requires analogously n_u active set changes. This paper proposes a different approach resulting in less active set changes for each transition from $k = N$ to $k = N \pm 1$; First, one single output which yields an observable state space model is introduced. This output can be the controlled output, as in Section 4, but this is not strictly necessary. If more outputs would be needed to obtain observability, a straightforward modification of the approach described below has to be applied. Second, (5) is extended by introducing extra variables $(u_{N_{\max}}, \ldots, u_{N_{\max}+n-2})$ and $(x_{N_{\max}+1}, \ldots, x_{N_{\max}+n-1})$ and constraints (5e) and (5f) are replaced by:

$$y_{N+k} = y_{\text{ref}} \quad \text{for} \quad k = 0, \ldots, n-1, \tag{8a}$$
$$u_{N+k} = u_{\text{ref}} \quad \text{for} \quad k = 0, \ldots, N_{\max} - N - 1, \tag{8b}$$

where $y_{\text{ref}} = Cx_{\text{ref}}$ is the output corresponding to the steady state x_{ref}. Lemma 3 proves that these sets of constraints are equivalent.

Lemma 3. *If the observability matrix C_o:*

$$C_o = [C^T \; A^T C^T \ldots (A^{n-1})^T C^T]^T \tag{9}$$

is of full rank, requirements (8) are equivalent with (5e) and (5f).

Proof. (5e)–(5f) ⇒ (8): trivial.

(8) ⇒ (5e) and (5f): All constraints on u are the same.

Because of the system dynamics (5c), it is known:

$$x_{N+1} = Ax_N + Bu_N \tag{10a}$$
$$= Ax_N + Bu_{\text{ref}} \tag{10b}$$
$$= A(x_N - x_{\text{ref}}) + x_{\text{ref}} \tag{10c}$$
$$\iff x_{N+1} - x_{\text{ref}} = A(x_N - x_{\text{ref}}), \tag{10d}$$

where (10a) is equivalent to (10b) because of constraint (8b) and (10b) is equivalent to (10c) by definition of x_{ref} (6). By induction, (10d) can be extended to:

$$x_{N+k} - x_{\text{ref}} = A^k (x_N - x_{\text{ref}}). \tag{11}$$

Hence, (11) allows to rewrite (8a) into:

$$C(x_{N+k} - x_{\text{ref}}) = CA^{k-1}(x_N - x_{\text{ref}}) = 0 \quad \text{for} \quad k = 0, \ldots, n-1, \tag{12}$$

which is equivalent to:

$$[C^T \; A^T C^T \ldots (A^{n-1})^T C^T]^T (x_N - x_{\text{ref}}) = 0. \tag{13}$$

Since the observability matrix (9) is of full rank, (13) has only one solution $x_N \equiv x_{\text{ref}}$. □

The advantage of formulation (8) in comparison with constraints (5e) and (5f) is that only $2 + n_u$ instead of $2n + n_u$ constraints have to be added to or removed from the active set when the value of N increases or decreases by one, which improves the numerical efficiency considerably. For example, for the first test case considered in Section 4 and a reference step of 10 cm, the worst case computation time (which typically occurs at the time sample when a new reference step is applied) reduces with a factor 4. In addition to this measure to reduce the computation time, the similarity between two consecutive problems can be exploited. Since the value of N changes by one in the series of optimization problems, the optimal solution of each problem has $2 + n_u$ known active set changes, which can be imposed directly and simultaneously [37]. E.g. if N is reduced to $N-1$, the following active set changes can be imposed: fix $u_{N-1} = u_{\text{ref}}$, fix $y_{N-1} = y_{\text{ref}}$ and relax y_{N+n-1}. By imposing these active set changes directly, and starting from the corresponding active set, the optimization procedure is accelerated further, resulting in an extra reduction of 40% of the worst case computation time for the same test case considered above. In comparison to an interior point solver as e.g. Mosek [38], the worst case computation time is 10 times smaller. Table 1 shows the relative comparison of worst case computation times in Matlab, for the considered case using Mosek and the three dicussed alternative implementations of TOMPC using qpOASES as a solver. These simulations are performed on a non-dedicated

Table 1
Relative comparison of worst case computation times in Matlab for the linear motor drive system discussed in Section 4 for a reference step of 10 cm, using four different solution methods.

Method	Computation time (–)
Mosek	10.17
qpOASES based on (5e) and (5f)	8.25
qpOASES based on (8a) and (8b)	1.83
qpOASES based on (8a) and (8b) with directly imposed active set changes	1

computer with a 2 GHz processor and 1.99 GB RAM. It should be noted that in the real-time implementation of the TOMPC controller, the optimization procedure may be terminated prematurely due to computation time limitations if a bad guess of $N^{\star}$ is used as also implemented in qpOASES.

Combination of the above measures to speed up the solution yields that the optimization problem underlying the TOMPC can be solved in real-time at 200 Hz and 60 Hz for respectively the linear drive system and the overhead crane as discussed in Section 4.

It should be noted that MPC based on active set numerical solution methods has been applied with sampling frequencies in the kHz-range [21]. The sampling frequencies and hence speed of response obtained with TOMPC are considerably lower. This is (i) because in the cases considered in this paper, the number of decision variables and constraints are respectively approximately 4 and 10 higher than in the case considered in [21], and (ii) because TOMPC has to solve a sequence of feasibility problems in order to obtain time optimal behavior.

3.5. Efficient extension of the prediction horizon

A disadvantage of the above described TOMPC algorithm, is the dependence of the largest possible reference step on the value of $N_{\max}$ because endpoint constraints (5e) and (5f) must be satisfied for $N \leqslant N_{\max}$, that is, the system must be able to reach this largest possible reference position in no more than $N_{\max}$ time steps without violating the system constraints (5d). $N_{\max}$ itself determines the size of the optimization problem and hence the worst case computation time of the TOMPC solution method which is limited by the sampling period. Hence, the sampling period limits the largest possible reference step, which can be small for systems that require high sampling frequencies. In order to overcome this limitation, non-equidistant time steps or time gridding is applied such that larger horizons can be considered without increasing the total number of discretization points $N_{\max}$ and hence the number of optimization variables. In the first part of the horizon up to typically $N_{\min}$, the time step corresponds to the sampling period. Thereafter, the time steps are gradually increased up to 10 times the sampling time. For the overhead crane test setup, this allows us to increase the prediction horizon by a factor of 4, extending the attainable range to its maximum of 70 cm.

It should be noted that the rigorous asymptotic convergence results presented in Section 2.3 do not hold strictly when a non-equidistant grid is used. However, the practical closed-loop performance is nearly identical as when a fine equidistant grid is used for the same horizon length, while the computation time is significantly reduced.

4. Validation

This section validates TOMPC on a linear motor drive and an overhead crane. First, in Section 4.1, the linear motor drive setup is presented, followed by an experimental validation of TOMPC. Section 4.2 discusses the application of TOMPC on an overhead crane, showing the capabilities of this technique to control a system with a highly undamped mode.

4.1. Linear motor drive

4.1.1. Test setup

The considered test setup is the linear motor shown in Fig. 4. The control input to the system is the reference motor current applied to the internal current loop [A], and the output is the position of the linear drive [m] measured with a linear encoder (LIA 21 of Numerik Jena) with a resolution of 2 nm. Due to peak current limitations, the input is limited to ±17 A, and the input slew rate is limited to ±2 A/ms by the bandwidth of the internal current controller. The system controllers are embedded on a SpeedGoat real-time target machine which contains a 2.13 GHz processor with 1 GB RAM and they are implemented through C++ functions in the XPC-target environment of Simulink. A discrete-time third order state space model with a sampling period of $T_s = 5$ ms is identified for this system, using a nonlinear least squares frequency domain identification approach based on frequency response function (FRF) measurements that are obtained from multisine excitations with a frequency content between 0.01 Hz and 15 Hz [39]. The system model contains one sample delay and one integrator pole at $z = 1$. This model is used for both traditional MPC and TOMPC.

4.1.2. Controller setup

TOMPC and MPC are implemented taking the input constraints described above into account, and in addition an overshoot and undershoot constraint of 10% of the reference step is imposed. The TOMPC controller is designed with a prediction horizon $N_{\max} = 45$ and weights $Q = 4\ \mathrm{m}^{-1}$ and $R = 5\ \mathrm{A}^{-1}$. The underlying optimization problem P_A contains 272 constraints (5d) (92 for overshoot and undershoot, and 90 for respectively the input u and the slew rate of the input Δu). It should be noted that the aggressive time optimal behavior is obtained by the shrinking horizon N and not by the choice of the weights Q and R. Hence, these weights can be selected by considering sensor noise sensitivity and disturbance regulation only, e.g. according to general tuning rules [1], taking into account the range of the different variables, which can be normalized by applying a proper scaling. This is not the case for the traditional MPC, where Q and R also determine the response speed. The traditional MPC is tuned very aggressively by trial-and-error, with a much higher weight on the output error than the input cost, in order to obtain approximately time optimal behavior (a Q–R ratio of 400,000 A/m). Since only the motor position is measurable, the state variables are determined using a state

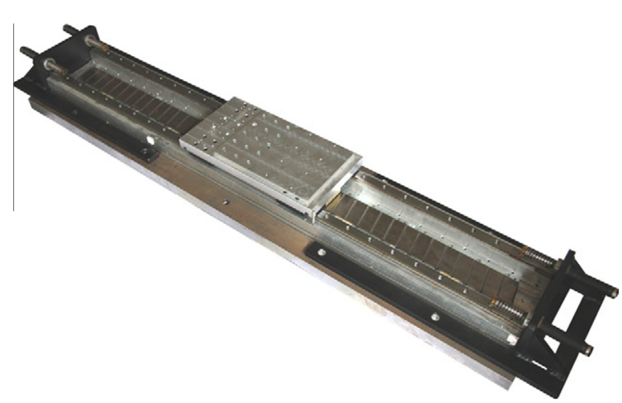

Fig. 4. Picture of the linear motor drive test-setup in the PMA-lab at K.U. Leuven.

estimator, which is designed using pole placement with a bandwidth of approximately 85 Hz.

4.1.3. Experimental validation and comparison with MPC on a linear motor drive

TOMPC and MPC are compared and validated experimentally on the given linear motor drive system. Three sets of experiments are performed. In a first set of experiments, TOMPC is compared with traditional MPC for several different displacements. The resulting 1% settling-times are summarized in Table 2. Fig. 5 shows the output motion for a step of 10 cm. Although the rise time obtained with MPC (gray solid line) is even slightly lower than with TOMPC (black solid line), the settling time with TOMPC is considerably reduced. This is more clearly shown in Fig. 6, which presents the output error relative to the applied step on a logarithmic scale. The remaining output error is due to model-plant mismatch and nonlinear disturbances like cogging and friction. This static error can be eliminated by introducing disturbance states, see e.g. [27], which is similar to introducing integrating action in linear controllers. Fig. 7 shows the corresponding inputs to the system. Comparable inputs are applied during the first part of motion, however when the system settles, TOMPC (black line) obtains a different control input than traditional MPC (gray line). This similar behavior during the first part of the motion (up to 0.13 s) is because both controllers hit the input constraints during acceleration and deceleration (because of respectively the endpoint constraint and high Q–R-ratio). When this is not the case, the controllers behave differently.

A second set of experiments shows the importance of the choice of $N_{\min}$. Due to the measurement noise feedback and disturbances, TOMPC behaves too aggressively if $N_{\min}$ is chosen too small because the too short horizon will enforce deadbeat reactions whereas a longer horizon enables a relaxed response. Fig. 8 illustrates this by showing the input applied to the actuators of the system for two different choices of $N_{\min}$, namely $N_{\min} = 8$ (black line) and $N_{\min} = 6$ (gray line). When a step is requested to the system (time 0 s), both controllers request the same system input and hence yield the same rise and settling time. However, close to the setpoint (from time 0.2 s) the controller with $N_{\min} = 8$ is much less sensitive. This effect of measurement noise and disturbances can also be reduced by decreasing the bandwidth of the state estimator, however this only reduces the effect and can not eliminate it.

A third experiment shows that although this controller is designed for completing steps with minimal settling time, it can accept a new reference step before the previous one is completed. Fig. 9a shows this for two steps of respectively 5 cm and 7 cm. Fig. 9b shows the corresponding system input. By comparing this figure with Fig. 8, it can be seen that when the second reference step is requested, the input reaches its maximal value again in order to fully accelerate instead of decelerate.

4.2. Overhead crane

4.2.1. Test setup

The second test setup is the overhead crane with fixed cable length shown in Fig. 10a. Fig. 10b shows a schematic representation of this system. The actuator of this system is a velocity con-

Table 2
The 1% settling time for different steps using TOMPC and MPC.

Step (cm)	TOMPC (s)	MPC (s)
1	0.12	0.16
5	0.16	0.22
10	0.18	0.25
15	0.22	0.30

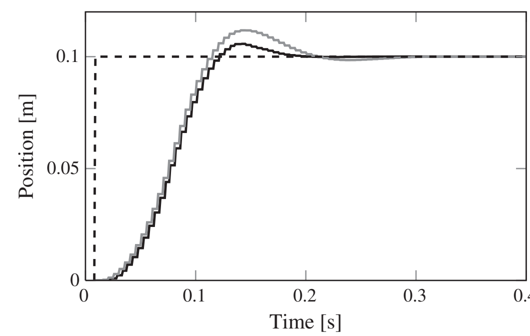

Fig. 5. Linear drive system: Drive system motion obtained with TOMPC (black solid line) and MPC (gray solid line) for a reference step of 10 cm (dashed line).

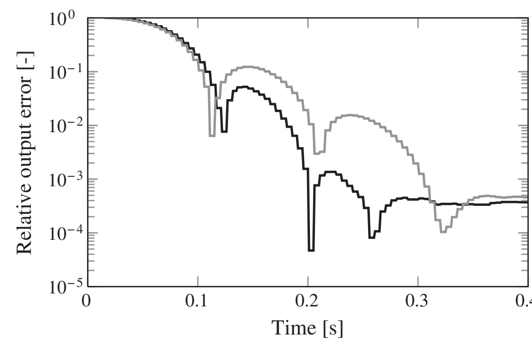

Fig. 6. Linear drive system: Relative output error obtained by control with TOMPC (black line) and MPC (gray line) for a reference step of 10 cm.

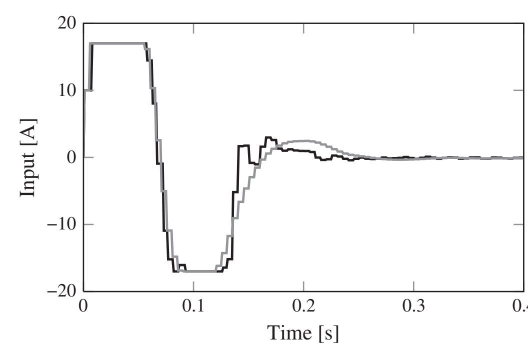

Fig. 7. Linear drive system: Control input signal obtained by control with TOMPC (black line) and MPC (gray line) for a reference step of 10 cm.

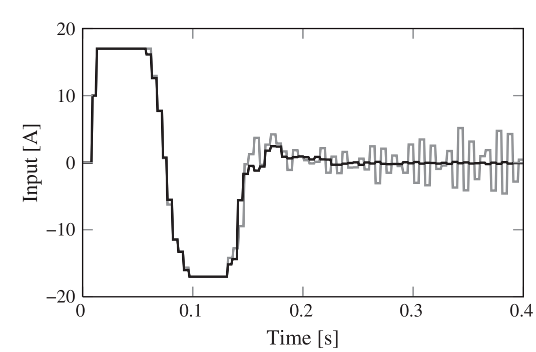

Fig. 8. Linear drive system: Control input applied to the system with a TOMPC controller with $N_{\min} = 8$ (black line) and $N_{\min} = 6$ (gray line).

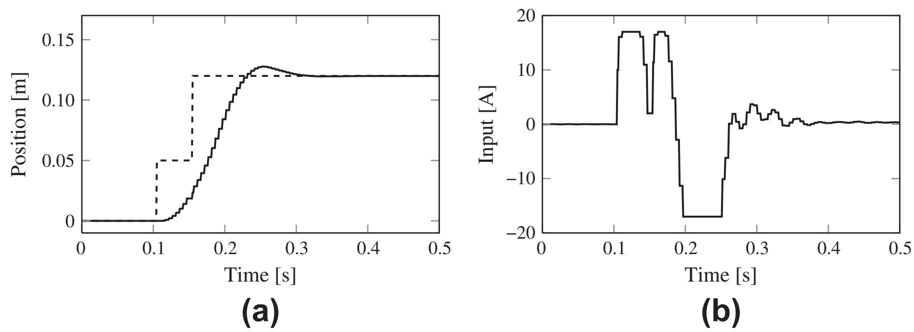

Fig. 9. Linear drive system: Drive system motion (full line) Fig. 9a and control input Fig. 9b obtained when a reference trajectory of two steps is requested (dashed line). The second step is requested before the first one is completed.

trolled DC-motor that drives a trolley through a rack and pinion. The position of the trolley x is measured using an angular encoder mounted on the DC-motor axle, yielding a position measurement resolution of 3 μm. The swing angle θ is measured using a rotative encoder mounted on the axle to which the cable is attached, yielding an angular resolution of 0.0009°. The input to this system is a voltage u, which is a reference applied to the 25 Hz bandwidth internal velocity loop. The input is limited to ±1 V and the input slew rate is limited to ±6 V/s due to limitations of the motor current amplifier. The maximal range of the trolley is 70 cm. The length of the cable is fixed to 450 mm. The system controllers are embedded on a dSPACE board DS1103 which contains a 1 GHz processor with 90 MB RAM, and they are implemented through C++ functions in the real-time-target environment of Simulink. The system controllers are applied at a sampling frequency of 60 Hz.

The dynamics of this system can be modeled by a fourth order discrete time model, of which the parameters are obtained using the same nonlinear least square frequency domain identification approach of Section 4.1.1 based on FRF measurements that are obtained from multisine excitations with a frequency content between 0.05 Hz and 5 Hz [39]. The identified model has three outputs; the position of the trolley, the swing angle and the position of the lower mass. The estimated resonance frequency is 0.74 Hz which corresponds to the theoretical value $\frac{1}{2\pi}\sqrt{\frac{g}{L}}$ with $L = 450$ mm. The damping of the estimated resonance frequency is $\zeta = 0.00168$ which is extremely low.

4.2.2. TOMPC controller

For this system, a TOMPC controller is developed taking into account the above mentioned constraints on input and input slew rate. As the states of the system model are not directly measurable, first a Luenberger observer [40] is designed. For the estimation of the states, the measurements of both the position of the trolley x and the swing angle θ are used. The best results are obtained using a state observer with a bandwidth of 0.95 Hz. Higher observer bandwidths yield a too nervous or even unstable behavior. It should be noted that this bandwidth corresponds to the maximal obtainable bandwidth with unconstrained linear controllers. In order to apply transformation (8), the position of the lower mass $y = x - L\theta\frac{\pi}{180}$ is chosen as the output which yields an observable state space model. As the solution of optimization problem (3d) requires a large computation time, the control input is only available near the end of every sample period, and this introduces an additional delay. To account for this delay, the state space model is extended with one delay state. Based on this fifth order model, the TOMPC is implemented. Figs. 11 and 12 show the response of the system with TOMPC on a reference step of 10 cm. Fig. 13 shows the corresponding input to the system. The constraints on the input amplitude are not reached. The input slew rate however, shown in Fig. 14, reaches the slew rate constraints during almost the whole duration of the motion, showing that the system is working at its limits which indicates that the controller steers the system as fast as possible to the desired endpoint.

Figs. 15–17 show respectively the position of the trolley x, the swing angle θ and the input u for several reference steps ranging from 20 cm to 50 cm. Fig. 17 and more clearly Fig. 18 show that the controller is continuously hitting the input constraints during the motion, showing its time optimality for all reference steps. A similar time optimal behavior can also be attained using a linear controller in combination with an optimized reference trajectory (e.g. [5,6]). However with this linear controller approach, the time optimal reference trajectories have to be computed off-line, e.g. in between two point-to-point motions and can therefore not be applied for new previously unknown reference step requests without introducing a computational delay and/or suboptimal behavior.

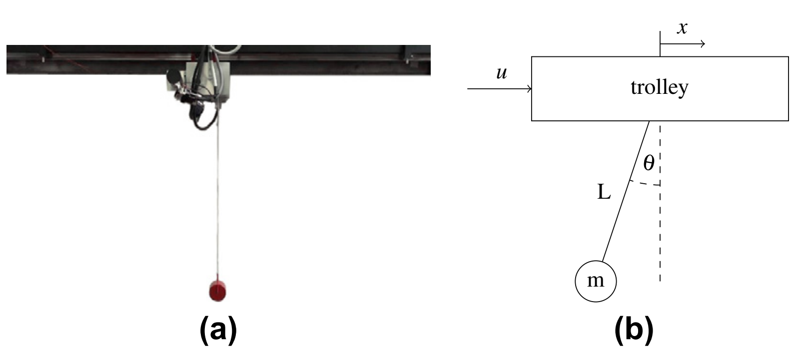

Fig. 10. Picture Fig. 10a and schematic drawing Fig. 10b of the overhead crane in the PMA-lab at K.U. Leuven.

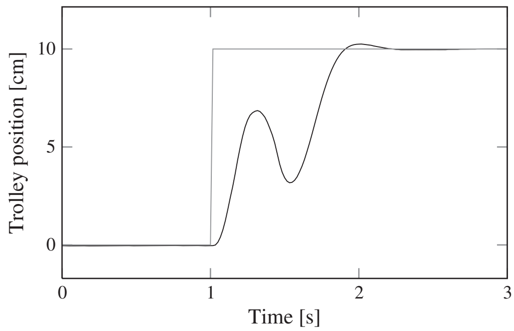

Fig. 11. Overhead crane: Position of the trolley (black) controlled by a TOMPC controller for a reference step of 10 cm (gray).

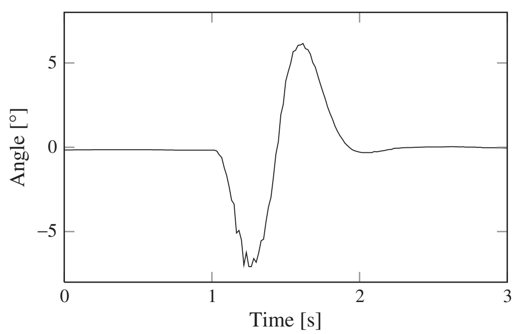

Fig. 12. Overhead crane: Swing angle of the TOMPC controlled system for a reference step of 10 cm.

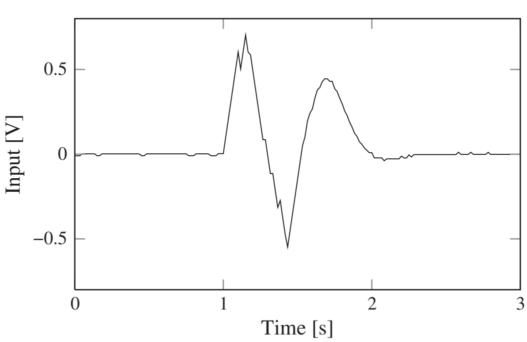

Fig. 13. Overhead crane: Input applied to the overhead crane for a requested step of 10 cm controlled by a TOMPC controller.

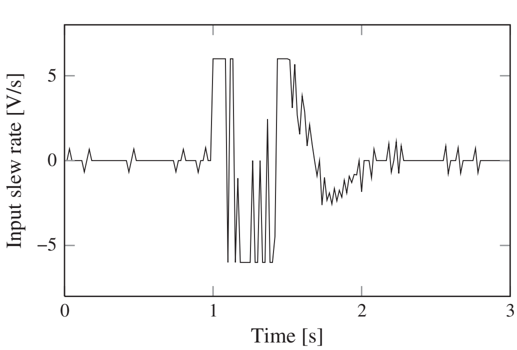

Fig. 14. Overhead crane: Slew rate of the input applied to the overhead crane for a requested step of 10 cm controlled by a TOMPC controller.

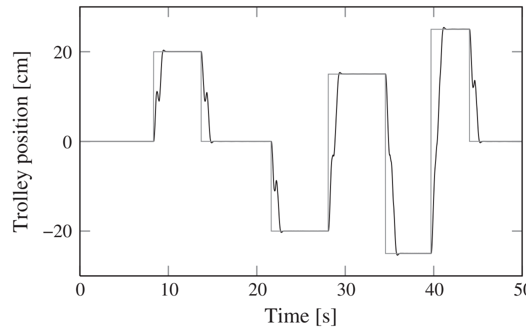

Fig. 15. Overhead crane: Position of the trolley (black) controlled by a TOMPC controller for a series of reference steps (gray).

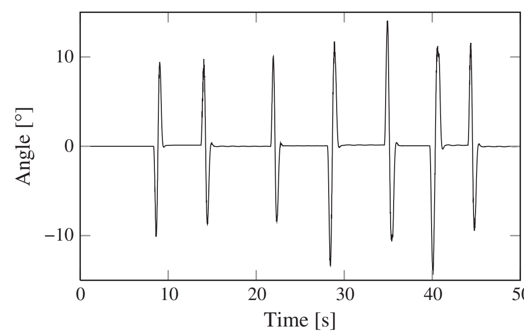

Fig. 16. Overhead crane: Swing angle of the TOMPC controlled system for a series of reference steps.

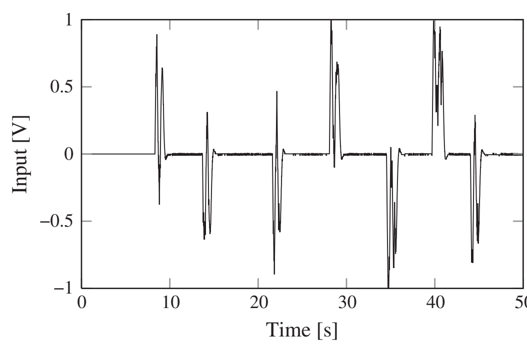

Fig. 17. Overhead crane: Input applied to the overhead crane controlled by a TOMPC controller for a series of reference steps.

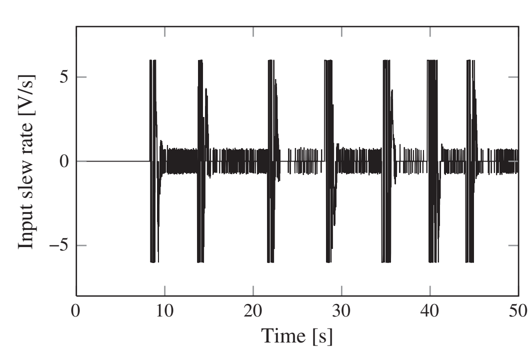

Fig. 18. Overhead crane: Slew rate of the input applied to the overhead crane controlled by a TOMPC controller for a series of reference steps.

5. Conclusion

This paper presents and experimentally validates a new type of MPC approach, called 'time optimal MPC' (TOMPC) aiming at performing time optimal point-to-point motion. Time-optimal behavior is achieved via a two level optimization problem which formulates settling time optimization as a feasibility problem based on a traditional MPC problem with quadratic objective function and system constraints. Through a careful formulation of the optimization problem and implementation of the active set numerical solution method, the worst case solution time is decreased by a factor of 8. As a result, sampling rates up to 200 Hz are achievable for systems with orders up to 5 and prediction horizons up to 45 time steps, yielding that TOMPC becomes feasible for simple mechatronic applications.

Acknowledgement

Lieboud Van den Broeck is funded by a Ph.D. fellowship of the Research Foundation - Flanders (FWO - Vlaanderen). This work benefits from K.U.Leuven-BOF PFV/10/002 Center-of-Excellence Optimization in Engineering (OPTEC), the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO), research project FP7-HD-MPC, research project EMBOCOM FP7-ICT-2009-4 248940, research project IWT-SBO 80032 (LeCoPro) of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen), research projects G0.0320.08, G.0558.08, G.0422.08, and G.0377.09 of the Research Foundation - Flanders (FWO - Vlaanderen), and K.U.Leuven's Concerted Research Action GOA/10/11. The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007–2013 under grant agreement number FP7-ICT-2009-4 248940.

References

[1] Franklin GF, Powell DJ, Emami-Naeini A. Feedback control of dynamic systems. Prentice Hall PTR; 2001.
[2] Zhou T, Doyle J, Glover K. Robust and optimal control. Prentice Hall; 1995.
[3] Morari M, Zafiriou E. Robust process control. Englewood Cliffs: Prentice Hall; 1989.
[4] Verscheure D, Demeulenaere B, Swevers J, Schutter JD, Diehl M. Time-optimal path tracking for robots: a convex optimization approach. IEEE Trans Autom Control 2009;54:2318–27.
[5] Demeulenaere B, Caigny JD, Pipeleers G, Schutter JD, Swevers J. Optimal splines for rigid motion systems: benchmarking and extensions. Trans ASME J Mech Des 131.
[6] Henrion D, Lasserre J. LMIs for constrained polynomial interpolation with application in trajectory planning. Syst Control Lett 2006;55(6):473–7.
[7] Boyd S, Vandeberghe L. Convex optimization. Cambridge University Press; 2004.
[8] Singer NC, Seering WP. Preshaping command inputs to reduce system vibration. Trans ASME J Dyn Syst Meas Control 1990;112:76–82.
[9] Singhose W, Seering WP, Singer N. Residual vibration reduction using vector diagrams to generate shaped inputs. J Dyn Syst Meas Control – Trans ASME 1994:654–9.
[10] Pao L, Lau M. Robust input shaper control design for parameter variations in flexible structures. Trans ASME J Dyn Syst Meas Control 2000;122:63–70.
[11] Robertson MJ, Erwin RS. Command shapers for systems with actuator saturation. In: Proceedings of the American control conference, New York City; 2007. p. 760–5.
[12] Van den Broeck L, De Caigny J, Pipeleers G, et al. A linear programming approach to robust input shaping. In: Proceedings of the AMC, Trento; 2008. p. 80–5.
[13] Singhose W, Pao L. A comparison of input shaping and time-optimal flexible-body control. Control Eng Pract 1997;5(4):459–67.
[14] Lambrechts P, Boerlage M, Steinbuch M. Trajectory planning and feedforward design for electromechanical motion systems. Control Eng Pract 2005;13(2):145–57.
[15] Zanasi R, Lo Bianco C, Tonielli A. Nonlinear filters for the generation of smooth trajectories. Automatica 2000;36(3):439–48.
[16] Maciejowski J. Predictive control with constraints. Prentice Hall; 2000.
[17] Garćia C, Prett D, Morari M. Model predictive control: theory and practice – a survey. Automatica 1989;25:335ff.
[18] Qin S, Badgwell T. A survey of industrial model predictive control technology. Control Eng Pract 2003;11:733–64.
[19] Bemporad A, Morari M, Dua V, Pistikopoulos E. The explicit linear quadratic regulator for constrained systems. Automatica 2002;38:3–20.
[20] Ferreau H, Bock H, Diehl M. An online active set strategy to overcome the limitations of explicit mpc. Int J Robust Nonlinear Control 2008;18(8):816–30.
[21] Wills A, Bates D, Fleming A, Ninness B, Moheimani S. Application of mpc to an active structure using sampling rates up to 25 kHz. In: Proceedings of the conference on decision and control and European control conference ECC'05, Seville; 2005.
[22] Ferreau H, Ortner P, Langthaler P, del Re L, Diehl M. Predictive control of a real-world diesel engine using an extended online active set strategy. Ann Rev Control 2007;31(2):293–301.
[23] Zhao J, Diehl M, Longman R, Bock H, Schlöder J. Nonlinear model predictive control of robots using real-time optimization. In: Proceedings of the AIAA/AAS astrodynamics conference, Providence, RI; 2004.
[24] Grieder P, Morari M. Complexity reduction of receding horizon control. In: Decision and control, 2003. Proceedings. 42nd IEEE conference on, vol. 3; 2003. p. 3179–90.
[25] Van den Broeck L, Diehl M, Swevers J. Embedded optimization for input shaping. IEEE Trans Control Syst Technol 2010;18(5):1146–54.
[26] Keerthi S, Gilbert E. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. J Optim Theory Appl 1988;57(2):265–93.
[27] Pannocchia G, Rawlings J. Disturbance models for offset-free model-predictive control. AIChE J 2003;49:426–37.
[28] Magni L, De Nicolao G, Magnani L, Scattolini R. A stabilizing model-based predictive control for nonlinear systems. Automatica 2001;37(9):1351–62.
[29] Mayne D, Rawlings J, Rao C, Schockaert P. Constrained model predictive control: stability and optimality. Automatica 2000;36:789–814.
[30] Marruedo D, Alamo T, Camacho E. Stability analysis of systems with bounded additive uncertainties based on invariant sets: stability and feasibility of MPC. In: Proceedings of The 2002 American control conference; 2002. p. 364–9.
[31] Morari M, Lee JH. Model predictive control: past, present and future. Comput Chem Eng 1999;23(4–5):667–82.
[32] Bellman R. Dynamic programming. Princeton Univ. Press; 1957.
[33] Kailath T. Linear systems. Prentice Hall; 1980.
[34] Nocedal J, Wright S. Numerical optimization. Springer series in operations research and financial engineering. Springer; 2006.
[35] Bartlett R, Biegler L. QPSchur: a dual, active set, Schur complement method for large-scale and structured convex quadratic programming algorithm. Optim Eng 2006;7:5–32.
[36] Gill P, Murray W, Saunders M, Wright M. Procedures for optimization problems with a mixture of bounds and general linear constraints. ACM Trans Math Softw 1984;10(3):282–98.
[37] Ferreau H. qpOASES User's Manual. <http://www.qpOASES.org/ (2007–2009). http://homes.esat.kuleuven.be/optec/software/qpOASES/download/manual-1.1.pdf>.
[38] ApS M. The mosek optimisation tools manual.
[39] Pintelon R, Schoukens J. System identification: a frequency domain approach. IEEE Press; 2001.
[40] Luenberg D. Introduction to observers. IEEE Trans Autom Control 1971;AC16(6):596–602.