$\mathcal{N}$IPM-HLSP: An Efficient Interior-Point Method for Hierarchical Least-Squares Programs

Kai Pfeiffer[1*], Adrien Escande[2] and Ludovic Righetti[3]

[1*]School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore.
[2]Inria Center of Grenoble Alpes University, Montbonnot, France.
[3]Tandon School of Engineering, New York University, New York, USA.

*Corresponding author(s). E-mail(s): kaipfeifferrobotics@gmail.com;

Abstract

Hierarchical least-squares programs with linear constraints (HLSP) are a type of optimization problem very common in robotics. Each priority level contains an objective in least-squares form which is subject to the linear constraints of the higher priority levels. Active-set methods are a popular choice for solving them. However, they can perform poorly in terms of computational time if there are large changes of the active set. We therefore propose a computationally efficient primal-dual interior-point method (IPM) for dense HLSP's which is able to maintain constant numbers of solver iterations in these situations. We base our IPM on the computationally efficient nullspace method as it requires only a single matrix factorization per solver iteration instead of two as it is the case for other IPM formulations. We show that the resulting normal equations can be expressed in least-squares form. This avoids the formation of the quadratic Lagrangian Hessian and can possibly maintain high levels of sparsity. Our solver reliably solves ill-posed instantaneous hierarchical robot control problems without exhibiting the large variations in computation time seen in active-set methods.

Keywords: Numerical optimization; real time robot control; hierarchical least-squares programming; nullspace method; lexicographical optimization; multi objective optimization;

1 Introduction

1.1 Context and contribution

Hierarchical or lexicographic multi-objective optimization (LMOO) is the prioritized separation of constraints and objectives over any number of priority levels p. In its most generic form, LMOO can be written as (lexmin.: lexicographically minimize)

$$\begin{aligned} \underset{u}{\text{lexmin.}} \quad & f_{\mathbb{E}_1}(u), \dots, f_{\mathbb{E}_p}(u) \\ \text{s.t} \quad & f_{\mathbb{I}_{\cup p}}(u) \geq 0 \end{aligned} \tag{LMOO}$$

$u \in \mathbb{R}^n$ is a variable vector. Each (possibly) non-linear *objective* $f_{\mathbb{E}_l} \in \mathbb{R}^{m_{\mathbb{E}_l}}$, which represent the set of equality constraints $\mathbb{E}_l$ of level l ($|\mathbb{E}_l| = m_{\mathbb{E}_l}$), needs to be minimized. This comes to the extent of not influencing the optimality of the infinitely more important *constraints* $f_{\mathbb{E}_{\cup l-1}} \in \mathbb{R}^{m_{\mathbb{E}_{\cup l-1}}}$ of levels 1 to $l-1$. $\mathbb{E}_{\cup l-1}$ represents the set union $\mathbb{E}_{\cup l-1} := \bigcup_{i=1}^{l-1} \mathbb{E}_i = \mathbb{E}_1 \cup \dots \cup \mathbb{E}_{l-1}$ of the sets of equality constraints of levels 1 to $l-1$. At the same time, inequality constraints $f_{\mathbb{I}_{\cup p}} \in \mathbb{R}^{m_{\mathbb{I}_{\cup p}}}$ shall not be violated.

To date, no method has been proposed to efficently solve LMOO as is. Early beginnings of LMOO have been reported for example in Sherali and Soyster (1983) in the context of lexicographic linear programming. The authors propose a set of appropriate weights to solve the prioritized multi-objectives as an equivalent weighted one. A solution in a variable reducing fashion has been proposed in Holder (2006). In this case the problem dimensions decrease as the solver progresses through the lexicographical program. This is one of the fundamental ideas in efficient hierarchical programming. Extensions to non-linear objective functions based on optimal Pareto front and for lexicographic programs and non-smooth functions can be found in Evtushenko and Posypkin (2014), Lai et al (2021) and Ansary (2023), respectively. A solution can also be obtained by evolutionary algorithms with great efficiency in identifying trade-off solutions. An evaluation on the scheduling problem for satellite communication is given in Petelin et al (2021).

A specific form of LMOO is hierarchical linear least-squares programming (HLSP), which is the main focus of this work. Here all constraints are linear and the objectives are in least-squares form. HLSP's have seen a sharp rise in popularity in the robotics community over the recent years. Different works tackled the incorporation of infeasible inequality constraints on any priority level (Kanoun et al, 2009), or proposed very efficient solvers for these types of problems (Escande et al, 2014). The aforementioned HLSP solver (Escande et al, 2014) enables high frequency non-linear whole-body robot control when used within a sequential hierarchical least-squares programming (S-HLSP)

solver to solve non-linear hierarchical least-squares programs (NL-HLSP). S-HLSP with local convergence properties for example based on a real-time capable trust-region adaptation method (Pfeiffer et al, 2018) has been applied in many robot control works (Herzog et al (2014) and references therein). In contrast, a globally converging method for non-linear least-squares programming based on branch-and-bound has been proposed in Amaran and Sahinidis (2012). However, this method is applicable only to unconstrained problems.

In this work we propose an efficient, dense HLSP solver based on the interior point method (IPM). This has numerical advantages with respect to active-set method (ASM) based solvers like Escande et al (2014). The *active-set method* separates inequality constraints into *active* constraints, which are violated, and *inactive* constraints, which are satisfied. Violated constraints are commonly referred to as *infeasible*. In ill-posed robot control scenarios our IPM based solver maintains constant number of iterations and computation times as opposed to the ASM. This is favorable in real-time robot control where only limited computational resources are available and a control loop has a fixed time budget at each iteration.

The solver's C++ code based on the Eigen library (Guennebaud et al, 2010) is available at https://www.github.com/pfeiffer-kai/NIPM-HLSP. The nomenclature and variable naming used hereafter is listed below.

Nomenclature

l	Current priority level
l^*	Virtual priority level
p	Overall number of priority levels, excluding the trust region constraint on $l = 0$
n	Number of variables
r	Rank of matrix
n_r	Number of remaining variables after nullspace projections
m	Number of constraints
$x \in \mathbb{R}^n$	Primal of HLSP
$\Delta x \in \mathbb{R}^n$	Primal Newton step of HLSP
Δz	Primal nullspace step
$f(u) \in \mathbb{R}^m$	Non-linear constraint function of variable vector $u \in \mathbb{R}^n$
$\mathbb{E}_l$	Set of $m_{\mathbb{E}}$ equality constraints (eq.) of level l
$\mathbb{I}_l$	Set of $m_{\mathbb{I}}$ inequality constraints (eq.) of level l
$\mathcal{I}_l$	Set of $m_{\mathcal{I}}$ inactive inequality constraints (ineq.) of level l
$\mathcal{A}_l$	Set of $m_{\mathcal{A}}$ active equality and inequality constraints of level l
$\mathbb{E}_{\cup l}$ (or $\mathcal{E}_{\cup l}$)	Set union $\mathbb{E}_{\cup l} := \bigcup_{i=1}^{l} \mathbb{E}_i = \mathbb{E}_1 \cup \cdots \cup \mathbb{E}_l$ with $m_{\mathbb{E}_{\cup l}}$ constraints
$A_{\mathbb{E}} \in \mathbb{R}^{m_{\mathbb{E}} \times n}$	Matrix representing a set $\mathbb{E}$ of $m_{\mathbb{E}}$ linear constraints
$b_{\mathbb{E}} \in \mathbb{R}^{m_{\mathbb{E}}}$	Vector representing a set $\mathbb{E}$ of $m_{\mathbb{E}}$ linear constraints
$\mathcal{N}(A_{\mathcal{A}_l})$	Operator to compute the nullspace basis $Z_{\mathcal{A}_l}$ and the rank r of a matrix $A_{\mathcal{A}_l}$

$Z_{\mathcal{A}_l} \in \mathbb{R}^{n \times n_r}$ Nullspace basis of matrix $A_{\mathcal{A}_l} \in \mathbb{R}^{m_{\mathcal{A}_l} \times n}$ with rank r, $n_r = n - r$ and $A_{\mathcal{A}_l} Z_{\mathcal{A}_l} = 0$

$N_{\mathcal{A}_{\cup l}} \in \mathbb{R}^{n \times n_r}$ Accumulated nullspace basis $N_{\mathcal{A}_{\cup l}} = Z_{\mathcal{A}_1} \dots Z_{\mathcal{A}_l}$

$\tilde{M} \in \mathbb{R}^{m \times n_r}$ Matrix $\tilde{M} = MN$ projected into the nullspace basis $N \in \mathbb{R}^{n \times n_r}$ of a matrix $A \in \mathbb{R}^{m \times n}$ of rank r; $n_r = n - r$ (variable elimination)

$v \in \mathbb{R}^m$ Slack variable

$V \in \mathbb{R}^{m,m}$ Diagonal matrix equivalent $V = \mathrm{diag}(v)$ of vector $v \in \mathbb{R}^m$

$v^* \in \mathbb{R}^m$ Optimal slack variable

$\lambda \in \mathbb{R}^m$ Lagrange multiplier

$\mathcal{L}$ Lagrangian

K Gradient of Lagrangian $K \coloneqq \nabla \mathcal{L}$

ι Newton iteration or active-set iteration of HLSP solver

μ Duality measure

σ Centering parameter

ρ Trust region radius

ξ Activation threshold of inequality constraints

ϵ Convergence threshold for Newton's method

$e \in \mathbb{R}^m$ Vector of ones

$a \odot b$ Element-wise multiplication between two vectors a and b

1.2 Hierarchical least-squares programming

A HLSP is a sub-form of LMOO with linear constraints and least-squares objectives as follows

$$\begin{aligned} \underset{x, v_{\mathbb{E}_{\cup p}}, v_{\mathbb{I}_{\cup p}}}{\text{lexmin.}} \quad & \|v_{\mathbb{E}_1}\|^2 + \|v_{\mathbb{I}_1}\|^2, \dots, \|v_{\mathbb{E}_p}\|^2 + \|v_{\mathbb{I}_p}\|^2 \\ \text{s.t} \quad & A_{\mathbb{E}_{\cup p}} x - b_{\mathbb{E}_{\cup p}} = v_{\mathbb{E}_{\cup p}} \\ & A_{\mathbb{I}_{\cup p}} x - b_{\mathbb{I}_{\cup p}} \geq v_{\mathbb{I}_{\cup p}} \end{aligned} \tag{1}$$

The variable vector $x \in \mathbb{R}^n$ consists of n variables. The linear constraint sets $\mathbb{E}_l$ and $\mathbb{I}_l$ of each level $l = 1, \dots, p$ are represented by the constraint matrices and vectors $A_{\mathbb{E}_l} \in \mathbb{R}^{m_{\mathbb{E}_l}}$ and $b_{\mathbb{E}_l} \in \mathbb{R}^{m_{\mathbb{E}_l}}$, respectively. The slack variables $v_{\mathbb{E}_l} \in \mathbb{R}^{m_{\mathbb{E}_l}}$ and $v_{\mathbb{I}_l} \in \mathbb{R}^{m_{\mathbb{I}_l}}$ relax the equality and inequality constraints $\mathbb{E}_l$ and $\mathbb{I}_l$ in case of infeasibility. This relaxation allows handling of infeasible inequality constraints on any priority level (Kanoun et al, 2011). In contrary, LMOO only allows the incorporation of inequality constraints on the problem variables (Lai et al, 2021). On each level $l = 1, \dots, p$ the optimal slacks $v^*_{\mathbb{E}_l}$ and $v^*_{\mathbb{I}_l}$ need to be identified. At the same time the optimal slacks $v^*_{\mathbb{E}_{\cup l-1}}$ and $v^*_{\mathbb{I}_{\cup l-1}}$ of the previous levels 1 to $l-1$ must not be violated.

In essence, a HLSP with p levels is an optimization problem composed of p consecutive least-squares programs (LSP). Each level l consists of a least squares objective which is subject to the linear constraints associated with the higher priority levels 1 to $l-1$ (except for the first level 1 which does not

carry any constraints from previous levels). The corresponding LSP's of levels $l = 1, \ldots, p$ can be written as

$$\begin{aligned} \min_{x, v_{\mathbb{E}_l}, v_{\mathbb{I}_l}} \quad & \frac{1}{2}\|v_{\mathbb{E}_l}\|^2 + \frac{1}{2}\|v_{\mathbb{I}_l}\|^2 \qquad l = 1, \ldots, p \\ \text{s.t.} \quad & A_{\mathbb{E}_l} x - b_{\mathbb{E}_l} = v_{\mathbb{E}_l} \\ & A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} \geq v_{\mathbb{I}_l} \\ & A_{\mathbb{E}_{\cup l-1}} x - b_{\mathbb{E}_{\cup l-1}} = v^*_{\mathbb{E}_{\cup l-1}} \\ & A_{\mathbb{I}_{\cup l-1}} x - b_{\mathbb{I}_{\cup l-1}} \geq 0 \end{aligned} \tag{2}$$

$v^*_{\mathbb{E}_{\cup l-1}}$ indicates the optimal slacks that were identified for the equality constraints of the previous levels 1 to $l-1$. Inequality constraints are assumed to be feasible on each priority level. In Sec. 2 we introduce a relaxation of this assumption based on the active-set method such that the original lexicographical problem (1) is represented exactly.

The authors in Kanoun et al (2009) consecutively solve each level l from the first to the last level p of (2) in a cascade-like manner. The disadvantage of this approach is that each optimization problem grows in the number of constraints since the ones from the previous levels need to be carried over. In De Lasa and Hertzmann (2009) this is avoided by solving each level in the nullspace of the active constraints of the previous levels. This way the problem dimensions are progressively reduced. This nullspace projection and consequent variable reduction is commonly referred to as the *nullspace method* (Nocedal and Wright, 2006). Additionally, only feasible inequality constraints are considered on the first level. The approach considered in Escande et al (2014) is based on this same principle but enables the incorporation of inequality constraints on any priority level. Unlike the previous cascade-like approaches (Kanoun et al, 2009; De Lasa and Hertzmann, 2009), their active-set search solves the whole hierarchy at once which adds further efficiency by unifying the active-set search of all the levels into one.

The dedicated HLSP solver in Escande et al (2014) is based on the ASM. At each solver iteration an equality only problem consisting of the active constraints $\mathcal{A}_{\cup p}$ of the priority levels 1 to p is solved. $\mathcal{A}_{\cup p}$ assembles equality constraints $\mathbb{E}_{\cup p}$ and active inequality constraints of $\mathbb{I}_{\cup p}$. Based on the violation or feasibility of inequality constraints $\mathbb{I}_{\cup p}$, the active set $\mathcal{A}_{\cup p}$ is composed accordingly. The ASM converges if no constraint needs to be added to or removed from the active set. In ASM's based on the nullspace method, all constraint matrices of levels l to p are projected into the nullspace of the active constraints $\mathcal{A}_{\cup l-1}$ of levels 1 to $l-1$ (Coleman, 1984; Benzi et al, 2005). With the right choice of nullspace basis this leads to a possibly significant reduction of variables especially on lower priority levels in case of a large number of linearly independent active constraints. This makes the solvers very efficient as the matrix factorization for the linear system solution is cubically dependent of the number of variables. While the work in Escande et al (2014)

uses orthogonal nullspace bases, the authors in Dimitrov et al (2015) implement non-orthogonal bases with further computational advantage due to their block structure. The disadvantage of non-orthogonal bases is that the primal x is not of minimum norm but this characteristic can be easily enforced by regularization.

ASM based solvers are most efficient on problems with limited changes of the active-set between HLSP problem instances. This is usually the case for parametric problems like robot control scenarios where the problem variables evolve slowly. In this case the ASM can be warm started by setting the active set of the current problem with the active set from the previous one (Gill et al, 1986). However, in robot real-time control there are cases where large shifts of the active set occur, for example in instances close to contact shifting or oscillations in case of numerical instabilities due to ill-posed constraint matrices (Pfeiffer et al, 2018; Pfeiffer et al, 2023). In these cases the number of iterations of the ASM may be exponential in the number of constraints as all possible active-set combinations need to be explored (Rao et al, 1998). Numerical issues like cycling (repeated activation and deactivation of the same constraint) can further increase the number of active set iterations such that it becomes impractical for real-time robot control. While methods mitigating such effects like decomposition updates (Hammarling and Lucas, 2008) and cycling handling (Gill et al, 1989) exist, they might not be enough to make up for the possibly large number of active set iterations until convergence (Pfeiffer et al, 2023).

In contrast, the IPM has been shown to robustly converge in a deterministic number of iterations independent of problem conditioning (Nesterov and Nemirovskii, 1994; Bartlett et al, 2000). The IPM has been first developed for linear programming (Karmarkar, 1984) but has seen extensions to quadratic programs (which LSP's are a sub-form of, Vanderbei (1999)) or non-linear programming (Wächter and Biegler, 2006). The ASM only considers inequality constraints which are deemed active in the current guess of the active-set. In contrast, the IPM considers all constraints including all inactive inequality constraints. In primal-dual formulations of the IPM, the primal and dual variables thereby move within the interior of the feasible region. This is first achieved by maintaining feasibility conditions by line search or more complex methods based on non-linear arc-searches (Yang, 2022). Furthermore, violations are penalized for example by a log-barrier function. A function with an upper bound on solver iterations has been proposed for linear programs in Fathi Hafshejani et al (2020). A summary of different functions is given in Li et al (2021). A single iteration of the IPM is relatively expensive but convergence is achieved robustly. The authors in Kuindersma et al (2014) use this fact to switch from the ASM to the reliable IPM in case of ASM failures.

1.3 Our contribution

While the IPM has been developed for LSP (Vanderbei, 1999) and could be used to solve (2) directly by solving each level in sequence (Kanoun et al,

2009), there exist no dedicated IPM based solvers for HLSP. In this work we provide the theoretical foundations for an efficient IPM for HLSP. The IPM based HLSP solver provides predictability in terms of computation times due to its constant number of Newton iterations even in case of ill-posed constraints matrices. This can be important for example when critical safety constraints need to be dealt with but the ASM fails to find an optimal point in a reasonable number of active set iterations.

Our contributions are threefold:

- We formulate the IPM for HLSP. It reliably resolves ill-posed HLSP's without significant fluctuations in solver iterations or computation time. We show that this enables a humanoid robot to handle large changes of the active set after a strong push.
- We suggest to apply the nullspace method (Nocedal and Wright, 2006) to the IPM for HLSP instead of the commonly used Schur complement (Wang and Boyd, 2010; Domahidi et al, 2012). This transfers the same variable reducing quality to the IPM for HLSP as seen for the ASM for HLSP. While this has been previously done for QP's in Model predictive control (MPC) (Frison and Diehl, 2020), we give a more detailed explanation for example on our choice of nullspace basis. With this formulation we reduce the necessary number of decompositions of the KKT system per Newton iteration from two to one. We show that this is computationally efficient for almost all problem constellations. Furthermore, the Lagrange multipliers associated with active constraints do not need to be evaluated by virtue of an adapted IPM convergence test.
- We show that the IPM can be expressed in least squares form. This way the formation of expensive matrix products is avoided which is of advantage for problems with high number of variables but low number of constraints. This also potentially preserves a large degree of sparsity of the constraint matrices which we aim to exploit in a future sparse version of the proposed solver.

1.4 Overview

This article is structured as follows: In Sec. 2 we first introduce the notion of active sets into the optimization problem (2). We then continue to outline the formulation of the IPM for HLSP. Section 3 oversees the efficient computation of the IPM solver iterations. First, the overall algorithm is outlined in Sec. 3.1. We apply the nullspace method which is a common tool in hierarchical programming, see Sec. 3.2. This requires a concept we refer to as 'virtual priority level' and which is detailed in Sec. 3.3. With a special convergence test the calculation of the dual associated with the active constraints can be avoided (Sec. 3.4). In Sec. 3.5 we show that the solver iterations can be expressed in efficient least-squares form. Section 3.6 oversees the development of Mehrotra's predictor-corrector-algorithm for HLSP's. Finally, we give a computational comparison between the different solver formulations (Sec. 4). The proposed

algorithms are evaluated in Sec. 5. We conclude the article with some remarks and considerations for future work (see Sec. 6).

2 Formulating the interior point method for HLSP

We first separately consider the top three lines of the optimization problem (2). This problem corresponds to finding a *feasible point* ($v^*_{\mathbb{E}_l} = 0$, $v^*_{\mathbb{I}_l} \geq 0$) or *optimal infeasible point* ($v^*_{\mathbb{E}_l} \neq 0$, $v^*_{\mathbb{I}_l} < 0$ with minimal $\|v_{\mathbb{E}_l}\|^2$, $\|v_{\mathbb{I}_l}\|^2$) of the constraints. In the context of linear programming the self-dual embedding model (Ye et al, 1994) has been proposed. Similarly to Wang and Boyd (2010) for quadratic programming, infeasible initial points with respect to the equality constraints are handled but inequality constraints are assumed to be feasible with $A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} \geq 0$. An algorithm overcoming this issue was proposed in Gill et al (1986) where the initial feasible point is determined by minimizing the sum of infeasibilities. However, the algorithm fails when no feasible but only an optimal infeasible point $A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} < 0$ exists. The solvers proposed in Kanoun et al (2011); Escande et al (2014); Dimitrov et al (2015) are based on the ASM and are able of handling all the above cases by virtue of relaxation with slack variables. Similarly, we introduce the notion of active constraints into (2) ((2) is equivalent to (1) only if all inequality constraints on all levels are feasible):

$$
\begin{aligned}
\underset{x, v_{\mathbb{E}_l}, v_{\mathbb{I}_l}}{\text{min.}} \quad & \frac{1}{2}\|v_{\mathbb{E}_l}\|^2 + \frac{1}{2}\|v_{\mathbb{I}_l}\|^2 \qquad l = 1, \ldots, p \qquad \text{(HLSP)} \\
\text{s.t.} \quad & A_{\mathbb{E}_l} x - b_{\mathbb{E}_l} = v_{\mathbb{E}_l} \\
& A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} \geq v_{\mathbb{I}_l} \\
& A_{\mathcal{A}_{\cup l-1}} x - b_{\mathcal{A}_{\cup l-1}} = v^*_{\mathcal{A}_{\cup l-1}} \\
& A_{\mathcal{I}_{\cup l-1}} x - b_{\mathcal{I}_{\cup l-1}} \geq 0
\end{aligned}
$$

The active set $\mathcal{A}_{\cup l-1}$ represents all equality constraints $\mathbb{E}_{\cup l-1}$ and the inequality constraints of $\mathbb{I}_{\cup l-1}$ that were infeasible and violated by $v^*_{\mathcal{A}_{\cup l-1}}$ at the optimal points of the levels 1 to $l-1$. Consequently, $\mathcal{I}_{\cup l-1}$ are the remaining inequality constraints that were feasible and satisfied at the optimal points of the levels 1 to $l-1$ and which are labeled as 'inactive'. More details on setting these active sets are given in Sec. 3.3.

The aim of the optimization problem is

1. to find the feasible or optimal infeasible point $v^*_{\mathbb{E}_l}$ and $v^*_{\mathbb{I}_l}$ of the equality and inequality constraints $\mathbb{E}_l$ and $\mathbb{I}_l$ of the current level l
   $\rightarrow$ IPM
2. while keeping the inactive inequality constraints $\mathcal{I}_{\cup l-1}$ of the previous levels $1, \ldots, l-1$ feasible with $A_{\mathcal{I}_{\cup l-1}} x - b_{\mathcal{I}_{\cup l-1}} \geq 0$
   $\rightarrow$ IPM

3. and while keeping the active constraints $\mathcal{A}_{\cup l-1}$ of the previous levels $1, \ldots, l-1$ optimal at the current violation $v^*_{\mathcal{A}_{\cup l-1}}$
   $\rightarrow$ Nullspace method

The first and second point we achieve by the IPM as outlined below. The third point is outlined in Sec. 3.2.

2.1 The Newton's method in the IPM

We introduce two positive slack variables, $w_{\mathbb{I}_l}$ for the inequality constraints on the current level l, and $w_{\mathcal{I}_{\cup l-1}}$ for the inactive inequality constraints of the previous levels. They are then penalized by the log function to avoid values approaching zero ('log-barrier'):

$$
\begin{aligned}
\min_x \; & \frac{1}{2}\|v_{\mathbb{E}_l}\|^2 + \frac{1}{2}\|v_{\mathbb{I}_l}\|^2 - \sigma_{\mathbb{I}_l}\mu_{\mathbb{I}_l}\sum \log(w_{\mathbb{I}_l}) - \sigma_{\mathcal{I}_{\cup l-1}}\mu_{\mathcal{I}_{\cup l-1}}\sum \log(w_{\mathcal{I}_{\cup l-1}}) \\
\text{s.t.} \; & A_{\mathbb{E}_l}x - b_{\mathbb{E}_l} = v_{\mathbb{E}_l} \\
& A_{\mathbb{I}_l}x - b_{\mathbb{I}_l} - v_{\mathbb{I}_l} = w_{\mathbb{I}_l} \\
& w_{\mathbb{I}_l} \geq 0 \\
& A_{\mathcal{A}_{\cup l-1}}x - b_{\mathcal{A}_{\cup l-1}} = v^*_{\mathcal{A}_{\cup l-1}} \\
& A_{\mathcal{I}_{\cup l-1}}x - b_{\mathcal{I}_{\cup l-1}} = w_{\mathcal{I}_{\cup l-1}} \\
& w_{\mathcal{I}_{\cup l-1}} \geq 0
\end{aligned} \tag{3}
$$

The Lagrangian of the optimization problem (3) is

$$
\begin{aligned}
\mathcal{L} \coloneqq & \frac{1}{2}\|v_{\mathbb{E}_l}\|^2 + \frac{1}{2}\|v_{\mathbb{I}_l}\|^2 - \sigma_{\mathbb{I}_l}\mu_{\mathbb{I}_l}\sum \log(w_{\mathbb{I}_l}) - \sigma_{\mathcal{I}_{\cup l-1}}\mu_{\mathcal{I}_{\cup l-1}}\sum \log(w_{\mathcal{I}_{\cup l-1}}) \\
& - \lambda^T_{\mathbb{E}_l}(A_{\mathbb{E}_l}x - b_{\mathbb{E}_l} - v_{\mathbb{E}_l}) - \lambda^T_{\mathbb{I}_l}(A_{\mathbb{I}_l}x - b_{\mathbb{I}_l} - v_{\mathbb{I}_l} - w_{\mathbb{I}_l}) \\
& - \lambda^T_{\mathcal{A}_{\cup l-1}}(A_{\mathcal{A}_{\cup l-1}}x - b_{\mathcal{A}_{\cup l-1}} - v^*_{\mathcal{A}_{\cup l-1}}) - \lambda^T_{\mathcal{I}_{\cup l-1}}(A_{\mathcal{I}_{\cup l-1}}x - b_{\mathcal{I}_{\cup l-1}} - w_{\mathcal{I}_{\cup l-1}})
\end{aligned} \tag{4}
$$

λ are the Lagrange multipliers associated with the corresponding constraints $\mathbb{E}_l$, $\mathbb{I}_l$, $\mathcal{A}_{\cup l-1}$ and $\mathcal{I}_{\cup l-1}$.

The first order optimality or KKT conditions $K \coloneqq \nabla_q \mathcal{L} = 0$ with the variable vector q

$$
q \coloneqq \begin{bmatrix} x^T & v^T_{\mathbb{E}_l} & v^T_{\mathbb{I}_l} & w^T_{\mathbb{I}_l} & \lambda^T_{\mathcal{A}_{\cup l-1}} & \lambda_{\mathcal{I}_{\cup l-1}} & w^T_{\mathcal{I}_{\cup l-1}} \end{bmatrix}^T \tag{5}
$$

are (after applying the substitutions $v_{\mathbb{E}_l} = -\lambda_{\mathbb{E}_l}$ and $v_{\mathbb{I}_l} = -\lambda_{\mathbb{I}_l}$)

$$K_l(q) = \begin{bmatrix} K_{x,l} \\ K_{v_{\mathbb{E}_l},l} \\ K_{v_{\mathbb{I}_l},l} \\ K_{w_{\mathbb{I}_l},l} \\ K_{\lambda_{\mathcal{A}\cup l-1},l} \\ K_{\lambda_{\mathcal{I}\cup l-1},l} \\ K_{w_{\mathcal{I}\cup l-1},l} \end{bmatrix} \coloneqq \begin{bmatrix} A_{\mathbb{E}_l}^T v_{\mathbb{E}_l} + A_{\mathbb{I}_l}^T v_{\mathbb{I}_l} - A_{\mathcal{A}\cup l-1}^T \lambda_{\mathcal{A}\cup l-1} - A_{\mathcal{I}\cup l-1}^T \lambda_{\mathcal{I}\cup l-1} \\ b_{\mathbb{E}_l} - A_{\mathbb{E}_l} x + v_{\mathbb{E}_l} \\ b_{\mathbb{I}_l} - A_{\mathbb{I}_l} x + v_{\mathbb{I}_l} + w_{\mathbb{I}_l} \\ w_{\mathbb{I}_l} \odot v_{\mathbb{I}_l} + \sigma_{\mathbb{I}_l} \mu_{\mathbb{I}_l} e \\ b_{\mathcal{A}\cup l-1} - A_{\mathcal{A}\cup l-1} x + v_{\mathcal{A}\cup l-1}^* \\ b_{\mathcal{I}\cup l-1} - A_{\mathcal{I}\cup l-1} x + w_{\mathcal{I}\cup l-1} \\ \lambda_{\mathcal{I}\cup l-1} \odot w_{\mathcal{I}\cup l-1} - \sigma_{\mathcal{I}\cup l-1} \mu_{\mathcal{I}\cup l-1} e \end{bmatrix} = 0 \tag{KKT}$$

The operator $\odot$ indicates the element-wise multiplication between two vectors. e is a vector of 1's of appropriate dimensions. The duality measures μ and the centering parameter σ are given by (Vanderbei, 2013)

$$\mu_{\mathbb{I}_l} \coloneqq \lambda_{\mathbb{I}_l}^T w_i / (n + m_{l,i}) \tag{6}$$

$$\mu_{\mathcal{I}\cup l-1} \coloneqq \lambda_{\mathcal{I}\cup l-1}^T w_{\mathcal{I}\cup l-1} / (n + m_{\mathcal{I}\cup l-1}) \tag{7}$$

$$\sigma_{\mathbb{I}_l}, \sigma_{m_{\mathcal{I}\cup l-1}} \in [0, 1] \tag{8}$$

The values can also be determined by Mehtrotra's predictor-corrector algorithm (Mehrotra, 1992), see Sec. 3.6.

The KKT conditions are non-linear. We linearize them by the Newton's method by applying the Newton step

$$K_l(q + \Delta q) \approx K_l(q) + \nabla_q K_l(q) \Delta q \tag{9}$$

The Lagrangian Hessian is given by

$$\nabla_q K_l(q) \coloneqq \begin{bmatrix} 0 & A_{\mathbb{E}_l}^T & A_{\mathbb{I}_l}^T & 0 & -A_{\mathcal{A}\cup l-1}^T & -A_{\mathcal{I}\cup l-1}^T & 0 \\ -A_{\mathbb{E}_l} & I & 0 & 0 & 0 & 0 & 0 \\ -A_{\mathbb{I}_l} & 0 & I & I & 0 & 0 & 0 \\ 0 & 0 & W_{\mathbb{I}_l} & V_{\mathbb{I}_l} & 0 & 0 & 0 \\ -A_{\mathcal{A}\cup l-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ -A_{\mathcal{I}\cup l-1} & 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & W_{\mathcal{I}\cup l-1} & \Lambda_{\mathcal{I}\cup l-1} \end{bmatrix} \tag{10}$$

The capital variables $W = \text{diag}(w) \in \mathbb{R}^{m,m}$, $V = \text{diag}(V) \in \mathbb{R}^{m,m}$ and $\Lambda = \text{diag}(\lambda) \in \mathbb{R}^{m,m}$ are the diagonal matrix equivalents of their vectors $w \in \mathbb{R}^m$, $v \in \mathbb{R}^m$ and $\lambda \in \mathbb{R}^m$. I is an identity matrix of appropriate dimensions.

We sequentially apply substitutions for $\Delta v_{\mathbb{E}_l}$, $\Delta v_{\mathbb{I}_l}$, $\Delta w_{\mathcal{I}_{\cup l-1}}$, $\Delta w_{\mathbb{I}_l}$ and $\Delta\lambda_{\mathcal{I}_{\cup l-1}}$ which leads to the *hierarchical augmented system*

$$\begin{bmatrix} C_l & -A^T_{\mathcal{A}_{\cup l-1}} \\ -A_{\mathcal{A}_{\cup l-1}} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda_{\mathcal{A}_{\cup l-1}} \end{bmatrix} = \begin{bmatrix} r_{l,1} \\ r_{l,2} \end{bmatrix} \tag{11}$$

with

$$C_l \coloneqq A^T_{\mathbb{E}_l} A_{\mathbb{E}_l} + A^T_{\mathbb{I}_l} \left(I + (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1} W_{\mathbb{I}_l}\right) A_{\mathbb{I}_l} + A^T_{\mathcal{I}_{\cup l-1}} w^{-1}_{\mathcal{I}_{\cup l-1}} \lambda_{\mathcal{I}_{\cup l-1}} A_{\mathcal{I}_{\cup l-1}} \tag{12}$$

and the right hand side

$$r_{l,1} \coloneqq A^T_{\mathcal{A}_{\cup l-1}} \lambda_{\mathcal{A}_{\cup l-1}} + A^T_{\mathcal{I}_{\cup l-1}} F + A^T_{\mathbb{E}_l} (b_{\mathbb{E}_l} - A_{\mathbb{E}_l} x) + A^T_{\mathbb{I}_l} G \tag{13}$$

$$r_{l,2} \coloneqq A_{\mathcal{A}_{\cup l-1}} x - b_{\mathcal{A}_{\cup l-1}} - w^*_{\mathcal{A}_{\cup l-1}} \tag{14}$$

G and F are given by

$$F_l \coloneqq \lambda_{\mathcal{I}_{\cup l-1}} + W^{-1}_{\mathcal{I}_{\cup l-1}} (\lambda_{\mathcal{I}_{\cup l-1}} \odot (b_{\mathcal{I}_{\cup l-1}} - A_{\mathcal{I}_{\cup l-1}} x) + \sigma_{\mathcal{I}_{\cup l-1}} \mu_{\mathcal{I}_{\cup l-1}} e) \tag{15}$$

$$G_l \coloneqq b_{\mathbb{I}_l} - A_{\mathbb{I}_l} x + w_{\mathbb{I}_l} - (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1} (\sigma_{\mathbb{I}_l} \mu_{\mathbb{I}_l} e + w_{\mathbb{I}_l} \odot (A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} - w_{\mathbb{I}_l})) \tag{16}$$

respectively.

2.2 The iterative nature of the IPM

The linear hierarchical augmented system (11) of level l is now repeatedly solved for the primal and dual steps Δq. The IPM's for finding the initial feasible or optimal infeasible point require following *dual feasibility* conditions to hold

$$v_{\mathbb{I}_l} \leq 0 \qquad \text{and} \qquad w_{\mathbb{I}_l} \geq 0 \tag{17}$$

For the IPM of the inactive constraints we have

$$\lambda_{\mathcal{I}_{\cup l-1}} \geq 0 \qquad \text{and} \qquad w_{\mathcal{I}_{\cup l-1}} \geq 0 \tag{18}$$

These conditions are maintained by line search. The computed primal and dual steps Δq are then scaled by the line search factor α and added to the current estimates of the primal and dual $q \leftarrow q + \alpha\Delta q$. The Newton's method terminates once the norm of the non-linear KKT conditions $\|K_l(q)\|_2 < \epsilon$ (KKT) is below a certain threshold $\epsilon = 10^{-12}$.

3 Computing the IPM for HLSP

Solving the augmented system (11) directly in a dense manner is inefficient as the zero block in the lower right corner of the left hand side matrix would

be ignored (a sparse solver has been proposed for example in Pandala et al (2019)). One way to circumvent this is to form the Schur complement (Bartlett and Biegler, 2006; Domahidi et al, 2012)

$$\Delta x = C_l^{-1}(r_{l,1} + A_{\mathcal{A}_{\cup l-1}}^T \Delta\lambda_{\mathcal{A}_{\cup l-1}}) \tag{19}$$

The resulting 'Schur' normal equations are given below (IPM-snf: Schur normal form or Schur normal equations):

$$A_{\mathcal{A}_{\cup l-1}} C_l^{-1} A_{\mathcal{A}_{\cup l-1}}^T \Delta\lambda_{\mathcal{A}_{\cup l-1}} = -r_{l,2} - A_{\mathcal{A}_{\cup l-1}} C_l^{-1} r_{l,1} \tag{IPM-snf}$$

In order to obtain the primal step Δx two decompositions per Newton iteration are required, one for C_l^{-1} and one for $A_{\mathcal{A}_{\cup l-1}} C_l^{-1} A_{\mathcal{A}_{\cup l-1}}^T$. Efficient strategies especially in the context of Model-Predictive-Control (MPC) leveraging banded matrix structures have been proposed (Domahidi et al, 2012). However, this still can prove inefficient, especially if the dual $\lambda_{\mathcal{A}_{\cup l-1}}$ is not required for later use (they might be required for example for Hessian computations (Pfeiffer et al, 2023)). We therefore describe in the following an algorithm based on the nullspace method that only requires a single decomposition per Newton iteration.

3.1 Algorithm for NIPM-HLSP

An overview of our algorithm NIPM-HLSP to resolve HLSP's is given below:

1. Go through the hierarchy HLSP from level 1 to p.
2. For each level l, repeatedly compute the Newton step by solving NIPM-nf or NIPM-ls formulated in Sec. 3.2 and Sec. 3.5, respectively. In case of the Mehrotra's predictor corrector algorithm (Sec. 3.6), first compute the affine step $\Delta\_^{\text{aff}}$ and then the centered one $\Delta\_$. The symbol $\_$ is a placeholder for the different variables $x, v_{\mathbb{I}_l}, w_{\mathbb{I}_l}, w_{\mathcal{I}_{\cup l-1}}$ and $\lambda_{\mathcal{I}_{\cup l-1}}$.
3. Line search for dual feasibility (17) and (18). Add step scaled by the line search factor to current primal and dual estimate.
4. Upon convergence of the Newton's method $\|\tilde{K}\| \leq \epsilon$, gather all inactive inequality constraints from higher priority levels $\mathcal{I}_{\cup l-1}$ that are saturated and add them to the active set $\mathcal{A}_{l^*}$ of the virtual priority level l^*. The active set assembly including the concept of virtual priority levels is explained in Sec. 3.3. $\tilde{K}$ are the projected KKT conditions described in Sec. 3.4.
5. Compute its nullspace $Z_{\mathcal{A}_{l^*}}, r \leftarrow \mathcal{N}(A_{\mathcal{A}_{l^*}})$ (the operator $\mathcal{N}(A)$ returns the rank and a basis of the nullspace Z of the input matrix A) and project lower priority levels and the remaining inactive inequality constraints into it. Augment $N_{\mathcal{A}_{\cup l^*}} \leftarrow N_{\mathcal{A}_{\cup l-1}} Z_{\mathcal{A}_{l^*}}$.
6. Add all equalities and the violated inequality constraints from level l to the active set $\mathcal{A}_l$ of level l. This is described in Sec. 3.3.
7. Compute its nullspace $Z_{\mathcal{A}_l}, r \leftarrow \mathcal{N}(A_{\mathcal{A}_l})$ and project lower priority levels and the remaining inactive inequality constraints into it. Augment $N_{\mathcal{A}_{\cup l}} \leftarrow N_{\mathcal{A}_{\cup l^*}} Z_{\mathcal{A}_l}$.

A pseudo-implementation of the algorithm is given in App. C.

3.2 $\mathcal{N}$IPM-HSLP: The nullspace method based IPM for HLSP

We first assume that $r_{l,2} = 0$ (14). This condition holds as x is updated after every Newton iteration during the resolution of the higher priority levels $l = 1, \ldots, l-1$ and is therefore feasible with respect to the active constraints $\mathcal{A}_{\cup l-1}$. We then apply the nullspace method (Nocedal and Wright, 2006) by first introducing the variable change

$$\Delta x = N_{\mathcal{A}_{\cup l-1}} \Delta z \tag{20}$$

The nullspace basis $N_{\mathcal{A}_{\cup l-1}}$ of $A_{\mathcal{A}_{\cup l-1}}$ fulfills the condition $A_{\mathcal{A}_{\cup l-1}} N_{\mathcal{A}_{\cup l-1}} = 0$. This means that $A_{\mathcal{A}_{\cup l-1}} \Delta x = 0$ such that the condition $r_{l,2} = 0$ continues to be fulfilled. An additional multiplication of the first row of (11) by $N^T_{\mathcal{A}_{\cup l-1}}$ from the left results in the 'projected' normal equations

$$N^T_{\mathcal{A}_{\cup l-1}} C_l N_{\mathcal{A}_{\cup l-1}} \Delta z = N^T_{\mathcal{A}_{\cup l-1}} r_{l,1} \qquad (\mathcal{N}\text{IPM-nf})$$

As can be seen, the primal step Δz is directly obtained without the intermediate step of computing the Lagrange multipliers $\lambda_{\mathcal{A}_{\cup l-1}}$ corresponding to the active constraints $\mathcal{A}_{\cup l-1}$. They can be obtained by solving

$$A^T_{\mathcal{A}_{\cup l-1}} \Delta \lambda_{\mathcal{A}_{\cup l-1}} = C_l \Delta x - r_{l,1} \tag{21}$$

An efficient recursive method of calculation is detailed in Appendix A. As we detail in Sec. 3.4, we do not necessarily need to compute these Lagrange multipliers after all.

We use the nullspace basis described in Nocedal and Wright (2006)

$$Z_{\mathcal{A}_l} = P \begin{bmatrix} -R^{-1}T \\ I \end{bmatrix} \quad \text{with} \quad A_{\mathcal{A}_l} = Q \begin{bmatrix} R & T \\ 0 & 0 \end{bmatrix} P^T \tag{22}$$

Resulting from a rank-revealing QR decomposition (RRQR) of $A_{\mathcal{A}_l}$, Q is an orthogonal matrix, R is upper triangular, T is a rectangular matrix and P is a permutation matrix. The bottom zero row is due to possible linear dependencies in $A_{\mathcal{A}_l}$. This basis has variable reducing qualities (Björck, 1996) (i.e. projected matrices are reduced in variables), projections are computed cheaply due to the lower identity matrix and it can be reused for the calculation of the Lagrange multipliers (App. A). While this basis is partly sparse, its projections are dense if there is any density in the factors R or T. The accumulated nullspace basis $N_{\mathcal{A}_{\cup l}}$ is computed by $N_{\mathcal{A}_{\cup l}} = Z_{\mathcal{A}_1} \ldots Z_{\mathcal{A}_l}$.

3.3 Active set assembly and virtual priority levels

Once the Newton's method of a priority level l has converged, inactive constraints $\mathcal{I}_{\cup l-1}$ from previous levels 1 to $l-1$ are either saturated with $A_{\mathcal{I}_{\cup l-1}}x - b_{\mathcal{I}_{\cup l-1}} = 0$ or satisfied with $A_{\mathcal{I}_{\cup l-1}}x - b_{\mathcal{I}_{\cup l-1}} > 0$. Additionally, inequality constraints $\mathbb{I}_l$ from the current level l are either satisfied with $A_{\mathbb{I}_l}x - b_{\mathbb{I}_l} \geq 0$ or violated with $A_{\mathbb{I}_l}x - b_{\mathbb{I}_l} < 0$. In order to be able to proceed with the cascade-like resolution of the next priority level we need to determine two distinct active sets $\mathcal{A}_{l^*}$ and $\mathcal{A}_l$ (such that $\mathcal{A}_{\cup l} \coloneqq \mathcal{A}_{1^*} \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{l^*} \cup \mathcal{A}_l$). The index l^* represents a 'virtual' priority level whose active set $\mathcal{A}_{l^*}$ contains saturated constraints from $\mathcal{I}_{\cup l-1}$. This is necessary as the projection of $\mathcal{I}_{\cup l-1}$ into the nullspace basis of the active constraints $\mathcal{A}_l$ of level l would contradict the priority order. Therefore, a virtual priority level l^* has lower priority than level $l-1$ but higher priority than level l whose active set $\mathcal{A}_l$ comprises of the equality constraints $\mathbb{E}_l$ and the activated constraints from $\mathbb{I}_l$.

We activate a constraint $c \in \mathcal{I}_{\cup l-1}$ if the corresponding pair of slack and Lagrange multiplier fulfills the conditions

$$w_{\mathcal{I}_{\cup l-1}}(c) < \xi \qquad \text{and} \qquad \lambda_{\mathcal{I}_{\cup l-1}}(c) > \xi \tag{23}$$

While $w_{\mathcal{I}_{\cup l-1}}(c) < \xi$ ensures that only saturated constraints are activated, the condition $\lambda_{\mathcal{I}_{\cup l-1}}(c) > \xi$ only selects constraints that are in significant conflict with constraints from level l and are not just accidentally saturated, for example by a randomly chosen initial point. Otherwise the ability of resolving the lower priority level l is artificially restricted by an unnecessarily inflated active set $\mathcal{A}_{\cup l-1}$.

The threshold $\xi = 10^{-8}$ is necessary as the Newton's method is usually not run to complete convergence $\|K_l\|_2 = 0$ but stopped earlier with some threshold $\|K_l\|_2 < \epsilon$. The small choice for ξ requires the Newton's method to converge to a similarly accurate degree $\epsilon = 10^{-12}$ in order to have an accurate value of the dual $w_{\mathcal{I}_{\cup l-1}}$ and $\lambda_{\mathcal{I}_{\cup l-1}}$.

The following steps are conducted for the virtual priority level l^*:

1. Add the newly activated constraints from $\mathcal{I}_{\cup l-1}$ to $\mathcal{A}_{l^*}$ as a 'virtual' priority level l^*.
2. Calculate the nullspace basis $Z_{\mathcal{A}_{l^*}}$ of $A_{\mathcal{A}_{l^*}} N_{\mathcal{A}_{\cup l-1}}$ and project lower priority levels and the remaining inactive inequality constraints into it.
3. Augment $N_{\mathcal{A}_{\cup l-1}}$ to $N_{\mathcal{A}_{\cup l^*}} = N_{\mathcal{A}_{\cup l-1}} Z_{\mathcal{A}_{l^*}}$

Now we handle the violated constraints from level l for the active set $\mathcal{A}_l$. A constraint $c \in \mathbb{I}_l$ is activated if the following conditions are fulfilled:

$$w_{\mathbb{I}_l}(c) < \xi \qquad \text{and} \qquad v_{\mathbb{I}_l}(c) < -\xi \tag{24}$$

Following steps complete the active set assembly:

1. Assemble the active set $\mathcal{A}_l$ of level l with all equality constraints and the active inequality constraints.

2. Inactive constraints are added to $\mathcal{I}_{\cup l}$. Note that in the case of bound constraints we check whether the variable is already constrained by the same constraint. If this is the case, no new constraint is added and the tighter bound of the two is used as the new right hand side $b_{\mathcal{I}_{\cup l}}$. This prevents unnecessarily increasing the size of $\mathcal{I}_{\cup l}$. Note that this is already implied by the set union symbol $\cup$.
3. Calculate the nullspace basis $Z_{\mathcal{A}l}$ of $A_{\mathcal{A}_l} N_{\mathcal{A}_{\cup l^*}}$ and project lower priority levels and the remaining inactive inequality constraints into it.
4. Augment $N_{\cup \mathcal{A}_{l^*}}$ to $N_{\mathcal{A}_{\cup l}} = N_{\mathcal{A}_{\cup l^*}} Z_{\mathcal{A}_l}$.

3.4 Avoiding the calculation of $\Delta\lambda_{\mathcal{A}_{\cup l-1}}$

The cost of calculating the dual step $\Delta\lambda_{\mathcal{A}_{\cup l-1}}$ associated with the equality constraints $\mathcal{A}_{\cup l-1}$ is of magnitude $O(r^2_{\mathcal{A}_{\cup l-1}})$. The rank of $A_{\mathcal{A}_{\cup l-1}}$ is given as $r_{\mathcal{A}_{\cup l-1}}$. However, we do *not necessarily* need to update the Lagrange multipliers $\lambda_{\mathcal{A}_{\cup l-1}}$ since none of the other primal or dual variables depend on them. Additionally, we can explicitly calculate $\lambda_{\mathcal{A}_{\cup l-1}}$ by solving $K_{x,l} = 0$ (KKT). This is in contrast to the augmented system (11) or the Schur normal equations (IPM-snf) which require the calculation of the dual step in order to obtain the primal step.

The dual $\lambda_{\mathcal{A}_{\cup l-1}}$ is only necessary for the evaluation of the norm of K_l (KKT) to determine whether the IPM has converged with $\|K_l\|_2 < \epsilon$. However, we can instead use the projected KKT conditions (we only show the relevant components)

$$\tilde{K}_{x,l} \coloneqq N^T_{\mathcal{A}_{\cup l-1}}(A^T_{\mathbb{E}_l} v_{\mathbb{E}_l} + A^T_{\mathbb{I}_l} v_{\mathbb{I}_l} - A^T_{\mathcal{I}_{\cup l-1}} \lambda_{\mathcal{I}_{\cup l-1}}) \tag{25}$$

$$\tilde{K}_{\lambda_{\mathcal{A}_{\cup l-1}},l} \coloneqq N^T_{\mathcal{A}_{\cup l-1}}(b_{\mathcal{A}_{\cup l-1}} - A_{\mathcal{A}_{\cup l-1}} x + v^*_{\mathcal{A}_{\cup l-1}}) = 0 \tag{26}$$

The nullity of the second equation holds since we already have ensured primal feasibility $v^*_{\mathcal{A}_{\cup l-1}}$ of the active constraints $\mathcal{A}_{\cup l-1}$ when resolving the previous levels $1, \ldots, l-1$. Furthermore, any nullspace step $\Delta x = N_{\mathcal{A}_{\cup l-1}} \Delta z$ does not influence $K_{\lambda_{\mathcal{A}_{\cup l-1}},l}$ since $A_{\mathcal{A}_{\cup l-1}}(x + N_{\mathcal{A}_{\cup l-1}} \Delta z) = A_{\mathcal{A}_{\cup l-1}} x$.

3.5 The least-squares form of the $\mathcal{N}$IPM-HLSP

$\mathcal{N}$IPM-nf can be rewritten to least squares form

$$\min_{\Delta z} \left\| \begin{bmatrix} \sqrt{w^{-1}_{\mathcal{I}_{\cup l-1}} \lambda_{\mathcal{I}_{\cup l-1}}} \tilde{A}_{\mathcal{I}_{\cup l-1}} \\ \sqrt{I + (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1} W_{\mathbb{I}_l}} \tilde{A}_{\mathbb{I}_l} \\ \tilde{A}_{\mathbb{E}_l} \end{bmatrix} \Delta z - \begin{bmatrix} \sqrt{w_{\mathcal{I}_{\cup l-1}} \lambda^{-1}_{\mathcal{I}_{\cup l-1}}} F \\ \sqrt{I + (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1} W_{\mathbb{I}_l}}^{-1} G \\ b_{\mathbb{E}_l} - A_{\mathbb{E}_l} x \end{bmatrix} \right\|^2 \tag{$\mathcal{N}$IPM-ls}$$

F and G are defined in (15) and (16), respectively. We use the notation $\tilde{M} \coloneqq M N_{\mathcal{A}_{\cup l-1}}$. The above is well defined as we show in the following.

Theorem 1. *The expressions under the square root of NIPM-ls are non-negative.*

Proof $W^{-1}_{\mathcal{I}_{\cup l-1}}\Lambda_{\mathcal{I}_{\cup l-1}} \geq 0$ and $W_{\mathcal{I}_{\cup l-1}}\Lambda^{-1}_{\mathcal{I}_{\cup l-1}} \geq 0$ follow directly from (18). Furthermore, for $\sqrt{I + (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1}W_{\mathbb{I}_l}}$ we have $V_{\mathbb{I}_l} - W_{\mathbb{I}_l} \leq 0$ because of the dual feasibility conditions (17). This leads to

$$I + (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1}W_{\mathbb{I}_l} \geq 0 \quad \leftrightarrow \quad V_{\mathbb{I}_l} \leq 0 \tag{27}$$

which is true and ensured by (17). □

The least-squares form has the advantage that the (substituted) Lagrangian Hessian (12) does not need to be computed. On the other hand, solving the system NIPM-ls requires a more expensive rank revealing decomposition (for example QR decomposition). Following our considerations from Sec. 4, we use the condition

$$n_r > \frac{3}{5}(m_{\mathcal{I}_{\cup l-1}} + m_{\mathbb{E}_l} + m_{\mathbb{I}_l}) \tag{28}$$

to decide whether the least-squares form (NIPM-ls) is more appropriate than the projected normal equations (NIPM-nf).

The above implies that the least-squares form is more efficient for a high ratio of number of variables to number of constraints ($n_r \gg m$). Yet, efficient parallel decomposition methods for tall matrices ($m \gg n_r$) have been evaluated for example in Sauk et al (2020). Furthermore, the above condition neglects the circumstance that the least-squares form can potentially maintain a high degree of sparsity of the constraint matrices A. In future work we therefore aim to implement a sparse version of the solver, for example based on sparsity maintaining nullspace bases as proposed in Gill et al (1987).

3.6 Mehrotra's predictor corrector algorithm

In the following we adapt Mehrotra's predictor-corrector algorithm (Mehrotra, 1992) to HLSP. It enables fast convergence of the Newton's method by choosing $\sigma_{\mathcal{I}_{\cup l-1}}$ and $\sigma_{\mathbb{I}_l}$ appropriately. The algorithm for LSP is described in Nocedal and Wright (2006) (Alg. 16.4) and only requires slight adaptations for HLSP which are detailed in Appendix B.

Even though the primal ($O(n_r^2)$) needs to be computed twice per Newton iteration this effort is negligible in comparison to the cost of computing a decomposition ($O(n_r^3)$) of the linear systems NIPM-nf or NIPM-ls.

4 Computational complexity of NIPM-HLSP

4.1 Computational complexity of single Newton iteration

Table 1 details the number of operations per Newton iteration for the Schur (IPM-snf) and projected normal equations (NIPM-nf) and the least squares form (NIPM-ls). The projected normal equations (4 steps) and the least squares form (3 steps) require less steps per Newton iteration than

Table 1 Necessary steps (#) per Newton iteration and their number of operations for the different IPM formulations for HLSP.

#	IPM-snf	$\mathcal{N}$IPM-nf	$\mathcal{N}$IPM-ls
1.	Form C_l (12): $O((m_{\mathbb{E}_l} + m_{\mathbb{I}_l} + m_{\mathcal{I}_{\cup l-1}})n^2)$	Form $N^T_{\mathcal{A}_{\cup l-1}} C_l N_{\mathcal{A}_{\cup l-1}}$: $O((m_{\mathbb{E}_l} + m_{\mathbb{I}_l} + m_{\mathcal{I}_{\cup l-1}})n_r^2)$	QR decomposition of $\mathcal{N}$IPM-ls: $O(2n_r^2(m_{\mathcal{I}_{\cup l-1}} + m_{\mathbb{E}_l} + m_{\mathbb{I}_l}) - 2n_r^3/3)$
2.	LDLT decomposition of C_l: $O(n^3/3)$	LDLT decomposition of $N^T_{\mathcal{A}_{\cup l-1}} C_l N_{\mathcal{A}_{\cup l-1}}$: $O(n_r^3/3)$	Solve for Δz: $O(n_r^2)$
3.	Form matrix $A_{\mathcal{A}_{\cup l-1}} C_l^{-1} A^T_{\mathcal{A}_{\cup l-1}}$: $O(m_{\mathcal{A}_{\cup l-1}} n^2 + m^2_{\mathcal{A}_{\cup l-1}} n)$ (Domahidi et al, 2012)	Solve for Δz: $O(n_r^2)$	$\Delta x = N\Delta z$: $O(nn_r)$
4.	LDLT decomposition of $A_{\mathcal{A}_{\cup l-1}} C_l^{-1} A^T_{\mathcal{A}_{\cup l-1}}$: $O(m^3_{\mathcal{A}_{\cup l-1}}/3)$	$\Delta x = N_{\mathcal{A}_{\cup l-1}} \Delta z$: $O(nn_r)$	
5.	Calculate dual step $\Delta\lambda_{\mathcal{A}_{\cup l-1}}$ (IPM-snf): $O(m^2_{\mathcal{A}_{\cup l-1}})$		
6.	Calculate primal step Δx (19): $O(n^2)$		
$\sum$	$O((m_{\mathbb{E}_l} + m_{\mathbb{I}_l} + m_{\mathcal{I}_{\cup l-1}} + n/3 + m_{\mathcal{A}_{\cup l-1}} + 1)n^2 + (n + m_{\mathcal{A}_{\cup l-1}}/3 + 1)m^2_{\mathcal{A}_{\cup l-1}})$	$O((m_{\mathbb{E}_l} + m_{\mathbb{I}_l} + m_{\mathcal{I}_{\cup l-1}} + 1 + n_r)n_r^2 + nn_r)$	$O(2n_r^2(m_{\mathcal{I}_{\cup l-1}} + m_{\mathbb{E}_l} + m_{\mathbb{I}_l}) - 2n_r^3/3 + n_r^2 + nn_r)$

the Schur normal equations (6 steps). However, both the projected normal equations ($\mathcal{N}$IPM-nf) and the least squares form ($\mathcal{N}$IPM-ls) require the calculation of a nullspace basis $Z_{\mathcal{A}_{l-1}}$ of $A_{\mathcal{A}_{\cup l-1}}$ in $O(2m^2_{\mathcal{A}_{\cup l-1}} n - 2/3 m^3_{\mathcal{A}_{\cup l-1}})$ at the beginning of the Newton's method of each level. Furthermore, the projections $A_{\mathbb{E}_l} Z_{\mathcal{A}_{l-1}}$ ($O(2m_{\mathbb{E}_l} nn_r)$), $A_{\mathbb{I}_l} Z_{\mathcal{A}_{l-1}}$ ($O(2m_{\mathbb{I}_l} nn_r)$) and $A_{\mathcal{I}_{\cup l-1}} Z_{\mathcal{A}_{l-1}}$ ($O(2m_{\mathcal{I}_{\cup l-1}} nn_r)$) are required. Nonetheless, with the additional effect of variable reduction due to the nullspace projections (Björck, 1996) we demonstrate in our evaluation (Sec. 5.1) that $\mathcal{N}$IPM-HLSP is equivalently fast or faster than the Schur normal form. This holds even for equality only problems where convergence is achieved within one Newton iteration without offsetting the overhead of the nullspace method over several Newton iterations.

4.2 $\mathcal{N}$IPM-HLSP's overall performance

A measure for the operational cost of $\mathcal{N}$IPM-HLSP dependent on the number of Newton iterations is given by

$$c_{\mathcal{N}\text{IPM-HLSP}} \coloneqq \tag{29}$$

$$\sum_{l=1}^{p} \iota_l \cdot c_{\text{IPM}}(n_r, m_{\mathcal{I}_{\cup l-1}} + m_{\mathbb{I}_l}, m_{\mathbb{E}_l}) + c_{\mathcal{N},\text{IPM}}(m_{\mathcal{A}_{l^*}}, m_{\mathcal{A}_l}, m_{\mathcal{I}_{\cup l}}, m_{l+1:p}, n_r)$$

The first component reflects the cost c_{IPM} of resolving the Newton's method of each level with ι_l Newton iterations as is listed in Tab. 1. The main computational load originates from solving the linear systems $\mathcal{N}$IPM-nf or $\mathcal{N}$IPM-ls. The cost is a function of the number of remaining variables n_r, the number of active constraints $m_{\mathcal{A}_{l^*}}$ and $m_{\mathcal{A}_l}$ of each respective level, the number of inactive constraints from the previous levels $m_{\mathcal{I}_{\cup l-1}}$, the number of equalities and inequalities of the current level $m_{\mathbb{I}_l}$ and $m_{\mathbb{E}_l}$ and finally the number of all equality and inequality constraints from the remaining levels $m_{l+1:p}$. Secondly, once the Newton's method of level l converges, the nullspace bases of the active-sets $\mathcal{A}_{l^*}$ and $\mathcal{A}_l$ have to be computed. Furthermore, the remaining levels $l+1, \ldots, p$ and inactive constraints $\mathcal{I}_{\cup l}$ need to be projected into it. This is reflected in the nullspace projection cost $c_{\mathcal{N},\text{IPM}}$.

In comparison, the pseudo-cost of the ASM in resolving the HLSP is composed as follows:

$$c_{\text{ASM}} \coloneqq \sum_{i=1}^{\iota} \sum_{l=1}^{p} c_{\mathcal{N},\text{ASM}}(m_{\mathcal{A}_l}, m_{l+1:p}, n_r) \tag{30}$$

ι are the number of active set iterations. Note that the cost of the ASM c_{ASM} is included in the projection cost $c_{\mathcal{N},\text{ASM}}$ (computation of a rank revealing QR decomposition of the current level with $m_{\mathcal{A}_l}$ active constraints; $m_{\mathcal{A}_l}$ also counts the activated constraints from higher priority levels 1 to $l-1$). As can be seen, the nullspace projection of the whole active set needs to be repeated for each active set iteration whereas $\mathcal{N}$IPM-HLSP needs to project the whole HLSP (including inactive constraints) only once per HLSP resolution. This leads to less operations for $\mathcal{N}$IPM-HLSP in case of large number of active set iterations for the ASM. Note that unlike the IPM, the ASM does not need to consider the $m_{\mathbb{I}_{\cup l}}$ inactive inequality constraint. This can be computationally advantageous depending on the problem's dimensions (see condition (28) as a reference).

5 Evaluation

We employ our solver $\mathcal{N}$IPM-HLSP ($\mathcal{N}$IPM-nf: projected normal form, $\mathcal{N}$IPM-ls: least-squares form) first on equality only constrained HLSP's of two levels $p = 2$ to gain some insights into its computational efficiency (Sec. 5.1). Especially, we are interested in how it fares in comparison to the Schur normal equations (IPM-snf: Schur normal form).

Secondly, we use the solver within the sequential hierarchical least-squares programming solver with trust region (S-HLSP) presented in Pfeiffer et al

(2023) to solve NL-HLSP's of the form (as a sub-form of LMOO)

$$\begin{aligned} \min_{u, v_{\mathbb{E}_l} v_{\mathbb{I}_l}} \quad & \frac{1}{2} \left\| v_{\mathbb{E}_l} \right\|^2 + \frac{1}{2} \left\| v_{\mathbb{I}_l} \right\|^2 \qquad l = 1, \ldots, p \qquad \text{(NL-HLSP)} \\ \text{s.t.} \quad & f_{\mathbb{E}_l}(u) \;=\; v_{\mathbb{E}_l} \\ & f_{\mathbb{I}_l}(u) \;\geq\; v_{\mathbb{I}_l} \\ & f_{\mathcal{A}_{\cup l-1}} = v^*_{\mathcal{A}_{\cup l-1}} \\ & f_{\mathcal{I}_{\cup l-1}} \geq v_{\mathcal{I}_{\cup l-1}} \end{aligned}$$

$f \in \mathbb{R}^m$ are non-linear task functions dependent on the variable vector $u \in \mathbb{R}^n$. $v \in \mathbb{R}^m$ are slack variables and $v^* \in \mathbb{R}^m$ are the optimal ones. S-HLSP repeatedly linearizes the NL-HLSP to HLSP 'sub-problems' around the current working point u (with the constraint matrices A and vectors b representing this linearization, for example as Jacobians and Hessians of f). The resulting step x from a HLSP sub-problem solver (for example $\mathcal{N}$IPM-HLSP) is then used to make an approximate step $u \leftarrow u + x$ within the non-linear model. The trust region radius limits the step $\|x\|_\infty < \rho$ in order to maintain validity of the linear approximation (HLSP) with respect to the original problem (NL-HLSP). In all following examples, the trust region radius constraint is formulated on level 0 of the HLSP's. S-HLSP employs a real-time capable trust-region radius adaptation method with local convergence properties. Other non-linear programming methods like filter methods with global convergence properties (Fletcher et al, 2002) require the recomputation of sub-problem steps x with different trust-region radii. Since this is not possible in real-time control due to computational limitations, every step is accepted. Instead, the trust region radius is only adapted (increased for valid step, decreased for invalid step) in the following control iteration (for more details see Pfeiffer et al (2018)).

Given the conceptual context of S-HLSP as in Pfeiffer et al (2023), we accordingly sample our simulations from different robot control scenarios. All constraints $f(u)$ therefore consist of either non-linear trigonometric functions or linear bound constraints on the robot variables u (joint velocities, joint torques, contact forces). This lets us design ill-posed problems commonly found in instantaneous robot control resulting from singular kinematic configurations of the robot. In these situations, first-order linearizations (Jacobians of the constraints $f(u)$) of the NL-HLSP become ill-posed or singular. Effective solutions to counter these situtations have been proposed, for example in form of regularization (Chiaverini, 1997) or higher-order approximations (Pfeiffer et al, 2023). For all robot control scenarios, one HLSP sub-problem is solved per control iteration.

We first evaluate a kinematic robot control example ($p = 5$, see Sec. 5.2). The evaluation is continued with a dynamic robot simulation depicting the effect of a larger number of variables and inequality constraints and ill-posed constraints due to singularities on the solver timings ($p = 6$, Sec. 5.3). The

evaluation is concluded with a push simulation in order to observe the distinctive difference in behavior between the ASM and IPM in case of a singular instance of a large change of the active set ($p = 5$, Sec. 5.4). For the simulations we use the HRP-2Kai humanoid robot with 38 degrees of freedom (DoF) (Kaneko et al, 2015).

For comparison, we solve the HLSP's directly with the ASM solver LexLS ((Dimitrov et al, 2015), https://github.com/jrl-umi3218/lexls), and in sequence (Kanoun et al, 2009) by the quadratic programming (QP) solvers OSQP (Stellato et al, 2020) based on the alternating direction of multipliers method (ADMM) and the proprietary barrier solver GUROBI (Gurobi Optimization, 2021). Unlike our dense solver $\mathcal{N}$IPM-HLSP, the two latter are sparse solvers. For OSQP, warm-starting of each level is disabled since the problem size can change over the course of the simulation due to constraint activations and deactivations. Since ADMM based solvers have the tendency to converge with only moderate accuracy, OSQP possesses the functionality to make an educated guess about the active set ('polishing'). We enable this functionality for the second (Sec. 5.3) and third simulation (Sec. 5.4). Otherwise, LexLS (iteration limit: 200), OSQP and GUROBI are used at their standard settings. In case of $\mathcal{N}$IPM-HLSP we limit the number of IPM iterations to 20. The projected normal equations ($\mathcal{N}$IPM-nf) are solved by a regularized LDLT decomposition (robust Cholesky decomposition with pivoting (Bunch and Parlett, 1971), plus diagonal regularization with a small weight of $1 \cdot 10^{-6}$). The least-squares form ($\mathcal{N}$IPM-ls) is solved by the rank revealing QR decomposition.

We depict the overall solver times, the required number of iterations (for OSQP both iterations and number of factorization updates as a dashed line), the time per Newton iteration (solver time divided by number of iterations; not for LexLS and OSQP) and the maximum KKT norm of all the levels (not for LexLS since it does not keep an update of the dual corresponding to inactive inequality constraints). The simulations are run on an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz processor with 32 Gb of RAM.

5.1 Solving equality only LSP's

Figure 1 shows the computation times for the different solvers when an equality only, dense LSP is solved ($p = 2$, $n = 60$ variables). The number of constraints of the first level $A_{\mathbb{E}_1} \in \mathbb{R}^{m_{\mathbb{E}_1}, n}$ is chosen to $m_{\mathbb{E}_1} = 0, 15, 30, 45, 60$. The number of constraints $m_{\mathbb{E}_2}$ on the second level ranges from 0 to 240.

It can be observed that the problems are solved the fastest for most problems by our solver $\mathcal{N}$IPM-nf. The LSP is solved at the same speed by IPM-snf when there are no equality constraints on the first level. In this case both problems are equivalent since IPM-snf only needs to do steps 2 and 6 of Tab. 1. Similarly, $\mathcal{N}$IPM-nf only needs to do steps 2 and 3. For the same problem constellation, both OSQP and GUROBI require much greater computation times due to computational overhead of handling dense problems sparsely. Therefore, for $m_{\mathbb{E}_1} > 0$ we do not report their results for the sake of better

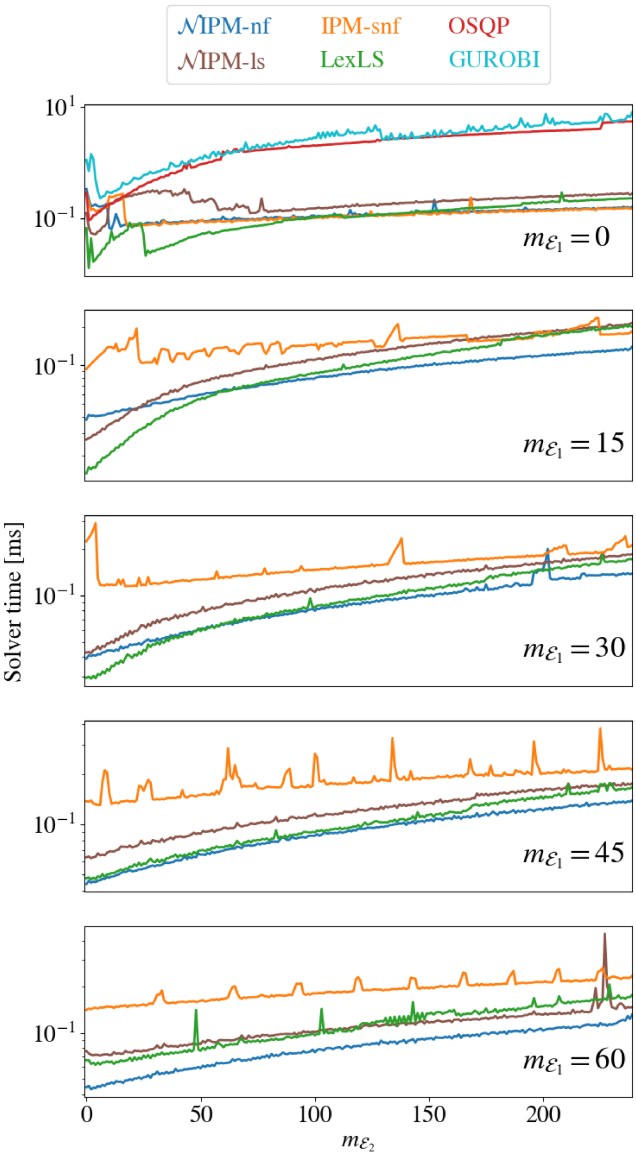

Fig. 1 Computation times of solving equality only, dense HLSP's ($p = 2$). The number of equality constraints on the first level are $m_{\mathbb{E}_1} = 0, 15, 30, 45, 60$ from top to bottom. The x-axis shows the number of equality constraints on the second level $m_{\mathbb{E}_2}$.

readability. Note that GUROBI includes bound constraints on the problem variables. Therefore any problem becomes an inequality constrained problem which requires more than one solver iteration. This is in contrast to $\mathcal{N}$IPM-nf, $\mathcal{N}$IPM-ls and LexLS which all converge after one iteration for equality only problems.

$\mathcal{N}$IPM-ls and LexLS are slower than $\mathcal{N}$IPM-nf except if both $m_{\mathbb{E}_1}$ and $m_{\mathbb{E}_2}$ are low. At the same time $\mathcal{N}$IPM-ls is slower than LexLS which re-uses the QR decomposition for resolving the primal step to compute the nullspace basis of the active constraints. Additionally, LexLS implements a very efficient in-place block decomposition with minimal memory access overhead.

As the number of equality constraints $m_{\mathbb{E}_1}$ increases, all nullspace method based solvers $\mathcal{N}$IPM-nf, $\mathcal{N}$IPM-ls and LexLS make up for the computation time of the nullspace basis computation $Z_{\mathbb{E}_1}$ and projection $A_{\mathbb{E}_2} Z_{\mathbb{E}_1}$ as increasingly smaller problems $n_r < n$ need to be solved on the second level. This is in contrast to IPM-snf where an increasingly more expensive second Cholesky decomposition of $A_{\mathbb{E}_1} C_2^{-1} A_{\mathbb{E}_1}$ needs to be computed. Eventually, $n_r = 0$ for $m_{\mathbb{E}_1} = n = 60$ which means that $\mathcal{N}$IPM-nf, $\mathcal{N}$IPM-ls and LexLS finish after solving the first level.

This evaluation shows that the projected normal equations ($\mathcal{N}$IPM-nf) are computationally equivalent or superior with respect to the Schur normal form (IPM-snf) in the limit case of equality constraints only. Since in the presence of inequalities the computational burden of the nullspace method for $\mathcal{N}$IPM-nf will be further offset over the solver iterations, we do not further consider IPM-snf throughout the remainder of the evaluation.

5.2 Kinematic robot control

Table 2 NL-HLSP A with $p = 5$ and $n = 38$ for the humanoid robot HRP-2.

l	Constraint $f_l(u) \geqq v_l$
1	Lower and upper joint angle limits (76 ineq.)
2	Two feet and left hand position and orientation (18 eq.)
3	Center of mass, medial and lateral (2 ineq.)
4	Right hand position (3 eq.)
5	Joint velocity regularization (38 eq.)

In this example, we demonstrate the computational efficiency of $\mathcal{N}$IPM-HLSP within the S-HLSP solver (Pfeiffer et al, 2023) for non-linear real-time robot control problems. We design a stretch demonstration with the HRP-2Kai robot according to the control hierarchy A given in Tab. 2. The constraints f can represent equality and inequality constraints which is indicated by the symbol $\geqq$. Each constraint's type (inequality, ineq.; equality, eq.) and dimension is given in brackets. The first level limits the robot's joint angles. The second level contains kinematic position constraints for both feet to be on the ground and the left hand to be grabbing onto a pole in front of the robot. On the next level the center of mass (CoM) is asked to stay within the 2D projected outer area of the feet on the ground. On the next level the right hand aims to reach an out-of-reach target on the upper right side of the robot. Finally, a regularization task on the joint velocities ensures full rank of the overall hierarchy.

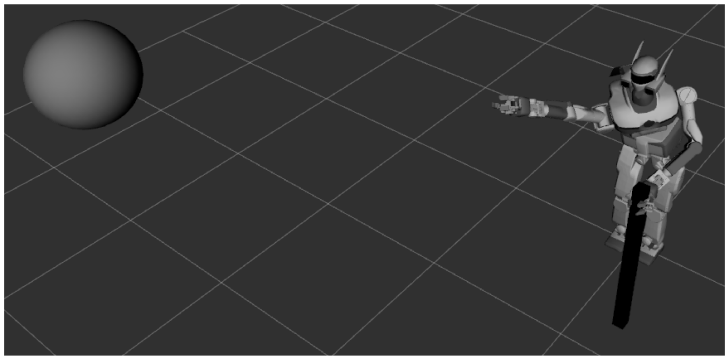

Fig. 2 HRP-2 stretching into kinematic singularity in order to move the right hand as close as possible to the ball. For the simulation in Sec. 5.3, the robot oscillates in a high frequency and low amplitude manner due to numerical instabilities resulting from unregularized singularities.

This leads to the robot taking the posture depicted in Fig. 2. By virtue of the hierarchical Newton's method (Pfeiffer et al, 2023), the robot control stays numerically stable despite the kinematic singularity of the stretched right arm.

The results are given in Fig. 3. The top graph shows the computation times of the five different solvers in every control time step while the second and third graph show the associated solver iterations and the computation time per solver iteration. The bottom graph shows the sum of the KKT norms of each priority level at convergence.

Among the IPM methods, our solver $\mathcal{N}$IPM-nf has the lowest solver time per solver iteration. The overall solver time is slightly higher than GUROBI since more iterations are required until convergence. All IPM solvers converge to a high degree with the KKT norm consistently being lower than 10^{-8}. This is in contrast to OSQP where in the majority of control iterations the KKT norm remains at levels of approximately 10^{-3}. Note that polishing is disabled as can be seen from the low number of factorization updates (dashed red line in second from top graph).

The S-HLSP sub-problems are solved fastest by LexLS. After an initial high number of active set changes in the first control iteration (8 active set iterations), the number of active set changes is limited to four, leading to low computation times (< 0.2 ms) due to warm-starting the active set of each control iteration with the previous one.

5.3 Dynamic robot control - A numerically unstable robot movement

The following simulation shows the behavior of the solvers if the problem to solve is scaled up in number of variables and constraints. Additionally, a singular task causes highly dynamic behavior with large changes of the active contact force, joint angle, velocity and torque constraints between the control

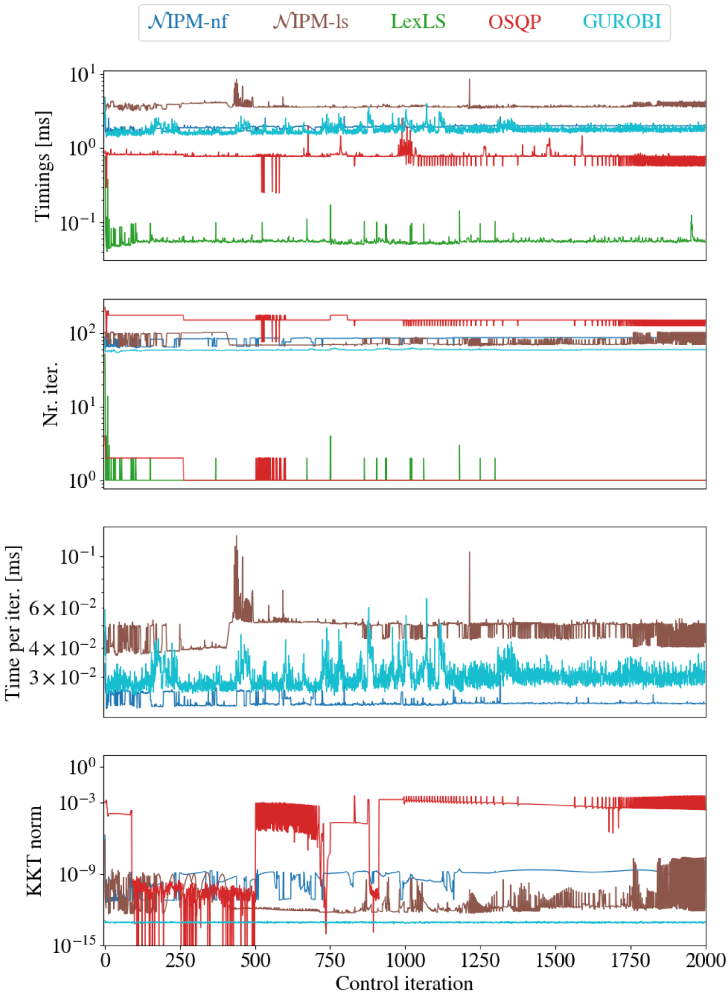

Fig. 3 NL-HLSP A, Tab. 2. $\mathcal{N}$IPM-nf exhibits lower computation times per solver iteration than GUROBI.

Table 3 NL-HLSP B with $p = 6$ and $n = 72$ for the humanoid robot HRP-2.

l	Constraint $f_l(u) \geqq v_l$
1	Lower and upper joint angle and velocity limits (152 ineq.)
	Joint torque limits (76 eq.)
	Unilateral contact forces (32 ineq.)
2	Two feet and left hand position and orientation (18 eq.)
3	Center of mass, medial and lateral (2 ineq.)
4	Right hand position (3 eq.)
5	Joint velocity regularization (38 eq.)
6	Contact force regularization (36 eq.)

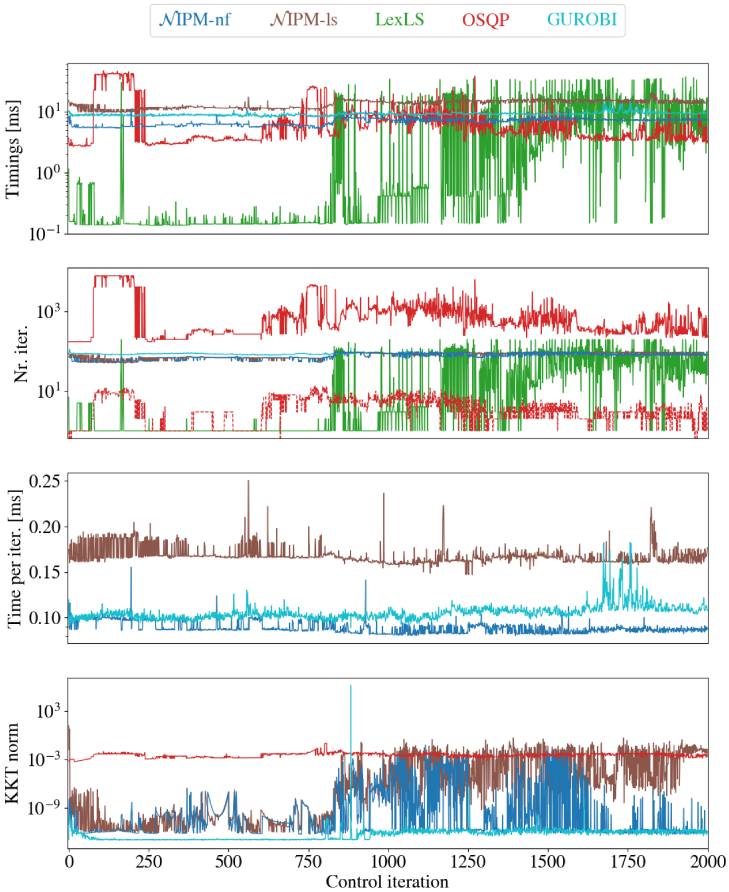

Fig. 4 NL-HLSP B, Tab. 3. The right arm starts stretching at control iteration 750 after which the robot behavior becomes numerically unstable.

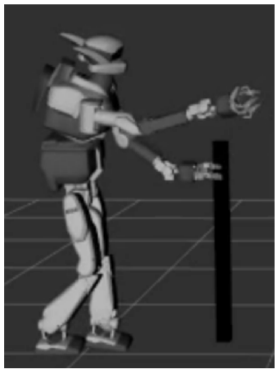

Fig. 5 NL-HLSP B, Tab. 3, LexLS. HRP-2 looses contact of the feet with the ground during the stretch task due to sub-optimal active-set identification resulting from constraint singularities. Also, the right arm is not properly stretching towards the goal on the far right top of the robot but swaying around uncontrollably.

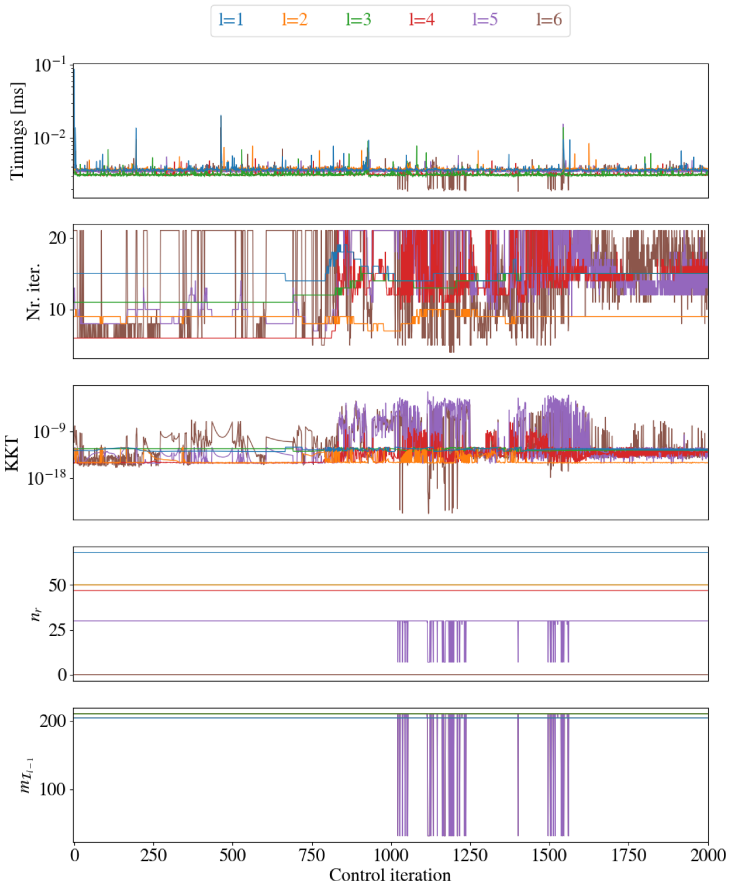

Fig. 6 NL-HLSP B, Tab. 3. $\mathcal{N}$IPM-nf's behavior for each priority level. n_r is the number of remaining variables on each respective priority level while m_{inact} corresponds to the number of inactive constraints $m_{\mathcal{I}_{\cup l-1}}$.

iterations. The complete control hierarchy B is given in Tab. 3 and the resulting robot posture is given in Fig. 2. We include the equation of motion with 3 contact points on the two feet and the left hand. The joint torques are subsequently substituted by the equation of motion (Herzog et al, 2016) such that we end up with 74 variables consisting of 38 joint velocities and 36 contact forces for the two feet and the left hand. The singular task is added to the hierarchy in form of a stretch task trying to reach for a Cartesian position outside of the workspace of the robot. If such a task is not regularized (Chiaverini, 1997), it leads to numerical instabilities in form of high frequency oscillations in presence of a trust region joint velocity constraint (Pfeiffer et al, 2023).

The computation times of the five solvers are given in Fig. 4. $\mathcal{N}$IPM-nf resolves the hierarchy in about 7 ms while the robot is numerically stable. Once the right arm starts stretching at control iteration 750 there is a slight uptick in computation times of about 2 ms due to an increased number of

Newton iterations of the levels containing the ill-posed arm stretch task and below ($l \geq 4$), see Fig. 6. The fifth level suffers from bad convergence with a maximum KKT norm of around 0.06 (14.5 for $\mathcal{N}$IPM-ls). Nonetheless, for both $\mathcal{N}$IPM-nf and $\mathcal{N}$IPM-ls the robot continues to behave as expected with both feet keeping contact to the ground and high frequency oscillations especially on the right arm.

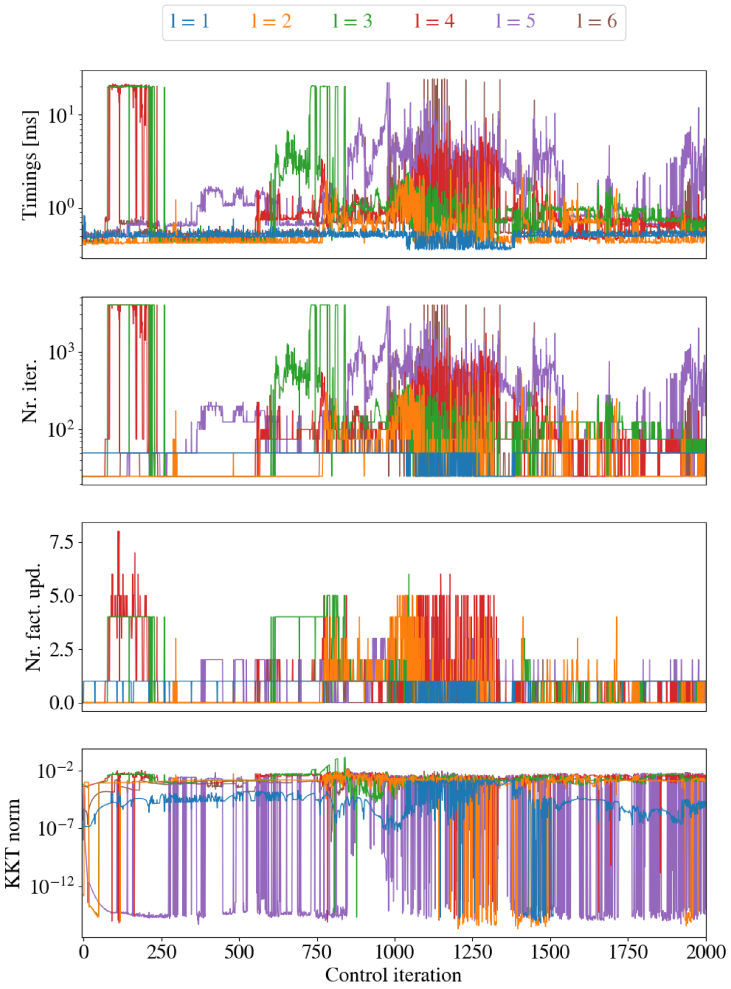

Fig. 7 NL-HLSP B, Tab. 3. OSQP's behavior for each priority level. 'Nr. fact. upd.' (in linear scale) indicates the number of (expensive) factorization updates per priority level.

GUROBI constantly shows a very high convergence rate (except for a single peak of 1.5×10^6 due to numerical difficulties). GUROBI efficiently leverages sparsity of the constraint matrices which is not possible for $\mathcal{N}$IPM-HLSP due to dense nullspace projections. Nonetheless, $\mathcal{N}$IPM-nf exhibits approximately

the same computation time per solver iteration as GUROBI. This can be explained by the large degree of variable elimination of 32% from 74 to 50 variables after the first priority level, see second graph from the bottom of Fig. 6.

LexLS fails to find a reasonable approximation of the true active set within the given limit of 200 iterations in many control iterations. The high frequency oscillations of the robot due to the kinematic singularities are associated with large changes of the active-set. This leads to the feet losing contact and the right arm swaying uncontrollably, see Fig. 5. Also, such a large number of active set iterations causes LexLS to violate the real-time constraint with computation times of about 37 ms.

OSQP converges with a maximum KKT norm of about 0.14 on level 3. While the hierarchy is mostly resolved around 2 ms faster than by IPM-ASM, there are many instances where OSQP requires both a high number of iterations and factorization updates. This leads to computation times of up to 42 ms (8175 iterations, 9 factorization updates). We observed that the main challenge of solving HLSP by the ADMM is to maintain feasibility of the subsequent LSP's due to the inherent moderate convergence accuracy of the ADMM. As can be seen from the KKT norms in Fig. 7, there is always a level where the polishing process fails such that the corresponding level converges at about 10^{-3}. This negatively influences the determination of the optimal infeasible points. While this may be acceptable for LSP's it is problematic for HLSP's since wrongly determined violations $v^*_{\mathcal{A}_{\cup l-1}}$ render the subsequent priority levels infeasible and eventually lead to solver failure. We explicitly recalculate the violations $v_{\mathbb{E}_l} = A_{\mathbb{E}_l} x - b_{\mathbb{E}_l}$ and $v_{\mathbb{I}_l} = A_{\mathbb{I}_l} x - b_{\mathbb{I}_l}$ at convergence of each priority level which can slightly mitigate this circumstance. Promising progress in terms of accuracy of the ADMM has been documented for example in Bambade et al (2022).

5.4 Dynamic robot control - Handling a push

While the previous example in Sec. 5.3 is interesting from a numerical point of view, there exist various regularization methods to prevent such numerically unstable behavior (Chiaverini, 1997; Pfeiffer et al, 2023). This last example therefore looks at a situation which may occur in a real robot situation, namely handling a push. For this we use the control hierarchy B given in Tab. 3 with slight modifications. The level $l = 4$ is omitted such that $p = 5$. Instead of an inequality constrained CoM, the CoM position is fully controlled by equality constraints in all three spatial directions ($l = 3$).

The robot is asked to lower its CoM. During this process at control iteration 400, a push from behind of magnitude 800 N is applied to the upper body. As can be seen from Fig. 8, this leads to a singular instance of a large number of 132 active set iterations which takes LexLS approximately 50 ms to resolve. This comes from an unfavorable interplay between the trust region and the joint torque limits caused by possibly ill conditioned equations of motion (Maciejewski, 1990).

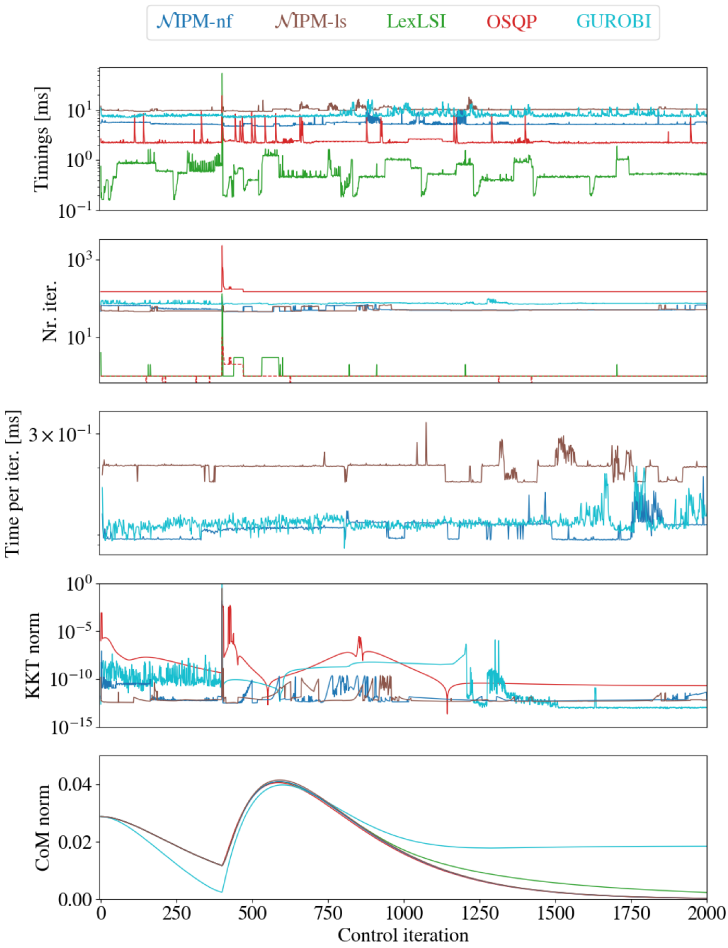

Fig. 8 NL-HLSP B (Tab. 3) without right hand task. At control iteration 400 a push from behind of magnitude 400 N is applied to the robot body.

This also influences the convergence behavior of OSQP negatively. While OSQP manages successful polishing in most control iterations it does not do so during the push and exposes a higher KKT norm in these instances. The high number of iterations and factorization updates require around 19 ms to resolve.

In contrast, such behavior is observed only to a small degree for the IPM. As can be seen from the second and third graph from the bottom, the number of iterations of $\mathcal{N}$IPM-nf, $\mathcal{N}$IPM-ls and GUROBI are barely influenced. Similarly to the previous example, the KKT norm increases sharply in this control instance for all IPM solvers but can be attributed to bad convergence of lower priority levels corresponding to the regularization task. As can be seen from the bottom graph, the CoM trajectory remains physically reasonable as it recovers from the push.

The push leads to the CoM of the robot being shifted to the front and increases the error norm of the CoM task as can be seen from the bottom graph. The difference in CoM behavior seen for GUROBI is due to a regularization term on the joint velocities and contact forces, which amounts to the reformulated cost function $\min._{x,v_{\mathbb{E}_l},v_{\mathbb{I}_l}} \quad \frac{1}{2}10^{-5}\|x\|^2+\frac{1}{2}\|v_{\mathbb{E}_l}\|^2+\frac{1}{2}\|v_{\mathbb{I}_l}\|^2$, $l = 1, \ldots, p$ for the HLSP. Otherwise we observed numerical difficulties.

6 Conclusion

With this work we have formulated the IPM for HLSP resulting in the solver $\mathcal{N}$IPM-HLSP based on the nullspace method. It requires only a single decomposition of the KKT system per Newton iteration instead of two. This proves computationally equivalent or superior with respect to the IPM in Schur normal form. Our simulations showed that $\mathcal{N}$IPM-HLSP resolves ill-posed problems without significant variations in solver iterations or computation times. In contrast, the ASM tends to require high number of active set iterations in dynamic or numerically unstable control situations due to ill-posed constraint matrices. Our $\mathcal{N}$IPM-HLSP formulation therefore may be preferred to the ASM if solver predictability is regarded more important than very fast computation times but limited by instances of unsuccessful active set searches.

While $\mathcal{N}$IPM-HLSP is reasonably efficient, we see further potential for algorithmic improvements, for example by a sparse solver formulation, heuristically reducing the number of inactive constraints or restricting the number of Newton iterations as seen in Wang and Boyd (2010). The last point may require a primal feasible formulation (primal-barrier interior-point method) of our solver. We further believe that our formulation of the nullspace method based IPM for HLSP is not only relevant for instantaneous robotic control but for example in model predictive control (MPC). Special attention requires its block diagonal structure which may be exploited for computational efficiency for example by tailored nullspace bases.

7 Declarations

Part of this work was supported by New York University NSF grants 1925079, 1825993.

Appendix A Recursive computation of the Lagrange multipliers associated with active constraints

(21) can be rewritten to

$$A^T_{\mathcal{A}_{\cup l-1}} \Delta\lambda_{\mathcal{A}_{\cup l-1}} = \begin{bmatrix} A_{\mathcal{A}_{\cup l-1}} \\ A_{\mathcal{I}_{\cup l-1}} \\ A_{\mathbb{I}_l} \\ A_{\mathbb{E}_l} \end{bmatrix}^T \begin{bmatrix} -\lambda_{\mathcal{A}_{\cup l-1}} \\ D \\ E \\ A_{\mathbb{E}_l}(x+\Delta x) - b_{\mathbb{E}_l} \end{bmatrix} \tag{A1}$$

with

$$D \coloneqq -\lambda_{\mathcal{I}_{\cup l-1}} - W^{-1}_{\mathcal{I}_{\cup l-1}} (\lambda_{\mathcal{I}_{\cup l-1}} \odot (b_{\mathcal{I}_{\cup l-1}} - A_{\mathcal{I}_{\cup l-1}}(x+\Delta x)) + \sigma_{\mathcal{I}_{\cup l-1}} \mu_{\mathcal{I}_{\cup l-1}} e) \tag{A2}$$

and

$$\begin{aligned} E \coloneqq &\left(I + (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1} W_{\mathbb{I}_l}\right) A_{\mathbb{I}_l} \Delta x + A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} - w_{\mathbb{I}_l} \\ &+ (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1} (\sigma_{\mathbb{I}_l} \mu_{\mathbb{I}_l} e + w_{\mathbb{I}_l} \odot (A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} - w_{\mathbb{I}_l})) \end{aligned} \tag{A3}$$

The Lagrange multipliers can be calculated recursively by

$$N^T_{\mathcal{A}_{\cup j-1}} A^T_{\mathcal{A}_j} \Delta\lambda_{\mathcal{A}_j} = N^T_{\mathcal{A}_{\cup j-1}} \left(\begin{bmatrix} A_{\mathcal{I}_{\cup l-1}} \\ A_{\mathbb{I}_l} \\ A_{\mathbb{E}_l} \end{bmatrix}^T \begin{bmatrix} D \\ E \\ A_{\mathbb{E}_l}(x+\Delta x) - b_{\mathbb{E}_l} \end{bmatrix} - \sum_{k=j+1}^{l-1} A^T_{\mathcal{A}_k} \Delta\lambda_{\mathcal{A}_k} \right) \tag{A4}$$

with $j = l-1, \ldots, 1$. For nullspace basis Z_l of the form (22), the QR decompositions of $A_{\mathcal{A}_l} N_{\mathcal{A}_{\cup l-1}}$ can be reused.

Appendix B Mehrotra's predictor corrector algorithm for NIPM-HLSP

First, a decomposition of the projected normal equations (NIPM-nf) or the least squares form (NIPM-ls) is computed. Note that this only needs to be done once per Newton iteration. It is then used to first calculate the affine scaling step Δz_{aff}, Δx_{aff}, $\Delta w^{\text{aff}}_{\mathbb{I}_l}$, $v^{\text{aff}}_{\mathbb{I}_l}$, $\Delta w^{\text{aff}}_{\mathcal{I}_{\cup l-1}}$ and $\Delta\lambda^{\text{aff}}_{\mathcal{I}_{\cup l-1}}$ with

$$\begin{aligned} &\sigma_{\mathcal{I}_{\cup l-1}} = 0 \quad \text{and} \quad \sigma_{\mathbb{I}_l} = 0 \\ &F_{\text{aff}} \coloneqq \lambda_{\mathcal{I}_{\cup l-1}} - W^{-1}_{\mathcal{I}_{\cup l-1}} \lambda_{\mathcal{I}_{\cup l-1}} (A_{\mathcal{I}_{\cup l-1}} x - b_{\mathcal{I}_{\cup l-1}}) \end{aligned} \tag{B5}$$

$$G_{\text{aff}} \coloneqq b_{\mathbb{I}_l} + w_{\mathbb{I}_l} - A_{\mathbb{I}_l} x - (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1} W_{\mathbb{I}_l} (A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} - w_{\mathbb{I}_l})$$

Line search α_{aff} is conducted in order to keep the dual feasible with $w_{\mathbb{I}_l} + \alpha_{\text{aff}} \Delta w_{\mathbb{I}_l}^{\text{aff}} \geq 0$, $v_{\mathbb{I}_l} + \alpha_{\text{aff}} \Delta v_{\mathbb{I}_l}^{\text{aff}} \leq 0$, $w_{\mathcal{I}_{\cup l-1}} + \alpha_{\text{aff}} \Delta w_{\mathcal{I}_{\cup l-1}}^{\text{aff}} \geq 0$ and $\lambda_{\mathcal{I}_{\cup l-1}} + \alpha_{\text{aff}} \Delta \lambda_{\mathcal{I}_{\cup l-1}}^{\text{aff}} \geq 0$.

With this information we calculate the corrector step Δz, Δx, $\Delta v_{\mathbb{I}_l}$, $w_{\mathbb{I}_l}$, $\Delta w_{\mathcal{I}_{\cup l-1}}$ and $\Delta \lambda_{\mathcal{I}_{\cup l-1}}$ with

$$\begin{aligned}
\sigma_{\mathcal{I}_{\cup l-1}} &\coloneqq (\mu_{\mathcal{I}_{\cup l-1}}^{\text{aff}} / \mu_{\mathcal{I}_{\cup l-1}})^3 \quad \text{and} \quad \mu_{\mathcal{I}_{\cup l-1}} \coloneqq \lambda_{\mathcal{I}_{\cup l-1}}^T w_{\mathcal{I}_{\cup l-1}} / m_{\mathcal{I}_{\cup l-1}} \\
\mu_{\mathcal{I}_{\cup l-1}}^{\text{aff}} &\coloneqq (\lambda_{\mathcal{I}_{\cup l-1}} + \alpha_{\text{aff}} \Delta \lambda_{\mathcal{I}_{\cup l-1}}^{\text{aff}})^T (w_{\mathcal{I}_{\cup l-1}} + \alpha_{\text{aff}} \Delta w_{\mathcal{I}_{\cup l-1}}^{\text{aff}}) / m_{\mathcal{I}_{\cup l-1}} \\
\sigma_{\mathbb{I}_l} &\coloneqq (\mu_{\mathbb{I}_l}^{\text{aff}} / \mu_{\mathbb{I}_l})^3 \quad \text{and} \quad \mu_{\mathbb{I}_l} = -v_{\mathbb{I}_l}^T w_{\mathbb{I}_l} / m_{\mathbb{I}_l} \\
\mu_{\mathbb{I}_l}^{\text{aff}} &\coloneqq (v_{\mathbb{I}_l} + \alpha_{\text{aff}} \Delta v_{\mathbb{I}_l}^{\text{aff}})^T (w_{\mathbb{I}_l} + \alpha_{\text{aff}} \Delta w_{\mathbb{I}_l}^{\text{aff}}) / m_{\mathbb{I}_l}
\end{aligned} \tag{B6}$$

For the corrector step we furthermore have

$$\begin{aligned}
K_{w_{\mathcal{I}_{\cup l-1}}, l} &\coloneqq v_{\mathbb{I}_l} \odot w_{\mathbb{I}_l} + \Delta v_{\mathbb{I}_l}^{\text{aff}} \odot \Delta w_{\mathbb{I}_l}^{\text{aff}} + \sigma_{\mathbb{I}_l} \mu_{\mathbb{I}_l} e \\
K_{w_{\mathbb{I}_l}, l} &\coloneqq \lambda_{\mathcal{I}_{\cup l-1}} \odot w_{\mathcal{I}_{\cup l-1}} + \Delta \lambda_{\mathcal{I}_{\cup l-1}}^{\text{aff}} \odot \Delta w_{\mathcal{I}_{\cup l-1}}^{\text{aff}} - \sigma_{\mathcal{I}_{\cup l-1}} \mu_{\mathcal{I}_{\cup l-1}} e
\end{aligned} \tag{B7}$$

such that

$$\begin{aligned}
D_{\text{cor}} \coloneqq & -\lambda_{\mathcal{I}_{\cup l-1}} - W_{\mathcal{I}_{\cup l-1}}^{-1} (\lambda_{\mathcal{I}_{\cup l-1}} \odot (b_{\mathcal{I}_{\cup l-1}} - A_{\mathcal{I}_{\cup l-1}} (x + \Delta x)) \\
& - \Delta \lambda_{\mathcal{I}_{\cup l-1}}^{\text{aff}} \odot \Delta w_{\mathcal{I}_{\cup l-1}}^{\text{aff}} + \sigma_{\mathcal{I}_{\cup l-1}} \mu_{\mathcal{I}_{\cup l-1}} e) \\
F_{\text{cor}} \coloneqq & \lambda_{\mathcal{I}_{\cup l-1}} + W_{\mathcal{I}_{\cup l-1}}^{-1} (\lambda_{\mathcal{I}_{\cup l-1}} \odot (b_{\mathcal{I}_{\cup l-1}} - A_{\mathcal{I}_{\cup l-1}} x) \\
& - \Delta \lambda_{\mathcal{I}_{\cup l-1}}^{\text{aff}} \odot \Delta w_{\mathcal{I}_{\cup l-1}}^{\text{aff}} + \sigma_{\mathcal{I}_{\cup l-1}} \mu_{\mathcal{I}_{\cup l-1}} e) \\
E_{\text{cor}} \coloneqq & -\left(I + (V_{1,\mathcal{I}} - W_{1,\mathcal{I}})^{-1} W_{\mathbb{I}_l}\right) A_{\mathbb{I}_l} - A_{\mathbb{I}_l} x + b_{\mathbb{I}_l} + w_{\mathbb{I}_l} - (V_{1,\mathcal{I}} - W_{1,\mathcal{I}})^{-1} \\
& (\sigma_{\mathbb{I}_l} \mu_{\mathbb{I}_l} e + \Delta v_{\mathbb{I}_l}^{\text{aff}} \odot \Delta w_{\mathbb{I}_l}^{\text{aff}} + w_{\mathbb{I}_l} \odot (A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} - w_{\mathbb{I}_l})) \\
G_{\text{cor}} \coloneqq & b_{\mathbb{I}_l} + w_{\mathbb{I}_l} - A_{\mathbb{I}_l} x - (V_{\mathbb{I}_l} - W_{\mathbb{I}_l})^{-1} \\
& (\sigma_{\mathbb{I}_l} \mu_{\mathbb{I}_l} e + \Delta v_{\mathbb{I}_l}^{\text{aff}} \odot \Delta w_{\mathbb{I}_l}^{\text{aff}} + w_{\mathbb{I}_l} \odot (A_{\mathbb{I}_l} x - b_{\mathbb{I}_l} - w_{\mathbb{I}_l}))
\end{aligned} \tag{B8}$$

Appendix C Algorithm

Here we detail the implementation of $\mathcal{N}$IPM-HLSP which can also be found at https://www.github.com/pfeiffer-kai/NIPM-HLSP. Algorithm 1 describes the overall routine. As input it requires a HLSP which contains information about the number of priority levels p, the number of variables n and the linear constraints represented by A and b. Note that the active and inactive sets $\mathcal{A}$ and $\mathcal{I}$ and the optimal slacks v^* are not known yet. Alg. 2 summarizes the predictor and corrector step calculation. Alg. 3 details the active set compositions and projections. The symbol _ is a placeholder for the different variables $x, v_{\mathbb{I}_l}, w_{\mathbb{I}_l}, w_{\mathcal{I}_{\cup l-1}}$ and $\lambda_{\mathcal{I}_{\cup l-1}}$.

Algorithm 1 $\mathcal{N}$IPM-HLSP

Input: HLSP
Output: x, $\lambda_{\mathcal{A}_{\cup p}}$

$r = 0$
$\iota = 0$
$N = I$
$\tilde{A} = A_p$
for $l = 1 : p$ **do**
  while $\|\tilde{K}_l\|_2 > \epsilon$ & $\iota <$ maxIter **do**
    $\alpha^{\text{aff}}, \Delta x^{\text{aff}}, \Delta v_{\mathbb{I}_l}^{\text{aff}}, \Delta w_{\mathbb{I}_l}^{\text{aff}}, \Delta w_{\mathcal{I}_{\cup l-1}}^{\text{aff}}, \Delta \lambda_{\mathcal{I}_{\cup l-1}}^{\text{aff}} \leftarrow$ `solve`('predictor')
    $\alpha, \Delta x, \Delta v_{\mathbb{I}_l}, \Delta w_{\mathbb{I}_l}, \Delta w_{\mathcal{I}_{\cup l-1}}, \Delta \lambda_{\mathcal{I}_{\cup l-1}} \leftarrow$ `solve`('corrector', $\alpha^{\text{aff}}, \Delta\_^{\text{aff}}$)
    Make step with $\_ = \_ + \alpha \Delta\_$ for new x, $w_{\mathbb{I}_l}$, $v_{\mathbb{I}_l}$, $w_{\mathcal{I}_{\cup l-1}}$, $\lambda_{\mathcal{I}_{\cup l-1}}$
    $\iota \leftarrow \iota + 1$
  end while
  $r, N, \tilde{A} \leftarrow$ `project`('inactive', $r, w_{\mathcal{I}_{\cup l-1}}, \lambda_{\mathcal{I}_{\cup l-1}}, N, \tilde{A}$)
  if $r \geq n$ **then return** x, $\lambda_{\mathcal{A}_{\cup p}}$ **end if**
  $r, N, \tilde{A} \leftarrow$ `project`('this level', $r, w_{\mathcal{I}_l}, v_{\underline{\mathcal{I}}_l}, N, \tilde{A}$)
  if $r \geq n$ **then return** x, $\lambda_{\mathcal{A}_{\cup p}}$ **end if**
end for
return x, $\lambda_{\mathcal{A}_{\cup p}}$

Algorithm 2 `solve`

Input: type, α^{aff}, $\Delta\_^{\text{aff}}$
Output: α, Δx, $\Delta v_{\mathbb{I}_l}$, $\Delta w_{\mathbb{I}_l}$, $\Delta w_{\mathcal{I}_{\cup l-1}}$, $\Delta \lambda_{\mathcal{I}_{\cup l-1}}$

if type = 'predictor' **then**
  Solve $\mathcal{N}$IPM-nf or $\mathcal{N}$IPM-ls for Δz using B5
else
  Solve $\mathcal{N}$IPM-nf or $\mathcal{N}$IPM-ls for Δz using (B6), (B7) and (B8) with α^{aff} and $\Delta\_^{\text{aff}}$
end if
$\Delta x = N_{\mathcal{A}_{\cup l-1}} \Delta z$
Calculate $\Delta v_{\mathbb{I}_l}$, $\Delta w_{\mathbb{I}_l}$, $\Delta w_{\mathcal{I}_{\cup l-1}}$, $\Delta \lambda_{\mathcal{I}_{\cup l-1}}$ with Δx
Line search for α such that (17) and (18) are fulfilled
return α, Δx, $\Delta v_{\mathbb{I}_l}$, $\Delta w_{\mathbb{I}_l}$, $\Delta w_{\mathcal{I}_{\cup l-1}}$, $\Delta \lambda_{\mathcal{I}_{\cup l-1}}$

References

Amaran S, Sahinidis N (2012) Global optimization of nonlinear least-squares problems by branch-and-bound and optimality constraints. Top 20:1–19. https://doi.org/10.1007/s11750-011-0178-8

Ansary MAT (2023) A newton-type proximal gradient method for nonlinear multi-objective optimization problems. Optimization Methods and Software

Algorithm 3 `project`

Input: type, r, w, λ, N, $\tilde{A}$
Output: r, N, $\tilde{A}$

$\mathcal{A}_l = \{\}$
$\mathbb{I}_l = \{\}$
if type = 'inactive' **then**
  for all $c \in \mathcal{I}_{\cup l-1}$ **do**
    if $w(c) < \xi$ and $\lambda(c) > \xi$ **then**
      $\mathcal{A}_{l^*} \leftarrow \{c, \mathcal{A}_{l^*}\}$
      $\mathcal{I}_{\cup l-1} \leftarrow \mathcal{I}_{\cup l-1} \setminus c$
    end if
  end for
  $\hat{r}, Z_{\mathcal{A}_{l^*}} \leftarrow \mathcal{N}(A_{\mathcal{A}_{l^*}} N_{\mathcal{A}_{\cup l-1}})$
  $N_{\mathcal{A}_{\cup l^*}} \leftarrow N_{\mathcal{A}_{\cup l-1}} Z_{\mathcal{A}_{l^*}}$
  $\tilde{A} \leftarrow \tilde{A} Z_{\mathcal{A}_l^*}$
else
  $\mathcal{A}_l \leftarrow \mathbb{E}_l$
  for all $c \in \mathcal{I}_l$ **do**
    if $w_{\mathcal{I}}(c) < \xi$ and $v_{\mathcal{I}}(c) < -\xi$ **then**
      $\mathcal{A}_l \leftarrow \{c, \mathcal{A}_l\}$
    else
      $\mathcal{I}_l \leftarrow \{c, \mathcal{I}_l\}$
    end if
  end for
  $\hat{r}, Z_{\mathcal{A}_l} \leftarrow \mathcal{N}(A_{\mathcal{A}_l} N_{\mathcal{A}_{\cup l^*}})$
  $N_{\mathcal{A}_{\cup l}} \leftarrow N_{\mathcal{A}_{\cup l^*}} Z_{\mathcal{A}_l}$
  $\tilde{A} \leftarrow \tilde{A} Z_{\mathcal{A}_l}$
end if
$r = r + \hat{r}$
return r, $N_{\mathcal{A}_{\cup l^*}}$ or $N_{\mathcal{A}_{\cup l}}$

0(0):1–21

Bambade A, El-Kazdadi S, Taylor A, et al (2022) PROX-QP: Yet another Quadratic Programming Solver for Robotics and beyond. In: RSS 2022 - Robotics: Science and Systems, New York, United States, URL https://hal.inria.fr/hal-03683733

Bartlett R, Biegler L (2006) Qpschur: A dual, active-set, schur-complement method for large-scale and structured convex quadratic programming. Optimization and Engineering 7:5–32. https://doi.org/10.1007/s11081-006-6588-z

Bartlett RA, Wachter A, Biegler LT (2000) Active set vs. interior point strategies for model predictive control. In: Proceedings of the 2000 American

Control Conference. ACC (IEEE Cat. No.00CH36334), pp 4229–4233 vol.6

Benzi M, Golub GH, Liesen J (2005) Numerical solution of saddle point problems. Acta Numerica 14:1–137. https://doi.org/10.1017/S0962492904000212

Björck A (1996) Numerical Methods for Least Squares Problems. Society for Industrial and Applied Mathematics, https://doi.org/10.1137/1.9781611971484

Bunch JR, Parlett BN (1971) {D}irect methods for solving symmetric indefinite systems of linear equations. {SIAM} {J}ournal of {N}umerical {A}nalysis 8(4):639–655. https://doi.org/10.1137/0708060

Chiaverini S (1997) Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. IEEE Transactions on Robotics and Automation 13(3):398–410. https://doi.org/10.1109/70.585902

Coleman TF (1984) Large Sparse Numerical Optimization. Springer-Verlag, Berlin, Heidelberg

De Lasa M, Hertzmann A (2009) Prioritized optimization for task-space control. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009 3(2):5755–5762. https://doi.org/10.1109/IROS.2009.5354341

Dimitrov D, Sherikov A, Wieber PB (2015) Efficient resolution of potentially conflicting linear constraints in robotics, URL https://hal.inria.fr/hal-01183003

Domahidi A, Zgraggen AU, Zeilinger MN, et al (2012) Efficient interior point methods for multistage problems arising in receding horizon control. In: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), pp 668–674

Escande A, Mansard N, Wieber PB (2014) Hierarchical quadratic programming: Fast online humanoid-robot motion generation. The International Journal of Robotics Research 33(7):1006–1028. https://doi.org/10.1177/0278364914521306

Evtushenko Y, Posypkin M (2014) A deterministic algorithm for global multiobjective optimization. Optimization Methods and Software 29(5):1005–1019

Fathi Hafshejani S, Peyghami MR, Jahromi A (2020) An efficient primal-dual interior point method for linear programming problems based on a new kernel function with a finite exponential-trigonometric barrier term. Optimization and Engineering 21. https://doi.org/10.1007/s11081-019-09436-3

Fletcher R, Leyffer S, Toint PL (2002) On the global convergence of a filter–sqp algorithm. SIAM Journal on Optimization 13(1):44–59. https://arxiv.org/abs/https://doinot.org/10.1137/S105262340038081X

Frison G, Diehl M (2020) Hpipm: a high-performance quadratic programming framework for model predictive control∗∗this research was supported by the german federal ministry for economic affairs and energy (bmwi) via eco4wind (0324125b) and dyconpv (0324166b), and by dfg via research unit for 2401. IFAC-PapersOnLine 53(2):6563–6569. https://doi.org/https://doi.org/10.1016/j.ifacol.2020.12.073, 21th IFAC World Congress

Gill P, Hammarling S, Murray W, et al (1986) LSSOL (Version 1.0): a Fortran Package for Constrained Linear Least-Squares and Convex Quadratic Programming. User's Guide. Defense Technical Information Center

Gill P, Murray W, Saunders M, et al (1989) Practical anti-cycling procedure for linearly constrained optimization. Mathematical Programming 45:437–474. https://doi.org/10.1007/BF01589114

Gill PE, Murray W, Saunders MA, et al (1987) Maintaining lu factors of a general sparse matrix. Linear Algebra and its Applications 88-89:239–270

Guennebaud G, Jacob B, et al (2010) Eigen v3. http://eigen.tuxfamily.org

Gurobi Optimization L (2021) Gurobi optimizer reference manual

Hammarling S, Lucas C (2008) Updating the qr factorization and the least squares problem

Herzog A, Righetti L, Grimminger F, et al (2014) Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. IEEE International Conference on Intelligent Robots and Systems pp 981–988. https://doi.org/10.1109/IROS.2014.6942678

Herzog A, Rotella N, Mason S, et al (2016) Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. Autonomous Robots 40(3):473–491. https://doi.org/10.1007/s10514-015-9476-6

Holder A (2006) Partitioning multiple objective optimal solutions with applications in radiotherapy design. Optimization and Engineering 7. https://doi.org/10.1007/s11081-006-0352-2

Kaneko K, Morisawa M, Kajita S, et al (2015) Humanoid robot hrp-2kai - improvement of hrp-2 towards disaster response tasks -. https://doi.org/10.1109/HUMANOIDS.2015.7363526

Kanoun O, Lamiraux F, Wieber PB, et al (2009) Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots. 2009 IEEE International Conference on Robotics and Automation (May):2939–2944. https://doi.org/10.1109/ROBOT.2009.5152293

Kanoun O, Lamiraux F, Wieber PB (2011) Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks. IEEE Trans on Robotics 27(4):785–792. https://doi.org/10.1109/TRO.2011.2142450

Karmarkar N (1984) A new polynomial-time algorithm for linear programming. Combinatorica 4:373–395

Kuindersma S, Permenter F, Tedrake R (2014) An efficiently solvable quadratic program for stabilizing dynamic locomotion. In: IEEE International Conference on Robotics and Automation, Hong Kong, China

Lai L, Fiaschi L, Cococcioni M, et al (2021) Handling priority levels in mixed pareto-lexicographic many-objective optimization problems. In: Ishibuchi H, Zhang Q, Cheng R, et al (eds) Evolutionary Multi-Criterion Optimization. Springer International Publishing, Cham, pp 362–374

Li M, Zhang M, Huang K, et al (2021) A new primal-dual interior-point method for semidefinite optimization based on a parameterized kernel function. Optimization and Engineering 22. https://doi.org/10.1007/s11081-020-09516-9

Maciejewski AA (1990) Dealing with the ill-conditioned equations of motion for articulated figures. IEEE Computer Graphics and Applications 10(3):63–71. https://doi.org/10.1109/38.55154

Mehrotra S (1992) On the implementation of a primal-dual interior point method. SIAM Journal on Optimization 2:575–601. https://doi.org/10.1137/0802028

Nesterov Y, Nemirovskii A (1994) Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics, https://doi.org/10.1137/1.9781611970791

Nocedal J, Wright SJ (2006) Numerical Optimization, 2nd edn. Springer, New York, NY, USA

Pandala AG, Ding Y, Park HW (2019) qpswift: A real-time sparse quadratic program solver for robotic applications. IEEE Robotics and Automation Letters 4(4):3355–3362. https://doi.org/10.1109/LRA.2019.2926664

Petelin G, Antoniou M, Papa G (2021) Multi-objective approaches to ground station scheduling for optimization of communication with satellites. Optimization and Engineering 24. https://doi.org/10.1007/s11081-021-09617-z

Pfeiffer K, Escande A, Kheddar A (2018) Singularity resolution in equality and inequality constrained hierarchical task-space control by adaptive nonlinear least squares. IEEE Robotics and Automation Letters 3(4):3630–3637. https://doi.org/10.1109/LRA.2018.2855265

Pfeiffer K, Escande A, Gergondet P, et al (2023) The hierarchical newton's method for numerically stable prioritized dynamic control. IEEE Transactions on Control Systems Technology pp 1–14. https://doi.org/10.1109/TCST.2023.3234492

Rao CV, Wright SJ, Rawlings JB (1998) Application of interior-point methods to model predictive control. JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS 99:723–757. https://doi.org/10.1023/A:1021711402723

Sauk B, Ploskas N, Sahinidis N (2020) Gpu parameter tuning for tall and skinny dense linear least squares problems. Optimization Methods and Software 35(3):638–660. https://doi.org/10.1080/10556788.2018.1527331

Sherali HD, Soyster AL (1983) Preemptive and nonpreemptive multi-objective programming: Relationship and counterexamples. Journal of Optimization Theory and Applications 39:173–186. https://doi.org/10.1007/BF00934527

Stellato B, Banjac G, Goulart P, et al (2020) OSQP: an operator splitting solver for quadratic programs. Mathematical Programming Computation 12(4):637–672. https://doi.org/10.1007/s12532-020-00179-2

Vanderbei R (2013) Linear Programming: Foundations and Extensions. International Series in Operations Research & Management Science, Springer US

Vanderbei RJ (1999) Loqo:an interior point code for quadratic programming. Optimization Methods and Software 11(1-4):451–484. https://doi.org/10.1080/10556789908805759

Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106:25–57

Wang Y, Boyd S (2010) Fast model predictive control using online optimization. IEEE Transactions on Control Systems Technology 18(2):267–278. https://doi.org/10.1109/TCST.2009.2017934

Yang Y (2022) A polynomial time infeasible interior-point arc-search algorithm for convex optimization. Optimization and Engineering https://doi.org/10.1007/s11081-022-09712-9

Ye Y, Todd M, Mizuno S (1994) An o(nl)-iteration homogeneous and self-dual linear programming algorithm. Mathematics of Operations Research - MOR 19. https://doi.org/10.1287/moor.19.1.53