A Stabilizing Nonlinear Model Predictive Control Scheme for Time-optimal Point-to-point Motions

Robin Verschueren¹, Hans Joachim Ferreau², Alessandro Zanarini², Mehmet Mercangöz², Moritz Diehl¹

Abstract— This paper formulates a new scheme for time-optimal nonlinear model predictive control (TONMPC). In contrast to other TONMPC schemes for point-to-point motions found in the literature, it does not require a time transformation or a time scaling, which facilitates a stability proof. The practical applicability is shown with hardware-in-the-loop (HIL) experiments on an embedded platform using an illustrative control example, along with a comparison to one other time-optimal controller.

I. INTRODUCTION

Time-optimal control is one of the oldest optimal control problem classes, both in continuous time and discrete time, and an ample literature on this topic exists. Often the solution techniques are based on indirect methods (see e.g. [3]), and assume some properties on the optimal control a priori (for example, a bang-bang structure, i.e. the controls are at their limits almost always). For nonlinear systems, it is even more challenging to arrive at a time-optimal control law in closed form; for some special classes of nonlinear systems, this has been achieved, see e.g. [14] and the references therein.

For problems in discrete time, it was Kalman who first derived a time-optimal controller for linear time-invariant (LTI) systems (also sometimes called dead-beat control, for an excellent overview of such controllers, see [16]). Although time-optimal control in both continuous and discrete time are similar in nature, an important difference is that for discrete time systems, the time-optimal control is generally non-unique and not always bang-bang (see [7]).

In this paper, we focus on time-optimal control of discretized nonlinear dynamic systems, using direct methods. More specifically, we will tackle time-optimal point-to-point motions, in contrast to time-optimal motions along or in the neighborhood of a pre-specified path [21], [13], [22]. Point-to-point motions are a sizable application field, for example in robotic manipulators [10], wafer steppers [12] and cranes [23].

Some applications use a receding horizon technique, e.g. model predictive control (MPC). In [20], a two-level time-optimal MPC scheme for linear systems is presented and experimentally validated, where the upper level determines the optimal horizon length, and the lower level is a tracking MPC with fixed horizon length. One drawback of this method is that the horizon length potentially keeps changing from one problem to the next. In [25], a nonlinear MPC (NMPC) scheme for time-optimal control has been proposed, based on a varying time scaling (the horizon length is a decision variable). Close to the desired end point, the horizon is kept fixed and a standard regulator NMPC is used to keep the system there. A different but related idea is proposed in [18], where the continuous dynamics are scaled with a constant factor, such that the horizon length becomes fixed. This scaling factor then enters the objective function linearly, such that time-optimality is recovered. One disadvantage of this method, is that the underlying optimization problem becomes nonlinear, even for linear systems. Furthermore, a stability proof of an NMPC scheme using such time-scaled dynamics is not known to the authors.

The main focus of this paper is a novel NMPC scheme for time-optimal point-to-point motions. For this scheme, we make two theoretical contributions: we show that nominal stability of the end point holds (for simplicity of the exposition, the end point is assumed to be the origin), and we prove that it is time-optimal, if a certain tuning parameter is chosen sufficiently high. The scheme is based on an optimal control problem (OCP) formulation employing an exponential increase of the stage costs along the horizon, based on an idea as presented (but not further developed) in [19]. The stage costs are chosen to be the l₁-norm of the difference between the end point and the state at each stage. As such, exponentially increasing weights encourage the motion to arrive at the end point "as early as possible", thus recovering time-optimality. The resulting NMPC scheme inherits this property of time-optimality. The practical usefulness of the scheme is illustrated with numerical experiments.

The outline of the paper is as follows. In Section II, we state the problem we want to solve and present one state-of-the-art method. In Section III we describe the new time-optimal formulation and prove its time-optimality. Subsequently, we define an NMPC scheme based on this OCP and show a basic stability result in Section IV. Some numerical simulations are carried out, both on a desktop computer and on an embedded computing platform, and are presented in Section V. The paper is concluded in Section VI.

II. PROBLEM FORMULATION

We consider nonlinear system models of the following form:

\[
\frac{dx(t)}{dt} = f_c(x(t), u(t)), \quad \text{for } t \in [0, T],
\]

where \( t \) is the time, \( x(t) \in \mathbb{R}^n_x \) is the state vector and \( u(t) \in \mathbb{R}^n_u \) the vector of controls. Throughout this paper, we assume that \( f_c \) is continuous and \( f_c(0, 0) = 0 \). We define the continuous-time time-optimal control problem as follows:

\[
\text{minimize} \quad \int_0^T 1 \, dt \quad \quad \quad \quad \quad \quad (1a)
\]

subject to

\[
\dot{x}(t) = f_c(x(t), u(t)), \quad \text{for } t \in [0, T],
\]

\[
c(x(t), u(t)) \leq 0, \quad \text{for } t \in [0, T],
\]

\[
x(0) = \mathbf{x},
\]

\[
x(T) = \mathbf{y},
\]

\[
0 \leq T,
\]

\[
0 \leq T,
\]

\[
0 \leq T,
\]

Support by the EU via ITN-TEMPO (607957) and ITN-AWESCO (642682), by DFG via Research Unit FOR 2401, and by BMWi via the TEMPO Initial Training Network funded by the EU.

¹ J. Verschueren and M. Diehl are with the chair of Systems Control and Optimization, Department for Microsystems Engineering, University of Freiburg, Germany
robin.verschueren@imtek.uni-freiburg.de

² J. Ferrea, A. Zanarini and M. Mercangöz are with the ABB Corporate Research Centre, Baden-Dättwil, Switzerland
joachim.ferreau@ch.abb.com
where $\tau \in \mathbb{R}^{n_n}$ is the initial state, and the final state is 0, without loss of generality, as we can introduce a different final state by a simple state transformation. The inequality constraints are defined by $c : \mathbb{R}^{n_x+n_u} \to \mathbb{R}^{n_{ineq}}$. It is well-established in the literature that the solution of (1) for linear systems with linear constraints gives rise to bang-bang solutions [3], and uniqueness of the optimal solution, under some conditions, can be established by the uniqueness of a boundary-value problem (BVP). In the case of nonlinear systems, these properties cannot be guaranteed, but the solution is often still found to be of bang-bang type.

In this paper, we focus on a receding horizon formulation, more specifically time-optimal nonlinear model predictive control (TON-MPC), which consists in repeatedly solving discretized versions of OCP (1). We treat two different approaches; the first can be found in the literature [18], the second one (presented in Section III) is a new scheme.

**Time Scaling Approach**

One straightforward approach to solve OCP (1) is to introduce a time transformation, e.g. $\tau = t/T$, such that the continuous-time system becomes

$$\frac{dx(t)}{d\tau} = f_c(x(t), u(t)) \cdot T, \quad \text{for} \quad \tau \in [0, 1].$$

(2)

One advantage of doing so is that the horizon length $T$ becomes independent of the pseudo-time $\tau$ over which we integrate the continuous-time system. To discretize the continuous-time system, we choose a multiple shooting discretization [5]. Since $\tau$ is now the independent variable, we choose a shooting interval length of $\Delta \tau = 1/N$, with $N$ the number of shooting intervals. Discretizing OCP (1) becomes

minimize

$$T = \sum_{k=0}^{N-1} T_k$$

subject to

$$x_{k+1} = f_T(x_k, u_k, T_k), \quad k = 0, \ldots, N-1,$$

$$c(x_k, u_k) \leq 0, \quad k = 0, \ldots, N-1,$$

$$x_0 = \bar{x},$$

$$x_N = 0,$$

$$T_k = T_{k+1}, \quad k = 0, \ldots, N-2,$$

$$0 \leq T_k, \quad k = 0, \ldots, N-1,$$

(3a)

(3b)

(3c)

(3d)

(3e)

(3f)

(3g)

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$ are the vectors of states and controls, and we introduce decision variables $T_k$ on each shooting interval in order to preserve the problem structure. Function $f_T : \mathbb{R}^{n_x+n_u+1} \to \mathbb{R}^{n_x}$ is the discrete-time representation of the time-scaled dynamic system (2), e.g. obtained by numerical integration. The inequality constraints are denoted by $c : \mathbb{R}^{n_x+n_u} \to \mathbb{R}^{n_{ineq},k}$. We would like to point out that equality constraint (3f) is not strictly necessary, but it is advisable to include it, as the optimizer might exploit the time warping to “cut corners” as to accommodate the path constraints (3c).

In an NMPC setting, we repeatedly solve OCP (3), feeding back the optimal control $u^*_0$ to the system. However, we need to include one additional constraint, such that the sampling time $\Delta t$ of the real control system matches that of the first time interval in OCP (3), namely $T_0 = N \cdot \Delta t$.

Note that such a fixed first interval makes a possible proof of nominal stability for the NMPC scheme more intricate, in particular because the property of recursive feasibility is difficult to establish. Indeed, such a stability proof does not yet exist, to the authors’ knowledge.

III. TIME-OPTIMAL OCP WITH EXPONENTIAL WEIGHTING

The second formulation, by contrast, does not employ a time-scaling and uses the discrete-time dynamics

$$x_{k+1} = f_d(x_k, u_k), \quad k = 0, 1, \ldots,$$

obtained by e.g. numerically simulating the continuous-time dynamics $f_c(x(t), u(t))$ over one sampling interval $\Delta t$. We make the following standard assumptions on the system.

**Assumption 1:** We assume that $f_d$ is continuous, and that $f_d(0, 0) = 0$.

The new time-optimal scheme is based on the following discrete time minimum-time problem which is another discretization of continuous-time problem (1):

$$N^*(\bar{x}) = \min_{N,x_0,\ldots,x_N \in \mathbb{R}^{n_x}} \quad \min_{0 \leq u_0,\ldots,u_N \leq 1} \quad N$$

s.t.

$$x_0 = \bar{x},$$

$$x_{k+1} = f_d(x_k, u_k), \quad k = 0, \ldots, N-1,$$

$$c(x_k, u_k) \leq 0, \quad k = 0, \ldots, N-1,$$

$$x_N = 0,$$

(4)

where $N \in \mathbb{N}_0$ (the set of nonnegative integer numbers), and $x_k, u_k, \bar{x}$ as before.

If we found a solution such that the state arrives at the origin at stage $N^*$ (we will sometimes use $N^*$ as a shorthand for $N^*(\bar{x})$ in the following), we want to keep it there, such that we make the following assumption:

**Assumption 2:** For the inequality constraints $c(x, u)$ it holds that $c(0, 0) \leq 0$.

The following definition will prove to be useful in the subsequent discussion.

**Definition 1:** We define a time-optimal solution subject to discrete dynamical system $x_{k+1} = f_d(x_k, u_k)$ as any solution to (4) that brings the system from $\bar{x}$ to the origin in $N^*(\bar{x})$ steps, where $N^*(\bar{x})$ is the solution to (4). Furthermore, let $\mathcal{X}_N^*$ denote the set of states $\bar{x}$ such that the optimal value of (4) is smaller than or equal to $N^*(\bar{x})$.

The fact that the horizon length is not fixed in OCP (4) is cumbersome in an algorithmic setting, because the problem dimensions will change in each iteration of the solution method. By contrast, we introduce our time-optimal formulation with fixed horizon length $N$ (possibly much larger than $N^*$) as follows:

$$V^*_N(\bar{x}) = \min_{x_0,\ldots,x_N \in \mathbb{R}^{n_x}} \quad \min_{0 \leq u_0,\ldots,u_N \leq 1} \quad \sum_{k=0}^{N-1} \theta^k \|x_k\|_1$$

s.t.

$$x_0 = \bar{x},$$

$$x_{k+1} = f_d(x_k, u_k), \quad k = 0, \ldots, N-1,$$

$$c(x_k, u_k) \leq 0, \quad k = 0, \ldots, N-1,$$

$$x_N = 0,$$

(5a)

(5b)

(5c)

(5d)

(5e)

where $\theta \in \mathbb{R}$ is a fixed parameter. Note that we fix $x$ to zero at a later stage than $N^*$. With that regard, an interesting connection between problems (4) and (5) is stated in the following theorem.

**Theorem 1:** Assume OCP (4) is feasible and choose $N \geq N^*(\bar{x})$. There exists a number $\theta_1$ such that, for all $\theta \geq \theta_1$, the solution of (5) satisfies $x^*_N = 0$, i.e. the solution is time-optimal with respect to Definition 1.
Naturally, this discrete-time time-optimal solution is a mere approximation of the continuous-time problem (1) but the approximation gets better as the fixed sampling time \( \Delta t \) gets smaller and simultaneously \( N \) gets bigger. In order to prove Theorem 1, we will first present a result for which we consider the following

\[
\phi(x_0, \ldots, x_{N-1}) := \sum_{k=0}^{N-1} \theta^k \|x_k\|_1, \quad \text{and we relax the terminal constraint to be } x_{N^*} = 0, \text{ such that the OCP becomes:}
\]

\[
\begin{align*}
\text{minimize} & \quad \sum_{k=0}^{N-1} \theta^k \|x_k\|_1 \\
\text{subject to} & \quad x_0 = \pi, \\
& \quad x_{k+1} = f_k(x_k, u_k), k = 0, \ldots, N - 1, \\
& \quad x_{N^*} = 0.
\end{align*}
\]

with \( \epsilon \in \mathbb{R}^{n_x} \). The optimal solution depends parametrically on \( \epsilon \), and we remark that the dual of the 1-norm, see \([15]\). As such, we have that the theorem holds. For \( N > N^* \), for all \( \theta \geq \theta_1 \), the term \( \theta^N \|x_N\|_1 \) represents an exact penalty term for the constraint \( x_N = \epsilon = 0 \) in (6), by Lemma (1) and the fact that the \( \theta \)-norm is the dual of the 1-norm, see \([15]\). As such, we have that \( x_N = 0 \). For stages \( k = N^* + 1, \ldots, N \), the solution will stay at the origin, as it is feasible (by Assumption 2) and optimal (because of the 1-norm in the objective). This proves the theorem. 

\[ \blacksquare \]

IV. NMPC FORMULATION

The time-optimality of OCP (5) has been established in Theorem 1. Next, we will use this in a receding horizon fashion, i.e. we formulate the NMPC problem. The NMPC scheme consists of the following steps:

1. Estimate the state \( \pi \) at the current time \( t_k \).
2. Solve OCP (5) for \( u_0^* \).
3. Apply control \( u_N(\pi) := u_0^* \) to \( f_{\pi} \).
4. Proceed to the next time point \( t_{k+1} \).

Let \( X_N \) be the set of states \( \pi \) for which (5) has a solution. We make the following (standard) assumption, refer to e.g. \([17]\):

\[ \text{Assumption 4:} \text{ There exists a } \kappa_\infty \text{ function } \alpha(\cdot) \text{ such that } V_N(\pi) \leq \alpha(\pi(\pi)). \]

The stability of the NMPC scheme is established in the next theorem.

\[ \text{Theorem 2:} \text{ Take } \theta > 1. \text{ Then, the origin is asymptotically stable for the system } x_{\text{next}} = f_{\pi}(\pi, u_N(\pi)). \]

\[ \text{Proof:} \text{ We follow a standard Lyapunov argument, using the notation as in \([17]\). We need to show that the optimal value function in (5) satisfies:} \]

\[ V_N(\pi) \geq \alpha_1(\pi) \]

\[ V_N(\pi) \leq \alpha_2(\pi) \]

\[ V_N(\pi) \leq \alpha_3(\pi) \]

for \( \kappa_\infty \) functions \( \alpha_1(\cdot), \alpha_2(\cdot) \). The optimal cost decrease \((9c)\) is proved as follows: consider the optimal value function

\[ V_N(\pi, u^*) := V_N(\pi) = \sum_{k=0}^N \theta^k \|x_k\|_1, \]

with \( u^* := [u_0^*, \ldots, u_{N-1}^*] \). Going to the next time step, with feasible but suboptimal control sequence \( \tilde{u} := [u_1^*, \ldots, u_{N-1}^*] \),
We define the state vector \( \mathbf{x} \) and the control vector \( \mathbf{u} \) as:

\[
\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}
\]

We then run NMPC scheme (8), with parameters \( N = 50 \), \( \theta = 1.6 \), starting from three different points at \( t_0 = 0 \) s. The inequality constraints are

\[
c(x, u) = \begin{bmatrix} u - 10 \\ u - 10 \end{bmatrix} \leq 0.
\]

For each of the three scenarios, we also compute \( N^* \), the minimum time. The closed-loop trajectories are shown in Figure 1. Notice how all the state trajectories go to the origin (and not earlier) at \( k = N^* \) and remain there, which illustrates Corollary 1. Furthermore, it is interesting to look at the control trajectories: the closed-loop control lies against its limits \([-10, 10]\) except for some points, which usually holds for discrete-time time-optimal control (see [7]).

### B. Hanging Pendulum

A second system we apply our time-optimal NMPC scheme on is a hanging pendulum with varying length, that moves from start to end point, which are both equilibria for the system. The continuous-time equations of motion are

\[
\begin{align*}
\dot{x}_t &= v_t, & \dot{v}_t &= a_t, \\
\dot{x}_c &= v_c, & \dot{v}_c &= a_c, \\
\dot{\phi} &= \omega, & \\
\dot{\omega} &= \frac{-2\omega v_c + a_t \cos(\phi) + g \sin(\phi)}{x_c}.
\end{align*}
\]

with \( x_t[m], v_t[m/s], a_t[m/s^2] \) the horizontal displacement, velocity and acceleration, respectively. We assume that we can control the acceleration directly, as in [4]. Same holds for the cable length \( x_c[m] \), with corresponding cable (un)rolling velocity \( v_c[m/s] \) and acceleration \( a_c[m/s^2] \). The angle that the pendulum makes with the vertical is denoted by \( \phi[\text{rad}] \), its angular velocity is \( \omega[\text{rad/s}] \). We define the state vector \( \mathbf{x} = [x_t, v_t, x_c, v_c, \phi, \omega] \) and controls

\[
u = [a_t, a_c] \] .

The height of the pendulum is \( H = 0.5 \) m, and the following bounds apply on states and controls:

\[
\begin{align*}
-0.8 \text{ m/s} &\leq v_t \leq 0.8 \text{ m/s} \\
0 \text{ m} &\leq x_c \leq 0.5 \text{ m} \\
-0.5 \text{ m/s} &\leq v_c \leq 0.5 \text{ m/s} \\
-1 \text{ m/s} &\leq a_t \leq 1 \text{ m/s}^2 \\
-1 \text{ m/s} &\leq a_c \leq 1 \text{ m/s}^2.
\end{align*}
\]

Furthermore, we introduce a static obstacle that the pendulum

For all the three scenarios, we also compute \( N^* \), the minimum time. The closed-loop trajectories are shown in Figure 1. Notice how all the state trajectories go to the origin (and not earlier) at \( k = N^* \) and remain there, which illustrates Corollary 1. Furthermore, it is interesting to look at the control trajectories: the closed-loop control lies against its limits \([-10, 10]\) except for some points, which usually holds for discrete-time time-optimal control (see [7]).
should avoid touching, of the form
\[
H - x_c \cos(\varphi) \geq \
\frac{0.4}{3} \left( \arctan(200 * (x_t + x_c \sin(\varphi) - 0.5)) - \arctan(200 * (x_t + x_c \sin(\varphi) - 0.7)) \right).
\] (13)

1) Optimal Control Problem: We compare optimal control formulations (3) and (5), with the following parameters: \( N = 25 \), starting point is \( \pi = [0, 0, 0.4, 0, 0, 0] \), end point \( x_{end} = [1, 0, 0.4, 0, 0, 0] \), and for the exponential formulation we take \( \theta = 1.5 \), \( \Delta t = 0.1 \text{s} \). Constraints (12) apply to both formulations.

The results can be seen in Figure 2a. We note that both trajectories are almost equal, as they should be (the difference arises from the general non-uniqueness of time-optimal control in discrete time). The optimal time for the time scaling method is \( T^* = 2.219 \text{s} \), which compares with the optimal time of the exponential method, \( N' \cdot \Delta t = 2.3 \text{s} \); the difference in optimal value for the time is smaller than one sampling time interval. Also the optimal state and control trajectories look very similar, the only difference being the horizontal acceleration \( a_t \); the exponential weighting method decides to swing more when going over the obstacle. Although the trajectories could be brought closer to each other by additional control penalties, we show this result to illustrate the non-uniqueness in the time-optimal controller.

![Optimal control trajectories](image)

(a) Optimal control trajectories in the \( X \) \( Y \) plane, with obstacle as in (13).

![Optimal state and control trajectories](image)

(b) Optimal state and control trajectories.

2) Hardware in the loop simulations: As a last result, we show hardware in the loop (HIL) simulations of an efficient NMPC implementation. For this, we use the open source ACADO Code

Fig. 3: CPU times and number of QP iterations against number of steps of the NMPC.

Generation Toolkit [11]. This tool automatically produces efficient C90 code implementing the real-time iteration (RTI) scheme [8], when given a high-level description of the problem. In the inner iterations, we use qpOASES [9] as QP solver. We then run this code on ABB’s high performance controller AC 800PEC, which is typically used for time- and safety-critical applications in the power electronics domain. It features a dual-core CPU running at a clock speed of up to 1200 MHz as well as a field-programmable gate array. In our simulations, both the RTI scheme as well as the simulation model are running on a single CPU core at a clock speed of 800 MHz. Simulation results are obtained via Ethernet using a dedicated communication protocol.

For the HIL-simulation, we show one run on the pendulum without constraint (13). We show the CPU times of the closed loop controller in Figure 3. As we can see, in the beginning there are a lot of active set changes in the active set QP solver (right hand axis), such that the CPU time taken for one RTI step are relatively high (more than 700 ms). However, there is a very steep drop afterwards. Ultimately, the number of QP iterations goes to zero as the pendulum arrives at its end point. It should be noted that the number of active set changes (and thus the CPU time spent in the QP solver) at the beginning of the run could be drastically reduced by first executing a few “warm-starting” NMPC runs, where the initial value is not updated.

VI. CONCLUSION

We have presented a new NMPC scheme for time-optimal control for point-to-point motions. We show two corresponding theoretical results: a general stability proof of the end point (origin) is given, and time-optimality is established. The practical usefulness of this method is shown with HIL-simulations on an embedded platform.

Subject of further research is a generalization of the stability result, as well as development of a tailored code to push down computation times, as well as an application to an experimental setup.

REFERENCES


