Model Predictive Trajectory Set Control with Adaptive Input Domain Discretization

Artemi Makarow, Martin Keller, Christoph Rösmann and Torsten Bertram

The 2018 American Control Conference, Milwaukee, WI, USA

"© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works."

Model Predictive Trajectory Set Control with Adaptive Input Domain Discretization

Artemi Makarow, Martin Keller, Christoph Rösmann and Torsten Bertram

***Abstract*— From a theoretical point of view, model predictive control (MPC) promises a high control quality since the future system performance is predicted and evaluated in every sampling interval. Additionally, desired optimization objectives can be realized while explicitly adhering to state and control input constraints. This control concept has the potential to set a new standard in industrial applications. In practical terms, functioning and properties of online optimization algorithms must be familiar to the developer to adjust the control performance of a mechatronic system. Moreover, it is challenging to realize a computationally expensive control concept for mechatronic systems with fast dynamics due to the limited computing power. It is worth mentioning that industrial systems usually exhibit hardware with low computing power. This contribution presents the model predictive trajectory set control (MPTSC) that constitutes a sub-optimal MPC with a sparse discretization of the control input domain. The optimal control input is determined without the use of iterative optimization techniques. To mimic quasi-continuous control input values similar to MPC, an adaptive input domain discretization is developed. MPTSC includes most advantages of MPC and is still computationally efficient. The implementation is less complex and error-prone and thus addresses especially industrial applications. The performance and the computation time are evaluated in comparison to MPC with nonlinear benchmark systems. Furthermore, the approach is tested experimentally on an industrial plant emulator with a sample rate of 100 Hz.**

I. INTRODUCTION

The control task of mechatronic systems in industrial applications is usually performed with cascaded control architectures based on PID controllers. They are widely applied due to their well-studied theory and moderate computational effort [1]. To enhance the closed-loop performance and to meet all the specified requirements for strongly nonlinear systems, classic PID controllers are extended with nonlinear characteristic curves for the integral and proportional amplification. Thus, the controller adapts the parameters as a function of the control error $e(t)$. With an increasing number of parameters, higher demands on the control performance can be satisfied. However, since the complexity increases, the parameterization becomes a challenging process. Parameterization is either carried out by a process expert with a high time expenditure or an automated Hardware-in-the-Loop meta optimization by applying multi-criteria evolutionary algorithms [2]. However, the control performance of classic control concepts is limited since no respectively vague information about the possible future system evolution is considered during runtime. Moreover, state and control input constraints are only implicitly taken into account. The ability to operate close to the system's physical constraints is limited.

Model predictive control (MPC) predicts the future system evolution and minimizes a user-defined but usually smooth objective function during runtime. It repeatedly solves an optimal control problem over a moving finite horizon in each sampling interval [3]. Consequently, a high control performance can be achieved. The prediction process uses a dynamic plant model. It is to be emphasized that MPC explicitly adheres to state and control input constraints. However, solving an optimal control problem in every sampling interval requires a high computing power which is usually not available in current industrial applications. Thus, researchers focus on developing numerical efficient realizations and approximations of MPC. In [4] a multiple-shooting method is presented which exhibits faster convergence properties compared to single-shooting. The real-time iterations scheme further reduces computation time by only performing a warm-started single sequential quadratic programming step in each sampling interval [5]. A combination with first-order methods for embedded MPC applications is presented in [6]. Further methods efficiently deal with sparse interior-point-methods [7], [8] or projected gradients [9]. Time-optimal MPC is realized in [10], [11].

To further reduce the computational burden of MPC, sub-optimal solutions are obtained with move-blocking strategies in which the number of control interventions along the horizon is limited [12]. Concerning the available degrees of freedom, this strategy reduces the dimension of the optimization problem and hence the computation time but also leads to slower control performance. In extreme cases, a single degree of freedom and thus a constant control input over the entire prediction horizon is considered for optimization.

In the context of power electronics, MPC has also been introduced successfully although high sampling rates are required [13], [14]. The finite control set MPC (FCS-MPC) makes use of the limited number of possible switching states and thus operates quickly. The optimal switching states of a converter are the optimization variables which can be set as active directly after solving the optimal control problem. A modulator which converts a continuous signal into corresponding switching states is not required. An extreme move-blocking strategy is not applied since the switching states are responsible for a rotating magnetic field.

Model predictive trajectory set control (MPTSC) reduces the complexity of a controller significantly and addresses mechatronic systems with fast dynamics in particular.

The authors are with the Institute of Control Theory and Systems Engineering, TU Dortmund University, D-44227, Germany, artemi.makarow@tu-dortmund.de

MPTSC has its origin in the development of an emergency steering assist. The model predictive trajectory set approach (MPTSA), developed for emergency maneuvers of vehicles in critical traffic situations, combines the planning and control of a collision avoidance trajectory in a single step [15]. In contrast to MPTSA, MPTSC is meant to be a tracking controller. MPTSC mimics the move-blocking strategy and makes use of the advantages of FCS-MPC. In [16], MPTSC has proved its applicability for a fast-acting hydraulic valve. The basic idea of MPTSC is to discretize the control input domain sparsely. Every available input candidate is kept constant over the entire prediction horizon. Thus a discrete set of trajectory candidates is predicted in every sampling interval. The control input of the trajectory with the smallest objective function value is selected and applied for control. Note, this concept exhibits the advantageous characteristics of predicting the plant's future behavior during runtime similar to conventional MPC. However, a gradient-based algorithm for solving the optimal control problem is not required. MPTSC is not meant to replace MPC in general, in fact it is well known from the theory of MPC that short horizons respectively less control interventions cannot stabilize every system with arbitrary constraints [17]. However, for small- to mid-scale system models with only a few control input variables and bounds on states and controls, MPTSC outperforms MPC regarding computational efficiency and simplicity. This approach is suitable especially in the case of stable open-loop dynamics. These particular systems mentioned before typically arise in mechatronic applications. Electromagnetic actuator dynamics and mechanical motions are usually defined in terms of a second or third order system of differential equations with a nonlinear character. Consequently, MPTSC intends to address a wide range of these mechatronic systems. MPTSC is suitable to replace complex native controllers in cascaded control structures. In this case, the number of input candidates to be tested is manageable without the use of further search algorithms like in [18]. The execution time is deterministic since the number of required computations is known in advance.

To achieve a nearly optimal closed-loop performance with the (sub-optimal) MPTSC, the basic concept needs to be extended with an adaptive discretization strategy. The contribution of this paper is to demonstrate that MPTSC with an adaptive input domain discretization requires just a couple of trajectory candidates to mimic the optimal control performance of the MPC with a control horizon $n_\mathrm{c} = 1$. MPTSC with an adaptive discretization does not result in a noticeably higher computation time compared to a linear discretization since only a functional mapping is performed.

The next section describes MPTSC and introduces the adaptive input domain discretization in detail. Section III provides a performance comparison with MPC (with $n_\mathrm{c} = 1$) for two nonlinear benchmark systems which incorporate a wide range of mechatronic characteristics. For experimental validation the industrial plant emulator with a sampling rate of $f_\mathrm{s} = 100\,\mathrm{Hz}$ is utilized in section IV. Finally, section V summarizes the results and provides an outlook.

II. MODEL PREDICTIVE TRAJECTORY SET CONTROL

Fundamental Formulation

In this contribution, a single-input system is investigated. The discrete state equations with p states and a single input are obtained by finite-differences and sample time ΔT:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{f}(\mathbf{x}_k, u_k)\,\Delta T, \quad \mathbf{x}_{k=0} = \mathbf{x}_0. \tag{1}$$

Thereby, $\mathbf{x}_k \in \mathbb{R}^p$ denotes the state vector at time instance k and $u_k \in \mathbb{R}$ the corresponding control input. Control inputs are further limited by the following box constraint:

$$u_\mathrm{min} \leq u_k \leq u_\mathrm{max}. \tag{2}$$

For the optimal control problem, the control input domain $\mathbb{R}$ is discretized such that $\mathbb{A}_k \subset \mathbb{R}$ and $u \in \mathbb{A}_k$. The (sub-)optimal control input u_k^* is obtained as follows:

$$u_k^* = \arg\min_{u \in \mathbb{A}_k} J(\mathbf{x}_k, u) \tag{3}$$

subject to (1), (2) and the state box constraints

$$\mathbf{x}_\mathrm{min} \leq \mathbf{x}_k \leq \mathbf{x}_\mathrm{max}. \tag{4}$$

At every time step k, the implicit control law is defined as $u_k = u_k^*$. The control input is kept constant over the prediction horizon t_p with $n_\mathrm{c} = 1$. To solve the optimal control problem in (3) a set of trajectory candidates is predicted by applying the possible control inputs $u \in \mathbb{A}_k$. The trajectory candidates are evaluated with the objective function value $J(\cdot)$. The candidate that exhibits the least objective function value is selected, and the corresponding input u_k^* is applied to the plant. It is important to emphasize that the requirements on $J(\cdot)$ are fairly mild when utilizing the MPTSC since no gradient-based optimization is required. Non-smooth cost functions might be utilized depending on the actual application.

Adaptive Input Domain Discretization

In the case of a sparse equidistant (or linear) discretization of the input domain the available input candidates are defined as follows:

$$u_\mathrm{lin} \in \mathbb{D} := \{u_\mathrm{min}, u_\mathrm{min} + \Delta u, u_\mathrm{min} + 2\,\Delta u, ..., u_\mathrm{max}\} \tag{5}$$

Since the number of inputs is finite, the MPTSC leads to a undesired performance. If the discretization step size Δu is not small enough, the controller has to switch the input frequently. According to the system's steady-state, the equidistant discretization only addresses systems without self-compensation since Δu can be adjusted such that the zero input is in the linear set $\mathbb{D}$. To smooth the control interventions and to reach every possible steady-state of systems with self-compensation, an adaptive input domain discretization is developed. The idea is to find a function which maps the linear discretized inputs to a nonlinear representation in order to achieve a fine discretization in the proximity of the previous input u_{k-1}. Consequently, the mapping is performed adaptively in order to seek for a quasi-continuous input range. In this contribution, polynomial functions are designed according to various conditions. To ensure that the box constraint in (2) is not violated a single curve is described by two m-order polynomials with $f(u_{\mathrm{lin}}) : \mathbb{D} \rightarrow \mathbb{A}_k$:

$$f(u_{\mathrm{lin}}) = u = \begin{cases} p_1\, u_{\mathrm{lin}}^m + p_2\, u_{\mathrm{lin}} + p_3, & \text{if } u_{\mathrm{lin}} \geq 0 \\ p_4\, u_{\mathrm{lin}}^m + p_2\, u_{\mathrm{lin}} + p_3, & \text{if } u_{\mathrm{lin}} < 0. \end{cases} \tag{6}$$

The set $\mathbb{A}_k$ changes in every sample step k with $u \in \mathbb{A}_k$. The polynomials are parameterized according to the conditions:

$$\begin{gathered} f(u_{\max}) = u_{\max},\, f(u_{\min}) = u_{\min}, \\ f(0) = u_{k-1},\, f'(0) = \alpha. \end{gathered} \tag{7}$$

The variable α describes the desired slope in $\big(0, u_{k-1}\big)$. In this contribution, α is set to zero. However, it is conceivable to adapt α depending on the control error $\mathbf{e}(t)$. The polynomial order is set to $m = 3$. Figure 1 shows the adaptive mapping for three different previous values u_{k-1}. The linear gray curve represents the parameter set $\alpha = 1$ and $f(0) = 0$ and would deactivate the adaptive discretization.

III. Simulative Analysis of the MPTSC

The simulative analysis is performed with a zero model mismatch. Thus, the plant and prediction model are identical and are solved numerically using the explicit Euler method with a dedicated step size $\Delta T = 0.01\,\mathrm{s}$. For small time steps ΔT the following numerical approximation of the chosen objective function is adequate:

$$\begin{aligned} J(\mathbf{x}, u) &= \int_{t=0}^{t=t_{\mathrm{p}}} \big((\hat{\mathbf{x}} - \mathbf{x}_{\mathrm{f}})^{\intercal} \mathbf{Q}\, (\hat{\mathbf{x}} - \mathbf{x}_{\mathrm{f}}) \big)\, \mathrm{d}t \\ J(\mathbf{x}_k, u) &\approx \Delta T \sum_{l=1}^{l=n_{\mathrm{p}}} \big((\hat{\mathbf{x}}_{k,l} - \mathbf{x}_{\mathrm{f},k})^{\intercal} \mathbf{Q}\, (\hat{\mathbf{x}}_{k,l} - \mathbf{x}_{\mathrm{f},k}) \big) \end{aligned} \tag{8}$$

with $\hat{\mathbf{x}}_{k,0} = \mathbf{x}_k$ and reference target state $\mathbf{x}_{\mathrm{f}}$. The index k denotes the runtime sampling and the index l the time steps on the prediction horizon. The matrix $\mathbf{Q}$ denotes at least a quadratic positive semi-definite weight matrix for the penalization of the state error $\hat{\mathbf{x}} - \mathbf{x}_{\mathrm{f}}$. The global control task is the transition between the initial states $\mathbf{x}_0$ and the final states $\mathbf{x}_{\mathrm{f}}$. The first set-point state x_{ref} is defined as a function of time to provide a step sequence. Note, during prediction the sampled reference states $\mathbf{x}_{\mathrm{f},k}$ are treated as constant over the prediction horizon since the future course of the reference is not known in advance in this contribution. To provide an absolute closed-loop reference performance, a classic MPC with an extreme move-blocking strategy, $n_{\mathrm{c}} = 1$, is utilized. In this case the optimal control input u_k^* is obtained as:

$$u_k^* = \arg\min_{u \in \mathbb{R}} J(\mathbf{x}_k, u) \tag{9}$$

subject to (1), (2) and (4). The optimal control problem is solved by the iterative algorithm IPOPT [19]. IPOPT applies the interior point method for sparse nonlinear programs and provides warm starts. IPOPT is configured to reach convergence in every sampling interval. The remaining settings are the same for both control concepts. HSL-MA57 is utilized as internal linear solver [20]. For a comparative analysis of the MPC and MPTSC regarding the control performance the following merit function is applied:

$$NRMSE = 100\,\% \left(\frac{\|\mathbf{y}_{\mathrm{MPC}} - \mathbf{y}_{\mathrm{MPTSC}}\|_2}{\big(y_{\mathrm{MPC,max}} - y_{\mathrm{MPC,min}}\big)\sqrt{n}} \right). \tag{10}$$

The vector $\mathbf{y}$ denotes a chosen state signal for comparison with n elements. In this contribution, the performance is evaluated concerning conventional design criteria for mechatronic systems like rise time, settling time and overshoot characteristic. The weights in $\mathbf{Q}$ and the prediction horizon t_{p} are tuned manually. Since the control horizon is set to $n_{\mathrm{c}} = 1$, the prediction horizon t_{p} becomes a controller parameter which has an impact on the dynamics of the closed-loop performance. Hence, a terminal cost as typically applied in literature [17], $(\hat{\mathbf{x}}_{k,n_{\mathrm{p}}} - \mathbf{x}_{\mathrm{f},k})^{\intercal}\mathbf{S}(\hat{\mathbf{x}}_{k,n_{\mathrm{p}}} - \mathbf{x}_{\mathrm{f},k})$, to approximate the cost for an infinite horizon and ensure stability according to the Riccati equation is omitted in (8). For the comparison of computation times, a further hard constraint solver is used. The sequential programming approach SQP is realized using *qpOases* [21] and backtracking linesearch according to a ℓ_1 merit function [22]. Both IPOPT and SQP are limited to a single optimization iteration in every sampling interval to reach short computation times. Computations are performed in C++ (PC: 3,4 GHz Intel i7-6700 CPU, Ubuntu). The open-loop results are evaluated by ten initial ($k = 0$) optimization problems. The closed-loop results are evaluated by three entire simulations.

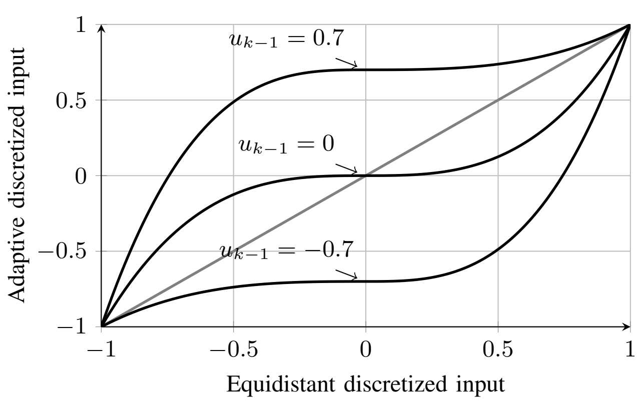

Fig. 1: Adaptive discretization of the control input domain applying polynomial functions with $u_{\min} = -1$, $u_{\max} = 1$.

Van-der-Pol Oscillator

The fast and precise control of the Van-der-Pol Oscillator is a challenging task. This system provides a common benchmark in the MPC literature. It is an oscillatory dynamic system with nonlinear damping and self-compensation. The second order differential equation is defined by:

$$\ddot{x} + \big(x^2 - 1\big)\,\dot{x} + x = u. \tag{11}$$

Transforming this equation into a state space representation with state vector $\mathbf{x} = [x, \dot{x}]^{\intercal} = [x_1, x_2]^{\intercal}$ results in a second order continuous-time system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \big[\dot{x}, \quad -\big(x^2 - 1\big)\,\dot{x} - x + u\big]^{\intercal}. \tag{12}$$

The input variable is limited to $|u| \leq 1$ and the box constraint $|x_2| \leq 0.3$ is set as active. The weight parameters of the Matrix $\mathbf{Q}$ are set to $q_{11} = 1$, $q_{22} = 0.2$, $q_{12} = q_{21} = 0$ and $t_\mathrm{p} = 0.5\,\mathrm{s}$. The reference vector is $\mathbf{x}_\mathrm{f} = [x_\mathrm{ref}, 0]^\intercal$. For a more detailed illustration of the functionality respectively the open-loop performance of MPTSC the set of predicted trajectories of the first state x_1 is depicted for multiple control sampling intervals in Figure 2. Figure 3 shows the simulated closed-loop performance for this nonlinear system. Both the MPC and the MPTSC illustrate the ability to adhere to constraints. Moreover, this evaluation reports the potential for improvement of the sub-optimal characteristic of the basic MPTSC when utilizing the adaptive input domain discretization. The subset $\mathbb{A}_k$ in every time step k consists of 11 possible input candidates regardless of the discretization method. The lack of smoothness of the control input u_k of the MPTSC with linear input domain discretization leads to a rather damped performance during the transition phase and an oscillation around the steady-states. The performance of the MPTSC with an adaptive input domain discretization results only in a slight deviation from the MPC. In Figure 4 the deviation of the MPTSC performance from the optimal MPC performance over the number of input candidates is presented. When utilizing the adaptive input domain discretization, only a small number of input candidates is needed to reach the performance saturation and to approximate the optimal MPC performance sufficiently well. Nonetheless, a minimum number of five input candidates is necessary. This fact addresses real-time applications regarding mechatronic systems with fast dynamics. The number of input candidates in set $\mathbb{A}_k$ must be chosen according to the desired control quality and available computational resources. However, recursive feasibility and stability must be ensured. Theoretical stability analysis is beyond the scope of this paper but generally, the stability results of MPC with $n_\mathrm{c} = 1$ apply if the control discretization is sufficiently small [17]. The analysis of the computational effort can be extracted from Table I. The renouncement of optimization algorithms leads to very short computation times.

Industrial Plant Emulator ECP 220

This mechatronic system is designed to emulate automated industrial processes and is shown in Figure 5. It allows an advanced development of control and model identification

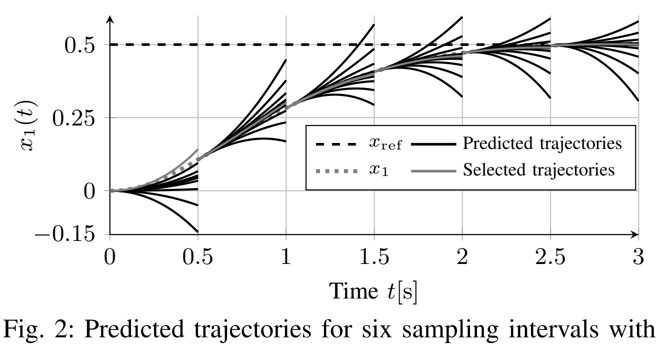

Fig. 2: Predicted trajectories for six sampling intervals with a prediction horizon of $t_\mathrm{p} = 0.5\,\mathrm{s}$.

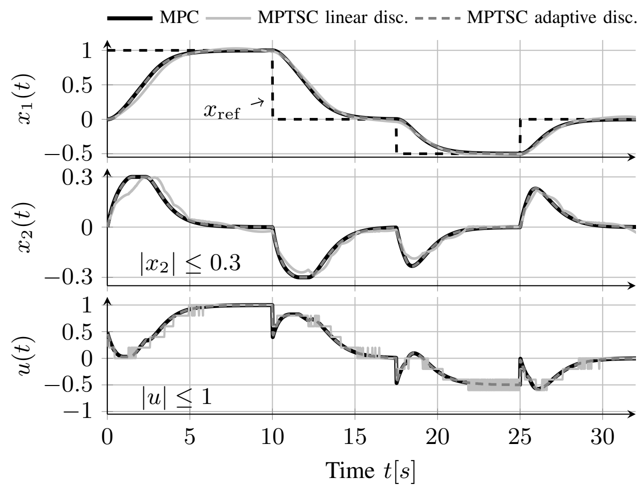

Fig. 3: Simulated closed-loop performance of the MPTSC for the Van-der-Pol Oscillator in comparison to MPC.

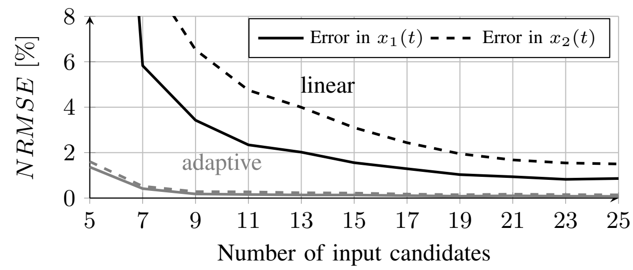

Fig. 4: Deviation of the sub-optimal MPTSC from the optimal MPC for the Van-der-Pol Oscillator.

concepts for mechatronic systems with comparable characteristics. It is possible to modify the mechanical configuration and therefore to manipulate the system performance. The system consists of two load plates actuated by brushless DC drives which are coupled via elastic transmission belts. The drive angles can be measured with two incremental encoders. An internal digital signal processor (DSP) provides the angular velocity. Thus the system states can be determined without the use of a model-based state observer. In this contribution, the first motor provides the driving torque while the second motor is not operated actively. To reach a high level of nonlinear characteristic the more distant second encoder is used as the measurement unit. The modeling process is related to the axis where the second encoder is located. The ECP model describes the system behavior between the reference current (input) u and the position x_1. The present plant is highlighted in gray in Figure 5. This model represents a system without self-compensation. The second order differential equation is defined by:

$$J_2\,\ddot{x} + \tau_\mathrm{c}\,\tanh(\beta\,\dot{x}) + d\,\dot{x} = 4\,k_1\,u. \tag{13}$$

The variable $J_2 = 3.39 \cdot 10^{-2}\,\mathrm{Nm}\cdot\mathrm{s}^2\cdot\mathrm{rad}^{-1}$ is the overall load inertia in the second axis, $k_1 = 9.5 \cdot 10^{-2}\,\mathrm{Nm}\cdot\mathrm{A}^{-1}$ is the motor constant, $\tau_\mathrm{c} = 9.39 \cdot 10^{-2}\,\mathrm{Nm}$ is the static friction torque and $d = 1.93 \cdot 10^{-2}\,\mathrm{Nm}\cdot\mathrm{s}\cdot\mathrm{rad}^{-1}$ is a damping constant. The discontinuous static friction representation is approximated with a trigonometric function. The variable

TABLE I: Van-der-Pol Oscillator computation times

Approach	First open-loop	Closed-loop
MPC with IPOPT	$(4133 \pm 900)\,\mu s$	$(855 \pm 72)\,\mu s$
MPC with SQP	$(1324 \pm 107)\,\mu s$	$(167 \pm 43)\,\mu s$
Adaptive MPTSC with 25 cand.	$(54 \pm 8)\,\mu s$	$(11 \pm 6)\,\mu s$
Adaptive MPTSC with 11 cand.	$(25 \pm 2)\,\mu s$	$(6 \pm 3)\,\mu s$

$\beta = 5.37\,\mathrm{s} \cdot \mathrm{rad}^{-1}$ affects the approximation error of the static friction model. Note, this approximation is applied for both MPC and MPTSC to ensure a comparable analysis. Nevertheless, MPTSC could trivially handle the non-smooth version as well. Transforming equation (13) into a state space representation with state vector $\mathbf{x} = [x, \dot{x}]^\intercal = [x_1, x_2]^\intercal$ results in a second-order continuous-time system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \left[\dot{x}, \quad -\frac{\left(\tau_c \tanh(\beta\,\dot{x}) + d\,\dot{x}\right)}{J_2} + \frac{4\,k_1\,u}{J_2}\right]^\intercal. \quad (14)$$

The input variable, the electrical set-point current, is limited to $|u| \leq 1\,\mathrm{A}$ and the box constraint $|x_2| \leq 5\,\mathrm{rad/s}$ is set as active. The free parameters are set to $q_{11} = 1,\ q_{22} = 0.006,\ q_{12} = q_{21} = 0$ and $t_\mathrm{p} = 0.1\,\mathrm{s}$. Figure 6 shows the simulated closed-loop performance for this further benchmark system with zero model mismatch. The reference vector is $\mathbf{x}_\mathrm{f} = [x_\mathrm{ref}, 0]^\intercal$. Figure 6 demonstrates the capability of the MPTSC to handle state box constraints once again for the presented example system. Figure 6 and Figure 7 have similar results as Figure 3 and Figure 4. The subset $\mathbb{A}_k$ at every time step k consists of 15 possible input candidates regardless of the discretization method. The system performance differs between these two evaluated benchmark systems. The ECP system is a fairly inert system. Hence, the prediction horizon can be chosen five times shorter as in the case of the Van-der-Pol Oscillator. The error regarding the state vector $\mathbf{x}$ in Figure 6 and Figure 7 of the MPTSC with a linear input discretization reaches a level comparable to the MPTSC with an adaptive input domain discretization. If the frequent switching of the control input has no negative impact on the system properties, the MPTSC with a linear discretization can also be applied to meet the requirements regarding the rise time, the settling time and the overshoot characteristic. In Table II, the computation time of MPTSC

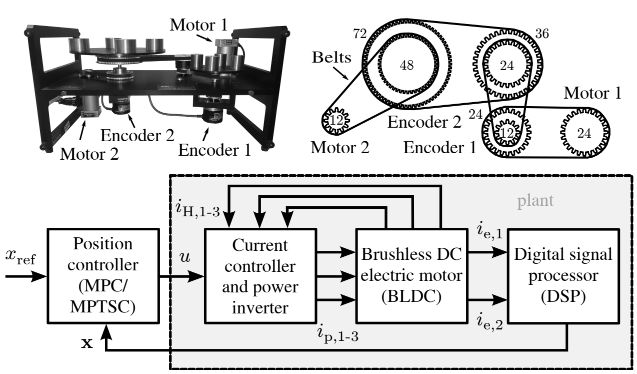

Fig. 5: Top: Industrial plant emulator ECP 220. Bottom: Block diagram when only one motor is used for control with the phase currents $i_{\mathrm{p},1-3}$, the signals of the hall-type sensors $i_{\mathrm{H},1-3}$ and the encoder signals $i_{\mathrm{e},1-2}$.

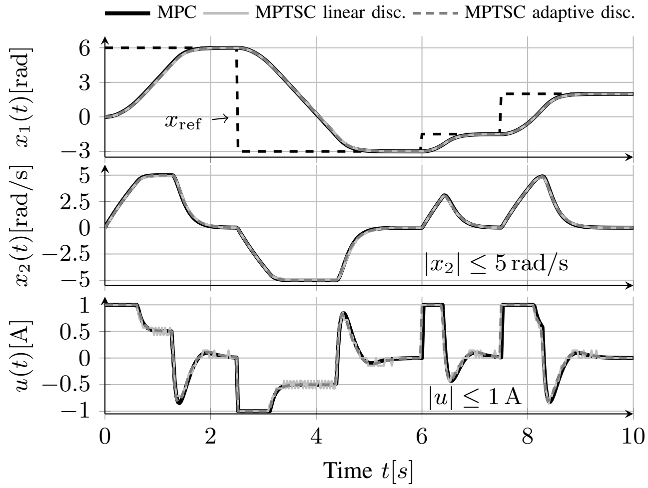

Fig. 6: Simulated closed-loop performance of the MPTSC for the ECP model.

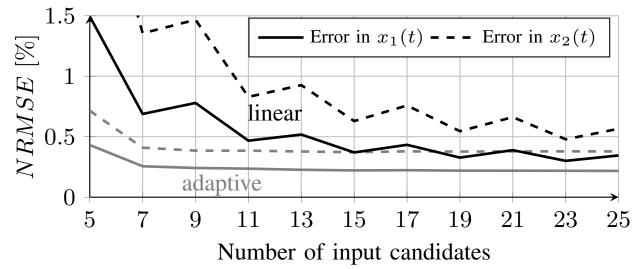

Fig. 7: Deviation of the sub-optimal MPTSC from the optimal MPC for the ECP model.

is significantly lower as the computation time of the MPC (hard constraint solver with just one allowed optimization iteration). The simulative analysis demonstrates promising closed-loop performances and leads to ideal preconditions to control a real system with mechatronic properties.

IV. Experimental Results

Theoretical investigations usually assume zero model mismatch. The model predictive control concepts focus on solving the optimal control problem efficiently by utilization of fast optimization techniques. In practice, a prediction model does never exhibit a zero error. Moreover, as the model accuracy increases the model becomes more complex. Thus, the computational effort increases, too. Furthermore, state estimation is performed by state observers or measurement units with limited accuracy. MPC approaches with $n_\mathrm{c} = 1$ lead to a lower closed-loop performance in comparison to MPC approaches with more degrees of freedom, especially in the case of zero model-mismatch. However, these approaches require iterative optimization algorithms. Section III demonstrated that the MPTSC achieves similar performance compared to MPC for two nonlinear systems. This section shows its performance in a real closed-loop experiment. Figure 8 illustrates the closed-loop performance of the MPTSC and the MPC for the ECP system close to the physical limits. Furthermore, the simulation results indicate the influence of the model mismatch, the encoder and DSP error. Both MPTSC and MPC are invoked with a sampling rate of $f_\mathrm{s} = 100\,\mathrm{Hz}$.

TABLE II: ECP model computation times

Approach	First open-loop	Closed-loop
MPC with IPOPT	$(3200 \pm 753)\,\mu s$	$(651 \pm 102)\,\mu s$
MPC with SQP	$(560 \pm 48)\,\mu s$	$(66 \pm 33)\,\mu s$
Adaptive MPTSC with 25 cand.	$(78 \pm 9)\,\mu s$	$(17 \pm 10)\,\mu s$
Adaptive MPTSC with 15 cand.	$(48 \pm 2)\,\mu s$	$(11 \pm 7)\,\mu s$

The adaptive MPTSC utilizes 21 input candidates. A closer look at the experimental results reveals that the system is initially operated at its maximum control bounds until state bound $|x_2| \leq 5$ rad/s becomes active. In comparison, time-optimal control would lead to slightly faster switches in the control input sequence. However, the extra effort in terms of implementing and setting up a real-time capable time-optimal controller with continuous optimization methods might not be justified from an industrial point of view.

V. CONCLUSIONS AND FUTURE WORK

The model predictive trajectory set control with an adaptive input domain discretization is presented in this paper. Polynomial functions are designed to enable the adaptive selection of possible input candidates to achieve a quasi-continuous input domain. Apparently, the sub-optimal MPTSC reproduces the closed-loop performance of the optimal MPC with one degree of freedom on the prediction horizon sufficiently well for different benchmark systems. At the same time, it is less time-consuming and less complex to implement without requiring any gradient-based optimization algorithms. These advantages address industrial applications in particular. In comparison to a classic nonlinear PID controller, the complexity regarding the number of design parameters can be reduced significantly. The experimental results with an industrial plant emulator emphasize these statements and demonstrate the superior closed-loop performance concerning the defined control task. Although MPTSC is less complex, it still integrates the systems state prediction with state feedback subject to constraints. It has thus shown to be suitable for replacing complex controllers in existing control architectures.

Future work is concerned with further comparisons to model predictive and alternative control concepts for a wider range of benchmark systems. Furthermore, a major interest is the theoretical analysis regarding stability and robustness.

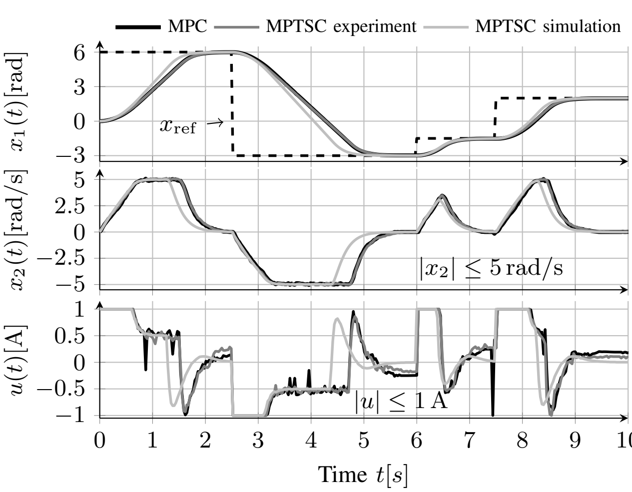

Fig. 8: Experimental closed-loop performance of the MPTSC for the industrial plant emulator ECP 220.

REFERENCES

[1] K. Åström and T. Hägglund, "The future of PID control," *Control Engineering Practice*, vol. 9, no. 11, pp. 1163–1175, 2001.

[2] J. Krettek, D. Schauten, F. Hoffmann, and T. Bertram, "Evolutionary hardware-in-the-loop optimization of a controller for cascaded hydraulic valves," in *International Conference on Advanced Intelligent Mechatronics (AIM)*, 2007.

[3] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[4] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, 2002.

[5] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[6] D. Kouzoupis, H. J. Ferreau, and M. Diehl, "First-order methods in embedded nonlinear model predictive control," in *European Control Conference (ECC)*, 2015.

[7] Y. Wang and S. P. Boyd, "Fast model predictive control using online optimization," in *IFAC World Congress*, vol. 17, 2008, pp. 6974–6979.

[8] A. Richards, "Fast model predictive control with soft constraints," in *European Control Conference (ECC)*, 2013.

[9] K. Graichen and B. Käpernick, "A real-time gradient method for nonlinear model predictive control," in *Frontiers of Model Predictive Control*, 2012.

[10] L. Van den Broeck, M. Diehl, and J. Swevers, "A model predictive control approach for time optimal point-to-point motion control," *Mechatronics*, vol. 21, no. 7, pp. 1203–1212, 2011.

[11] C. Rösmann, A. Makarow, F. Hoffmann, and T. Bertram, "Sparse shooting at adaptive temporal resolution for time-optimal model predictive control," in *Conference on Decision and Control (CDC)*, 2017.

[12] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," in *IEEE Conference on Decision and Control (CDC)*, vol. 2, 2004, pp. 2023–2028.

[13] H. A. Young, M. A. Perez, J. Rodriguez, and H. Abu-Rub, "Assessing finite-control-set model predictive control: A comparison with a linear current controller in two-level voltage source inverters," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 44–52, 2014.

[14] S. Kouro, M. A. Perez, J. Rodriguez, A. M. Llor, and H. A. Young, "Model predictive control: MPC's role in the evolution of power electronics," *IEEE Industrial Electronics Magazine*, vol. 9, no. 4, pp. 8–21, 2015.

[15] M. Keller, C. Hass, A. Seewald, and T. Bertram, "A model predictive approach to emergency maneuvers in critical traffic situations," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[16] A. Makarow, M. Keller, C. Rösmann, T. Bertram, G. Schoppel, and I. Glowatzky, "Model predictive trajectory set control for a proportional directional control valve," in *Conference on Control Technology and Applications (CCTA)*, 2017.

[17] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[18] D. Dunlap, C. V. Caldwell, and E. G. Collins, "Nonlinear model predictive control using sampling and goal-directed optimization," in *International Conference on Control Applications (CCA)*, 2010, pp. 1349–1356.

[19] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[20] "HSL. A collection of Fortran codes for large scale scientific computation." [Online]. Available: http://www.hsl.rl.ac.uk/

[21] 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.

[22] J. Nocedal and S. J. Wright, *Numerical optimization*, ser. Springer series in operations research. New York: Springer, 1999.