Multi-Contact Agile Whole-Body Motion Planning via Contact Sequence Discovery and SE(3) Tangent-Space Trajectory Optimization

Ioannis Tsikelis, Evangelos Tsiatsianas, Chairi Kiourt, Serena Ivaldi, Konstantinos Chatzilygeroudis, Enrico Mingo Hoffman

Multi-Contact Agile Whole-Body Motion Planning via Contact Sequence Discovery and SE(3) Tangent-Space Trajectory Optimization

Ioannis Tsikelis[1], Evangelos Tsiatsianas[2,3], Chairi Kiourt[3,4], Serena Ivaldi[1]
Konstantinos Chatzilygeroudis[2,3], and Enrico Mingo Hoffman[1]

***Abstract*— Planning agile whole-body motions for legged and humanoid robots is a fundamental capability for enabling dynamic tasks such as running, jumping, and fast reactive maneuvers. In this work, we present a multi-contact motion planning framework based on bi-level optimization that integrates a contact sequence discovery mechanism, using the Mixed Distribution Cross-Entropy Method (CEM-MD), and an efficient trajectory optimization scheme, which parameterizes the robot's poses and motions in the tangent space of SE(3). The proposed framework permits the automatic generation of feasible contact timings and locations, with associated whole-body dynamic transitions. We validate our approach on a set of challenging agile motion planning tasks for humanoid robots, demonstrating that contact sequence discovery combined with tangent-space parameterization leads to highly dynamic motion sequences while remaining computationally efficient.**

I. Introduction

Humanoid robots are ideal for deployment in the real world. Their human-like form allows them to use existing tools, interact with common objects, and move in environments built for humans. This compatibility significantly reduces integration costs. However, the environments these robots are expected to operate in can be cluttered and complex to navigate. Moving in places like construction sites, factories, and disaster areas requires sophisticated maneuvers, which are generally difficult to plan and execute.

Typical methods for multi-contact loco-manipulation can be divided into three main paradigms: the stance-before-motion paradigm, where the sequence of stances is selected first, followed by a sequence of compatible motions, approaches where motions and contacts are planned simultaneously, or a combination of both.

In the first case, path planning methods are employed to find a feasible path around any possible obstacles, fully defining the contact sequence and positions [1]. This initial planning phase breaks down the task into simpler motions that are executed by a locomotion controller, built to navigate in relatively smooth terrain.

In the second case, models of the robot and the environment are used to plan motions that respect some kinodynamic constraints, usually through the help of a numerical optimizer. This family of methods, also known as trajectory optimization, can generate more complex and dynamic motions involving multiple contacts [2]. However, to be computationally efficient, most of them depend on simplified models of the system's dynamics, or some form of manually defined body motions or contact sequences.

The third case covers numerous methods that integrate techniques used by both of the above approaches. These methods usually split the problem in two parts, initially deciding some gait parameters, including manual selection [3], heuristics, or the use of some integer or black-box optimizer [4], [5], before planning the robot's motion. Execution of each part can be performed numerous times, to fine-tune the executed motion.

We propose a motion planning framework capable of efficiently generating agile, multi-contact motions for humanoid robots in complex environments. Inspired by previous work [4], which introduced a bi-level optimization framework combining template modeling with a black-box optimization method to independently discover feasible contact sequences for motion tasks, we extend this approach by integrating it with whole-body dynamics trajectory optimization scheme. This enhancement enables the method to be applied to more complex systems, such as humanoid robots, rather than being limited to light-limbed legged robots.

Thanks to the black-box optimization method's ability to evaluate multiple contact sequences in parallel and an efficient custom implementation of a trajectory optimizer on SE(3), our method manages to determine feasible solutions within minutes (around 300 s) for the average motion planning task.

II. Related Work

Multi-contact motion planning can be categorized based on how the contact sequence is determined. In general, approaches fall into three categories: those that first select contact points and then plan the motion, also known as the stance-before-motion paradigm; those that incorporate the contact sequence directly into the optimization problem and solve for it jointly; and hybrid methods that fix some contact parameters while optimizing others.

*This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program, and by the JCJC Project ANR-24-CE33-0753-01 MeRLin: Multi-limbed Robots empowered by whole-body Loco-manipulation.

[1]Université de Lorraine, CNRS, Inria, LORIA, Nancy, F-54000, France, ioannis.tsikelis@inria.fr, serena.ivaldi@inria.fr, enrico.mingo-hoffman@inria.fr

[2]Laboratory of Automation and Robotics (LAR) in the Department of Electrical & Computer Engineering, University of Patras, GR-26504 Patras, Greece, etsiatsianas@ac.upatras.gr, costashatz@upatras.gr

[3]Archimedes/Athena RC, Greece

[4]Athena - Research and Innovation Center in Information, Communication and Knowledge Technologies, Xanthi, Greece, chairiq@athenarc.gr

The stance-before-motion paradigm, introduced in [6], fully determine the contact sequence prior to motion planning. Heuristics and sampling-based algorithms [7] are often employed to identify feasible contact sequences, which are then connected using optimization-based planners. For instance, in [1], the authors represent contact configurations as nodes in a graph and apply graph search techniques to find a viable sequence that leads to the desired final configuration. Similarly, [8] also introduces a reachability reduction property that decreases the problem's dimensionality, thereby enabling more efficient planning.

The second category incorporates contact planning directly into the optimization problem. In these approaches, gait parameters, including the sequence, duration, and location of contacts, are introduced either explicitly as optimization variables [9] or implicitly through cost functions and constraints [10]. These problems are generally considered hard to solve because of the additional complexity introduced. Nevertheless, recent advances have made progress in addressing these challenges through techniques such as bi-level optimization [11], relaxation of the linear complementarity constraints introduced in the problem [12], or the use of variational integrators [13].

Finally, hybrid methods integrate elements from both previously described approaches by pre-selecting certain gait parameters while optimizing others. This category encompasses multiple approaches, including manual gait definition [3], [14], formulation of the combinatorial problem as a mixed integer program [15], [5]. In [4], the authors introduced a bi-level hybrid optimization scheme that combines a black-box optimization method with optimization-based motion planning to independently discover feasible contact sequences for given motion tasks.

Despite the wide range of proposed methodologies, there is no universally accepted approach to the contact planning problem. Many methods perform well only over short time horizons or for relatively simple motions, and often rely heavily on engineering heuristics and task-specific tuning to generate valid results.

Another important consideration when setting up a motion optimization problem is how to model the robot. Depending on the complexity of the desired motion and computational constraints, one may select between simplified, i.e, template, models and more accurate physics representations.

In [3], the single rigid body dynamics (SRBD) model is used alongside a cubic spline parameterization of end-effector positions, velocities, and contact forces to generate agile motions for mono-, bi-, or quadrupedal robots.

However, generating more dynamic and realistic motions, especially for humanoid robots, often requires a more detailed model that captures the inertial effects of moving limbs, which are neglected when using the SRBD. The centroidal dynamics model, analyzed in [16], models variable inertia based on the robot's current kinematic configuration, thereby enabling more accurate representation of whole-body momentum.

In [17], the authors implement an optimization-based motion planning framework using the centroidal dynamics model to evaluate the use of analytical Hessians and the inclusion of footstep positions as optimization variables.

The full body dynamics model, as described in [18], accounts for all the links' masses and inertia along a kinematic chain. Despite its higher physical fidelity, it is less commonly used in long-horizon motion planning due to its computational complexity and the need for careful cost function tuning. The full body model is available for use in [19], a motion planning suite that supports multiple numerical solvers. Additional frameworks are presented in [20], [21]. These also include a specialized solver that takes into account the special Euclidean group SE(3) of the robot's orientation representation [22].

In these respects, we propose a motion planning framework capable of efficiently generating agile, multi-contact whole-body motions for humanoid robots in complex environments. To achieve this, we employ a black-box optimization method to evaluate multiple contact sequences in parallel. For the sequence evaluation, we use a custom implementation of a full-body dynamics motion planner that exploits a linearization property of the SE(3), leading to considerably faster optimization times [23]. We assess our method in several multi-contact scenarios that otherwise require good initial guesses and meticulous cost tuning.

III. Background

This section introduces the two methods used in the framework, namely the mixed-distribution cross-entropy black box optimization method and whole-body trajectory optimization. The presentation order reflects the order these two are integrated in our framework.

A. Black Box Optimization and CEM-MD

Black box or derivative-free optimization refers to a subclass of optimization algorithms that do not utilize any derivative information about the cost function (also called fitness function) during the optimization process. This feature allows the use of a bigger subset of fitness functions, including functions that are discrete, non-convex, or non-differentiable.

Cross-entropy methods (CEM) are a family of black-box optimization algorithms that treat the optimization variables as a sample from a parameterized probability distribution $\mathcal{L}_p$. At each iteration, a collection (population) of N samples is drawn from the distribution and given a fitness score from an arbitrary fitness function. The best M population members (the elites) are used to calculate the new parameters p of the distribution. This procedure repeats for K iterations, or until some convergence criterion is met. A useful advantage of this method is that the population evaluation at each iteration can be performed in parallel, minimizing total computation time.

As in [4], we make use of the mixed distribution cross-entropy method (CEM-MD) [24], detailed in algorithm (1). CEM-MD is a member of the CEM family that uses two parameterized distributions: a categorical and a continuous one; Gaussian in our case. This allows the use of both integer

and real-valued variables in our problems (lines 4 and 5). The population evaluation and elite selection is performed on all variable pairs (line 7 and line 8) and the distribution parameter updates are performed independently for the two distributions (in lines 9and 11), at each iteration.

Algorithm 1 Mixed Distribution Cross Entropy Method (CEM-MD)

1: **procedure** CEM-MD(N, K, M, J, C)
2: Initialize $\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1, \boldsymbol{p}_1$
3: **for** $k = 1 \rightarrow K$ **do** ▷ K iterations
4: $\mathcal{P}_{\text{cont}} = \{\boldsymbol{z}_1^c, \ldots, \boldsymbol{z}_N^c\}$, where $\boldsymbol{z}_i^c \sim \mathcal{N}(\boldsymbol{\mu}_k, \text{diag}(\boldsymbol{\sigma}_k^2))$ ▷ Generate continuous population of size N
5: $\mathcal{P}_{\text{discrete}} = \{\boldsymbol{z}_1^d, \ldots, \boldsymbol{z}_N^d\}$, where $\boldsymbol{z}_i^d \sim \text{Categorical}(\boldsymbol{p}_k)$ ▷ Generate discrete population of size N
6: $\mathcal{P} = \{\boldsymbol{z}_1 = (\boldsymbol{z}_1^c, \boldsymbol{z}_1^d), \ldots, \boldsymbol{z}_N = (\boldsymbol{z}_N^c, \boldsymbol{z}_N^d)\}$
7: $\mathcal{F} = \{J(\boldsymbol{z}_1), \ldots, J(\boldsymbol{z}_N)\}$ ▷ Evaluate individuals with objective function J
8: $\mathcal{P}_{\text{elite}} = \{\boldsymbol{z}_1^{\text{elite}}, \ldots, \boldsymbol{z}_M^{\text{elite}}\}$ ▷ Select M elites
9: $\boldsymbol{\mu}_{k+1} = \frac{1}{M} \sum_{\boldsymbol{z}_i^{\text{elite},c} \in \mathcal{P}_{\text{elite}}} \boldsymbol{z}_i^{\text{elite},c}$ ▷ Update continuous mean
10: $\boldsymbol{\sigma}_{k+1}^2 = \frac{1}{M} \sum_{\boldsymbol{z}_i^{\text{elite},c} \in \mathcal{P}_{\text{elite}}} (\boldsymbol{z}_i^{\text{elite},c} - \boldsymbol{\mu}_k)^2$ ▷ Update continuous variance
11: **for** $i = 1 \rightarrow C$ **do** ▷ C categories
12: $\boldsymbol{p}_{k+1}^i = \frac{\text{count\_occurrences(Category}_i\text{)}}{M}$

B. Whole-body Optimal Control and Trajectory Optimization

Optimal control refers to determining the optimal sequence of actions to guide a system to a desired state within a finite time horizon, subject to various constraints. These constraints can include the system's dynamics, action limits, and environmental structure. When optimal control is applied specifically to generate reference trajectories, the process is known as trajectory optimization.

In the context of humanoid or legged robots, this problem can be formally defined as an Optimal Control Problem (OCP) [2]:

$$\begin{aligned} \min_{\boldsymbol{x}(t), \boldsymbol{u}(t)} \quad & \int_{t_0}^{t_f} \ell(\boldsymbol{x}(t), \boldsymbol{u}(t))\, dt + \ell_f(\boldsymbol{x}(t_f)) \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\ & \begin{bmatrix} \mathbf{0}_6 \\ \boldsymbol{\tau}_{\min} \end{bmatrix} \leq \boldsymbol{\tau}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \begin{bmatrix} \mathbf{0}_6 \\ \boldsymbol{\tau}_{\max} \end{bmatrix} \\ & \boldsymbol{x}_{\min} \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_{\max} \\ & \boldsymbol{u}_{\min} \leq \boldsymbol{u}(t) \leq \boldsymbol{u}_{\max} \\ & \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \mathbf{0} \\ & \boldsymbol{x}(t_0) = \boldsymbol{x}_{init} \end{aligned} \tag{1}$$

with:

$$\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{M}(\boldsymbol{q}(t))\dot{\boldsymbol{v}}(t) + \boldsymbol{h}_{\boldsymbol{g}}(\boldsymbol{q}(t), \boldsymbol{v}(t)) - \boldsymbol{J}(\boldsymbol{q}(t))^\top \boldsymbol{\lambda}(t) \tag{2}$$

Here, $\boldsymbol{x}(t) = \begin{bmatrix} \boldsymbol{q}(t)^\top & \boldsymbol{v}(t)^\top \end{bmatrix}^\top$ and $\boldsymbol{u}(t) = \begin{bmatrix} \dot{\boldsymbol{v}}(t)^\top & \boldsymbol{\lambda}(t)^\top \end{bmatrix}^\top$ represent the state and control vectors with $\boldsymbol{q}(t) \in \mathbb{R}^n \times SE(3)$, $\boldsymbol{v}(t) \in \mathbb{R}^{n+6}$ and $\boldsymbol{\lambda}(t) \in \mathbb{R}^{n_c}$ being the robot's configuration, velocities, accelerations and contact forces respectively, n the total actuated degrees of freedom (DOFs), and n_c the total number of contact points. $\boldsymbol{f}(\cdot, \cdot)$ is the dynamical system's ordinary differential equation (ODE) [1] and (2) describes the whole body equations of motion and maps states and actions to the joint torques $\boldsymbol{\tau} \in \mathbb{R}^n$; with $\boldsymbol{M} \in \mathbb{R}^{n+6 \times n+6}$ describing the floating-base robot's inertia matrix, $\boldsymbol{h}_{\boldsymbol{g}} \in \mathbb{R}^{n+6}$ the Coriolis and gravity induced forces and $\boldsymbol{J} \in \mathbb{R}^{3n_c \times n+6}$ a matrix resulting from stacking the contact Jacobians from the root link to the contact points. Finally, $\ell(\cdot), \ell_f(\cdot)$ define running and final costs, and $\boldsymbol{g}(\cdot, \cdot)$ defines any additional constraints, possibly including terrain height and friction, environment bounds, and contact collisions.

The problem presented in (1) is continuous and needs to be *transcribed* in discrete form to be solved with numerical methods. An OCP can be discretized in numerous ways, explained in detail in [25]. For example, in direct transcription methods, e.g., direct multiple shooting, the continuous trajectory is split into N segments (or nodes), with piecewise linear control, and connected with additional defect constraints in the form of numerical integration. The OCP problem thus becomes:

$$\begin{aligned} \min_{\boldsymbol{x}, \boldsymbol{u}} \quad & \sum_{k=1}^{N} \ell(\boldsymbol{x}_k, \boldsymbol{u}_k) + \ell_f(\boldsymbol{x}_{k+1}) \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{integrate}(\boldsymbol{x}_k, \boldsymbol{u}_k, dt) \\ & \begin{bmatrix} \mathbf{0}_6 \\ \boldsymbol{\tau}_{\min} \end{bmatrix} \leq \boldsymbol{\tau}(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \begin{bmatrix} \mathbf{0}_6 \\ \boldsymbol{\tau}_{\max} \end{bmatrix} \ \forall k \in \{1 \ldots K\} \\ & \boldsymbol{x}_{\min} \leq \boldsymbol{x}_k \leq \boldsymbol{x}_{\max} \ \forall k \in \{1 \ldots K\} \\ & \boldsymbol{u}_{\min} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\max} \ \forall k \in \{1 \ldots K\} \\ & \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \mathbf{0} \ \forall k \in \{1 \ldots K\} \\ & \boldsymbol{x}_0 = \boldsymbol{x}_{init} \end{aligned} \tag{3}$$

We can then employ a numerical solver to solve (3) and generate optimal trajectories for our robot.

One considerable aspect when solving the optimization problem above is the fact that $\boldsymbol{q} \in \mathbb{R}^n \times SE(3)$, with $SE(3)$ being the special Euclidean group that describes rotations and translation in three-dimensional space. There exist multiple ways to represent the three-dimensional position and orientation of a body [26]. Usually, for the base link of humanoid robots, a vector $\boldsymbol{r} \in \mathbb{R}^3$ and a unit-norm quaternion $\boldsymbol{\rho} \in SO(3)$ are used to represent its position and rotation, respectively. This particular choice guarantees a representation that is free of singularities. However, since the quaternion is a four-element vector, special care must be taken when integrating it. The generic operation that describes the numerical integration is:

$$\begin{aligned} \dot{\boldsymbol{\rho}}_k &= \frac{1}{2} \boldsymbol{\rho}_k \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_k \in \mathbb{R}^3 \end{bmatrix}, \\ \boldsymbol{\rho}_{k+1} &= \boldsymbol{\rho}_k + \dot{\boldsymbol{\rho}}_k dt, \end{aligned} \tag{4}$$

[1] We denote time derivatives with an upper dot: *e.g.* $\dot{x}$.

Algorithm 2 Trajectory Optimization With Contact Sequence Discovery

procedure TOCSD
  for $k = 1 \rightarrow K$ **do**
    $\boldsymbol{\mathcal{P}} = \text{SamplePop}()$ ▷ Sample contact configurations
    $\boldsymbol{Res}$ = SolveTrajopt($\boldsymbol{\mathcal{P}}$) ▷ Solve TO in parallel
    $\boldsymbol{\mathcal{F}} = \boldsymbol{J}(\boldsymbol{Res})$ ▷ Evaluate TO results
    UpdateDistributions($\boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{F}}$) ▷ Update CEM-MD distributions

which is usually followed by a normalization procedure $\boldsymbol{\rho}_{k+1} \leftarrow \frac{\boldsymbol{\rho}_{k+1}}{\|\boldsymbol{\rho}_{k+1}\|}$ or a constraint, to ensure the resulting quaternion being unit norm. These additions introduce non-linearities to the problem, slowing down convergence [23].

IV. Method Overview

Our method incorporates a bi-level optimization structure to generate feasible contact-rich, agile, whole-body motions for humanoid robots. The outer level samples a population of candidate contact sequences from a model distribution. Each population member is used to solve a trajectory optimization problem. The members are then evaluated based on the quality of each one's solutions, and the population distribution is updated based on the members with the best scores. The procedure is repeated until a convergence criterion is met or after a set number of outer-level iterations. A high-level description of the algorithm can be found in 2, while the population selection and trajectory optimization parts are analyzed below.

A. Contact Sequence Discovery

Inspired by [4], we employ CEM-MD to discover feasible contact sequences for a multi-contact task. As presented in Section III, a population member of the CEM-MD is defined by a categorical and a continuous vector of variables. In our case, these vectors are of the same dimension, and each categorical-continuous element pair defines a contact configuration C:

$$C = (ee_c,\ d), \tag{5}$$

with $ee_c \in \mathbb{N}$ and $d \in \mathbb{R}$ being *encodings* for the end-effectors in contact and for the configuration duration, respectively. For a robot with K end-effectors, we constrain $ee_c \in [0, 2^K - 1] \subset \mathbb{N}$ such that, in the binary encoding of ee_c, a "1" in the k-th digit means that the k-th end-effector is in contact. Moreover, like [4], we employ a log space optimization of the durations, meaning that $d \in [l, u]$, with $l, u \in \mathbb{R}^-$, such that $e^l = t_c^{\min}$ and $e^u = t_c^{\max}$, $t_c \in \mathbb{R}^+$ denoting the duration of the contact configuration.

This approach differs from [4], where the authors use CEM-MD to define the number of steps for each end-effector and the durations for the maximum number of phases, leading to possibly unused continuous variables. Our encoding is more concise and drops the need for heuristics to narrow down the search space.

B. SE(3) Tangent-Space Trajectory Optimization

Evaluating the candidate solutions that are generated by the outer loop requires a fast and reliable motion planner. We employ the whole-body dynamics transcription described in Eq. (3) to generate more realistic motions for humanoid robots. To address the hybrid motion planning problem, we split the problem into phases of pre-specified durations. Each phase defines its own set of constraints, based on the end-effectors that are in contact or flight, and is connected to its neighboring phases through additional constraints. For each end-effector i in phase node k we define the following constraints when:

$$\text{in contacts:} \quad \begin{cases} \boldsymbol{\lambda}_i \in \mathcal{FC}(\mu, \boldsymbol{n}), \\ \boldsymbol{\lambda}_i \cdot \boldsymbol{n} > 0, \\ p_{i_z} = h(pi_x, p_{i_y}), \quad i \in \mathcal{C}, \\ \dot{\boldsymbol{p}}_i = \boldsymbol{0} \end{cases} \tag{6}$$

$$\text{not in contact:} \quad \begin{cases} \boldsymbol{\lambda}_i = \boldsymbol{0}, \\ p_{i_z} > h(p_{i_x}, p_{i_y}), \quad i \notin \mathcal{C}, \end{cases} \tag{7}$$

where $\mathcal{C}$ is the set of end-effectors in contact, $\boldsymbol{p}_i \in \mathbb{R}^3$ describes the end-effector position in three-dimensional space, $\mathcal{FC}$, the linearized Coulomb friction cone, with associated $\boldsymbol{n}$, the ground normal vector and μ friction coefficient, and $h(\cdot, \cdot)$ the terrain height at coordinates x, y. We should note that $\boldsymbol{p}_i$ is calculated by the robot's forward kinematics $\boldsymbol{k}(\boldsymbol{q})$ from the base link to the end-effector frame.

1) Special Euclidean Group: The Special Euclidean group SE(3) models rigid-body poses as a Lie group whose tangent at the identity, $\mathfrak{se}(3)$, is a 6-dimensional Lie algebra. Using a compact vector coordinate $\boldsymbol{\xi} \in \mathbb{R}^6$ for twists, one has [27], [28]:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ \boldsymbol{0} & 1 \end{bmatrix}, \quad \boldsymbol{R} \in \text{SO}(3),\ \boldsymbol{p} \in \mathbb{R}^3, \tag{8}$$

$$\text{Exp}(\boldsymbol{\xi}) = \boldsymbol{T}, \quad \text{Log}(\boldsymbol{T}) = \boldsymbol{\xi}, \tag{9}$$

$$\boldsymbol{T}_{k+1} = \boldsymbol{T}_k \oplus \Delta\boldsymbol{\xi} = \boldsymbol{T}_k \,\text{Exp}(\Delta\boldsymbol{\xi}), \tag{10}$$

$$\Delta\boldsymbol{\xi} = \boldsymbol{T}_{k+1} \ominus \boldsymbol{T}_k = \text{Log}\big(\boldsymbol{T}_k^{-1}\, \boldsymbol{T}_{k+1}\big). \tag{11}$$

Left and right Jacobians then map small twists into manifold perturbations, enabling efficient gradient-based trajectory optimization.

2) SE(3) Tangent Space Operations: From Eq. (3), three key choices arise: (1) the optimization variables, (2) the method for measuring state differences (e.g. for constraint enforcement or cost evaluation), and (3) the integration scheme. For the SE(3) Tangent-space, we make the following choices (for each node k):

$$\begin{aligned} &\text{Optimization Variables:} && \boldsymbol{\xi}_k \\ &\text{Difference Operator:} && \text{Exp}(\boldsymbol{\xi}_2) \ominus \text{Exp}(\boldsymbol{\xi}_1) \\ &\text{Integration Operator:} && \text{Log}(\text{Exp}(\boldsymbol{\xi}) \oplus \mathcal{V}_b h) \end{aligned} \tag{12}$$

where we focus on the representation of the floating-base (i.e., we show only the variables related to the floating-base

pose representation), and $\mathcal{V}_b \in \mathbb{R}^6$ is the body twist of the floating-base (part of the generalized velocities vector). The integration approach in Eq. (12) is exact (given the time discretization), and does not suffer from inaccuracies of the integrator scheme as the quaternion integration in Eq. (4).

V. Experimental Evaluation

We showcase our method's ability to generate feasible multi-contact motions in two different environments: a straight corridor equipped with handrails, where the robot can utilize both its feet and arms to propel itself forward, and a chimney-like environment, where the robot must ascend by pushing against opposing surfaces. In all our evaluations, we are interested in quickly discovering a feasible motion profile, thus, we evaluate our CEM-MD population members based on the number of constraint violations after a set number of iterations in the trajectory optimization numerical solver.

In all scenarios, our approach successfully finds feasible solutions most of the times, within the few iterations of CEM-MD specified. For our experiments, we used the model of the robot Talos from PAL Robotics [29], a 1.75 m tall, 95 kg, 32-DOFs humanoid. During the optimization loop, we were not checking for self collisions of the robot's links. The evaluation code for our framework was written in Python using the respective Python bindings of the Pinocchio dynamics library [30], for calculating the whole-body dynamics and their derivatives, and the Ipopt non-linear numerical solver [31] with the HSL MA97 linear solver internally. The experiments were ran on a Dell Precision 3591 laptop computer equipped with an 16-core Intel Core Ultra 7 155H CPU that can reach up to 4.8GHz clock time. In the supplementary video, you can find visualizations of the generated behaviors.

The hyper-parameters used for the CEM-MD algorithm are listed in I:

TABLE I
CEM-MD Hyper-parameters

	Hyper-parameters Per Environment	
Parameter	Sec. V-A	Sec. V-B
Population size (N)	8	8
Elites size (M)	4	4
Maximum Iterations (K)	5	5
Termination Condition	*Feasible Solution Found*	
$\boldsymbol{\mu}_1$	-1.1 ($\approx 0.3\,s$)	-0.625 ($\approx 0.5\,s$)
Minimum value	-1.6 ($\approx 0.2\,s$)	-1.6 ($\approx 0.2\,s$)
Maximum value	-0.69 ($\approx 0.5\,s$)	-0.35 ($\approx 0.7\,s$)
$\boldsymbol{\sigma}_1$	1	1
$\boldsymbol{p}_1$	$\frac{1}{16}$	$\frac{1}{16}$
Minimum Discrete Value	0 (no ee in contact)	
Maximum Discrete Value	15 (all ees in contact)	
α_{penalty}	-10	-10

A. Handrail Corridor

We initially evaluate our framework in a handrail environment where the robot is positioned in a corridor and can use both its legs and hands to locomote. The handrails were modeled as two long surfaces with upward-facing normals. In order to further speed up convergence and avoid naive random solutions, such as when the feet make contact with the handrail, the robot's hands are constrained to only use the rails as contact surfaces, while the feet can only make contact with the ground plane. The handrails environment was selected to showcase our method's ability to quickly generate a diverse set of multi-contact motions.

The robot is initially configured with both feet in contact with the ground. From this starting pose, our method generates a contact sequence to move the robot forward by a target distance of three meters. For this experiment, we specify the whole target configuration of our robot, which is the same as the initial pose, translated three meters forward. We also define the final contact configuration manually to ensure that the robot will conclude its motion in a stable pose.

As it can be seen in 3, our method manages to efficiently generate motions in the handrails scenario relatively quickly, with a mean time of around $100\ s$. Specifically, our method managed to find a feasible gait in all (100% success) out of the ten times that it was executed with a different random seed, within the maximum allowed five CEM-MD iterations.

B. Chimney Environment

To approximate a chimney-like environment, we model the terrain as two vertical, parallel walls, each positioned $0.5\ m$ away from the robot. The robot is initialized with its hands and feet on the walls and is tasked with climbing to a predefined height. The initialization was achieved by manually adding contact stances in our gait profile to achieve the desired configuration; this choice was made for practical reasons in the code, more specifically, the way the end-effector position constraint vector is generated, as described in Eq. (6) and Eq. (7). The target height was added as an additional constraint on the final node of the trajectory, in the form of a displacement $\Delta z \in \mathbb{R}$ added to the initial height of the robot's base link.

For the scope of this work, we present the results of two experiments in the chimney-like environment; one where $\Delta z = 1\ m$ and another where $\Delta z = 3\ m$, while keeping the number of CEM-MD optimized contact configurations the same for both experiments. Consequently, our method managed to generate some very fluid motions of the robot moving left to right on the walls before reaching the lower height, whereas the motions generated to reach the higher target position were more deliberate to gain altitude.

The time results of the chimney environment experiments are shown in 3. For a target of $\Delta z = 1\ m$, our method managed to find a solution eight out of ten times (80% success) within the five CEM-MD iteration limit while for a target of $\Delta z = 3\ m$ it managed to find a feasible solution five out of ten times (50% success) within the first five CEM-MD iterations. Moving in such an environment requires more complex contact sequences and and dexterous whole body motions. However, when the method succeeded in finding a feasible contact sequence, the results were very agile and dynamic.

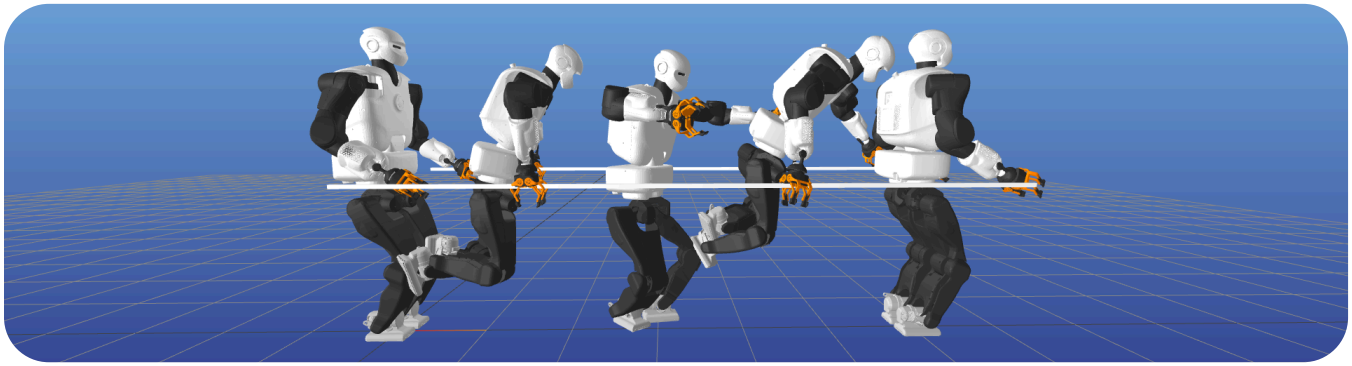

Fig. 1. Snapshots of a generated agile motion sequence for the handrails environment V-A.

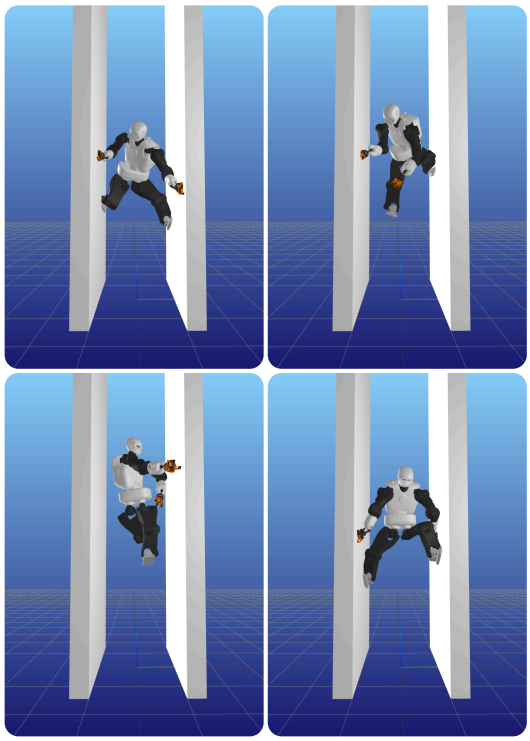

Fig. 2. Snapshots of a resulting motion sequence generated by our framework for the chimney environment V-B.

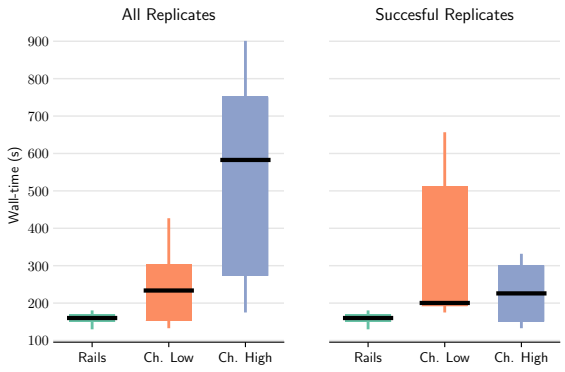

Fig. 3. Wall-time performance for handrails and chimney (low and high targets) environments. The box plots on the left show the median (black line) and the interquartile range (25th and 75th percentiles) over 10 replicates. The box plots on the right show the same information, only for the successful replicates.

This particular scenario has been also considered in previous work [1], with similar solution times (order of hundreds of seconds). It is worth noting, though, that the proposed approach is capable of generating dynamic motions, whereas the work in [1] focused only on planning quasi-static motions.

C. Ablation Study

During our tests, we performed an ablation study on the percentage of elites selected from the population. In the CEM-MD algorithm, during each evaluation round, a subset of members with the best fitness score, i.e., the elites, are selected to update the distribution parameters. The number of the elites is selected as a percentage of the total population. We evaluated three different percentages of elites: 30%, 50%, and 80% of the total population in the chimney environment with the 3 m target base height. We allowed our method to run for a total of five CEM-MD iterations, since we are interested in the short-term performance of our framework to generate feasible gaits quickly. The results are plotted in Figure V-C and show that picking a percentage of around 50% of the population for the elites generally leads to better odds of minimizing the constraint violations and finding a feasible solution earlier. When fewer elites are selected, the updated distribution parameters of the CEM-MD are more heavily influenced by the top members of the population; a disadvantage if their parameters are not near the possible solutions distribution. On the other hand, selecting a bigger percentage of the population as the elites showed slower overall convergence up until the iteration limit we set.

VI. Conclusion

This work presented a motion planning framework based on bi-level optimization for generating agile, multi-contact motions for humanoid robots in complex environments. It was primarily inspired by [4], with the explicit goal of extending that approach to more complex robots, specifically by overcoming the limitations of the SRBD model through the use of full-body trajectory optimization. Given the nature of the outer optimization loop, implemented via the CEM-MD algorithm, which evaluates multiple trajectories in parallel,

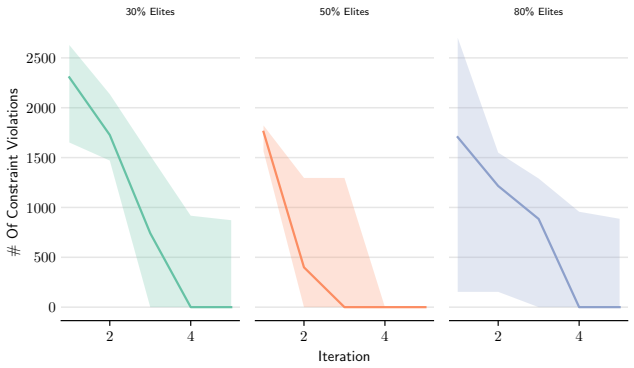

Fig. 4. Ablation study results (Sec. V-C). Number of constraint violations after k-th CEM-MD iteration(s). Solid lines are the median over 10 replicates, and the shaded regions are the regions between the 25-th and 75-th percentiles.

we chose to employ a trajectory optimization method that parameterizes the base pose in the tangent space of SE(3), thereby accelerating the inner-loop optimization in the full robot model. We also improved the contact selection process in the outer loop of CEM-MD by refining it to act at specific locomotion phases. We demonstrated the effectiveness of our approach through experiments involving agile, whole-body multi-contact trajectories in two scenarios: a handrail corridor and a chimney. The results are comparable with the current state-of-the-art in loco-manipulation planning.

Future work will explore the stabilization of the planned trajectories, potentially using Model Predictive Control or Reinforcement Learning techniques. Additionally, we aim to extend the method to handle non-continuous and/or non-convex environment descriptions, which currently remain a limitation of the proposed approach.

REFERENCES

[1] P. Ferrari, L. Rossini, F. Ruscelli, A. Laurenzi, G. Oriolo, N. G. Tsagarakis, and E. Mingo Hoffman, "Multi-contact planning and control for humanoid robots: Design and validation of a complete framework," *Robotics and Autonomous Systems*, vol. 166, 2023.

[2] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2024.

[3] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[4] I. Tsikelis and K. Chatzilygeroudis, "Gait optimization for legged systems through mixed distribution cross-entropy optimization," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 1011–1018, 2024.

[5] A. K. Valenzuela, *Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain*. PhD thesis, Massachusetts Institute of Technology, 2016.

[6] T. Bretl, "Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem," *The International Journal of Robotics Research*, vol. 25, no. 4, pp. 317–342, 2006.

[7] A. Orthey, C. Chamzas, and L. E. Kavraki, "Sampling-based motion planning: A comparative review," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, no. Volume 7, 2024, pp. 285–310, 2024.

[8] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, 2018.

[9] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph.*, vol. 31, July 2012.

[10] G. Kim, D. Kang, J.-H. Kim, S. Hong, and H.-W. Park, "Contact-implicit model predictive control: Controlling diverse quadruped motions without pre-planned contact modes or trajectories," *The International Journal of Robotics Research*, vol. 44, no. 3, pp. 486–510, 2025.

[11] S. Le Cleac'h, T. A. Howell, S. Yang, C.-Y. Lee, J. Zhang, A. Bishop, M. Schwager, and Z. Manchester, "Fast contact-implicit model predictive control," *IEEE Transactions on Robotics*, 2024.

[12] S. Dafarra, G. Romualdi, and D. Pucci, "Dynamic complementarity conditions and whole-body trajectory optimization for humanoid robot locomotion," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3414–3433, 2022.

[13] Z. Manchester, N. Doshi, R. J. Wood, and S. Kuindersma, "Contact-implicit trajectory optimization using variational integrators," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1463–1476, 2019.

[14] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," 2019.

[15] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 279–286, 2014.

[16] D. E. Orin and A. Goswami, "Centroidal momentum matrix of a humanoid robot: Structure and properties," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 653–659, 2008.

[17] A. Papatheodorou, W. Merkt, A. L. Mitchell, and I. Havoutis, "Momentum-aware trajectory optimisation using full-centroidal dynamics and implicit inverse kinematics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11940–11947, 2024.

[18] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer US, 2008.

[19] F. Ruscelli, A. Laurenzi, N. G. Tsagarakis, and E. Mingo Hoffman, "Horizon: A trajectory optimization framework for robotic systems," *Frontiers in Robotics and AI*, vol. 9, 2022.

[20] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[21] W. Jallet, A. Bambade, S. El Kazdadi, C. Justin, and M. Nicolas, "aligator."

[22] W. Jallet, A. Bambade, E. Arlaud, S. El-Kazdadi, N. Mansard, and J. Carpentier, "Proxddp: Proximal constrained trajectory optimization," *IEEE Transactions on Robotics*, vol. 41, pp. 2605–2624, Mar. 2025.

[23] E. Tsiatsianas, C. Kiourt, and K. Chatzilygeroudis, "A Comparative Study of Floating-Base Space Parameterizations for Agile Whole-Body Motion Planning," *preprint*, 2025.

[24] T. Xue, A. Razmjoo, S. Shetty, and S. Calinon, "Logic-skill programming: An optimization-based approach to sequential skill planning," in *Robotics: Science and Systems*, 07 2024.

[25] M. P. Kelly, "Transcription Methods for Trajectory Optimization: A beginners tutorial," Feb. 2015.

[26] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer-Verlag, 2007.

[27] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *CoRR*, vol. abs/1812.01537, 2018.

[28] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2024.

[29] O. Stasse, T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. Del Prete, P. Souères, N. Mansard, F. Lamiraux, *et al.*, "Talos: A new humanoid research platform targeted for industrial applications," in *IEEE-RAS International Conference on Humanoid Robotics (Humanoids)*, pp. 689–695, 2017.

[30] J. Carpentier, F. Valenza, N. Mansard, *et al.*, "Pinocchio: fast forward and inverse dynamics for poly-articulated systems," 2015–2021.

[31] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.