# A Feasible Sequential Linear Programming Algorithm with Application to Time-Optimal Path Planning Problems

David Kiessling[1], Andrea Zanelli[2], Armin Nurkanović[3], Joris Gillis[1], Moritz Diehl[3,4], Melanie Zeilinger[2], Goele Pipeleers[1], Jan Swevers[1]

***Abstract*— In this paper, we propose a Feasible Sequential Linear Programming (FSLP) algorithm applied to time-optimal control problems (TOCP) obtained through direct multiple shooting discretization. This method is motivated by TOCP with nonlinear constraints which arise in motion planning of mechatronic systems. The algorithm applies a trust-region globalization strategy ensuring global convergence. For fully determined problems our algorithm provides locally quadratic convergence. Moreover, the algorithm keeps all iterates feasible enabling early termination at suboptimal, feasible solutions. This additional feasibility is achieved by an efficient iterative strategy using evaluations of constraints, i.e., zero-order information. Convergence of the feasibility iterations can be enforced by reduction of the trust-region radius. These feasibility iterations maintain feasibility for general Nonlinear Programs (NLP). Therefore, the algorithm is applicable to general NLPs. We demonstrate our algorithm's efficiency and the feasibility update strategy on a TOCP of an overhead crane motion planning simulation case.**

## I. Introduction

We aim at solving Nonlinear Programs (NLP) arising in time-optimal motion planning of mechatronic systems. The TOCP we are looking at in this paper are obtained by the direct multiple shooting discretization [3] and written as follows

$$\min_{\substack{x_0,\dots,x_N\\ u_0,\dots,u_{N-1}\\ s_0,\,s_N,\,T}} \quad T + \mu_0^\top s_0 + \mu_N^\top s_N \tag{1a}$$

$$\text{s.t.} \quad -s_0 \le x_0 - \overline{x}_0 \le s_0, \tag{1b}$$

$$x_{k+1} = f(x_k,\, u_k,\, \tfrac{T}{N}),\ k = 0,\dots,N-1, \tag{1c}$$

$$u_k \in \mathbb{U}_k, \qquad k = 0,\dots,N-1, \tag{1d}$$

$$x_k \in \mathbb{X}_k, \qquad k = 0,\dots,N, \tag{1e}$$

$$e(x_k,\, u_k) \le 0, \qquad k = 0,\dots,N-1, \tag{1f}$$

$$-s_N \le x_N - \overline{x}_N \le s_N. \tag{1g}$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, $s_0$, $s_N \in \mathbb{R}^{n_x}$ denote the state, control, and slack variables for horizon length $N \in \mathbb{N}$. The time horizon is given by $T \in \mathbb{R}_{>0}$ and the multiple shooting time interval size is given by $h := \frac{T}{N}$. We denote the start and end points by $\bar{x}_0$, $\bar{x}_N \in \mathbb{R}^{n_x}$. Let $f\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $e\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_e}$ denote the system dynamics and the stage constraints, respectively. Additionally, let $\mu_0$, $\mu_N \in \mathbb{R}^{n_s}_{>0}$ denote penalty parameters and let $\mathbb{U}_k$, $\mathbb{X}_k$ denote convex polytopes.

Among other alternatives, Sequential Quadratic Programming (SQP) methods can be used to solve (1). Due to the nonlinearities introduced in the system dynamics and motion planning constraints in (1) and the difficulty of finding a good initial guess, it is often required to introduce a globalization strategy that ensures global convergence of the iterates. For an overview of different globalization strategies for SQP methods we refer to [5] and [12]. These standard methods often require tuning of many parameters to ensure global convergence. Moreover, the iterates are in general infeasible, which can lead to inconsistent inputs in a real-time control context where an SQP method may need to be terminated early [14].

In [16], the so-called Feasibility-Perturbed Sequential Quadratic Programming (FP-SQP) method was introduced. This method overcomes many of the aforementioned drawbacks by keeping all iterates feasible. This avoids the occurrence of infeasible, physically impossible states and controls. Retaining feasibility is achieved by an additional projection step onto the feasible set. A forward simulation of the system dynamics with the inputs of the SQP step can achieve feasibility with respect to the dynamic constraints. In [14], a more advanced forward simulation is used that also ensures feasibility with respect to linear input and path constraints. In [2], feasibility iterations based on repeated evaluation of zero-order information were introduced in the context of real-time Nonlinear Model Predictive Control (NMPC) algorithms under the name "feasibility improvement iterates". In [17], a feasible SQP method was introduced that uses these feasibility iterations to ensure the feasibility of every SQP iterate.

Due to the linear objective function in (1), we propose a Feasible Sequential Linear Programming (FSLP) algorithm that maintains the global convergence properties of FP-SQP. SLP methods were first introduced in [8] and in the case that the optimal solution is fully determined by the active set of constraints, they achieve locally quadratic convergence [11].

The proposed algorithm consists of outer iterations calculating a standard SLP step and inner feasibility iterations

This work has been carried out within the framework of Flanders Make SBO DIRAC: DIRAC - Deterministic and Inexpensive Realizations of Advanced Control. Flanders Make is the Flemish strategic research centre for the manufacturing industry.

[1] Department of Mechanical Engineering, KU Leuven, and DMMS lab, Flanders Make, Leuven, Belgium david.kiessling@kuleuven.be

[2] Institute for Dynamic Systems and Control, ETH Zurich, Switzerland

[3] Department of Microsystems Engineering (IMTEK), University of Freiburg, 79110 Freiburg, Germany

[4] Department of Mathematics, University of Freiburg, 79110 Freiburg, Germany

projecting the step onto the feasible set. The inner iterations repeatedly solve Linear Programs (LP) and converge linearly towards a feasible point. During this process, only zero-order information of the constraints, i.e., their residuals need to be reevaluated. In contrast to [17], the convergence of the iterations is achieved by a trust-region. We emphasize that the feasibility iterations do not depend on the optimal control problem structure and are suitable for general NLP. The overall algorithm is free of second-order derivative information and solves only LPs. Therefore, the overall computational cost of the proposed algorithm is low.

The paper is structured as follows. In Section II, the outer feasible sequential linear programming algorithm and its convergence behavior are presented. Section III discusses the inner feasibility iteration and the main local convergence results are derived. Simulation results on time-optimal motion planning of an overhead crane are presented in Section IV. This paper concludes with Section V.

## II. Feasible Sequential Linear Programming

In this section, we first introduce the problem formulation and the notation. Then, we describe the outer feasible sequential linear programming algorithm and state its local and global convergence properties.

### A. Notation & Preliminaries

For simplicity of presentation, we bring (1) in a more general form. All state and control variables as well as the step size are stacked into a vector $y \in \mathbb{R}^{n_y}$. The slack variables in the TOCP are stacked in a vector $s_{\mathrm{OCP}} \in \mathbb{R}^{n_s}$. We introduce additional slack variables $s_{\mathrm{NLP}} \in \mathbb{R}^{n_z}$ to bring (1) in the desired problem structure and let $s := (s_{\mathrm{OCP}}, s_{\mathrm{NLP}}) = [s_{\mathrm{OCP}}^\top, s_{\mathrm{NLP}}^\top]^\top$. Then, the decision variable $w \in \mathbb{R}^{n_w}$ is defined by $w := (y, s)$. The TOCP (1) can be reformulated in the following way

$$\min_{w \in \mathbb{R}^{n_w}} \; c^\top w \quad \text{s.t. } Cw + g(P_y w) = 0, \;\; Aw + b \leq 0. \tag{2}$$

Here, let $c \in \mathbb{R}^{n_w}$, $b \in \mathbb{R}^{n_b}$, $A \in \mathbb{R}^{n_b \times n_w}$ with full row rank $n_b$ and let $g\colon \mathbb{R}^{n_y} \to \mathbb{R}^{n_g}$ with $C \in \mathbb{R}^{n_g \times n_w}$. The projection matrix $P_y \in \mathbb{R}^{n_y \times n_w}$ is a sparse matrix, composed of rows with each a single 1-entry, that selects the non-slack variables of $w$. Additionally, we will always choose the minimal slack variables with respect to the objective function in the FSLP algorithm. That is

$$\begin{aligned} s^*(y) := \underset{s \in \mathbb{R}^{n_s + n_z}}{\operatorname{argmin}} \quad & c_s^\top s \\ \text{s.t.} \quad & A_y y + A_s s + b \leq 0, \\ & C_y y + C_s s + g(y) = 0, \end{aligned}$$

with $A = (A_y \,|\, A_s)$, $C = (C_y \,|\, C_s)$, $c = (c_y, c_s)$. From this follows that there exists a constant $\xi > 0$ such that

$$\xi^{-1} \|w_1 - w_2\|_2 \leq \|P_y(w_1 - w_2)\|_\infty \leq \xi \|w_1 - w_2\|_2 \tag{3}$$

for all $w_1, w_2 \in \{w \in \mathbb{R}^{n_w} | y \in \mathbb{R}^{n_y} : w = (y, s^*(y))\}$. We define the measure of infeasibility by

$$h(w) := \|Cw + g(P_y w)\|_\infty + \|[Aw + b]^+\|_\infty \tag{4}$$

where $[Aw + b]^+ := [\max\{A_i w + b_i, 0\}]_{i=1}^{n_b}$. Let the Lagrangian be defined by $\mathcal{L}(w, \lambda, \pi) := c^\top w + g(x)^\top \lambda + (Aw + b)^\top \pi$ with Lagrange multiplier vectors $\lambda \in \mathbb{R}^{n_g}$ and $\pi \in \mathbb{R}^{n_b}$. The KKT-conditions for (2) are given by

$$\begin{aligned} &\nabla_w \mathcal{L}(w, \lambda, \pi) = 0, \; Cw + g(P_y w) = 0, \; Aw + b \leq 0 \\ &\pi \geq 0, \quad \pi_i (A_i w + b_i) = 0, \quad \forall i = 1, \ldots, n_b. \end{aligned} \tag{5}$$

The feasible set is denoted by $\mathcal{F} := \{w \in \mathbb{R}^{n_w} \,|\, Cw + g(P_y w) = 0, \, Aw + b \leq 0\}$. In Section IV, we make use of a more restrictive definition of feasibility. Let the *zero slack feasible set* with respect to the OCP (1) be $\mathcal{F}_{\mathrm{OCP}} := \{w \in \mathbb{R}^{n_w} \,|\, w \in \mathcal{F}, \, s_{\mathrm{OCP}} = 0\}$. For a given initial guess $\hat{w}_0 \in \mathcal{F}$ we define the level set $L_0(\hat{w}_0) \subset \mathcal{F}$ by $L_0(\hat{w}_0) := \{w \in \mathbb{R}^{n_w} \,|\, w \in \mathcal{F}, \, c^\top w \leq c^\top \hat{w}_0\}$. We denote the closed ball around $\hat{w}$ with radius $\gamma > 0$ by $\mathcal{B}(\hat{w}, \gamma) := \{w \,|\, \|\hat{w} - w\|_2 \leq \gamma\}$. When it is not further specified $\|\cdot\|$ denotes the Euclidean norm.

### B. Description of the algorithm

Here, we propose the FSLP algorithm which is a special case of the algorithmic framework FP-SQP [16] that keeps all iterates feasible due to a *feasibility perturbation step*. In [16], there is no specific feasibility perturbation technique given. We introduce a novel feasibility perturbation technique in Section III. Due to the trust-region used in FP-SQP, there are no strong requirements on the Hessian approximation. Therefore, choosing a zero matrix in the SQP subproblem yields an LP and transforms FP-SQP into an FSLP algorithm while keeping its global convergence theory valid. FSLP is described in Algorithm 1.

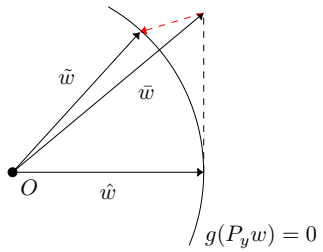



Fig. 1. Visualization of feasibility perturbation.

Let $\hat{w} \in \mathcal{F}$ the point of linearization, then the trust-region LP is defined as follows

$$\begin{aligned} \min_{w \in \mathbb{R}^{n_w}} \quad & c^\top w && \text{(6a)} \\ \text{s.t.} \quad & Cw + \nabla g(P_y \hat{w})^\top P_y (w - \hat{w}) = 0, && \text{(6b)} \\ & Aw + b \leq 0, && \text{(6c)} \\ & ||P_y(w - \hat{w})||_\infty \leq \Delta. && \text{(6d)} \end{aligned}$$

After solving (6), its solution $\bar{w} \in \mathbb{R}^{n_w}$ is projected onto the feasible set. The feasibility perturbed iterate is denoted by $\tilde{w} \in \mathbb{R}^{n_w}$. The connection of these three iterates is visualized in Fig. 1. For the global convergence theory in [16] to hold, it is required that $\tilde{w} \in \mathcal{F}$ and that the *projection ratio* is below a certain threshold, i.e., there exists a continuous monotonically increasing function $\phi\colon [0, \Delta_{\max,2}] \to [0, \frac{1}{2}]$

with $\phi(0) = 0$ such that

$$\frac{\|\bar{w} - \tilde{w}\|_2}{\|\bar{w} - \hat{w}\|_2} \leq \phi(\|\bar{w} - \hat{w}\|_2), \tag{7}$$

where $\Delta_{\max,2} > 0$ will be defined in Theorem 3. As globalization strategy, a trust-region approach is used. Since every iterate remains feasible, the merit function is chosen as the objective. The ratio of actual to predicted reduction decides upon step acceptance or rejection. It is defined by

$$\rho(\hat{w},\, \bar{w},\, \tilde{w}) = \frac{c^\top(\hat{w} - \tilde{w})}{c^\top(\hat{w} - \bar{w})}. \tag{8}$$

As termination criterion, we use the linear model of the objective function

$$m(\bar{w};\, \hat{w}) := c^\top(\bar{w} - \hat{w}).$$

If the model does not decrease through a new LP iterate $\bar{w}$, an optimal point was found and the FSLP algorithm is terminated.

**Algorithm 1** Feasible Sequential Linear Programming

**Input:** initial point $\hat{w}_0 \in \mathcal{F}$, projection matrix $P_y$, initial trust-region radius $\Delta_0 \in (0,\, \tilde{\Delta}]$, trust-region upper bound $\tilde{\Delta} \geq 1$, $\sigma \in (0, 1/4)$, $\sigma_{\text{outer}} \in (0, 10^{-5})$, $0 < \alpha_1 < 1 < \alpha_2 < \infty$, $0 < \eta_1 < \eta_2 < 1$;

**for** $k = 0,\, 1,\, 2, \ldots$ **do**
  Obtain $\bar{w}_k$ by solving (6)
  **if** $|m(\bar{w}_k,\, \hat{w}_k)| \leq \sigma_{\text{outer}}$ **then**
    **STOP**
  **end if**
  Seek $\tilde{w}_k \in \mathcal{F}$ that fulfills (7) with Algorithm 2
  **if** no such $\tilde{w}_k$ is found **then**
    $\Delta_{k+1} \leftarrow \alpha_1 ||P_y(\bar{w}_k - \hat{w}_k)||_\infty$; $\hat{w}_{k+1} \leftarrow \hat{w}_k$
  **else**
    Calculate $\rho_k$ from (8)
    **if** $\rho_k < \eta_1$ **then**
      $\Delta_{k+1} \leftarrow \alpha_1 ||P_y(\bar{w}_k - \hat{w}_k)||_\infty$
    **else if** $\rho_k > \eta_2$ and $||P_y(\bar{w}_k - \hat{w}_k)||_\infty = \Delta_k$ **then**
      $\Delta_{k+1} \leftarrow \min(\alpha_2 \Delta_k, \tilde{\Delta})$
    **else**
      $\Delta_{k+1} \leftarrow \Delta_k$
    **end if**
    **if** $\rho_k > \sigma$ **then**
      $\hat{w}_{k+1} \leftarrow \tilde{w}_k$
    **else**
      $\hat{w}_{k+1} \leftarrow \hat{w}_k$
    **end if**
  **end if**
**end for**

### C. Global convergence of FP-SQP and FSLP

The global convergence theory of FP-SQP can directly be applied to FSLP. In order to prove global convergence the following assumptions are made in [16]:

*Assumption 1:* For a given $\hat{w}_0$, the level set $L_0(\hat{w}_0)$ is bounded, and the function $g$ in (2) is twice continuously differentiable in an open neighborhood of $L_0(\hat{w}_0)$.

*Assumption 2:* For every point $\hat{w} \in L_0(\hat{w}_0)$, there are constants $\zeta,\, \hat{\Delta} > 0$ such that for all $w \in \mathcal{B}(\hat{w},\, \xi\hat{\Delta})$ we have

$$\min_{v \in \mathcal{F}} \|v - w\|_2 \leq \zeta(\|g(w)\|_2 + \|[Aw + b]^+\|_2),$$

where $\xi$ was defined in (3).

The last assumption requires that the distance of $\bar{w}$ to the closest feasible point is bounded by the constraint violation of $\bar{w}$. This ensures the existence of a feasible point $\tilde{w}$ satisfying (7) [16].

Since FSLP is a special case of FP-SQP we apply the following global convergence result:

*Theorem 1 (Global convergence of FP-SQP [16]):* Suppose that Assumptions 1, 2 hold then all limit points of Algorithm 1 either are KKT points or else fail to satisfy the Mangasarian-Fromowitz constraint qualification.

### D. Local convergence of FSLP

In the following, we show that in the case where the solution of an NLP is fully determined by the active constraints, following [10] and [11], we obtain local quadratic convergence if the iterates are not projected on the feasible set. In practice, local quadratic convergence is often obtained for projected iterates as we show in section IV. In the fully determined case, solving the NLP (2) is equivalent to finding a feasible point to the active constraints. Applying the classical Newton method to this root-finding problem yields quadratic convergence [11]. If an LP has at least one solution, then at least one solution lies in a vertex of the feasible set. In the fully determined case, the NLP solution is locally unique and lies in a vertex of the feasible set. Therefore, the unprojected iterates of FSLP converge quadratically. We adopt the convergence result of [11] to our algorithm.

*Corollary 1 (Local quadratic convergence):* Assume $z^* = (\hat{w}^*,\, \lambda^*, \pi^*) = (\hat{w}^*,\, \vartheta^*)$ is a KKT point of (2), at which LICQ and strict complementarity hold. If $\hat{w}^*$ is fully determined by the active constraints, then $\hat{w}^*$ is a local minimizer. Additionally, there exists a trust-region radius $\tilde{\Delta} > 0$ and a $\bar{k} \in \mathbb{N}$ such that for all $\hat{w}_k$ with $\|P_w(\hat{w}_k - \hat{w}^*)\| \leq \tilde{\Delta}$ and $k > \bar{k}$, then FSLP converges Q-quadratically in the primal variable $\hat{w}$, and R-quadratically in the dual variable $\vartheta$, i.e., there are constants $c_1,\, c_2 \in \mathbb{R}_+$ such that

$$\begin{aligned} &\|\hat{w}_{k+1} - \hat{w}^*\| \leq c_1 \|\hat{w}_k - \hat{w}^*\|^2 \\ \text{while} \quad &\|\vartheta_{k+1} - \vartheta^*\| \leq c_2 \|\hat{w}_k - \hat{w}^*\|. \end{aligned}$$

*Proof:* Due to the trust-region radius $\tilde{\Delta}$ all iterates $\hat{w}_k$ for $k > \bar{k}$ are in the region of local convergence around $\hat{w}^*$. Then, the proof follows from [11]. ■

## III. Inner Feasibility Iterations

In this section, we present the inner feasibility iterations to obtain a feasible step $\tilde{w} \in \mathcal{F}$ from the LP step $\bar{w}$.

This is the main algorithmic contribution of this paper. After introducing the algorithm, we propose a termination heuristic for efficient implementation and we prove local convergence of the inner iterates towards a feasible point. In particular, we show that this step fulfills the projection ratio condition (7).

### A. Description of the algorithm

The feasibility iterations are closely related to second-order corrections which were first introduced in [7] in order to avoid slow convergence in globalized SQP methods. In fact, if the algorithm is started at $\bar{w}$ the very first feasibility iteration coincides with a second-order correction. Further iterations perform higher-order corrections of the constraints. Our approach is very similar to [17], but it differs in the use of a trust-region as a means to impose local convergence in the inner iterations and global convergence in the outer iterations. Let $\hat{w} \in \mathcal{F}$ be the outer iterate and $w_l \in \mathbb{R}^{n_w}$ be an inner iterate for a given inner iterate counter $l \in \mathbb{N}$. We fix the Jacobian of $g$ at $P_y\hat{w}$, i.e., $G^\top := \nabla g(P_y\hat{w})^\top P_y$, and define

$$\delta(w_l,\, \hat{w}) := g(P_y w_l) - g(P_y\hat{w}) - G^\top(w_l - \hat{w}). \tag{9}$$

Let $\delta_l = \delta(w_l,\, \hat{w})$, then we define the parametric linear program $\mathrm{PLP}(\delta_l;\, \hat{w}, \Delta)$ as

$$\begin{aligned} \min_{w \in \mathbb{R}^{n_w}} \quad & c^\top w && \text{(10a)} \\ \text{s.t.} \quad & \delta_l + Cw + G^\top(w - \hat{w}) = 0, && \text{(10b)} \\ & Aw + b \leq 0, && \text{(10c)} \\ & ||P_y(w - \hat{w})||_\infty \leq \Delta. && \text{(10d)} \end{aligned}$$

Its solution will be denoted by $w^*_{\mathrm{PLP}}(\delta_l;\, \hat{w}, \Delta)$. We note that for $w_l \leftarrow \hat{w}$, we obtain LP (6) and for $w_l \leftarrow \bar{w}$, we obtain a standard second-order correction problem as defined in [5] or [12]. The algorithm is based on the iterative solution of PLPs. In every iteration the nonlinear constraints are re-evaluated at $w_l$ in the term $\delta_l$ and another PLP is solved. The solution of this problem is denoted by $w_{l+1}$ and the procedure is repeated. We will show that for $l \to \infty$ the limit of the sequence $\{w_l\}_{l \in \mathbb{N}}$ is $\tilde{w} \in \mathcal{F}$. Algorithm 2 presents this feasibility improvement strategy in detail. The feasibility iterations are repeated until convergence towards a feasible point of (2) is achieved. If the iterates are diverging the algorithm is terminated, the inner algorithm returns to the outer algorithm Algorithm 1, and the trust-region radius is decreased. This strategy is presented in detail in the following subsection III-B.

Particularly, in every iteration of Algorithm 2, the constraints are re-evaluated. This is advantageous in applications where the evaluation of first- and second-order derivatives is expensive. Assuming that the solutions of the PLPs do not differ much, the solution of the feasibility problem can be computed quite cheaply by using an active set solver [17].

The subsequent sections focus on the termination heuristic for Algorithm 2 and on its local convergence behavior. We state that the limit of the feasibility iterations is indeed a feasible point. Afterwards, we prove local convergence and the satisfaction of the projection ratio condition (7).

**Algorithm 2** Inner Feasibility Iterations

**Input:** $\hat{w} \in \mathcal{F}$, fixed Jacobian $G^\top = \nabla g(\hat{w})^\top P_y$. Let $\bar{w}$ be the solution of (6) at $\hat{w}$. $n_{\mathrm{watch}} \in \mathbb{N}$, $\kappa_{\mathrm{watch}} < 1$, $\sigma_{\mathrm{inner}} \in (0,\, 10^{-5})$;
**Output:** $\tilde{w}$

```
w_0 ← w̄
for l = 0, 1, 2, ... do
    if h(w_l) ≤ σ_inner and ‖w̄ − w_l‖/‖w̄ − ŵ‖ < 1/2 then
        w̃ ← w_l
        STOP
    end if
    Solve PLP(δ_l, ŵ, Δ)
    if iterates w_l are diverging according to subsection III-B then
        STOP
    end if
    w_{l+1} ← w*_PLP(δ_l; ŵ, Δ)
end for
```

### B. Termination heuristic

For an efficient FSLP algorithm, it is crucial to stop the inner feasibility iterations early if the iterates are not converging. We recall that the optimal solution must be feasible, i.e., $w^* \in \mathcal{F}$ and that the projection ratio condition (7) must be satisfied. The termination heuristic consists of the following steps.

As shown in Section IV, the projection ratio does not change much for converging feasibility iterations. Therefore, we observe the projection ratio for all inner iterates, i.e., $\|\bar{w} - w_l\|/\|\bar{w} - \hat{w}\|$. If it is higher than $1.0$, the inner iterations are aborted and the trust-region radius is decreased.

Additionally, we estimate the contraction rate $\kappa$ of the algorithm with the following formula

$$\kappa_l = \frac{\|w_{l+1} - w_l\|}{\|w_l - w_{l-1}\|}. \tag{11}$$

The convergence of the inner iterations is observed through a watchdog strategy. After $n_{\mathrm{watch}}$ inner iterations, the contraction rate for these $n_{\mathrm{watch}}$ steps is checked, if it is not below a threshold $\kappa_{\mathrm{watch}}$ the iterations are aborted. The projection ratio is also checked. If it is not below $0.5$, the algorithm also terminates.

If the feasibility measure $h(w_l)$ is below a feasibility tolerance $\sigma_{\mathrm{inner}}$, the algorithm terminates. If the maximum number of iterations $n_{\max}$ are reached without achieving convergence, the inner iterations are aborted and the trust-region radius is decreased.

### C. Limit of feasibility iterations

For completeness of presentation, we state a result about the limit of the feasibility iterations, first derived in [2]:

*Lemma 1 (Limit of feasibility improvement):* Assume that for fixed $\hat{w}$, the sequence of feasibility iterates $\{w_l\}_{l\in\mathbb{N}}$ converges towards a $w^*$, and let $\pi^*$ be the corresponding Lagrange multipliers of (10b) in $w^*$. Then $w^*$ and $\pi^*$ belong to a KKT point of the problem

$$\min_{w\in\mathbb{R}^{n_w}} \quad (c + (G - P_y^\top \nabla g(P_y w^*))\pi^*)^\top w \tag{12a}$$

$$\text{s.t.} \quad Cw + g(P_y w) = 0, \quad Aw + b \leq 0, \tag{12b}$$

$$\|P_y(w - \hat{w})\|_\infty \leq \Delta. \tag{12c}$$

*Proof:* The proof follows by comparison of the KKT-conditions of (10) and (12) in the same fashion as in [2]. ∎

### D. Local convergence of feasibility iterations

Next, we provide a proof of the local convergence of the feasibility iterations. In order to prove local contraction, we make use of the following assumption which is in the context of generalized equations referred to as strong regularity [13].

*Assumption 3 (Strong regularity):* For all $\hat{w} \in \mathcal{F}$ exist $L_1, L_2, \bar{\Delta} > 0$ such that for all $\Delta \leq \bar{\Delta}$ and for all $\delta_1, \delta_2 \in \mathcal{B}(0, L_2\,\Delta)$ it holds that $\|w^*_{\mathrm{PLP}}(\delta_1; \hat{w}, \Delta) - w^*_{\mathrm{PLP}}(\delta_2; \hat{w}, \Delta)\| \leq L_1\|\delta_1 - \delta_2\|$.

We note that $w_{l+1} = w^*_{\mathrm{PLP}}(\delta_l; \hat{w}, \Delta)$. Then, we can state the local contraction result:

*Theorem 2 (Local linear convergence proportional to $\Delta$):* Let $\hat{w} \in \mathcal{F}$ and let Assumption 3 hold, then there exist $\Delta_{\max,1}, L > 0$ such that for all $\Delta \leq \Delta_{\max,1}$ the iterates $\{w_l\}_{l\in\mathbb{N}}$ of Algorithm 2 converge towards a point $w^* \in \mathcal{F}$ and the contraction rate is proportional to $\Delta$, i.e., $\|w_{l+1} - w^*\| \leq L\Delta\|w_l - w^*\|$.

*Proof:* Let $\Delta \leq \bar{\Delta}$. Using the definition of the solution of $\mathrm{PLP}(\delta; \hat{w}, \Delta)$ and Assumption 3 yields $\|w_{l+1} - w^*\| = \|w^*_{\mathrm{PLP}}(\delta_l; \hat{w}, \Delta) - w^*_{\mathrm{PLP}}(\delta^*; \hat{w}, \Delta)\| \leq L_1\|\delta_l - \delta^*\|$. For simplicity of notation we define $\tilde{G}\colon w \mapsto P_y \nabla g(P_y w)$ and observe $\tilde{G}(\hat{w}) = G$. We note that by applying the fundamental theorem of calculus and using Lipschitz continuity of $\tilde{G}$ with Lipschitz constant $\tilde{L} > 0$ we obtain

$$\begin{aligned}
&\|\delta_l - \delta^*\| = \|g(P_y w_l) - g(P_y w^*) - G^\top(w_l - w^*)\| \\
&\leq \int_0^1 \|(\tilde{G}(w^* + t(w_l - w^*)) - G)^\top\|\,\|w_l - w^*\|\mathrm{d}t \\
&\leq \int_0^1 \tilde{L}\|w^* + t(w_l - w^*) - \hat{w}\|\mathrm{d}t\,\|w_l - w^*\| \\
&\leq \tilde{L}\,\Delta\,\|w_l - w^*\|.
\end{aligned}$$

Defining $L = L_1\,\tilde{L}$ yields

$$\|w_{l+1} - w^*\| \leq L\Delta\,\|w_l - w^*\|.$$

From this follows $\Delta_{\max,1} < \frac{1}{L}$. ∎

### E. Projection ratio

In this section, we provide a proof that condition (7) about the projection ratio holds for the optimal solution of the inner feasibility iterations. We note that $w_0 = \hat{w}$, $w_1 = \bar{w}$, and $w_l \to \tilde{w}$, i.e., $\hat{w} = w^*$. For an illustration of the three different iterates, we refer to Fig. 1.

*Theorem 3 (Projection ratio):* Let Assumption 3 hold. Moreover, let $\tilde{w}$ be the limit of Algorithm 2, $\bar{w}$ be the solution of (6), and let $\hat{w}$ be the point of linearization, then there exists $\Delta_{\max,2}$ such that a function $\phi\colon \mathbb{R} \to [0,\, 1/2]$ satisfying condition (7) exists.

*Proof:* We interpret the solution of (6) $\bar{w}$ as the first iterate of Algorithm 2, i.e., $w_0 = \hat{w}$ and $w_1 = \bar{w}$. We also note that $\tilde{w} = w^*$. In the same way, as done in Theorem 2, we obtain the following estimate

$$\begin{aligned}
&\|\delta_{l+1} - \delta_l\| \\
&\quad\leq \int_0^1 \tilde{L}\|w_l + t(w_{l+1} - w_l) - \hat{w}\|\mathrm{d}t\,\|w_{l+1} - w_l\| \\
&\quad\leq \tilde{L}\int_0^1 \|w_l - w_0\| + t\|w_{l+1} - w_l\|\mathrm{d}t\,\|w_{l+1} - w_l\| \\
&\quad\leq \tilde{L}(\|w_l - w_0\| + \frac{1}{2}\|w_{l+1} - w_l\|)\,\|w_{l+1} - w_l\|.
\end{aligned}$$

From this follows

$$\begin{aligned}
&\|w_{l+2} - w_{l+1}\| \\
&\qquad\leq L(\|w_l - w_0\| + \frac{1}{2}\|w_{l+1} - w_l\|)\,\|w_{l+1} - w_l\|.
\end{aligned}$$

For simplicity, we define

$$\kappa_l := L(\|w_l - w_0\| + \frac{1}{2}\|w_{l+1} - w_l\|),$$

which gives

$$\|w_{l+2} - w_{l+1}\| \leq \kappa_l\,\|w_{l+1} - w_l\|.$$

We also define $\kappa_{\max} := 2L\Delta$. For the case $l = 0$ we get $\kappa_0 = \frac{1}{2}L\|w_1 - w_0\|$ such that

$$\|w_2 - w_1\| \leq \tau\|w_1 - w_0\|^2$$

where $\tau := \frac{1}{2}L$. We proceed by a telescoping sum argument. It holds

$$\sum_{l=0}^{K} w_{l+2} - w_{l+1} = w_{K+2} - w_1$$

and

$$\|w_{l+2} - w_{l+1}\| \leq (\kappa_{\max})^l\|w_2 - w_1\|.$$

Then,

$$\begin{aligned}
\|w_{K+2} - w_1\| &\leq \sum_{l=0}^{K}\|w_{l+2} - w_{l+1}\| \\
&\leq \sum_{l=0}^{K}(\kappa_{\max})^l\|w_2 - w_1\| \\
&\leq \left(\sum_{l=0}^{K}(\kappa_{\max})^l\right)\tau\|w_1 - w_0\|^2 \\
&\leq \frac{\tau}{1 - \kappa_{\max}}\|w_1 - w_0\|^2.
\end{aligned}$$

For $K \to \infty$ and with the original notation of (7), we get

$$\|\tilde{w} - \bar{w}\| \leq \frac{\tau}{1 - \kappa_{\max}} \|\bar{w} - \hat{w}\|^2.$$

From this we define $\phi$ as

$$\phi(\|\bar{w} - \hat{w}\|) = \frac{0.5L\|\bar{w} - \hat{w}\|}{1 - 2L^2\Delta},$$

where $\phi$ is always smaller than $\frac{0.5L\Delta}{1-2L^2\Delta}$. If we choose $\Delta_{\max,2} \leq \frac{1}{(1+2L)L}$, then $\phi(\cdot) \leq 1/2$. ■

## IV. Numerical Example

In this section, we present simulation results for the FSLP algorithm. We first demonstrate the quadratic convergence behavior of FSLP on an illustrative example of a fully determined system. Subsequently, we test FSLP on a time-optimal point-to-point motion problem of an overhead crane. We show, that the theoretically derived contraction properties of the inner feasibility iterations can be empirically verified on the test example and that the desired projection ratio property holds. Finally, we compare FSLP with the state-of-the-art NLP solver `Ipopt` [15] on 100 different problem instances. `Ipopt` uses the linear solver `ma57` from the HSL library [9].

### A. Implementation

We use the Python interface of the open source software `CasADi` [1] to model the optimization problem and for a prototypical implementation of the FSLP solver. As LP solver, we use the dual simplex algorithm of `CPLEX` version 12.8 [6] which can be called from `CasADi`. The implementation is open-source and can be found at `https://github.com/david0oo/fslp`. As parameters in Algorithm 1 we chose: $\Delta_0 = 1$, $\alpha_1 = \eta_1 = 0.25$, $\alpha_2 = 2$, $\eta_2 = 0.75$, $\sigma = 10^{-8}$, $\sigma_{\text{outer}} = 10^{-8}$. In Algorithm 2, we chose $n_{\text{watch}} = 5$, $\kappa_{\text{watch}} = 0.3$ and $\sigma_{\text{inner}} = 10^{-7}$. The simulations were carried out on a Intel Core i7-10810U CPU.

### B. Illustrative example of quadratic convergence

In this subsection, we illustrate the quadratic convergence of the outer iterations of FSLP in the case of a fully determined system. Therefore, we define the following parametric optimization problem

$$\min_{w \in \mathbb{R}^2} w_2 \quad \text{s.t.} \quad w_2 \geq w_1^2,\ w_2 \geq 0.1w_1 + \varepsilon \qquad (13)$$

with $\varepsilon \in \mathbb{R}$. It is easy to see that the optimal solution of (13) for $\varepsilon = -0.06$ is $w^* = (0,\ 0)$ and for $\varepsilon = 0.06$, it is $w^* = (-0.2,\ 0.04)$. For $\varepsilon = 0.06$, the solution is fully determined by both constraints and lies in a vertex of the linearized constraints at the solution. For $\varepsilon = -0.06$, the solution is not fully determined and does not lie in a vertex of the linearized constraints at the solution. In Fig. 2, we see that in the case of the fully determined NLP, Algorithm 1 converges quadratically towards the optimal solution. In the case of the not fully determined problem, the algorithm converges linearly towards the optimal solution, but at a certain accuracy, it is not possible to improve accuracy since all iterates of FSLP lie in a vertex of the boundary of the trust-region. If a step is accepted, the trust-region is increased which results in an overshoot of the steps such that the accuracy decreases again. The convergence results are shown in Fig. 2. The FSLP algorithm was initialized at $\hat{w}_0 = (2,\ 10)$ for both problems.

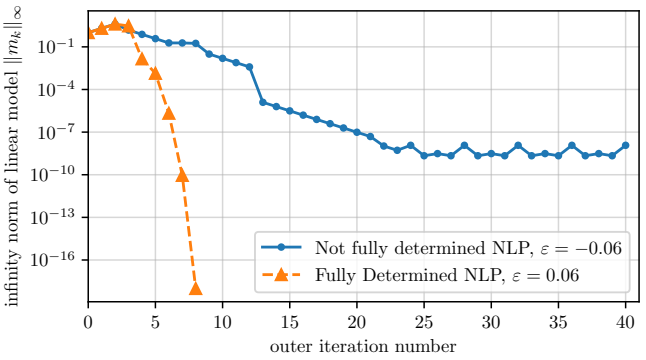



Fig. 2. Comparison of the convergence of FSLP on a fully determined NLP and a not fully determined NLP.

### C. Point-to-point motion of an overhead crane

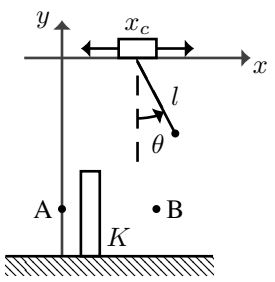



Fig. 3. Schematic illustration of the overhead crane.

As main test example, we present the time-optimal point-to-point motion of an overhead crane as shown in Fig. 3. The crane can move its position $x_c\,[\mathrm{m}]$ on a rail. The length of the crane hoist is denoted by $l\,[\mathrm{m}]$ and its angle with respect to the rail by $\theta\,[\mathrm{rad}]$. The payload of the crane has the position $p = (x_c + l\sin(\theta), -l\cos(\theta))$. The control inputs are the acceleration of the cart $\ddot{x}_c\,[\mathrm{m/s^2}]$ and the winding acceleration $\ddot{l}\,[\mathrm{m/s^2}]$ of the hoist. Our goal is to move the payload from point of rest A to point of rest B in minimal time $t\,[s]$. The system dynamics are given by

$$l\ddot{\theta} = \cos(\theta)\ddot{x}_c - 2\dot{l}\dot{\theta} - g\sin(\theta),$$

where $g = 9.81\mathrm{kg\,m/s^2}$ is the gravitational acceleration. The rectangular obstacle $K \subset \mathbb{R}^2$ in Fig. 3 is represented by its vertices $v_1, v_2, v_3, v_4 \in \mathbb{R}^2$. In order to model obstacle avoidance, we apply the separating hyperplane theorem [4]. We introduce the radius of the load $r_{\text{load}}\,[\mathrm{m}]$ and for every time instant $k = 1, \ldots, N$, we introduce hyperplane variables $u_k^{\text{h}} \in \mathbb{R}^2$, $u_k^{\text{c}} \in \mathbb{R}$ that separate the payload from

the obstacle. The constraints are of the following form:

$$p_k^\top u_k^{\mathrm{h}} - u_k^{\mathrm{c}} \leq -r_{\mathrm{load}},$$
$$v_i^\top u_k^{\mathrm{h}} - u_k^{\mathrm{c}} \geq 0 \quad \forall\, i = 1, \ldots, 4,$$
$$\|u_k^h\|_\infty \leq 1, \quad \|u_k^c\|_\infty \leq 1$$

for all $k = 1, \ldots, N$. Algorithm 1 needs to be initialized with a feasible trajectory. Since it is difficult to provide an initial guess satisfying the start and end position constraints, we have to relax both conditions, i.e,

$$-s_0 \leq x_0 - \bar{x}_0 \leq s_0, \quad -s_f \leq x_N - \bar{x}_f \leq s_f,$$

with $s_0,\, s_f \in \mathbb{R}^{n_x}_{\geq 0}$. The slack variables are penalized by the factor $10^5$ in the objective. Furthermore, $N = 20$ and $r_{\mathrm{load}} = 0.08\,\mathrm{m}$. Table I shows the parameters for the box constraints on the variables. The system dynamics were integrated with a Runge Kutta 4 scheme with 20 internal steps per multiple shooting interval. Altogether, we obtain a TOCP formulation as in (1).

In the simulations, we initialize with the trajectory of iteration zero as in Fig. 4. These can be created by initializing with initial time 2.5 s, i.e., $h = 0.125\,\mathrm{s}$ and forward simulating the system dynamics with the constant control $(0\,\mathrm{m/s^2},\, 0.1\,\mathrm{m/s^2})$ from the initial payload position $(0\,\mathrm{m},\, -0.6\,\mathrm{m})$. The trajectories of some iterates of FSLP are shown in Fig. 4. FSLP terminates after eleven iterations at the optimal solution. After three iterations, the starting condition is already satisfied, after six iterations the start and end conditions are satisfied, i.e., a zero slack feasible solution is found, and after nine iterations, the suboptimal iterate and the optimal solution are barely distinguishable. In the following, we always mean a zero slack feasible solution, if a feasible solution is mentioned.

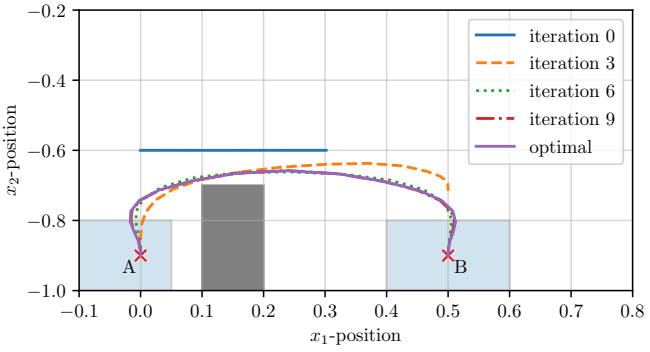



Fig. 4. Payload trajectories of the iterates of FSLP on the overhead crane problem.

*1) Local contraction and projection ratio:* At first, we investigate the local convergence behavior of the inner iterations for different trust-region radii. As an example problem, we use the first outer iteration of FSLP initialized with the trajectory of iteration zero as shown in Fig. 4. In Fig. 5, we observe the contraction and projection ratio for different trust-region radii. We see that the iterates converge locally and fulfill (7) for trust-region radii below a certain threshold which confirms the theoretical results of subsections III-D and III-E.

TABLE I
BOUNDARIES OF VARIABLES.

| Lower Bound | Description | Upper Bound |
|---|---|---|
| $-0.1\,\mathrm{m}$ | $x_c$ | $0.6\,\mathrm{m}$ |
| $-0.4\,\mathrm{m/s}$ | $\dot{x}_c$ | $0.4\,\mathrm{m/s}$ |
| $10^{-2}\,\mathrm{m}$ | $l$ | $2\,\mathrm{m}$ |
| $-0.25\,\mathrm{m/s}$ | $\dot{l}$ | $0.25\,\mathrm{m/s}$ |
| $-0.75\,\mathrm{rad}$ | $\theta$ | $0.75\,\mathrm{rad}$ |
| $-5\,\mathrm{m/s^2}$ | $\ddot{x}_c$ | $5\,\mathrm{m/s^2}$ |
| $-5\,\mathrm{m/s^2}$ | $\dot{l}$ | $5\,\mathrm{m/s^2}$ |
| 0 | $s_0,\, s_N$ | $+\infty$ |

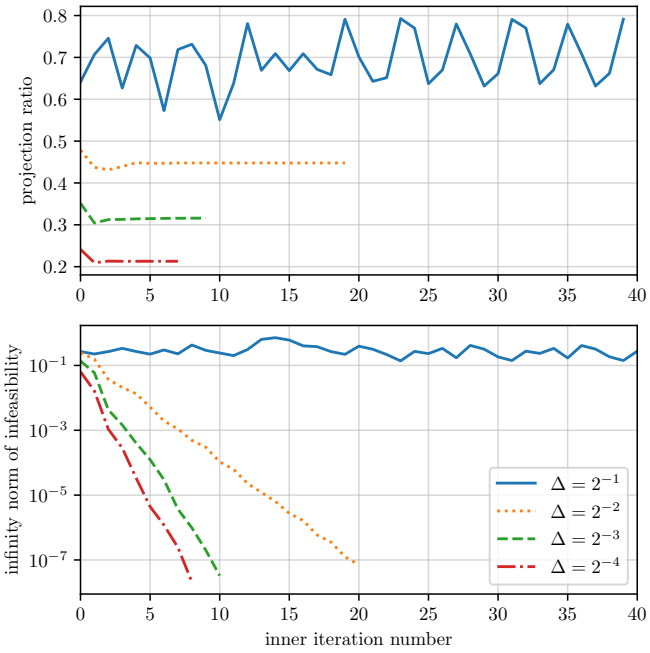



Fig. 5. Comparison of local convergence and projection ratio of feasibility iterations for different trust-region radii. For all trust-region radii below a certain value between $2^{-1}$ and $2^{-2}$ we obtain linear convergence and satisfaction of condition (7).

*2) Comparison to Ipopt:* In the next experiment, we take ten random perturbations of starting position A and ten perturbations of the end position B resulting in 100 different optimization problems. The perturbations are chosen to be uniformly distributed over the blue areas in Fig. 4. The performance of FSLP and `Ipopt` is presented in Fig. 6. `Ipopt` - feasible solution denotes the value at the iteration from which all subsequent iterations stay below the given feasibility tolerance. We can see that the SLP algorithm needs in almost all cases fewer outer iterations than `Ipopt` to solve the TOCP problems. In contrast, FSLP needs more constraint evaluations due to the inner feasibility iterations. Since FSLP needs fewer outer iterations than `Ipopt`, FSLP needs fewer evaluations of derivative information and especially no evaluations of the Hessian. Stopping FSLP at a suboptimal point satisfying start and end position constraints reduces the constraint evaluations and outer iterations.

Additionally, we timed the wall time of the Python FSLP prototype against `Ipopt` for solving an instance of the 100 different test problems. One big difference between FSLP and `Ipopt` is the use of derivative information, therefore we investigate the wall times on the 100 different problems

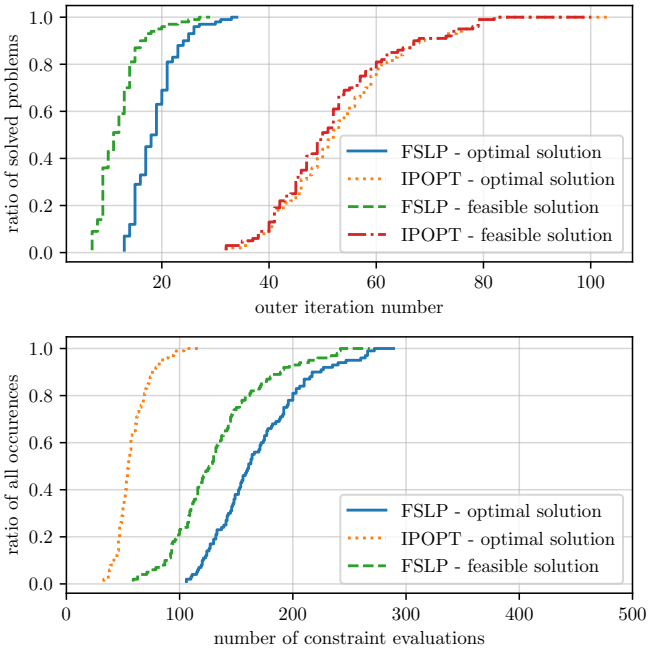



Fig. 6. Comparison of outer iterations and constraint evaluations of FSLP and IPOPT on 100 different overhead crane problems.

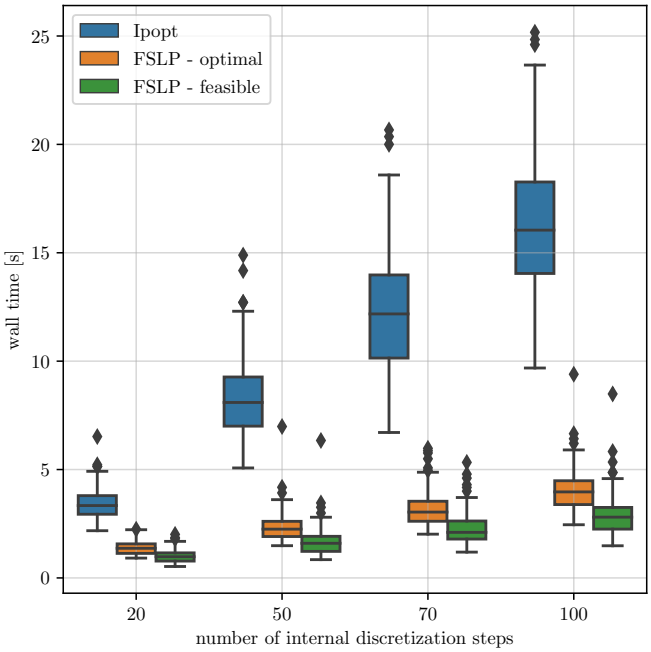



Fig. 7. Comparison of wall times of FSLP and IPOPT on 100 different overhead crane problems for different number of discretization steps in the Runge Kutta scheme.

for different numbers of internal discretization steps in the Runge Kutta scheme. The more steps are used, the more expensive it gets to evaluate the constraints and especially the derivatives. The results are illustrated in a boxplot in Fig. 7. We see that the run time of the prototypical implementation scales better than the run time of `Ipopt` for increasing numbers of discretization steps. These are promising results for future research and for an efficient implementation for real-time optimization by exploiting problem structure.

At last, we remark that the initial time was chosen close to the optimal time of the original overhead crane problem in Subsection IV-C. If the initial time is chosen larger, then a zero-slack feasible solution can be reached faster due to more freedom in feasible trajectories, but the run time to find an optimal solution will in general increase.

## V. Conclusion

In this paper, we proposed a novel globally convergent feasible sequential linear programming algorithm for time-optimal control problems. In the case of a fully determined system, we even obtain quadratic local convergence. We show that the algorithm can maintain feasible iterates also for problems with nonlinear constraints. In a numerical case study, the performance of the algorithm and its potential to stop iterations early was demonstrated.

## References


[1] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[2] H. G. Bock, M. Diehl, E. A. Kostina, and J. P. Schlöder. Constrained optimal feedback control of systems governed by large differential algebraic equations. In *Real-Time and Online PDE-Constrained Optimization*, pages 3–22. SIAM, 2007.

[3] H. G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the IFAC World Congress*, pages 242–247. Pergamon Press, 1984.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. University Press, Cambridge, 2004.

[5] A.R. Conn, N. Gould, and P.L. Toint. *Trust-Region Methods*. MPS/SIAM Series on Optimization. SIAM, Philadelphia, USA, 2000.

[6] IBM ILOG Cplex. V12.8: User's manual for cplex. *International Business Machines Corporation*, 2017.

[7] R. Fletcher. Second order corrections for non-differentiable optimization. In G. Alistair Watson, editor, *Numerical Analysis*, pages 85–114, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[8] Ro E Griffith and RA Stewart. A nonlinear programming technique for the optimization of continuous processing systems. *Management science*, 7(4):379–392, 1961.

[9] HSL. A collection of Fortran codes for large scale scientific computation., 2011.

[10] Taedong Kim and Stephen J. Wright. An s l1 lp-active set approach for feasibility restoration in power systems. *Optimization and Engineering*, 17:385–419, 6 2016.

[11] Florian Messerer, Katrin Baumgärtner, and Moritz Diehl. Survey of sequential convex programming and generalized Gauss-Newton methods. *ESAIM: Proceedings and Surveys*, 71:64–88, 2021.

[12] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.

[13] Stephen M. Robinson. Strongly regular generalized equations. *Mathematics of Operations Research*, 5:43–62, 2 1980.

[14] M.J. Tenny, S.J. Wright, and J.B. Rawlings. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Computational Optimization and Applications*, 28:87–121, 2004.

[15] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[16] Stephen J. Wright and Matthew J. Tenny. A feasible trust-region sequential quadratic programming algorithm. *SIAM Journal on Optimization*, 14:1074–1105, 1 2004.

[17] Andrea Zanelli. *Inexact methods for nonlinear model predictive control: stability, applications, and software*. PhD thesis, University of Freiburg, 2021.