# Timed-Elastic-Bands for Time-Optimal Point-to-Point Nonlinear Model Predictive Control

Christoph Rösmann, Frank Hoffmann and Torsten Bertram

*Abstract*— This contribution presents a novel approach for nonlinear time-optimal model predictive control (MPC) based on Timed-Elastic-Bands (TEB). The TEB merges the states, control inputs and time intervals into a joint trajectory representation which enables planning of time-optimal trajectories in the context of model predictive control. Model predictive control integrates the planning of the optimal trajectory with state feedback in the control loop. The TEB approach formulates the fixed horizon optimal control problem for point-to-point transitions as a nonlinear program. The comparative analysis of the TEB approach with state-of-the-art approaches demonstrates its computational efficiency. The TEB approach generates a trajectory that approximates the analytical time-optimal trajectory in few iterations. This efficiency enables the refinement of the planned state and control sequence within the underlying closed-loop control during runtime.

## I. INTRODUCTION

As technical processes become progressively more complex, the demands for their control and automation increase as well. Hence advanced control concepts that explicitly consider constraints on control and state variables gain importance in research and applications. In this context *model predictive control* (MPC) provide a means to repeatedly solve a receding horizon optimal control problem for nonlinear dynamic processes with multiple inputs and outputs [1]. In process automation, especially in the field of chemical engineering, model predictive controllers are well established [2]. Solving optimal control problems under constraints is computationally demanding and thus until recently has been only applied to slow processes.

In the past decade the advancement of control schemes for mechatronic systems caused an interest in numerically efficient implementations of MPC. The majority of approaches in MPC are concerned with the minimization of quadratic cost functionals with respect to control error and effort. These conventional formulations of MPC objectives are not directly applicable to time-optimal control tasks.

Direct methods discretize the states and controls and thereby approximate the calculus of variations by a finite parameter nonlinear program. Diehl et al. solve the optimal control problem with an efficient multiple-shooting approach [3]. The approach rests upon multiple partitions of the time horizon into discrete intervals, for which isolated initial value problems are solved in parallel. In [4] the original approach is extended to a real-time iteration scheme, which subsequently refines an initial coarse approximation at runtime. Recent methods reformulate the optimization problem

The authors are with the Institute of Control Theory and Systems Engineering, Technische Universität Dortmund, D-44227, Germany, `christoph.roesmann@tu-dortmund.de`

in order to allow a more efficient computation based on inner-point-methods [5], [6], [7]. The nonlinear program is simultaneously solved with respect to controls and states. Although the resulting problem possesses more variables, its structure remains sparse which enables its numerical efficient optimization. Vukov et al. treat long horizons with automatic code generation strategies that explicitly exploit structure [8]. An efficient method based on projected gradients is presented in [9] within a real-time capable MPC scheme. Regarding linear systems, the approach by Zeilinger et al. guaranties stability and feasibility under hard real-time conditions [10].

Explicit MPCs solve the general parameterized optimization problem offline. The optimal control action for the current state and target is extracted from a precomputed lookup table [11]. The explicit approach requires an a-priori discretization of the operational space. The lookup table representation has the drawback that the memory requirements grow exponentially with the dimension of the state space, thus limiting the approach to low order dynamics.

The above mentioned approaches mainly consider cost functionals that penalize a combination of control error and effort. In case of time-optimal point-to-point transitions, conventional approaches are not readily applicable for real-time applications. Explicit cost terms that minimize the settling time are considered in [12] for a quasi time-optimal control of a spherical robot. The approach is based on an indirect solution of the problem in terms of calculus of variations and a time transformation of the continuous system dynamics. The cost functional is adapted in the vicinity of the goal state in order to minimize the control effort rather than time for the final control steps. A drawback of indirect methods is the difficulty of handling inequality constraints and the strong dependence on the initial solution.

The concept of direct MPC is extended to time-optimal point-to-point transitions in [13]. The method called TOMPC minimizes the settling time in a two layer optimization routine. The outer loop incrementally reduces the horizon of the control sequence until the inner loop nonlinear program no longer generates a feasible solution for the allocated time horizon. Since the objective function minimizes the distance of discrete states to the final state, the solution with the shortest feasible horizon is quasi time-optimal. Due to the lower bound on the time horizon, the algorithm behaves like a conventional MPC in the vicinity of the goal state and therefore guarantees stability. The runtime strongly depends on the initial estimation of the settling time, as it determines the number of iterations in the outer loop time horizon reduction. An alternative approach that follows a reference

path in minimal time is presented in [14]. Time-optimality is nearly achieved in case of long time horizons. In applications such as race car automatic control, MPC methods are used to minimize the lap time [15], [16]. Explicit MPC is extended to the offline computation of time-optimal tasks for linear time-invariant and piecewise affine systems [17].

This work presents a novel approach to time-optimal MPC for point-to-point transitions of nonlinear dynamic systems. The concept is based on *Timed-Elastic-Bands* (TEB) for the representation of the underlying state and control sequence. Originally, the TEB approach is designed to deform and optimize trajectories of mobile platforms. The classic *Elastic-Bands* developed by Quinlan et al. deform an initially coarsely planned, pure geometric path under the presence of internal and external forces [18]. Internal forces contract the path to obtain the shortest path without detours, while external forces maintain a separation between the path and obstacles. [19] presents the TEB approach as an optimization based trajectory deformation with a sparse structure and the explicit incorporation of temporal information. Kinodynamic constraints and nonholonomic kinematics are taken into account. Keller et al. extend the TEB to the planning of optimal collision maneuvers for automotive applications [20].

The utilization of TEBs for MPCs requires the extension of the trajectory representation to state and control sequences that are mutually constrained by the system dynamics while preserving a sparse structure and temporal representation. The optimization problem is formulated as a nonlinear program that is efficiently solved with online active-set or interior-point methods. In contrast to conventional MPCs that operate either in continuous time domain or in discrete time domain using a fixed sample rate, the TEB approach retains the discrete time interval between consecutive states as an explicit optimization parameter. This strategy allows the contraction and expansion of the control and state sequence with respect to the transition time, leading to a quasi time-optimal control. The analogy to deforming elastic bands with fixed start and goal in state space motivates the term TEB for the presented MPC strategy.

The next section introduces the general concept of the TEB based MPC approach, in particular formulating and solving the open-loop optimization problem and performing the closed-loop control. Section III presents examples and simulations of the proposed method with a focus on the comparison with a state-of-the-art approach for time-optimal MPC called TOMPC [13]. Finally, section IV summarizes the results and provides an outlook on further work.

## II. TIMED-ELASTIC-BAND BASED MPC APPROACH

### A. Formulation of the Open-Loop Optimization Problem

A nonlinear, autonomous dynamic system with $p$ states and $q$ inputs is defined by:

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

in which $\mathbf{x} \in \mathbb{R}^p$ denotes the time dependent state and $\mathbf{u} \in \mathbb{R}^q$ the corresponding control input. $\mathbf{x}_s \in \mathbb{R}^p$ denotes

the initial state at time $t = 0\,\mathrm{s}$. This system of continuous differential equations is approximated and discretized by finite differences. The approach assumes forward differences $\dot{\mathbf{x}}(t) \approx \Delta T^{-1}(\mathbf{x}_{k+1} - \mathbf{x}_k)$ since for most practical scenarios their accuracy is sufficient to correct the model mismatch during closed loop control. The discretized system dynamics is given by:

$$\Delta T^{-1}(\mathbf{x}_{k+1} - \mathbf{x}_k) = \tilde{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) \quad k = 1, 2, \ldots, n-1 \quad (2)$$

This approximation allows a discrete parameterization of the state and control sequence with $n$ states and $n-1$ controls in combination with a strictly positive temporal discretization $\Delta T \in \mathbb{R}^+$. These parameters are lumped together into the TEB set $\mathcal{B} \subseteq \mathbb{R}^{\tilde{d}}$ with $\tilde{d} = np + (n-1)q + 1$:

$$\mathcal{B} := \{\mathbf{x}_1, \mathbf{u}_1, \mathbf{x}_2, \mathbf{u}_2, \ldots, \mathbf{x}_{n-1}, \mathbf{u}_{n-1}, \mathbf{x}_n, \Delta T\} \quad (3)$$

In contrast to the TEB formulation in [19], which reconstructs the control sequence $\mathbf{u}_k$ by inverting (2), our novel approach considers the control as an explicit optimization parameter. The TEB strategy incorporates the temporal discretization $\Delta T$ into the optimization problem. Furthermore, within each sampling interval of the closed loop control, the number of samples $n$ is adjusted such that it approximately matches the underlying sample rate. This flexibility with respect to the temporal discretization offers advantages for time-optimal control compared with MPC approaches that rely on a fixed temporal discretization.

The open loop optimization task involves the planning of the controls for (2) in order to transit from the initial state $\mathbf{x}_s$ to the final target state $\mathbf{x}_f$ in minimal time $T$. According to (3), the transition time $T \in \mathbb{R}^+$ is determined by $T \approx (n-1)\Delta T$. The optimal control sequence is obtained by solving the following nonlinear program:

$$V^*(\mathcal{B}) = \min_{\mathcal{B}} \quad (n-1)\Delta T \quad (4)$$

subject to

$$\mathbf{x}_1 = \mathbf{x}_s, \quad \mathbf{x}_n = \mathbf{x}_f, \quad \Delta T > 0$$
$$\mathbf{h}_k(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k, \Delta T) = \mathbf{0} \quad (k = 1, 2, \ldots, n-1)$$
$$\mathbf{g}_1(\mathbf{u}_1) \geq \mathbf{0}$$
$$\mathbf{g}_k(\mathbf{x}_k, \mathbf{u}_k) \geq \mathbf{0} \quad (k = 2, 3, \ldots, n-1)$$

Initial $\mathbf{x}_1$ and final TEB state $\mathbf{x}_n$ are constrained by $\mathbf{x}_s$ and $\mathbf{x}_f$ and thus not subject to TEB optimization. The equality constraints $\mathbf{h}_k(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k, \Delta T) = \Delta T^{-1}(\mathbf{x}_{k+1} - \mathbf{x}_k) - \tilde{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{0}$ originate from the system dynamics (2). Inequality constraints $\mathbf{g}_1 : \mathbb{R}^q \to \mathbb{R}^r$ are imposed on the first control $\mathbf{u}_1$. Further controls and states are restricted by inequality constraints $\mathbf{g}_k : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^r, k = 2, 3, \ldots, n-1$. The linear objective function $V(\mathcal{B}) = (n-1)\Delta T$ is bounded from below since $\Delta T$ is strictly positive. Notice, that even for linear systems, which implies that (1) is linear, the optimization problem (4) contains nonlinear equality constraints and therefore can no longer be solved as a basic linear program.

### B. Optimality of the TEB Open-Loop Optimization Problem

Initial state $\mathbf{x}_1$ and final state $\mathbf{x}_n$ are fixed during optimization and substituted in the cost function and constraints

by the constants $\mathbf{x}_s$ and $\mathbf{x}_f$. The reduced optimization vector $\mathbf{b} \in \mathcal{B} \setminus \{\mathbf{x}_1, \mathbf{x}_n\} \subseteq \mathbb{R}^d$ with $d = p(n-2) + q(n-1) + 1$ is defined by

$$\mathbf{b} := [\mathbf{u}_1, \mathbf{x}_2, \mathbf{u}_2, \ldots, \mathbf{x}_{n-1}, \mathbf{u}_{n-1}, \Delta T]^\mathsf{T} \qquad (5)$$

The vector-valued equality constraints are given by $\mathbf{h}(\mathbf{b}, \mathbf{x}_s, \mathbf{x}_f) = [\mathbf{h}_1^\mathsf{T}, \mathbf{h}_2^\mathsf{T}, ..., \mathbf{h}_{n-1}^\mathsf{T}]^\mathsf{T}$ and the inequalities by $\mathbf{g}(\mathbf{b}, \mathbf{x}_s, \mathbf{x}_f) = [\mathbf{g}_1^\mathsf{T}, \mathbf{g}_2^\mathsf{T}, ..., \mathbf{g}_{n-1}^\mathsf{T}, \Delta T]^\mathsf{T}$. For the sake of readability arguments in both functions are omitted in the following.

The Lagrangian of problem (4) is given by:

$$\mathcal{L}(\mathbf{b}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = (n-1)\Delta T - \boldsymbol{\mu}^\mathsf{T}\mathbf{h} - \boldsymbol{\lambda}^\mathsf{T}\mathbf{g} \qquad (6)$$

$\boldsymbol{\mu} \in \mathbb{R}^{p(n-1)}$ and $\boldsymbol{\lambda} \in \mathbb{R}^{r(n-1)+1}$ are referred to as constraint multiplier vectors. A local solution $\mathbf{b}^*$ of (4) has to satisfy the first order necessary conditions, known as the *Karush-Kuhn-Tucker* (KKT) conditions. Assuming some constraint qualifications (CQ) hold at $\mathbf{b}^*$ (see below), multiplier vectors $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}^*$ exist which satisfy the following conditions for $(\mathbf{b}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ [21]:

$$\nabla_{\mathbf{b}}\mathcal{L}(\mathbf{b}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*) = 0, \qquad (7)$$

$$g_i(\mathbf{b}^*) \geq 0, \quad \lambda_i^* \geq 0 \qquad (8)$$

$$h_i(\mathbf{b}^*) = 0, \quad \mu_i^* h_i(\mathbf{b}^*) = 0, \quad \lambda_i^* g_i(\mathbf{b}^*) = 0 \qquad (9)$$

$g_i, h_i, \lambda_i, \mu_i$ denote the elements of $\mathbf{g}, \mathbf{h}, \boldsymbol{\lambda}, \boldsymbol{\mu}$ respectively. The *linear independence QC* (LICQ) is a common constraint qualification for nonconvex, nonlinear constraints, which holds if the set of active constraint gradients is linearly independent at $\mathbf{b}^*$ [21].

*Remark 1:* Constrained optimization algorithms usually do not cover strictly positive inequality constraints. The difference quotient in (2) is not defined for $\Delta T = 0$. Hence, $\Delta T > 0$ is satisfied implicitly by preserving the difference quotient instead of eliminating the fraction and in addition by starting initially with $\Delta T > 0$. The implemented line-search based on an exact merit function $\Phi(\cdot)$ (see Section II-C) ensures $\lim_{\Delta T \to 0+} \Phi(\cdot) = \infty$ and hence it rejects step lengths causing $\Delta T = 0$.

To formally confirm, that $\mathbf{b}^*$ actually constitutes a local minimizer of (4), sufficient conditions must hold. A second order sufficient condition requires the Hessian of the Lagrangian $\nabla_{\mathbf{bb}}\mathcal{L}(\mathbf{b}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ to be positive semidefinite at $(\mathbf{b}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ [21].

### C. Solution of the TEB Open-Loop Optimization Problem

Solving problem (4) requires constrained optimization algorithms that are well known in literature and for which many numerical software packages are available. For the implementation of a real-time capable MPC, common solvers either employ *interior-point methods* that exploit a particular problem structure exploitation, or *online-active-set methods* that facilitate hot-starting from previous solutions. Recent implementations especially for nonlinear MPC operate either in continuous or in discrete time domain [22]. Our approach utilizes a *sequential programming approach (SQP)* based on the line-search SQP algorithm described in [21]. The SQP
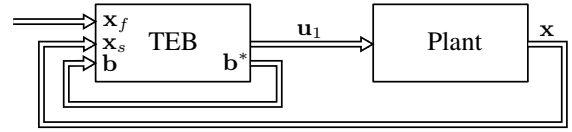


Fig. 1. Closed-loop TEB control

iteration problem (4) is approximated w.r.t. the objective function by a quadratic model and w.r.t. the constraints by a linear approximation. The resulting quadratic program is solved by *qpOases*, an homotopy based *online-active-set* implementation [23]. It allows hot-starts for subsequent SQP iterations. *qpOases* handles semi-definite Hessians that may result due to the linear objective function. TOMPC as well successfully uses *qpOases* for time-optimal MPC [13]. In order to execute a sufficient step along the gradient descent direction of problem (4) based on the solution of the quadratic approximation, a $l_1$ exact merit function $\Phi(\cdot)$ is minimized in the underlying backtracking line-search [21]. To speed up the optimization, a common BFGS quasi-newton approximation for an efficient estimate of the Hessian is applied. Note, that the TEB has been tested successfully with Matlab®'s constrained optimization algorithms.

### D. Closed-Loop TEB Control

This section describes the integration of the TEB open-loop optimization based trajectory planning into the closed-loop control. Figure 1 shows the block diagram of the underlying control architecture. The TEB algorithm identifies the (quasi) optimal trajectory $\mathbf{b}^*$ with respect to problem (4) in order to regulate the transition to the final state $\mathbf{x}_f$ depending on the current plant state $\mathbf{x}_k$. At each sampling interval, the imminent control action $\mathbf{u}_1$ of the planned sequence is applied as input to the plant. The state $\mathbf{x}_k$ is either directly measurable or estimated by a state observer. The optimized trajectory $\mathbf{b}^*$ is stored, respectively fed back, in order to hot-start subsequent optimizations at subsequent sampling intervals. The algorithm operates according to:

1: **procedure** TIMEDELASTICBAND($\mathbf{b}, \mathbf{x}_s, \mathbf{x}_f$)
2:     Initialize or update trajectory
3:     **for all** Iterations 1 to $I_{teb}$ **do**
4:         Adjust length $n$, resp. $d$ of the trajectory
5:         $\mathbf{b}^* \leftarrow$ SOLVENLP($\mathbf{b}$)            ▷ solve (4)
6:         Check feasibility
        **return** (sub-) optimal TEB $\mathbf{b}^*, \mathbf{u}_1^*$

The first sampling interval requires an initial guess of $\mathbf{b}$. A first analysis shows that for most time optimal control problems a linear interpolation $\mathbf{x}_k = \mathbf{x}_s + kn^{-1}(\mathbf{x}_f - \mathbf{x}_s)$ between start and final state with zero controls $\mathbf{u}_k = \mathbf{0}$ is sufficient for convergence to the global optimal solution. Obviously, this initialization strategy does not guarantee global convergence in the general case, since numerically solving (4) leads to a local minimizer only with respect to the initialization $\mathbf{b}$.

The previous solution $\mathbf{b}^*$ provides the initialization to the optimal control problem (4) at the next sampling interval.

The new initial trajectory is updated by replacing $\mathbf{x}_1$ and $\mathbf{x}_n$ by the most recent state estimate $\mathbf{x}$ and $\mathbf{x}_f$ respectively.

The initialization is followed by an outer optimization loop that iterates $I_{teb}$ times. At first the trajectory length in terms of the number of state and control samples $n$ is adapted depending on the current time increment $\Delta T$:

- If $\Delta T > \Delta T_{ref} + \Delta T_{hyst} \wedge n > n_{min}$, remove a state.
- If $\Delta T < \Delta T_{ref} - \Delta T_{hyst} \wedge n < n_{max}$, insert new state.

$\Delta T_{ref}$ denotes a time increment (usually related to the inherent sampling time), $\Delta T_{hyst}$ introduces a hysteresis in order to avoid oscillations when adding and removing states. $n_{min}$ and $n_{max}$ provides a lower and upper bound on the number of samples. At each instance either a new state is inserted or an obsolete state is removed from the TEB. The complete trajectory is re-sampled by linearly interpolating each state and control sequence with respect to time. Thus, the updated time interval is set to $\Delta T_{new} = \Delta T \frac{n}{n_{new}}$.

Adapting the length of the TEB has several advantages: The number of samples decreases as the current state converges towards the target state. Assuming fixed computational resources, the number of feasible iterations to improve the quality and accuracy of the trajectory increases as the computational demand for a single iteration with fewer samples decreases. In contrast, changes on the final state or disturbances may require a longer trajectory, respectively a finer discretization. As an example consider a mobile robot navigation scenario, in which a dynamic obstacle penetrates the original trajectory requiring an expansion of the path. In case the number of samples remains constant, the distance between two consecutive states increases with increasing path length, resulting in a non smooth trajectory. In the worst-case the obstacle slips through the trajectory due to a coarse resolution of samples.

The TEB algorithm repeatedly solves the underlying nonlinear program (4) by invoking an SQP solver $I_{sqp}$ times. The TEB algorithm leads to an elastic band in state- and control space, which convergence behavior is mainly influenced by the choice of $I_{teb}$ and $I_{sqp}$ (despite of specific solver configurations).

*E. Stability of Closed-Loop TEB Control*

Stability analyses of MPCs are often realized by use of *control Lyapunov functions* [24]. Consider finite horizon problems with final equality constraints ($\mathbf{x}_n = \mathbf{x}_f$) such that (4). The solution $\mathbf{b}_j^*$ is only valid if it satisfies the system dynamics, the final state equality constraint, each inequality constraint and results in a feasible control sequence. The principle of optimality demands that the objective function $V(\mathcal{B})$ is monotonically decreasing, such that the problem converges to $\mathbf{x}_f$ [24]. Figuratively, the TEB contracts between fixed $\mathbf{x}_s$ and $\mathbf{x}_f$ implying convergence due to the formulation of (4) that minimizes $\Delta T$.

Note, that the approach allows an asynchronous integration of control and planning rather than to operate with constant time slots for planning. In contrast to conventional MPC formulations it is possible to decouple the underlying rates of control and planning. The control sample rate is adapted to $\Delta T$ in order to decrease the number of recalculations and interventions as often as necessary. Considering the assumptions above, convergence is preserved which immediately follows by the convergence proofs of sample-based MPC [25] and MPC with asynchronous measurements and control [26].

In practice, numerical instabilities might emerge in the TEB as the actual state converges towards the target state $\mathbf{x}_f$. While converging to $\mathbf{x}_f$ and $n = n_{min}$, it follows $\Delta T \rightarrow 0$ which is undefined for the difference quotient in (2). Although according to section II-B $\Delta T > 0$ is ensured, too small time intervals $\Delta T$ are avoided as they cause numerical instabilities. This lower bound is accomplished by the inequality to $\Delta T - \epsilon > 0$ with $0 < \epsilon < \Delta T_{ref} - \Delta T_{hyst}$. Future work is concerned with the analysis of integrated switching strategies that utilize quadratic objective functions while converging towards the target $\mathbf{x}_f$.

## III. EXAMPLES

This section analyses the performance and potential of the TEB algorithm on the control of two nonlinear systems. The behavior of the closed loop control with TEB according to Fig. 1 is investigated in simulation. The simulations are done in Matlab (PC: $3,4$ GHz Intel i7 CPU) with a problem specific, but fixed sampling time $\Delta T_{ref}$. The SQP algorithm is compiled as a C++ program using code generation and the underlying quadratic program is solved by *qpOases* using a mex-C++-Interface. Note, the sparse structure of the optimization problem is not exploited within this work. However, dedicated solvers might reduce the computational effort of the TEB optimization significantly.

The TEB algorithm is compared to TOMPC that constitutes a state-of-the-art MPC extension to time-optimal point-to-point control (see description in Section I). TOMPC is reimplemented by the authors within the same framework by means of sharing the same SQP algorithm, initialization phase and data structures to allow a consistent and fair evaluation and comparison with the TEB formulation.

State trajectories are initialized as a straight line (see Section II-D). Unless stated otherwise, $I_{teb} = 2$, $I_{sqp} = 2$ and $I_{tompc} = 3$ are applied. The number of total SQP iterations $I_{teb} \cdot I_{sqp} = 4$ is chosen to be increased by one in comparison to TOMPC, since the TEB may adapt the length occasionally. It is $\Delta T_{hyst} = 0.1\Delta T_{ref}$ and $n_{min} = 3$.

*A. Van-der-Pol Oscillator*

The Van-der-Pol oscillator constitutes an oscillatory dynamic system with nonlinear damping. The system is described by $\ddot{x}(t) + (x^2(t) - 1)\dot{x}(t) + x(t) = u(t)$. The evolution of the state vector $\mathbf{x} = [x(t), \dot{x}(t)]^\mathsf{T}$ is governed by the state-space representation:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = [\dot{x}, \quad -(x^2 - 1)\dot{x} - x - u]^\mathsf{T} \qquad (10)$$

Discretizing the system with finite differences leads to:

$$\mathbf{h}_k(\mathbf{x}_{k+1}, \mathbf{x}_k, u_k, \Delta T) = \Delta T^{-1}(\mathbf{x}_k - \mathbf{x}_{k+1}) + \tilde{\mathbf{f}}(\mathbf{x}_k, u_k) \qquad (11)$$
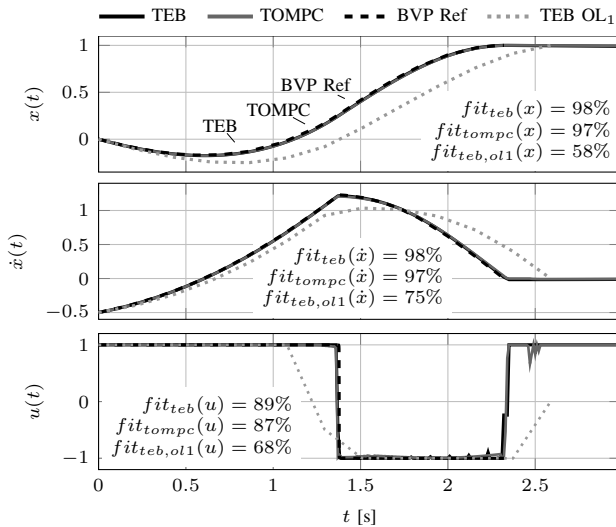
Fig. 2.  Closed-loop control of the Van-der-Pol oscillator



Fig. 3.  TEB trajectory optimization refinement

The control input is bounded to $-1 \leq u \leq 1$ that is captured by the inequalities $\mathbf{g}_k(u) = [u + 1, -u + 1]^\intercal \geq 0$.

As a reference for TEB and TOMPC, the actual time-optimal trajectory is obtained by solving a continuous *boundary-value-problem* (BVP) with Matlab's *bvp4c* algorithm. To create a suitable BVP, the control input sequence is modeled as a sign-function that is smoothed by an expansion of the state dimension. Time-optimality is accomplished using a time transformation. The interested reader is referred to [27] for more details.

The task is to transit the oscillator from the initial state $\mathbf{x}_s = [0, -0.5]^\intercal$ to the target state $\mathbf{x}_f = [1, 0]^\intercal$ in minimum time. The sampling time is set to $\Delta T_{ref} = 0.01$ s. Figure 2 shows the resulting state and control input trajectories of the system for the closed-loop TEB control, the closed-loop TOMPC control and the BVP reference. In addition the planned open-loop (OL) trajectory obtained after the first two TEB iterations for just $n = 10$ samples, is plotted. In order to compare the trajectories to the BVP reference, the quality of the fit between the executed trajectory $x(t)$ and the known optimal trajectory is given by $fit(x) = 100(1 - NRMSE(x))$. $NRMSE(x)$ denotes the normalized root mean square error. The reference argument is omitted. The results in Fig. 2 confirm that both TEB and TOMPC approximate the time-optimal control equally well.

The following analysis investigates the convergence behavior of the planned trajectory with respect to the number of optimization iterations. Initially, the trajectory length is set to $n = 10$, which is obviously a poor guess since it implies $n \cdot \Delta T_{ref} = 0.1$ s across the entire transition. The trajectory is optimized with $I_{teb} = I_{sqp} = 2$ which amounts to a CPU time of 4 ms. The resulting trajectory is shown in Fig. 2. Afterwards, the optimization is repeated multiple times to refine the current trajectory until $\Delta T = \Delta T_{ref}$. Figure 3 shows the fit value with respect to the number of invocations of the underlying optimization routine and the associated time for computation. The corresponding trajectory length is indicated by a second abscissa. Surprisingly,
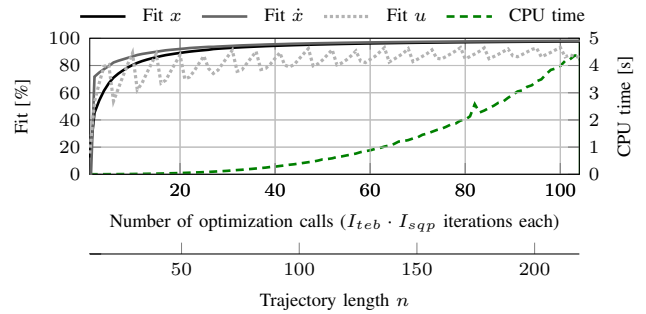
the state trajectories converge rapidly towards a fit value of $\approx 90\%$ and therefore approximate the time-optimal transition within a few iterations. Even the first optimization call with $n = 10$ results in feasible although suboptimal trajectories. The control input $u$ is optimal for the imminent control cycle 1 s, hence there is no degradation in control performance while the TEB is further refined across the next control cycles. Note, as a result the TEB transfers the idea of *real-time-iteration schemes* to time-optimal MPC.

In contrast, solving the same problem with $n_{init} = 10$, TOMPC requires 146 s to generate time-optimal trajectories. TOMPC increases $n$ by one each outer optimization loop until the problem is feasible. Due to a fixed $\Delta T = \Delta T_{ref}$ all trajectories (attempt to be) planned before are infeasible (below $n = 235$) and hence provide no meaningful controls. Considering Fig. 3, the TOMPC CPU time is proportional to the integral of the TEB CPU time starting from $n_{init}$. Applied to closed-loop control, TOMPC induces similar results in fit and CPU time if a suitable $n$ is determined in advance. However with changes in the target state or external disturbances the CPU time is significantly larger.

### B. Free-space Rocket System

The second example comprises the control of a free-space rocket system that is frequently mentioned in the MPC literature. The nonlinear systems equations are given by:

$$\dot{s}(t) = v(t) \tag{12}$$
$$\dot{v}(t) = (u(t) - 0.2v(t)^2)/m(t) \tag{13}$$
$$\dot{m}(t) = -0.01u(t)^2 \tag{14}$$

The mass $m$ is bounded to $-0.5 \leq m \leq 1.7$. $s$ denotes the distance covered, $v$ is the velocity. $-1.1 \leq u \leq 1.1$ denotes the system input. $\mathbf{h}_k$ and $\mathbf{g}_k$ are obtained according to the procedure in Section III-A. The sampling time is set to $\Delta T_{ref} = 0.1$.

For the following closed-loop control simulation (see Fig. 4), a sequence of two final states is given. The system starts at $\mathbf{x}_s = [x_s, v_s, m_s]^\intercal = [0, 0, 1]^\intercal$. The controller is only aware of the current final state (marked with dashed lines as reference), which switches after 10 s. In contrast to the original problem formulation (4), the final state $m$ is no longer constant to add a degree of freedom, since the final mass is a priori unknown. A step disturbance is applied to the system output $s(t = 6 \text{ s})$. The required computation
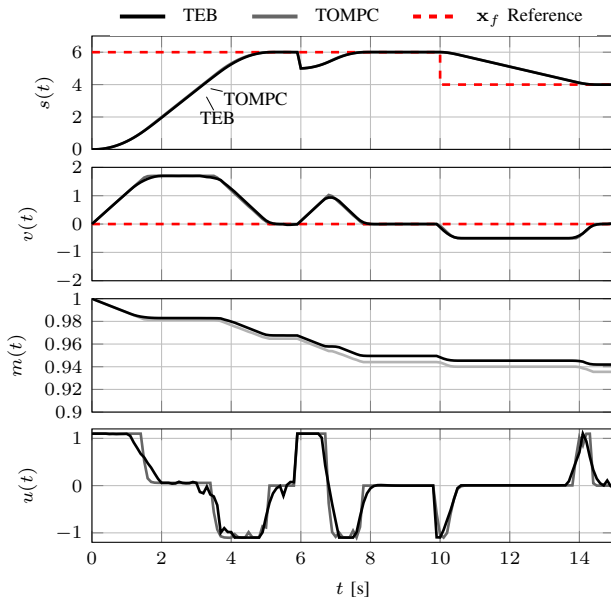
Fig. 4.  Closed-loop control of the free-space rocket system

TABLE I

COMPUTATION TIME OF ROCKET SYSTEM OPEN-LOOP CONTROL

| $n_{init}$ | TEB CPU Time | TOMPC CPU Time |
|---|---|---|
| 26 | **0.019 s** | 0.587 s |
| 51 | **0.123 s** | 0.148 s |
| 52 | 0.131 s | **0.111 s** |
| 53 | **0.138 s** | 0.186 s |
| 104 | **1.400 s** | 12.721 s |

time to generate a first feasible trajectory for varying $n_{init}$ is shown in Table I. The given sampling time is reached for $n_{init} = 52$.

## IV. CONCLUSIONS AND FUTURE WORK

The extension of TEBs to time-optimal MPC tasks performs well on two nonlinear systems presented in this paper. The results are evaluated using ground-truth time-optimal trajectories as a reference. In comparison to a state-of-the-art approach the TEB is able to generate a feasible approximation to time-optimal control already after a few iterations. The resulting state and control input trajectories are subsequently refined during closed-loop control.

Future work is concerned with exploiting the sparse structure of the TEB optimization problem in order to reduce the computational effort. Furthermore, switching strategies to quadratic cost functionals are considered to guarantee asymptotic stability upon convergence to the target state.

## 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] U. Piechottka and V. Hagenmeyer, "A discussion of the actual status of process control in theory and practice: a personal view from german process industry," *at – Automatisierungstechnik*, vol. 62, no. 2, 2014.

[3] 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.

[4] 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.

[5] Y. Wang and S. P. Boyd, "Fast model predictive control using online optimization," in *Proc. of the 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] M. Vukov, A. Domahidi, H. J. Ferreau, M. Morari, and M. Diehl, "Auto-generated algorithms for nonlinear model predictive control on long and on short horizons," in *IEEE Annual Conference on Decision and Control*, 2013, pp. 5113–5118.

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

[10] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones, "On real-time robust model predictive control," *Automatica*, vol. 50, no. 3, pp. 683–694, 2014.

[11] K. I. Kouramas, N. P. Faísca, C. Panos, and E. N. Pistikopoulos, "Explicit/multi-parametric model predictive control (mpc) of linear discrete-time systems by dynamic and multi-parametric programming," *Automatica*, vol. 47, no. 8, pp. 1638–1645, 2011.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] P. Grieder and M. Morari, "Complexity reduction of receding horizon control," in *IEEE International Conference on Decision and Control*, 2003, pp. 3179–3190.

[18] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," in *Int. Conf. on Robotics and Autom.*, 1993, pp. 802–807.

[19] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," in *European Conference on Mobile Robots*, 2013, pp. 138–143.

[20] M. Keller, F. Hoffmann, T. Bertram, C. Hass, and A. Seewald, "Planning of optimal collision avoidance trajectories with timed elastic bands," in *Proc. of the IFAC World Congress*, 2014, pp. 9822–9827.

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

[22] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear mpc in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.

[23] 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.

[24] 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.

[25] F. A. C. C. Fontes, L. Magni, and É. Gyurkovics, "Sampled-data model predictive control for nonlinear time-varying systems: Stability and robustness," in *Assessment and Future Directions of Nonlinear Model Predictive Control*.   Springer, 2007, vol. 358, pp. 115–129.

[26] P. Varutti and R. Findeisen, "Predictive control of nonlinear chemical processes under asynchronous measurements and controls," in *Advanced Control of Chemical Processes*, 2009, pp. 153–158.

[27] S. N. Avvakumov and Kiselev, Yu. N., "Boundary value problem for ordinary differential equations with applications to optimal control," in *World Multi-Conf. on Systemics, Cybernetics and Informatics*, 2004.