Multi-Contact Motion Retargeting using Whole-body Optimization of Full Kinematics and Sequential Force Equilibrium

Quentin Rouxel, Kai Yuan, Ruoshi Wen, Zhibin Li

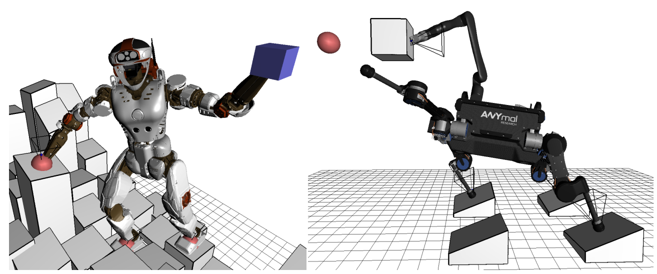
***Abstract*—This paper presents a multi-contact motion adaptation framework that enables teleoperation of high degree-of-freedom (DoF) robots, such as quadrupeds and humanoids, for loco-manipulation tasks in multi-contact settings. Our proposed algorithms optimize whole-body configurations and formulate the retargeting of multi-contact motions as sequential quadratic programming, which is robust and stable near the edges of feasibility constraints. Our framework allows real-time operation of the robot and reduces cognitive load for the operator because infeasible commands are automatically adapted into physically stable and viable motions on the robot. The results in simulations with full dynamics demonstrated the effectiveness of teleoperating different legged robots interactively and generating rich multi-contact movements. We evaluated the computational efficiency of the proposed algorithms, and further validated and analyzed multi-contact loco-manipulation tasks on humanoid and quadruped robots by reaching, active pushing and various traversal on uneven terrains.**

***Index Terms*—Teleoperation; Motion Retargeting; Multi-Contact; Humanoid; Legged robot;**

I. Introduction

Human-in-the-loop approaches for controlling robots are of essential importance in safety-critical, cognitively challenging and high-risk tasks [1], [2], which can be achieved through high-level supervision and/or online commands. Human's involvements in the control loop complement robots' abilities in perception and motor actions. It also provides robots with versatile motor skills for unforeseen situations and dexterous interactions in uncertain environments. Contextual understanding and safe decisions are required to deploy robots in remote tasks, such as distant planetary exploration [3], subsea inspection and nuclear decommissioning. For example, the ESA's METERON project developed a teleoperation system where the operators' skills in dealing with different situations were used to control planetary robots from the orbit [4].

The development of robotic teleoperation is advancing towards improving their versatile capabilities with complex platforms. For example, legged robots with manipulators and arms can be teleoperated to perform loco-manipulation tasks in challenging, unstructured, and natural terrains. However, legged robots, e.g. humanoids and quadrupeds, have a high number of degrees of freedom, making it difficult for operators to command all these joints directly while satisfying the balancing criteria. Besides, robotic systems are also subject to physical constraints, such as joint limits, actuator power limits and non slipping contacts, and the teleoperation system must consider all of these to ensure safety. When human operators make mistakes, the system should be robust and able to deal with any dangerous or infeasible commands.



(a) Valkyrie (b) ANYmal with a robot arm

Fig. 1: Teleoperation of multi-contact interactions – locomotion and manipulation on uneven terrains for humanoid and quadruped robots.

To address these problems, the teleoperation system needs to retarget and adapt desired motions into a specific robot's morphology, while considering its physical limitations. While the balance on flat terrain can be analyzed by simple geometric criteria, such as the projection of the Center of Mass (CoM), the contact wrenches and force distribution need to be considered in complex non-coplanar multi-contact cases. Unlike the planning problem with known future states [5], online teleoperation system is an interactive scheme where only the current command of the operator is known. Therefore, the retargeting method must adapt the operator's input reactively to enforce safety in real time.

In this work, we developed a novel formulation to solve the motion retargeting as an optimization problem efficiently, with which operator's commands can be adapted and then executed on robots in real time – guaranteeing all the feasibility constraints, and the balance of floating base robots as well.

The automatic adaptation and enforcement, including balance and other hard physical constraints, alleviate human mental load and allow the operator to focus on high-level supervision for solving complex loco-manipulation tasks – shared control where humans provide task-level skills, and the algorithms resolve the local control of the high degree-of-freedom (DoF) robots and their physical constraints. Our method is suited for difficult high-DoF teleoperation where safety, feasibility, and effective prevention of erroneous commands are crucial.

A. Related Works

Conventional retargeting schemes for teleoperation use Inverse Kinematics (IK) extensively to compute whole-body joint positions from desired end-effectors and Center of Mass (CoM) references. For humanoids on flat ground [6], [7], [8], [9], [10], quasi-static equilibrium is formulated by constraining the CoM projection within the support polygon.

However, IK-based schemes only consider kinematics constraints, multi-contact loco-manipulation on uneven surfaces require the use of force-related quantities to guarantee feasibility. OpenSoT proposed an IK formulation to constrain both kinematic and dynamic quantities by integrating the joint accelerations at the velocity level [11], but the contact wrenches are neither constrained nor optimized and must be provided as inputs additionally.

The balance of dynamic motions was studied on flat ground using the Zero Moment Point (ZMP) [12] and the Linear Inverted Pendulum Model (LIPM) [13], based on which the dynamic filter [14] was proposed to generate balanced motions by transforming the ZMP references commanded by the operator. Similarly, the Divergent Component of Motion (DCM) was applied to predict the evolution of the system for long-term balance [15]. In [16], human motions were captured online and transferred to a humanoid robot on flat ground. The above methods used simplified models to avoid the computationally expensive nonlinear whole-body model, but are limited to flat (or coplanar) surfaces and do not address multi-contact cases. The complex centroidal model [17] and energy state based model [18] for fall prediction are promising alternatives.

The retargeting in [19] extended humanoid teleoperation to the multi-contact case where the contact switching, as well as the constraints of kinematics, torque and contact were realized by an inverse dynamics Quadratic Programming (QP) controller. However, only coplanar contact surfaces were considered and the balance criterion purely relied on the kinematics of the projected CoM, so the method cannot address unstructured uneven terrains.

There are previous research in planning similar to our proposed scheme. The work in [20] solved kinematics and force-related quantities in a constrained nonlinear Sequential Quadratic Programming (SQP), assuming the static equilibrium. A sequence of keyframe configurations were optimized in [21], contact stance poses were solved in [22], and uneven multi-contact postures were computed by using analytical partial derivatives in [23]. Compared to these works designed for offline planning, our proposed scheme is developed specifically for online real-time applications as an interactive teleoperation process. We developed novel techniques detailed in the following to enable fast real-time computation.

B. Contribution

This work proposes an optimization-based motion retargeting to teleoperate robots and achieve physically feasible, safe and balanced multi-contact tasks – *Sequential Equilibrium and Inverse Kinematics Optimization (SEIKO)* – applicable and suitable for combined locomotion and manipulation on uneven surfaces, where only quasi-static and/or slow-medium speeds are required, and safety and risk mitigation are more critical.

Our contributions are summarized as follows:

1) **A new algorithmic formulation of SEIKO (Section III-A–III-E)** with real-time performance to optimize the whole robot configuration of joint positions, torques, and contact forces under strict feasibility constraints.
2) **Smooth multi-contact switching algorithm (Section III-F)** for transitions in-between adding–removing new physical contacts using SEIKO.
3) **An integrated motion retargeting teleoperation framework (Section II-B, III-G)** for safe and robust interactive loco-manipulation tasks in multi-contact scenarios.

The proposed SEIKO is validated on floating-base robots (humanoid, quadruped) on various multi-contact tasks, and considering both plane and point contacts (see Fig. 1). The framework has flexibility to use different low-level controllers, e.g. inverse dynamics [24] or admittance control [25] to track references of robot posture and contact forces for stabilization.

Inverse dynamic controllers are designed to track dynamic motions and guarantee instantaneous dynamic stability, but can not guarantee the long-term balance alone. They react aggressively at the edge of the feasibility boundary and eventually fail when the input reference is physically infeasible. Usually, high-level planners can take care of the feasibility by pre-computing viable trajectories offline (either quasi-static or dynamic), but this approach is not applicable to online teleoperation, because the future operator's commands are unknown and subject to any changes. Hence, our proposed motion retargetting serves as a safety layer for interactive online teleoperation in the context of multi-contact.

Compared to previous works based on IK, the formulated SEIKO includes both kinematics and contact forces. This allows to undertake a broader set of tasks such as the contact switching on uneven multi-contact surfaces, postural optimization for minimizing joint torques or pushing tasks (Section IV-C2)), while safely ensuring the balance equilibrium.

The remainder of this paper is organized as follows. The teleoperation scheme is detailed in Section II. The core algorithmic details of SEIKO for retargeting and the contact switching are formulated in Section III. The validation is presented, evaluated and analyzed in Section IV with loco-manipulation tasks demonstrated in simulations. The limitations are discussed in Section V. Finally, we concluded and suggested future work in Section VI.

II. Multi-contact Teleoperation Framework

A. Command Paradigm

As depicted in Fig. 2, the operator commands a high-DoF robot by mapping the end-effector motions to the whole-body configuration while guaranteeing feasibility, safety, and balance. The robot establishes supporting contacts with the environment through the end-effectors, i.e. feet and hands. Each contact is categorized either as a planar (e.g., rectangular foot) or a point contact (e.g., hand stump), and has one contact state which is either *enabled* or *disabled* (i.e. free end-effector). The operator can provide three types of commands: the position and orientation for each free end-effector, a discreet contact switching trigger to remove or add a contact, and in an optional mode, a reference for the force applied by a specific contact. The operator continuously commands the desired poses of the free end-effectors while the retargeting method optimizes the

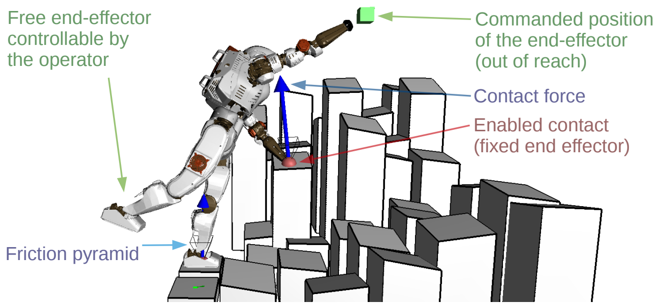


Fig. 2: User interface for our multi-contact teleoperation, where an operator commands the pose of free end-effectors and can trigger contact switches.

interaction forces at the enabled contacts. At any time a contact switching smooth transition can be triggered (Section III-F) to add or remove a selected end-effector. The operator can also optionally define a desired normal contact force for a specific enabled contact to achieve pushing tasks. Note that for point contacts, the surface orientation has to be externally provided. As investigated in [26], the same formulation can allow the operator to command other predefined links of the robot such as the head, pelvis or shoulders.

B. Design of the Control Architecture

Fig. 3 shows our 2-stage architecture of retargeting and control. The human operator continuously provides a high level Cartesian command $\boldsymbol{X}^{\text{target}}$ (pose of the end-effectors). In stage 1, the proposed SEIKO method retargets the commanded poses into best matching whole-body configuration. First an incremental configuration change [$\Delta\boldsymbol{q}$, $\Delta\boldsymbol{\lambda}$] is optimized with respect to all the physical constraints. The change is then integrated to produce a feasible desired configuration [$\boldsymbol{q}^{\text{d}}$, $\boldsymbol{\lambda}^{\text{d}}$, $\boldsymbol{\tau}^{\text{d}}$]. In stage 2, this desired configuration is tracked by a whole-body dynamic controller based on inverse dynamics that solves a Quadratic Programming (QP) [24]. The controller solves and computes joint accelerations, joint torques and contact wrenches while optimizing a set of weighted tasks such as the positions of joints, CoM and contact forces. The joint torque efforts are then sent to the robot system. In supplementary materials, we discuss in Section 9 the limitations of QP controllers, and we show in Section 10 how essential it is to enforce the feasibility of input references.

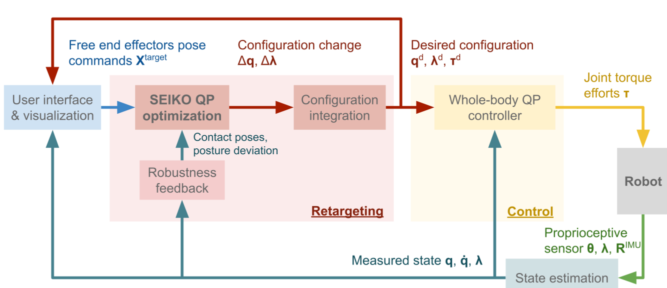


Fig. 3: Two-stage retargeting and control architecture for multi-contact teleoperation. The operator commands the target poses $\boldsymbol{X}^{\text{target}}$ of the end-effectors. At each time step, SEIKO computes the incremental changes $\Delta\boldsymbol{q}, \Delta\boldsymbol{\lambda}$ which are integrated into the desired base and joint position $\boldsymbol{q}^d$, joint torque $\boldsymbol{\tau}^d$ and contact force $\boldsymbol{\lambda}^d$. Given the desired configuration, a whole body dynamic controller computes the joint torques $\boldsymbol{\tau}$ which are sent to the robot.

III. Principles and Formulation of SEIKO

The physics of the robot is governed by the nonlinear equation of motion [27]:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{S}\boldsymbol{\tau} + \boldsymbol{J}(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{\lambda}, \tag{1}$$

where $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{C}$ is the vector of centrifugal and Coriolis forces, $\boldsymbol{G}$ is the gravitational vector, $\boldsymbol{S}$ is the selection matrix for the underactuated floating base, $\boldsymbol{q}$ is the vector of generalized degrees of freedom positions including the pose of the floating base and joint positions (denoted as $\boldsymbol{\theta}$), $\boldsymbol{\tau}$ is the joint torques, $\boldsymbol{J}$ is the stacked Jacobian matrices of all contact points and $\boldsymbol{\lambda}$ is the stacked contact wrenches.

By limiting to slow and continuous commanded motions, we handle the unknown intention of the operator with one step ahead online optimization. With restrictions to quasi-static motions such that $\ddot{\boldsymbol{q}} \approx \dot{\boldsymbol{q}} \approx \boldsymbol{0}$, the terms related with acceleration and centrifugal and Coriolis forces become zero.

Classical dynamic QP controllers solve for the decision variable $\left[\ddot{\boldsymbol{q}}, \boldsymbol{\tau}, \boldsymbol{\lambda}\right]^{\mathsf{T}}$. Hence, if the QP controllers as in [24], [28], [29] implement inequality constraints, they often only operate within a conservative subspace and remain away from their feasibility boundaries.

On the contrary, our formulations particularly address the requirements from multi-contact teleoperation, where stable configuration needs to be planned near and on the edge of the feasibility boundaries. This provides more possibilities to the operator to safely reach and operate at the boundaries, compared to the over restriction in the conventional formulation. We used an active set algorithm for solving the QP which has better numerical stability than interior point methods at the boundaries of inequality constraints.

A. Optimization Formulation

The posture retargeting is formulated as constrained nonlinear optimization, which is solved by a sequence of QP problems. As operator's commands are constantly changing in real time, the problem is continuously updated at each time step. As our teleoperation use case requires interactive and reactive control, fast computation speed is critical.

The proposed SEIKO differs from the classic SQP in three aspects. Firstly, we only compute one SQP iteration (one linearization and QP solution) per control loop. This allows online execution with a fast update frequency, i.e. 1 kHz. Second, the problem is updated at each control loop with the continuously changing commands of the operator. Third, classic SQP schemes use line search [30] to improve convergence speed, i.e. the scalar step length is optimized in the gradient direction to minimize the cost function. However, in our case, our study found that line search increases computational time and is not needed, because the converged errors are sufficiently small (see Section IV-B). Therefore, our formulation keeps the SQP step length constant and equal to 1, as this is the best [31] when the configuration is close to the optimal solution, which is our case because the problem (thus the optimal solution) changes slowly when continuously updated at a high frequency, and the initial configuration is always initialized from the measured robot state.

The SEIKO's QP problem is formulated as a constrained least square optimization that is solved at each control loop:

$$\begin{aligned} & \min_{\Delta \boldsymbol{x}} \ \|\boldsymbol{C}_{\text{cost}}(\boldsymbol{x})\Delta\boldsymbol{x} - \boldsymbol{c}_{\text{cost}}(\boldsymbol{x})\|^2_{\boldsymbol{w}} \quad \text{s.t.} \\ & \boldsymbol{C}_{\text{eq}}(\boldsymbol{x})\Delta\boldsymbol{x} + \boldsymbol{c}_{\text{eq}}(\boldsymbol{x}) = \boldsymbol{0}, \\ & \boldsymbol{C}_{\text{ineq}}(\boldsymbol{x})\Delta\boldsymbol{x} + \boldsymbol{c}_{\text{ineq}}(\boldsymbol{x}) \geqslant \boldsymbol{0}, \\ & \text{where } \boldsymbol{x} = \begin{bmatrix} \boldsymbol{q}^{\text{d}} \\ \boldsymbol{\tau}^{\text{d}} \\ \boldsymbol{\lambda}^{\text{d}} \end{bmatrix}, \Delta\boldsymbol{x} = \begin{bmatrix} \Delta\boldsymbol{q} \\ \Delta\boldsymbol{\tau} \\ \Delta\boldsymbol{\lambda} \end{bmatrix}. \end{aligned} \tag{2}$$

Here, $\boldsymbol{x}$ is the current desired configuration, and the incremental change $\Delta\boldsymbol{x}$ is the decision variable. $\boldsymbol{C}_{\text{cost}}, \boldsymbol{c}_{\text{cost}}, \boldsymbol{C}_{\text{eq}}, \boldsymbol{c}_{\text{eq}}, \boldsymbol{C}_{\text{ineq}}, \boldsymbol{c}_{\text{ineq}}$ are the matrices and vectors defining the cost, equality and inequality constraints respectively. Section III-B to III-C describe the tasks and constraints which are stacked to define $\boldsymbol{C}_{\text{cost}}, \boldsymbol{c}_{\text{cost}}, \boldsymbol{C}_{\text{eq}}, \boldsymbol{c}_{\text{eq}}, \boldsymbol{C}_{\text{ineq}}, \boldsymbol{c}_{\text{ineq}}$ and provided as input to the QP solver.

The motion equation under the quasi-static assumption is linearized and approximated at the first-order, and the analytical derivatives are used for better computation speed and stability. The quadratic cost function and linear constraints are thus formulated with respect to the decision variables.

In contrast to the usual QP formulation, our decision variable here is the incremental change $\left[\Delta\boldsymbol{q}, \Delta\boldsymbol{\tau}, \Delta\boldsymbol{\lambda}\right]^{\mathsf{T}}$, which is equivalent as optimizing the rate change of the configuration $\left[\boldsymbol{q}^{\text{d}}, \boldsymbol{\tau}^{\text{d}}, \boldsymbol{\lambda}^{\text{d}}\right]^{\mathsf{T}}$. The resulting configuration is then updated as:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \Delta\boldsymbol{x}. \tag{3}$$

In the proposed formulation, each new solution is a full configuration set which includes joint positions, joint torques, and contract wrenches $\left[\boldsymbol{q}^{\text{d}}, \boldsymbol{\tau}^{\text{d}}, \boldsymbol{\lambda}^{\text{d}}\right]^{\mathsf{T}}$ at a *stable static equilibrium*. The QP is guaranteed to have a solution because the solution $\left[\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}\right]^{\mathsf{T}}$ of no configuration changes is always valid. This satisfies the particular requirement for safety-critical teleoperation tasks where the system states always need to be stable so that the robot can halt instantly in case of emergency. The quasi-static motion allows the safe emergency stop at any time or when the feasibility boundary is reached.

The following sections describe the weighted cost function and constraints of the optimization (see detailed notations in supplementary materials Section 1). Note that the expressions of spatial algebra are simplified and the formal Lie algebra operations are in the supplementary materials. In the following sections, the desired configurations being optimized are denoted as $\boldsymbol{\theta}, \boldsymbol{q}, \boldsymbol{\tau}, \boldsymbol{\lambda}$, instead of $\boldsymbol{\theta}^d, \boldsymbol{q}^d, \boldsymbol{\tau}^d, \boldsymbol{\lambda}^d$ for clarity.

B. Optimization Formulation

The optimization aims to minimize the weighted tasks:

$$\begin{aligned} \min \ & \left\|\dot{\boldsymbol{\theta}}\right\|^2_{\boldsymbol{w}_{\text{velocity}}} + \|\boldsymbol{\tau}\|^2_{\boldsymbol{w}_{\text{torque}}} + \sum_i \left\|\boldsymbol{\lambda}_i^{\text{target}} - \boldsymbol{\lambda}_i\right\|^2_{\boldsymbol{w}_{\text{contact, i}}} + \\ & \left\|\mathsf{Clamp}\left(\boldsymbol{\theta}^{\text{target}} - \boldsymbol{\theta}\right)\right\|^2_{\boldsymbol{w}_{\text{posture}}} + \\ & \sum_i \left\|\mathsf{ClampNorm}\left(\boldsymbol{X}_i^{\text{target}} \ominus \boldsymbol{X}_i(\boldsymbol{q})\right)\right\|^2_{\boldsymbol{w}_{\text{position,i}}, \boldsymbol{w}_{\text{orientation, i}}} \end{aligned} \tag{4}$$

Table I: Typical parameters used during the humanoid and quadruped experiments. ($\mathbf{1}_n$ is the vector of ones of size the number of joint n)

Parameter	Value
$\boldsymbol{w}_{\text{velocity}} = w_{\text{velocity}}\mathbf{1}_n$	10^4
$\boldsymbol{w}_{\text{posture}} = w_{\text{posture}}\mathbf{1}_n$	1
$\boldsymbol{w}_{\text{position}} = w_{\text{position}}\mathbf{1}_3$	10^3
$\boldsymbol{w}_{\text{orientation}} = w_{\text{orientation}}\mathbf{1}_3$	1 – 10^2
$\boldsymbol{w}_{\text{torque}} = w_{\text{torque}}\mathbf{1}_n$	10^{-5}
$\boldsymbol{w}_{\text{contact}} = w_{\text{contact}}\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 0.01\end{bmatrix}^{\mathsf{T}}$	10^{-5} – 1
$w_{\text{contact}}^{\text{enabled}}$	10^{-5}
$w_{\text{contact}}^{\text{disabled}}$	1
clamp bound for joint angular position	0.1 rad
clamp bound for Cartesian position	0.01 m
clamp bound for Cartesian orientation	0.1 rad
α (contact switching transition factor)	1.005

The joint velocities $\left\|\dot{\boldsymbol{\theta}}\right\|^2_{\boldsymbol{w}_{\text{velocity}}}$ are minimized to enforce the quasi-static motions and to improve the optimization stability around kinematic singularities and feasibility boundaries.

The joint positions $\|\mathsf{Clamp}\left(\boldsymbol{\theta}^{\text{target}} - \boldsymbol{\theta}\right)\|^2_{\boldsymbol{w}_{\text{posture}}}$ are attracted toward a default posture to regularize the nullspace of the end-effector's pose. This term typically helps the end-effector to recover its nominal pose after undergoing a highly singular motion. $\boldsymbol{\theta}^{\text{target}} \in \mathbb{R}^n$ is a nominal joint position vector.

The pose of the ith free end-effector $\left\|\mathsf{ClampNorm}\left(\boldsymbol{X}_i^{\text{target}} \ominus \boldsymbol{X}_i(\boldsymbol{q})\right)\right\|^2_{\boldsymbol{w}_{\text{position,i}}, \boldsymbol{w}_{\text{orientation, i}}}$ is driven towards a target pose in the Cartesian space, where $\boldsymbol{X}_i(\boldsymbol{q}) \in SE(3)$ is the current Cartesian pose measured by forward kinematics and $\boldsymbol{X}_i^{\text{target}} \in SE(3)$ is the target Cartesian pose in world frame. The distance vectors in the joint and Cartesian space are clamped to prevent unbounded numerical values in the QP solver, which improves the stability of solving the optimization when the robot is operating or stuck at the feasibility boundaries. The clamping function $\mathsf{Clamp}()$ thresholds the absolute value of all individual input vector components, while the function $\mathsf{ClampNorm}()$ bounds the norm of the input vector in $\mathbb{R}^3$.

The orientations of enabled contact points are not constrained. We regulate the orientations towards the surface normals to avoid unexpected collisions with the environment. The joint torques $\|\boldsymbol{\tau}\|^2_{\boldsymbol{w}_{\text{torque}}}$ are minimized so that he optimized posture is regularized toward an energy efficient configuration.

The wrenches and forces of every enabled plane and point contacts $\left\|\boldsymbol{\lambda}_i^{\text{target}} - \boldsymbol{\lambda}_i\right\|^2_{\boldsymbol{w}_{\text{contact, i}}}$ are optimized respectively as close to a target wrench/force $\boldsymbol{\lambda}_i^{\text{target}} \in \mathbb{R}^6/\ \mathbb{R}^3$ as possible. The target $\boldsymbol{\lambda}_i^{\text{target}}$ is $\boldsymbol{0}$ for an idle, non-contact end-effector, and it can be used to generate a desired contact force or a center of pressure (CoP). The weights for each contact associated to the task $\boldsymbol{w}_{\text{contact, i}}$ are used to regulate the force distribution among the contacts. The parameters of the cost function used in the experiments of the humanoid robot are listed in Table I. From extensive tests, these parameters were robust to various teleoperation tasks and requires no additional fine tuning to be transferred between two robots.

C. Optimization Constraints

Several types of constraints that assure the feasibility of the configuration are: kinematic constraints (joint position and velocity limits), actuator power constraints (joint torque limits) and balance constraints. The balance constraints require two conditions in the quasi-static case: first, all the external forces acting on the robot and the joint torques must follow the equilibrium equation; second, each contact must be stable, i.e. not slipping, not tilting and not pulling from the surface.

All these constraints are formulated as linear equality and inequality constraints with respect to the decision variables, as in [32]. By constraining the system configuration within these bounds, we can guarantee that the robot is statically balanced with physically feasible postures while satisfying actuation requirements.

The equation of motion enforced at the static balancing equilibrium reduces to:

$$\boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{S}\boldsymbol{\tau} + \boldsymbol{J}(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{\lambda}. \tag{5}$$

In the world frame, the pose of the ith contact points and planes is defined with the kinematic constraints:

$$\begin{aligned} \boldsymbol{X}_i^{\text{target}} \ominus \boldsymbol{X}_i(\boldsymbol{q}) &= \boldsymbol{0} \quad \text{for plane contacts,} \\ \boldsymbol{p}_i^{\text{target}} - \boldsymbol{p}_i(\boldsymbol{q}) &= \boldsymbol{0} \quad \text{for point contacts,} \end{aligned} \tag{6}$$

where $\boldsymbol{X}_i^{\text{target}} \in SE(3)$ and $\boldsymbol{p}_i^{\text{target}} \in \mathbb{R}^3$ are the pose and position measured from the robot's state when the ith contact is established.

The classical inequality constraints (detailed in supplementary materials Section 2 and in [32]) enforce the joint position and torque limits of the system and enable feasible contact conditions with the constrained normal force, center of pressure, friction pyramid and torsional torque.

D. Partial Derivatives

The cost, equality and inequality constraints of the optimization in Section III-B, Section III-C are written in the nonlinear form for clarity. However, solving the QP at each control loop for the posture change $\Delta\boldsymbol{x}$, requires the expressions of their first order differentiation. Most of the differentiated terms are simple. We focus on $\mathsf{ClampNorm}\left(\boldsymbol{X}_i^{\text{target}} \ominus \boldsymbol{X}_i(\boldsymbol{q})\right)$ from (4) and (5) and (6) which are non-trivial.

The target cost for free end-effector pose i in (4) is differentiated as

$$\mathsf{ClampNorm}\left(\boldsymbol{X}_i^{\text{target}} \ominus \boldsymbol{X}_i(\boldsymbol{q})\right) + \boldsymbol{J}_i(\boldsymbol{q})\Delta\boldsymbol{q}, \tag{7}$$

and the plane contact i in (6) for kinematics constraint yields:

$$\boldsymbol{X}_i^{\text{target}} \ominus \boldsymbol{X}_i(\boldsymbol{q}) + \boldsymbol{J}_i(\boldsymbol{q})\Delta\boldsymbol{q} = \boldsymbol{0}, \tag{8}$$

where $\boldsymbol{J}_i(\boldsymbol{q}) \in \mathbb{R}^{6\times(6+n)}$ is the Jacobian of the end-effector frame i expressed in world frame. For point contacts, only the linear part is used.

The equilibrium in (5) is nonlinear in the gravitational term. Using the incremental change in the configuration $\Delta\boldsymbol{x}$, we differentiate the equilibrium equation as:

$$\boldsymbol{G}(\boldsymbol{q}+\Delta\boldsymbol{q}) = \boldsymbol{S}(\boldsymbol{\tau}+\Delta\boldsymbol{\tau}) + \boldsymbol{J}(\boldsymbol{q}+\Delta\boldsymbol{q})^{\mathsf{T}}(\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}). \tag{9}$$

If (5) is differentiated only by $\Delta\boldsymbol{q}$, the term $\left(\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}}\Delta\boldsymbol{q}\right)^T\boldsymbol{\lambda}$ will appear which is bilinear in $(\Delta\boldsymbol{q},\boldsymbol{\lambda})$ and cannot be expressed by the formulation of linear equality constraint of the QP. Therefore, (5) is differentiated by $\Delta\boldsymbol{q}, \Delta\boldsymbol{\tau}, \Delta\boldsymbol{\lambda}$. By using the first-order terms, (9) is approximated as:

$$\begin{aligned} \boldsymbol{G}(\boldsymbol{q}) + \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}\Delta\boldsymbol{q} =& \boldsymbol{S}\boldsymbol{\tau} + \boldsymbol{S}\Delta\boldsymbol{\tau} + \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\lambda} \\ &+ \boldsymbol{J}(\boldsymbol{q})^T\Delta\boldsymbol{\lambda} + \left(\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}}\Delta\boldsymbol{q}\right)^T\boldsymbol{\lambda} \end{aligned} \tag{10}$$

where $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}(\boldsymbol{q}) \in \mathbb{R}^{(6+n)\times(6+n)}$ is the partial derivatives of $\boldsymbol{G}(\boldsymbol{q})$ and $\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}}(\boldsymbol{q}) \in \mathbb{R}^{l\times(6+n)\times(6+n)}$ the kinematics Hessian tensor of the stacked Jacobian of contacts $\boldsymbol{J}(\boldsymbol{q})$.

The Hessian tensor product $\boldsymbol{H} \in \mathbb{R}^{(6+n)\times(6+n)}$ can be rewritten such as:

$$\left(\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}}\Delta\boldsymbol{q}\right)^T\boldsymbol{\lambda} = \boldsymbol{H}\Delta\boldsymbol{q}, \tag{11}$$

$$\text{with} \quad H_{ij} = \sum_{k=1}^{l}\left(\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}}\right)_{kij}\lambda_k = \left(\left(\frac{\partial \boldsymbol{J}}{\partial q_j}\right)^{\mathsf{T}}\boldsymbol{\lambda}\right)_i. \tag{12}$$

The differentiated equation of motion can then be linearly expressed with respect to $\Delta\boldsymbol{x}$ as:

$$\begin{aligned} \left[\left(\tfrac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}} - \boldsymbol{H}\right) \quad -\boldsymbol{S} \quad -\boldsymbol{J}(\boldsymbol{q})^{\mathsf{T}}\right]\Delta\boldsymbol{x} + \boldsymbol{G}(\boldsymbol{q}) \\ - \boldsymbol{S}\boldsymbol{\tau} - \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\lambda} = \boldsymbol{0}. \end{aligned} \tag{13}$$

E. Decomposition of the Equation of Motion

Utilizing the selection matrix, the equation of motion can be split into upper rows (floating base) and lower rows (joint space), as in [33]. This decomposition allows to express the joint torques $\boldsymbol{\tau}$ linearly by the contact forces $\boldsymbol{\lambda}$, so the QP is solved much faster by removing joint torques from the decision variables. The same approach is applied to our differentiated equation of motion by slicing the 6 floating base rows (B) from the n joints rows (J):

$$\begin{aligned} &\boldsymbol{G} = \begin{bmatrix}\boldsymbol{G}_{\mathsf{B}} \\ \boldsymbol{G}_{\mathsf{J}}\end{bmatrix}, \boldsymbol{J} = \begin{bmatrix}\boldsymbol{J}_{\mathsf{B}} & \boldsymbol{J}_{\mathsf{J}}\end{bmatrix}, \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}} = \begin{bmatrix}\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}_{\mathsf{B}} \\ \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}_{\mathsf{J}}\end{bmatrix} \boldsymbol{H} = \begin{bmatrix}\boldsymbol{H}_{\mathsf{B}} \\ \boldsymbol{H}_{\mathsf{J}}\end{bmatrix} \\ &\boldsymbol{G}_{\mathsf{B}} \in \mathbb{R}^6, \boldsymbol{G}_{\mathsf{J}} \in \mathbb{R}^n, \boldsymbol{J}_{\mathsf{B}} \in \mathbb{R}^{l\times 6}, \boldsymbol{J}_{\mathsf{J}} \in \mathbb{R}^{l\times n}, \\ &\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}_{\mathsf{B}}, \boldsymbol{H}_{\mathsf{B}} \in \mathbb{R}^{6\times(n+6)}, \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}_{\mathsf{J}}, \boldsymbol{H}_{\mathsf{J}} \in \mathbb{R}^{n\times(n+6)}. \end{aligned} \tag{14}$$

By applying such upper-lower partitions, the differentiated equation of motion (13) can be replaced by the two partitioned equations below:

$$\left[\left(\tfrac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}_{\mathsf{B}} - \boldsymbol{H}_{\mathsf{B}}\right) \quad -\boldsymbol{J}_{\mathsf{B}}(\boldsymbol{q})^{\mathsf{T}}\right]\begin{bmatrix}\Delta\boldsymbol{q} \\ \Delta\boldsymbol{\lambda}\end{bmatrix} + \boldsymbol{G}_{\mathsf{B}}(\boldsymbol{q}) - \boldsymbol{J}_{\mathsf{B}}(\boldsymbol{q})^T\boldsymbol{\lambda} = \boldsymbol{0}, \tag{15}$$

$$\begin{aligned} &\boldsymbol{\tau} + \Delta\boldsymbol{\tau} = \boldsymbol{T}\begin{bmatrix}\Delta\boldsymbol{q} \\ \Delta\boldsymbol{\lambda}\end{bmatrix} + \boldsymbol{t}, \quad \text{where} \\ &\boldsymbol{T} = \left[\left(\tfrac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}_{\mathsf{J}} - \boldsymbol{H}_{\mathsf{J}}\right) \quad -\boldsymbol{J}_{\mathsf{J}}(\boldsymbol{q})^{\mathsf{T}}\right] \in \mathbb{R}^{n\times(6+n+l)}, \\ &\boldsymbol{t} = \boldsymbol{G}_{\mathsf{J}}(\boldsymbol{q}) - \boldsymbol{J}_{\mathsf{J}}(\boldsymbol{q})^T\boldsymbol{\lambda} \in \mathbb{R}^n. \end{aligned} \tag{16}$$

As shown in (15), the QP only needs to optimize $\left[\Delta\boldsymbol{q}, \Delta\boldsymbol{\lambda}\right]^T \in \mathbb{R}^{6+n+l}$ and the resulting joint torques $\boldsymbol{\tau} + \Delta\boldsymbol{\tau} \in \mathbb{R}^n$ can be equivalently and linearly expressed by (16).

Algorithm 1: Disable contact i

$w_{\text{contact, i}} \leftarrow w_{\text{contact}}^{\text{enabled}}$ `// Initial weighting value before switching`
while $w_{\textit{contact, i}} < w_{\textit{contact}}^{\textit{disabled}}$ **do**
  $w_{\text{contact, i}} \leftarrow \alpha w_{\text{contact, i}}$ (with $\alpha > 1$)
  SEIKO online retargeting:
  $\begin{cases} \text{Solve } (\Delta\boldsymbol{q}, \Delta\boldsymbol{\lambda}) \leftarrow \mathsf{SEIKO\_QP}(\boldsymbol{q}, \boldsymbol{\tau}, \boldsymbol{\lambda}, \boldsymbol{w}) \\ \text{Compute } \Delta\boldsymbol{\tau} \text{ from } (\Delta\boldsymbol{q}, \Delta\boldsymbol{\lambda}) \\ \text{Integrate state } (\boldsymbol{q}, \boldsymbol{\tau}, \boldsymbol{\lambda}) \mathrel{+}= (\Delta\boldsymbol{q}, \Delta\boldsymbol{\tau}, \Delta\boldsymbol{\lambda}) \end{cases}$
end
if $\|\boldsymbol{\lambda}_i\| < \epsilon$ **then**
  Disable contact i
  return Success
else
  Slowly decrease $w_{\text{contact, i}}$ to come back to $w_{\text{contact}}^{\text{enabled}}$
  return Failure
end

F. Contact Switching

The ability to add and remove contacts during teleoperation allows a broader range of manipulation and locomotion tasks, but requires a smooth transition and enforced feasibility constraints. Both the force distribution and the kinematic posture have to change, in order to free a contact point, which cannot be achieved by a pure IK formulation.

Removing a contact needs to smoothly bring contact forces to zero, see the procedure in algorithm 1. The transition is implemented by exponentially increasing the penalty weight associated to the contact force regularization $w_{\text{contact, i}}$ from $w_{\text{contact}}^{\text{enabled}}$ to $w_{\text{contact}}^{\text{disabled}}$ (see parameters in Table I). The duration of this transition is defined by the transition factor α and the update frequency. In our tests, this procedure runs online, and the parameter $w_{\text{contact, i}}$ is being changed while SEIKO optimization keeps running continuously.

Our formulation, which combines both kinematics and force quantities, naturally shifts the posture and force distribution toward the remaining contacts when the wrench penalty on a specific contact is increased. This transition motion is induced by optimizing the regularization terms $\|\boldsymbol{\tau}\|^2_{\boldsymbol{w}_{\text{torque}}}, \|\boldsymbol{\lambda}_i^{\text{target}} - \boldsymbol{\lambda}_i\|^2_{\boldsymbol{w}_{\text{contact, i}}}$ in (4) and the equilibrium equality constraint (5).

To add a new contact, this procedure simply needs to run reversely by changing from $w_{\text{contact}} = w_{\text{contact}}^{\text{disabled}}$ to $w_{\text{contact}}^{\text{enabled}}$, and the posture will change and the contact forces will smoothly redistribute. Note that removing a contact point by smoothly bring contact forces to zero is not always feasible. Such cases may occur when the inequality constraints prevent the posture and the force distribution from fully transferring to other supporting contacts, and the algorithm will fail to solve and remains at the initial contact state.

G. Improvements of Robustness

The two-stage architecture in Fig. 3 consists of motion retargeting and control execution. First, the desired configuration is optimized by SEIKO; second, the measured configuration is estimated from the sensors and used by the dynamic controller to track the desired configuration. To provide a useful and relevant reference, the desired configuration must be consistent with the actual measured state of the robot. This consistency deteriorates when the pose of the contacts mismatch between these two configurations, for example in case of external pushes, slipping contact, or tracking errors. Hence, we take advantage of the online computation of SEIKO to formulate two feedback actions to improve the robustness of teleoperation.

The measured pose of each enabled contact is estimated, filtered and used in the kinematic constraint (6) of the desired configuration $\boldsymbol{X}_i^{\text{target}} = \boldsymbol{X}_i^{\text{measured}}$. Note that this does not generate drifting motion of end-effectors, because both SEIKO and the QP tracking controller assume fixed contacts.

In case of external pushes, the real posture of the robot can deviate from the desired one. We clamp the maximum angular distance between the desired and measured joint positions $\boldsymbol{q} = \boldsymbol{q}^{\text{measured}} + \mathsf{Clamp}(\boldsymbol{q} - \boldsymbol{q}^{\text{measured}})$. Within this angular range and thanks to the controller, the desired posture in joint space acts as a spring-damper attractor. When the angular distance becomes larger than the threshold, the desired posture follows the measured one and acts as a saturation. This feature is useful for safe physical interactions.

H. Implementation

Note that instead of building the costly full kinematic Hessian tensor (11) and (12), only the Hessian-vector product is computed from the differentiation of the Recursive Newton-Euler Algorithm (RNEA) by setting $\ddot{\boldsymbol{q}} = \dot{\boldsymbol{q}} = \boldsymbol{0}$:

$$\frac{\partial \mathsf{ID}}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{M}}{\partial \boldsymbol{q}} \ddot{\boldsymbol{q}} + \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} + \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}} - \frac{\partial \boldsymbol{J}^{\mathsf{T}}}{\partial \boldsymbol{q}} \boldsymbol{\lambda}. \tag{17}$$

The proposed algorithms were implemented in C++ using *RBDL* [34] and *Pinocchio* [35] rigid-body libraries. *Pinocchio* provides efficient and analytical computation of the partial derivatives of the equation of motion [36], hence the matrices $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}$ and $\frac{\partial \boldsymbol{J}^{\mathsf{T}}}{\partial \boldsymbol{q}} \boldsymbol{\lambda}$ in (13) can be quickly retrieved. The QP solver uses *EiQuadProg++* based on the algorithm in [37].

Let n, m_{plane} and $m_{\text{point}} \in \mathbb{N}$ be the number of joints, the numbers of currently enabled plane and point contacts respectively. At each time step, we solve a QP problem of $6 + n + 6m_{\text{plane}} + 3m_{\text{point}}$ decision variables. The total number of equality constraints is $m_{\text{eq}} = 6 + 6m_{\text{plane}} + 3m_{\text{point}}$. Each plane contact generates 18 inequality constraints while each point contact generates 6. In total, the number of inequalities is $m_{\text{ineq}} = 2n + 2n + 18m_{\text{plane}} + 6m_{\text{point}}$.

IV. Results

This section presents the validation results of our proposed teleoperation framework on two types of legged robots – a humanoid and a quadruped – performing complex multi-contact motions on uneven terrains (see the real-time performance in the attached video). The retargeting capabilities were teleoperated online using real-time implementation with all feasibility constraints enforced. The accompanying video of this paper summaries our approach and demonstrates all the validations of both robots during several teleoperated tasks.

Table II: Average and maximum computing time for one control/optimization step (32 DOFs including the floating base).

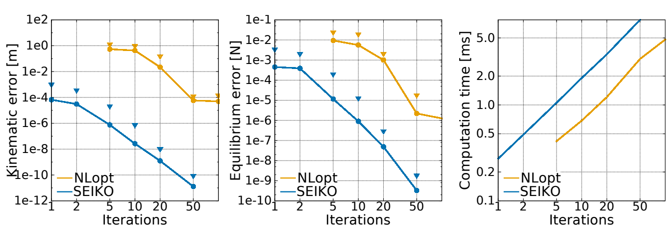
Task	**Avg (max) time (ms)**	**Ratio (%)**
State estimation (filtering and model update)	0.05 (0.09)	
Proposed SEIKO method	**0.47 (0.89)**	100%
– Jacobian and gravity vector (*RBDL*)	0.03 (0.05)	6%
– Analytical partial derivatives (*Pinocchio*)	0.06 (0.13)	12%
– Cost, equalities and inequalities matrices	0.11 (0.24)	23%
– QP solver	0.16 (0.31)	34%
– Joint torques	0.11 (0.19)	23%
Inverse dynamic QP controller	0.28 (0.55)	
Total control cycle	0.81 (1.48)	

A. Computational Time

The average and maximum computing times measured on an embedded mini-PC (Intel NUC, Intel Core i7-3615QE (2.30 GHz)) with a real-time Linux kernel for one control cycle is given in Table II. The proposed method achieved good real-time performances thanks to the analytical partial derivatives and the decomposition of the motion equations (15) and (16). According to our extensive tests, the average computing time is fairly stable; however, the maximum time can vary depending on the kernel, CPU core binding, and other scheduling configurations of the operating system.

As previously mentioned in Section III-F, the contact switching algorithm 1 runs online and does not cause additional computing cost. Also, it can run offline for verification by computing only SEIKO's retargeting without executing the controller, to see if a contact switch is feasible without actually moving the robot. Assuming an update frequency of $1000\,\mathrm{Hz}$, the transition requires 2309 iterations to finish with the parameters listed in Table I. The transition duration is therefore $2.309\,\mathrm{s}$, which can be computed offline in about $0.560\,\mathrm{s}$ only, in the case of the Valkyrie robot.

B. Convergence of the Equality Constraints



(a) Kinematics (b) Equilibrium (c) Computing time

Fig. 4: Comparison of SEIKO and SLSQP NLopt by number of iterations: The average (dots) and maximum errors (triangles) of the kinematic (a) and equilibrium (b) constraints; (c) average computational time per control loop.

We compared SEIKO with the constrained nonlinear optimization SLSQP algorithm [31] provided by the NLopt library as a baseline. The SLSQPT was implemented with the same constraints (Section III-C) and the same objectives (Section III-B), but employs direct decision variables as the configuration of the system $\left[\boldsymbol{q}^{\mathrm{d}}, \boldsymbol{\tau}^{\mathrm{d}}, \boldsymbol{\lambda}^{\mathrm{d}}\right]^{\mathsf{T}}$, compared to our formulation of using increments. Fig. 4 shows the residual errors of the kinematic (6) and equilibrium (5) constraints as well as the computing time for one time step. On the contrary to SEIKO, SLSQP relies on a line search algorithm to improve its convergence. Our tests showed that SLSQP requires multiple iterations to converge to satisfying constrain errors, where a single iteration does not produce a viable result.

In contrast, with only one iteration, our proposed SEIKO can achieve the position error below $1\,\mathrm{mm}$ for kinematics, and the force error below $0.01\,\mathrm{N}$ for the equilibrium constraints, which are all small enough to ensure accuracy for the teleoperation. These sub decimal scale of errors are especially negligible, compared to the uncertainties in the model and the controller. It shall be noted that in some cases, SLSQP produces discontinuous successive solutions and thus jerky motions when the solution lies on the edge of the feasibility boundaries. The saturation of inequality constraints severely impedes the convergence of the SLSQP algorithm even with an increased number of iterations. More details are given on this phenomenon in supplementary materials in Section 7.

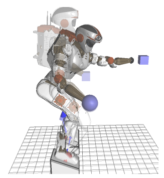
C. Validation of Online Teleoperation

The capabilities of our formulation are evaluated on four tasks: (a) extreme reaching motions beyond the feasibility boundary, (b) hand pushing, (c) contact switching, and locomotion on complex uneven terrain. We validated the whole control architecture in the Pybullet simulator, including SEIKO and a dynamic controller for tracking of the desired configuration. Both SEIKO retargeting and the whole body dynamic controller run at $1000\,\mathrm{Hz}$. We show the teleoperation of Valkyrie in Fig. 5 and ANYmal in Fig. 6. For these experiments, the operator commanded the robot in real-time though a visualization and keyboard interface.

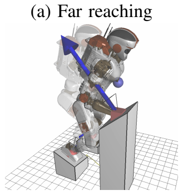
We evaluate the stability of the contacts using the Center of Pressure (CoP) expressed as $CoP_x = \frac{|\tau_y|}{f_z}, CoP_y = \frac{|\tau_x|}{f_z}$, and the friction ratio as $\eta = \frac{\max(|f_x|,|f_y|)}{f_z}$, where $\tau_x, \tau_y \in \mathbb{R}$ are the plane contact lateral torques, $f_x, f_x \in \mathbb{R}$ are the tangential contact forces and $f_z \in \mathbb{R}$ is the normal contact force. The non-tilting condition is satisfied for the feet of the humanoid, when $CoP_x < l_x, CoP_x < l_y$ and the non-sliding conditions is met when $\eta < \mu$, where $l_x, l_y \in \mathbb{R}$ are the foot plane lengths and $\mu \in \mathbb{R}$ is the friction coefficient limit. For the humanoid $l_x = 0.11\,\mathrm{m}$, $l_y = 0.07\,\mathrm{m}$ and $\mu = 0.5$; and for the point-foot quadruped, a more conservative value of $\mu = 0.3$ is chosen.

1) Multi-contact reaching motion under constraints: When the operator commands an extreme forward reaching motion (Fig. 5(a) and Fig. 6(a)), SEIKO updates the desired posture which is tracked by the whole body inverse dynamic controller. The desired and measured positions follow the commanded one until the saturation of some constraints (Section III-C) blocks its further motion to prevent any balance or physical limits violation. The CoP X position of Valkyrie's left foot saturates at the foot edge at $11\,\mathrm{cm}$ (Fig. 5(d), bottom plot) while the reaching motion of ANYmal is constrained by the minimum contact forces (Fig. 6(d), bottom plot).

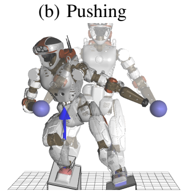
At every time step, SEIKO provides a statically balanced and feasible whole body configuration. The robot can come to rest at any time because of a constraint saturation or a commanded stop and still be safe. Supplementary materials in Section 4 present additional results on the smooth adapted



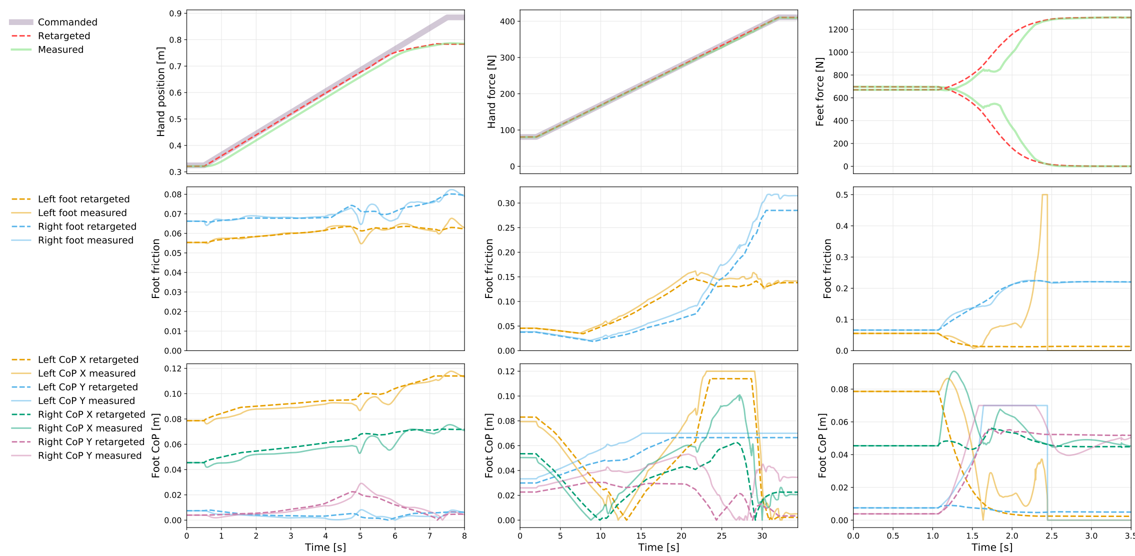
(a) Far reaching



(b) Pushing

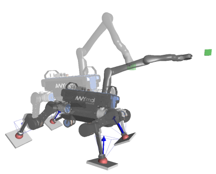


(c) Contact switching

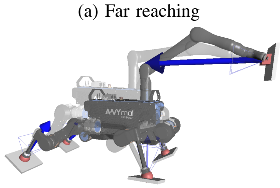


(d) Far reaching (e) Pushing (f) Contact switching

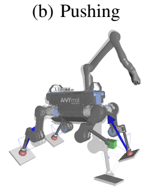
Fig. 5: Multi-contact teleoperation of the Valkyrie robot in Pybullet simulation: (a) far reaching with the left hand; (b) pushing with the right hand; (c) contact switching to disengage and lift the left foot. We compare the operator's command, the desired configuration optimized by SEIKO and the measured one tracked by the dynamic controller, as shown in data plots from (d-f): (d) the position of the left hand during reaching; (e) the contact force of the right hand during pushing; (f) the force distribution among the two feet during the contact switching. For each task, the top row shows the retargeted and measured signals specific to the task, the middle row shows the friction ratio of the feet, and the bottom row shows the CoP.



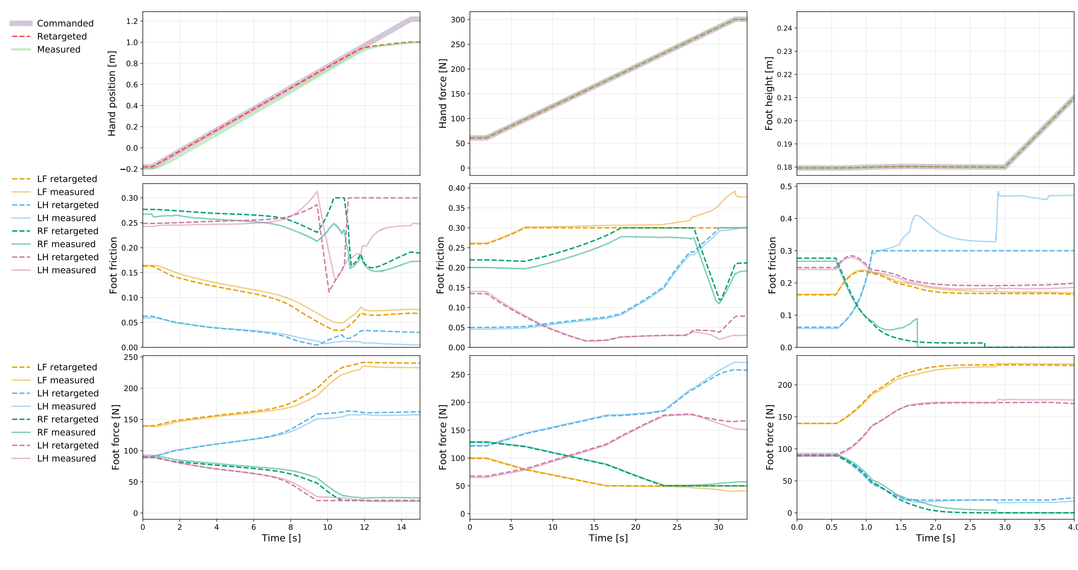
(a) Far reaching



(b) Pushing



(c) Contact switching



(d) Far reaching (e) Pushing (f) Contact switching

Fig. 6: Multi-contact teleoperation of the ANYmal robot in Pybullet simulation: (a) far reaching; (b) pushing; (c) contact switching to lift the front right foot (RF) at 3 s. We compare the operator's command, the desired configuration optimized by SEIKO and the measured one tracked by the dynamic controller, as shown in the data plots from (d-f): (d) the position of the hand during reaching; (e) the contact force of the hand during pushing; (f) the height of the front right foot during contact switching and lifting. For each task, the top row shows the retargeted and measured signals specific to the task, the middle row shows the friction ratio, and the bottom row shows the normal contact force distribution among the feet.

motions produced by SEIKO in response to discontinuous commands. Fast dynamic motions that violate the quasi-static assumption are analyzed and compared in Sections 6 and 11 in the supplementary materials. An additional application for retargeting human motion capture into the humanoid's morphology is also presented in supplementary materials Section 5.

2) Pushing tasks and force manipulability: The operator teleoperated pushing tasks with the right hand in Fig. 5(b) and with the arm's hand in Fig. 6(b) through the commanded normal contact force. We used the optional mode that can command of the contact force (see Section II-A). The commanded force $\boldsymbol{\lambda}_{\text{hand}}^{\text{target}}$ in (4) was gradually increased by the operator and the associated weight $\boldsymbol{w}_{\text{contact, hand}}$ was set to 10^4 to prioritize

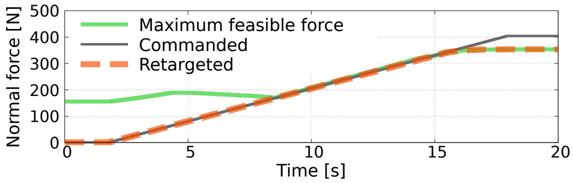


Fig. 7: Contact force from Valkyrie's pushing task. Commanded and retargeted normal forces are compared with the maximum feasible force at the edge of the manipulability polytope.

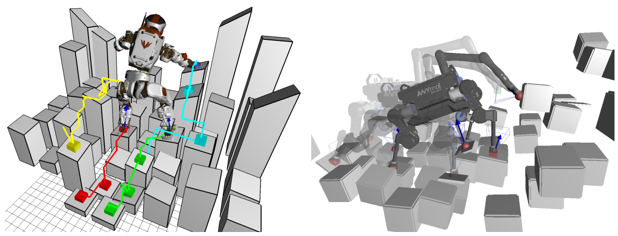


Fig. 8: Teleoperated multi-contact locomotion on uneven terrains using hands and feet for Valkyrie and ANYmal robots. The operator can manually chose the contact sequence, command the motion of the end-effector and activate the contact transition.

the pushing task.

Fig. 5(e) and Fig. 6(e) show on upper row the commanded, retargeted and realized contact force. At first the system is able to increase the applied force by only redistributing the forces among all the contacts with marginal posture change. When the commanded force in Fig. 7 reaches the *maximum feasible normal force* (see supplementary materials Section 3 and [38], [39]), SEIKO tends to update the whole posture by "sliding along the constraints". This postural adaptation increases the maximum feasible force and allows the robot to apply more force until being blocked further by the saturation of the kinematic or contact constraints.

3) Contact switching: The operator triggers the contact switching mechanism detailed in Section III-F to smoothly remove and lift the left foot in Fig. 5(c) and the front right foot in Fig. 6(c). The wrench penalty weight w_{contact} increases exponentially from 1 to 10^5 and drives both the posture change and the force redistribution among the remaining contacts (Fig. 5(f) upper plot and Fig. 6(f) lower plot). With one contact point removed, the saturation of the constraints (friction, CoP, minimal normal force) tends to increase.

4) Locomotion: The command of individual end-effector motions combined with contact switching allows the robot to locomote over uneven terrains, assuming static equilibrium. The operator selects the sequence and commands the location of the multi-contact stances by reaching, probing and triggering contact switch of the end-effectors. The sequences of locomotion for both robots are in Fig. 8, and the details of the retargeted constraints and tracking for the ANYmal robot can be found in Section 8 of the supplementary materials..

V. Discussion

The method we proposed is particularly suitable for interactive teleoperation where high level commands from the operator are retargeted for the robot's morphology. Unlike offline planners which computes the feasible trajectories offline, SEIKO computes feasible references online to be tracked by a dynamic controller. In slow motion cases, both SEIKO and existing approaches work equally well when commanded motions remain within the physical limits of the robot, because no safety precautions are actually required (see Section 10 of supplementary materials). However, when the limits of robots are violated, or the target position commands are infeasible due to operator's mistake, without SEIKO, the whole-body QP controller itself cannot maintain the robot's balance. In contrast, the proposed SEIKO can successfully restrict references within safety boundaries and guarantees the robot's long-term stability.

SEIKO is applicable to multi-contact teleoperation cases which have not been solved by previous approaches, based on two main assumptions: the quasi-static equilibrium and the possible state discrepancy at contact initiation. The quasi-static assumption is required to guarantee the long-term balance because future commands are unknown in the context of teleoperation. Despite this assumption, we have demonstrated that our scheme still achieved acceptable operating velocities in conjunction with the whole-body QP controller (see Fig. 5-6). In Sections 6 and 11 of the supplementary material, we showed that the Valkyrie humanoid's hand can safely reach the velocity of $30\,\mathrm{cm/s}$ in reaching tasks, using parameters that trade off the maximum reachable distance for more conservative postures. Hence, SEIKO still works well for motions of moderate speeds, which suits for a large range of practical loco-manipulation applications, where feasibility and safety are more important than the speed.

The success of teleoperation tasks intrinsically puts dependencies on human motor skills, e.g., our work relies on humans for perception of the environment. To avoid the discrepancies between the model and the actual contact state that could destabilize the robot, the second assumption is that the human operator should command the robot's end-effector to be in contact with a new surface before triggering the contact switching. If the operator makes mistakes such as colliding the end-effector with the environment, the system will rely on the dynamic controller to attenuate such disturbances since the impact force is not part of SEIKO's formulation. Detecting and dealing with unexpected external force perturbations or collisions in the motion retargeting is a promising direction for future work.

SEIKO guarantees the generation of safe and feasible postures, and enhances the safety of intuitive, interactive multi-contact teleoperation, which is robust to human mistakes. This is an advantageous feature in teleoperation, especially when the communication latency can be a major source of human-factor risks. If operator's commands are erroneous due to an impeded communication link, the proposed SEIKO will automatically adapt and convert wrong commanded motions into viable solutions to ensure safety.

Extensive experiments in this work show that few adjustments are needed to transfer the parameters to a different robot once they are tuned (listed in Table I). These parameters have physical significance on how to affect the optimization outcome, so the adjustments are straightforward. Alternatively, Bayesian Optimization could be used for automatic tuning [40].

Similar to general nonlinear optimizations, the proposed scheme exhibits local minimum problems, e.g., robotic arms may occasionally have trouble in returning to their initial poses in a near-singular posture. We have mitigated this problem by regularizing the joint position to a default position as a low-weighted task (with parameter $\boldsymbol{w}_{\text{posture}}$).

VI. Conclusion and Future Work

This paper presents an optimization-based motion retargeting method which is suitable for teleoperation of quasi-static multi-contact tasks, such as loco-manipulation – a combination of locomotion and manipulation. We proposed the Sequential Equilibrium and Inverse Kinematics Optimization (SEIKO) to map and adapt the operator's commands into feasible retargeted configurations, as well as a smooth contact switching and transitions for multi-contact tasks. This method has been applied to teleoperate both the Valkyrie humanoid and the ANYmal quadruped robot. The online teleoperation was achieved and validated in the Pybullet and Gazebo simulators, which demonstrated the effectiveness of the proposed method to guarantee the kinematic and dynamic feasibility.

As the future work, it would be beneficial to estimate external contact forces, which can be included as a bias vector in the equilibrium equation of SEIKO. Hence, we can automatically trigger contact switching when an end-effector pushes on the environment. Also, the presented formulation enforces all contact constraints and joint limits, without considering collisions. Since our retargeting formulation is compatible with self-collision avoidance as implemented in [11], [19], [41], collision avoidance can be a future extension as well.

Acknowledgment

Authors would like to thank Justin Carpentier and the developers of the Pinocchio library, and Steve Tonneau for the valuable discussions. This research is supported by the EPSRC Future AI and Robotics for Space (EP/R026092/1), and EU Horizon2020 project Harmony (101017008).

References

[1] M. DeDonato, V. Dimitrov, R. Du, R. Giovacchini, K. Knoedler, X. Long, F. Polido, M. A. Gennert, T. Padır, S. Feng *et al.*, "Human-in-the-loop control of a humanoid robot for disaster response: A report from the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 275–292, 2015.

[2] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman *et al.*, "Team ihmc's lessons learned from the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.

[3] P. Schmaus, D. Leidner, T. Krüger, A. Schiele, B. Pleintinger, R. Bayer, and N. Y. Lii, "Preliminary insights from the meteron supvis justin space-robotics experiment," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3836–3843, 2018.

[4] F. B. de Frescheville, S. Martin, N. Policella, D. Patterson, M. Aiple, and P. Steele, "Set-up and validation of meteron end-to-end network for robotic experiments," in *ASTRA Conference*, 2011.

[5] I. Chatzinikolaidis, Y. You, and Z. Li, "Contact-implicit trajectory optimization using an analytically solvable contact model for locomotion on variable ground," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6357–6364, 2020.

[6] F.-J. Montecillo-Puente, M. N. Sreenivasa, and J.-P. Laumond, "On real-time whole-body human to humanoid motion transfer," in *ICINCO*, 2010.

[7] J. Koenemann, F. Burget, and M. Bennewitz, "Real-time imitation of human whole-body motions by humanoids," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2806–2812.

[8] K. Darvish, Y. Tirupachuri, G. Romualdi, L. Rapetti, D. Ferigo, F. J. A. Chavez, and D. Pucci, "Whole-body geometric retargeting for humanoid robots," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2019, pp. 679–686.

[9] F. Abi-Farraj, B. Henze, A. Werner, M. Panzirsch, C. Ott, and M. Roa, "Humanoid teleoperation using task-relevant haptic feedback," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'18*, 2018.

[10] J. Oh, O. Sim, B. Cho, K. Lee, and J.-H. Oh, "Online delayed reference generation for a humanoid imitating human walking motion," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 1, pp. 102–112, 2021.

[11] E. M. Hoffman, A. Rocchi, N. G. Tsagarakis, and D. G. Caldwell, "Robot dynamics constraint for inverse kinematics," *Advances in Robot Kinematics 2016*, vol. 4, p. 275, 2018.

[12] M. Vukobratović and B. Borovac, "Zero-moment point—thirty five years of its life," *International journal of humanoid robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[13] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 2001, pp. 239–246.

[14] K. Yamane and Y. Nakamura, "Dynamics filter-concept and implementation of online motion generator for human figures," *IEEE transactions on robotics and automation*, vol. 19, no. 3, pp. 421–432, 2003.

[15] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Bipedal oriented whole body master-slave system for dynamic secured locomotion with LIP safety constraints," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 376–382.

[16] L. Penco, N. Scianca, V. Modugno, L. Lanari, G. Oriolo, and S. Ivaldi, "A multimode teleoperation framework for humanoid loco-manipulation: An application for the icub robot," *IEEE Robotics Automation Magazine*, vol. 26, no. 4, pp. 73–82, Dec 2019.

[17] A. Del Prete, S. Tonneau, and N. Mansard, "Zero step capturability for legged robots in multicontact," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1021–1034, 2018.

[18] Z. Li, C. Zhou, J. Castano, X. Wang, F. Negrello, N. G. Tsagarakis, and D. G. Caldwell, "Fall prediction of legged robots based on energy state and its implication of balance augmentation: A study on the humanoid," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5094–5100.

[19] A. Di Fava, K. Bouyarmane, K. Chappellet, E. Ruffaldi, and A. Kheddar, "Multi-contact motion retargeting from human to humanoid robot," in *IEEE-RAS 16th International Conference on Humanoid Robots*, 2016, pp. 1081–1086.

[20] T. Bretl and S. Lall, "Testing static equilibrium for legged robots," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 794–807, 2008.

[21] R. Shigematsu, M. Murooka, Y. Kakiuchi, K. Okada, and M. Inaba, "Generating a key pose sequence based on kinematics and statics optimization for manipulating a heavy object by a humanoid robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3852–3859.

[22] K. Bouyarmane and A. Kheddar, "Humanoid robot locomotion and manipulation step planning," *Advanced Robotics*, vol. 26, no. 10, pp. 1099–1126, 2012.

[23] S. Brossette, A. Escande, and A. Kheddar, "Multicontact postures computation on manifolds," *IEEE Transactions on Robotics*, 2018.

[24] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the atlas robot," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014.

[25] Z. Li, C. Zhou, N. Tsagarakis, and D. Caldwell, "Compliance control for stabilizing the humanoid on the changing slope based on terrain inclination estimation," *Autonomous Robots*, vol. 40, 08 2016.

[26] E. S. Neo, K. Yokoi, S. Kajita, F. Kanehiro, and K. Tanie, "A switching command-based whole-body operation method for humanoid robots," *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 5, pp. 546–559, 2005.

[27] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[28] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, “A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks,” in *2009 International Conference on Advanced Robotics*. IEEE, 2009, pp. 1–6.
[29] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. G. Tsagarakis, “Opensot: A whole-body control library for the compliant humanoid robot coman,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
[30] P. E. Gill and E. Wong, “Sequential quadratic programming methods,” in *Mixed integer nonlinear programming*. Springer, 2012, pp. 147–224.
[31] D. Kraft *et al.*, “A software package for sequential quadratic programming,” 1988.
[32] S. Caron, Q.-C. Pham, and Y. Nakamura, “Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
[33] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, “Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid,” *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.
[34] M. L. Felis, “Rbdl: an efficient rigid-body dynamics library using recursive algorithms,” *Autonomous Robots*, pp. 1–17, 2016.
[35] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, “The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives,” in *IEEE International Symposium on System Integrations (SII)*, 2019.
[36] J. Carpentier and N. Mansard, “Analytical derivatives of rigid body dynamics algorithms,” in *Robotics: Science and Systems (RSS 2018)*, 2018.
[37] D. Goldfarb and A. Idnani, “A numerically stable dual method for solving strictly convex quadratic programs,” *Mathematical programming*, vol. 27, no. 1, pp. 1–33, 1983.
[38] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. G. Caldwell, and C. Semini, “Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3363–3370, 2018.
[39] W. J. Wolfslag, C. McGreavy, G. Xin, C. Tiseo, S. Vijayakumar, and Z. Li, “Optimisation of body-ground contact for augmenting the whole-body loco-manipulation of quadruped robots,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3694–3701.
[40] K. Yuan, I. Chatzinikolaidis, and Z. Li, “Bayesian optimization for whole-body control of high-degree-of-freedom robots through reduction of dimensionality,” *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2268–2275, 2019.
[41] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet *et al.*, “Multi-contact vertical ladder climbing with an hrp-2 humanoid,” *Autonomous Robots*, vol. 40, no. 3, pp. 561–580, 2016.

Quentin Rouxel is currently a post-doctoral researcher at the School of Informatics at the University of Edinburgh. He received his PhD degree in 2017 at the University of Bordeaux in the robotic team Rhoban and the computer science laboratory of Bordeaux (LaBRI) and participated several years to the RoboCup competition in the humanoid kid-size league. His research interests are in control and learning for legged and multi-contacts robotic systems.

Kai Yuan received his M.Sc. from the University of Edinburgh in Robotics and Autonomous Systems, and an M.Sc. and B.Sc. in Engineering Cybernetics from the University of Stuttgart.

He is currently a PhD student at the University of Edinburgh with research interests in the control and optimisation of robots and the application of Machine Learning to further enhance the autonomy and intelligence of robots.

Ruoshi Wen is currently a postdoctoral research associate in the School of Informatics at the University of Edinburgh. Her research interests include human robot interaction and collaboration, biomedical signal processing, Bayesian inference, and machine learning. She is dedicated to enhance the grasping and manipulation ability of robots by integrating human intelligence.

Zhibin (Alex) Li is an assistant professor at the School of Informatics, University of Edinburgh. He obtained an joint PhD degree in Robotics at the Italian Institute of Technology (IIT) and University of Genova in 2012. His research interests are in creating intelligent behaviors of dynamical systems with human comparable abilities to move and manipulate by inventing new control, optimization and deep learning technologies.

Supplementary Materials:
Multi-Contact Motion Retargeting using Whole-body Optimization of Full Kinematics and Sequential Force Equilibrium

Quentin Rouxel, Kai Yuan, Ruoshi Wen, Zhibin Li

1 Notations and Definitions

The robotic system is modeled by a kinematic tree of $n \in \mathbb{N}$ joints and positioned in world frame with a floating base of 6 DoFs. Let $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}} \in \mathbb{R}^n$ be the joints' position and velocity vectors, $\boldsymbol{p}_b \in SE(3)$ the pose of the floating base and $\boldsymbol{\nu}_b = \begin{bmatrix} \boldsymbol{\omega}_b \\ \boldsymbol{v}_b \end{bmatrix} \in \mathbb{R}^6$ its Cartesian velocity in world frame. Angular and linear components are expressed using spatial algebra notation. The state position vector of the system is denoted as $\boldsymbol{q} = \begin{bmatrix} \boldsymbol{p}_b \\ \boldsymbol{\theta} \end{bmatrix} \in \mathbb{R}^{7+n}$ with the floating base orientation expressed by a quaternion. The state velocity vector is written as $\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{\nu}_b \\ \dot{\boldsymbol{\theta}} \end{bmatrix} \in \mathbb{R}^{6+n}$. The exponential and logarithmic map of the Lie algebra of $SO(3)$ are denoted $Exp : \mathbb{R}^3 \to SO(3)$ and $Log : SO(3) \to \mathbb{R}^3$. The following notation is used:

$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + \Delta \boldsymbol{q} = \boldsymbol{q}_t + \dot{\boldsymbol{q}} \Delta t \triangleq \begin{bmatrix} \boldsymbol{p}_b \oplus \boldsymbol{\nu}_b \Delta t \\ \boldsymbol{\theta} + \dot{\boldsymbol{\theta}} \Delta t \end{bmatrix},$$

where $\oplus$ is the increment on the Lie algebra of $SE(3)$. The operator $\ominus$ used to compute the change between two poses in $SE(3)$ is defined as:

$$\boldsymbol{X}_1 \ominus \boldsymbol{X}_2 \triangleq \begin{bmatrix} Log(\boldsymbol{R}_1 \boldsymbol{R}_2^\intercal) \\ \boldsymbol{p}_1 - \boldsymbol{p}_2 \end{bmatrix} \in \mathbb{R}^6,$$

where $\boldsymbol{R}_1, \boldsymbol{R}_2 \in SO(3)$ and $\boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathbb{R}^3$ the associated orientations and positions.

We formulate two types of contact that the system can interact with the environment: plane and point contact.

Plane contact generates a 6 DoFs kinematic constraint (e.g., a foot surface). The associated frame's Z axis is normal to the contact surface. Torques and forces applied on the environment are written $\boldsymbol{\lambda}_{\text{plane}} = \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{f} \end{bmatrix} \in \mathbb{R}^6$. The associated Jacobian matrix is written in local frame (with spatial algebra convention) $\boldsymbol{J}_{\text{plane}} = \begin{bmatrix} \boldsymbol{J}_R^{\text{body}} \\ \boldsymbol{J}_P^{\text{body}} \end{bmatrix} \in \mathbb{R}^{6 \times (6+n)}$.

Point contact generates a 3 DoFs kinematic constraint (e.g., a hand fist or point foot). To express the linear contact forces $\boldsymbol{\lambda}_{\text{point}} = \boldsymbol{f} \in \mathbb{R}^3$ within the contact surface frame, the orientation of the surface $\boldsymbol{R}_{\text{body}}^{\text{world}} \in SO(3)$ in world frame must be externally provided. The associated Jacobian (with only linear part) is expressed in world frame $\boldsymbol{J}_{\text{point}} = \boldsymbol{J}_P^{\text{world}} \boldsymbol{R}_{\text{body}}^{\text{world}} \in \mathbb{R}^{3\times(6+n)}$.

Let m_{plane} and m_{point} be respectively the number of plane and point contacts that are currently enabled. The contact wrenches/forces and Jacobian are stacked as:

$$
\begin{aligned}
& l = 6m_{\text{plane}} + 3m_{\text{point}} \in \mathbb{N} \\
& \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_{\text{plane},1}^{\mathsf{T}} & \dots & \boldsymbol{\lambda}_{\text{point},1}^{\mathsf{T}} & \dots \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^l \\
& \boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_{\text{plane},1}^{\mathsf{T}} & \dots & \boldsymbol{J}_{\text{point},1}^{\mathsf{T}} & \dots \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{l\times(6+n)}
\end{aligned} \tag{1}
$$

2 Optimization Constraints

Each joint position $\boldsymbol{\theta}$ is bounded by its physical range:

$$\boldsymbol{\theta}_{\text{lower}} \leqslant \boldsymbol{\theta} \leqslant \boldsymbol{\theta}_{\text{upper}}, \tag{2}$$

where $\boldsymbol{\theta}_{\text{lower}}, \boldsymbol{\theta}_{\text{upper}}$ are the minimum and maximum joint position. A velocity bound can also be introduced similarly when needed.

Each joint torque is constrained by the maximum torque $\boldsymbol{\tau}_{\max}$ limited by the actuators:

$$-\boldsymbol{\tau}_{\max} \leqslant \boldsymbol{\tau} \leqslant \boldsymbol{\tau}_{\max}. \tag{3}$$

For each plane contact, the wrench is denoted as $\boldsymbol{\lambda}_i = [\tau_x, \tau_y, \tau_z, f_x, f_y, f_z]^{\mathsf{T}} \in \mathbb{R}^6$. With $f_{\min} \in \mathbb{R} > 0$ being the minimum normal force applied on a contact surface, the unilaterality of the surface normal force is enforced (no pulling from a contact) as:

$$f_z \geqslant f_{\min}. \tag{4}$$

The Center of Pressure (CoP) is bounded within the contact surface to avoid tilting around its edges. $X, Y \in \mathbb{R} > 0$ denotes the half lengths of the contact surface along x and y axes in local frame:

$$-\begin{bmatrix} Xf_z \\ Yf_z \end{bmatrix} \leqslant \begin{bmatrix} -\tau_y \\ \tau_x \end{bmatrix} \leqslant \begin{bmatrix} Xf_z \\ Yf_z \end{bmatrix}. \tag{5}$$

The tangential forces are bounded within the friction cone linearized as a conservative friction pyramid to prevent sliding:

$$-\begin{bmatrix} \mu f_z \\ \mu f_z \end{bmatrix} \leqslant \begin{bmatrix} f_x \\ f_y \end{bmatrix} \leqslant \begin{bmatrix} \mu f_z \\ \mu f_z \end{bmatrix}, \tag{6}$$

where $\mu \in \mathbb{R} > 0$ is the friction pyramid coefficient.

Finally, the torsional torque is bounded to avoid rotational sliding as:

$$
\begin{aligned}
& \tau_z^{\min} \leqslant \tau_z \leqslant \tau_z^{\max} \quad \text{with} \\
& \tau_z^{\min} = -\mu(X+Y)f_z + |Yf_x - \mu\tau_x| + |Xf_y - \mu\tau_y| \\
& \tau_z^{\max} = +\mu(X+Y)f_z - |Yf_x + \mu\tau_x| - |Xf_y + \mu\tau_y|.
\end{aligned} \tag{7}
$$

The same formulation for plane contacts can be applied for the point contacts, with only the consideration of the normal force unilaterality and the friction pyramid constraints.

3 Maximum Feasible and Admissible Force

In a given posture, the maximum admissible force of the contact normal is computed by a Linear Program (LP) subject to the static equilibrium equality constraint, joint torque and contact inequality constraints. For example, for the robot's left hand, it is formulated as:

$$
\begin{aligned}
&\min_{\boldsymbol{\tau},\boldsymbol{\lambda},\lambda_{\text{left\_hand}}} \ -\lambda_{\text{left\_hand}} \quad \text{such that,} \\
&\boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{S}\boldsymbol{\tau} + \boldsymbol{J}(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{\lambda} + \boldsymbol{J}_{\text{left\_hand}}(\boldsymbol{q})^{\mathsf{T}} \left(\lambda_{\text{left\_hand}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \\
&\boldsymbol{C}_{\text{ineq}} \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \\ \lambda_{\text{left\_hand}} \end{bmatrix} + \boldsymbol{c}_{\text{ineq}} \geqslant \boldsymbol{0}
\end{aligned}
\tag{8}
$$

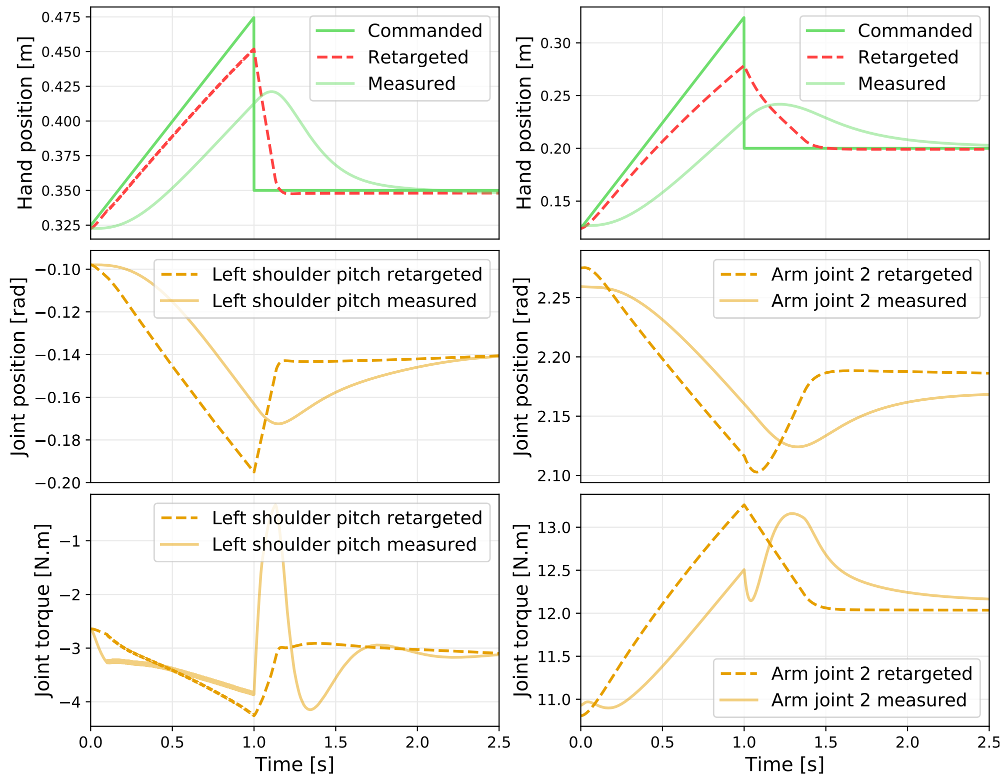
Here, $\boldsymbol{C}_{\text{ineq}}, \boldsymbol{c}_{\text{ineq}}$ represent the convex inequality constraints imposed on the maximum joint torques, the contacts unilaterality and no sliding and tilting conditions. Given the posture, these constraints spawn a convex polytope over the force feasibility set applicable by the end-effector. The maximum feasible normal contact force is the distance to the polytope edge along the normal direction of the surface.

4 Robustness to Discontinuous Commands

Our retargeting optimization can prevent drastic or discontinuous commanded motions from causing the robots to distablize or fall, and hence guarantee the stability and smoothness of the re-adapted motions and be error-proof.

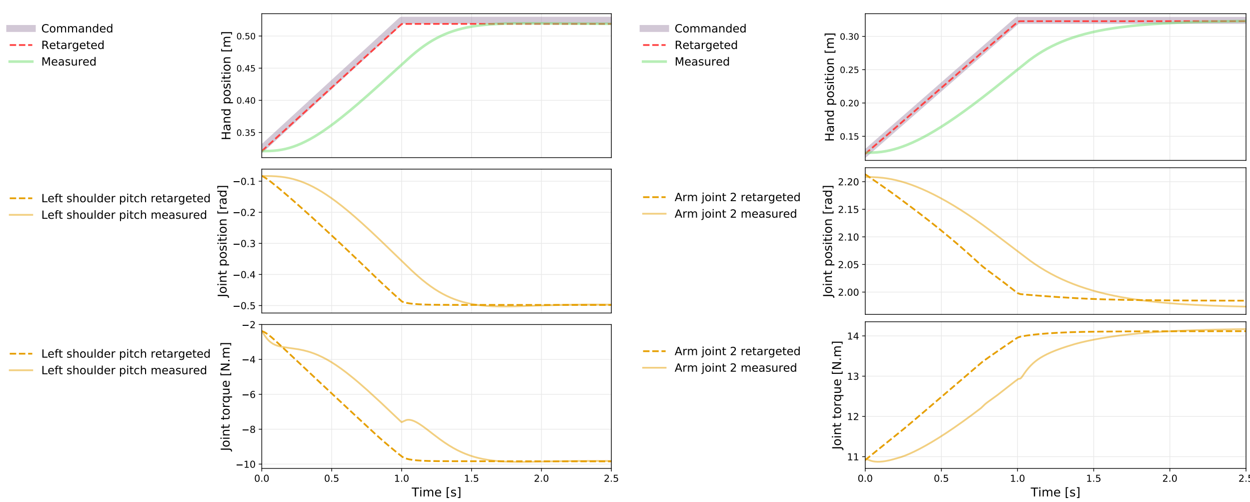
For both Valkyrie and ANYmal robots, discontinuous position commands were intentionally tested on SEIKO, and the resulted data and snapshots from different tests were shown in Fig. 1 and Fig. 2. SEIKO has retargeted the discontinuous position command to the smoother desired position, which avoided spikes in the measured torques. This behavior is regularized by the regularization parameter $\boldsymbol{w}_{\text{velocity}}$.

In the same way, a higher penalty can also be set on the rate of change of contact forces $\Delta\boldsymbol{\lambda}$. This can slow down the change of the desired torque-force configuration e.g. slower CoP change or slower weight redistribution among the contacts. The desired configuration is then easier to track and gain a more robust control and behavior, by trading off the reaction speed.



(a) Valkyrie (b) ANYmal

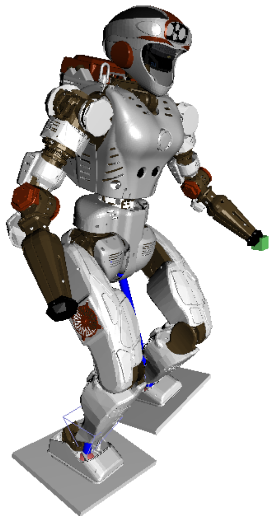
Fig. 1: Smooth trajectories produced by SEIKO retargeting when the operator provides discontinuous position commands. The velocity penalty weight is set to $w_{\text{velocity}} = 10^6$. After receiving a discontinuous position command, the retargeted hand, joint positions, and joint torques of Valkyrie and ANYmal are shown in (a) and (b) respectively. Both are simulated and controlled in PyBullet through real-time teleoperation.



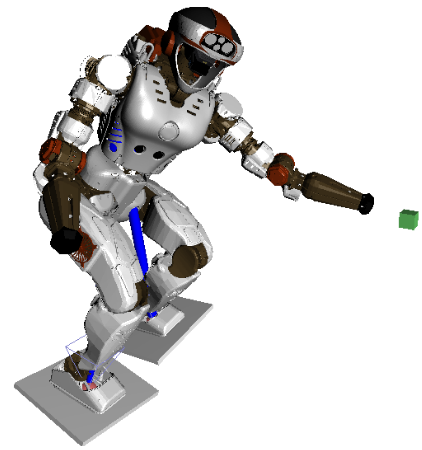
(a) Valkyrie robot (b) ANYmal robot

Fig. 3: Responses of key joints in presence of an emergency stop of the hand (dropping velocity from $0.2\,\mathrm{m\,s^{-1}}$ to $0\,\mathrm{m\,s^{-1}}$) for both (a) Valkyrie and (b) ANYmal robots.

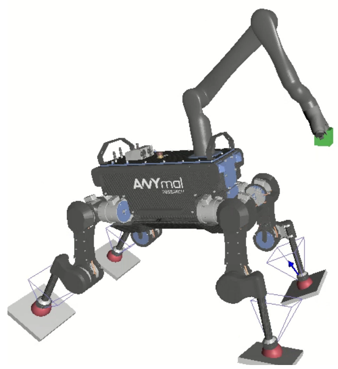
At every time step, SEIKO provides a statically balanced and feasible whole body con-



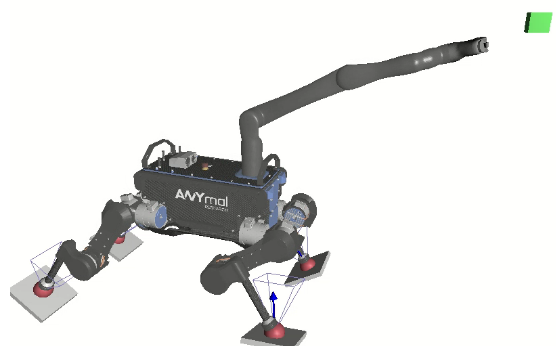
(a) Valkyrie's initial posture



(b) Valkyrie's final posture



(c) ANYmal's initial posture



(d) ANYmal's final posture

Fig. 2: Smooth adaptation of retargeted motions in response to discontinuous commands. The initial posture of Valkyrie and ANYmal are shown in (a) and (c) respectively. After receiving a discontinuous command which has the target position outside the reachable space, the SEIKO optimization automatically adapts the movements of Valkyrie and ANYmal smoothly towards the desired targets as close as possible, with the final postures shown by (b) and (d) from the PyBullet physics simulation.

figuration. The robot can come to a stop at any time, because of a saturating constraint or a commanded stop, and still be safe. Fig. 3 highlights the tracking by the inverse dynamics QP controller at a commanded emergency stop, and also show how the measured state converge towards the desired one.

5 Extended Retargetting Application using Motion Capture

In addition to the pose of the end effectors, the proposed SEIKO method can adapt the whole body motion using human motion capture as input and retarget the motion to a humanoid robot. Online retargeting of human movements on flat ground to the Valkyrie robot on uneven surfaces is shown in Fig. 4. Key features of the human body are measured

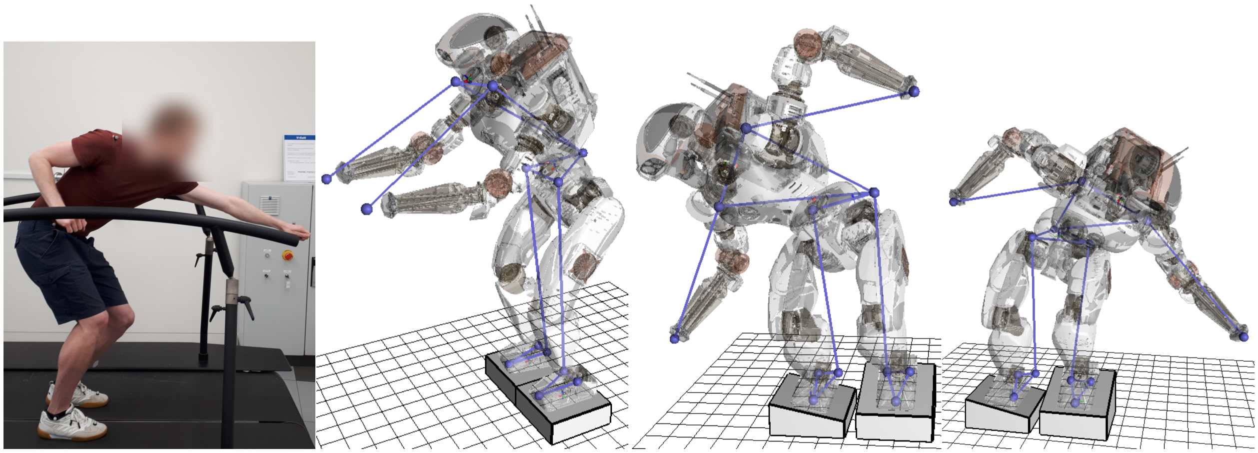


Fig. 4: An additional application of SEIKO for whole body motion retargeting using motion capture. The blue skeleton represents the normalized Cartesian space data measured on the human body. SEIKO can successfully retarget the motion recorded from a flat ground case to uneven terrains by enforcing feasible equilibrium for keeping balance.

and then normalized to match the morphology of Valkyrie. These pose references are added in the cost function of SEIKO optimization with tuned weights.

6 Robustness against Non Quasi-Static Motions in the Gazebo Simulator

Fig. 5 presents the tracking of retargeted motions which are fast and beyond the quasi-static assumption. We use here the simulation environment provided by NASA along with the Valkyrie robot based on the Gazebo simulator. On the contrary to PyBullet, this Gazebo environment simulates the low level delays as well as the noise level of the sensors. The state feedback of the floating base is obtained by state estimation from the noisy IMU sensor. This environment is specifically designed by the engineers of the robot and provides a more realistic setting for validation purposes.

The humanoid performs a forward reaching task at different speeds (similar to Fig. 2(b)). Left hand's velocity range from quasi-static velocity ($4\,\mathrm{cm\,s^{-1}}$) to fast motion ($30\,\mathrm{cm\,s^{-1}}$) plotted in column. The sagittal tracking of the left hand position and the CoP under the left foot are compared. Two sets of SEIKO parameters are also compared.

Default parameters set (upper two rows) uses $w_{\text{contact}} = 10^{-5}$. The conservative parameters set (lower two rows) uses $w_{\text{contact}} = 10^{-3}$ and produces a safer whole body posture more robust to dynamical effects because the CoPs are closer to the center of the feet.

The maximum forward position reached by the left hand is independent of the motion velocity: $1.0\,\mathrm{m}$ for the default parameters and $0.92\,\mathrm{m}$ for the conservative ones. With increased command velocity, the CoP tracking performance worsens, and the realized CoP reaches the limit enforced by the dynamic controller and saturates (in Fig. 5, second row, third and fourth column).

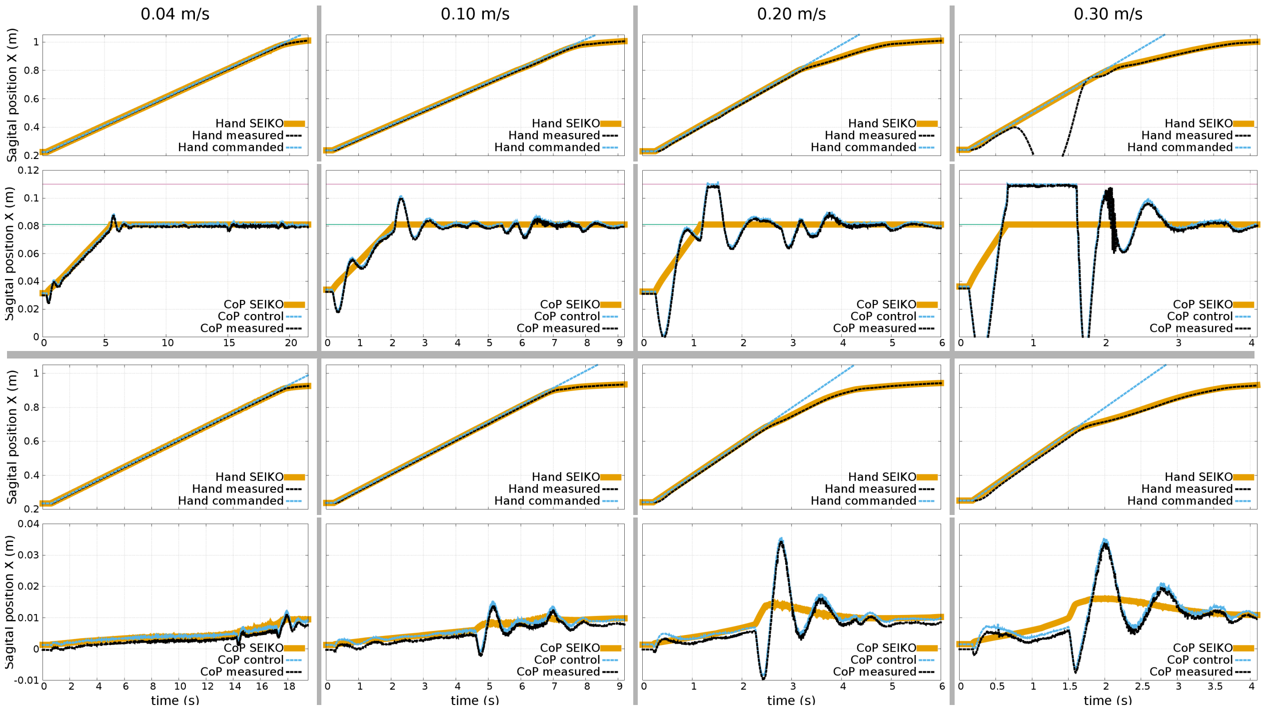


Fig. 5: Tracking performance at different velocities in Gazebo simulator. Default SEIKO parameters are used for the first two rows while more conservative parameters are chosen for the last two rows. First and third rows show the sagittal Cartesian positions of the left hand. Second and fourth rows show the sagittal CoP under the left foot. Note, that the time scale is different for each column.

In special cases such as at the boundaries of the constraints, the tracking of the whole body posture is traded off to maintain the system balance which is more critical, resulting in a large tracking error on the left hand position. With the conservative parameters (last two rows), the desired optimized CoP position remains closer to the foot center, the controlled CoP does not saturate, and the posture tracking is more robust during fast motions. Despite our quasi-static assumption, SEIKO can be tuned to produce conservative retargeted postures to achieve fast motions.

7 Additional Comparison Between SEIKO and SLSQP NLopt

The Fig. 6 shows the time evolution of errors of the equality constraints and computation time, regarding the forward reaching motion of the hand of the Valkyrie robot.

During the motion, the robot was bending forward such that both the torso angular range constraint and the forward CoP constraint were saturated. The configuration of the system was thus lying on the edge of the feasibility space. Non-smooth trajectories produced by SLSQP can be seen on the top right plot. SLSQP is unable to fully and repetitively converge when the solution lies on the constraint boundaries which prevents the position of the hand

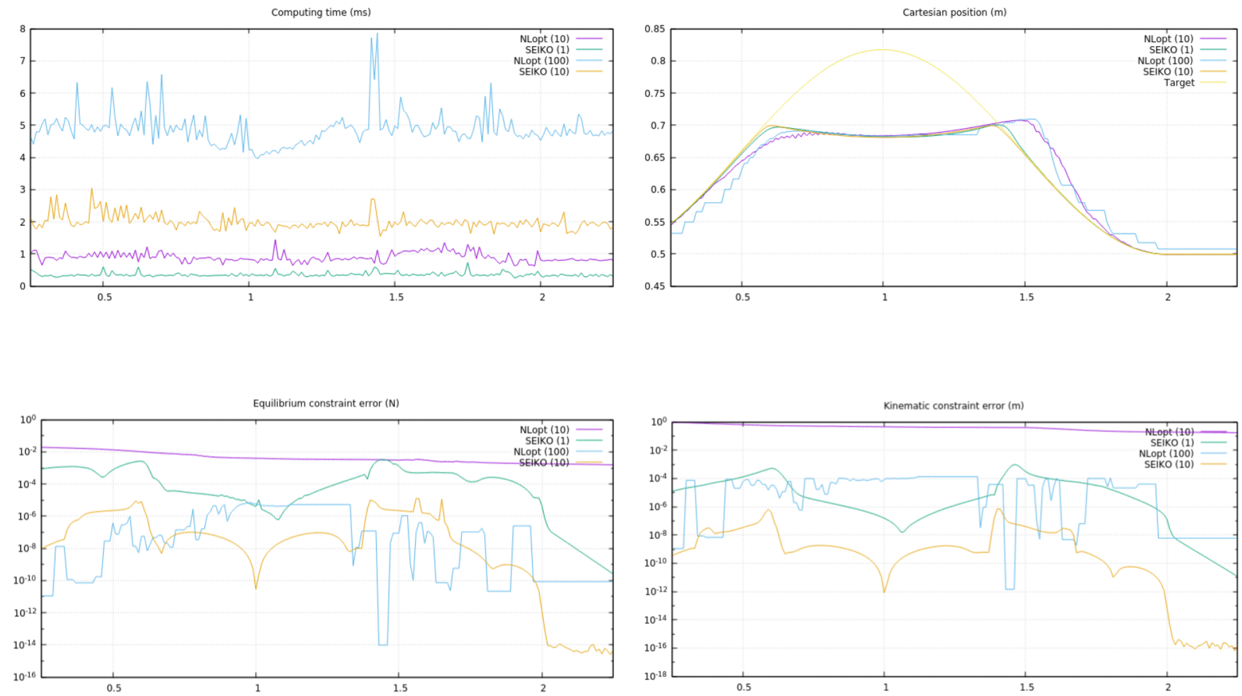


Fig. 6: Comparison between SEIKO and SLSQP NLopt on a forward reaching task: (top left) the computing time; (top right) hand forward position; (bottom left) equilibrium error; and (bottom right) kinematic errors over time (in seconds). We compare SEIKO(1) and SEIKO(10) that run 1 and 10 iterations per time step, and NLopt(10) and NLopt(100) that run 10 and 100 iterations per time step.

to accurately track the commanded target. At the end of the motion, the target position stops to move: the constraint errors of SEIKO for both equilibrium and kinematics converge to zero; in contrast, the errors for SLSQP remain “blocked” because of convergence issues.

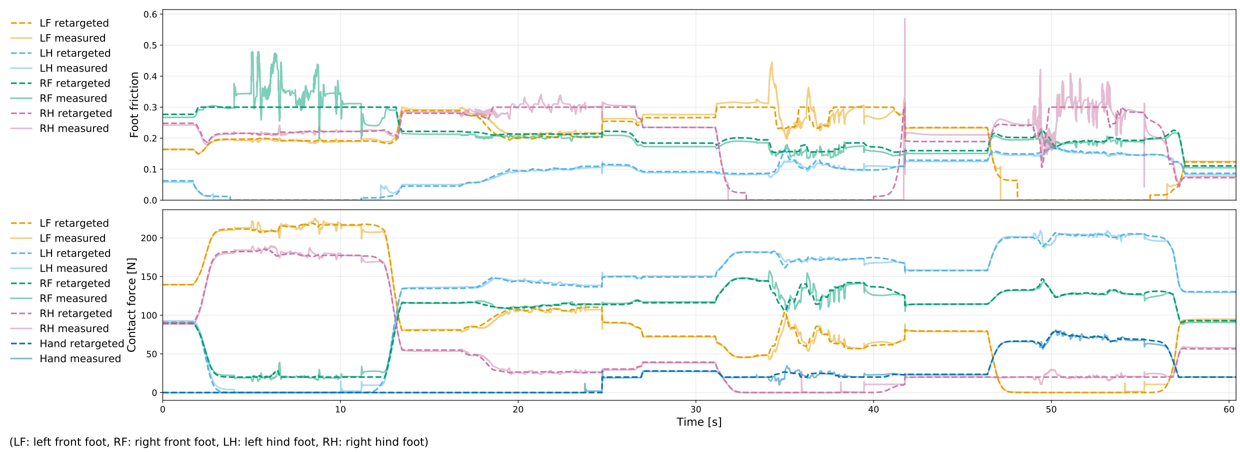


Fig. 7: Teleoperated multi-contact locomotion on uneven terrains of the ANYmal robot. The desired configuration optimized by SEIKO and the measured one tracked by the dynamic controller are compared. In order, the left hind foot (LH) is lifted up, moved and reengaged, the hand contact is moved and enabled, the right hind foot (RH) and then the left front foot (LF) are disengaged, moved and reengaged.

8 Multi-Contact Locomotion on Uneven Terrain

The Fig. 7 details the retargeted and tracked configuration of the ANYmal robot during a short teleoperated locomotion sequence in the Pybullet simulator. The locomotion sequence on very uneven terrain is pictured on the right panel of Fig. 8 in the main document. The operator manually selects the contact sequence, commands the motion of the end-effectors, and disables/activates the contact transitions. The friction ratio (top) and the contact force distribution of the feet and Kinova hand (bottom) are plotted.

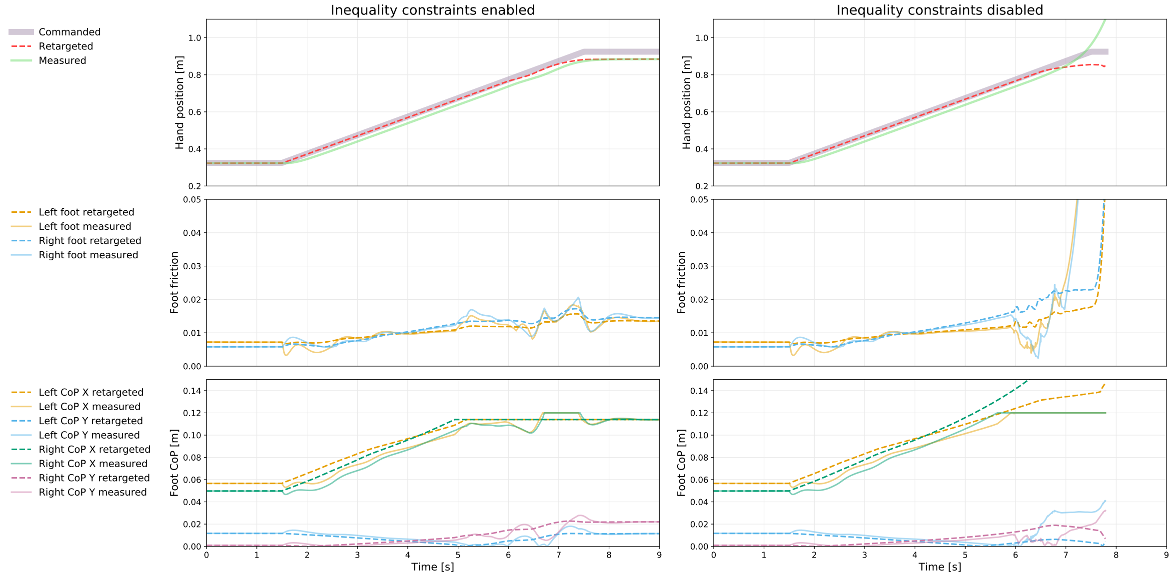
9 Limitations of the Inverse Dynamic QP Controller

As depicted in the architecture diagram in Fig. 3 of the manuscript, our proposed scheme uses a dynamic whole-body controller to realize and track the desired configuration optimized by SEIKO. Our method is agnostic to the choice of the low-level controllers, as long as they track the posture and the contact forces. In this work, we use the "task space inverse dynamic" QP controller which is now the de facto standard approach for torque-controlled humanoid robots. This model-based controller is implemented as a QP which solves the joint accelerations $\ddot{\boldsymbol{q}}$, the joint torques $\boldsymbol{\tau}$ and the contact wrenches $\boldsymbol{\lambda}$ simultaneously while optimizing a set of weighted tasks (see [24] for details). The joint torques $\boldsymbol{\tau}$ are then sent to the actuators of the robot.

For teleoperation applications, we use the QP controller to track desired joint positions, Cartesian positions of end-effectors and the CoM produced by SEIKO while regularizing joint accelerations to prevent too aggressive motions. An additional control task regularizes the contact wrenches to follow the desired contact wrenches, and improves the tracking performance during contact switching when the normal contact forces distribution needs to be changed.

Although the QP controller can track dynamic motions and ensure instantaneous stability, it cannot guarantee long term balance. In fact, the controller reacts aggressively at the edge of the feasibility boundary and eventually fails when the input reference becomes too physically infeasible. Even for dynamic motions, the QP controller requires the reference trajectory to be feasible. Therefore, a high level offline planner usually needs to be implemented to pre-compute a feasible trajectory (either quasi-static or dynamic) for the QP controller. However, offline planners cannot be applied in teleoperation because the operator's future commands are unknown. Especially if the operator commands fast motions, the trajectory sent to the QP controller is dynamically infeasible which causes stability issues. All these unresolved problems motivate us to develop the SEIKO to optimize the feasible configuration in real time.

SEIKO guarantees the static balance of the optimized desired configurations. However, fast trajectories break the quasi-static assumption, and thus are likely to be dynamically infeasible. To address the imperfect tracking of these trajectories, we introduced a conservative margin between the limits used in SEIKO and the physical limits used in the dynamic QP controller. We apply a factor of 0.9 to all limits used in SEIKO (10% margin). E.g., we use a slightly smaller area for the limit of the CoP of the foot defined by the geometry of the rectangle surface to account for tracking errors. By using this factor, we have shown that the teleoperated motion with moderate speed is stable even when reaching the feasibility boundary (see Section 6 and 11).



(a) with SEIKO

(b) without SEIKO

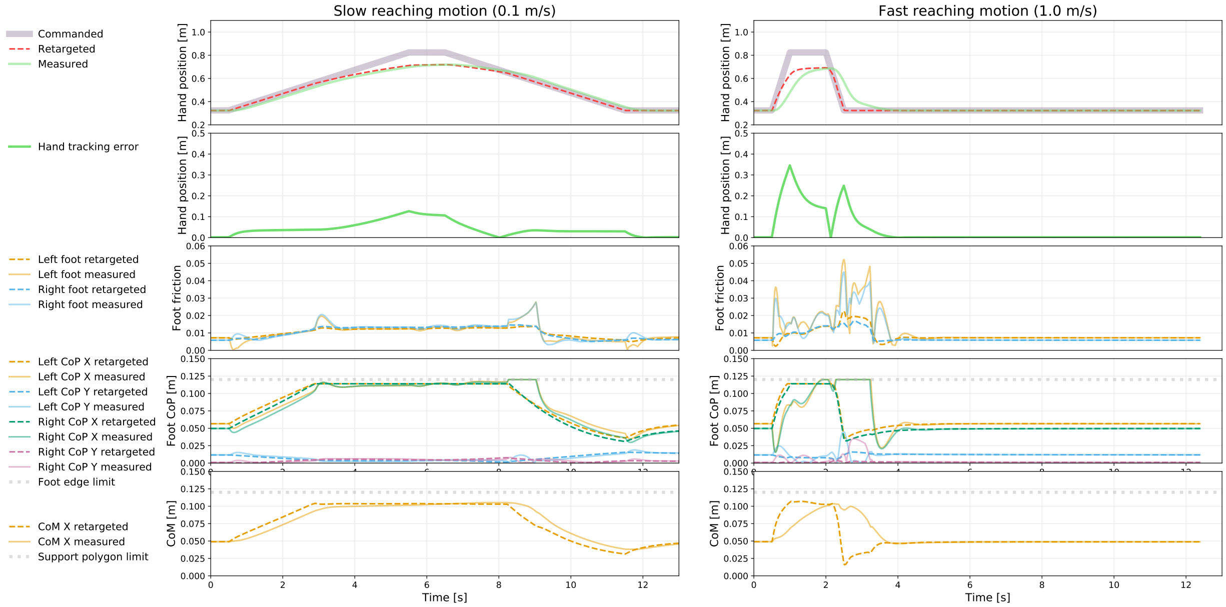
Fig. 8: Comparison with (a) and without (b) feasibility enforcement on Valkyrie humanoid robot performing forward reaching tasks.

10 Importance of SEIKO's Feasibility Constraints

Inverse dynamic QP controllers require the input reference to be feasible. In Fig. 8, we have demonstrated that it is important to restrict the desired configuration to be feasible for the system's safety, even in the quasi-static case. Unstable motions can easily occur after sending quasi-static infeasible references to the controller without SEIKO.

As a comparison study to highlight the effectiveness of SEIKO, we designed an experiment where we commanded the Valkyrie humanoid robot to perform a forward reaching task with its left hand until the target position could not be reached. To compare with SEIKO, we removed the enforcement of physical limits before the commands were sent to the QP controller. When the operator tried to reach a target out of the robot's workspace, references given to the controller were statically infeasible. Without the enforcement of physical limits in SEIKO, the retargeting started to output infeasible configurations once the limits were violated, and the controller quickly became unstable. The tracking of the hand was lost, the feet started to tilt and the robot fell. When we enabled the feasibility constraints, the desired configuration remained bounded and the robot did not fall. Without SEIKO, the control can remain stable only if the physical limits of the robot are not reached, which is a strong assumption because these limits can be very unintuitive for the human operator.

To summarize, the QP controller does not require any safety precautions when we command slow motions within the physical limits of the robot. However, it cannot maintain the robot's balance by itself when the limits are violated, or the target position commands are infeasible due to operator's mistakes. The proposed SEIKO can successfully restrict references within safety boundaries and guarantees the robot's long-term stability.



(a) Quasi-static reaching motion

(b) Dynamic reaching motion

Fig. 9: Comparison between (a) slow quasi-static (10 cm/s) and (b) fast dynamic (100 cm/s) motions with the Valkyrie humanoid robot performing a forward and backward reaching task with its left hand.

11 Validation of SEIKO in Both Quasi-Static and Dynamic Motions

We tested SEIKO in reaching tasks that required quasi-static and dynamic motions. In these two tasks, all settings are exactly the same except the motion speed for comparison. In the PyBullet simulation of the Valkyrie humanoid, we used SEIKO with its default parameters (see Table I in the manuscript) and the inverse dynamic QP controller to realize the motion.

We commanded slow (10 cm/s) and fast (100 cm/s) motions in reaching tasks where the left hand was going out of the reachable workspace and then coming back to the original position. In Fig. 9, we compared the task space performance (end-effector position tracking errors), and the stability performance (feet friction ratio, CoP and projection of the CoM). We considered that the total tracking error between the operator's command and the motion was realized on the robot.

In both cases, SEIKO filtered out and adapted the unreachable target. The tracking error reached its maximum when the commanded position was outside of the workspace. Compared to the slow motion case, the tracking performance for the hand position, the CoPs and the center of mass (CoM) degraded in the fast motion case because the quasi-static assumption was violated.

The plot on the third row in Fig. 9(b) shows that the CoP was saturating at the foot edge (0.12 m), and deviating from its desired reference after 2 s. This indicates that the controller gave more priority to the dynamic balance than the tracking of the desired motion when the balance reduces.

SEIKO is designed for teleoperation cases where the operator's commanded motions are slow, less dynamic, and the feasibility of motions need to be automatically adapted for safety reasons. The effectiveness of SEIKO is validated by its good performance when teleoperating the humanoid Valkyrie in reaching tasks with both quasi-static and dynamic motions.