Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Integrated Hierarchical Decision-Making in Inverse Kinematic Planning and Control

Kai Pfeiffer*, Quan Zhang*, Yuqing Chen*, Gordon Owusu Boateng*, Yuquan Wang†, Vincent Bonnet‡§, Abderrahmane Kheddar¶

*School of Advanced Technology, Xi'an Jiatong Liverpool University, China †X Square Robot, China
‡ CNRS, IPAL, Singapore §LAAS-CNRS, Université de Toulouse, France
¶CNRS-University of Montpellier, LIRMM, Montpellier, France

***Abstract*—This work presents a novel and efficient nonlinear programming framework that tightly integrates hierarchical decision-making with whole-body inverse kinematic planning and control. Decision-making plays a central role in many aspects of robotics, from sparse inverse kinematic control with a minimal number of joints, to inverse kinematic planning while simultaneously selecting a discrete end-effector location from multiple candidates. Current approaches often rely on heavy computations using mixed-integer nonlinear programming, separate decision-making from inverse kinematics (some times approximated by reachability methods), or employ efficient but less versatile ℓ_1-norm formulations of linear sparse programming, without addressing the underlying nonlinear problem formulations. In contrast, the proposed sparse hierarchical nonlinear programming solver is efficient, versatile, and accurate by exploiting sparse hierarchical structure and leveraging the ℓ_0-norm which is rarely used in robotics. The solver efficiently tackles complex nonlinear hierarchical decision-making problems previously unaddressed in the literature, such as inverse kinematic planning with simultaneous prioritized selection of end-effector locations from a large set of candidates, or inverse kinematic control with simultaneous selection of bi-manual grasp locations on a randomly rotated box.**

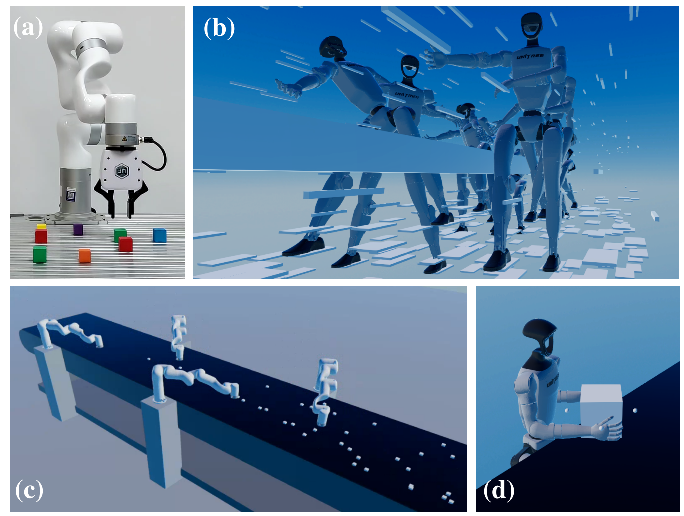

Fig. 1: The newly proposed sparse hierarchical nonlinear programming framework (SH-NLP) enables the following use-cases: **(a)** Subsequent pick-and-place of several objects with UFactory's `xarm6` (Sec. VIII-A). **(b)** Unitree's `G1` plans IK postures while simultaneously choosing one of 200 possible discrete locations for the higher-priority foot and the lower-priority hand placement tasks. 10 postures are computed by iteratively solving 763 SHQP's within 2.12 s by the proposed solver $\mathcal{N}$QP (Sec. VIII-C). Both **(a)** and **(b)** represent planning applications (SHIK-P). **(c)** An array of four UFactory `xarm6`'s equipped with vacuum tubes clears a conveyor belt loaded with produce like cashew nuts. **(d)** A `G1` places its hands on two of the four sides of a randomly rotated box. In both **(c)** and **(d)**, the robot reacts to changes of the objects with immediate and continuous decision-making within the IK control loop (SHIK-C; see Sec. VIII-F). Rendering: `Raisim` [16].

I. Introduction

Robotics often involves aspects of decision-making, for example in inverse kinematics (IK) control with a minimal number of active joints (e.g., in highly redundant robotic system embedded with brakes at the actuator level). Exploiting sparsity in such problems can lead to more economical [12] and human-like motions [3]. Furthermore, IK planning may involve multiple feasible scenarios, for instance when a robot may place its end-effector at any of several possible discrete locations without compromising the overall objective. While optimization methods like mixed-integer nonlinear programming (MINLP) can solve such decision-making problems to global optimality [4, 28], they are computationally taxing. A more efficient approach is to rely on sparse optimization methods, as proposed in [29] for footstep planning with robustness to sparse gradients, unlike reinforcement learning based methods [33]. While the use of the ℓ_1-norm in [29] is effective for this purpose, some inaccuracies remain, and redundant allocation of end-effector locations was reported. Furthermore, kinematic reachability approximations are used while the robot's whole-body IK are treated separately. This introduces the risk that a selected end-effector location may be unreachable if the approximation is not conservative enough, or overly conservative if chosen too restrictively, thus under-utilizing the robot's workspace.

This work addresses these limitations by developing a novel framework for hierarchical nonlinear decision-making, which, to the best of our knowledge, is the first in robotics (and beyond) to address such problems. The following use-cases are considered, as illustrated in Fig. 1:

- *Planning (SHIK-P):* directly solving nonlinear sparse hierarchical IK and decision-making planning problems.

- *Control (SHIK-C):* solving sparse hierarchical IK and decision-making instantaneously for real-time control.

The main contributions of this work are as follows:

- We propose a novel and efficient sequential sparse hierarchical quadratic programming solver (S-SHQP) for sparse hierarchical nonlinear programs (SH-NLP). This enables robot IK planning and control with autonomous and reliable hierarchical decision-making, for example when choosing among multiple Cartesian locations with different prioritized end-effector tasks. Prioritization overcomes the difficulty of weighting tasks against logarithmic surrogate functions of the used ℓ_0-norm formulations.
- Our method enables real-time decision-making in milliseconds for complex whole-body robotic systems due to linear dependency on the number of candidates. This enables efficient selection from a large set of candidate locations without relying on reachability approximations. By simultaneously considering whole-body IK, an immediate certificate of kinematic reachability is obtained.
- Unlike sparse programming methods based on the ℓ_1-norm, our ℓ_0-norm based formulation is versatile and agnostic to the specific formulation of the decision-functions. For example, it enables decision-making with vector-valued functions which have been turned into scalars by least-squares.
- Our method enables parallel decision-making from the same set of candidates while avoiding double-allocation. This is important in robotics as end-effectors like the left and right feet typically choose from the same set of candidate locations.
- A wide range of whole-body IK planning and control applications with integrated hierarchical decision-making for both humanoids and manipulators are presented, while reiterating on the advantages of the ℓ_0-norm over the ℓ_1-norm and MINLP.

S-SHQP extends nonlinear hierarchical least-squares programming (ℓ_2-norm [23, 21]), which is solved via sequential hierarchical least-squares programming using a hierarchical step filter (HSF) and a trust-region constraint [21]. Here, we introduce the required adaptations for ℓ_0-norm programming and hierarchical decision-making (see Sec. V and VI). In its main computation step, S-SHQP iteratively solves sparse hierarchical quadratic programs (SHQP). While standard QP solvers such as PIQP [27] or MOSEK [2] can be used, they do not exploit the special structure of SHQP. Our proposed solver, $\mathcal{N}$QP, leverages this structure for improved computational efficiency (Sec. VII). For example, SHIK-C with selection among 100 possible locations on Unitree's `G1` robot is run at a control loop time of 1.5 ms, compared to 2.2 ms and 8.3 ms for PIQP and MOSEK, respectively (Sec. VIII-E).

The remainder of this letter is organized as follows. Sec. II reviews related work, while Sec. III recalls the basics of sparse programming based decision-making. This is extended to hierarchical decision-making via SH-NLP, which is outlined in Sec. IV. Sec. V presents the S-SHQP solver, followed in Sec. VI by additions for parallel and hierarchical decision-making. Sec. VII details the interior-point method for SHQP, and Sec. VIII evaluates the algorithm on benchmark functions and robot planning and control problems.

NOMENCLATURE

$\chi/x \in \mathbb{R}^n, \nu/v, \tau/t$	SH-NLP and SHQP primal variable equivalents
$\bullet^* \in \mathbb{R}^n$	Optimal SH-NLP or SHQP value $\bullet$ found for level l
$\mathbb{C}_l \in \mathbb{R}^{\|\mathbb{C}_l\|}$	Constraint set composed of equality and inequality constraints $\mathbb{C}_l = \{\mathbb{E}_l, \mathbb{I}_l\}$ of level l
$\mathcal{I}_{\cup l-1}/\mathcal{A}_{\cup l-1}$	Set union $\mathcal{I}_{\cup l-1} := \mathcal{I}_1 \cup \ldots \cup \mathcal{I}_{l-1}$ of inactive / active constraints
$\mathcal{T}_l \in \mathbb{R}^{2\|\mathbb{C}_l\|}$	Set of ℓ_0-norm auxiliary constraints
$\omega_{\mathbb{C}_l}^{(*)} \in \mathbb{R}^{\|\mathbb{C}_l\|}$	ℓ_0-weights; (*): optimal SH-NLP value found for level l
$f_{i/j/k} \in \mathbb{R}$	Nonlinear functions associated with constraint sets $i \in \mathbb{C}_l$ / $j \in \mathcal{I}_{\cup l-1}$ / $k \in \mathcal{A}_{\cup l-1}$
$(f_i) \in \mathbb{R}^{\|\mathbb{C}_l\|}$	Vector accumulation $(f_i) := (f_i)_{i \in \mathbb{C}_l}$ or $(f_i) := [f_{i_1} \ \ldots]^T$
$a_i \in \mathbb{R}^n, b_i \in \mathbb{R}$	Linearization of non-linear function f_i
$N_l \in \mathbb{R}^{n \times n_r}$	Nullspace of matrix $[a_{k_1}^T \ \ldots]^T$ of active constraints $\mathcal{A}_l$
$\tilde{a}_i \in \mathbb{R}^{n_r}$	Nullspace projected vector $\tilde{a}_i^T = a_i^T N_{l-1}$
$\breve{b}_i \in \mathbb{R}$	Value b_i augmented with terms from nullspace projections
$\lambda \in \mathbb{R}^m$	Lagrange multipliers
$w \in \mathbb{R}^m$	Slack variables for inequality constraints
$\gamma_i^+/\nu_i^+, \gamma_i^-/\nu_i^- \in \mathbb{R}$	Lagrange multipliers / slack variables for upper and lower auxiliary ℓ_0 constraints; summarized by $\gamma_i^\pm, \nu_i^\pm$
$\Psi_i, \Gamma_i \in \mathbb{R}$	Auxiliary variables $\Psi_i := (w_i/\lambda_i)$ and $\Gamma_i := (\nu_i/\gamma_i)$

II. RELATED WORK

The concept of sparse programming originates from statistical analysis and decision-making problems such as compressed sensing [6] and portfolio optimization [32]. In these domains, unconstrained linear regression problems are solved to fit model parameters to observed data. By introducing sparsity-promoting regularization, the optimization is biased toward selecting a minimal subset of parameters sufficient to explain the data. Sparsity is typically defined via the ℓ_0-norm, which counts the number of non-zero entries in a vector χ. Since the ℓ_0-norm is a pseudo-norm (violating properties such as homogeneity, i.e., $\|\alpha\chi\|_{\ell_0} \neq |\alpha|\|\chi\|_{\ell_0}$ for $\alpha \neq 1$) its direct minimization requires a combinatorial search that becomes intractable for large-scale problems. To overcome this, the ℓ_0-norm is often approximated by (weighted) ℓ_1-norm minimization [6], which can be efficiently solved using interior-point methods [18].

In robotics, sparse control has been explored primarily through sequential optimization schemes. At each iteration, a sparse convex subproblem is solved by linearizing constraints related to robot dynamics and kinematics around the current operating point [23]. In [12], the sparsity-inducing properties of the simplex method are exploited to achieve sparse robot control with minimal joint activation via ℓ_1-norm regularization. The methods proposed in [25] and [15] formulate sparse control and contact force selection as NP-hard mixed-integer linear programs, while in [26] sparse ℓ_1-norm terms are incorporated at multiple levels of a hierarchical controller. However, these approaches rely on ℓ_1-norm relaxations and do not exploit true ℓ_0-norm formulations, as considered here, which are required for versatile decision-making.

Although the aforementioned methods enable sparse control, they lack nonlinear solver components such as Fletcher *et al.*'s filter-based globalization technique [10]. This limits their applicability to nonlinear SHIK-P or sparse optimal control problems. Differential dynamic programming with linear ℓ_1-norm regularization has been used to achieve sparse robot joint engagement [8]; however, it cannot enforce sparsity in nonlinear cost functions. In [14], ℓ_0-norm regularization is used for the same purpose, but relies on a continuous surrogate function. The optimization-based formulation in [34] addresses nonlinear constrained sparse programming; however, nonlinear constraints are restricted to equalities, and sparsity is limited to constraints that are linear in the decision variables.

In summary, while sparse methods have been widely explored in control and estimation, their application to hierarchical, nonlinear robotic problems, particularly those requiring simultaneous autonomous location selection and IK reasoning, remains largely unexplored. This motivates the development of a dedicated nonlinear solver capable of handling sparse hierarchical programming with ℓ_0-norm formulations.

III. Sparse programming based decision-Making

Sparse programming can be applied to select a subset of feasible constraints from a group of similar-type ones for efficient *decision-making*. In robotics, this concept naturally fits autonomous location selection, where a robot must identify a single feasible end-effector location from multiple potential candidates, e.g., from sensor detection or candidate grasp poses. We denote such a group of selection constraints as $\mathbb{S} \subseteq \mathbb{C}$. Each constraint in $\mathbb{S}$ is expressed as the following ℓ_2-norm entry:

$$f_s(\chi_{\mathbb{S}}) = \|g_{\mathbb{S}}(\chi_{\mathbb{S}}) - g_{d,s}\|_2^2, \qquad s \in \mathbb{S} \tag{1}$$

The function $g_{\mathbb{S}}(\chi_{\mathbb{S}}) \in \mathbb{R}^{m_g}$ is representative for all entries in $\mathbb{S}$ and depends on a subset of variables $\chi_{\mathbb{S}} \subseteq \chi$, e.g., the joint variables of a specific kinematic chain (e.g., right arm of a humanoid). The desired value $g_{d,s} \in \mathbb{R}^{m_g}$ is constant and can represent a Cartesian candidate location for the end-effector. Minimizing $\|(f_s)_{s\in\mathbb{S}}\|_{\ell_0}$ (with $(f_s)_{s\in\mathbb{S}} := [f_{s_1} \ \dots]^T \in \mathbb{R}^{|\mathbb{S}|}$) with respect to the ℓ_0-norm enables a unique selection –or *decision-making*– within the group $\mathbb{S}$, as formalized below.

Proposition 1: If all desired values $g_{d,s}$ are distinct and at least one constraint in $\mathbb{S}$ is feasible, then $\min_{\chi_{\mathbb{S}}}\|(f_s)_{s\in\mathbb{S}}\|_{\ell_0} = |\mathbb{S}|-1$ is a global solution.

Proof: From (1) and the given assumptions, only one entry s can satisfy $f_s = 0$ with $g_{\mathbb{S}} = g_{d,s}$. ■

The identity $\|(f_s)_{s\in\mathbb{S}}\|_{\ell_1} = \sum_{s\in\mathbb{S}}|f_s| = \sum_{s\in\mathbb{S}} f_s$ shows that the ℓ_1-norm does not affect the decision-functions of the form (1), which emphasizes the versatility of the ℓ_0-norm, as it is agnostic to their specific formulation.

IV. Sparse Hierarchical nonlinear Programming

We introduce a novel optimization formalism for hierarchical decision-making coined SH-NLP. The formulation is rooted in optimization theory but directly corresponds to hierarchical robotic planning and control problems and forms the basis of the proposed sparse hierarchical inverse kinematics with autonomous location selection (Sec. VI):

$$\begin{aligned} \min_{\chi,(\nu_i)_{i\in\mathbb{C}_l}} \quad & \|(\nu_i)_{i\in\mathbb{C}_l}\|_{\ell_0}, \qquad l = 1,\dots,p \\ \text{s.t.} \quad & f_i(\chi) \geqq \nu_i, \quad i \in \mathbb{C}_l \\ & f_j(\chi) \geq 0, \quad j \in \mathcal{I}_{\cup l-1} \\ & f_k(\chi) = \nu_k^*, \quad k \in \mathcal{A}_{\cup l-1} \end{aligned} \tag{SH-NLP}$$

The function $f_i(\chi) \in \mathbb{R}$ represents a nonlinear task constraint from the set $i \in \mathbb{C}_l$, e.g., an end-effector position error. The function is parameterized by the variable vector $\chi \in \mathbb{R}^n$ (e.g., the robot's joint state). The constraint set $\mathbb{C}_l = \{\mathbb{E}_l, \mathbb{I}_l\}$ includes equalities and inequalities, denoted by the symbol $\geqq$. The slack vector $(\nu_i)_{i\in\mathbb{C}_l}$ allows controlled relaxation of these constraints, e.g., to handle unreachable position targets. At each priority level l, the goal is to find the optimal feasible point $(\nu_i^*)_{i\in\mathbb{C}_l} = 0$ (all constraints satisfied) or the optimal infeasible point $(\nu_i^*)_{i\in\mathbb{C}_l} \neq 0$ (with the fewest possible violations). Constraints that were active or inactive at previous levels (1 to $l-1$) must remain consistent, represented by the active and inactive sets $\mathcal{A}_{\cup l-1}$ and $\mathcal{I}_{\cup l-1}$, respectively. The symbol $\cup$ denotes the union over previous levels: $\mathcal{I}_{\cup l-1} := \mathcal{I}_1 \cup \dots \cup \mathcal{I}_{l-1}$. Once $(\nu_i^*)_{i\in\mathbb{C}_l}$ is identified, all equality and violated inequality constraints are added to the new active set $\mathcal{A}_l$ with corresponding optimal slack $(\nu_i^*)_{i\in\mathbb{C}_{\cup l-1}}$, while satisfied inequalities form the inactive set $\mathcal{I}_l$. The process continues recursively for the next level $l \leftarrow l+1$.

In the following, we omit the specification of set memberships $i \in \mathbb{C}_l$, $s \in \mathbb{S}$, $j \in \mathcal{I}_{\cup l-1}$, and $k \in \mathcal{A}_{\cup l-1}$ for brevity.

Directly solving SH-NLP is combinatorial and thus intractable for large-scale problems. Following the approach in [6] (originally designed for unconstrained linear programs), we adopt a continuous approximation of the ℓ_0-norm minimization through a logarithmic surrogate, leading to:

$$\begin{aligned} \min_{\chi,(\nu_i),(\tau_i)} \quad & \sum\nolimits_{i\in\mathbb{C}_l} \log(\tau_i + \xi), \qquad l = 1,\dots,p \\ \text{s.t.} \quad & -\tau_i \leq \nu_i \leq \tau_i \\ & f_i(\chi) \geqq \nu_i \\ & f_j(\chi) \geq 0 \\ & f_k(\chi) = \nu_k^* \end{aligned} \tag{2}$$

The summation of entry-wise logarithms, $\sum \log(\cdot)$, rewards entries in (ν_i) with increasingly negative cost as they approach zero, thus promoting sparsity. The small constant $\xi > 0$ ensures numerical stability. This emphasizes the benefit of prioritization, as weighting strategies are difficult to tune in this regime. The auxiliary variable τ_i and the corresponding bound constraints $\mathcal{T}_l$ on ν_i provide a smooth and continuous approximation of the discontinuous absolute value function

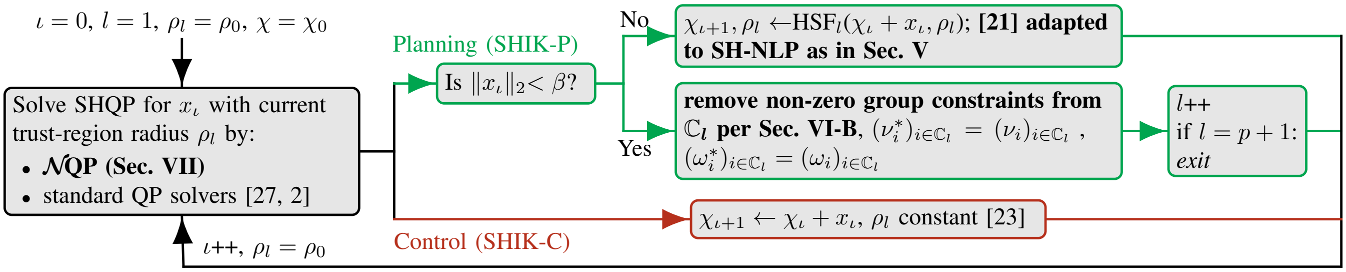

Fig. 2: Symbolic overview of the sequential sparse hierarchical quadratic programming (S-SHQP) algorithm with trust region and hierarchical step-filter (HSF) [21], adapted from the SQP step-filter of Fletcher *et al.* [10]. The framework efficiently solves sparse hierarchical nonlinear programs (SH-NLP) across p priority levels. New contributions are printed in bold.

$|\nu_i|$. This formulation maintains the hierarchical structure of the original sparse problem while enabling efficient numerical solutions through nonlinear programming.

V. Sequential Sparse Hierarchical Quadratic Programming

We solve (2) by our nonlinear sparse solver S-SHQP, which is based on the principles of sequential quadratic programming (see [20], Ch. 18). As depicted in Fig. 2, at each iteration ι, a SHQP subproblem approximates (2) around the current state $\begin{bmatrix}\chi_\iota^T & \nu_\iota^T & \tau_\iota^T\end{bmatrix}^T$, which is solved for the local step $\begin{bmatrix}x_\iota^T & v_\iota^T & t_\iota^T\end{bmatrix}^T$ [20, 6, 23]. SHQP results from the first-order conditions of (2) when Newton's method is applied to them:

$$\begin{aligned}
\min_{z_l,(v_i),(t_i)} \quad & \textstyle\sum_{i\in\mathbb{C}_l} \omega_i t_i, \quad l = 1,\dots,p \qquad \text{(SHQP)}\\
& +0.5(x_{l-1}^* + N_{l-1}z_l)^T H_l (x_{l-1}^* + N_{l-1}z_l)\\
\text{s.t.} \quad & -t_i \leq v_i \leq t_i\\
& \tilde{a}_i^T z_l - \breve{b}_i \geqq v_i\\
& \tilde{a}_j^T z_l - \breve{b}_j \geq 0\\
& \|x_l\|_\infty < \rho
\end{aligned}$$

The weights $\omega_i > 0$ are defined as:

$$\omega_i = 1/t_i + \xi \tag{3}$$

Setting $(\omega_i) = 1$ recovers the ℓ_1-norm case of SH-NLP.

The change of variables $x_l = x_{l-1}^* + N_{l-1}z_l$ (with $x := x_p$) represents a projection into the nullspace of the active constraints $\mathcal{A}_{\cup l-1}$. Here, x_{l-1}^* is the optimal primal of levels 1 to $l-1$, and $N_{l-1} \in \mathbb{R}^{n\times n_r}$ is a nullspace basis of the vectors $a_k \in \mathbb{R}^n$ ($a_k^T N_{l-1} = 0$; $N_0 = I$). The vector $z_l \in \mathbb{R}^{n_r}$ is the projected primal of level l. This projection eliminates both active constraints and already occupied variables such that the number of remaining variables is $n_r < n$.

The hierarchical Lagrangian Hessian H_l [23] captures second-order terms of the constraints, and is crucial for ensuring linear constraint qualifications (LCQ) of the vectors $a_i = \text{grad} f_i(\chi_\iota)$ [10] (with projected counterparts $\tilde{H}_l = N_{l-1}^T H_l N_{l-1}$ and $\tilde{a}_i^T = a_i^T N_{l-1}$). The term is highlighted in gray to emphasize the difference between Gauss-Newton (no H_l) and full Newton steps (with H_l). The choice between the two depends on linearized constraint feasibility $\|(v_i^*)\|_2 \leq / > \epsilon$ [22], where $\epsilon > 0$ is a numerical threshold. In practice, Newton's method is needed in singular robot tasks in the presence of unreachable targets.

The value b_i is defined as $b_i := f_i(\chi_\iota)$, and $\breve{b}_i$ denotes the nullspace projection offset from previous priority levels: $\breve{b}_i := b_i - a_i^T x_{l-1}^*$. The primal x_l is restricted within a trust region of radius ρ, which enforces local validity of the SHQP model at χ_ι (omitted hereon for brevity). S-SHQP differs from NL-HLSP, which solves a hierarchical least-squares problem similar to SHQP, but with quadratic cost $\|(v_i)\|_2^2$ and no auxiliary constraints $\mathcal{T}_l$.

Depending on the operational mode, the algorithm proceeds as follows:

1) Planning: The proposed step is accepted or rejected by the HSF [10, 21], which evaluates both feasibility and optimality. The filter is updated with the values $\sum_{i\in\mathbb{C}_l} \log(|f_i^{\geq 0}| + \epsilon)$ for each level l, requiring new points to strictly improve over prior ones. The expression $f_i^{\geq 0} := \max(0, f_i^T)$ maintains positive values of the inequality constraints $i \in \mathbb{I}_l$. Model validity is checked by comparing the nonlinear values $f_i^{\geq 0}$ to the linearized approximations $\omega_i(a_i^T x - b_i)^{\geq 0}$. The trust-region radius is then adapted (expanded for accepted steps, reduced otherwise). Once the S-SHQP has converged with $\|x_\iota\|_2 < \beta$ ($\beta > 0$), the optimal slacks are stored. Sparse constraints s with $\nu_s^* = 0$ in group $s \in \mathbb{S}$ (Sec. VI) trigger removal of non-sparse ones $\nu_s \neq 0$ from $\mathbb{C}_l$. Corresponding weights are fixed as $\omega_s^* = \omega_{s,\iota}$, while ω_s is continuously updated according to (3) for unsolved levels ($l+1$ to p). Feasibility of (2) is guaranteed only at full HSF convergence.

2) Control: Each step is directly accepted. With a properly tuned constant trust-region radius, the resulting state remains approximately feasible for (2), which makes the update suitable for closed-loop robotic control [23].

VI. Parallel Hierarchical Decision-Making

A. Avoiding Double Allocation in parallel decision-making

In robotics, it may be necessary for several selection groups $\mathbb{S}_{c=1,2,\dots} \subset \mathbb{S}$ to choose from the same set of potential candidates. To avoid *double allocation*, e.g., when both robot feet or arms attempt to select the same candidate, the following modified version of the weights (ω_{s_c}) with $s_c \in \mathbb{S}_c$ is used:

$$(\tilde{\omega}_{s_c}) = (\phi_{s_c}) \odot (\omega_{s_c}), \quad (\phi_{s_c}) = \min((f_{s_c}))\,|(f_{s_c})|^{-1} \tag{4}$$

The symbol $\odot$ symbolizes the entry-wise multiplication between two vectors. The operation $|(f_{s_c})|^{-1}$ returns the reciprocal of each individual entry of the vector. The masks ϕ_{s_c} are computed in sequence and represent the minimum value of the constraints in each set $\mathbb{S}_c$. In order to prevent multiple groups from converging to the same candidate, each minimum index of each set $\mathbb{S}_c$ is chosen uniquely over all sets, while already identified indices from other sets are set to zero in (ϕ_{s_c}).

Since in each S-SHQP iteration the mask $(\phi_{\mathbb{S}_\mathbb{C}})$ is treated as constant, the resulting SHQP subproblem becomes a less accurate local approximation of the original problem (2). However, the HSF robustly compensates for this, and no adverse effects on convergence were observed in practice.

B. Implications for Newton's Method and Hierarchical Decision-making

In the case of Newton's method (see Sec. V), the Lagrangian Hessian is non-zero and full-rank over the variables $\chi_\mathbb{S}$ of a group $\mathbb{S}$. This guarantees solver convergence via LCQ, but also prevents reuse of these variables in lower-priority levels. As a result, an unnecessary switch of Newton's method can degrade accuracy on lower priority levels in *hierarchical decision-making*. To mitigate this, we leverage feasible entries within $\mathbb{S}$ whenever available.

- Run the HSF for level l and identify the optimally infeasible point (ν_s^*) of group $\mathbb{S}$.
- If a feasible constraint s exists (i.e., $\nu_s \leq \zeta$, $\zeta > 0$), add it to the active set $\mathcal{A}_l$ and discard all infeasible ones.
- If no feasible constraint exists, add one representative constraint from $\mathbb{S}$ to $\mathcal{A}_l$ and discard the others.

Discarding constraints is justified by Theorem 1: once a single constraint s of $\mathbb{S}$ is included in the active set, all other slacks in $\mathbb{S}$ are uniquely determined. Specifically, if $\|g_\mathbb{S}(\chi_\mathbb{S}) - g_{d,s_1}\|_2^2 = \nu_{s_1}$ holds for some $\chi_\mathbb{S}$ and $s_1 \in \mathbb{S}$, then $\|g_\mathbb{S}(\chi_\mathbb{S}) - g_{d,s_2}\|_2^2 = \nu_{s_2}$ is implicitly defined for any other $s, s_2 \in \mathbb{S}$. Selecting the zero constraint s_1 (the feasible one) avoids activating Newton's method since the switching condition $\|\nu_{s_1}^*\| \ngtr \epsilon$ remains unsatisfied.

VII. An Interior-Point Method for SHQP

Efficient solvers have been proposed to solve HLSP as subproblems of NL-HLSP, using either active-set [9] or interior-point methods [24]. In these hierarchical formulations, each level l is solved sequentially, and its least-squares objective becomes a linear constraint for the lower levels $l+1$ to p. However, this cascaded scheme cannot be directly applied to SHQP, since its objective includes the linear term $\sum_{i\in\mathbb{C}_l}\omega_i t_i$, which prevents reformulation into a least-squares problem. This limitation can be resolved by the following observation:

Theorem 1: v_i is strictly upper or lower bounded at t_i or $-t_i$ for $\omega_i > 0$ and the infeasible case $v_i \neq 0$. Otherwise $t_i = v_i = 0$.

As a result, the inactive opposing constraint will never be active and can be omitted in $\mathcal{I}_{\cup l-1}$. For example, if $t_i - v_i = 0$ holds for constraint i, then $t_i + v_i \geq 0$ always holds. Furthermore, this allows one not only to store the optimal slack v_i^*, but also to set the optimal auxiliary variable $t_i^* = |v_i^*|$. Since the term $\omega_i t_i^*$ in the cost function is now constant, it can be omitted. The cost function of each level is then a least-squares problem $(\|R_l(x_{l-1}^* + N_{l-1}z_l)\|_2^2)$ and becomes the linear constraint $R_l N_{l-1} z_l = R_l x_{l-1}^*$ for all subsequent levels $l+1$ to p. Here, R_l is a factor of the semi-positive definite Hessian such that $H_l = R_l^T R_l$. This reformulation enables the design of an efficient interior-point solver, denoted as $\mathcal{N}$QP. The proof of Theorem 1 follows after some preliminaries of the solver have been outlined.

In the following, the slack variables $h_i^\pm := [h_i^+ \; h_i^-]^T$, w_i (for $i \in \mathbb{I}_l$ only, highlighted with gray crosshatch pattern), and w_j are introduced for the different inequality constraints in SHQP. These are penalized using a log-barrier function with centering parameter σ and duality measure μ [24]. This leads to the following problem formulation:

$$\min_{z,(v_i),(t_i),(h_i^\pm),(w_{i/j})} \quad \sum_{i\in\mathbb{C}_l}\omega_i t_i \qquad l = 1,\dots,p \tag{5}$$

$$+0.5(x_{l-1}^* + N_{l-1}z_l)^T H_l (x_{l-1}^* + N_{l-1}z_l)$$

$$-\sigma\mu\left(\sum_{i\in\mathbb{C}_l}\log(h_i^\pm) + \sum_{i\in\mathbb{I}_l}\log(w_i) + \sum_{j\in\mathcal{I}_{\cup l-1}}\log(w_j)\right)$$

$$\begin{aligned}\text{s.t.}\quad & t_i - v_i = h_i^+, \qquad t_i + v_i = h_i^- \\ & \tilde{a}_i^T z_l - \breve{b}_i = v_i + w_i \\ & \tilde{a}_j^T z_l - \breve{b}_j = w_j \\ & h_i^\pm \geq 0, \quad w_i \geq 0, \quad w_j \geq 0 \end{aligned} \tag{6}$$

Variables with superscripts $\pm$ indicate upper and lower bounds in the constraint set $\mathcal{T}_l$.

The corresponding Lagrangian of (5) is given by:

$$\begin{aligned}\mathcal{L}_l := & \sum_{i\in\mathbb{C}_l}\omega_i t_i + 0.5(x_{l-1}^* + N_{l-1}z_l)^T H_l (x_{l-1}^* + N_{l-1}z_l) \\ & -\sigma\mu\left(\sum_{i\in\mathbb{C}_l}\log(h_i^\pm) + \sum_{i\in\mathbb{I}_l}\log(w_i) + \sum_{j\in\mathcal{I}_{\cup l-1}}\log(w_j)\right) \\ & -\sum_{i\in\mathbb{C}_l}\gamma_i^+(t_i - v_i - h_i^+) + \sum_{i\in\mathbb{C}_l}\gamma_i^-(t_i + v_i - h_i^-) \\ & -\sum_{i\in\mathbb{C}_l}\lambda_i\left(\tilde{a}_i^T z_l - \breve{b}_i - v_i - w_i\right) \\ & -\sum_{j\in\mathcal{I}_{\cup l-1}}\lambda_j(\tilde{a}_j^T z_l - \breve{b}_j - w_j) \end{aligned} \tag{7}$$

γ and λ are the Lagrange multipliers associated with the constraints $\mathcal{T}_l$, $\mathbb{C}_l$, and $\mathcal{I}_{\cup l-1}$. We summarize the problem variables as:

$$q_l = [z_l^T \, (v_i)^T \, (t_i)^T \, (\gamma_i^\pm)^T \, (\lambda_i)^T \, (\lambda_j)^T \, (h_i^\pm)^T \, (w_i)^T \, (w_j)^T]^T \tag{8}$$

The first-order optimality, or Karush-Kuhn-Tucker (KKT), conditions are expressed as:

$$K_{q_l} := \nabla_{q_l}\mathcal{L}_l(q_l) = 0 \tag{KKT}$$

The components of the Lagrangian gradient $K_l(q_l)$ are

$$K_{z_l} = \tilde{H}_l z_l + N_{l-1}^T H_l x_l - \sum_{i\in\mathbb{C}_l}\tilde{a}_i\lambda_i - \sum_{j\in\mathcal{I}_{\cup l-1}}\tilde{a}_j\lambda_j \tag{9}$$

$$K_{\lambda_i} = -\tilde{a}_i^T z_l + b_i + v_i + w_i \tag{10}$$

$$K_{\lambda_j} = -\tilde{a}_j^T z_l + b_j + w_j \tag{11}$$

$$K_{v_i} = \gamma_i^+ - \gamma_i^- + \lambda_i \tag{12}$$

$$K_{t_i} = \omega_i - \gamma_i^+ - \gamma_i^- \tag{13}$$

$$K_{\gamma_i^+} = t_i - v_i - h_i^+, \qquad K_{\gamma_i^-} = t_i + v_i - h_i^- \tag{14}$$

$$K_{w_{i/j}} = \lambda_{i/j} w_{i/j} - \sigma\mu, \qquad K_{h_i^\pm} = \gamma_i^\pm h_i^\pm - \sigma\mu \tag{15}$$

With this foundation, the proof of Theorem 1 is given:

Proof: We prove Theorem 1 by contradiction, considering the KKT conditions of the SHQP. If both the lower and upper bounds of the auxiliary constraint are inactive, we obtain $\gamma_i^+ = \gamma_i^- = 0$. This leads to a conflict between the conditions $K_{t_i} = \omega_i = 0$ and $\omega_i > 0$. Now, consider the case of double-sided activity, $\gamma_i^+, \gamma_i^- > 0$. Due to complementary slackness, we must have $h_i^+ = h_i^- = 0$. This contradicts the condition $K_{\gamma_i^\pm} = 0$, which yields $t_i = v_i$ and $t_{\mathbb{C}_l} = -v_{\mathbb{C}_l}$. This can only hold in the feasible case $v_i = t_i = 0$. Therefore, necessarily, only one bound of each constraint pair $-t_i < v_i < t_i$ is active whenever $v_i \neq 0$. ■

Applying Newton's method to the nonlinear KKT system yields the linearized update:

$$K_{q_l}(q_l + \Delta q_l) \approx K_{q_l}(q_l) + \nabla_{q_l} K_{q_l}(q_l)\Delta q_l = 0 \tag{16}$$

Following variable substitutions can be obtained. We define $\Psi_i \coloneqq w_i/\lambda_i$ and $\Gamma_i^\pm = h_i^\pm/\gamma_i^\pm$.

$$\begin{aligned} \Delta\gamma_i^- = {} & 1/(4\Psi_i + \Gamma_i^+ + \Gamma_i^-)(-2a_i^T \Delta z \\ & - 2(-\Psi_i(K_{t_i} + K_{v_i}) - K_{w_i}/\lambda_i + K_{\lambda_i}) \\ & - (-\Psi_{t_i}^+ K_{t_i} - K_{h_i^+}/\gamma_i^+ - K_{\gamma_i^+}) - K_{h_i^-}/\gamma_i^- - K_{\gamma_i^-}) \end{aligned} \tag{17}$$

$$\Delta\gamma_i^+ = -\Delta\gamma_i^- + K_{t_i} \tag{18}$$

$$\Delta\lambda_i = \Delta\gamma_i^+ - \Delta\gamma_i^- + K_{v_i} \tag{19}$$

$$\Delta\lambda_j = 1/\Psi_j(a_j^T \Delta z - K_{w_j}/\lambda_j + K_{\lambda_j}) \tag{20}$$

$$\Delta w_{i/j} = -(w_{i/j}\Delta\lambda_{i/j} + K_{w_{i/j}})/\lambda_{i/j} \tag{21}$$

$$\Delta h_i^\pm = -(h_i^\pm \Delta\gamma_i^\pm + K_{h_i^\pm})/\gamma_i^\pm \tag{22}$$

$$\Delta v_i = a_i^T \Delta z_l + \Delta w_i + K_{\lambda_i} \tag{23}$$

$$\Delta t_i = \Delta v_i + \Delta h_i^+ - K_{\gamma_i^-} \tag{24}$$

This leads to the main computational step of the interior-point method:

$$\begin{aligned} & (\tilde{H}_l + \textstyle\sum_{i\in\mathbb{C}_l} C_i + \sum_{j\in\mathcal{I}_{\cup l-1}} C_j)\Delta z_l \\ & = - K_z + \textstyle\sum_{i\in\mathbb{C}_l} r_i + \sum_{j\in\mathcal{I}_{\cup l-1}} r_j \end{aligned} \tag{25}$$

The left-hand side matrices C are defined below:

$$C_i \coloneqq 4/(\Gamma_i^+ + \Gamma_i^- + 4\Psi_i)\tilde{a}_i\tilde{a}_i^T \tag{26}$$

$$C_j \coloneqq 1/\Psi_j \tilde{a}_j\tilde{a}_j^T \tag{27}$$

The corresponding right-hand side vectors r are given by:

$$\begin{aligned} r_i \coloneqq {} & \tilde{a}_i(2/(4\Psi_i + \Gamma_i^+ + \Gamma_i^-)(\Gamma_i^+ K_{t_i} \\ & + K_{h_i^+}/\gamma_i^+ + 2(K_{\lambda_i} - \Psi_i(-K_{t_i} - K_{v_i}) - K_{w_i}/\lambda_i) \\ & - K_{h_i^-}/\gamma_i + K_{\gamma_i^+} - K_{\gamma_i^-}) - K_{t_i} - K_{v_i}) \end{aligned} \tag{28}$$

$$r_j \coloneqq \tilde{a}_j(-K_{w_j}/\lambda_j + K_{\lambda_j})/\Psi_j \tag{29}$$

Computing the Newton step Δz_l in (25) requires $O(n_r^3)$ operations, for example when using a Cholesky decomposition. The number of sparse constraints $|\mathbb{C}_l|$ affects the matrix-matrix multiplications in (25) only linearly ($O(n_r^2|\mathbb{C}_l|)$). This scaling behavior aligns with results from unconstrained ℓ_1-regularization methods [18, 19, 5], but contrasts with common robotics formulations [3, 25], which exhibit cubic scaling $O((n+|\mathbb{C}_l|)^3)$ because auxiliary variables are not eliminated.

The Newton step in (25) is then iteratively computed and added to the current primal variable z_l, using a line-search factor to maintain dual feasibility (6) and the condition $\gamma^\pm, \lambda \geq 0$. Once the nonlinear KKT conditions converge ($K_q(q_l) \approx 0$), the new active and inactive constraint sets $\mathcal{A}_l$ and $\mathcal{I}_l$ are assembled. The constraint matrices of the remaining hierarchy levels are then projected into the nullspace of the new active set. This process is repeated across all p levels. Further details on interior-point formulations for hierarchical optimization can be found in [24].

VIII. Evaluation

We evaluate the proposed S-SHQP framework on both hierarchies of numerical test functions and robotic applications, including SHIK-P and SHIK-C. In each SHQP, a trust-region constraint is defined at the zeroth level. Variable regularization or nonlinear robotics constraints like collision avoidance are defined in the ℓ_2-norm and can be solved as described in [21]. The nullspace basis of active constraints is computed following [24].

All algorithms are implemented in `C++` using the `Eigen` library [13]. In the robotic examples, the SH-NLP and SHQP sub-problems are computed using `Pinocchio` [7]. We compare our proposed $\mathcal{N}$QP solver to state-of-the-art QP solvers (H-)MOSEK [2] and (H-)PIQP [27] (H: hierarchical). Theorem 1 is applied to all solvers so that auxiliary variables and constraints are not propagated to lower-priority levels. Simulations are executed on an Intel Core Ultra 9 185H×22 CPU with 64 GB RAM.

First, a pick-and-place scenario with a manipulator as a simplified version of SHIK-P is used for comparison with state-of-the-art nonlinear programming (NLP) and MINLP solvers (Sec. VIII-A). S-SHQP is then verified on a hierarchy of typical test functions to evaluate convergence in hierarchical decision-making (Sec. VIII-B). Efficient SHIK-P with many candidate solutions is validated in humanoid planning for autonomous selection among many potential and prioritized end-effector locations (Sec. VIII-C). Real-time SHIK-C for a manipulator tracking task with sparse joint activation is presented in Sec. VIII-D. Efficient SHIK-C with a large-scale candidate sets is evaluated with a humanoid (Sec. VIII-E). Potential robotic applications are described in Sec. VIII-F. Real-world robot experiments on UFactory's `xarm6` are given for Sec. VIII-A and Sec. VIII-D (see video).

A. SHIK-P of manipulator for pick-and-place

To the best of our knowledge, there is currently no other solver except S-SHQP which can solve the approximation (2) of SH-NLP. In order to benchmark our method nonetheless, we compare it with the nonlinear solvers IPOPT [31] and NLOPT [17] (using Method of Moving Asymptotes [30]) by

$\underline{l}$		ℓ	$f_i(\chi) \leqq \nu_i, \quad i \in \mathbb{C}_l$
$\underline{1}$	Joint ang. lim.	2	$\underline{\chi} \leq \chi \leq \overline{\chi}$
	Coll. av.	2	$f_{\text{coll}}(\chi) \geq 0$
$\underline{2}$	EF	0 / 1 / 2	$\lVert f_{\text{EF}}(\chi) - f_{\text{EF},d,i} \rVert_2^2 = \nu_i$

TABLE 1: SHIK-P for pick-and-place of $|\mathbb{S}| = 10, ..., 100$ objects.

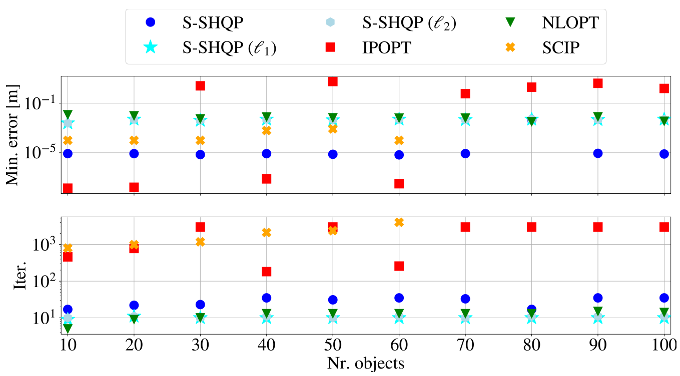

Fig. 3: SHIK-P for pick-and-place: number of nonlinear solver iterations (Iter.) and minimum decision-making error when picking one out of 10 to 100 objects.

solving a simplified version of (2) with two priority levels, which effectively reduces to a NLP:

$$\begin{aligned} \min_{\chi,\nu_2,\tau_2} \quad & \textstyle\sum_{i\in\mathbb{C}_2} \log(\tau_i + \xi) && (30) \\ \text{s.t.} \quad & -\tau_i \leq \nu_i \leq \tau_i, \quad f_i(\chi) \geqq \nu_i, \quad i \in \mathbb{C}_2 \\ & f_j(\chi) \geq 0, \quad j \in \mathcal{I}_1 \end{aligned}$$

Unlike (2), the inequalities $f_{\mathcal{I}_1} \geq 0$ need to be feasible.

UFactory's `xarm6` (end-effector $f_{\text{EF}}(\chi)$, $\chi \in \mathbb{R}^6$) needs to decide on which one of 10 to 100 objects to pick-up, while simultaneously computing the corresponding inverse kinematic posture (see Tab. 1 and Fig. 1 (a)). The inequality constraints f_j represent joint angle limits and collision avoidance.

S-SHQP is shown in Fig. 3 to reliably converge to a minimum decision-making error of less than $1 \cdot 10^{-5}$ m. It can be observed that SH-NLP in combination with both the ℓ_1-norm and ℓ_2-norm leads to identically inaccurate results. This can be attributed to the quadratic formulation in (1), which undermines the sparsity-promoting properties of the ℓ_1-norm. This highlights the versatility and robustness of the ℓ_0-norm, whose effectiveness is agnostic towards the specific problem formulation. Similarly, IPOPT manages to converge accurately for only 4 instances (10, 20, 40, 60 objects) at an approximate error of $3 \cdot 10^{-8}$ m, while exceeding the number of maximum iterations of 3000 in the other cases. On the contrary, S-SHQP requires between 37 and 67 iterations for all samples. NLOPT finishes within 5 to 15 iterations, but manages to converge to a decision-making error of only 3 mm to 1.2 cm (solver tolerances: $1 \cdot 10^{-5}$). All solvers use BFGS Lagrangian Hessian approximations.

Furthermore, the following representative MINLP is solved by SCIP [4]:

$$\begin{aligned} \min_{\chi,\theta_i} \quad & \sum_{i=1}^{m} \theta_i && (31) \\ \text{s.t.} \quad & \underline{\chi} \leq \chi \leq \overline{\chi} \\ & \lVert f_{\text{EF}}(\chi) - f_{\text{EF},d,i} \rVert_2^2 \leq 4\theta_i \\ & \theta_i \in \{0, 1\} \end{aligned}$$

This example uses a planar robot with $n = 2$ and link length 1 m. The Big M scalar 4 is derived from the reachable workspace of the robot. Despite this simpler robot, the solver reaches time-out for 70 objects with $2.9 \cdot 10^6$ branch-and-bound nodes explored, and 4854 IPOPT iterations for NLP relaxations (displayed in Fig. 3).

B. Hierarchical decision-making with test functions

$\underline{l}$	ℓ	$f_i(\chi) \leqq \nu_i, \quad i \in \mathbb{C}_l$	$\lVert(\nu_i^*)\rVert_2$	Iter.
$\underline{1}$	0	$\chi_1^2 + \chi_2^2 - {}^{1.9}/_{2} \leq \nu_{i_1/i_2}$	$^{0}/_{0}$	1
$\underline{2}$	0	$(1-\chi_1)^2 + 100(\chi_2 - \chi_1^2)^2 + {}^{0}/_{5} = \nu_{i_1/i_2}$	$^{2.9\cdot 10^{-4}}/_{5.0}$	72
$\underline{3}$	0	$\chi_1^2 + \chi_2^2 - {}^{0.9}/_{1} = \nu_{i_1/i_2}$	$^{1}/_{0.9}$	1
$\underline{4}$	0	$\chi_2^2 + \chi_3^2 - {}^{1}/_{1.1} = \nu_{i_1/i_2}$	$^{0}/_{0.1}$	7
$\underline{5}$	0	$\chi_4^2 + {}^{1}/_{1.1} \leq \nu_{i_1/i_2}$	$^{1}/_{1.1}$	7
$\underline{6}$	0	$\chi_5^2 + {}^{+1}/_{-1} \leq \nu_{i_1/i_2}$	$^{1}/_{0}$	5
$\underline{7}$	0	$\chi_6^2 + \chi_7^2 + \chi_8^2 - {}^{4}/_{5} = \nu_{i_1/i_2}$	$^{1}/_{1.3\cdot 10^{-8}}$	4
$\underline{8}$	0	$(1-\chi_6)^2 + 100(\chi_7 - \chi_6^2)^2 + {}^{0}/_{4} = \nu_{i_1/i_2}$	$^{2.3\cdot 10^{-7}}/_{4}$	2
$\underline{9}$	0	$\sin(\chi_9+\chi_{10})+(\chi_9-\chi_{10})^2-1.5x\chi_9+2.5\chi_{10}+1=\nu_{i_1}$	21.9	29
	0	$\chi_9^2 + \chi_{10}^2 - 2 = \nu_{i_2}$	$2.8 \cdot 10^{-17}$	
$\underline{10}$	2	$\chi = (\nu_i)$	3.01	1
Σ				129

TABLE 2: SH-NLP with $p = 10$ and $n = 10$, composed of test function selection constraints (Rosenbrock, etc.) as equalities and inequalities, solved by S-SHQP with $\mathcal{N}$QP. The norm $\ell = 0, 2$, optimal slacks ν^*, and iteration counts (Iter.) are shown. On each priority level l, there are two constraints i_1 and i_2 summarized as $^{i_1}/_{i_2}$. Zero constraints are highlighted in blue.

To validate solver behavior for hierarchical decision-making, we define a hierarchy of common optimization test functions (Rosenbrock, etc.) with $p = 10$ levels and $n = 10$ variables (Tab. 2). The levels include identical (levels 1-8) or mixed (level 9) selection constraints. The results for S-SHQP with $\mathcal{N}$QP confirm the expected behavior:

- Zero constraints are correctly identified for both equality (levels 2, 4, 7, 8, 9) and inequality (level 6) constraints.
- Infeasible selection constraints are removed to prevent Hessian activation (e.g., level 7), improving hierarchical decision-making on lower levels (see Sec. VI-B).

C. SHIK-P of humanoid robot with many candidate locations

We compute the posture $\chi \in \mathbb{R}^{31}$ of the humanoid robot `G1` while autonomously selecting among Cartesian locations (see Fig. 1 (b)). Each end-effector (LF, RF, LH, RH) is assigned 200 potential locations (Tab. 3), for example identified by exteroceptive sensors. Unlike ℓ_2-norm minimization, ℓ_0 optimization enables choosing a single feasible location rather than averaging over all. The feet placement is thereby prioritized over the hand placement for hierarchical decision-making.

l		ℓ	$f_i(\chi) \leqq \nu_i, \quad i \in \mathbb{C}_l$	$\|(\nu_i^*)\|_2$	Iter.
1	Joint ang. lim.	2	$\underline{\chi} \leq \chi \leq \overline{\chi}$	0	1
2	Coll. av.	2	$f_{\text{coll}}(\chi) \geq (\nu_{\text{coll}})$	$2.4 \cdot 10^{-4}$	1
3	CoM$_z$	2	$f_{\text{CoM}_z}(\chi) - f_{\text{LF/RF},z}(\chi) - 0.2 \geq (\nu_{\text{CoM}})$	0	1
4	LF	0	$\|f_{\text{LF}}(\chi) - f_{\text{LF},d,s_1}(t)\|_2^2 = \nu_{s_1}, \quad \|\mathbb{S}_1\| = 200$	$3.8 \cdot 10^{-6}$	37
	RF	0	$\|f_{\text{RF}}(\chi) - f_{\text{RF},d,s_2}(t)\|_2^2 = \nu_{s_2}, \quad \|\mathbb{S}_2\| = 200$	$4.1 \cdot 10^{-6}$	
5	LH	0	$\|f_{\text{LH}}(\chi) - f_{\text{LH},d,s_1}(t)\|_2^2 = \nu_{s_1}, \quad \|\mathbb{S}_1\| = 200$	$5.1 \cdot 10^{-4}$	31
	RH	0	$\|f_{\text{RH}}(\chi) - f_{\text{RH},d,s_2}(t)\|_2^2 = \nu_{s_2}, \quad \|\mathbb{S}_2\| = 200$	$6.5 \cdot 10^{-5}$	
Σ					71

TABLE 3: SHIK-P of humanoid robot, $p = 5$ and $n = 31$, solved in 71 iterations (0.17 s, $\mathcal{N}$QP).

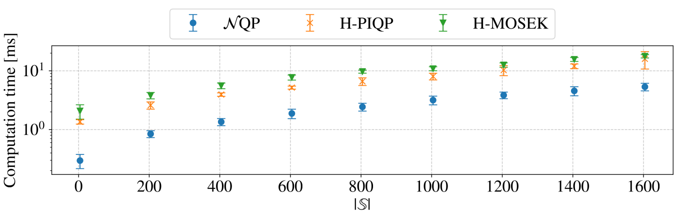

Fig. 4: Computation times of solving SHQP's for humanoid robot inverse kinematics hierarchy (Tab. 3, levels 4-5 only). The number of sparse equality constraints $|\mathbb{S}|$ is increased from 1 to 1604.

The hierarchy in Tab. 3 includes joint limits ($\underline{\chi}$, $\overline{\chi}$), collision avoidance (self and a horizontal bar), a center-of-mass (CoM) to feet minimum distance constraint in vertical direction to enforce balanced upright postures, and finally the selection constraints for each limb. The problem is solved in 0.23 s (89 S-SHQP iterations) by $\mathcal{N}$QP. For comparison, H-PIQP converges in 90 iterations / 0.33 s, and H-MOSEK in 46 iterations / 0.59 s. Each end-effector successfully selects a unique feasible location, with results over 10 different initial configurations visualized in Fig. 1 (b).

As the proposed planner is local, some of the selected locations end up being infeasible due to conflict with same or higher priority constraints (see Fig. 5).

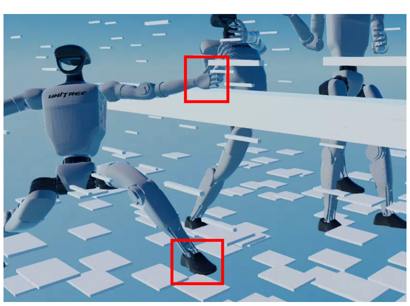

Fig. 5: S-SHQP is a local solver which can misguide the robot to reach for out-of-reach targets. Global understanding of the scene would encourage the robot to aim for feasible targets closer to the robot torso. Note that the rightmost robot posture has not converged yet.

To evaluate the impact of the selection constraint size $|\mathbb{S}| = 4, \dots, 1604$ (corresponding to $1, \dots, 400$ possible locations per end-effector) on computation time, the hierarchy in Tab. 3 is solved considering only the equality constraints on levels 4 and 5. Figure 4 confirms a linear relationship between computation time and the number of sparse ℓ_0-norm constraints. The proposed $\mathcal{N}$QP solver consistently outperforms H-PIQP and H-MOSEK, both of which do not exploit the structured sparsity of SHQP problems.

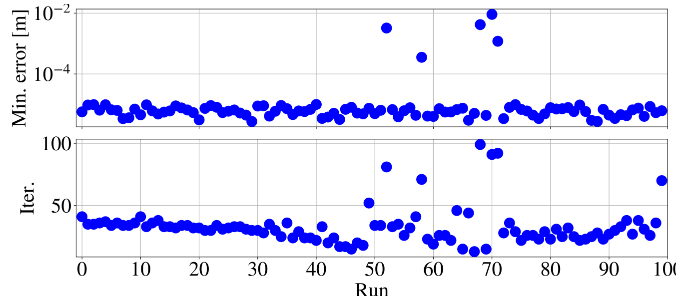

Fig. 6: S-SHQP is run a 100 times on level 4 only of hierarchy Tab. 3.

Lastly, we conduct 100 simulations on G1 foot location planning with different initial configurations in order to test the robustness of our solver S-SHQP. Infeasibility due to conflict with higher priority levels is avoided by solely considering level 4 of the hierarchy in Tab. 3. The robot succeeds in 95 simulations with a DM error smaller than $1 \cdot 10^{-5}$ m on both feet (see Fig. 6), while considering cost encoding via the mask in (4) to avoid feet allocation to the same location. Similarly to Fig. 5, the failure cases can be explained by the local nature of the solver.

D. SHIK-C of manipulator for object tracking

l		ℓ	$f_i(\chi) \leqq \nu_i, \quad i \in \mathbb{C}_l$
1	Joint ang. lim.	2	$\underline{\chi} \leq \chi \leq \overline{\chi}$
2	Coll. av.	2	$f_{\text{coll}}(\chi) \geq (\nu_{\text{coll}})$
3	CoM$_z$	2	$f_{\text{CoM}_z}(\chi) - f_{\text{Shoulder},z}(\chi) - 0.1 \geq (\nu_{\text{CoM}})$
4	EF	0	$\|f_{\text{EF}}(\chi) - f_{\text{EF},d,s_1}(t)\|_2^2 = \nu_{s_1}, \quad \|\mathbb{S}_1\| = 2$
	Reg	0	$1 \cdot 10^{-3}\chi = (\nu_{s_2}), \quad \|\mathbb{S}_2\| = n$
5	Reg.	2	$\chi = \nu_{\text{Reg}}$

TABLE 4: SHIK-C of manipulator: Task hierarchy for tracking of two targets with $p = 5$ and $n = 6$.

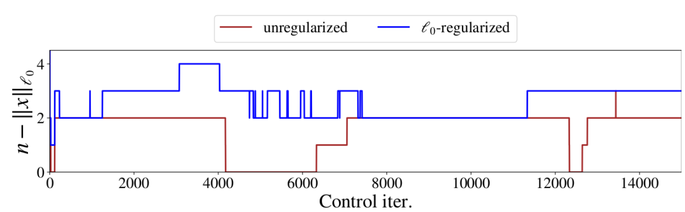

Fig. 7: SHIK-C of manipulator: Number of joints with zero angle for parsimonious kinematic control [12].

This test considers selective tracking of one of two moving targets with a real-world UFactory `xarm6` manipulator (see video). The two Cartesian locations $f_{\text{EF},d,i}(t) \in \mathbb{R}^3$ $(i = 1, 2)$ move along opposing circular trajectories, partially occluded by a wall. The CoM of the robot must remain at least 0.1 m above the shoulder link. Optionally, an ℓ_0-norm regularization term with small weight $1 \cdot 10^{-3}$ is added to achieve sparse kinematic control (colored in blue in Tab. 4). Such concurrent decision-making, parsimonious control, and presence of constraints is not possible in approaches like [12].

The control loop time is around 0.8 ms using $\mathcal{N}$QP. Figure 7 shows the number of joints remaining at zero throughout execution. The ℓ_0-regularized formulation leads to sparser actuation, e.g., joints χ_4, χ_5, and χ_6 remain inactive when possible. Tracking accuracy decreases slightly, highlighting the challenge of adequately balancing between sparsity triggered by the logarithmic function in (2), and precision.

E. SHIK-C of humanoid robot with many candidate locations

$\underline{l}$		ℓ	$f_i(\chi) \leqq \nu_i, \quad i \in \mathbb{C}_l$
$\underline{1}$	Joint ang. lim.	2	$\underline{\chi} \leq \chi \leq \overline{\chi}$
$\underline{2}$	Coll. av.	2	$f_{\text{coll}}(\chi) \geq (\nu_{\text{coll}})$
$\underline{3}$	LF / RF / LH	2	$f_i(\chi) - f_{i,d}(t) = \nu_i, \quad i \in$ LF / RF / LH
$\underline{4}$	RH	0 / 1 / 2	$\Vert f_{\text{RH}}(\chi) - f_{\text{RH},d,s}(t) \Vert_2^2 = \nu_s$, $\vert\mathbb{S}\vert = 100$
$\underline{5}$	Reg.	2	$\chi - \chi_d = \nu_{\text{Reg}}$

TABLE 5: SHIK-C of humanoid robot with $p = 5$ and $n = 31$.

To assess solver performance in large-scale sparse control, we simulate a `G1` tasked with touching 100 objects moving downward, representing a dynamic catching scenario. The objects descend at constant velocity and are randomly distributed within reach of the robot.

The control hierarchy is shown in Tab. 5. End-effector location selection for the right-hand is formulated as a selection constraint with $|\mathbb{S}| = 100$ on level 4. When the distance to an object drops below 1 cm, the object is removed from the group $\mathbb{S}$ for a continuous autonomous selection of new targets.

If the fourth level is formulated in the ℓ_0-norm, the robot successfully interacts with 92 out of 100 objects before they pass beyond reach. If the fourth level is formulated as a linear or least-squares program (ℓ_1 / ℓ_2-norm), the robot fails to contact any (see Sec. III). The $\mathcal{N}$QP solver handles each SHQP in 1.6 ± 0.3 ms, outperforming H-PIQP (2.2 ± 0.7 ms) and H-MOSEK (8.3 ± 2.7 ms). These results confirm the efficiency and scalability of the proposed sparse hierarchical formulation for both planning and real-time control tasks.

F. Potential applications

Sorting and distribution logistics play a critical role in maintaining competitiveness in e-commerce and other domains such as automated waste sorting. The following demonstrates how such tasks can be formulated as sparse optimization problems with immediate and continuous decision-making (SHIK-C), rather than relying on heuristic approaches which consider IK separately.

In the first simulation, multiple manipulators jointly address a shared selection constraint in which as many passing objects as possible must be removed from a conveyor (see Fig. 1 (c)). Once an object has been removed by one of the robots, it is excluded from the selection constraint for all others. Such scenarios are representative of industrial sorting applications (e.g., e-commerce fulfillment or coal mine operations [11]), where different types of grasping end-effectors (such as suction cups, bi-manual grippers, or vacuum-based systems) may be employed depending on the nature of the objects.

$\underline{l}$		ℓ	$f_i(\chi) \leqq \nu_i, \quad i \in \mathbb{C}_l$
$\underline{1}$	Joint ang. lim.	2	$\underline{\chi} \leq \chi \leq \overline{\chi}$
$\underline{2}$	Coll. av.	2	$f_{\text{coll}}(\chi) \geq (\nu_{\text{coll}})$
$\underline{4}$	LF / RF	2	$\Vert f_i(\chi) - f_{i,d} \Vert_2^2 = \nu_i, \quad i \in$ LF / RF
$\underline{3}$	CoM	2	$f_{\text{CoM}}(\chi) \geq (\nu_{\text{CoM}})$
$\underline{4}$	LH / RH	0	$\left\Vert \begin{bmatrix} f_{\text{LH}}(\chi) - f_{\text{LH},d,s}(t) \Vert_2^2 \\ f_{\text{RH}}(\chi) - f_{\text{LH}}(\chi) - f_{\text{RH},d} \end{bmatrix} \right\Vert_2^2 = \nu_s, \quad \vert\mathbb{S}\vert = 4$
$\underline{5}$	Reg.	2	$\chi = \nu_{\text{Reg}}$

TABLE 6: SHIK-C of humanoid robot: Task hierarchy for grasping randomly rotated box.

The second simulation illustrates how the two distinct left and right arm motions of a humanoid can be formulated within the same ℓ_0-norm constraint (see Tab. 6). Each of the four vertical sides of a box, placed on a table and randomly rotated within the range $[0, 2\pi]$, is assigned a unique identifier. The left hand is tasked with selecting one of the sides, while the right hand must be placed on the opposite side, Fig. 1 (d).

IX. Conclusion

We have presented a sparse hierarchical optimization framework for hierarchical decision-making in model-based robotics. The SH-NLP formulation integrates sparsity-promoting principles within hierarchical nonlinear optimization, enabling efficient planning and control of redundant robots under large sets of constraints and potential discrete candidate locations. By exploiting the structure of the quadratic sub-problem SHQP, the proposed numerical scheme achieves linear computational scaling with respect to the number of sparse constraints. The approach was validated on hierarchies composed of standard nonlinear test functions and demonstrated on robot SHIK-P and SHIK-C tasks involving autonomous location selection, confirming both convergence and computational efficiency.

This work opens a promising direction towards unifying model-based, whole-body optimal control and autonomous discrete contact planning by leveraging sparse hierarchical nonlinear optimization. We envision an incorporation into a reinforcement learning pipeline to enhance robustness against sparse gradients. Notably, the ℓ_0-norm approximation is differentiable, unlike other decision-making methods like mixed-integer programming, and is therefore amenable to informed training regimes like Sobolev training [1].

X. Acknowledgements

This work was partly supported by the RDF-24-02-060 XJTLU Research Development Fund. This work was partly supported by the Schaeffler Hub for Advanced Research at Nanyang Technological University, under the ASTAR IAF-ICP Programme (grant number ICP1900093).

References

[1] Elisa Alboni, Gianluigi Grandesso, Gastone Pietro Rosati Papini, Justin Carpentier, and Andrea Del Prete. Cacto-sl: Using sobolev learning to improve continuous actor-critic with trajectory optimization. In *Conference on Learning for Dynamics & Control*, 2023.

[2] MOSEK ApS. *MOSEK Fusion API for C++ 10.1.12*, 2019. URL https://docs.mosek.com/latest/cxxfusion/index.html.

[3] Bastien Berret, Christian Darlot, Frédéric Jean, Thierry Pozzo, Charalambos Papaxanthis, and Jean Paul Gauthier. The inactivation principle: Mathematical solutions minimizing the absolute work and biological implications for the planning of arm movements. *PLOS Computational Biology*, 4(10):1–25, 2008.

[4] Ksenia Bestuzheva, Antonia Chmiela, Benjamin Müller, Felipe Serrano, Stefan Vigerske, and Fabian Wegscheider. Global optimization of mixed-integer nonlinear programs with scip 8. Technical Report 2301.00587, arXiv, 2023.

[5] Andrea Borsic and Andy Adler. A primal–dual interior-point framework for using the l1 or l2 norm on the data and regularization terms of inverse problems. *Inverse Problems*, 28(9):095011, August 2012.

[6] Emmanuel Candès, Michael Wakin, and Stephen Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2007.

[7] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. The Pinocchio C++ library— a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE/SICE International Symposium on System Integration*, pages 614–619, 2019.

[8] Traiko Dinev, Wolfgang Merkt, Vladimir Ivan, Ioannis Havoutis, and Sethu Vijayakumar. Sparsity-inducing optimal control via differential dynamic programming. In *IEEE International Conference on Robotics and Automation*, pages 8216–8222, 2021.

[9] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.

[10] Roger Fletcher, Sven Leyffer, and Philippe L. Toint. On the global convergence of a filter–sqp algorithm. *SIAM Journal on Optimization*, 13(1):44–59, 2002.

[11] Rui Gao, Mengcong Liu, Jingyi Du, Yifan Bao, Xudong Wu, and Jiahui Liu. Research on a cooperative grasping method for heterogeneous objects in unstructured scenarios of mine conveyor belts based on an improved matd3. *Sensors*, 25(22), 2025.

[12] Viniçius Gonçalves, Philippe Fraisse, André Crosnier, and Bruno Adorno. Parsimonious kinematic control of highly redundant robots. *IEEE Robotics and Automation Letters*, 1(1):65–72, 2016.

[13] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[14] Naoki Hayashi, Takuya Ikeda, and Masaaki Nagahara. Design of sparse control with minimax concave penalty. *IEEE Control Systems Letters*, 8:544–549, 2024.

[15] Enrico Mingo Hoffman, Matteo Parigi Polverini, Arturo Laurenzi, and Nikos G. Tsagarakis. A study on sparse hierarchical inverse kinematics algorithms for humanoid robots. *IEEE Robotics and Automation Letters*, 5(1):235–242, 2020.

[16] Jemin Hwangbo, Joonho Lee, and Marco Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018. URL www.raisim.com.

[17] Steven G. Johnson. The NLopt nonlinear-optimization package. https://github.com/stevengj/nlopt, 2007.

[18] Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. An interior-point method for large-scale ℓ_1-regularized logistic regression. *Journal of Machine Learning Research*, 8(54):1519–1555, 2007.

[19] Ignace Loris. On the performance of algorithms for the minimization of ℓ_1-penalized functionals. *Inverse Problems*, 25(3):035008, January 2009.

[20] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

[21] Kai Pfeiffer and Abderrahmane Kheddar. Sequential hierarchical least-squares programming for prioritized non-linear optimal control. *Optimization Methods and Software*, 39(4):1104–1142, 2024.

[22] Kai Pfeiffer, Adrien Escande, and Abderrahmane Kheddar. 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, 2018.

[23] Kai Pfeiffer, Adrien Escande, Pierre Gergondet, and Abderrahmane Kheddar. The hierarchical newton's method for numerically stable prioritized dynamic control. *IEEE Transactions on Control Systems Technology*, 31(4):1622–1635, 2023.

[24] Kai Pfeiffer, Adrien Escande, and Ludovic Righetti. $\mathcal{N}$IPM-HLSP: an efficient interior-point method for hierarchical least-squares programs. *Optimization and Engineering*, 25(2):759–794, 2024.

[25] Matteo Parigi Polverini, Enrico Mingo Hoffman, Arturo Laurenzi, and Nikos G. Tsagarakis. Sparse optimization of contact forces for balancing control of multi-legged humanoids. *IEEE Robotics and Automation Letters*, 4(2):1117–1124, 2019.

[26] Ajay Suresha Sathya, Goele Pipeleers, Wilm Decré, and Jan Swevers. A weighted method for fast resolution of strictly hierarchical robot task specifications using exact penalty functions. *IEEE Robotics and Automation Letters*, 6(2):3057–3064, 2021.

[27] Roland Schwan, Yuning Jiang, Daniel Kuhn, and

Colin N. Jones. PIQP: A proximal interior-point quadratic programming solver. In *IEEE Conference on Decision and Control*, pages 1088–1093, 2023.

[28] Yuki Shirai, Xuan Lin, Alexander Schperberg, Yusuke Tanaka, Hayato Kato, Varit Vichathorn, and Dennis Hong. Simultaneous contact-rich grasping and locomotion via distributed optimization enabling free-climbing for multi-limbed robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13563–13570, 2022.

[29] Daeun Song, Pierre Fernbach, Thomas Flayols, Andrea Del Prete, Nicolas Mansard, Steve Tonneau, and Young J. Kim. Solving footstep planning as a feasibility problem using l1-norm minimization. *IEEE Robotics and Automation Letters*, 6(3):5961–5968, 2021.

[30] Krister Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12: 555–573, 2002.

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

[32] Hao Wang, Wanying Zhang, Yuxin He, and Wenming Cao. ℓ_0-norm based short-term sparse portfolio optimization algorithm based on alternating direction method of multipliers. *Signal Processing*, 208:108957, 2023.

[33] Huayi Wang, Zirui Wang, Junli Ren, Qingwei Ben, Tao Huang, Weinan Zhang, and Jiangmiao Pang. BeamDojo: Learning agile humanoid locomotion on sparse footholds. In *Robotics: Science and Systems*, 2025.

[34] Chen Zhao, Naihua Xiu, Houduo Qi, and Ziyan Luo. A lagrange–newton algorithm for sparse nonlinear programming. *Mathematical Programming*, 195(1–2): 903–928, sep 2022.