PANTR: A proximal algorithm with trust-region updates for nonconvex constrained optimization

Alexander Bodard, Pieter Pas and Panagiotis Patrinos

***Abstract*— This work presents PANTR, an efficient solver for nonconvex constrained optimization problems, that is well-suited as an inner solver for an augmented Lagrangian method. The proposed scheme combines forward-backward iterations with solutions to trust-region subproblems: the former ensures global convergence, whereas the latter enables fast update directions. We discuss how the algorithm is able to exploit exact Hessian information of the smooth objective term through a linear Newton approximation, while benefiting from the structure of box-constraints or ℓ_1-regularization. An open-source C++ implementation of PANTR is made available as part of the NLP solver library ALPAQA. Finally, the effectiveness of the proposed method is demonstrated in nonlinear model predictive control applications.**

I. Introduction

A. Background and motivation

Various areas of science and engineering naturally give rise to constrained, potentially nonconvex optimization problems, motivating the need for efficient solvers. A prominent example is found in the field of model predictive control (MPC) [1]. Historically, there has been a strong focus on linear MPC, where linear system dynamics lead to convex (and often linear-quadratic) optimization problems. However, recent advances in hardware have sparked an increased interest in nonlinear MPC (NMPC), which produces more challenging nonconvex, nonlinear programs (NLPs).

In this context, not only the efficiency of the solver, but also its reliability and memory requirements are critical. Classical techniques for solving the NLPs include interior point (IP) methods and sequential quadratic programming (SQP) [2]. State-of-the-art IP solvers such as IPOPT [3] are reliable as general-purpose solvers, but have the disadvantage that they do not easily exploit warm starts and have large memory requirements. SQP involves repeatedly solving a quadratic program (QP), and relies on efficient QP solvers. An overview of recent advances in this area is given in [4]. As an alternative to these classical approaches, PANOC [5] is a first-order method that combines forward-backward (FB) iterations with quasi-Newton directions to attain fast local convergence. PANOC benefits from warm starts and has a much smaller memory footprint. Since PANOC was originally proposed, its effectiveness in real-time MPC applications has been shown in various works [6]–[8].

The underlying premise of PANOC is that the proximal mapping of the objective's nonsmooth term can be efficiently evaluated (§II). To handle more general constraints, the ALPAQA package [9] implements an augmented Lagrangian method (ALM) that uses PANOC as an inner solver. Both quasi-Newton and Gauss-Newton [10] variants of PANOC speed up convergence through approximate second-order information. In contrast, the present work applies directions generated by a linear Newton approximation (LNA) [11], with the goal of enabling faster convergence than PANOC through *exact Hessian information* of the smooth term of the cost. For nonconvex optimization problems, the LNA may have negative eigenvalues. We therefore propose a semismooth Newton scheme that computes update directions as solutions to trust-region (TR) subproblems, allowing the exploitation of directions of negative curvature of the LNA.

B. Contributions

The contributions of this work include: (*i*) we propose PANTR, a novel proximal method that combines FB iterations with solutions to TR subproblems and as such is able to exploit exact second-order information for nonconvex problems; (*ii*) we prove global convergence of PANTR, without requiring feasibility of the iterates; (*iii*) we show that if the TR constraint becomes inactive, PANTR generates fast Newton directions; (*iv*) an open-source C++ implementation of PANTR is made available as part of ALPAQA;[1] (*v*) we show the effectiveness of PANTR on a number of NMPC problems. In particular, PANTR appears to perform well on problems where PANOC struggles.

Notation

By $\mathbb{N}$, $\mathbb{R}$ and $\overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$ we denote the set of natural, real and extended real numbers respectively. The restriction of $\mathbb{N}$ to $[i, j]$ is written as $\mathbb{N}_{[i,j]} := \mathbb{N} \cap [i, j]$ and x_i denotes the i'th component of $x \in \mathbb{R}^n$. We write $x_{\mathcal{I}} = (x_i)_{i\in\mathcal{I}}$ for an index set $\mathcal{I} \subseteq \mathbb{N}_{[1,n]}$, and denote the Euclidean inner product and norm by $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ respectively. The proximal operator of a function $h : \mathbb{R}^n \to \overline{\mathbb{R}}$ is $\mathbf{prox}_h(x) := \mathbf{arg\,min}_u\{h(u) + \frac{1}{2}\|u - x\|^2\}$. By Π_C we denote the Euclidean projection on a set C. The set $\mathbf{fix}\, T := \{x \in \mathbb{R}^n \mid x \in T(x)\}$ contains the fixed points of an operator $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$. We denote the set

This work was supported by: Research Foundation Flanders (FWO) PhD grant No. 11M9523N and research projects G081222N, G033822N, G0A0920N; Research Council KU Leuven C1 project No. C14/18/068; EU's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 953348; Fonds de la Recherche Scientifique - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlaanderen under EOS project no 30468160 (SeLMA).

The authors are with the Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium. Email: {alexander.bodard,pieter.pas,panos.patrinos}@esat.kuleuven.be

[1] https://github.com/kul-optec/alpaqa

of k times continuously differentiable functions by $\mathcal{C}^k$. We say that $f \in \mathcal{C}^1$ is L_f-smooth when ∇f is L_f-Lipschitz continuous. We denote the Clarke generalized Jacobian of a function $F : \mathbb{R}^n \to \mathbb{R}^m$ by $\partial_C F$ [12]. *lsc* and *osc* refer to lower and outer semicontinuity as in [13].

II. Problem statement and preliminaries

ALPAQA [9] combines an augmented Lagrangian method with inner solvers like PANOC [5], [14] to tackle NLPs

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\textbf{minimize}} && f(x) \\ &\textbf{subject to} && \underline{x} \le x \le \overline{x}, \qquad \underline{z} \le g(x) \le \overline{z}, \end{aligned} \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ may be nonconvex. PANOC attains fast convergence in solving the inner problem (P-FB) through quasi-Newton or Gauss-Newton directions [10]. However, disadvantages of the existing methods using quasi-Newton and Gauss-Newton directions are that (*i*) the Hessian is kept positive definite, creating a potential disagreement with the curvature of the underlying function; (*ii*) no *exact* Newton directions are exploited. Hence, we propose an alternative solver that exploits *second-order structure* of the inner problem.

For nonconvex problems, using the exact Hessian or a possibly indefinite approximation thereof requires *regularization* whenever the quadratic model is unbounded below. Techniques like Levenberg-Marquardt or modification of the eigenvalues require careful selection of the regularization parameter to attain fast convergence. In this work, we choose to implicitly regularize by limiting the norm of the Newton step, which corresponds to solving a TR subproblem.

The remainder of this section briefly reviews the augmented Lagrangian method (ALM), existing FB schemes like PANOC, and TR methods.

A. Augmented Lagrangian method

Define $C := \{x \mid \underline{x} \le x \le \overline{x}\}$ and $D := \{z \mid \underline{z} \le z \le \overline{z}\}$. By use of a slack vector z, Problem (P) can be written as

$$\underset{x\in C, z\in D}{\textbf{minimize}} \quad f(x) \qquad \textbf{subject to} \qquad z = g(x). \tag{P-ALM}$$

Given a positive definite diagonal matrix $\Sigma \in \mathbb{R}^{m\times m}$, we define the augmented Lagrangian function with penalty factor Σ as $\mathcal{L}_\Sigma(x, z, y) := f(x) + \langle y, g(x) - z\rangle + \frac{1}{2}\|g(x) - z\|^2_\Sigma$. ALM applied to Problem (P-ALM) iteratively (*i*) minimizes $\mathcal{L}_\Sigma$ w.r.t. x and z; (*ii*) updates the Lagrange multipliers y; and (*iii*) updates the penalty factors Σ. For more detailed information on ALM in the context of ALPAQA, see [9].

B. Forward-backward schemes

Consider the composite minimization

$$\underset{x\in\mathbb{R}^n}{\textbf{minimize}} \quad \varphi(x) := \psi(x) + h(x) \tag{P-FB}$$

of two functions $\psi : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \overline{\mathbb{R}}$ where $\mathbf{prox}_{\gamma h}$ is efficiently evaluated. Throughout this work, we assume that (*i*) $\psi \in \mathcal{C}^1$ is L_ψ-smooth; (*ii*) h is proper, lsc and convex; and (*iii*) $\varphi \equiv \psi + h$ is lower bounded. These are standard assumptions for first-order splitting schemes, see e.g. [15, Assumption 10.1].[2] Note that (P-FB) reduces to *nonconvex constrained optimization* by defining h as the indicator function of a constraint set. Also observe that the ALM inner problem, i.e. the minimization of $\mathcal{L}_\Sigma$ with respect to x and z, can be formulated in the form (P-FB) by defining $\psi(x) = f(x) + \frac{1}{2}\mathbf{dist}^2_\Sigma(g(x) + \Sigma^{-1}y, D)$ and $h = \delta_C$, where $\mathbf{dist}_\Sigma(\cdot, D) := \mathbf{min}_{z\in D}\{\|z - \cdot\|_\Sigma\}$.[3]

Solutions to (P-FB) are fixed points of the *forward-backward operator* $T_\gamma(x) := \mathbf{prox}_{\gamma h}(x - \gamma\nabla\psi(x))$, or equivalently, zeros of the *fixed-point residual* $R_\gamma(x) := {}^1\!/_\gamma\,(x - T_\gamma(x))$, where $\gamma > 0$. The forward-backward splitting (FBS) scheme iteratively applies T_γ, and converges to a fixed-point of T_γ for sufficiently small step sizes γ. Various methods have been proposed to accelerate the convergence of FBS. For example, both PANOC [5], [14] and ZeroFPR [16] aim to find zeros of R_γ through a quasi-Newton line search procedure that uses the forward-backward envelope (FBE) [16]–[18] as a merit function.

Definition 1 (FBE)**.** *The FBE of φ with parameter $\gamma > 0$ is*

$$\varphi_\gamma(x) = \inf_{u\in\mathbb{R}^n} \psi(x) + \langle\nabla\psi(x), u - x\rangle + h(u) + \tfrac{1}{2\gamma}\|u - x\|^2.$$

The FBE is a useful metric, since its minimization is equivalent to the minimization of φ in the sense that $\mathbf{arg\,min}\,\varphi \equiv \mathbf{arg\,min}\,\varphi_\gamma$ for $\gamma \in (0, {}^1\!/_{L_\psi})$ [18, Pr. 2.3 iii]. Remark that when $\psi \in \mathcal{C}^2$, the FBE is continuously differentiable with $\nabla\varphi_\gamma(x) = Q_\gamma(x)R_\gamma(x)$ where $Q_\gamma(x) := \mathrm{I} - \gamma\nabla^2\psi(x)$ [18, Th. 2.6]. Yet, φ_γ is not twice differentiable in general, since R_γ is not generally differentiable.

C. Trust-region methods

To minimize a function $F : \mathbb{R}^n \to \mathbb{R}$, TR methods define a model m_k that locally approximates the objective function F, typically based on a second order Taylor expansion. A TR step d_k is then computed by minimizing the model m_k over all points within a distance Δ_k of the current iterate, i.e.

$$\begin{aligned} &\underset{d}{\textbf{minimize}} && m_k(d) := F(x_k) + \langle\nabla F(x_k), d\rangle + \tfrac{1}{2}\langle B_k d, d\rangle \\ &\textbf{subject to} && \|d\| \le \Delta_k \end{aligned} \tag{1}$$

where B_k is the (approximate) Hessian matrix of F evaluated at x_k. The TR radius Δ_k is updated depending on how well m_k approximates F, through the ratio

$$\rho_k = \tfrac{F(x_k) - F(x_k + d_k)}{m_k(0) - m_k(d_k)}. \tag{2}$$

The TR subproblem can be interpreted as adaptively regularizing B_k, since for any global minimizer of (1) [19, Cor. 7.2.2]

$$\exists\lambda \ge 0 : (B_k + \lambda\,\mathrm{I})\,d = -\nabla F(x_k) \;\text{ and }\; B_k + \lambda\,\mathrm{I} \succeq 0. \tag{3}$$

Approximately solving this subproblem efficiently is critical for the overall performance of TR methods. The *Steihaug*

[2] Section III-E relaxes the first assumption to include all functions ψ with *locally* Lipschitz-continuous gradients.

[3] Since $\mathbf{dist}^2_\Sigma(\cdot, D)$ has a piecewise linear gradient, it is Lipschitz-smooth. Hence, for *locally* Lipschitz-smooth functions g, also ψ is *locally* Lipschitz-smooth.

conjugate gradient (CG) method [20] is widely used, although other techniques exist. The interested reader is referred to [19] for an extensive study of TR methods.

III. PANTR

This section introduces the PANTR scheme for solving (P-FB). First, the semismooth Newton system for the root-finding problem $R_\gamma(x^\star) = 0$ is transformed into a minimization problem, and it is regularized by fitting it into a TR framework. Next, we globalize the scheme and analyze its convergence properties. Finally, we propose an adaptive step size procedure, such that no explicit knowledge of the Lipschitz constant L_ψ is needed.

A. Newtonian directions for the fixed-point residual

As a surrogate for solving (P-FB), we are interested in finding a point $x_\star$ such that $R_\gamma(x_\star) = 0$. PANOC addresses this problem using a Newton-type approach and computes update directions $d_k = -B_k^{-1} R_\gamma(x_k)$, where B_k is an invertible matrix. If B_k captures first-order information of R_γ, this enables superlinear convergence when sufficiently close to a strong local minimum. The operators B_k are typically defined using an L-BFGS scheme. However, this enforces symmetry and positive definiteness of B_k, which restricts the algorithm's ability to exploit directions of negative curvature.

Ideally, we are interested in performing Newton's method, which defines B_k as the exact Jacobian of the fixed-point residual $\mathrm{J}R_\gamma(\widehat{x})$. Unfortunately, R_γ is *not in general differentiable*. Hence, we propose to use a linear Newton approximation (LNA) [11] of R_γ instead. Define the generalized Hessian of ψ as $\partial^2\psi := \partial_C(\nabla\psi)$. Then

$$\hat{\partial} R_\gamma(x) := \left\{ \tfrac{1}{\gamma}\left(\mathrm{I} - \mathcal{P}_\gamma \mathcal{Q}_\gamma\right) \middle| \begin{smallmatrix} \mathcal{P}_\gamma \in \partial_C \, \mathbf{prox}_{\gamma h}(x - \gamma\nabla\psi(x)) \\ \mathcal{Q}_\gamma \in \mathrm{I} - \gamma\partial^2\psi(x) \end{smallmatrix} \right\} \quad (4)$$

is a LNA of R_γ at $x_\star$ under a mild semismoothness assumption [10, Prop. 1]. Remark that this is a generalization of the LNA originally proposed in [21, Prop. 4.13] for $\psi \in \mathcal{C}^2$.

At a given iterate $\widehat{x}$, using the residual $R_\gamma(\widehat{x})$ and its LNA $\mathcal{R}_\gamma \in \hat{\partial} R_\gamma(\widehat{x})$, we thus aim to compute an update direction d that solves the Newton system $\mathcal{R}_\gamma d = -R_\gamma(\widehat{x})$. To this end, the following theorem introduces a constrained minimization problem, the solution of which matches the solution to the Newton system of the fixed-point residual under the idempotence assumption $\mathcal{P}_\gamma^2 = \mathcal{P}_\gamma$. Observe that this condition can be satisfied when $h = \delta_S$ where S is a polyhedral set [21, §6.2b] and, by the Moreau decomposition, when h is the support function of a polyhedral set S.

Theorem 1. (Solutions of Newton system) *Select $H \in \partial^2\psi(\widehat{x})$ and $\mathcal{P}_\gamma \in \partial_C \, \mathbf{prox}_{\gamma h}(\widehat{x} - \gamma\nabla\psi(\widehat{x}))$, such that $\mathcal{R}_\gamma = \frac{1}{\gamma}(\mathrm{I} - \mathcal{P}_\gamma\mathcal{Q}_\gamma)$. Suppose that $\mathcal{P}_\gamma^2 = \mathcal{P}_\gamma$, and define $\mathcal{P}_\gamma^\perp := I - \mathcal{P}_\gamma$. Then, a solution $d^\star$ of the minimization problem*

$$\begin{aligned} \underset{d}{\textbf{minimize}} \quad & \tfrac{1}{2}\langle d, Hd\rangle + \langle R_\gamma(\widehat{x}), d\rangle \\ \textbf{subject to} \quad & \mathcal{P}_\gamma^\perp d = -\gamma \mathcal{P}_\gamma^\perp R_\gamma(\widehat{x}) \end{aligned} \quad (5)$$

also solves the Newton system of the fixed-point residual,

$$\mathcal{R}_\gamma \, d^\star = -R_\gamma(\widehat{x}). \quad (6)$$

Proof. Being the generalized Jacobian of a proximal mapping, $\mathcal{P}_\gamma$ is symmetric [21, Thm. 15.4.12]. The stationarity condition of problem (5) is given by $Hd^\star + R_\gamma(\widehat{x}) + \mathcal{P}_\gamma^\perp\lambda = 0$, for some multiplier λ. Multiplying by $\mathcal{P}_\gamma$ and using $\mathcal{P}_\gamma\mathcal{P}_\gamma^\perp = 0$ yields $\mathcal{P}_\gamma H d^\star = -\mathcal{P}_\gamma R_\gamma(\widehat{x})$. Replacing $\mathcal{P}_\gamma = I - \mathcal{P}_\gamma^\perp$ on the right-hand side and substituting the constraint results in $\mathcal{P}_\gamma H d^\star = -R_\gamma(\widehat{x}) - \gamma^{-1}\mathcal{P}_\gamma^\perp d^\star$, implying (6). □

B. Box constraints and ℓ_1-norm

We now specialize to two classes of functions whose proximal operators satisfy the idempotence assumption $\mathcal{P}_\gamma^2 = \mathcal{P}_\gamma$, i.e. the indicators of rectangular boxes, and the ℓ_1-norm. First, consider the case where h is the indicator of the set of box constraints $C = \{x \mid \underline{x} \le x \le \overline{x}\}$. The proximal operator then reduces to a projection onto C, i.e. $\mathbf{prox}_{\gamma h} = \Pi_C$ [13, Sec. 1.G]. Similarly to the derivation in [9, Sec. III], define the set $\mathcal{K} := \underline{\mathcal{K}}(\widehat{x}) \cup \overline{\mathcal{K}}(\widehat{x})$ of indices – for brevity we omit the argument of $\mathcal{K}$ – corresponding to *active* box constraints on $\widehat{x} - \gamma\nabla\psi(\widehat{x})$ (i.e. after a forward step on $\widehat{x}$) where

$$\begin{aligned} \underline{\mathcal{K}}(\widehat{x}) &:= \left\{ i \in \mathbb{N}_{[1,n]} \mid (\widehat{x})_i - \gamma\nabla_{x_i}\psi(\widehat{x}) \le \underline{x}_i \right\}, \\ \overline{\mathcal{K}}(\widehat{x}) &:= \left\{ i \in \mathbb{N}_{[1,n]} \mid \overline{x}_i \le (\widehat{x})_i - \gamma\nabla_{x_i}\psi(\widehat{x}) \right\}. \end{aligned} \quad (7)$$

We denote the complement of $\mathcal{K}$, containing the indices of the *inactive* constraints, by $\mathcal{J} = \mathbb{N}_{[1,n]} \backslash \mathcal{K}$. To simplify notation, consider a row permutation matrix $\mathrm{P}_{\mathcal{K}\mathcal{J}}$ that reorders the rows with indices $k \in \mathcal{K}$ before those with indices $j \in \mathcal{J}$. In this case, we can select $\mathcal{P}_\gamma \in \partial_C(\mathbf{prox}_{\gamma h}(\widehat{x} - \gamma\nabla\psi(\widehat{x})))$ to be $\mathcal{P}_\gamma = \mathrm{P}_{\mathcal{K}\mathcal{J}}^\top \left(\begin{smallmatrix} 0_{|\mathcal{K}|} & 0 \\ 0 & \mathrm{I}_{|\mathcal{J}|} \end{smallmatrix}\right) \mathrm{P}_{\mathcal{K}\mathcal{J}}$ [21, Sec. 6.2d], indeed satisfying the idempotence assumption of Theorem 1. Substituting this choice of $\mathcal{P}_\gamma$ into problem (5), the constraint can be written as $d_\mathcal{K} = -\gamma\,[R_\gamma(\widehat{x})]_\mathcal{K}$. It only remains to solve the, potentially smaller, unconstrained minimization problem

$$\underset{d_\mathcal{J}}{\textbf{minimize}} \quad \tfrac{1}{2}\langle d_\mathcal{J}, H_{\mathcal{J}\mathcal{J}} d_\mathcal{J}\rangle + \langle [R_\gamma(\widehat{x})]_\mathcal{J} + H_{\mathcal{J}\mathcal{K}} d_\mathcal{K}, d_\mathcal{J}\rangle. \quad (8)$$

Here, $H_{\mathcal{J}\mathcal{K}}$ and $H_{\mathcal{J}\mathcal{J}}$ are the bottom left and right blocks of $\mathrm{P}_{\mathcal{K}\mathcal{J}}^\top H \mathrm{P}_{\mathcal{K}\mathcal{J}}$ respectively.

The structure of $\mathcal{P}_\gamma$ for the case where the nonsmooth term consists of an ℓ_1-norm is similar to that for box constraints. When $h = \lambda\|\cdot\|_1$ with $\lambda > 0$, we have that $\mathbf{prox}_{\gamma h}(x)_i = \mathbf{sign}(x_i)\,\mathbf{max}\,(|x_i| - \gamma\lambda, 0)$. Defining $\mathcal{K} := \left\{ i \in \mathbb{N}_{[1,n]} \mid |(\hat{x}_k)_i| \le \gamma\lambda \right\}$ again yields problem (8).

C. Adaptive regularization

To deal with cases where $H_{\mathcal{J}\mathcal{J}}$ is not positive definite, we introduce a radius constraint to keep the solution of (8) well-defined, giving rise to the following TR subproblem.

$$\begin{aligned} \underset{d_\mathcal{J}}{\textbf{minimize}} \quad & q^\mathcal{J}(d_\mathcal{J}) := \tfrac{1}{2}\langle d_\mathcal{J}, H_{\mathcal{J}\mathcal{J}} d_\mathcal{J}\rangle \\ & \qquad + \langle [R_\gamma(\widehat{x})]_\mathcal{J} + H_{\mathcal{J}\mathcal{K}} d_\mathcal{K}, d_\mathcal{J}\rangle \\ \textbf{subject to} \quad & \|d_\mathcal{J}\| \le \Delta \end{aligned} \quad (9)$$

When the radius constraint is inactive, by Theorem 1, a solution of (9) is exactly a Newton step for the root-finding problem of the residual. Moreover, when $\psi \in \mathcal{C}^2$, it is also a Newton step for the problem of minimizing the FBE,

as can be verified using $\nabla\varphi_\gamma(x) = Q_\gamma(x)R_\gamma(x)$ where $Q_\gamma = \mathrm{I} - \gamma\nabla^2\psi$, and the fact that $\hat{\partial}^2\varphi_\gamma(x) := Q_\gamma(x)\hat{\partial}R_\gamma(\widehat{x})$ is a LNA for $\nabla\varphi_\gamma$ [21, Cor. 15.4.14].

D. Globalization

To summarize, thus far we described a way to compute update directions for solving the root-finding problem $R_\gamma(x^\star) = 0$ by solving a smaller TR subproblem. Whenever the block $H_{\mathcal{J}\mathcal{J}}$ of the Hessian is positive definite and the TR constraint is inactive, that direction is exactly the Newton direction, and whenever this is not the case, the radius constraint implicitly regularizes the problem.

Although we can reasonably expect a scheme using those directions to yield fast local convergence, it still lacks a *globalization strategy*. Hence, we propose to first perform a *forward-backward* (FB) step $\hat{x}_k \in T_\gamma(x_k)$ at the iterate x^k, and then compute a candidate *accelerated* step d_k as the solution to (9) at the point $\hat{x}_k$ (i.e., using $\widehat{x} = \hat{x}_k$ when evaluating $H, \mathcal{P}_\gamma, R_\gamma, \mathcal{K}$ and $\mathcal{J}$). Whether or not the candidate step d_k is accepted is determined by the ratio

$$\rho_k := \tfrac{\varphi_\gamma(\hat{x}_k) - \varphi_\gamma(\hat{x}_k + d_k)}{-q(d_k)}, \tag{10}$$

which verifies descent on the FBE, similar to PANOC. It would thus be natural to define a quadratic model of the FBE, e.g. using the LNA of [21, Cor. 15.4.14] as $q^{\varphi_\gamma}(d) := \frac{1}{2}\langle d, \mathcal{Q}_\gamma\mathcal{R}_\gamma\, d\rangle + \langle \mathcal{Q}_\gamma R_\gamma(\widehat{x}), d\rangle$. A solution $d^\star_{\mathcal{J}}$ to the reduced problem (9) satisfies $q^{\mathcal{J}}(d^\star_{\mathcal{J}}) \leq 0$, which does not guarantee $q^{\varphi_\gamma}(d^\star) \leq 0$. This is problematic, since a positive ρ_k need not imply descent on the FBE. Instead, we define $q(d) := q^{\mathcal{J}}(d_{\mathcal{J}}) - \frac{1}{2\gamma}\|d_{\mathcal{K}}\|^2$, for which we do have that $q(d) \leq q^{\mathcal{J}}(d_{\mathcal{J}})$. Remark that this model avoids the computation of (expensive) products $HR_\gamma(\widehat{x})$. Based on the ratio ρ_k, we update the radius Δ_k using

$$\Delta_{k+1} = \begin{cases} \mathbf{max}\{c_3\|d_k\|, \Delta_k\} & \rho_k \geq \mu_2 \\ c_2\Delta_k & \mu_1 \leq \rho_k < \mu_2 \\ c_1\|d_k\| & \rho_k < \mu_1. \end{cases} \tag{11}$$

Since the only purpose of updating Δ_k is to generate better candidate steps in the next iteration, various alternatives to this heuristic are possible. When combined, the FB step, the solution of a TR subproblem, and the update of the TR radius make up the PANTR scheme, as summarized in Algorithm 1.

Algorithm 1 PANTR

procedure PANTR($x_0, \Delta_0, \gamma, \mu_1, \mu_2, c_1, c_2, c_3$)
  for $k = 1, 2\ldots$ **do**
    Select $\hat{x}_k \in T_\gamma(x_k)$;
    Compute d_k as the solution to (9) at $\widehat{x} = \hat{x}_k$;
    Compute ρ_k as in (10) and update Δ_{k+1} as in (11);
    $x_{k+1} \leftarrow \begin{cases} \hat{x}_k + d_k & \rho_k \geq \mu_1 \\ \hat{x}_k & \rho_k < \mu_1; \end{cases}$

The reference implementation of PANTR solves subproblem (9) using the Steihaug CG method. This method requires only Hessian-vector products and has a limited memory footprint, making it suitable for embedded applications. Hessian-vector products $H_{\mathcal{J}\mathcal{J}}d_{\mathcal{J}}$ can be approximated by a quasi-Newton method on $\nabla\psi$, by finite differences, or using automatic differentiation. Note that positive definiteness of the quasi-Newton approximants is not required. The following theorem describes the global subsequential convergence of the presented method.

Theorem 2. *Let $\omega(x_k)$ denote the set of cluster points of $(x_k)_{k\in\mathbb{N}}$. Then for the iterates $(x_k)_{k\in\mathbb{N}}$ of* PANTR*:*
(i) The sequence $(\varphi_\gamma(x_k))_{k\in\mathbb{N}}$ is nonincreasing;
(ii) $R_\gamma(x_k) \to 0$ and $R_\gamma(\hat{x}_k) \to 0$ square summably;
(iii) $\omega(x_k) = \omega(\hat{x}_k) \subseteq \mathbf{fix}\, T_\gamma$;
(iv) The sequence $(\varphi_\gamma(x_k))_{k\in\mathbb{N}}$ converges to a finite value $\varphi_\star$, and so does the sequence $(\varphi(\hat{x}_k))_{k\in\mathbb{N}}$ when $(x_k)_{k\in\mathbb{N}}$ is bounded.

Proof. Let $\beta = \frac{1-\gamma L_\psi}{2}$. Claim *(i)* follows from the fact that

$$\varphi_\gamma(x_{k+1}) \leq \varphi_\gamma(\hat{x}_k) \leq \varphi(\hat{x}_k) \leq \varphi_\gamma(x_k) - \gamma\beta\|R_\gamma(x_k)\|^2, \tag{12}$$

where we consecutively used (10) and $q(d_k) \leq 0$, [18, Prop. 2.2 *(i)*], and [18, Prop. 2.2 *(ii)*]. As for *(ii)*, by telescoping (12) and using the lower-boundedness of φ – and thus of φ_γ – we prove $R_\gamma(x_k) \to 0$ square summably. For $R_\gamma(\hat{x}_k) \to 0$ square summably, the proof is similar, but instead uses

$$\varphi_\gamma(T_\gamma(\hat{x}_k)) \leq \varphi(T_\gamma(\hat{x}_k)) \leq \varphi_\gamma(\hat{x}_k) - \gamma\beta\|R_\gamma(\hat{x}_k)\|^2.$$

Assume that for some $x' \in \mathbb{R}^n$ and $K \subseteq \mathbb{N}$, we have $\{x_k\}_{k\in K} \to x'$. Then $\{\hat{x}_k\}_{k\in K} \to x'$ as well, since $\|\hat{x}_k - x_k\| = \gamma\|R_\gamma(x_k)\| \to 0$. The arbitrarity of x' implies that $\omega(x_k) \subseteq \omega(\hat{x}_k)$, and by a similar argument also the converse inclusion holds. Hence, $\omega(x_k) = \omega(\hat{x}_k)$. Moreover, $x_k = \mathbf{prox}_{\gamma h}(x_k - \gamma\nabla\psi(x_k)) + \gamma R_\gamma(x_k)$, and since $\{x_k - \gamma\nabla\psi(x_k)\}_{k\in K} \to x' - \gamma\nabla\psi(x')$, the outer semicontinuity of $\mathbf{prox}_{\gamma h}$ implies that $x' = \mathbf{prox}_{\gamma h}(x' - \gamma\nabla\psi(x'))$. Hence, $x' \in \mathbf{fix}\, T_\gamma$, proving *(iii)*. From (12) it follows that $\varphi_\gamma(x_k) \to \varphi_\star$. If $\{x_k\}_{k\in\mathbb{N}}$ is bounded, then so is $\{\hat{x}_k\}_{k\in\mathbb{N}}$ due to compact-valuedness of $\mathbf{prox}_{\gamma h}$ [13, Thm. 1.25]. By [16, Prop. 4.2], φ_γ is also M-Lipschitz continuous on a compact set containing $\{x_k\}_{k\in\mathbb{N}}$ and $\{\hat{x}_k\}_{k\in\mathbb{N}}$ for some $M > 0$. Hence,

$$\varphi_\gamma(x_k) - \varphi(\hat{x}_k) \leq \varphi_\gamma(x_k) - \varphi_\gamma(\hat{x}_k) \leq M\gamma\|R_\gamma(x_k)\| \to 0.$$

Thus we have that $\{\varphi(\hat{x}_k)\}_{k\in\mathbb{N}} \to \varphi_\star$, establishing *(iv)*. □

E. Adaptive step size procedure

To ensure convergence, the step size γ is required to be smaller than the inverse of the Lipschitz constant L_ψ, for example by selecting $\gamma = \alpha/L_\psi$ for some $\alpha \in (0, 1)$. When the true value of L_ψ is unknown, it can be estimated as follows. The step size γ is updated adaptively by verifying

$$\psi(\hat{x}_k) \leq \psi(x_k) + \langle\nabla\psi(x_k), \hat{x}_k - x_k\rangle + \tfrac{\alpha}{2\gamma}\|\hat{x}_k - x_k\|^2 \tag{13}$$

at the start of every iteration of PANTR. Whenever violated, we set $\gamma \leftarrow \gamma/2$. Remark that this can only happen a finite number of times, since when $\alpha/\gamma > L_\psi$, (13) is automatically satisfied [16]. Thus, after a finite number of iterations γ is constant and all convergence results remain valid as of then.

Moreover, by similar arguments as in [16], *local* Lipschitz-continuity of $\nabla\psi$ suffices whenever the candidate directions

d_k are bounded and h has a bounded domain. The latter is satisfied for $h = \delta_C$ with C bounded.

IV. Applications and Numerical Results

The effectiveness of our method is demonstrated in the setting of optimal control. We show that PANTR (Alg. 1) performs on par with, or greatly outperforms ALPAQA's existing inner solvers – PANOC [5], [14] and structured PANOC [9]. Run times of IPOPT [3], the default solver in open-source MPC toolboxes such as `do_mpc` and `rockit-meco`, are also reported.

We consider two MPC problems: one with a *hanging chain* model, for which PANOC performs well, and one with a *quadcopter* model and general constraints, selected because it is a problem for which PANOC seems to struggle.

Benchmark problems

Both problems can be stated using the general formulation

$$\begin{aligned} \underset{\mathbf{x},\mathbf{u}}{\textbf{minimize}} \quad & \sum_{k=0}^{N-1} \ell_k(x^k, u^k) + \ell_N(x^N) \\ \textbf{subject to} \quad & x^{k+1} = f(x^k, u^k), && \forall k \in \mathbb{N}_{[0,N-1]} \\ & \underline{u} \le u^k \le \overline{u}, && \forall k \in \mathbb{N}_{[0,N-1]} \\ & \underline{z} \le c(x^k) \le \overline{z}, && \forall k \in \mathbb{N}_{[0,N]}. \end{aligned} \tag{OCP}$$

The function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ models the discrete-time dynamics of the system. The matrices $\mathbf{x} := (x^1, \ldots, x^N) \in \mathbb{R}^{n_x \times N}$ and $\mathbf{u} := (u^0, \ldots, u^{N-1}) \in \mathbb{R}^{n_u \times N}$ contain the state and input sequences, respectively. The cost is a sum of the stage costs ℓ_k and the terminal cost ℓ_N. Additionally, the inputs u^k are constrained by a rectangular box, and more general state constraints can be included as well through $c : \mathbb{R}^{n_x} \to \mathbb{R}^{n_c}$. All solvers are applied to the quadcopter and hanging chain OCPs, varying the OCP horizon length from 1 to 60, and simulating the MPC controller for 60 time steps. In a first experiment, each solver is cold-started; in a second experiment, the solvers are warm-started using the solution and multipliers from the previous run, shifted by one time step.

As a first benchmark problem, we consider the *hanging chain model* described in [22], using the parameters and initial state listed in section III. The second benchmark problem under consideration is a simplified *quadcopter model*, governed by the following continuous-time dynamics:

$$\dot{p} = v, \qquad \dot{v} = R(\theta)\,(0, 0, a_t)^\top + g, \qquad \dot{\theta} = \omega. \tag{14}$$

The state vector $x := (p, v, \theta) \in \mathbb{R}^9$ consists of the position $p \in \mathbb{R}^3$, the velocity $v \in \mathbb{R}^3$, and the orientation $\theta \in \mathbb{R}^3$, represented using Euler angles. The input $u := (a_t, \omega) \in \mathbb{R}^4$ consists of the thrust $a_t \in \mathbb{R}$ and the angular velocity $\omega \in \mathbb{R}^3$, and will be determined by the controller as the solution to a finite-horizon optimal control problem (OCP). $R(\theta) \in \mathrm{SO}(3)$ represents a rotation matrix, and $g = (0, 0, -9.81\,\mathrm{m\,s^{-2}})$ is the acceleration due to gravity. The cost function of the OCP aims to minimize the distance to the target position $p_{\text{ref}} := (0.25, 0.25, 0.5)$, and penalizes high velocities, angles and angular velocities, specifically, $\ell_k(x, u) := 10\,\|p - p_{\text{ref}}\|^2 + \|v\|^2 + \|\theta\|^2 + 10\,\|\omega\|^2 + 10^{-4} a_t^2$ and $\ell_N(x) := 10\,\|p - p_{\text{ref}}\|^2 + \|v\|^2 + \|\theta\|^2$. Additionally, we impose constraints on the maximum thrust and on the rate of rotation by defining $\underline{u} := (0, -0.1, -0.1, -0.1)$ and $\overline{u} := (49, 0.1, 0.1, 0.1)$. The state constraints limit the tilt angles and avoid a cylindrical object located at the origin, $c(x) := (\theta_x, \theta_y, \cos(\theta_x)\cos(\theta_y), p_x^2 + p_y^2)$, $\underline{z} := (-\frac{\pi}{2}, -\frac{\pi}{2}, \cos(\pi/6), 0.1^2)$, $\overline{z} := (\frac{\pi}{2}, \frac{\pi}{2}, +\infty, +\infty)$. The system dynamics are discretized using an explicit fourth-order Runge-Kutta integrator with a time step $T_s = 0.1\,\mathrm{s}$.

Solvers

A single-shooting formulation of the OCP is used, eliminating the dynamics constraint in (OCP). This yields a NLP of the form (P). Remark that ALPAQA thus handles the general state constraint c using its ALM. However, when no general state constraint is present, (P) boils down to (P-FB) with $h = \delta_C$, which is solved directly by PANOC or PANTR. We use an L-BFGS buffer of length 50 for the PANOC-based solvers, and for the parameters in Algorithm 1, we use $c_1 = 0.35$, $c_2 = 0.99$, $c_3 = 10$, $\mu_1 = 0.2$, $\mu_2 = 0.5$. The step size is determined adaptively as described in Section III-E. For IPOPT, both the main tolerance and the constraint violation tolerance are set to 10^{-8}. The three other solvers declare convergence when $\left\|\mathbf{u} - \Pi_{[\underline{u},\overline{u}]}(\mathbf{u} - \nabla\psi(\mathbf{u}))\right\|_\infty \le 10^{-8}$ and $\left\|c(\mathbf{x}) - \Pi_{[\underline{z},\overline{z}]}(c(\mathbf{x}) + \Sigma^{-1}y)\right\|_\infty \le 10^{-8}$. The maximum number of inner iterations per ALM subproblem is set to 250, and the initial penalty for the state constraints is set to 10^4, with a penalty increase factor of 5. The initial inner tolerance is set to 100, lowering it by a factor of 10 on each ALM iteration. The necessary derivatives are computed and precompiled using CasADi [23], except for the hanging chain problem with IPOPT, where precompilation is not possible due to the large Hessian matrix, and CasADi's virtual machine (VM) is used for evaluation instead.[4]

Hanging chain results

For the hanging chain problem, where PANOC performs quite well, PANTR is on par with PANOC, and significantly faster than IPOPT. Fig. 1 shows the average run times. Note that solvers that fully exploit the OCP structure, like the Gauss-Newton variant of PANOC from [10], might offer even higher performance. We do not consider them here, because PANTR is applicable to a wider range of problems, whereas PANOC with Gauss-Newton only applies to problems with a particular OCP structure. Similar specialization and exploitation of OCP structure for PANTR are the subject of future research.

Quadcopter results

Average solver run times for the quadcopter benchmark are shown in Fig. 2. PANOC and PANOC with approximate structured L-BFGS directions do not perform particularly well here, especially for larger horizons. This served as one of the

[4]The C++ source code to reproduce the results in this section can be found at github.com/kul-optec/pantr-cdc2023-experiments. All experiments were carried out using an Intel Core i7-11700 CPU at 2.5 GHz.

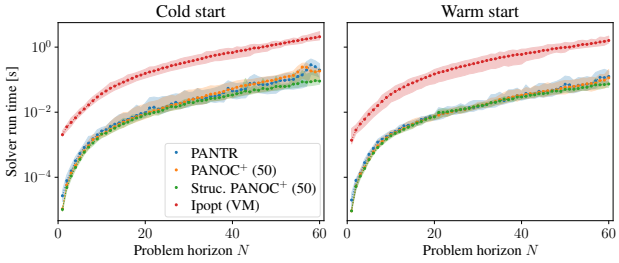

Fig. 1: Average solver run times for different horizon lengths of the hanging chain benchmark, with P5/P95 percentiles.

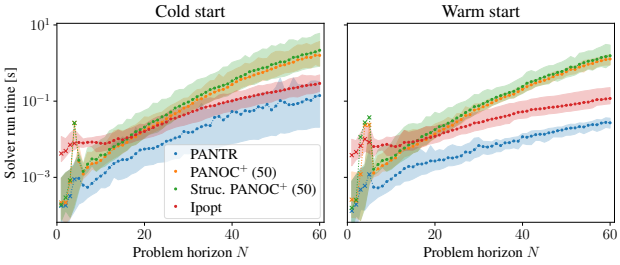

Fig. 2: Average solver run times for different horizon lengths of the quadcopter benchmark, with P5/P95 percentiles.

motivations for the development of PANTR. The latter is able to consistently outperform IPOPT, and scales better with the horizon N than PANOC. Although IPOPT does benefit from warm-starting slightly, the effect is much clearer for PANTR, which, when warm-started, achieves average run times that are around three times faster than IPOPT's for large problems.

Looking at the individual run times for horizon 60 in Fig. 3, it is clear that all solvers take more time during the first 12 MPC time steps, where the collision constraint is active (as shown in Fig. 4). For PANTR, this is alleviated by warm-starting, resulting in fast convergence throughout.

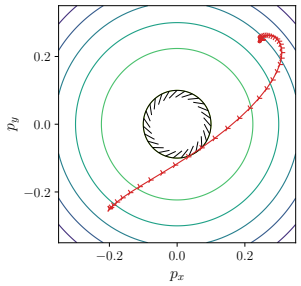

Fig. 4: Optimal trajectory of quadcopter ($N = 60$).

We observed that the average number of CG iterations per TR subproblem hovers around 10% of the number of variables. Evaluation of a Hessian-vector product using CasADi is only 1.2 to 3 times more expensive than a gradient evaluation, further motivating the use of a CG solver for (9).

V. Conclusion

This paper presented PANTR, a novel proximal algorithm for nonconvex constrained optimization that is well-suited as an ALM inner solver. The scheme locally performs Newton steps on an LNA of the fixed-point residual, and as such exploits exact Hessian information of the smooth cost term. Update directions are computed as solutions to TR subproblems, thus implicitly regularizing the corresponding Newton system. The presented scheme compares favorably

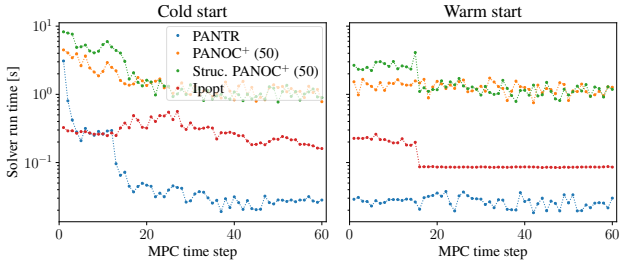

Fig. 3: Solver run time per MPC time step, for the quadcopter benchmark with horizon $N = 60$.

against state-of-the-art NLP solvers in NMPC applications, with the exact second-order information proving particularly beneficial for problems where first-order solvers struggle.

References

[1] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*, 2nd ed. Madison, WI: Nob Hill Publishing, 2017.

[2] J. Nocedal and S. J. Wright, *Numerical optimization*. NY: Springer, 2006.

[3] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[4] D. Kouzoupis, G. Frison, A. Zanelli, and M. Diehl, "Recent advances in quadratic programming algorithms for nonlinear model predictive control," *Vietnam Journal of Mathematics*, vol. 46, no. 4, pp. 863–882, Dec. 1, 2018.

[5] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, "A simple and efficient algorithm for nonlinear model predictive control," in *IEEE 56th Conference on Decision and Control (CDC)*, Dec. 2017, pp. 1939–1944.

[6] A. Sathya, P. Sopasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos, "Embedded nonlinear model predictive control for obstacle avoidance using PANOC," in *European Control Conference (ECC)*, Limassol: IEEE, Jun. 2018, pp. 1523–1528.

[7] E. Small, P. Sopasakis, E. Fresk, P. Patrinos, and G. Nikolakopoulos, "Aerial navigation in obstructed environments with embedded nonlinear model predictive control," in *18th European Control Conference (ECC)*, Jun. 2019, pp. 3556–3563.

[8] B. Lindqvist, S. S. Mansouri, J. Haluška, and G. Nikolakopoulos, "Reactive navigation of an unmanned aerial vehicle with perception-based obstacle avoidance constraints," *IEEE Trans Control Syst Technol*, vol. 30, no. 5, pp. 1847–1862, Sep. 2022.

[9] P. Pas, M. Schuurmans, and P. Patrinos, "Alpaqa: A matrix-free solver for nonlinear MPC and large-scale nonconvex optimization," in *European Control Conference (ECC)*, Jul. 2022, pp. 417–422.

[10] P. Pas, A. Themelis, and P. Patrinos, *Gauss-newton meets PANOC: A fast and globally convergent algorithm for nonlinear optimal control*, Dec. 8, 2022.

[11] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*, 2 vols. New York: Springer, 2003, vol. 1.

[12] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, Jan. 1990.

[13] R. T. Rockafellar and R. J. Wets, *Variational Analysis* (Grundlehren der mathematischen Wissenschaften). Springer, 1998, vol. 317.

[14] A. De Marchi and A. Themelis, "Proximal gradient algorithms under local lipschitz gradient continuity: A convergence and robustness analysis of PANOC," *J Optim Theory Appl*, vol. 194, no. 3, pp. 771–794, Sep. 2022.

[15] A. Beck, *First-Order Methods in Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics, Oct. 3, 2017.

[16] A. Themelis, L. Stella, and P. Patrinos, "Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone line-search algorithms," *SIAM J. Optim.*, vol. 28, no. 3, pp. 2274–2303, Jan. 2018.

[17] P. Patrinos and A. Bemporad, "Proximal newton methods for convex composite optimization," in *52nd IEEE Conference on Decision and Control*, Firenze: IEEE, Dec. 2013, pp. 2358–2363.

[18] L. Stella, A. Themelis, and P. Patrinos, "Forward–backward quasi-newton methods for nonsmooth optimization problems," *Comput Optim Appl*, vol. 67, no. 3, pp. 443–487, Jul. 1, 2017.

[19] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*. Society for Industrial and Applied Mathematics, Jan. 2000.

[20] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization," *SIAM Journal on Numerical Analysis*, Jul. 17, 2006.

[21] A. Themelis, M. Ahookhosh, and P. Patrinos, "On the acceleration of forward-backward splitting via an inexact newton method," in 2019, pp. 363–412.

[22] L. Wirsching, H. G. Bock, and M. Diehl, "Fast nmpc of a chain of masses connected by springs," 2006, pp. 591–596.

[23] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.