Johann Radon Institute for
Computational and Applied Mathematics
Austrian Academy of Sciences (ÖAW)

Smoothened Quasi-Time-Optimal Control for the Torsional Torque in a Two-Mass-System

Esteban Fuentes, Dante Kalise, Ralph Kennel

RICAM-Report 2015-01

Smoothened Quasi-Time-Optimal Control for the Torsional Torque in a Two-Mass-System

Esteban Fuentes, Dante Kalise and Ralph Kennel

January 29, 2015

Abstract

In this work, a quasi-time-optimal, nonlinear state-feedback control for the torsional torque of a two-mass system (TMS) is presented. The control scheme considers constraints in the stator voltage and currents of the driving machine: a permanent magnet synchronous motor (PMSM). Its design is based on results of the more general framework of optimal control and its implementation constitutes an ad hoc adaption of a numerical method, formulated to solve a more general class of time-optimal control problems. This method leverages the assumption of a bang-bang structure for the actuation and the proposed algorithm exploits the geometrical properties displayed by the system state trajectories in the state space, when steered under this regime, to render its implementation feasible. Once the state reaches a region near its target, the control task is taken over by a classical linear quadratic regulator, to avoid spurious chattering.

The algorithm outputs a reference for the electrical torque to be developed by the PMSM, which is controlled with a finite-set model predictive torque controller, modified to make use of a voltage modulation scheme.

The performance of the proposed controller is validated with experimental results, which consider an external proportional-integral controller for the load speed, tunned for relatively high bandwidth.

1 Introduction

This article deals with the problem of devising a feasible algorithm to time-optimally control the torsional torque in a TMS, driven by a PMSM (see Fig. 1), accounting for constraints in the stator voltage, generated by a two-level voltage source inverter (VSI), and currents. The goal is to speed the closed loop dynamics, to track step-references, up to its physical limits, avoiding the use of off-line computations and tunning.

The TMS is a convenient model for many mechanical systems ranging from wind mills to laser positioning systems, in general: any drive, where mechanical design or control specifications weaken the assumption of an infinitely stiff shaft. This system poses a challenge to control systems, since exciting it at its resonant frequency could destroy the shaft. Its time-optimal control is even more complex, since the whole state can not be accelerated in an arbitrary direction for every point in the state space [1].

Different approaches have been proposed to control the two-mass system, ranging from classical techniques for non-linear systems [2], sliding mode con-

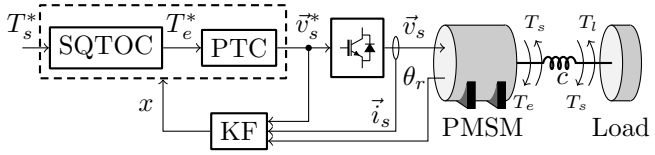

Figure 1: Diagram of the drive. The load is driven by a PMSM through a flexible shaft, with stiffness constant c. The PMSM is driven by a two-level VSI. The control system consists of the Smoothened Quasi-Time-Optimal Controller (SQTOC), a predictive torque controller (PTC), modified to make use of a modulation scheme, and a Kalman Filter (KF). The latter is used to reconstruct the state of the system x and feed it to the controller

trollers [3, 4], adaptive control [5, 6] and fuzzy control [7]. In [8] a comparison between classic state-feedback controllers and model predictive control is presented and in [9, 10] the authors of this work presented naive approaches using *finite-set model predictive control*. In all these works the two main recurring issues are tunning and taking constraints into account. The first one is usually tackled using frequency-domain criteria. The second issue is either ignored or dodged with patches (anti-windup). The approach presented in this work takes constraints explicitly into account and, for the most part, tunning is not necessary: the directing criteria is time-optimality. The latter does not hold, however, for fine control around the steady state, where a classic Linear Quadratic Regulator (LQR) takes over.

The design of the proposed controller, the SQTOC, is based on a convenient approximate model of the whole system dynamics: the abstract TMS, and a computational scheme: the switching time parametrization method (STPM), which can be used to solve a general class of time-optimal control problems, formulated as boundary value problems, assuming a bang-bang structure for the actuation. This method is implemented using a general optimization algorithm (COBYLA) to study the behaviour of the abstract TMS, when controlled to achieve time-optimal behaviour, but is not practical to be used in a real drive: the solution is valid only in a continuous-time regime and furthermore, the iterative nature of the optimization algorithm requires heavy computation. The SQTOC solves this by taking advantage of the geometrical properties of the trajectories developed by the abstract TMS, when time-optimally controlled. The iterations in the STPM are replaced by a prescribed sequence for the actuation, which is given by the position of the state in the state-space, with respect to the trajectory the system should develop when reaching its target, and a finite-set of long range predictions (for the application of this idea to approach the harmonic distortion problem in torque controllers refer to [11], for its application in the development of a position controller refer to [12]).

The resulting control scheme (see Fig. 1) takes the form of a state feedback (it has no internal states), with a cascaded structure: the PTC as inner controller and the SQTOC as outer controller. The workings of both controllers are, nevertheless, closely interwoven: the outer controller does consider the fundamental properties of the whole system, as well as the effect of the inner controller, whereas the latter allows to make assumptions that simplify the way the outer controller takes into account the dynamics of the *inner loop*.

The design of the proposed controller is based on the model of the system and the directing criteria is time-optimality, hence, no design parameters are introduced, except for the definition of a region in the state space where an LQR takes over to avoid chattering: in the same fashion as a sliding mode controller, the reference for T_e given by the SQTOC is always one of its extreme values $\pm\hat{T}_e$, therefore, when the system is near its steady state the reference for T_e should be smoother.

This paper is organized as follows: section 2 presents the model of the drive and a more abstract model, which is used as foundation for the proposed controller. In section 3 general ideas regarding time-optimal control are outlined and the time-optimal behavior for the abstract model of the TMS is obtained using a continuous time optimization scheme, assuming *bang-bang* structure. The workings of the proposed control method are described in section 4. Experimental results and general conclusions are presented in sections 5 and 6.

2 System Model

2.1 Drive Model

The dynamics of the drive, including the PMSM drive and the TMS, are described by the following dynamical system [13, 14]:

$$\frac{d}{dt}\begin{bmatrix} \vec{i}_s \\ \omega_m \\ \theta_m \\ T_s \\ \omega_l \end{bmatrix} = \begin{bmatrix} L_s^{-1}\left(\vec{v}_s - r_s\vec{i}_s - \dfrac{\partial L_s}{\partial\theta_m}\omega_m\vec{i}_s - J\omega_m\vec{\psi}\right) \\ \frac{p}{J_m}\left(T_e - T_s\right) \\ \omega_m \\ \frac{c}{p}\left(\omega_m - \omega_l\right) \\ \frac{p}{J_l}\left(T_s - T_l\right) \end{bmatrix} \tag{1}$$

where $\vec{i}_s = [i_\alpha\ i_\beta]^T$ and $\vec{v}_s = [v_\alpha\ v_\beta]^T$ are the stator current and voltage, ω_m and θ_m are the speed and position of the rotor of the motor. The last two states represent the torsional torque T_s and the load speed ω_l. For convenience the latter is expressed in electrical radians per second. The self-inductance $L_s = T\begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix}T^{-1}$ depends on the position of the rotor magnet θ_m, with $T = \begin{bmatrix} \cos(\theta_m) & -\sin(\theta_m) \\ \sin(\theta_m) & \cos(\theta_m) \end{bmatrix}$, while L_d and L_q are the equivalent inductances in the direction of the magnet and the quadrature direction respectively. The parameter r_s is the stator resistance, $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ and $\vec{\psi} = T\begin{bmatrix} \psi_m \\ 0 \end{bmatrix}$ is the flux produced by the rotor magnet of magnitude ψ_m.

The torque generated by the PMSM is given by:

$$T_e = \frac{3}{2}p(\psi_m i_q + (L_d - L_q)i_d i_q), \tag{2}$$

where p is the pole pair number and $\begin{bmatrix} i_d \\ i_q \end{bmatrix} = T^{-1}\vec{i}_s$.

The torsional torque is given by:

$$T_s = \frac{c}{p}\left(\theta_m - \theta_l\right), \tag{3}$$

where θ_l is the position of the load and c is the spring constant of the shaft.

The coupling between the mechanical variables gives rise to resonant behaviour with resonant angular frequency:

$$\omega_c = \sqrt{c\frac{J_m + J_l}{J_m J_l}}. \tag{4}$$

The input of the system $\vec{v}_s$, is synthesised using a two-level voltage source inverter.

Actuator - voltage source inverter

In the two-level VSI, $\vec{v}_s$ relates to the duty cycle applied on each of its three legs d_a, d_b and d_c, and to the dc-link v_{dc}, via:

$$\vec{v}_s = \frac{2}{3} v_{dc} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} d_a \\ d_b \\ d_c \end{bmatrix} \tag{5}$$

with $d_{\{a,b,c\}} \in [0, 1]$, thus generating an hexagon in the $\alpha\beta$ plane.

In the so-called *finite-set model predictive controllers* (FS-MPC) [15] only the vertices of the hexagon, together with $\vec{v}_s = \vec{0}$, are taken as the control set. Consequently, the overall performance is very coarse if the sampling period is relatively long. This is critical in the application at hand, since the electrical and mechanical time constants are of comparable magnitudes, hence relatively high precision is needed to achieve an acceptable performance. In order to avoid the hard trade-off between high-switching frequency, control precision and computational constraints, posed by the FS-MPC method, the proposed controller makes use of a PWM scheme to synthesize arbitrary voltage vectors inside the aforementioned hexagon. The core ideas from FS-MPC are still employed to simplify the solution of the optimal control problem, reducing its search space.

2.2 Abstract Description

In order to focus the analysis on the dynamics that are relevant to the desired behaviour, the above presented model is distilled through a set of assumptions, which we discuss in this section.

As stated before, both the stator voltage and currents can be described with two components ($\alpha\beta$ or dq). Nevertheless, only the current component that has an influence on the electrical torque is relevant to the higher order dynamics, namely, the mechanical variables. By neglecting the non-linearities and the resistive losses in the stator currents equations, and the reluctance component of the torque in (2), the relevant information of the first two equations of the system model (1) can be summarized in one simple equation:

$$\frac{dT_e}{dt} = \frac{3p\psi_m}{2L_q} v_q \tag{6}$$

To assume that the dynamics of T_e are described by (6) is very bold. However, it is useful to approximate the system behaviour during the transients and the errors introduced by this assumption are corrected by the use of an inner torque controller that takes all the aforementioned effects into account and regulates the

stator currents so as to obtain maximum torque per ampere operation (MTPA) [16].

With this, the dynamics of T_e (eq. (6)), henceforth x_0, can be synthesized the equation:

$$\frac{dx_0}{dt} = \frac{u}{\tau_0}. \tag{7}$$

The state x_0 is relevant for the control design insofar it stores the information regarding the satisfaction of the constraint on the stator currents, i.e., the constraint on T_e.

The next assumption is that the actuation belongs to a finite set:

$$u \in \{-\hat{u}, 0, \hat{u}\}, \tag{8}$$

with $\hat{u} = \frac{\sqrt{3}}{3} v_{dc}$, the maximum amplitude the two-level VSI can generate, when synthesising perfectly sinusoidal voltages. This assumption is leveraged to reduce the trajectories the state could follow to a manageable set, i.e. shrink the search space for the optimal actuation. This assumption has physical and theoretical meanings: on the one hand the actuator (the VSI) has a constrained nature, and on the other, the actuation synthesized by time-optimal controllers adopts a *bang-bang* form (this is discussed in section 3). This assumption stops serving its purpose near the switching curve and the steady state, where restricting the actuation to its limit values will produce chattering, specially in a discrete-time regime. In order to avoid this, when the predicted state falls near the Γ curve (this is introduced latter, but is related to the switching curve or surface, through long-term predictions), the reference for T_e is synthesized using a linear combination of the values predicted using the extreme values of the actuation and, near the steady state, the control is passed over to a classical LQR, which produces torque references in the continuous range $[-\hat{T}_e, \hat{T}_e]$.

Finally, considering the following change of variables

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} T_e \\ k_1 \left(T_e - \sigma T_s\right) \\ \omega_m - \omega_l \\ T_s \end{bmatrix}, \qquad d = T_l, \tag{9}$$

with

$$\sigma = \left(1 + \frac{J_m}{J_l}\right) \quad \text{and} \quad k_1 = \frac{p}{J_m \omega_c}, \tag{10}$$

an abstract model for the drive, including the mechanical dynamics can be written:

$$\frac{dx}{dt} = \begin{pmatrix} {}^{u}/_{\tau_0} \\ -\omega_c x_2 + k_u u \\ \omega_c x_1 + k_d d \\ {}^{x_2}/_{\tau_3} \end{pmatrix}, \tag{11}$$

where

$$\tau_0 = \frac{2L_q}{3p\psi_m}, \qquad k_u = \frac{3p^2\psi_m}{2L_q J_m \omega_c}, \tag{12}$$

$$\tau_3 = \frac{p}{c}, \qquad k_d = \frac{p}{J_l}. \tag{13}$$

Equations for states x_1 and x_2 correspond to the description of an harmonic oscillator with two sources, u and d, and angular frequency ω_c. As it will be discussed in the following section, this model shall be convenient to understand, formulate and solve the minimum-time control problem.

The *total energy* in this harmonic oscillator is related to the distance between the state coordinates in the x_1x_2 plane and the steady state point. Component x_1 pertains to the potential energy stored in the *shaft spring*, whereas component x_2 pertains to the kinetic energy in the motor and load inertiae. In the following sections the words *expansion* and *contraction* will be used in relation to the *kinetic energy* of the oscillator, meaning that the state is driven *away from* or *towards* the steady state point in the x_2 axis, respectively.

2.2.1 Steady State

The steady state and the control target for system (11) are reached at point

$$x^{ss} = \begin{bmatrix} \sigma x_3^* - (\sigma - 1)d \\ -\frac{k_d}{\omega_c} d \\ 0 \\ x_3^* \end{bmatrix}, \tag{14}$$

where x_3^* is the torsional torque reference. In terms of the drive model this means that in steady state

$$T_e^{ss} = \left(1 + \frac{J_m}{J_l}\right) T_s^* - \frac{J_m}{J_l} T_l, \tag{15}$$

this is, in steady state T_e should be proportional to the reference T_s^* and also compensate for the load torque T_l.

2.2.2 Continuous time model

The step response of the system described by eq. (11) can be solved analytically assuming

$$u(t) = \hat{u}\mu(t) \qquad \text{and} \qquad d(t) = \hat{d}\mu(t) \tag{16}$$

and an i.c. $x(0)$, yielding to

$$x(t) = S(t, x(0), \hat{d}, \hat{u})\,, \tag{17}$$

with

$$S(t, x(0), \hat{d}, \hat{u}) = \begin{pmatrix} \frac{\hat{u}}{\tau_0} t + x_0(0) \\ A\cos(\omega_c t) + B\sin(\omega_c t) - \frac{k_d}{\omega_c}\hat{d} \\ A\sin(\omega_c t) - B\cos(\omega_c t) + \frac{k_u}{\omega_c}\hat{u} \\ \frac{1}{\sigma}\left(x_0(t) - \frac{1}{k_1} x_1(t)\right) \end{pmatrix}, \tag{18}$$

where

$$A = \frac{k_d}{\omega_c}\hat{d} + x_1(0) \qquad \text{and} \qquad B = \frac{k_u}{\omega_c}\hat{u} - x_2(0)\,. \tag{19}$$

In the x_1x_2 plane, the trajectories developed by $S(\cdot)$ follow circular paths with center

$$(x_1, x_2) = \left(-\frac{k_d}{\omega_c}\hat{d}, \frac{k_u}{\omega_c}\hat{u} \right) \tag{20}$$

and angular frequency ω_c. This property is later exploited to invert part of the minimum time problem in the x_1x_2 plane: the time required to drive the state between two arbitrary points with $u = \pm\hat{u}$ corresponds to the angle subtended by the arc between the two points from the center of the corresponding circle. This model will also be useful to calculate the Γ curve (see sec. 4.1), which is the trajectory that the state should approach and follow in order to reach the target.

3 Time-Optimal Control

In this section, we review some classical results concerning the existence and synthesis of minimum time controllers which are relevant for the proposed design. Let us consider a continuous time dynamical system given by

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = y, \tag{21}$$

where $x(\cdot) \in X \subseteq \mathbb{R}^n$ denotes the state of the system and $u(\cdot) \in U \subseteq \mathbb{R}^m$ is an admissible control signal. Our control goal is to steer the system to a target $\mathcal{T} \subset X$ as for instance, to a ball around the origin. If for a control signal $u(t)$ we define the arrival time from y to the target as

$$t_y(u) = \inf_s \{ s \in \mathbb{R}^+ : x(s; u(s)) \in \mathcal{T} \}, \tag{22}$$

the value function of the minimum time problem is given by

$$T(y) := \inf_{u(\cdot)\in U} t_y(u). \tag{23}$$

To obtain the optimal controller two approaches are possible. On the one hand, it is possible to derive optimality conditions characterizing an open-loop controller via the application of the Pontryagin Maximum Principle [17]. Such an approach is relatively simple to implement, and can yield accurate trajectories at low cost. A different solution method relies on the application of the Dynamic Programming Principle (DPP) [18], and characterizes the value function as the viscosity solution of the following Hamilton-Jacobi equation over X,

$$\sup_{u\in U} \{ -\nabla T(x) \cdot f(x, u) \} = 1. \tag{24}$$

Since the value function is obtained for the whole state space (which can be computationally costly for high-dimensional dynamics), the optimal controller is expressed in feedback form

$$u^*(x) = \arg\min_{\pi\in U} \{ \nabla T(x) \cdot f(x, \pi) \}, \tag{25}$$

which can be implemented for online control once the value function has been computed. However, since an approximation of the state space must be introduced in order to solve eq. (24), the value function is defined over a finite grid

of points, producing grid-dependent trajectories, which can exhibit an spurious chattering along the switching curves. Therefore, it is relevant to achieve a balance between the robustness of a feedback control and an accurate approximation of the switching structure of the system in order to yield adequate trajectories.

It is well-known that minimum time optimal control often leads to bang-bang type of controllers. More precisely, the classical results asserts that for a linear system with real eigenvalues, the minimum time controller will be bang-bang with at most $n-1$ switchings, where n is the dimension of the state space; there are different extensions of this result for the nonlinear case, but no general assertion can be made regarding the number of switchings in the presence of constraints in the state. Assuming that this number is finite, the computation of the optimal controller is reduced to the identification of the switching times. A simple solution method is then given in the scalar control case [19], by prescribing a switching sequence

$$\mathbf{u} = \{-\hat{u}, \hat{u}, -\hat{u}, \ldots, \hat{u}\}\,, \tag{26}$$

of size $m \geq n$, with corresponding times

$$\mathbf{\Delta t} = \{\Delta t_1, \ldots, \Delta t_m\}\,, \tag{27}$$

As a result, the minimum time problem can be cast as

$$\min_{\mathbf{\Delta t}} T := \sum_{i=1}^{m} \Delta t_i \tag{28}$$

subject to

$$\dot{x} = f(x, \mathbf{u})\,, \tag{29}$$
$$x(0) = y\,, \tag{30}$$
$$x(T) \in \mathcal{T}\,, \tag{31}$$
$$\Delta t_i \geq 0\,, \forall i\,. \tag{32}$$

If a bang-bang structure is assumed, this idea can be computationally implemented with a redundant amount of switchings, which shall be shrunk to zero if the optimal solution is achieved. In the following this scheme will be referred to as *switching time parametrization method* (STPM).

Such a scheme was implemented to investigate the properties of the state trajectories under a time-optimal controller and to hold the solutions as reference. Broadly speaking, the scheme works as a shooting method: it solves the state equation for different variations in the switching times of the actuation until the target is reached in minimum-time. Results using this approach to control the system in (11) are presented in Fig. 2.

Time optimal behaviour of the two-mass-system

In the context of the tow-mass-system, the control goal *to drive T_s towards an arbitrary reference in minimum time*, can be paraphrased as *to store an arbitrary amount of energy in the shaft spring in minimum time*; this perspective can be helpful to understand the systems desired behavior. First, the rotor of the

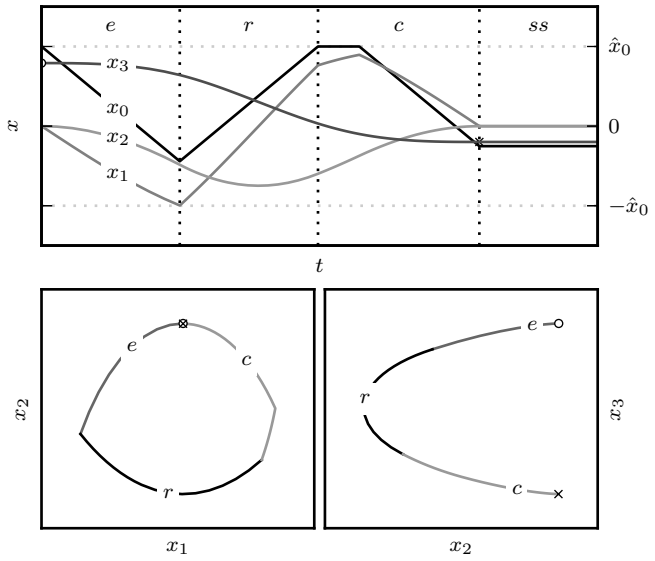

Figure 2: Time optimal trajectory for system (11) for an arbitrary i.c.. The plot on the top presents the system state in time. The plots in the bottom present the system state trajectories in the x_1x_2 and x_2x_3 planes, respectively. The control constraints are only represented in the first plot ($\hat{x}_0$). The i.c. and target for state x_3 are marked with the symbols $\bigcirc$ and $\times$. e, r, c and ss stand for *expansion*, *rollback*, *contraction* and steady state.

PMSM should be accelerated, so energy is transferred into kinetic energy. As the gap between θ_m and θ_l grows, energy is transferred to the *shaft spring*, where it takes the form of potential energy. which is then transformed back to kinetic energy as the load accelerates. At some point the kinetic energy in the rotor of the PMSM is more than enough to drive the potential energy in the rotor spring towards its desired level. Then, the surplus should be drawn by decelerating the rotor of the PMSM down to the point where the speeds and accelerations of both the rotor of the PMSM and the load match.

Fig. 2 presents plots of the trajectory developed by the state of system (11), when controlled to achieve this behavior, against time and in the x_1x_2 and x_2x_3 planes. Labels e, r and c account for three stages characterizing this trajectory. In terms of the model presented in 2.2 and the description in the last paragraph, they refer to stages of expansion (e) and contraction (c) of the kinetic energy in the harmonic oscillator x_1x_2, whereas r constitutes a rollback to a point in the state space where the contraction of the kinetic energy can be achieved as fast as the control constraints allow. For the development of the proposed method it is assumed that the system state must follow this pattern. This allows to reduce the optimal control problem to the evaluation of predictions of the system state, using only the set of sequences or configurations of the actuation that make the state follow this pattern.

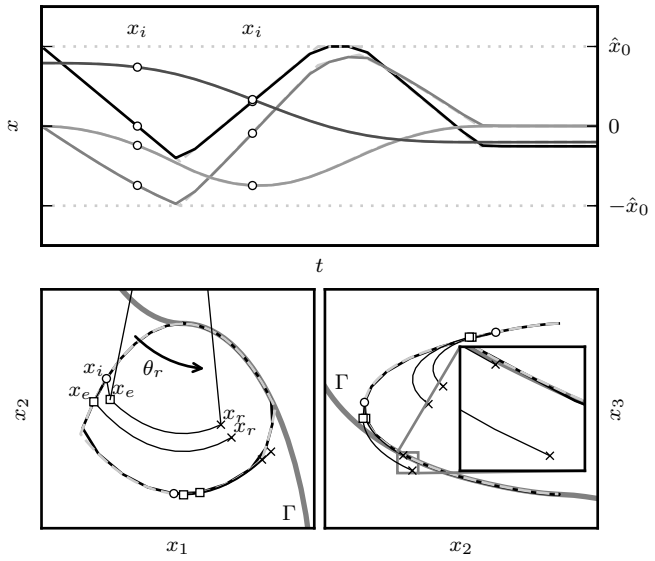

Figure 3: Approximation using the SQTOC method. x_i represents the i.c. for the method. x_e ($\square$) is the prediction for the *expansion* stage. Two predictions are shown: using $u = u_e$ (see eq. (33)) and $u = 0$. Points x_r ($\times$) pertain to the outcome of the long range predictions, with $u = -u_e$, representing the *rollback* stage. The time frame for this prediction is calculated from θ_r. The rays subtending θ_r intersect at $\left(-\frac{k_d}{\omega_c}d, -\frac{k_u}{\omega_c}u_e\right)$. The sign for u_e is calculated from the position of x_i with respect to the Γ curve (see sec. 4.1) in the x_2x_3 plane. The control decision is taken upon the position of x_r, with respect to the Γ curve in the x_2x_3 plane. In first case both predictions, with $u = u_e$ and $u = 0$ stay on the same side as the i.c. with respect to Γ. In this case the controller decides to go on with the expansion stage and applies $u = u_e$. In the second case (in the zoom box), the prediction with $u = u_e$ crosses Γ, whereas the prediction with $u = 0$ does not. In this case $u = 0$ is applied.

4 Smoothened Quasi-Time-Optimal Control

The design of the proposed control method is based in two main ideas:

- The *finite set assumption*. We assume that the value the actuation can take at any time belongs to a finite set and changes in the actuation can only occur at the sampling instants, which is a common setting for discrete-time control systems. The search space for the optimal control is then drastically reduced and, it can then be solved by direct evaluation of predictions of the system state using the values in this *finite set*. The actuation sequences for these predictions are configured so that the state follows the pattern described in the last section.

- Exploitation of the geometrical properties displayed by the state trajectory, when steered with maximum actuation.

The following section describes a key element of the proposed control method: the Γ curve. After this, the control algorithm is explained alongside technical implementation details.

4.1 Γ curve

The Γ curve corresponds to the *contraction* stage, described in last section (stage c in Fig. 2). It can be understood as the result of applying the backward induction principle: it corresponds to the response of the system (with the time running backwards and starting from steady state) to a step excitation of maximum magnitude $u = \pm\hat{u}$, up to the point where x_0 reaches its constraint, and $u = 0$ from then on. If the current state of the system is located at any point along this curve, it will follow it and reach the target in minimum time by applying maximum actuation. Consequently, and according to the DPP, for any i.c. not in this curve, the state should be first taken towards it (the rollback stage: r in Fig. 2), before it can be driven towards the target.

The position of the i.c. with respect to Γ determines the sign of the actuation for the expansion stage:

$$u_e = -\text{sgn}\left(\Gamma_{23}(x_2) - x_3\right)\hat{u}, \tag{33}$$

where Γ_{23} is the projection in the x_2x_3 plane of the Γ curve.

In order to better understand the form that Γ_{23} takes, some magnitudes relative to it are first introduced, in terms of the parameters of the system model (11). The quantity

$$r = \frac{k_u}{\omega_c}\hat{u}\,, \tag{34}$$

is the radius of the circular trajectory followed by the state in the x_1x_2 plane, starting from the equilibrium point (14), with center at $(x_1, x_2) = \left(-\frac{k_d}{\omega_c}d, \mp\frac{k_u}{\omega_c}\hat{u}\right)$, when maximum actuation ($u = \pm\hat{u}$) is applied, and the time is let to run backwards. Under these conditions and in the same plane, the trajectory will develop an arch of length $r\theta^{\pm}$, before x_0 reaches its constraint $x_0 = \pm\hat{x}_0$, with

$$\theta^{\pm} = 2\pi\frac{\tau_0}{\hat{u}}\left(\pm\hat{x}_0 - \sigma x_3^*\right). \tag{35}$$

Once x_0 reaches its constraint, the state would have reached one of these two points:

$$x^{\pm} = \begin{pmatrix} \pm\hat{x}_0 \\ k_1\left(\pm\hat{x}_0 - \sigma x_3^{\pm}\right) \\ \pm r\left(\cos(\theta^{\pm}) - 1\right) \\ \pm\frac{r}{\tau_3\omega_c}\left(\theta^{\pm} - \sin(\theta^{\pm})\right) + x_3^* \end{pmatrix}. \tag{36}$$

If, from then on, $u = 0$ is applied, x_0 will remain on its constraint, and the state will follow a circular trajectory in the x_1x_2 plane, centered at $(x_1, x_2) = \left(-\frac{k_d}{\omega_c}d, 0\right)$ with radius given by:

$$(r^{\pm})^2 = (x_1^{\pm})^2 + (x_2^{\pm})^2\,. \tag{37}$$

With this, and using an *inverse time* version of (17), the projection of the Γ curve in the x_2x_3 plane (see Fig. 4) can be written as a function of x_2:

$$\Gamma_{23}(x_2) = \begin{cases} x_3^* & \text{if } x_2 = 0 \\ -\text{sgn}(x_2)\frac{r}{\tau_3\omega_c}\left(\theta_\Gamma - \sin(\theta_\Gamma)\right) + x_3^* & \\ \qquad \text{if } x_2^+ \leq x_2 \leq x_2^- \wedge x_2 \neq 0 & \\ -\text{sgn}(x_2)\left(\frac{\hat{x}_0}{\sigma} - \frac{1}{\tau_3\omega_c}\sqrt{(r^{\{\text{sgn}(x_2)\}})^2 - x_2^2}\right) + \frac{\sigma-1}{\sigma}d & \\ \qquad \text{if } x_2 < x_2^+ \vee x_2^- > x_2 & \end{cases}, \tag{38}$$

where

$$\theta_\Gamma = \cos^{-1}\left(1 - \frac{|x_2|}{r}\right). \tag{39}$$

4.2 Control algorithm

The proposed method is an *ad-hoc* discrete-time adaption of the STPM, described in section 3, with a reduced computational overhead by making use of the known properties of the solutions. Similarly as in the STPM, the core of the proposed method is constituted by simulations, or predictions of the state with a given actuation sequence. In the STPM, the control sequence is defined in terms of time intervals during which $u = \pm\hat{u}$ or $u = 0$ is applied, and which are then minimized subject to state and terminal constraints. In the proposed method, the position of the i.c. relative to the Γ curve defines the actuation sequence and it is divided in the three stages: *expansion*, *rollback* and *contraction*. The time for the expansion stage is assumed to be one sampling period and it might or might not occur in the following sampling period. The time for the rollback stage is calculated exploiting the geometrical properties of (17). No prediction is carried out for the *contraction* stage, since it is already given by the Γ curve. These steps are depicted in Fig. 3.

The expansion stage is simulated with a prediction for the state from the i.c. x_i up to point x_e with $u = u_e$. Starting from x_e, a long term prediction up to point x_r is carried out, to account for the *rollback* stage. The sign of the actuation for this stage is calculated using the position of x_e relative to the Γ curve in the x_1x_2 plane:

$$u_r = \begin{cases} -\hat{u} & \text{if } R^+ \vee \overline{(R^+ \vee R^-)} \wedge (x_1 < -k_d d) \\ \hat{u} & \text{if } R^- \vee \overline{(R^+ \vee R^-)} \wedge (x_1 > -k_d d) \\ 0 & \text{if } x_1 = x_2 = 0 \end{cases}, \tag{40}$$

with

$$R^+ := (x_1 + k_d d)^2 + (x_2 + r)^2 \leq r^2 \tag{41}$$

$$R^- := (x_1 + k_d d)^2 + (x_2 - r)^2 \leq r^2. \tag{42}$$

The time span for the rollback stage, t_r, is calculated exploiting the geometry of the trajectories described by the state in the x_1x_2 plane: it is given by the

angle subtended by the arc between x_e and x_r in the circumference with center $C = \left(-\frac{k_d}{\omega_c}d, -\frac{k_u}{\omega_c}u_e\right)$ and radius $r_r = ||C - (x_{e1}, x_{e2})||$:

$$t_r = \frac{\theta_r}{\omega_c}. \tag{43}$$

The point x_r in the x_1x_2 plane is calculated as the intersection of two circumferences, described by the following equations:

$$\left(x_1 + \frac{k_d}{\omega_c}d\right)^2 + \left(x_2 + \frac{k_u}{\omega_c}u_e\right)^2 = r_r^2\,, \tag{44}$$

$$\left(x_1 + \frac{k_d}{\omega_c}d\right)^2 + \left(x_2 - \frac{k_u}{\omega_c}u_e\right)^2 = \left(\frac{k_u}{\omega_c}\hat{u}\right)^2\,. \tag{45}$$

The first circumference represents the trajectory of the state during the *rollback* stage, the second is the trajectory of the state during the *contraction* stage, or the projection of the Γ curve in the x_1x_2 plane (see Sec. 4.1).

The control decision is taken upon the position of x_r relative to Γ in the x_2x_3 plane, compared to the position of the i.c.: if the state does not cross the Γ_{23} curve, the kinetic energy in the harmonic oscillator can still be expanded before proceeding to the rollback stage. On the other hand, if the state goes beyond the Γ_{23} curve, x_3 will necessarily have to show overshoot before reaching its reference and therefore, in this situation no expansion stage should take place and the algorithm should proceed to the rollback stage and the state steered towards Γ.

The control decision takes the form of a reference for an inner controller for x_0: if it is decided to proceed with the *expansion*, the reference for x_0 takes the value of its prediction with $u = u_e$. On the other hand, if the decision is to *roll-back*, the reference takes the value of the prediction for x_0 with $u = -u_e$.

The constraint on the magnitude of x_0 is taken into account considering that both the expansion and rollback stages can only develop up to a point where the constraint in x_0 is upheld. To implement this, whenever the expansion stage prediction violates the constraint with $u = \pm\hat{u}$, the prediction up to point x_e is instead simulated assuming that the actuation takes the exact value that steers the electrical torque (x_0) towards its constraint. This violates the *finite-state* assumption, but it is necessary to ensure the satisfaction of the constraint and, at the same time, that the whole range of x_0 will be utilized. On the other hand, whenever the time for the *roll-back* stage t_r is longer than the time required to drive x_0 to its constraint, t_r is cut down to the time that takes x_0 exactly to its constraint:

$$t_c = \frac{\tau_0}{u_e}(\mathrm{sgn}(u_e)\hat{x}_0 - x_0) \tag{46}$$

The proposed controller is implemented as a sampled system and thus is only allowed to switch the actuation at the sampling instants. This is not enforced in the STPM, consequently, the time-optimal behaviour described in the last section requires that switches occur at arbitrary instants. In the proposed algorithm, when the optimal switching time falls between the current and the next sampling time, the x_r prediction crosses the Γ, but proceeding directly to the *rollback* stage would fall short. To solve this and approximate the behaviour

generated by the STPM, two new predictions are introduced, with $u = 0$ and $u = -u_e$ for the *expansion* stage. Then, if the prediction assuming $u = u_e$, x_r^+ crosses Γ_{23}, but the prediction assuming $u = 0$, x_r^0 does not, the actuation is calculated as a linear combination of both u_e and 0: the linear combination between x_r^+ and x_r^0 that exactly hits Γ_{23}. An equivalent procedure is carried out when the current state is near the Γ_{23} curve and the predictions for the expansion stage cross it. In this situation the linear combination is calculated between the prediction that crosses Γ_{23} and the prediction that does not.

Finally, there are regions in the state space where $u_e = u_r$ (see eqs. (33) and (40)), i.e., the sign for the actuation for the rollback and expansion stages are the same. This means that x_2 is increasing the error in x_3.and u_e must be is applied (a reference $x_0^* = \text{sgn}(u_e)\hat{x}_0$ is given to the inner controller) to revert this situation.

Fig. 3 presents simulation results using the proposed scheme to control the abstract system in eq. (11). The behaviour generated by the proposed algorithm effectively emulates the behaviour produced by the STPM method applied to the same system.

4.3 Smoothening near the steady state

Just as in a sliding mode controller [20], the reference for T_e (x_0) produced by the algorithm described above depends on the position of the state (in the proposed scheme: the predicted state) relative to a switching curve, and it is always the result of applying an extreme value of the control set. This, combined with noisy measurements and a discrete-time regime, results in chattering around the switching curve and poor steady state performance. To avoid this, a classical and smoother LQR is introduced to take over the control the task, when the state is near its desired steady state. The tuning of the LQR and the region where it should take over were done empirically through simulations. The goal being to minimize noticeable effects during the transitions from one controller to the other.

4.4 Inner Controller - Predictive Torque Control

The design of proposed method was developed independently from the inner torque controller, but under the assumption that the torque dynamics can be approximated by those of a simple integrator and that the controller will react to errors in T_e as if it were a dead-beat controller: by either applying maximum actuation or exactly the required value to eliminate the error in one sampling period. A torque controller which fairly approximates this behavior is the *finite-set predictive torque controller* FS-PTC, which is a receding horizon controller with a prediction horizon of one sampling period that solves the optimal control problem by using the *finite-set assumption* [15, 21].

In this work, the FS-PTC was modified to allow the use of any voltage the VSI can synthesize by PWM, enabling fast and precise control of the electrical torque, without the need of a very high switching frequency. This is achieved by first introducing two variables related to the error in T_e and the satisfaction

Table 1: Experimental setup parameters

$p = 3$	$J_m = 3.265 \times 10^{-3}\,[\mathrm{kgm}^2]$
$r_s = 2.2\,[\Omega]$	$J_l = 8.815 \times 10^{-3}\,[\mathrm{kgm}^2]$
$\psi_m = 0.226\,[\mathrm{Wb}]$	$c = 260.657\,[\mathrm{Nm/rad}]$
$L_d = 8.4\,[\mathrm{mH}]$	
$L_q = 11.1\,[\mathrm{mH}]$	$h = 46.088\,[\mu\mathrm{s}]$

of the condition for MTPA [16],

$$e_T = T_e^* - T_e\,, \tag{47}$$

$$e_d = i_d + \frac{L_d - L_q}{\psi_m}\left(i_d^2 - i_q^2\right)\,. \tag{48}$$

The state predictions (using the zero and the active vectors as actuation) generate an hexagon in the $e_T\, e_d$ plane , with the predictions generated by the active vectors on its vertices, and the prediction generated by the zero vector inside the hexagon. If the sampling period is relatively short, it can be assumed that any point inside this hexagon can be reached by a linear combination of these predictions (and the voltage vectors that generate them). The voltage to be synthesized by the VSI is then calculated solving a space vector modulation problem in the $e_T e_d$ plane to reach $(e_T, e_d) = \vec{0}$, or the point resulting from intersecting the boundary of the hexagon and the ray between the prediction produced by the zero voltage vector and $(e_T, e_d) = \vec{0}$, when the latter lays outside the hexagon.

5 Results

The performance of the proposed controller was tested using an experimental setup consisting mostly of industrial components. The PMSM is driven by an industrial VSI, modified for direct access to the gate signals of the IGBTs. A 2.2 [kW] induction machine, driven by an industrial VSI set for torque control, is used as load. These are coupled through a specially designed flexible shaft. The gating signals for the VSI driving the PMSM are produced by a real time computer with an Intel processor running at a clock frequency of 1.4 [GHz]. Real-time operation is achieved with a Linux kernel, modified with the RTAI package. The control algorithm was written in C, so that the same code could be used for experimental tests and simulations. The latter were implemented in Python, with the SciPy/Numpy [22] and SClib packages [23].

Technical limitations of the test bench forced the reduction of the voltage range to produce the experimental results presented here: when modulating relatively high voltages, the zero voltage vector is applied for very short time and the noise introduced by the switching elements is very noticeable in the currents measurements, which are synchronized to occur while the zero vector is applied. To tackle this, the value for the maximum actuation was set to 90% of $\hat{u}$ (see eq. (2.2)).

The relevant parameters of the test bench are presented in table 1.

Fig. 4 presents a step response of the system using the proposed control method. The reference for T_s goes from approximately 0 [Nm] to -7.3 [Nm]. The

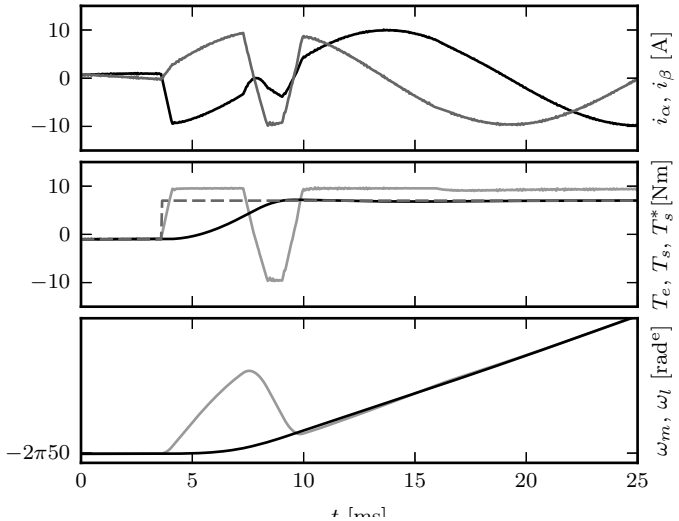

Figure 4: Reference Step change using the proposed method to control the torsional torque in a two-mass-system.

controller achieves reference tracking for T_s by forcing T_e towards its constraint for the whole transient. As a result T_s displays a sigmoid behavior, typical of time-optimal controllers.

It is possible to see some overshoot in T_s before it settles at its reference. This can be observed more explicitly in the difference between ω_m and ω_l. This may be the effect of neglecting the *back-emf* to calculate T_e^* in the SQTOC (see eq. (6)), since this effect is smaller, when operating at lower speeds. Otherwise, the obtained behaviour is as expected.

Fig. 5 presents experimental results using the proposed controller for T_s and an external PI controller for the load speed ω_l. The PI controller was tunned using pole placement for a plant with the form $G(s) = \frac{1}{J_l s}$, to obtain a closed loop natural frequency of approximately 8.67 [Hz], this is, only 6 times slower than then resonant frequency of the system. The test consists of a start-up maneuver, where the reference for ω_l goes from 0 to $2\,\pi 50\,[\text{rad}^e]$ (1000[rpm] mechanical), followed by a speed reversal and a load torque impact.

In Fig. 4, at $t \approx 16\,[\text{ms}]$ it is also possible to distinguish a small change in T_e. This is caused by the LQR, when it takes over. This effect is more patent in Fig. 5 at $t \approx 0.4\,[\text{s}]$ and $t \approx 0.75\,[\text{s}]$. Although this behaviour does not cause major problems, it would be desirable to have strategies to match both controllers at the transition boundary, or at least to verify that the LQR action will not take the state out of its region of activity resulting in chattering around the transition frontier.

6 Conclusions

The proposed controller effectively achieves the specified behavior and very closely approximates the behaviour generated by the STPM. The most notice-

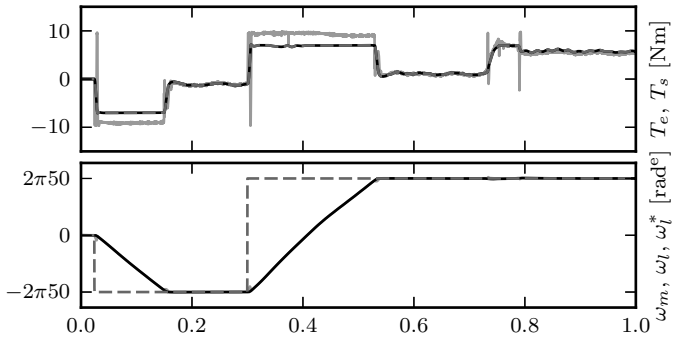

Figure 5: Load speed control for the two-mass system. The reference is given by a PI controller. The bandwidth for the speed controller was purposely set beyond the resonance frequency of the system: the inner controller successfully regulates oscillations at that frequency.

able difference is introduced by the restriction on the points in time where the proposed method is allowed to switch the actuation. The SQTOC scheme solves this by infringing the bang-bang structure when necessary.

At the core of the proposed method are simulations of the system dynamics, carried out with a sequence for the actuation, defined by the position of the state in the state space. This resembles the numerical implementation of the SPTM, which encompasses solving the system dynamics using different combinations for the actuation sequence. The formulation of the later method is very general and is valid for a broad family of systems. The information required by it is very scarce, it only requires a general specifications of the problem: system model, constraints and a criterion to evaluate a given solution and when an optimal solution is found, it is very reliable. However, this method requires many iterations, where different actuation sequences are tried out. The number of iterations may vary, depending on the set-up and the optimization scheme being used, but in general, heavy computational effort is required. This may render this approach unpractical, specially when the dynamics at hand are fast, as the dynamics of the motor. Nevertheless, the solutions found with this method do provide information on the nature of the expected closed loop dynamics. In this work, this was combined with a thorough study of the system dynamics to produce the proposed method, which can be feasible implemented.

This work illustrates how a general optimization method can be simplified and adapted to produce a feasible implementation for a particular problem.

References

[1] M. W. Spong, "Underactuated mechanical systems," in *Control Problems in Robotics and Automation.* Springer, 1998, pp. 135–150.

[2] S. Di Gennaro and M. Tursini, "Control techniques for synchronous motor with flexible shaft," in *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, Aug 1994, pp. 471–476 vol.1.

[3] P. Koronki, H. Hashimoto, and V. Utkin, "Direct torsion control of flexible shaft in an observer-based discrete-time sliding mode," *Industrial Electronics, IEEE Transactions on*, vol. 45, no. 2, pp. 291–296, Apr 1998.

[4] P. Koroundi, H. Hashimoto, and V. Utkin, "Discrete sliding mode control of two mass system," in *Industrial Electronics, 1995. ISIE '95., Proceedings of the IEEE International Symposium on*, vol. 1, Jul 1995, pp. 338–343 vol.1.

[5] K. Date, H. Ohmori, A. Sano, Y. Todaka, and H. Nishida, "Speed control of two-mass resonant system by new simple adaptive control scheme," in *Proc. IEEE International Conference on Control Applications*, vol. 2, 1–4 Sept. 1998, pp. 1120–1124.

[6] C. Hackl, H. Schuster, C. Westermaier, and D. Schroder, "Funnel-control with integrating prefilter for nonlinear, time-varying two-mass flexible servo systems," in *Advanced Motion Control, 2006. 9th IEEE International Workshop on.* IEEE, 2006, pp. 456–461.

[7] D. Szabo, S. Kerekes, O. Dranga, and T. Gajdar, "A fuzzy sliding mode approach for the two-mass system," in *Proc. IEEE International Symposium on Industrial Electronics ISIE '99*, vol. 1, 12–16 July 1999, pp. 348–352.

[8] S. Thomsen, N. Hoffmann, and F. W. Fuchs, "Pi control, pi-based state space control, and model-based predictive control for drive systems with elastically coupled loads—a comparative study," *IEEE Trans. Ind. Electron.*, vol. 58, no. 8, pp. 3647–3657, 2011.

[9] E. Fuentes, C. Silva, and J. Yuz, "Predictive speed control of a two-mass system driven by a permanent magnet synchronous motor," *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 7, pp. 2840–2848, July 2012.

[10] E. Fuentes and R. Kennel, "Finite-set model predictive control of the two-mass-system," in *Proc. IEEE 1th Workshop on Predictive Control of Electrical Drives and Power Electronics PRECEDE 2011*, 17–20 May 2011, pp. 390–395.

[11] T. Geyer, "Computationally efficient model predictive direct torque control," *Power Electronics, IEEE Transactions on*, vol. 26, no. 10, pp. 2804–2816, 2011.

[12] E. Fuentes and R. Kennel, "A finite-set model predictive position controller for the permanent magnet synchronous motor," in *Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE), 2013 IEEE International Symposium on*, Oct 2013, pp. 1–7.

[13] D. Paulus, P. Landsmann, and R. Kennel, "Sensorless field- oriented control for permanent magnet synchronous machines with an arbitrary injection scheme and direct angle calculation," in *Sensorless Control for Electrical Drives (SLED), 2011 Symposium on*, 2011, pp. 41–46.

[14] S. Villwock and M. Pacas, "Application of the welch-method for the identification of two- and three-mass-systems," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 1, pp. 457–466, Jan 2008.

[15] J. Rodriguez, J. Pontt, C. A. Silva, P. Correa, P. Lezana, P. Cortes, and U. Ammann, "Predictive current control of a voltage source inverter," *Industrial Electronics, IEEE Transactions on*, vol. 54, no. 1, pp. 495 –503, feb. 2007.

[16] M. Preindl and S. Bolognani, "Model predictive direct speed control with finite control set of pmsm drive systems," *Power Electronics, IEEE Transactions on*, vol. 28, no. 2, pp. 1007–1015, 2013.

[17] L. Pontryagin and L. Neustadt, *The Mathematical Theory of Optimal Processes*, ser. Classics of Soviet Mathematics. Gordon and Breach Science Publishers, 1962, no. Bd. 4.

[18] R. Bellman, "The theory of dynamic programming," *Bull. Amer. Math. Soc.*, vol. 60, pp. 503–515, 1954.

[19] C. Y. Kaya, S. K. Lucas, and S. T. Simakov, "Computations for bang−bang constrained optimal control using a mathematical programming formulation," *Optimal Control Applications and Methods*, vol. 25, no. 6, pp. 295–308, 2004. [Online]. Available: http://dx.doi.org/10.1002/oca.749

[20] V. Utkin, "Sliding mode control design principles and applications to electric drives," *Industrial Electronics, IEEE Transactions on*, vol. 40, no. 1, pp. 23–36, 1993.

[21] H. Miranda, P. Cortes, J. I. Yuz, and J. Rodriguez, "Predictive torque control of induction machines based on state-space models," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1916–1924, June 2009.

[22] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online]. Available: http://www.scipy.org/

[23] E. Fuentes and H. E. Martinez, "SClib, a hack for straightforward embedded C functions in Python," *ArXiv e-prints*, Dec. 2014. [Online]. Available: https://github.com/drestebon/SClib