Sparse Shooting at Adaptive Temporal Resolution for Time-Optimal Model Predictive Control

Christoph Rösmann, Artemi Makarow, Frank Hoffmann and Torsten Bertram

IEEE Conference on Decision and Control (CDC), Melbourne, Australia, 2017

Sparse Shooting at Adaptive Temporal Resolution for Time-Optimal Model Predictive Control

Christoph Rösmann, Artemi Makarow, Frank Hoffmann and Torsten Bertram

Abstract— **This contribution presents a novel approach for time-optimal model predictive control. The underlying optimal control problem rests upon an adaptive, local temporal discretization of the shooting grid and automatically determines optimal switching points in the control sequence. Furthermore, the grid size is adapted online in order to solve the control problem with a minimum number of control interventions. The approach offers significant advantages for bang-bang control tasks which exhibit few transitions between a discrete set of piece-wise constant control actions. Experiments and a comparative analysis on different nonlinear control tasks demonstrate the superiority of adaptive shooting grids w.r.t. state-of-the-art approaches in model predictive control.**

I. INTRODUCTION

Model predictive control (MPC) repeatedly solves a finite horizon optimal control problem by taking the predicted future evolution as well as constraints on states and control variables into account [1]. Due to their large computational burden, within the past decade researchers investigated numerical efficient realizations of (nonlinear) MPC to extend their application to nonlinear systems with fast dynamics. In the context of continuous-time dynamic models, classical indirect optimal control methods are based on the calculus of variations, whereas direct methods transform the underlying optimal control problem into a finite parameter nonlinear program. Direct methods are furthermore categorized into either a sequential (single-shooting) or simultaneous (multiple-shooting, collocation) strategy. The sequential strategy merely discretizes the control input trajectory and thus requires the ongoing simulation of the future state evolution at each solver step. In comparison, simultaneous strategies discretize both state and control trajectories and usually achieve better convergence due to their sparse albeit larger problem structure. Diehl et al. propose multiple-shooting in order to partition the time horizon into multiple discrete intervals for which isolated initial value problems are solved [2]. Furthermore, the real-time iteration scheme applies only a single warm-started sequential-quadratic-programming step at each sampling interval to subsequently refine previous solutions during runtime [3]. Condensing techniques exploit the sparsity structure of nonlinear programs [4]. Interior-point methods enable the exploitation of sparse problem structures [5], [6], [7]. Graichen et al. present an efficient real-time capable MPC method based on projected gradients in [8]. In [9] the optimal control problem is transformed to an unconstrained auxiliary problem with interior penalties which is solved by an efficient gradient method.

In the context of time-optimal MPC, Zhao et al. apply multiple-shooting for quasi time-optimal control of a spherical robot [10]. Time-optimal reference path tracking is achieved by choosing sufficiently large time horizons in [11]. In race car automatic control applications the MPC control task is concerned with the minimization of the lap time [12], [13]. Verschueren et al. compute time-optimal motions along a Cartesian path for robotic manipulators [14]. Hereby, the time horizon is transformed into a fixed integration grid along the unit interval by applying an established time-scaling technique. A method called TOMPC minimizes the settling time of point-to-point transitions [15]. An outer optimization loop sequentially reduces the time horizon until the underlying standard quadratic form optimal control problem is no longer feasible. This iterative adaptation determines the minimal feasible horizon length. The number of outer loop iterations is determined by the gap between the initial and minimal horizon length. Our previous approach [16] introduces a time-optimal nonlinear MPC in which the global uniform temporal resolution of the discretized state and control trajectories is subject to optimization. Consequently, time-scaling becomes obsolete though the numerical properties are similar. The approach adapts the number of discretized states and controls to adhere to the desired temporal resolution of the discretization grid.

Time-optimal controllers usually operate the plant at either control or state limits. Thus, many practical time-optimal control problems consist of either bang-bang, bang-singular-bang or a small finite set of piece-wise constant controls. For these types of problems the number of effective control interventions at which the control input changes is significantly smaller than the ordinary temporal resolution of the MPC horizon, e.g. in [16] and time-scaling with a previously allocated fixed discretization grid. Our contribution addresses this circumstance by reformulating time-optimal MPC with a dynamic shooting grid. Individual time intervals of the multiple shooting grid are subject to optimization in order to enable the optimizer to explicitly determine the optimal time instances of switching the control signal w.r.t. the overall minimum time objective. The main feature of our approach is an online adaptation of the shooting grid size. The objective is a shooting grid in which the number of parameters that discretize the controls in the optimization becomes identical to the effective number of control switches without any loss of optimality. We phrase the term minimal control interventions for such a minimal grid size control sequence. The grid size

This work is funded by the German Research Foundation (DFG, BE 1569/13-1). The authors are with the Institute of Control Theory and Systems Engineering, Technische Universität Dortmund, D-44227, Germany, christoph.roesmann@tu-dortmund.de

adaptation is intertwined with the MPC control loop.

The next section presents the novel time-optimal MPC with dynamic grids. Section III provides a comparative analysis with state-of-the-art approaches and experimental results of a closed-loop position control of a servo drive system. Finally, section IV summarizes the results and provides an outlook on further work.

II. Time-Optimal Model Predictive Control

A. Uniform Grid Time-Optimal Control Formulation

A continuous-time nonlinear, autonomous dynamic system with state vector $\mathbf{x}(t) \in \mathbb{R}^p$, control input $\mathbf{u}(t) \in \mathbb{R}^q$ and initial state $\mathbf{x}_s$ is defined by:

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big), \quad \mathbf{x}(t=0) = \mathbf{x}_s. \tag{1}$$

The optimization task involves the solution of a boundary value problem of system (1) with final state $x(t = T) = \mathbf{x}_f$ at time T. Multiple-shooting [2] partitions the overall interval $[0, T]$ into n subintervals:

$$\begin{aligned} 0 = t_0 \leq t_0 + \Delta T_0 \quad &= t_1, \\ t_1 \leq t_1 + \Delta T_1 \quad &= t_2, \\ \vdots \quad & \\ t_{n-1} \leq t_{n-1} + \Delta T_{n-1} \quad &= t_n = T. \end{aligned} \tag{2}$$

Hereby, time intervals ΔT_k are usually but not necessarily chosen uniform. The control input trajectory $\mathbf{u}(t)$ is composed of piece-wise constant signals $\mathbf{u}_k$ on a temporal grid $\{t_0, t_1, \ldots, t_n\}$:

$$\mathbf{u}(t) := \mathbf{u}_k \quad \text{for} \quad t \in [t_k, t_k + \Delta T_k]. \tag{3}$$

Since multiple-shooting solves isolated initial value problems for each time interval, so called shooting nodes $\mathbf{s}_k := \mathbf{x}(t_k)$ are introduced for states on the grid t_k. The initial value problem on interval $[t_k, t_k + \Delta T_k]$ with control $\mathbf{u}_k$ and initial state $\mathbf{s}_k$ becomes:

$$\mathbf{x}(t_k + \Delta T_k, \mathbf{s}_k, \mathbf{u}_k) = \int_{t_k}^{t_k + \Delta T_k} \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}_k\big) \mathrm{d}t. \tag{4}$$

Carathéodory's existence theorem addresses conditions for a unique solution of (4). For the sake of simplicity, it is assumed that $\mathbf{f}(\cdot)$ is continuous and Lipschitz in $\mathbf{x}(t)$. Connectivity between subsequent intervals is ensured if $\mathbf{s}_{k+1} = \mathbf{x}(t_k + \Delta T_k, \mathbf{s}_k, \mathbf{u}_k)$ holds for all k. These algebraic equations are incorporated in the following nonlinear program to achieve a minimum time transition from $\mathbf{x}_s$ to $\mathbf{x}_f$:

$$\tilde{V}^* = \min_{\mathbf{s}_k, \mathbf{u}_k, T} \int_{t=0}^{T} 1 \, \mathrm{d}t = \min_{\mathbf{s}_k, \mathbf{u}_k, T} T \tag{5}$$

subject to

$$\begin{aligned} &\mathbf{s}_0 = \mathbf{x}_s, \quad \mathbf{s}_n = \mathbf{x}_f, \\ &\mathbf{s}_{k+1} = \mathbf{x}(t_{k+1}, \mathbf{s}_k, \mathbf{u}_k), \\ &\mathbf{g}(\mathbf{s}_k, \mathbf{u}_k) \geq \mathbf{0} \qquad\qquad (k = 0, 1, \ldots, n-1). \end{aligned}$$

The minimum objective function value is denoted as $\tilde{V}^*$ which coincides with the optimal transition time T^*. Initial $\mathbf{s}_0$ and final shooting node $\mathbf{s}_n$ are constrained by $\mathbf{x}_s$ and $\mathbf{x}_f$. Furthermore, inequality constraints $\mathbf{g} : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}^r$ such as saturation limits on states and controls are included. If the number of grid partitions n respectively distinctive constant controls is sufficiently large (at worst full discretization), the minimizer of (5) coincides with the minimum time solution in the (quasi) continuous-time domain. For smaller but still feasible dimensions the best (albeit suboptimal) solution for the current grid is achieved. In case the number of controls is significantly smaller the strategy still provides a possible realization of move-blocking MPC in which the reduced number of controls is explicitly traded for faster computation times. Note, the existence of a true minimizer in (5) for particular systems and constraints requires certain optimality, feasibility and differentiability conditions to be hold (refer to [17], [18]).

In the literature time-scaling constitutes a common approach to solve (5). Hereby, the systems dynamic equations are scaled by T: $\dot{\mathbf{x}}(t) = T\mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big)$ in order to map the time interval $[0, T]$ to $[0, 1]$, thus operating with a fixed underlying grid size which is independent of T (e.g. see [14]). In [16] the grid is dependent on a common uniform time interval $\Delta T_k := \Delta T$. Since $T = n\Delta T$ the optimal control problem (5) is expressed in terms of ΔT and optimization is performed w.r.t. ΔT rather than T which obviates time-scaling. Notice, the formulation in [16] utilizes finite differences instead of numerical integration. Although the grid resolution is dynamic (ΔT is subject to optimization), we categorize the approach as a uniform grid formulation in order to distinguish it from the dynamic grid scheme proposed in the following.

B. Dynamic Grid Time-Optimal Control Formulation

In contrast to the previous approaches, individual time intervals $\Delta T_k \in \mathbb{R}_0^+$ for $k = 0, 1, \ldots, n-1$ in the shooting grid (2) are now retained as explicit parameters subject to optimization. The proposed modified nonlinear program is defined as follows:

$$V^* = \min_{\mathbf{s}_k, \mathbf{u}_k, \Delta T_k} \sum_{k=0}^{n-1} \big[\Delta T_k + \lambda r(\Delta T_k)\big] \tag{6}$$

subject to

$$\begin{aligned} &\mathbf{s}_0 = \mathbf{x}_s, \quad \mathbf{s}_n = \mathbf{x}_f, \quad 0 \leq \Delta T_k, \\ &\mathbf{s}_{k+1} = \mathbf{x}(t_k + \Delta T_k, \mathbf{s}_k, \mathbf{u}_k), \\ &\mathbf{g}(\mathbf{s}_k, \mathbf{u}_k, \Delta T_k) \geq \mathbf{0} \qquad\qquad (k = 0, 1, \ldots, n-1). \end{aligned}$$

The term $r \colon \mathbb{R}_0^+ \rightarrow \mathbb{R}$ denotes a regularization term weighted by $\lambda \in \mathbb{R}_0^+$. Inequalities $\mathbf{g}(\cdot)$ in (5) are defined for grid points $\mathbf{s}_k$ and the grid is fine-grained respectively fixed in the uniform case. In case of the dynamic grid in (6) the optimizer itself separately adjusts time increments ΔT_k and thus shifts the grids temporal foundation. Consequently, inequalities either have to be independent of variable interval increments ΔT_k or take them explicitly into account. In order to transform the optimal control problem (5) to (6) without changing $\mathbf{g}(\cdot)$, the following assumption ensures that inequality constraints are independent of ΔT_k:

Assumption 1: Inequality constraints $\mathbf{g}(\mathbf{s}_k, \mathbf{u}_k, \Delta T_k)$ are satisfied for all intermediate states and controls on the interval ΔT_k and hence $\mathbf{s}_k$ can be substituted by any $\mathbf{x}(t)$ from $t \in [t_k, t_{k+1}]$ without violating the corresponding inequality constraints in (5).

Remark 1: If constraints depend only on control inputs, Assumption 1 is fulfilled since $\mathbf{u}(t)$ is constant for each ΔT_k (see (3)). For constraints involving the state evolution, this property depends on the system equations (1). E.g. the assumption is not satisfied if the optimal state trajectory satisfies constraints at $\mathbf{s}_k$ and $\mathbf{s}_{k+1}$ but not in the interior of $t \in [t_k, t_{k+1}]$ for some k. From a practical and implementation point of view, the assumption might be ensured by sufficiently oversampling the state trajectory of the initial value problem in the interval ΔT_k and by including the corresponding intermediate solutions as additional constraints.

Problem (6) with $\lambda = 0$ can be interpreted as a concatenation of n individual time-optimal control tasks with a constant control $\mathbf{u}_k$ each. For a sufficiently high resolution, optimality w.r.t. the quasi continuous-time solution follows from Bellman's principle of optimality. The following cases are distinguished: 1. If the number of time intervals n is larger than the minimum number of control interventions n^*, $n - n^*$ time interval parameters become redundant and problem (6) is under-constrained. 2. If n is smaller than n^* the optimal control problem might be either suboptimal w.r.t. quasi continuous-time or infeasible.

Remark 2: In case the particular nonlinear program solver does not handle under-constrained problems well, an additional regularization term $r(\Delta T_k) = \Delta T_k^2$ with a small weight λ ensures feasibility of the optimization. Notice, for $\lambda \gg 1$ the objective becomes $V_n \approx \sum_k \Delta T_k^2$ for which $\Delta T_k^* = T^*/n$ constitutes the minimizer in terms of time (proof follows by adding $\sum_k \Delta T_k = T^*$ with the method of Lagrange multipliers). This solution exactly corresponds to the uniform grid.

Note, in previous work [16] a uniform grid is utilized in terms of a single temporal parameter ΔT subject to optimization. Nonlinear program (6) exhibits a larger dimension for same lengths n due to the inclusion of individual time increments ΔT_k. This is compensated and further reduced by a significantly smaller overall dimension n required for many time-optimal control problems, especially those in which the control is composed of bang and *nice* singular arcs. The approach leads to a block-diagonal sparsity structure in the primal optimization variables since each ΔT_k only effects the two consecutive shooting nodes $\mathbf{s}_k$ and $\mathbf{s}_{k+1}$.

C. Online Regulation of Minimal Control Interventions

In time-optimal control tasks the plant usually operates either at state or control saturation limits. The optimal trajectory often emerges from a sequence of few piecewise constant controls $\mathbf{u}_k$ that include controls $\mathbf{u}_{\min}$, $\mathbf{u}_{\max}$ (bang arcs) or controls from the interior (singular arc). This *bang-singular-bang property* is generally proven for single-input nonlinear control-affine systems in the plane with bounds only on controls [19]. Several publications verify this property for certain classes of nonlinear systems. A detailed summary on analytical synthesis of control tasks is beyond the scope of this paper. In the following, we present a regulation mechanism that adapts the grid size and resolution in terms of the number elements (interventions) in the control sequence n online. The objective is to seek a grid with a minimum number of control interventions to realize the underlying time optimal and time continuous control signal.

Assumption 2: There exists a finite $n > 0$ for which the optimal control problem (5) is feasible and its solution constitutes a unique minimizer such that necessary and sufficient conditions hold.

This assumption is essential for every direct optimal control formulation. The reader is referred to [17], [18], [20] for further details. The minimal number of control interventions with (6) is determined in an iterative manner by concurrently testing the effect on an increase of an interventions to $n+1$ or decrease to $n-1$ [21]. If the optimal transition time V^* is retained for a decrease, at least one control intervention is obsolete. If the optimal transition time V^* is improved with an increase, the current temporal grid structure is suboptimal. Testing and regulation of n requires multiple nonlinear programs to be solved in parallel and the linear search substantially depends on the initial length n_0 which is not preferable for an online integration. Therefore, redundant control interventions are identified by analyzing the control input trajectory obtained from the previous solution of the nonlinear program (6):

Algorithm 1 Regulation of number of control interventions

```
 1: procedure AdaptAndSolveNlp($\mathbf{b}, n_b^*$)
 2:   for all Iterations $i = 1, 2, \dots, I$ do
 3:     if $i > 1$ or $\mathbf{b}$ is a warm-start then
 4:       $\{n_b, \mathcal{K}\} \leftarrow$ Count validity of $|\mathbf{u}_{k+1} - \mathbf{u}_k| \leq \boldsymbol{\epsilon}\ \forall k$ in $\mathbf{b}$
 5:       if $n_b < n_b^*$ then
 6:         $\mathbf{b} \leftarrow$ insertVariables($\mathbf{b}, n_b^* - n_b$)
 7:       else if $n_b > n_b^*$ then
 8:         $\mathbf{b} \leftarrow$ erase $\mathbf{u}_{k+1}$, $\mathbf{s}_{k+1}$ and $\Delta T_{k+1}$ in $\mathbf{b}\ \forall k \in \mathcal{K}$

 9:     $\{V^*, \mathbf{b}^*\} \leftarrow$ solveNLP($\mathbf{b}$)                ▷ solve (6)

10:     if $V^*$ is non-feasible then
11:       $\mathbf{b} \leftarrow$ insertVariables($\mathbf{b}^*, 1$)
12:       Goto 9 and resolve
13:   return $\mathbf{b}^*$
```

Hereby, $\mathbf{b}$ denotes the current parameter vector subject to optimization: $\mathbf{b} = [\mathbf{s}_0^\intercal, \mathbf{u}_0^\intercal, \Delta T_0, \mathbf{s}_1^\intercal, \mathbf{u}_1^\intercal, \Delta T_1, \dots, \mathbf{s}_{n-1}^\intercal, \mathbf{u}_{n-1}^\intercal, \Delta T_{n-1}, \mathbf{s}_n^\intercal]^\intercal$ with $n = n_0$ at the first iteration. The second argument $n_b^* \in \mathbb{N}_{\geq 0}$ denotes a desired surplus in the number of control interventions as explained below. The initial $\mathbf{b}$ is obtained from a linear interpolation $\mathbf{s}_k = \mathbf{x}_s + kn^{-1}(\mathbf{x}_f - \mathbf{x}_s)$ between start and final state with zero controls $\mathbf{u}_k = \mathbf{0}$ in this work or later in section II-D from previous closed-loop sampling intervals (warm-start). If $\mathbf{b}$ does not constitute a warm-start, the procedure first solves problem (6) in line 9. In subsequent iterations the control input sequence is investigated for potentially redundant control interventions n_b in terms of equality of subsequent controls $|\mathbf{u}_{k+1} - \mathbf{u}_k| \leq \boldsymbol{\epsilon}$. Theoretically, one expects $\boldsymbol{\epsilon} = \mathbf{0}$ but in practical implementations a small but finite margin accounts for numerical imprecision. The indices k of redundant controls are gathered in set $\mathcal{K}$. If $n_b < n_b^*$, the grid is augmented by $n_b^* - n_b$ additional intermediary shooting nodes, controls and time intervals. Insertion is performed subsequently for the currently largest time interval $\max\{\Delta T_k | \forall k\}$ by linear interpolation. In case $n_b > n_b^*$, the redundant grid points and parameters with indices $\mathcal{K}$ are removed from the control sequence.

Remark 3: If there is no surplus of grid points $n_b^* = 0$ and the initial n_0 is overestimated, problem (6) is underconstrained such that $n_b > 0$ and redundant grid points are removed. At this point convergence is reached. On the other hand, if n_0 is underestimated such that the NLP solution becomes infeasible, a new intermediary grid point is inserted in line 11 and the nonlinear program is resolved with the augmented grid structure. In case n_0 is underestimated but the NLP solution is suboptimal albeit feasible, all subsequent controls differ ($\mathcal{K} = \emptyset$), thus the algorithm does not enhance the grid structure even though the true optimal solution requires additional interventions. Consequently, $n_b^* > 0$ is crucial for recovering from suboptimal solutions with too few interventions. n_b^* is chosen such that $V^*(n) > V^*(n + n_b^*)$ holds for all suboptimal n. In practice, $n_b^* = 1$ is often sufficient for a robust recovery.

Notice, for the online integration of Alg. 1 the nonlinear program solver in line 9 might be terminated prior to convergence. A suitable scheduling of solver iterations significantly increases the overall convergence speed. Due to the sparse structure, partial solutions are likely to indicate redundant control interventions at an early stage of convergence. These grid points are removed prior to complete convergence with the benefit of reducing the number of parameters in subsequent iterations. In case of limited computational resources, the number of control interventions is determined a priori by invoking Alg. 1 on samples of initial states.

D. Closed-Loop Control

In this section the open-loop optimization is integrated with state feedback in order to regulate system (1) to the final target state $\mathbf{x}_f$. At each sampling interval the algorithm operates according to:

Algorithm 2 MPC step invoked at each sampling instance

```
1: procedure FeedbackControlStep($\mathbf{b}, \mathbf{x}_s, \mathbf{x}_f$)
2:     Initialize or update trajectory
3:     $\mathbf{b}^* \leftarrow$ AdaptAndSolveNlp($\mathbf{b}, n_b^*$)
4:     return $\{\mathbf{u}_0^*, \mathbf{b}^*\}$        ▷ $\mathbf{u}_0^*$ is the first control in $\mathbf{b}^*$
```

The current plant state $\mathbf{x}_s$ is either directly measurable or estimated by a state observer. $\mathbf{b}$ denotes the parameter vector fed back from the previous sampling interval. The initial solution is constructed as described in section II-C. In line 2 the parameter vector is updated by replacing $\mathbf{s}_0$ and $\mathbf{s}_n$ by the most recent states $\mathbf{x}_s$ and $\mathbf{x}_f$ respectively. Optimization along with the grid structure adaption is performed in line 3. From the resulting parameter vector $\mathbf{b}^*$ the imminent control action $\mathbf{u}_0^*$ is applied as input to the plant.

Stability properties in the context of MPC are summarized in [18], [20]. MPC with final state constraints ($\mathbf{s}_n = \mathbf{x}_f$) as in (6) are shown to be stable in the absence of disturbances and model mismatch if the initial solution is already feasible [20]. The time-optimal objective function in (6) is strictly monotonically decreasing towards the target state $\mathbf{x}_f$. In case Alg. 1 converges in every MPC step (refer to the discussion in Remark 3) stability follows from Bellman's principle of optimality under the above model and disturbance assumptions.

III. Evaluation and Experiments

This section evaluates the proposed approach on two simulated nonlinear control problems and demonstrates its feasibility experimentally for feedback control of an industrial servo drive. Optimal control problem (6) is solved with IPOPT [22] and HSL-MA57 as internal linear solver [23]. IPOPT constitutes a C++ interior point solver for sparse nonlinear programs. For the evaluation part IPOPT is provided with numerically computed Jacobian and Hessian matrices to avoid convergence effects that tend to emerge in iterative BFGS methods [17]. In our implementation, the sparsity structure is further exploited by representing the complete nonlinear program as a hyper-graph which vertices denote optimization parameters in $\mathbf{b}$ and which edges denote the equality and inequality constraints as well as the summands of the objective function. Dense block Jacobian and Hessian submatrices are computed edge-wise and finally all submatrices are combined into the overall sparse Jacobian resp. Hessian matrices. For the examples below, a dedicated regularization is omitted with $\lambda = 0$. Computations are performed in C++ (PC: 3.4 GHz Intel i7 CPU, Ubuntu).

A. Free-Space Rocket System

The free-space rocket system constitutes a common benchmark in the MPC literature:

$$\begin{aligned} \dot{s}(t) &= v(t), \\ \dot{v}(t) &= (u(t) - 0.02\, v(t)^2)/m(t), \\ \dot{m}(t) &= -0.01\, u(t)^2. \end{aligned} \qquad (7)$$

Hereby, $s(t)$ denotes the distance traveled, $m(t) \geq 0$ denotes the mass and $v(t)$ the rocket's velocity which is bounded to $-0.5 \leq v(t) \leq 1.7$. The control input $u(t)$ is limited to $|u(t)| \leq 1$. With state vector $\mathbf{x}(t) = [s, v, m]^\intercal$ the system dynamics are expressed in the form $\mathbf{f}(\cdot)$. Furthermore, inequality constraints are combined to $\mathbf{g}(\mathbf{s}_k, u_k) = [u + 1, -u + 1, v + 0.5, -v + 1.7, m]^\intercal \geq \mathbf{0}$. Note, $\mathbf{g}(\cdot)$ satisfies Assumption (1). Whenever the velocity bound is active between two consecutive shooting nodes the constraint $\dot{v} = 0$ is imposed which is accomplished by $u_k = 0.02 \cdot 1.7^2 = 0.0578$ in case of the upper bound. This particular u_k is always feasible which applies to the other state bounds as well. The control task is to transit from the initial state $\mathbf{x}_s = [0, 0, 1]^\intercal$ to the target state $\mathbf{x}_f = [s_f, 0, \cdot]^\intercal$ in minimum time with $s_f \in \{10, 30\}$. As the final mass m_f is a priori unknown the final state m is subject to optimization. Numerical integration of (4) is performed with forward Euler and a step width

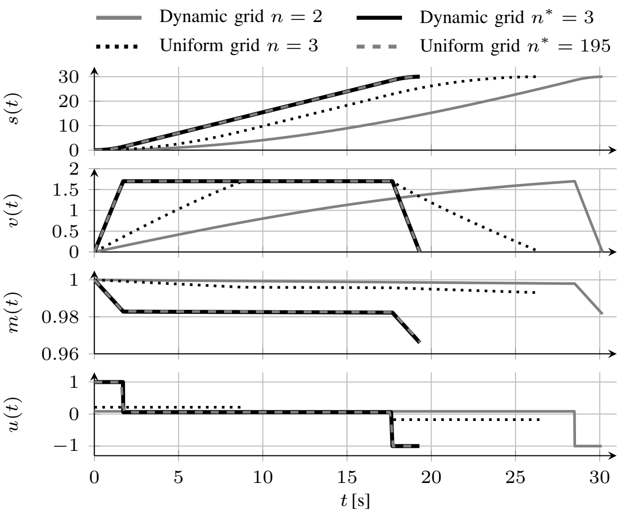

Fig. 1: Open-loop control of the free-space rocket system

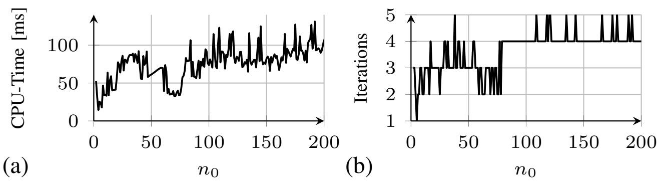

Fig. 2: Convergence analysis of Alg. 1: (a) CPU time and (b) number of iterations until n^* is reached

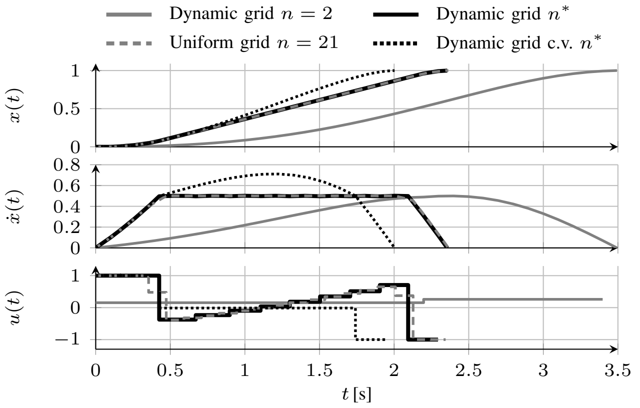

Fig. 3: Open-loop control of the Van-der-Pol Oscillator

of 0.1 s. Fig. 1 shows the resulting state and control input trajectories for different controllers and $s_f = 30$. For comparison, the uniform grid approach presented in [16], time-scaling and TOMPC are taken into account. Time-scaling and TOMPC are omitted in Fig. 1 since their trajectories coincide with the uniform grid solution for $n = 195$. The resolution of $n = 195$ is chosen in order to accomplish a comparable resolution of $\Delta T = 0.1\,\mathrm{s}$ (full discretization). For both uniform grid approaches fewer states sacrifice upon global optimality (refer to the uniform grid with $n = 3$). Note, TOMPC always converges towards the optimal trajectory with a linear search in terms of increasing or decreasing n. The dynamic grid approach perfectly coincides with the $n = 195$ uniform solution sequence with $n^* = 3$ states and hence three control interventions only. A suboptimal solution for $n = 2$ is shown as well. Table I reports computations times for both $s_f = 30$ and a closer target $s_f = 10$ with $T^* = 7.7\,\mathrm{s}$. References of computational performance are evaluated in two configurations: 1. full discretization, i.e. $n = 77$ for $s_f = 10$, and 2. at least a NRMSE of 0.5 % w.r.t. the dynamic grid, i.e. $n = 40$ for $s_f = 10$ and $n = 120$ for $s_f = 30$. Notice, the CPU time of TOMPC is significantly larger even for a perfect initial guess n^* since at each step at least two optimal control problems are solved and the objective function includes the evaluation of quadratic form terms for all shooting nodes. In order to analyze the convergence properties of Alg. 1 the control task with $s_f = 10$ is considered for varying initial control sequence lengths n_0. IPOPT is configured to terminate after ten iterations for each outer loop iteration of Alg. 1. Fig. 2a shows the CPU times until convergence w.r.t. n_0. It is quite remarkable that IPOPT displays a convergence boost for an approximately uniform discretization at around $n_0 = 77$. The number of outer loop iterations I until convergence towards $n^* = 4 + n_b^*$ with $n_b^* = 1$ is depicted in Fig. 2.

TABLE I: Rocket system computation times

Method	CPU Time $s_f = 10$	CPU Time $s_f = 30$
Dynamic grid $n^* = 3$	(15.3 ± 0.4) ms	(64.5 ± 0.7) ms
Uniform grid $n = 40$; 120	(42.1 ± 0.6) ms	(113.7 ± 1.2) ms
Uniform grid $n^* = 77$; 195	(65.1 ± 1.0) ms	(175.0 ± 1.8) ms
Time-scaling $n = 40$; 120	(43.7 ± 3.4) ms	(80.0 ± 3.1) ms
Time-scaling $n^* = 77$; 195	(70.8 ± 3.1) ms	(132.0 ± 3.6) ms
TOMPC $n^* = 77$; 195	(396.8 ± 1.9) ms	(436.0 ± 4.1) ms

B. Van-der-Pol Oscillator

The Van-der-Pol system $\ddot{x}(t) + (x^2(t) - 1)\dot{x}(t) + x(t) = u(t)$ constitutes an oscillatory dynamic system with nonlinear damping. With state vector $\mathbf{x} = [x(t), \dot{x}(t)]^\intercal$ and control input $|u(t)| \leq 1$ the state space model is defined by:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = [\dot{x}, \quad -(x^2 - 1)\dot{x} - x + u]^\intercal. \tag{8}$$

The second state is constrained to $|\dot{x}(t)| \leq 0.5$. Inequalities $\mathbf{g}(\cdot)$ are constructed according to the procedure in Section III-A. The optimal control problem demands a transition from $\mathbf{x}_s = [0, 0]^\intercal$ to $\mathbf{x}_f = [1, 0]^\intercal$ in minimum time. Notice, the state constraint introduces a non bang-bang control type since $-x(t) + u(t) = 0$ must hold for an active bound with $\dot{x}(t) = 0$ and implies a linear dependency on the non-constant state $x(t)$. Consequently, Assumption 1 might become invalid. According to Remark 1, two intermediate state constraints are incorporated for $\mathbf{g}(\cdot)$. The resulting trajectories are depicted in Fig. 3. The constraint violation case (c.v.) without the above modification is presented as well. The reference approach (uniform grid) with $n = 21$ samples exhibits the (quasi) linear arc in the control sequence for the active state bound. For the dynamic grid, Alg. 1 is invoked with $n_0 = 20$. Obviously, the dynamic grid matches the optimal theoretical switching points more precisely (the overall transition is 5 ms faster). The suboptimal solution for a dynamic grid with $n = 2$ is shown as well.

C. Closed-Loop Control Experiment

This section investigates feedback control of an *ECP Industrial Plant Emulator Model 220*. The system consists of

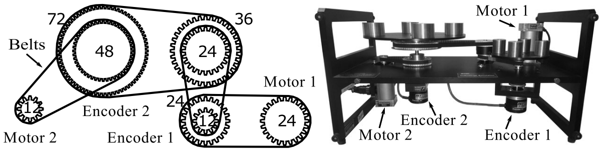

Fig. 4: ECP Industrial Plant Emulator Model 220

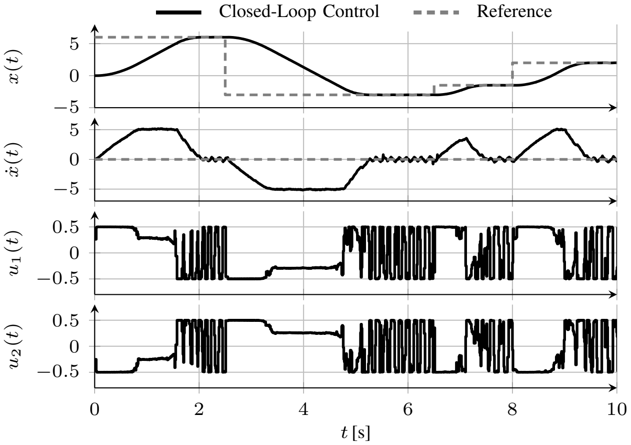

Fig. 5: Closed-loop control of the ECP220 system

two load plates actuated by motors which motion is coupled by transmission belts (see Fig. 4). Encoder readings provide a direct measurement of angular position and the angular velocity is estimated from encoder signals with a DSP.

In the experimental setup, both motor generates actuation torques in order to regulate the position and velocity of the plate of the secondary drive. The system dynamics with nonlinear damping and two control inputs are given by:

$$\ddot{x}(t) = 4(k_1 u_1(t) - k_2 u_2(t))/J - \big(\tau_c \tanh(\alpha \dot{x}(t) + d\dot{x}(t)\big)/J \tag{9}$$

with $k_1 = 9.5 \cdot 10^{-2}$, $k_2 = 7.5 \cdot 10^{-2}$, $J = 3.39 \cdot 10^{-2}$, $\tau_c = 9.39 \cdot 10^{-2}$, $\alpha = 5.37$ and $d = 1.93 \cdot 10^{-2}$. The optimal control problem is constructed similar to Section III-B with state vector $\mathbf{x}(t) = [x(t), \dot{x}(t)]^\intercal$, control bounds $|u(t)| \leq 0.5$ and velocity bounds $|\dot{x}(t)| \leq 5$. Note, that the simplified dynamic equations exhibit a non-trivial model mismatch w.r.t. the true coupled drive dynamics. For the closed-loop control experiment, four target positions are commanded as shown in Fig. 5. The controller is only aware of the current target position (marked with dashed lines) and does not look-ahead beyond the single step reference. Numerical integration is performed with 5th-order Runge-Kutta with a step width of $0.1\,\text{s}$ and the MPC operates at $100\,\text{Hz}$.

IV. Conclusions and Future Work

The comparative analysis of the proposed time-optimal control formulation confirms that the dynamic shooting grid achieves time-optimality with a significantly smaller number of control interventions compared to current state-of-the-art methods. This reduction in problem dimension is especially beneficial for optimal control sequences composed of bang and *nice* singular arcs. Furthermore, the online adaption of the shooting grid resolution automatically determines the minimum number of sufficient control interventions either during runtime or a-priori offline. The experimental results demonstrate the practical feasibility of the dynamic grid MPC formulation for feedback control of a servo drive.

Future work investigates collocation methods subject to dynamic discretization grids.

References

[1] 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.

[2] 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.

[3] 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.

[4] G. Frison, D. Kouzoupis, J. B. Jørgensen, and M. Diehl, "An efficient implementation of partial condensing for nonlinear model predictive control," in *IEEE Conference on Decision and Control (CDC)*, 2016.

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

[6] A. Richards, "Fast model predictive control with soft constraints," in *European Control Conference*, 2013, pp. 1–6.

[7] K. S. Pakazad, H. Ohlsson, and L. Ljung, "Sparse control using sum-of-norms regularized model predictive control," in *IEEE Conference on Decision and Control*, 2013.

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

[9] B. Käpernick and K. Graichen, "Nonlinear model predictive control based on constraint transformation," *Optimal Control Applications and Methods*, vol. 37, no. 4, pp. 807–828, 2016.

[10] J. Zhao, M. Diehl, R. Longman, H. G. Bock, and J. P. Schlöder, "Nonlinear model predictive control of robots using real-time optimization," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2004.

[11] D. Lam, "A model predictive approach to optimal path-following and contouring control," PhD Thesis, The University of Melbourne, 2012.

[12] D. P. Kelly and R. S. Sharp, "Time-optimal control of the race car: a numerical method to emulate the ideal driver," *Vehicle System Dynamics*, vol. 48, no. 12, pp. 1461–1474, 2010.

[13] J. P. Timings and D. J. Cole, "Minimum manoeuvre time of a nonlinear vehicle at constant forward speed using convex optimisation," in *International Symposium on Advanced Vehicle Control*, 2010.

[14] R. Verschueren, N. van Duijkeren, J. Swevers, and M. Diehl, "Time-optimal motion planning for n-dof robot manipulators using a path-parametric system reformulation," in *Proceedings of the American Control Conference (ACC)*, 2016.

[15] 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.

[16] C. Rösmann, F. Hoffmann, and T. Bertram, "Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control," in *European Control Conference (ECC)*, 2015.

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

[18] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*, 2nd ed., ser. Communications and Control Engineering. Springer, 2017.

[19] H. J. Sussmann, "The structure of time-optimal trajectories for single-input systems in the plane: The general real analytic case," *SIAM J. Control Optim*, vol. 25, no. 4, pp. 868–904, 1987.

[20] 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.

[21] C. Rösmann, A. Makarow, F. Hoffmann, and T. Bertram, "Time-optimal nonlinear model predictive control with minimal control inverentions," in *IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 19–24.

[22] 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.

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