

This is the post peer-review accepted manuscript of:

I. Notarnicola, F. A. Bayer, G. Notarstefano and F. Allgower, "Final-state constrained optimal control via a projection operator approach," 2016 European Control Conference (ECC), Aalborg, 2016, pp. 148-153.

The published version is available online at:

<https://doi.org/10.1109/ECC.2016.7810278>

© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Final-State Constrained Optimal Control via a Projection Operator Approach

Ivano Notarnicola¹, Florian A. Bayer², Giuseppe Notarstefano¹, and Frank Allgöwer²

Abstract—In this paper we develop a numerical method to solve nonlinear optimal control problems with final-state constraints. Specifically, we extend the PROjection Operator based Netwon’s method for Trajectory Optimization (PRONTO), which was proposed by Hauser for unconstrained optimal control problems. While in the standard method final-state constraints can be only approximately handled by means of a terminal penalty, in this work we propose a methodology to meet the constraints exactly. Moreover, our method guarantees recursive feasibility of the final-state constraint. This is an appealing property especially in realtime applications in which one would like to be able to stop the computation even if the desired tolerance has not been reached, but still satisfy the constraints. Following the same conceptual idea of PRONTO, the proposed strategy is based on two main steps which (differently from the standard scheme) preserve the feasibility of the final-state constraints: (i) solve a quadratic approximation of the nonlinear problem to find a descent direction, and (ii) get a (feasible) trajectory by means of a feedback law (which turns out to be a nonlinear projection operator). To find the (feasible) descent direction we take advantage of final-state constrained Linear Quadratic optimal control methods, while the second step is performed by suitably designing a constrained version of the trajectory tracking projection operator. The effectiveness of the proposed strategy is tested on the optimal state transfer of an inverted pendulum.

I. INTRODUCTION

Optimal control problems (OCPs) are an active field of research in the controls community since they may arise in many application areas as, e.g., Process Control, Robotics, Aerospace and Automotive. Throughout the last decades, many different approaches have been presented to solve these problems. A possible classification of these methods has been given in [1]: (i) Dynamic programming, (ii) Indirect Methods, and (iii) Direct methods. While methods in the first class solve the OCP by finding optimal input segments using the Principle of Optimality (see, e.g., [2], or [3]), the ones in the second area are based on solving the necessary conditions for optimality using a (two-point) boundary value problem, which can be solved by means of calculus of variations ([4], [5]) or Pontryagin’s Maximum Principle ([6], [7]). The third direction is the most investigated and simplifies the OCP

by parameterizing the control. According to the way the dynamics is handled, these methods are classified into fully discretized (or collocation) methods (see, e.g., [8]) and direct shooting methods, where the dynamics are included by some integration scheme (see, e.g., [9]). A detailed overview over Direct methods can, for example, be found in [10].

Of special interest for our paper is the PROjection Operator based Newton method for Trajectory Optimization (PRONTO) which was introduced in [11], see also [12]. In contrast to many other approaches solving optimal control problems, this method is able to guarantee feasibility of the dynamics after each iteration of the underlying Newton method using a “projection operator” defined by a feedback, closed-loop system. According to the classification in [1] this can be seen as a combination of shooting and collocation.

This method was designed to handle unconstrained optimal control problems (and extended to input-constrained problems in [13]), considering final-state constraints only approximately by means of a final penalty. Matching exactly final-state constraints is of interest in many control applications. This is the case, for example, in the field of hybrid systems, that is, systems that consist of continuous and discrete event dynamics (see, e.g., [14] and the references therein). Discontinuous jumps of continuous states may occur when the system state traverses a certain region of the state space. This demands for an exact satisfaction of constraints on the final state. Another field where this is of interest is the field of Model Predictive Control (MPC) (see, e.g., [15] and the references therein). In MPC, the system is controlled by means of repeatedly solving a finite-horizon OCP. In many approaches within MPC, convergence and stability can be guaranteed if a certain terminal condition is satisfied. This leads to the need of an algorithm being able to handle final state constraints.

A first approach to solve the nonlinear transfer problem was introduced in [16]. In there, the terminal constraint was satisfied asymptotically by iteratively choosing a terminal reference until the actual final state matches the target one.

The contribution of this paper is twofold. First, we introduce a new projection operator, inspired by the one presented in [12], such that not only the dynamics, but also the terminal constraint is satisfied after each iteration of the optimization algorithm. We reformulate the constrained projection as a root-finding of an infinite dimensional functional, which can be accomplished by means of a Newton root-finding in Banach spaces. Then, based on this new projection operator, as main contribution we propose an optimal control method solving final-state constrained problems which shows recursive feasibility. The proposed algorithm consists of two

¹Ivano Notarnicola and Giuseppe Notarstefano are with the Department of Engineering, Università del Salento, Lecce, Italy, name.lastname@unisalento.it

²Florian Bayer and Frank Allgöwer are with the Institute for Systems Theory and Automatic Control, University of Stuttgart, 70550 Stuttgart, Germany, {bayer, allgower}@ist.uni-stuttgart.de

This result is part of a project that has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART).

F. Bayer and F. Allgöwer would like to thank the German Research Foundation (DFG) for financial support within the Cluster of Excellence in Simulation Technology (EXC 310/2) at the University of Stuttgart.

steps. First, a feasible descent direction is determined using a quadratic approximation of the nonlinear problem. The descent direction is chosen such that the mismatch on the final state is zero. Second, the perturbed curve is projected on the feasible manifold such that the *dynamics and the terminal constraint* are satisfied.

An interesting feature of the proposed algorithm is that it is amenable to realtime, fast MPC schemes. Indeed, in many applications one may not be able to run the algorithm until convergence is achieved with a desired tolerance. Due to a reduced computation time it could be that a (much) shorter number of iterations can be run. Since feasibility of both the dynamics and the final state-constraint are guaranteed at each iteration one can stop the computation and still get a feasible trajectory.

The paper is organized as follows. In Section II we introduce the problem setup and recall how to solve final-state constrained linear quadratic optimal control problems. PRONTO is introduced in Section III. Our new final-state constrained PRONTO is presented in Section IV and a numerical simulation for the optimal state-transfer of an inverted pendulum is given in Section V.

Notation: Given a smooth vector field $f(x, u)$, we denote by $f_x(\bar{x}, \bar{u})$ its derivative with respect to x evaluated at (\bar{x}, \bar{u}) , and, consistently, by f_u its derivative with respect to u . For the curve $\xi = (x(\cdot), u(\cdot))$, we introduce the projections $\pi_1 = [I \ 0]$ and $\pi_2 = [0 \ I]$ such that $x(\cdot) = \pi_1 \xi$ and $u(\cdot) = \pi_2 \xi$. Given a functional $\mathcal{G} : X \rightarrow \mathbb{R}$, with X a Banach space, and a point $\xi \in X$, we denote by $D\mathcal{G}(\xi)$ the first Fréchet derivative of \mathcal{G} evaluated at ξ , and, consistently, by $D^2\mathcal{G}(\xi)$ its second Fréchet derivative, [17].

II. PROBLEM SETUP AND PRELIMINARIES

In this paper we consider a final-state constrained optimal control problem. That is, we aim at finding a trajectory of a dynamical system that minimizes a given objective functional while satisfying an initial and a terminal constraint. Formally, we consider the problem

$$\begin{aligned} & \underset{(x(\cdot), u(\cdot))}{\text{minimize}} \int_0^T \ell(x(\tau), u(\tau)) \, d\tau \\ & \text{subject to } \dot{x}(t) = f(x(t), u(t)) \\ & \quad x(0) = x_0, \quad x(T) = x_T, \end{aligned} \quad (1)$$

where $\ell : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ is the running cost, $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is the nonlinear vector field describing the control system, and $x_0 \in \mathbb{R}^n$ and $x_T \in \mathbb{R}^n$ are the initial and final fixed states respectively. We assume ℓ and f to be \mathcal{C}^2 functions. Notice that in the rest of the paper, for the sake of brevity, we will omit the dimensions of the quantities when it will be clear from the equations.

Before stating the main assumptions for problem (1), we recall some notation that will be also useful in the rest of the paper. Consider the Hamiltonian of (1) given by

$$H(x(t), p(t), u(t)) := \ell(x(t), u(t)) + p(t)^T f(x(t), u(t)), \quad (2)$$

where $p(\cdot)$ is the costate. Then, for $\xi = (\bar{x}(\cdot), \bar{u}(\cdot))$ define

$$q(\xi) \cdot (\zeta, \zeta) := \int_0^T \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix}^T \begin{bmatrix} H_{xx}(\tau) & H_{xu}(\tau) \\ H_{ux}(\tau) & H_{uu}(\tau) \end{bmatrix} \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix} d\tau, \quad (3)$$

where $\zeta = (z(\cdot), v(\cdot))$ is a (state-input) curve representing a variation from ξ , while $H_{xx}(t)$, $H_{xu}(t)$ and $H_{uu}(t)$ denote the appropriate second derivative of the H evaluated along the extremal state-control-costate trajectory, e.g., $H_{xx}(t) = H_{xx}(\bar{x}(t), \bar{p}(t), \bar{u}(t))$.

Given a dynamical system $\dot{x} = f(x, u)$, $x(0) = x_0$, we say that a state-input curve $\xi = (\bar{x}(t), \bar{u}(t))$ is a trajectory of the system if it satisfies the dynamics, i.e., $\dot{\bar{x}}(t) = f(\bar{x}(t), \bar{u}(t))$ for all $t \in [0, T]$ and $\bar{x}(0) = x_0$. We denote the (infinite-dimensional) manifold of all system trajectories by \mathcal{T} , so that we write $\xi \in \mathcal{T}$.

Given a trajectory $\xi = (\bar{x}(t), \bar{u}(t))$, we denote by $T_\xi \mathcal{T}$ the manifold of curves $\zeta = (z(\cdot), v(\cdot))$ satisfying the linearized dynamics

$$\dot{z} = f_x(\bar{x}(t), \bar{u}(t))z + f_u(\bar{x}(t), \bar{u}(t))v \quad (4)$$

with $z(0) = 0$ and for $v(\cdot) \in L_2$. We say that $T_\xi \mathcal{T}$ is the tangent space of the trajectory manifold at ξ .

Assumption 2.1 (Linear controllability): The system $\dot{x} = f(x, u)$ is linearly controllable around any trajectory. That is, for any $(\bar{x}(\cdot), \bar{u}(\cdot))$ defined on $[0, T]$, the linearized system (4) is controllable over $[0, T]$.

Assumption 2.2 (Second Order Sufficiency): Given a trajectory $\xi \in \mathcal{T}$, the Hamiltonian H satisfies $H_{uu}(t) \geq r_0 I$ for $t \in [0, T]$ and some $r_0 > 0$, and the quadratic functional q is positive-definite¹ on $T_\xi \mathcal{T}$. \square

Theorem 2.3 ([16, Theorem 2.1]): Let $\xi = (x(\cdot), u(\cdot))$ be a stationary trajectory of (1) with corresponding costate trajectory $p(\cdot)$. Suppose that Assumption 2.2 hold at ξ . If the system is linearly controllable around ξ , then ξ is an isolated local minimum of (1). \square

Remark 2.4: Assumption 2.1 not only is a sufficient condition for the theorem above, but also guarantees that the algorithm we propose will be solvable at each iteration. \square

A. Linear Quadratic (LQ) optimal state transfer problem

We start by considering a special version of problem (1) in which the cost is quadratic and the dynamics is linear and time-varying, i.e., we consider the problem

$$\begin{aligned} & \underset{(x(\cdot), u(\cdot))}{\text{minimize}} \int_0^T a(\tau)^T x(\tau) + b(\tau)^T u(\tau) \\ & \quad + \frac{1}{2} \begin{bmatrix} x(\tau) \\ u(\tau) \end{bmatrix}^T \begin{bmatrix} Q(\tau) & S(\tau) \\ S(\tau)^T & R(\tau) \end{bmatrix} \begin{bmatrix} x(\tau) \\ u(\tau) \end{bmatrix} d\tau \\ & \text{subject to } \dot{x} = A(t)x + B(t)u, \quad x(0) = x_0, \quad x(T) = x_T, \end{aligned} \quad (5)$$

where we assume that $a(\cdot)$ and $b(\cdot)$ are piecewise continuous vectors, and $A(\cdot)$, $B(\cdot)$, $Q(\cdot) = Q(\cdot)^T$, $R(\cdot) = R(\cdot)^T$, and $S(\cdot)$ are piecewise continuous matrices with $R(t) \geq r_0 I$, $t \in [0, T]$, for some $r_0 > 0$.

¹See, e.g., [17] for the definition of positive definite functional.

Remark 2.5: Problem (5) can be obtained as the linear-quadratic approximation of problem (1). In particular, $A(\cdot)$ and $B(\cdot)$ result from the linearization of the nonlinear dynamics f at a given trajectory, while Q , R , S , a and b define the quadratic approximation of the nonlinear cost functional ℓ at the same trajectory. \square

Theorem 2.6 ([16, Proposition 1.1]): If $(A(\cdot), B(\cdot))$ in (5) describes a controllable linear time-varying system over $[0, T]$ and q is positive definite on the space of the system trajectories, then problem (5) has a unique solution. \square

Next, we recall how to solve problem (5). We start by imposing the first-order necessary conditions of optimality.

Setting to zero the first variation of the Hamiltonian with respect to u , we obtain the optimal feedback law

$$u = -R^{-1}[S^T x + B^T p + b]. \quad (6)$$

By setting the first variations of the Hamiltonian with respect to x and p to zero and by using (6), we obtain the following linear two-point boundary value problem

$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} \tilde{A} & -BR^{-1}B^T \\ -\tilde{Q} & -\tilde{A}^T \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} + \begin{bmatrix} -BR^{-1}b \\ SR^{-1}b - a \end{bmatrix}, \quad \begin{matrix} x(0) = x_0 \\ p(T) = p_1 \end{matrix}, \quad (7)$$

where $p(t)$ is the costate, p_1 is a boundary value to be determined, $\tilde{A} := A - BR^{-1}S^T$ and $\tilde{Q} := Q - SR^{-1}S^T$.

It can be shown that p and x in (7) are related via an affine relation, i.e.,

$$p = Px + r. \quad (8)$$

By defining the gain matrix $K := R^{-1}(S^T + B^T P)$, the optimal input (6) results into the affine feedback law $u = -Kx - R^{-1}(B^T r + b)$. Then, equation (7) can be decoupled by means of the sweep method, [3], which leads to the following differential (Riccati) equations

$$-\dot{P} = A^T P + PA - K^T R K + Q, \quad P(T) = 0 \quad (9)$$

$$-\dot{r} = (A - BK)^T r - K^T b + a, \quad r(T) = p_1 \quad (10)$$

where the boundary conditions follow from (8).

The above equations should be integrated to determine the optimal control (6) and thus solve problem (5). However, the terminal vector p_1 is still unknown. Thus, we need to express explicitly the relation between p_1 and the terminal condition x_T . Plugging (8) into the first equation of (7), we obtain

$$\dot{x} = (A - BK)x - BR^{-1}(B^T r + b), \quad x(0) = x_0. \quad (11)$$

Next, we observe that

$$x(T) = x_u(T) + x_{f,b}(T) + x_{f,r}(T), \quad (12)$$

where $x_u(T)$ is the unforced response of system (11) at time $t = T$, whereas $x_{f,b}(T)$ and $x_{f,r}(T)$ are the forced responses due to the inputs $BR^{-1}b$ and $BR^{-1}B^T r$, respectively.

Focusing on $x_{f,r}(T)$, we note that it can be further split into two contributions related, respectively, to the forced and unforced responses of r . The latter contribution depends directly on p_1 and it can be shown that equation (12) can

be rewritten as $x(T) = x_u(T) + n(T) - W_c(T)p_1$, where $W_c(T)$ is the controllability Gramian matrix,

$$W_c(t) := \int_0^t \Phi_c(t, \tau) B(\tau) R(\tau)^{-1} B(\tau)^T \Phi_c(t, \tau)^T d\tau,$$

evaluated at time T , with Φ_c being the state transition function associated to closed-loop system with state matrix $A - BK$, while $n(T)$ denotes the terminal state of

$$\dot{n} = (A - BK)n - BR^{-1}(B^T r_f + b), \quad n(0) = 0,$$

where r_f denotes the forced response of r , i.e., it solves (10) with zero terminal condition.

To conclude, p_1 can be computed as

$$p_1 = W_c(T)^{-1} (x_T - x_u(T) - n(T)).$$

III. PROJECTION OPERATOR NEWTON METHOD FOR TRAJECTORY OPTIMIZATION (PRONTO)

PRONTO was introduced in [12] to solve the following finite-horizon optimal control problem

$$\begin{aligned} & \underset{(x(\cdot), u(\cdot))}{\text{minimize}} \int_0^T \ell(x(\tau), u(\tau)) d\tau + m(x(T)) \\ & \text{subject to } \dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \end{aligned} \quad (13)$$

which, differently from problem (1), has a terminal penalty $m: \mathbb{R}^n \rightarrow \mathbb{R}$ rather than a terminal constraint.

The key idea of PRONTO is to (i) convert the dynamically constrained (infinite-dimensional) optimization problem into an unconstrained one by means of a projection operator, and (ii) solve the unconstrained problem via an infinite-dimensional Newton method.

We start recalling the projection operator, which is based on a trajectory tracking feedback law.

A. The trajectory tracking nonlinear projection operator

Suppose that $\xi := (\alpha(\cdot), \mu(\cdot))$ (defined on $t \geq 0$) is a bounded state-input curve and let $\eta := (x(\cdot), u(\cdot))$ be the trajectory determined by the nonlinear feedback system

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)), & x(0) = \alpha(0) \\ u(t) = \mu(t) + K(t)[\alpha(t) - x(t)]. \end{cases} \quad (14)$$

Under suitable conditions on f and K , the feedback system in (14) defines a *continuous* nonlinear projection operator $\mathcal{P}: \xi = (\alpha(\cdot), \mu(\cdot)) \mapsto \eta = (x(\cdot), u(\cdot))$.

The operator \mathcal{P} is a projection since $\mathcal{P} = \mathcal{P} \circ \mathcal{P}$ on its domain. Indeed, independent of K , if ξ is a trajectory of f , then ξ is a *fixed point* of \mathcal{P} , i.e., $\xi = \mathcal{P}(\xi)$. As a consequence, a trajectory can be characterized in terms of the projection operator as $\xi \in \mathcal{T}$ if and only if $\xi = \mathcal{P}(\xi)$. In [12], the authors have proven that the projection operator \mathcal{P} is as smooth as f and one can compute (and analyze) its derivatives. In particular, if f is \mathcal{C}^1 , then the first derivative of the projection operator is the linear mapping $\zeta = (\beta(\cdot), \nu(\cdot)) \mapsto D\mathcal{P}(\xi) \cdot \zeta = (z(\cdot), v(\cdot))$ defined by

$$\begin{cases} \dot{z}(t) = f_x(x(t), u(t))z(t) + f_u(x(t), u(t))v(t), & z(0) = 0 \\ v(t) = \nu(t) + K(t)[\beta(t) - z(t)]. \end{cases}$$

which is obtained by linearizing (14) about $\xi \in \mathcal{T}$. It can be shown that $DP(\xi)$ is itself a projection, so that $\zeta \in T_\xi \mathcal{T}$ if and only if $\zeta = DP(\xi) \cdot \zeta$.

B. The PRONTO algorithm

Writing the cost in (13) as the functional

$$h(\xi) := \int_0^T \ell(x(\tau), u(\tau)) d\tau + m(x(T)),$$

we see that the optimal control problem (13) is equivalent to the constrained optimization problem $\min_{\xi \in \mathcal{T}} h(\xi)$. Using the trajectory characterization and defining $g(\xi) := h(\mathcal{P}(\xi))$ the constrained problem can be converted into an unconstrained one as $\min_{\xi \in \mathcal{T}} h(\xi) = \min_{\xi} g(\xi)$.

The PRONTO algorithm, stated in Algorithm 1, is based on a Newton method applied to $\min_{\xi} g(\xi)$ and includes two key steps. First, the search direction ζ_i is determined by an optimization problem considering the first and second derivatives of the nonlinear functional g . Since the derivatives of g are computed, the projection \mathcal{P} is inherently considered within the calculation of the search direction. Moreover, the search direction is limited to the tangent space of the trajectory manifold at the current trajectory ξ_i , that is, $\zeta_i \in T_{\xi_i} \mathcal{T}$. Second, the update is performed using the projection \mathcal{P} in (16), thus a feasible trajectory is determined after each iteration of the optimization algorithm.

Algorithm 1 PRONTO

GIVEN: initial trajectory $\xi_0 \in \mathcal{T}$

FOR: $i = 0, 1, 2, \dots$

redesign feedback K if desired/needed
search direction

$$\zeta_i = \underset{\zeta \in T_{\xi_i} \mathcal{T}}{\operatorname{argmin}} Dg(\xi_i) \cdot \zeta + \frac{1}{2} D^2g(\xi_i) \cdot (\zeta, \zeta) \quad (15)$$

step-size

$$\gamma_i = \underset{\gamma \in (0,1]}{\operatorname{argmin}} g(\xi_i + \gamma \zeta_i)$$

update

$$\xi_{i+1} = \mathcal{P}(\xi_i + \gamma_i \zeta_i) \quad (16)$$

Remark 3.1: Notice that step (15) consists of solving a (standard) LQR problem in the form

$$\begin{aligned} \underset{\zeta = (z(\cdot), v(\cdot))}{\operatorname{minimize}} & \int_0^T a(\tau)^T z(\tau) + b(\tau)^T v(\tau) \\ & + \frac{1}{2} \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix}^T \begin{bmatrix} Q(\tau) & S(\tau) \\ S(\tau)^T & R(\tau) \end{bmatrix} \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix} d\tau \\ & + z(T)^T P_1 z(T) + r_1^T z(T) \end{aligned}$$

subject to $\dot{z} = A(t)z + B(t)v$, $z(0) = 0$.

Step (16) consists of computing the updated trajectory $\xi_{i+1} = (x_{i+1}(\cdot), u_{i+1}(\cdot))$ by running the closed loop system (14) with (given) curve $(\alpha(\cdot), \mu(\cdot)) = \xi_i + \gamma_i \zeta_i = (x_i(\cdot) + \gamma_i z_i(\cdot), u_i(\cdot) + \gamma_i v_i(\cdot))$. \square

IV. FINAL-STATE CONSTRAINED PRONTO

In this section, we introduce an optimization algorithm which solves the nonlinear optimal state transfer problem. The key approach is to: (i) introduce a projection operator, inspired by the one introduced in [12] (and recalled in Section III), such that not only the dynamics, but also the terminal constraint is satisfied, and (ii) compute a descent direction that satisfies the final-state constraint to first-order.

A. Final-state constrained projection operator

The Projection Operator as recalled in Section III-A is not able to guarantee an exact matching of the terminal constraint. As a key step of our algorithm, we introduce a final-state constrained projection operator, $\xi = (\alpha(\cdot), \mu(\cdot)) \mapsto \mathcal{P}_c(\xi) = \eta = (x(\cdot), u(\cdot))$, satisfying $x(T) = \alpha(T)$ where, as usual, ξ is a curve while $\eta \in \mathcal{T}$ a trajectory. Our idea is to *design* the operator \mathcal{P}_c as an iterative routine in which, at each iteration: (i) we perturb the actual trajectory in order to hit exactly the terminal constraint and (ii) we project the resulting curve by means of the standard projection operator (14).

The final-state constrained projection can be formalized in terms of an infinite dimensional root-finding. Given $x_T \in \mathbb{R}^n$, let us define a functional \mathcal{F} which associates to a state-input curve $\xi = (\alpha(\cdot), \mu(\cdot))$ the difference between its terminal state $\alpha(T)$ and x_T . Hence, a *trajectory* η being a *root* of \mathcal{F} , i.e., such that $\mathcal{F}(\eta) = 0$, is exactly what we expect to be the result of the final-state constrained projection operator \mathcal{P}_c when applied to a curve ξ .

Following the same high level idea in Section III-B to derive the PRONTO algorithm, we convert the constrained root-finding of \mathcal{F} into the unconstrained root-finding of $\mathcal{G}(\cdot) := \mathcal{F}(\mathcal{P}(\cdot))$, with \mathcal{P} being the (unconstrained) projection operator introduced in (14).

Given an initial curve ξ , the root of the functional \mathcal{G} is found by means of an infinite-dimensional Newton method. Formally, at each iteration the perturbation ζ_k is obtained by setting to zero the first order approximation of the perturbed functional, i.e. by solving for ζ_k the following equation

$$D\mathcal{G}(\xi_k) \cdot \zeta_k = 0. \quad (17)$$

Using the chain rule, the linear mapping $D\mathcal{G}(\xi_k)$ applied to a state-input curve ζ_k can be expressed as $D\mathcal{G}(\xi_k) \cdot \zeta_k = D\mathcal{F}(\xi_k) \cdot DP(\xi_k) \cdot \zeta_k$. When ξ_k is a trajectory, the linear mapping $DP(\xi_k)$ is a projection on the tangent space $T_{\xi_k} \mathcal{T}$ (see [12]). Moreover, the first order expansion of the perturbed functional $\mathcal{F}(\xi_k + \zeta)$ turns out to be $D\mathcal{F}(\xi_k) \cdot \zeta = (\pi_1 \zeta)(T)$. Thus, we can conclude that equation (17) simply enforces a terminal condition on ζ_k , i.e., find the state component $z_k(\cdot)$ of $DP(\xi_k) \cdot \zeta_k \in T_{\xi_k} \mathcal{T}$ such that

$$x_k(T) - x_T + z_k(T) = 0. \quad (18)$$

Note that, since the linear mapping $D\mathcal{G}(\xi_k)$ is not invertible, the solution of (17) is not unique.

A finite dimensional counter-part of equation (17) is a linear system of the form $Mz + n = 0$. When $\ker M$

is non-empty, the equation has not a unique solution. A typical approach to overcome this problem is to consider the equivalent least-square problem, which selects the minimum norm solution of the linear system.

Motivated by this finite-dimensional observation, a reasonable choice is to select a $\zeta_k \in T_{\xi_k} \mathcal{T}$ satisfying condition (18) with minimum L_2 norm. It can be obtained solving the following linear quadratic optimal state transfer problem

$$\begin{aligned} \zeta_k := (z_k(\cdot), v_k(\cdot)) = \operatorname{argmin}_{(z(\cdot), v(\cdot))} & \frac{1}{2} \int_0^T \|z(\tau)\|^2 + \|v(\tau)\|^2 d\tau \\ \text{subj. to } & \dot{z} = A(t)z + B(t)v \\ & z(0) = 0, z(T) = -x(T) + x_T, \end{aligned}$$

where $A(\cdot)$ and $B(\cdot)$ result by the linearization of dynamics f around the current iterate ξ_k .

A pseudo code of the constrained projection operator \mathcal{P}_c is given in the following table (Algorithm 2).

Algorithm 2 Final-state constrained projection operator

GIVEN: a curve $\bar{\xi}$, a projection operator \mathcal{P} and a tolerance value tol , SET: $\xi_0 = \bar{\xi}$

FOR: $k = 0, 1, 2, \dots$
search direction

$$\begin{aligned} \zeta_k = \operatorname{argmin}_{\zeta} & \frac{1}{2} \|\zeta\|_{L_2}^2 \\ \text{subj. to } & \zeta \in T_{\xi_k} \mathcal{T} \\ & (\pi_1 \zeta)(T) = -\mathcal{F}(\xi_k) \end{aligned}$$

update

$$\xi_{k+1} = \mathcal{P}(\xi_k + \zeta_k)$$

IF: $\|\mathcal{F}(\xi_{k+1})\| < \text{tol}$, THEN: break.

SET: $\mathcal{P}_c(\bar{\xi}) = \xi^*$, being ξ^* the *last* iteration trajectory.

Remark 4.1: The convergence of Algorithm 2 can be guaranteed by satisfying the hypotheses of Newton-Kantorovich theorem (see, e.g., [18], [19]). \square

B. fsPRONTO Algorithm

We are ready to present the final-state constrained Projection Operator Newton method for Trajectory Optimization (fsPRONTO) algorithm which is an iterative algorithm able to solve problem (1). The algorithm extends the PRONTO outlined in Section III-B combining a particular descent direction and the final-state constrained projection operator presented in Section IV-A.

First, we search for a descent direction $\zeta_i \in T_{\xi_i} \mathcal{T}$ satisfying the final constraint to first-order by means of a linear-quadratic state transfer problem as in (5). Since each ξ_i is already feasible, in order to maintain feasibility to first order, the perturbation ζ_i must satisfy the terminal constraint $z_i(T) := (\pi_1 \zeta_i)(T) = 0$. Second, we perform a backtracking line-search to modulate the descent direction. Finally, we perform the projection step by means of the constrained projection operator described by Algorithm 2.

The fsPRONTO algorithm is formally stated in the following table (Algorithm 3).

Algorithm 3 Final-state constrained PRONTO.

GIVEN: initial trajectory ξ_0

FOR: $i = 0, 1, 2, \dots$

redesign feedback K if desired/needed
constrained search direction

$$\zeta_i = \operatorname{argmin}_{\zeta \in T_{\xi_i} \mathcal{T}} Dg(\xi_i) \cdot \zeta + \frac{1}{2} D^2g(\xi_i) \cdot (\zeta, \zeta) \quad (19)$$

subj. to $(\pi_1 \zeta_i)(T) = 0$

step-size

$$\gamma_i = \operatorname{argmin}_{\gamma \in (0,1]} g(\xi_i + \gamma \zeta_i) \quad (20)$$

constrained update

$$\xi_{i+1} = \mathcal{P}_c(\xi_i + \gamma_i \zeta_i) \quad (21)$$

In the following, we have a closer look at some of the specific aspects of our newly presented Algorithm 3.

Remark 4.2: Notice that step (19) consists of solving a linear quadratic optimal state transfer problem in the form

$$\begin{aligned} \text{minimize}_{\zeta=(z(\cdot), v(\cdot))} & \int_0^T a(\tau)^T z(\tau) + b(\tau)^T v(\tau) \\ & + \frac{1}{2} \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix}^T \begin{bmatrix} Q(\tau) & S(\tau) \\ S(\tau)^T & R(\tau) \end{bmatrix} \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix} d\tau \end{aligned}$$

subject to $\dot{z} = A(t)z + B(t)v$, $z(0) = 0$, $z(T) = 0$,

as discussed in detail in Section II-A. Step (21) consists of computing the updated trajectory $\xi_{i+1} = (x_{i+1}(\cdot), u_{i+1}(\cdot))$ via Algorithm 2 with a (given) curve $\xi = \xi_i + \gamma_i \zeta_i = (x_i(\cdot) + \gamma_i z_i(\cdot), u_i(\cdot) + \gamma_i v_i(\cdot))$. \square

V. NUMERICAL COMPUTATIONS

In this section we provide numerical computations showing the effectiveness of the proposed nonlinear algorithm. We solve the optimal state transfer problem for a driven inverted pendulum. We consider the problem

$$\begin{aligned} \text{minimize}_{(x(\cdot), u(\cdot))} & \int_0^T \frac{1}{2} \|x(\tau) - x_d(\tau)\|_Q^2 + \frac{1}{2} \|u(\tau) - u_d(\tau)\|_R^2 d\tau \\ \text{subject to } & \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{g}{L} \sin x_1 - \frac{u}{L} \cos x_1 \end{bmatrix}, \quad x(0) = x_0, \\ & x(T) = x_T, \end{aligned}$$

with $L = 0.5$ m being the length of the pendulum and g the gravity acceleration. We set the time horizon to $T = 20$ s. Moreover, $(x_d(\cdot), u_d(\cdot))$ is a (continuous) desired curve, $Q \in \mathbb{R}^{2 \times 2}$ is a symmetric, positive-definite matrix and R is a positive scalar.

Before testing the fsPRONTO algorithm, we highlight the applicability of the final-state constrained projection operator presented in Algorithm 2.

We consider a given curve ξ which is not a feasible trajectory of the inverted pendulum. The projected state x_1 is depicted in Figure 2. Both projections $\mathcal{P}(\xi)$ (in magenta)

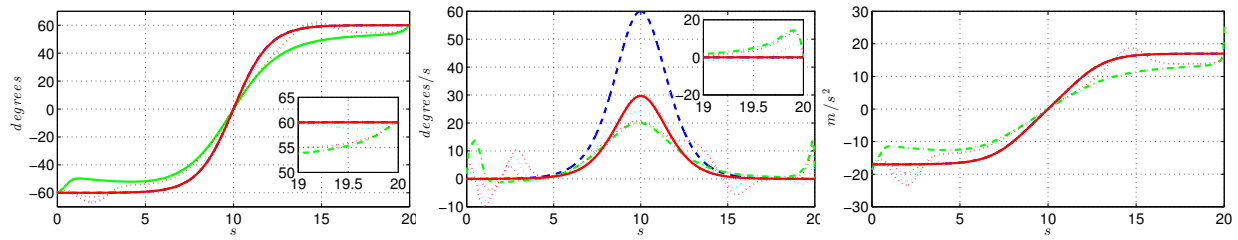


Fig. 1. Evolution of x_1 , x_2 and u through the iterations of fsPRONTO (Algorithm 3). The desired curve (dashed blue), the initial (feasible) trajectory (dashed-dot green) and the optimal trajectory (solid red) are depicted. Intermediate (feasible) trajectories are plotted with light dotted lines.

and $\mathcal{P}_c(\xi)$ (in red) provide a trajectory close to the curve ξ (in green). However, when closely checking the terminal state, one can see that only the trajectory projected under $\mathcal{P}_c(\xi)$ satisfies the terminal constraint.

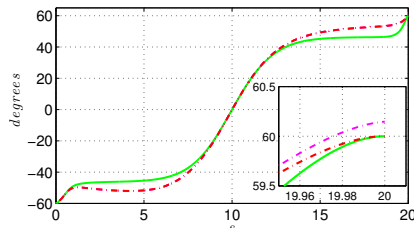


Fig. 2. Final-state constrained projection operator: x_1 state component. Specifically, the unfeasible curve ξ (solid green), the standard projection $\mathcal{P}(\xi)$ (dashed-dot magenta) and the constrained projection $\mathcal{P}_c(\xi)$ (dashed-dot red) are depicted.

Next, we apply the fsPRONTO (Algorithm 3) in order to optimize the trajectory of an inverted pendulum. We use $Q = \text{diag}(100, 1)$ and $R = 1$ as cost parameters. The choice of a higher penalty on the first component x_1 of the least-square distance will result in an optimal solution (solid red) which almost overlaps the first component of the desired curve (dashed-dot blue) as shown in Figure 1.

It is worth nothing that, as expected, the algorithm guarantees recursive feasibility. In fact, the terminal error, highlighted in the inset, is zero at each iteration for both the state components.

In Figure 3 the descent at each iteration, in logarithmic scale, is depicted. It gives a measure of the rate of convergence of the algorithm which appears to be quadratic.

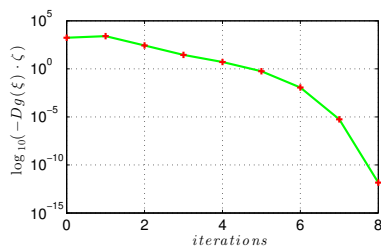


Fig. 3. Convergence Rate of fsPRONTO Algorithm.

VI. CONCLUSIONS

In this paper we have presented a new numerical approach for solving final-state constrained optimal control problems. The main advantage of the proposed method is that it

guarantees recursive feasibility of both the dynamics and the final-state constraint at each iteration. Specifically, we have proposed a Newton method, inspired to the one introduced in [11], based on: (i) the design of a final-state constrained projection operator, being able to find a trajectory satisfying the final constraint, and (ii) the computation of a descent direction satisfying the final constraint to first-order.

REFERENCES

- [1] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*. Springer, 2006, pp. 65–93.
- [2] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2005.
- [3] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control - Optimization, Estimation, and Control*. Hemisphere Publishing Cooperation, 1975.
- [4] D. E. Kirk, *Optimal Control Theory*. Prentice-Hall Inc., 1970.
- [5] A. P. Sage, *Optimum Systems Control*. Prentice-Hall, 1968.
- [6] L. S. Pontryagin, V. G. Boltyanskii, R. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*. Wiley (NY), 1962.
- [7] D. Liberzon, *Calculus of Variations and Optimal Control Theory: a Concise Introduction*. Princeton, NJ: Princeton University Press, 2012.
- [8] A. Cervantes and L. T. Biegler, "Large-scale DAE optimization using a simultaneous NLP formulation," *AIChE Journal*, vol. 44, no. 5, pp. 1038–1050, 1998.
- [9] H. G. Bock and K. J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proc. 9th IFAC World Congress Budapest (Hungary)*, July 1984, pp. 242 – 247.
- [10] J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2001.
- [11] J. Hauser, "A Projection Operator Approach to the Optimization of Trajectory Functionals," in *Proc. 15th IFAC World Congress*, vol. 15, 2002, pp. 310–315.
- [12] J. Hauser and D. G. Meyer, "The Trajectory Manifold of a Nonlinear Control System," in *Proc. 37th IEEE Conf. Decision and Control (CDC)*, vol. 1, 1998, pp. 1034 –1039 vol.1.
- [13] A. Saccon, J. Hauser, and A. Beghi, "A virtual rider for motorcycles: An approach based on optimal control and maneuver regulation," in *3rd International Symposium on Communications, Control and Signal Processing (ISCCSP)*, 2008, pp. 243 –248.
- [14] R. Goebel, R. Sanfelice, and A. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. New Jersey: Princeton University Press, 2012.
- [15] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill, 2009, vol. 1.
- [16] J. Hauser, "On the Computation of Optimal State Transfers with Application to the Control of Quantum Spin Systems," in *Proc. IEEE American Control Conference (ACC)*, vol. 3, 2003, pp. 2169–2174.
- [17] E. Zeidler, *Applied Functional Analysis: Applications to Mathematical Physics*. Springer, 1995, vol. 108.
- [18] L. V. Kantorovich, "Functional Analysis and Applied Mathematics," *Uspekhi Matematicheskikh Nauk*, vol. 3, no. 6, pp. 89–185, 1948.
- [19] J. M. Ortega, "The Newton-Kantorovich Theorem," *American Mathematical Monthly*, vol. 75, no. 6, pp. 658–660, 1968. [Online]. Available: <http://www.jstor.org/stable/2313800>