Noname manuscript No.
(will be inserted by the editor)

An Efficient and Direct Method for Trajectory Optimization of Robots Constrained by Contact Kinematics and Forces

Jaemin Lee[1] · **Efstathios Bakolas**[2] · **Luis Sentis**[2]

Received: date / Accepted: date

Abstract In this work, we propose a trajectory generation method for robotic systems with contact kinematics and force constraints based on optimal control and reachability analysis tools. Normally, the dynamics and constraints of a contact-constrained robot are nonlinear and coupled to each other. Instead of linearizing the model and constraints, we solve the optimal control problem directly to obtain feasible state trajectories and their corresponding control inputs. A tractable optimal control problem is formulated and subsequently addressed by dual approaches, which rely on sampling-based dynamic programming and rigorous reachability analysis tools. In particular, a sampling-based method together with a Partially Observable Markov Decision Process (POMDP) solution approach are used to break down the end-to-end trajectory generation problem by generating a sequence of subregions that the system's trajectory will have to pass through to reach its final destination. The distinctive characteristic of the proposed trajectory optimization algorithm is its ability to handle the intricate contact constraints, coupled with the system dynamics, in a computationally efficient way. We validated our method using extensive numerical simulations with a legged robot.

Jaemin Lee
jmlee87@utexas.edu
Efstathios Bakolas
bakolas@austin.utexas.edu
Luis Sentis
lsentis@austin.utexas.edu

[1] Department of Mechanical Engineering The University of Texas at Austin, TX, USA
[2] Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, TX, USA

Keywords Trajectory Optimization · Contact Constraints · Nonlinear Optimal Control

1 Introduction

This paper considers the issue of trajectory optimization problems for robotic systems with state, input, and contact force constraints. Often, it is required that legged or humanoid robots maintain stable foot or body contacts while executing given tasks. In such cases, contact forces constrain and determine the robot's state reachability together with other state and input constraints. Therefore, we seek to devise control algorithms that can generate trajectories for contact-constrained robots via formal state reachability analysis. Frequently, control studies for robotics assume that task trajectories are predefined Khatib [1987], Sentis and Khatib [2005], Mansard et al. [2009], Righetti et al. [2011], then attempt to find an instantaneously optimal solution to accomplish them. However, the desired trajectories are frequently infeasible and it is not straightforward to check the feasibility of trajectories under contact constraints a priori. Many motion planning and trajectory generation approaches for humanoid robots use very simple models that rely on the dynamics of the center of mass under contact constraints Kajita et al. [2003], Stephens and Atkeson [2010], Liu et al. [2015], Lee and Oh [2016]. However, these methods result in lower performance of the robots due to their inability to accurately capture the robot's kinematics and input constraints.

Our paper formulates a problem considering complex robotic systems constrained by contact forces coupled with the system dynamics, state, and input constraints. Before trajectory optimization, it can be ben-

eficial for end-users to check whether desired goals are reachable. Using reachability analysis is one way to guarantee the existence of feasible state trajectories. It can provide useful guidance for high level planning for instance. However, for our purpose, it is very difficult to check that the desired goal is reachable using methods that rely on the solution of the Hamilton-Jacobi-Bellman partial differential equation (PDE), since the contact force is time varying and the robots we consider in our problem are nonlinear and high-dimensional systems. Linearization of dynamics and core approximation of reachable sets with convex sets is not applicable to our problem because reachable sets of constrained nonlinear systems may not be convex. Thus, we devise a new method consisting of propagating system states to obtain the reachable set. After confirming that the goal is reachable via the proposed reachability analysis, we generate an optimal trajectory by using Nonlinear Programming (NLP) tools. Although nonlinear optimization tools and in particular, Sequential Quadratic Programming (SQP) have been utilized for trajectory optimization for robotic systems with contacts in the literature (see Posa et al. [2014]), the use of such methods poses significant computational challenges for high-dimensional systems. Nonlinear programming requires the use of many decision variables and constraints when incorporating both state of the full-body system and the constraint forces. Therefore, we aim to solve trajectory optimization problems for robotic systems with contact constraints based on NLP methods, but in a more computationally efficient way than state-of-the-art methods in the field.

Concretely, we propose an approach that combines sampling-based methods, quadratic programming (QP), NLP tools, and approximation techniques, which are based on the propagation of samples at boundary points of reachable sets to solve our problem. More specifically, we divide the end-to-end trajectory generation problem into small tractable sub-problems by using a sampling-based approach combined with quadratic programming (QP). Then, we reformulate the trajectory optimization problem as a Partially Observable Markov Decision Process (POMDP) with full-body dynamics of the robots considering obstacles in the system's output space. An optimal Markov policy resulting from the application of a dynamic programming (DP) algorithm provides a sequence of output subregions that the system's output has to visit while avoiding unsafe output regions, such as regions comprised of the locations of obstacles in the output space. In the next step, we utilize rigorous reachability analysis tools to determine whether given pairs of subregions can be connected with each other by means of feasible trajectories. In our proposed approach, we propagate the reachable sets of the system starting from a given initial state by propagating the boundary states of the reachable sets corresponding to previous time instances. In this way, we achieve a significant improvement of the algorithmic efficiency of our method, which is one of its distinctive features compared with other state-of-the-art techniques.

We summarize the main contributions of our work in this paper as follows:

- To the best of our knowledge, the proposed method is the first one in making the end-to-end trajectory generation problem with Contact Wrench Cone (CWC) constraints tractable.
- Thus, our method allows end-users to quickly verify that goal output states are reachable via the proposed reachability tool prior to execution of robot behaviors.
- Our method enables to consider a wide variety of robotic constraints, such as joint position/velocity/torque constraints, collision avoidance, contact kinematics and CWC constraints.
- We demonstrate the efficiency of our method by applying it to a 23-DOF humanoid robot.

This paper is organized as follows. We summarize prior work related to trajectory optimization and reachability analysis in Section 2. Section 3 defines our problem and the target class of systems. A sampling-based algorithm for obtaining approximations of the reachable set of constrained systems is proposed in Section 4, and a POMDP-based approach for obtaining an optimal Markov policy of the output sequence is described in Section 5. In Section 6, we propose an approach to obtain the reachable sets that we leverage in Section 7 to design an optimal controller based on NLP techniques. The proposed approach is validated by extensive simulations of a robotic legged system with contact force constraints in Section 7.

2 Related Work

2.1 Motion Planning and Trajectory Optimization

There is a rich literature on motion planning and trajectory optimization for robotic systems. In particular, kinodynamic motion planning has been widely studied and used for the computation of trajectories that satisfy constraints arising from both the kinematics and dynamics of robots. For instance, Rapidly exploring Random Tree (RRT) Kuffner and LaValle [2000], Karaman et al. [2011] and Probabilistic Road Map (PRM) Kavraki et al. [1996] algorithms are employed to solve

motion planning problems of robots. RRT*-smart is an improved version of RRT*, which is based on the combination of optimization techniques and an intelligent sampling strategy Islam et al. [2012]. In Sakcak et al. [2019], a variation of RRT* is proposed in which optimal trajectories are synthesized based on a pre-computed database of motion primitives. A semi-stochastic technique for generating a trajectory fulfilling kinodynamic constraints to reach a goal state is proposed in Sintov [2019]. However, these methods are not suitable for trajectory optimization problems for non-holonomic systems Wieber [2006] or for robotic systems with contact constraints, which are key attributes of various types of robots.

Optimal control is a widely used approach to solving motion planning problems. For instance, Lyapunov functions are utilized to track desired walking trajectories under significant physical perturbations Nguyen and Sreenath [2015], Nguyen et al. [2016]. Several studies have used Linear Quadratic Regulators (LQR), for instance to synthesize local controllers and compute the cost-to-go while fulfilling contact kinematic constraints Posa et al. [2016], or to robustly control trajectories under disturbances Manchester and Kuindersma [2019], or to achieve optimal momentum control under multiple contacts Herzog et al. [2015]. Reference Carius et al. [2018] leverages the iterative LQR algorithm to handle hard contacts with computational efficiency. Although these methods based on LQR are computationally efficient, they require preliminary steps to obtain nominal trajectories and rely on linear system approximations. In contrast, our method generates from scratch the robot trajectories using nonlinear whole-body dynamic models of the robot.

Nonlinear optimization methods are capable of directly handling complicated nonlinear dynamics with contact constraints Tassa et al. [2012], Posa et al. [2014], Mastalli et al. [2019]. In these works, contact kinematics and force constraints are expressed using inelastic impact and Coulomb friction in the form of Linear Complementary Problems (LCP) Tassa et al. [2012], Posa et al. [2014]. However, these state-of-the-art methods do not account for CWC constraints Caron et al. [2015] which better describe the contact states of robots by including horizontal and vertical moments with respect to the contact surfaces. Another difference with respect to these methods is that our proposed method provides a framework for computational efficiency based on sampling methods which the above references don't provide. In addition, Budhiraja et al. [2018], Mastalli et al. [2019] propose computationally efficient optimal control methods for robots with contacts via Differential Dynamic Programming (DDP) but at the cost of ignoring inequality or CWC constraints as we consider in our problem. As a result our method aims at providing a more comprehensive trajectory generation tool that includes more realistic models and a richer set of constraints.

Many studies on legged robots have employed simplified models, e.g., the Linear Inverted Pendulum Model (LIPM) or the Reaction Mass Pendulum Model (RMPM) based on centroidal dynamics, to plan walking motions instead of using full-body models. LIPM-based planners generate Center of Mass (CoM) behavior efficiently due to employing a low-dimensional state, and the planned CoM is tracked via Whole-Body Control (WBC) for both bipedal Kim et al. [2019] and quadrupedal Mastalli et al. [2020] robots. In addition, the centroidal dynamics model accounts for the centroidal momentum to describe upper-body behaviors or to make robots more robust against push disturbances Wensing and Orin [2016]. Based on the centroidal dynamics model, approximated CoM proxy and CWC constraints are employed to generate legged locomotion behaviors by sampling CoM states Carpentier and Mansard [2018]. Also, Fernbach et al. [2020] leverages the centroidal dynamics model to guarantee that CoM trajectories fulfill constraints while making contact transitions.

Although the above methods are practical and computationally efficient, two significant issues arise Lee et al. [2020]. First, there exists a discrepancy between simplified models and more complex full-body models of robots. These methods cannot guarantee that the desired CoM behaviors and contact locations are feasible because simplified models do not explicitly handle joint position/velocity/torque constraints that occur in real robots. Secondly, using simplified models prevents controllers to verify whether multiple task trajectories can be simultaneously executed.

In our approach, in order to strictly enforce the explicit constraints and account for the nonlinearities of the full-body model of robots, thus ensuring our ability to generate feasible trajectories, we propose to directly solve the resulting optimal control problem after converting it to a Nonlinear Programming (NLP) problem. Although many NLP solvers, such as SNOPT Gill et al. [2005] and IPOPT Wächter and Biegler [2006], they involve complex computations that prevent them from being practical for robotic systems. One of the main contributions of this work is the significant mitigation of the computational burden of the nonlinear trajectory generation process based on NLP applied to various robotic systems.

2.2 Reachability Analysis

Reachability or feasibility tools are beneficial to plan complex and dynamic motions of robotic systems fulfilling constraints such as loco-manipulation Jorgensen et al. [2020], Burget and Bennewitz [2015], Yang et al. [2017]. In optimal control studies, reachability analysis is frequently implemented by solving Hamilton-Jacobi-Bellman (HJB) PDE Asarin et al. [2000], Mitchell et al. [2005], Kariotoglou et al. [2013]. Although these methods are effective for low-dimensional dynamical systems, it is hard to perform the HJB-based reachability analysis for high-dimensional constrained systems such as humanoid robots. Other than the HJB-based methods, many approaches have been proposed to compute reachable sets by exploiting mathematical techniques, optimization, inherent characteristics of systems, etc. The logarithmic norm of a type of system's Jacobian is utilized to obtain over-approximated reachable sets for nonlinear continuous-time systems Maidens and Arcak [2015] and that norm is utilized for simulation-based reachability analysis Arcak and Maidens [2017]. Reachability analysis for uncertain nonlinear systems based on conservative approximations can be found in Rungger and Zamani [2018]. Also, for continuous-time piecewise affine systems, linear matrix inequalities (LMI) are employed to characterize the bounds of reachable regions in Hamadeh and Goncalves [2008]. Another class of reachability analysis uses convex approximations of the reachable sets in terms of ellipsoidal sets Kim [2008], polytopes, zonotopes Girard [2005], and support functions Le Guernic and Girard [2010]. In Liebenwein et al. [2018], a sampling-based method is employed to approximately compute reachable sets. However, the method proposed in this reference is not applicable to contact-constrained robotic systems.

Although these approaches are capable of applying nonlinear dynamical systems, they are not suitable for complex robots due to the high-dimensional state space dynamics and intricate constraints. Instead, the reachability analysis is typically carried out on the configuration space of robots. Reachability space is merely employed to speed up an inverse kinematics solver for dexterous manipulation Vahrenkamp et al. [2009]. Reachability indices of a mobile manipulator at potential poses of the base in $SE(2)$ are computed to find the suitable base pose for a given grasping mission Vahrenkamp et al. [2013]. For humanoid robots, a simple sampling method is utilized to generate a sparse reachable area of end-effector with joint limit and self-collision avoidance Lofaro et al. [2012]. Also, a reachability map of a full-body humanoid is proposed to select a stance pose for achieving a grasping task Burget and Bennewitz [2015]. Furthermore, Yang et al. [2017] combines both an upper-body inverse dynamic reachability map and a lower-body dynamic reachability map to generate loco-manipulation behavior. Guan et al. [2008] obtains the reachable space of a humanoid robot using optimization and the Monte-Carlo method. After solving the inverse kinematics of legged parts, random configurations of the upper-body are tested to check whether the CoM position belongs to the support polygon. Jorgensen et al. [2020] proposes a method to find the loco-manipulability region based on the readability of both manipulation and locomotion contact transition.

Reachability has contributed to improving the performance of motion planning in robotics. Nevertheless, most approaches rely on robots' kinematics and simple constraints in position or velocity levels. We need to exploit not only kinematics but also the dynamics of robots to properly handle contact constraints in the reachability tools. However, it is well known that it is difficult to do reachability analysis for high-dimensional systems due to the computational complexity. Therefore, in this paper, we will employ optimal control methods, which will be applied to nonlinear models of the robotic systems and will also implement the reachability analysis of contact-constrained robots in a computationally efficient way, namely through the boundary-states propagation method.

3 Problem Formulation

3.1 Notation

We denote the set of real n-dimensional vectors and the set of real $n \times m$ matrices by $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$, respectively. The set of natural numbers and the set of integer numbers are denoted by $\mathbb{N}$ and $\mathbb{Z}$, respectively. The set of positive definite and positive semi-definite $n \times n$ matrices are denoted by $\mathbb{S}^n_{>0}$ and $\mathbb{S}^n_{\geq 0}$. When considering $z_1, z_2 \in \mathbb{N}$ with $z_2 > z_1$, the discrete interval between z_1 and z_2 is defined as $[z_1, z_2]_{\mathbb{N}} \coloneqq \{z_1, z_1+1, \ldots, z_2-1, z_2\}$. In case that z_1 and z_2 are non-negative real numbers, $[z_1, z_2]^{\Delta}_{d} \coloneqq \{z_1, z_1 + \Delta, \ldots, z_2 - \Delta, z_2\}$ denotes a discrete interval with Δ being the increment. When n real numbers $a_1, \ldots, a_n$ are consider, $\mathrm{Vec}[a_i]^n_{i=1} \in \mathbb{R}^n$ represents a vector whose i-th element is a_i. Given $n \times m$ real numbers $a_{11}, \ldots, a_{mn}$, a matrix whose (i, j) element is a_{ij} is denoted by $\mathrm{Mat}[a_{ij}]^{n,m}_{i,j=1} \in \mathbb{R}^{n\times m}$. Given a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathrm{tr}(\mathbf{A})$ denotes its trace. $\overline{\sigma}(\mathbf{A})$ and $\underline{\sigma}(\mathbf{A})$ represent the largest and smallest singular values of $\mathbf{A}$, respectively. Given matrices $\mathbf{A}_i \in \mathbb{R}^{n_i\times m}$ $i \in [1, z]_{\mathbb{N}}$, $\mathrm{Vertcat}(\mathbf{A}_1, \ldots, \mathbf{A}_z) \in \mathbb{R}^{(n_q+\cdots+n_z)\times m}$ indicates a block matrix constructed by vertically concatenating the matrices $\mathbf{A}_i$ $i \in [1, z]_{\mathbb{N}}$. Given a set of

real vectors $\mathcal{A} \subseteq \mathbb{R}^n$, card($\mathcal{A}$) denotes its cardinality. When considering particular cases such that $\mathcal{A} \subset \mathbb{R}^n$ with $n \in [1,3]_{\mathbb{N}}$, ghull($\mathcal{A}$) and gbd($\mathcal{A}$) represent the general hull and the set of vectors closest the boundary of $\mathcal{A}$. $\mathbb{E}[.]$ represents the probabilistic expectation operator.

3.2 Nonlinear Robotic System with Contacts

We characterize the equation of motion for general robotic systems with contact forces and assuming rigid body linkages as follows:

$$\mathbf{M}(q)\ddot{q} + b(\dot{q}, q) = \mathbf{S}^\top u + \mathbf{J}_c^\top(q) F_c \tag{1}$$

where $q \in \mathbb{R}^{n_q}$, $\mathbf{M}(q) \in \mathbb{S}^{n_q}_{>0}$, $b(\dot{q}, q) \in \mathbb{R}^{n_q}$, $\mathbf{S} \in \mathbb{R}^{n_u \times n_q}$, $u \in \mathbb{R}^{n_u}$, $\mathbf{J}_c(q) \in \mathbb{R}^{n_c \times n_q}$, and $F_c \in \mathbb{R}^{n_c}$ denote the joint variable, sum of Coriolis/centrifugal and gravitational forces, selection matrix for the actuation, input actuating joint torques, contact Jacobian matrix, and contact force, respectively. We can bring the differential equation (1) into a state space form by defining the state $x \coloneqq [x_1^\top x_2^\top]^\top \in \mathbb{R}^{n_x}$ where $x_1 = q$ and $x_2 = \dot{q}$:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} x_2 \\ \mathbf{M}^{-1}(x_1)\left(\mathbf{S}^\top u + \mathbf{J}_c^\top(x_1) F_c - b(x_2, x_1)\right) \end{bmatrix} \\ &= f(x, u, F_c). \end{aligned} \tag{2}$$

The continuous state space model of the robot can be discretized as follows:

$$x(t_{k+1}) = f_D\left(x(t_k), u(t_k), F_c(t_k)\right) \tag{3}$$

where $x(t_k)$, $u(t_k)$, and $F_c(t_k)$ denote the state, input and contact force at time t_k. In addition, $f_D : \mathbb{R}^{n_x + n_u + n_c} \mapsto \mathbb{R}^{n_x}$ is continuous and $\Delta t = t_{k+1} - t_k$. Since the robotic systems are controlled in operational space, the output of the system is a function of the state, $y(t) = f_y(x(t))$, where $f_y : \mathbb{R}^{n_x} \mapsto \mathbb{R}_{n_y}$ is C_2. For instance, we can set the output of the system as the position and orientation of the end-effector, that is $y \in SE(3)$, then the robot would be controlled to achieve the desired output at specific time instance: $y(t_f) = y^g$ where y^g and t_f represent the goal output vector and final time, respectively.

3.3 Constraints of the System

We refer to h_e and h_i as the equality constraint function and the inequality constraint function, respectively, where we assume that $h_e(x) = 0$ and $h_i(x) \leq 0$. The state equality constraint function, h_e, should contain the contact kinematics constraints as follows:

$$\begin{aligned} \phi_i(q) &= 0 \\ \dot{\phi}_i(q, \dot{q}) &= \frac{d\phi_r}{dt} = \mathbf{J}_{\phi_i}(q)\dot{q} = 0 \\ \ddot{\phi}_i(q, \dot{q}, \ddot{q}) &= \frac{d^2\phi_i}{dt^2} = \frac{d\mathbf{J}_{\phi_i}(q)}{dt}\dot{q} + \mathbf{J}_{\phi_i}(q)\ddot{q} = 0 \end{aligned} \tag{4}$$

where $\phi_i : \mathbb{R}^{n_q} \mapsto \mathbb{R}^6$ denotes a continuous mapping for position and orientation errors of the i-th body with respect to the planned contact location and $\mathbf{J}_{\phi_i} = \mathbf{J}_{c_i}$. Also, we consider multiple inequality constraints in joint space as follows:

$$\begin{aligned} &q_{LB} \leq q \leq q_{UB}, \quad \dot{q}_{LB} \leq \dot{q} \leq \dot{q}_{UB}, \\ &u_{LB} \leq u \leq u_{UB} \end{aligned} \tag{5}$$

where subscripts $(.)_{LB}$ and $(.)_{UB}$ refer to lower and upper bounds, respectively. In this paper, it is necessary to explicitly consider the contact force constraint to prevent slip and flit on act surface. The frictional contact wrench cone constraint is shaped by using the state and contact force generated by the robotic system:

$$h_c(x, F_c) \leq 0, \quad h_c(x, F_c) \coloneqq \mathbf{W}_c(x) F_c \tag{6}$$

where $\mathbf{W}_c(x) \in \mathbb{R}^{16 \times n_c}$ is a matrix describing the unilateral constraint using a polyhedral approximation of the friction cone of a surface Caron et al. [2015]. The contact force is coupled to the system dynamics as shown in (1) so that this constraint makes the trajectory optimization problem more difficult.

3.4 Problem Definition

Here we specify the end-to-end problem we are addressing and break it down into tractable subproblems. The end-to-end trajectory generation problem assumes that an initial state, $x_0 = x(t_0)$, and a goal output state, y^g, are given. The problem consists then on obtaining optimal state and input trajectories satisfying state, input, and contact constraints over a finite-time horizon. The end-to-end trajectory generation problem is written as follows:

Problem 1 Consider an initial state $x(t_0)$ fulfilling state equality and inequality constraints, a goal output vector y^g, and finite time interval $T = [t_0, t_f]_d^\Delta$. Our problem is to obtain a locally optimal trajectory for the discretized system in (3) satisfying all constraints to reach the goal output at time instance t_f, that is $\mathbf{X} \coloneqq \{x^d(t_0), \cdots, x^d(t_f)\}$ where $f_y(x^d(t_f)) = y^g$.

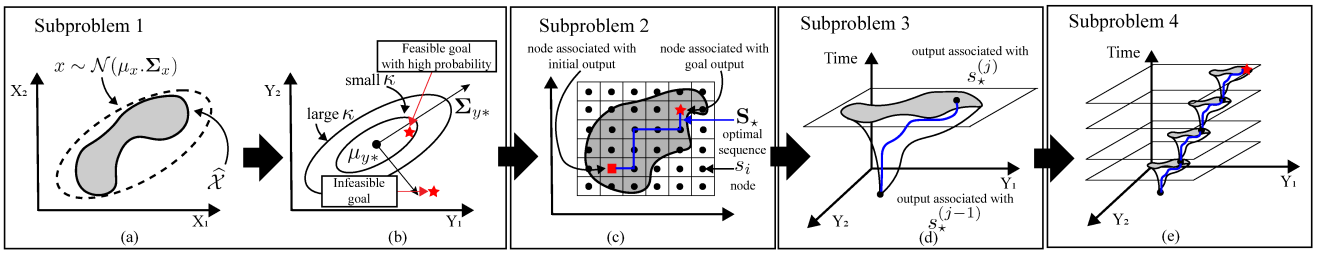

Fig. 1 Subproblems and Proposed method: (a) Generating random state vectors and obtaining a set of sample-wisely feasible states, Section 4.1, (b) Checking the feasibility goal output via output approximation, Section 4.2, (c) Solving POMDP to obtain an optimal sequence of nodes, Section 5, (d) Obtaining reachable sets and optimizing trajectories between sequential nodes, Section 6 and 7 (e) Consecutively executing trajectory optimizations with reachable sets, Section 6 and 7

This problem is implicit and hard to solve in general. In addition, we do not know if the goal output is reachable in first place. Therefore, we divide the entire problem into four tractable subproblems as follows:

Subproblem 1 *Given constraint functions h_e, h_i, h_u, and h_c, we aim to obtain a set of states, $\mathcal{X}$, fulfilling all constraints. We define this set in Definition 1 as the Samplewise Reachable Set.*

Subproblem 2 *Given a set of states fulfilling constraints, $\mathcal{X}$, and an initial state, $x(t_0)$, the problem is now to find an optimal sequence of subregions in the output space such that y^g is reachable.*

Subproblem 3 *Suppose that the above optimal sequence of subregions in output space has been found and we are given the initial state $x(t_0)$, the next subproblem is to check whether the goal output, y^g, is reachable via reachability analysis.*

Subproblem 4 *After checking the goal output, y^g, we obtain a local optimal trajectory, $\mathbf{X}$, by solving a nonlinear constrained optimal control problem.*

We employ optimal control to generate short trajectories based on quadratic cost terms involving output errors. We break down the motion planning problem into various subproblems in output space instead of configuration space. This strategy provides a near-optimal solution for output space trajectory generation. To tackle these subproblems we employ sampling-based optimizations, a POMDP representation, reachability analysis, and nonlinear optimal control. The overall procedure is depicted in Fig. 1.

4 Sample-Based Optimization via QP

In this section, we solve the defined Subproblem 1 by using random samples and QP. The solution of Subproblem 1 is a set of states precisely defined as follows:

Definition 1 (Samplewise Reachable Set) Given constraint functions h_e, h_i, h_u, and h_c, Samplewise Reachable Set (SRS) is defined as a collection of the states satisfying all constraint.

$$\begin{aligned}\mathcal{X} \coloneqq \{x \in \mathbb{R}^{n_x} : h_e(x) = 0,\ h_i(x) \le 0,\ \exists v \in \mathbb{R}^{n_x},\\ v = f_D(x, u, F_c),\ h_e(v) = 0,\ h_i(v) \le 0,\\ h_u(u) \le 0,\ h_c(x, F_c) \le 0,\ u \in \mathbb{R}^{n_u},\ F_c \in \mathbb{R}^{n_c}\}\end{aligned} \tag{7}$$

By the definition of SRS, all elements in $\mathcal{X}$ are able to propagate to the next state fulfilling the addressed constraints by appropriate input and contact force.

In this section, we will obtain SRS by using sampling-based method and a least-square QP process.

4.1 Update for Random State Samples

We start by creating random samples from a Gaussian distribution $z \sim \mathcal{N}(\mu_x, \mathbf{\Sigma}_x)$ where $\mu_x \in \mathbb{R}^{n_x}$ and $\mathbf{\Sigma}_x \in \mathbb{S}^{n_x}_{>0}$ are the mean and the covariance matrix, respectively, that is, $\mu_x \coloneqq \mathbb{E}[z]$ and $\mathbf{\Sigma}_x \coloneqq \mathbb{E}[(z-\mu_x)(z-\mu_x)^\top]$. We compute μ_x by considering the median of Range of Motion (RoM) for each robot joint or initial configuration in relation to joint constraints (6). We assume that $\mathbf{\Sigma}_x$ is a diagonal matrix whose elements are variances of joint positions and velocities. When drawing multiple random samples, we assume that the random samples are statistically independent and uncorrelated. As a first step, we gather the states fulfilling the addressed constraints among the sampled random states. Of course the Monte-Carlo method can be one of candidates to collect the state samples however it is very inefficient. Instead of the Monte-Carlo method, a least-square QP and the gradient descent method are combined to update the sampled states for fulfilling the state constraints.

Firstly, let us consider the state constraint functions h_e, and h_i. Then, the corresponding Jacobian matrices are defined as follows:

$$\mathbf{J}_e(x) \coloneqq \frac{\partial h_e}{\partial x}(x), \quad \mathbf{J}_i(x) \coloneqq \frac{\partial h_i}{\partial x}(x). \tag{8}$$

All constraints are independent then the Jacobian matrices are full row rank. Then, we iteratively update the sampled states as follows:

$$z_{k+1} = z_k + \gamma_k \Delta z_k \tag{9}$$

where z_0 denotes the initially sampled state and $\gamma_k \in \mathbb{R}_{\geq 0}$. For our gradient decent method, we deploy a particular method for computing γ_k.

$$\gamma_k = \frac{|(z_k - z_{k-1})^\top (\Delta z_k - \Delta z_{k-1})|}{\|\Delta z_k - \Delta z_{k-1}\|^2} \tag{10}$$

Δz_k is computed by the following QP.

$$\begin{aligned} \min_{\Delta z_k, w_k} \quad & \|w_k\|^2 \\ \text{s.t.} \quad & \mathbf{J}_e(z_k)\Delta z_k \leq -h_e(z_k) + w_k, \\ & \mathbf{J}_i(z_k)\Delta z_k \leq \varphi_i^d - h_i(z_k) \end{aligned} \tag{11}$$

where φ_i^d refers the desired vector computed by using the initial sampled state as follows:

$$\min_{\varphi_i^d} \quad \|\varphi_i^d - h_i(z_0)\|^2, \quad \text{s.t.} \quad \varphi_i^d \leq \delta_i \tag{12}$$

where $\delta_i = [\alpha, \cdots, \alpha]^\top \in \mathbb{R}^{\dim(h_i(z_0))}$ and $\alpha < 0$. The proposed method is a numerical iteration combined with QP so that we have to determine the termination tolerance of this iterative process. We terminate the iteration if the error $\|h_e(z_k)\| \leq \varepsilon$ where ε is a very small numerical tolerance. Then, it is evaluated whether the inequality constraints are fulfilled in terms of the update state. We obtain a set of the states updated by the proposed method and satisfying the addressed constrained.

$$\begin{aligned} \mathcal{X}_{\text{state}} \coloneqq \{ & z_k \in \mathbb{R}^{n_x} : \|h_e(z_k)\| \leq \varepsilon,\ h_i(z_k) \leq 0, \\ & z_k \text{ updated from } z_0 \sim \mathcal{N}(\mu_x, \mathbf{\Sigma}_x)\} \end{aligned} \tag{13}$$

For the next step, we take all elements of $\mathcal{X}_{\text{state}}$ and check whether there exist the appropriate input and contact force to propagate the state which also fulfills the state constraints. Considering the input and contact force constraints, we formulate a QP problem by taking two elements v_1 and v_2 from $\mathcal{X}_{\text{state}}$.

$$\begin{aligned} \min_{u, F_c} \quad & F_c^\top \mathbf{Q}_c F_c + u^\top \mathbf{Q}_u u \\ \text{s.t.} \quad & v_2 = f_D(v_1, u, F_c), \\ & h_u(u) \leq 0,\ h_c(v_1, F_c) \leq 0 \end{aligned} \tag{14}$$

where $\mathbf{Q}_c \in \mathbb{S}_{>0}^{n_c}$ and $\mathbf{Q}_u \in \mathbb{S}_{>0}^{n_u}$ are weighting matrices for the cost. If there exists an optimal solution of the problem in (14), we collect all v_1 in (14) into SRS, $\mathcal{X}$. Otherwise, we discard the updated state samples. It is key to draw sufficient state samples on the constrained manifold using the proposed optimization-based method because we break down the end-to-end trajectory optimization problem via the use of POMDP, which will rely on the probabilistic distribution over the sampled set. If the state set is sparse, there is less chance of finding a feasible sequence of output regions.

4.2 Output Space Approximation

In this subsection, we check the feasibility of reaching the desired goal output. To do so, we approximate the output samples with a Gaussian distribution $y^* \sim \mathcal{N}(\mu_{y^*}, \mathbf{\Sigma}_{y^*})$ Hendeby and Gustafsson [2007]. From the obtained SRS $\mathcal{X}$, we compute sample mean and covariance of $\mathcal{X}$, μ_x^* and $\mathbf{\Sigma}_x^*$. Based on the sample mean and covariance, the mean and covariance of the output obtained after neglecting higher order terms are

$$\mu_{y^*} \coloneqq f_y(\mu_x^*) + \text{Vec}\left[\text{tr}(\mathbf{H}_{y,i}(\mu_x^*)\mathbf{\Sigma}_x^*)\right]_{i=1}^{n_y} \tag{15a}$$

$$\begin{aligned} \mathbf{\Sigma}_{y^*} \coloneqq & \mathbf{J}_y(\mu_x^*)\mathbf{\Sigma}_x^*\mathbf{J}_y^\top(\mu_x^*) \\ & + \frac{1}{2}\text{Mat}\left[\text{tr}(\mathbf{\Sigma}_x^*\mathbf{H}_{y,i}(\mu_x^*)\mathbf{\Sigma}_x^*\mathbf{H}_{y,j}(\mu_x^*))\right]_{i,j=1}^{n_y, n_y} \end{aligned} \tag{15b}$$

where $\mathbf{J}_y(\mu)$ and $\mathbf{H}_{y,i}(\mu)$ denote the Jacobian matrix of the output function $f_y(\mu)$ and the 2nd derivative matrix of the output function $f_{y,i}(\mu)$, for the i-th element. In particular,

$$\begin{aligned} \mathbf{J}_y(\mu) &\coloneqq \frac{\partial f_y}{\partial x}(\mu), \\ \mathbf{H}_{y,i}(\mu) &\coloneqq \begin{bmatrix} \frac{\partial^2 f_{y,i}(\mu)}{\partial x_1^2} & \cdots & \frac{\partial^2 f_{y,i}(\mu)}{\partial x_1 \partial x_{n_x}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f_{y,i}(\mu)}{\partial x_{n_x} \partial x_1} & \cdots & \frac{\partial^2 f_{y,i}(\mu)}{\partial x_{n_x}^2} \end{bmatrix} \end{aligned} \tag{16}$$

where $\mathbf{J}_y(\mu) \in \mathbb{R}^{n_y \times n_x}$ and $\mathbf{H}_{y,i}(\mu) \in \mathbb{R}^{n_x \times n_x}$ is a symmetric matrix. We construct a probabilistic ellipsoid in the output space to approximate whether an output sample y^* is feasible. We define a set of outputs that lie inside an ellipsoid $\mathcal{E}_\kappa$ with

$$\mathcal{E}_\kappa \coloneqq \{y \in \mathbb{R}^{n_y} : (y - \mu_{y^*})^\top \mathbf{\Sigma}_{y^*}^{-1}(y - \mu_{y^*}) \leq \kappa\} \tag{17}$$

where κ is a coefficient determined by the cumulative probability of the Chi-square distribution. For instance, $\kappa = 5.991$ for $\Pr(y^* \in \mathcal{E}_\kappa) = 0.95$ and $y^* \in \mathbb{R}^2$. Our method to check if a goal output y^g is interior to $\mathcal{E}_\kappa$ is more efficient than using a Monte Carlo method, because we only need to compute μ_{y^*} and $\mathbf{\Sigma}_{y^*}$ using the mean and covariance matrix of the samples using (15).

5 Optimal Sequence of Subregions via POMDP

After checking that the desired output goal y^g is located at the interior of the ellipsoid $\mathcal{E}_\kappa$ in (17), we solve Subproblem 2 to find an optimal sequence of subregions in the output space. We will find the optimal policy to move from the output subregion containing $y(t_0) = f_y(x(t_0))$ to one where y^g is located.

5.1 POMDP Setup

To start the process solving Subproblem 2, we define output subregions:

$$Y_i \coloneqq \{y \in \mathbb{R}^{n_y} : \|y - y_i^c\|_\infty \leq \varepsilon_y\} \tag{18}$$

where $y_i^c \in \mathbb{R}^{n_y}$ denotes the center of the output subregion Y_i and $i \in [1, m]_\mathbb{N}$ where $\cup_{i\in[1,m]_\mathbb{N}} Y_i \subset \mathbb{R}^{n_y}$. Sides of the subregions are $2\varepsilon_y$. Also, we obtain a set of outputs as follows:

$$\mathcal{Y} \coloneqq \hat{f}_y(\mathcal{X}) = \{y \in \mathbb{R}^{n_y} : y = f_y(x), x \in \mathcal{X}\} \tag{19}$$

where $\hat{f}_y : \mathcal{X} \rightrightarrows \mathcal{Y}$ denotes a set-value mapping. To formulate our problem as a POMDP, we define discrete nodes associated with the previous subregions as follows:

$$s_i = \text{node}(Y_i), \quad i \in [1, m]_\mathbb{N} \tag{20}$$

where $\mathcal{S} \coloneqq \{s_1, \ldots, s_m\}$. Based on these nodes, we transform the problem to a POMDP. We will formulate the probability of observations using the sampled states. Before formulating the detailed POMDP, we introduce several definitions used in this section

Definition 2 (POMDP) Partially Observable Markov Decision Process is defined as a tuple $\mathbf{P} = (\mathcal{S}, \mathcal{A}, \mathcal{O}_b, \mathbf{T}, \mathbf{Z})$:

- $\mathcal{S}$ is a finite set of nodes, $\mathcal{S} \coloneqq \{s_1, \cdots, s_{m_s}\}$
- $\mathcal{A}$ is a finite set of actions, $\mathcal{A} \coloneqq \{a_1, \cdots, a_{m_a}\}$
- $\mathcal{O}_b$ is a finite set of observations, $\mathcal{O}_b \coloneqq \{o_1, \cdots, o_{m_o}\}$
- $\mathbf{T}$ is the transition dynamics $\mathbf{T}(s', s, a)$ defining the transition from $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ after taking an action $a \in \mathcal{A}$.
- $\mathbf{Z}$ is the observation $\mathbf{Z}(s, a, o)$ consisting of the probability of observing $o \in \mathcal{O}_b$ after taking an action $a \in \mathcal{A}$ from node $s \in \mathcal{S}$.

Definition 3 (Markov Policy) A Markov policy $\mathbf{\Pi}$ is defined as a sequence: $\mathbf{\Pi} \coloneqq \{a^{(1)}, \cdots, a^{(n)}\}$. $a^{(j)} \in \mathcal{A}$, where $a^{(j)} : \mathcal{S} \to \mathcal{S}$ is a measurable map from a node to another one, $j \in [1, n]_\mathbb{N}$.

Definition 4 (Principle Singular Vector) Consider a node $s_i \in \mathcal{S}$ associated with an output subregion Y_i and a set $\mathcal{Y} = \hat{f}_y(\mathcal{X})$. Consider $\mathbf{\Sigma}_{\mathcal{Y}_i}$ being the covariance matrix for the set $Y_i \cap \mathcal{Y}$, that is, $\mathbf{\Sigma}_{\mathcal{Y}_i} = \mathbb{E}[(y - \mu_y)(y - \mu_y)^\top]$ and $\mu_y = \mathbb{E}[y]$ where $y \in Y_i \cap \mathcal{Y}$. A Principle Singular Vector (PSV) is defined as

$$\mathcal{V}(s_i) = \text{col}(\mathbf{V}_{\mathcal{Y}_i})_k, \quad \sigma_k(\mathbf{\Sigma}_{\mathcal{Y}_i}) = \overline{\sigma}(\mathbf{\Sigma}_{\mathcal{Y}_i}) \tag{21}$$

where $\mathbf{\Sigma}_{\mathcal{Y}_i} = \mathbf{V}_{\mathcal{Y}_i}^\top \mathbf{\Lambda}_{\mathcal{Y}_i} \mathbf{V}_{\mathcal{Y}_i}$, $\mathbf{\Lambda}_{\mathcal{Y}_i} = \text{diag}(\sigma_1, \ldots, \sigma_{n_y})$, and σ_k denotes the singular value of $\mathbf{\Sigma}_{\mathcal{Y}_i}$.

5.2 POMDP Formulation

The problem concerning this section is on finding a sequence of feasible subregions towards an output goal using POMDP tools and analysis. To do that, we convert the POMDP into a belief MDP. Belief $b[s_i]$ is defined with respect to discrete nodes $s_i \in \mathcal{S}$. Let suppose $b = b[s^{(j)}]$, $b' = b[s^{(j+1)}]$, and $a = a^{(j)}$ where $s^{(j)}$ represents the node for the j-th step of the POMDP. The belief transition function, $\mathbf{\Gamma}(b, a, b')$, is equal to

$$\mathbf{\Gamma}(b, a, b') = \sum_{o\in\mathcal{O}_b} \Pr(b'|b, a, o)\Pr(o|b, a) \tag{22a}$$

$$\Pr(b'|b, a, o) = \begin{cases} 1, \text{ if belief update returns } b' \\ 0, \text{ otherwise} \end{cases} \tag{22b}$$

$$\Pr(o|b, a) = \sum_{s'\in\mathcal{S}} \mathbf{Z}(s', a, o) \sum_{s\in\mathcal{S}} \mathbf{T}(s', s, a)b. \tag{22c}$$

The key challenges of this POMDP are on defining meaningful observations and on finding their conditional probability. Let us consider that $Y = Y^{(j)}$ and $Y' = Y^{(j+1)}$ associated with the nodes $s^{(j)}$ and $s^{(j+1)}$. We propose to define observations as the set of feasible states after taking an action a, i.e.

$$\begin{aligned} \hat{O} \coloneqq \{&v_1 \in \mathbb{R}^{n_x} : v_1 \in Y \cap \mathcal{Y},\ v_2 \in Y' \cap \mathcal{Y}, \\ &v_2 = f_D(v_1, u, F_c),\ h_u(u) \leq 0, \\ &h_c(v_1, F_c) \leq 0,\ u \in \mathbb{R}^{n_u},\ F_c \in \mathbb{R}^{n_c}\} \end{aligned} \tag{23}$$

where Y is the subregion before taking the action a. If $v_1 \in \hat{O}$, it holds that there exists at least one sample connecting $f_y(v_1)$ to another output $f_y(v_2)$ in the subregion $\mathcal{Y}'$ satisfying the constraints as illustrated in Fig 2(a). Otherwise, v_1 is excluded from the observation $\hat{O}$. Considering the above observations, we define the conditional probability as

$$\mathbf{Z}(s', a, o) \coloneqq \Pr(o|s', a) = \text{card}(\hat{O})/\text{card}(Y' \cap \mathcal{Y}). \tag{24}$$

Let us focus on the reward and transition dynamics. As a heuristic, a higher number of feasible samples falling

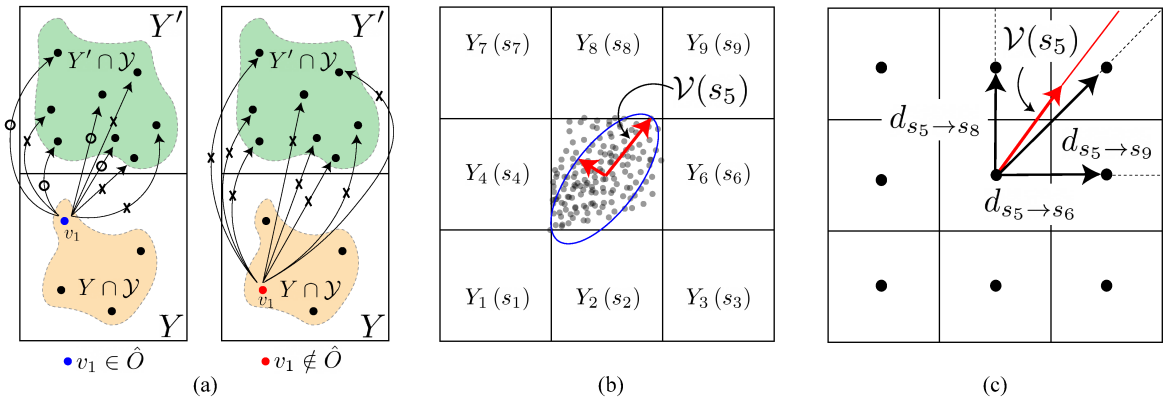

Fig. 2 Basic use of a POMDP process: (a) Observations are samples that can reach neighboring output subregions, (b) Given a set of observations for a node $\mathcal{V}(s_i)$ we compute their principle singular vector, (c) Using this vector we compute the transition dynamics and the action vectors in output space.

into a subregion implies a higher probability of reaching it. Therefore we define the reward

$$\mathfrak{R}(s_i, a) \coloneqq K_r \frac{\text{card}(Y_i \cap \mathcal{Y})}{\text{card}(\mathcal{Y})} + \eta_i \tag{25}$$

where $K_r \in \mathbb{R}_{>0}$ and $\eta_i \in \mathbb{R}$ are the gain and reward offset, respectively. We take η_i to be large when $y^g \in \mathcal{Y}_i$. Also, to avoid unsafe output regions we set η_i to large negative values.

For defining the transition dynamics, let an action $a \in \mathbf{A}$ map state $s \in \mathcal{S}$ to $s' \in \mathcal{S}$. $d_{s \to s'}$ is a vector in our node space associated with $d_{Y \to Y'} \coloneqq (y'^c - y^c)$ where y^c and y'^c are the centers of subregions Y and Y', respectively. We define the transition dynamics as

$$\mathbf{T}(s', s, a) \coloneqq \begin{cases} \mathcal{T}(s', s, a)/\varpi, & \text{if } \varpi \neq 0, \\ 0, & \text{else} \end{cases} \tag{26a}$$

$$\mathcal{T}(s', s, a) \coloneqq \max\{0, \langle \mathcal{V}(s), \mathbf{a} \rangle\} \tag{26b}$$

where $\varpi = \sum_{a \in \mathbf{A}} \mathcal{T}(s', s, a)$ denotes a normalization constant and $\mathbf{a}$ is the action vector of the action a. In Fig. 2(b), we illustrate $\mathcal{V}(s)$ and it is intuitive that the samples are densely scattered along PSV. Fig 2(c) interprets the meaning of the tansition dynamics. In detail, the angle between PSV and the action vector is smaller, in other words $\langle \mathcal{V}(s), \mathbf{a} \rangle$ is bigger, and it is easier to make a transition in the action direction $\mathbf{a}$. As according to Proposition 1, we cannot construct the transition dynamics by the proposed method if first two singular values of $\mathbf{\Sigma}_{\mathcal{Y}_i}$ are similar or the output samples are uniformly distributed. In these cases, the transition dynamics is considered as deterministic.

Proposition 1 *Let $s = s_i \in \mathcal{S}$ and the corresponding subregion be Y_i. If the output samples are uniformly distributed in Y_i, then $\mathbf{T}(s', s, a) = 0$.*

Proof See Appendix A.

We now solve a finite-horizon belief MDP. The optimal policy, denoted by $\mathbf{\Pi}_\star$, is obtained by solving the Bellman equation as follows:

$$\mathfrak{D}_\star(b) = \max_{a \in \mathcal{A}} \Bigg[\sum_{s \in \mathcal{S}} b(s) \mathfrak{R}(s, a) + \beta \sum_{o \in \mathcal{O}_b} \Pr(o|b, a) \mathfrak{D}_\star(\mathbf{\Gamma}(b, a, o)) \Bigg] \tag{27}$$

where β is a discount coefficient. The result of the DP provides an optimal Markov policy which we transform to a sequence of nodes as $\mathbf{S}_\star = \{s_\star^{(1)}, \cdots, s_\star^{(n_\pi)}\}$. The sequence of subregions in output space is converted form the optimal sequence of the nodes, $\mathbf{Y}_\star = \{\text{node}^{-1}(s_\star^{(1)}), \cdots, \text{node}^{-1}(s_\star^{(n_\pi)})\}$. Based on the generated sequence of subregions, we will implement reachability analysis for each connection between sequential two subregions in $\mathbf{Y}_\star$.

6 Reachability Analysis

It is extremely hard to obtain the reachable set in continuous time because our problem incorporates the nonlinear dynamics with intricated state-input constraints. In this section, we define discrete-time reachable sets and propose the way to obtain the reachable sets via optimizations. To overcome the computational complexity of propagation algorithm for the reachable sets, a method propagating the boundary samples is proposed and analyzed in the views of computational complexity.

6.1 Optimization-based Reachability Analysis

To start solving Subproblem 3, we define a discrete-time reachable set at time instance t in terms of the initial

state $x_0 = x(t_0)$. We assume that the given initial state x_0 fulfills all state constraints.

Definition 5 (Discrete-time Reachable Set) Let x_0 be an initial state and $t \in T_d^{\Delta t} = [t_0, t_f]_d^{\Delta t}$ be an arbitrary time interval. We define a reachable set in discrete time domain as:

$$\begin{aligned}\mathcal{R}_x(t, x_0) \coloneqq \{x(t) : \exists u([t_0, t]_d^{\Delta t}), \exists F_c([t_0, t]_d^{\Delta t}),\\ h_e(x(\tau)) = 0,\ h_i(x(\tau)) \leq 0,\\ h_u(u(\tau)) \leq 0,\ h_c(x(\tau), F_c(\tau)) \leq 0,\\ x(\tau + \Delta t) = f_D(x(\tau), u(\tau), F_c(\tau)),\\ x(t_0) = x_0,\ \tau \in [t_0, t]_d^{\Delta t}\}\end{aligned} \tag{28}$$

where $\Delta t > 0$ is the discretization step or sampling period for our discrete model.

We extend the reachable set defined above for the finite discrete time interval $T_d^{\Delta t} = [t_0, t_f]_d^{\Delta t}$ as

$$\mathcal{R}_x(T_d^{\Delta t}, x_0) \coloneqq \bigcup_{t \in T_d^{\Delta t}} \mathcal{R}_x(t, x_0). \tag{29}$$

For any $t_f < \infty$, the reachable set satisfies the following bound $\mathcal{R}_x(T_d^{\Delta t},\ x_0) \subseteq \mathcal{R}_x([t_0, +\infty), x_0)$. By using the set-value mapping $\hat{f}_y$, we can express the reachable sets in the output space such as $\mathcal{R}_y(t, x_0) = \hat{f}_y(\mathcal{R}_x(t, x_0))$ and $\mathcal{R}_y(T_d^{\Delta t}, x_0) = \hat{f}_y(\mathcal{R}_x(T_d^{\Delta t}, x_0))$. We clearly show the following two corollaries related to the defined discrete-time reachable sets.

Corollary 1 *The reachable set over a finite time horizon $T_d^{\Delta t} = [t_0, t_f]_d^{\Delta t}$, $\mathcal{R}_y(T_d^{\Delta t}, x_0)$, is compact.*

Proof See Appendix B.

Corollary 2 *Suppose that $\{x_0\} = \mathcal{R}_x(t_0, x_0)$ and f_y is continuous. Then, a set, $\mathcal{R}_y(T_d^{\Delta t}, x_0)$ where $T_d^{\Delta t} = [t_0, t_f]_d^{\Delta t}$, is connected.*

Proof See Appendix C.

Above two corollaries are useful to check whether the goal output is reachable using the proposed reachable sets. Based on the corollaries, we address a theorem to confirm that the goal output is reachable in terms of the system dynamics, constraints, and a specific time horizon.

Theorem 1 *Suppose that the initial state, x_0, the goal output, y^g, and discrete-time interval, $T_d^{\Delta t} = [t_0, t_f]_d^{\Delta t}$, are given. Let us assume that the set, $\mathcal{R}_y(T_d^{\Delta t}, x_0)$, is compact, connected, and $y^g \in \mathcal{R}_y(T_d^{\Delta t}, x_0)$. Then, at lest one trajectory $\boldsymbol{\Psi} \coloneqq \{\xi(t_0), \ldots, \xi(\tau)\}$ exists such that $f_y(\xi(\tau)) = y^g$ where $\tau \leq t_f$.*

Proof See Appendix D.

6.2 Forward Propagation of Reachable Set

We employ the sampling-base method and a QP to propagate the state in the discrete-time domain. A random input is drawn from a Gaussian distribution $u \sim \mathcal{N}(\mu_u, \boldsymbol{\Sigma}_u)$ at each instant of time with the input set $\mathcal{U}$ defined as the collection of inputs fulfilling input constraint. Let us consider x_0 and $\mathcal{R}_x(T_d^{\Delta t}, x_0)$ where $T_d^{\Delta t} = [t_0, t_f]_d^{\Delta t}$. We define a QP to check for feasible contact forces, i.e.

$$\begin{aligned}\min_{F_c, x_{k+1}} \quad & F_c^\top \mathbf{Q}_c F_c + (x_{k+1} - x_k)^\top \mathbf{Q}_x (x_{k+1} - x_k)\\ \text{s.t.} \quad & x_{k+1} = f_D(x_k, u, F_c),\\ & h_e(x_{k+1}) = 0,\ h_i(x_{k+1}) \leq 0,\\ & h_c(x_k, F_c) \leq 0,\ u \in \mathcal{U}\end{aligned} \tag{30}$$

where $x_k \in \mathcal{R}_x(T_d^{\Delta t}, x_0)$. If there exist the optimal decision variable $F_c^\star$ and $x_{k+1}^\star$, $x_{k+1}^\star$ becomes an element of $\mathcal{R}_x(t_{k+1}, x_0)$. For all sampled inputs $u \in \mathcal{U}$ and the reachable states $x_k \in \mathcal{R}_x(T_d^{\Delta t}, x_0)$, the QP in (30) is repeatedly solved to collect the optimal decision variables $x_{k+1}^\star$ in $\mathcal{R}_x(t_{k+1}, x_0)$:

$$\begin{aligned}\mathcal{R}_x(t_{k+1}, x_0) = \{x_{k+1}^\star : (F_c^\star,\ x_{k+1}^\star) \leftarrow \text{QP (30)},\\ \forall x_k \in \mathcal{R}_x(T_d^{\Delta t}, x_0),\ \forall u \in \mathcal{U}\}\end{aligned} \tag{31}$$

The reachable set $\mathcal{R}_x([t_0, t_{k+1}]_d^{\Delta t}, x_0) = \mathcal{R}_x(T_d^{\Delta t}, x_0) \cup \mathcal{R}_x(t_{k+1}, x_0)$. This is a computationally efficient method to compute the reachable sets because the QP is computationally cheap. However, the number of state samples in the reachable sets exponentially increases in terms of the number iteration. To reduce the growing computation burden, we propose a particular method in the following section.

6.3 Propagation of Boundary States

The basic algorithm for reachability analysis suffers from exponential complexity with respect to the number of time steps. Although the previous POMDP contributes to reducing the time horizon to be checked for reachability analysis, full-state propagation would still result in heavy computational burden. In this section, we propose a method for reducing the computational complexity of the algorithm by only propagating selected states. This approach results in more conservative reachable sets.

To implement the forward propagation of boundary state samples, we define a set by collecting the boundary samples of an reachable set $\mathcal{R}_x(t, x_0)$ as follows:

$$\begin{aligned}\mathcal{B}_x(t, x_0) \coloneqq \{x \in \mathbb{R}^{n_x} : x \in \mathcal{R}_x(t, x_0),\\ f_y(x) \in \text{gbd}(\mathcal{R}_y(t, x_0))\}.\end{aligned} \tag{32}$$

Algorithm 1: Computing Reachable Sets by Propagation of Boundary States

Data: q_0, $\dot{q}_0$, $\mathbf{Q}_c$, $\mathbf{Q}_x$, $T_d^{\Delta t}$, y^g
Result: $\mathcal{R}_x(T_d^{\Delta t}, x_0)$
$\mathbf{M}_0$, $b_0 \leftarrow$ Update dynamics properties w.r.t. q_0, $\dot{q}_0$;
$\mathbf{J}_c \leftarrow$ Update contact Jacobian w.r.t. q_0 ;
Check Contact Kinematics in (4) w.r.t. q_0, $\dot{q}_0$;
if *Feasible* **then**
  $\mathcal{R}_x([t_0, t_0]_d^{\Delta t}, x_0) = \mathcal{R}_x(t_0, x_0) = \{x_0\}$;
  for $k \leftarrow 0$ **to** N_t **do**
    $T_{k,d}^{\Delta t} = [t_0, t_k]_d^{\Delta t}$;
    $\mathcal{U} \leftarrow$ Generate N_u input samples;
    $\mathcal{B}_x(T_{k,d}^{\Delta t}, x_0) \leftarrow$ (32) by $\mathcal{R}_x(T_{k,d}^{\Delta t}, x_0)$;
    $N_{\mathcal{R}_x} \leftarrow$ Get card$(\mathcal{B}_x(T_{k,d}^{\Delta t}, x_0)$;
    for $j \leftarrow 1$ **to** $N_{\mathcal{R}_x}$ **do**
      $x_k \leftarrow j$-th sample in $\mathcal{R}_x(T_{k,d}^{\Delta t}, x_0)$;
      for $i \leftarrow 1$ **to** N_u **do**
        $u \leftarrow i$-th sample in $\mathcal{U}$;
        solve (30) with x_k and u ;
        if $\exists(F_c^\star, x_{k+1}^\star)$ **then**
          add $x_{k+1}^\star$ to $\mathcal{R}_x^{\mathcal{B}}(t_{k+1}, x_0)$;
        end
      end
    end
    $\mathcal{R}_x([t_0, t_{k+1}]_d^{\Delta t}, x_0) \leftarrow$
     $\mathcal{R}_x(T_{k,d}^{\Delta t}, x_0) \cup \mathcal{R}_x^{\mathcal{B}}(t_{k+1}, x_0)$;
  end
end
else
  Terminate due to the infeasible initial state
end
;

This definition can be extended to the reachable set over a finite time horizon, $\mathcal{B}_x(T_d^{\Delta t}, x_0)$, in the same manner. We solve the QP (30) for only the state samples in $\mathcal{B}_x(T_d^{\Delta t}, x_0)$ to obtain $\mathcal{R}_x^{\mathcal{B}}(t_{k+1}, x_0)$ and iterate this process to expand the reachable set forward. In this way, the computational complexity becomes linear with respect to the number of boundary samples, card$(\mathcal{B}_x(T_d^{\Delta t}, x_0))$. The detailed process of our algorithm is described in Algorithm 1. In order to replace full-state propagation with boundary-state propagation, we show that the set of reachable outputs is compact and connected. First, the set $\mathcal{R}_y^{\mathcal{B}}(T_d^{\Delta t}, x_0)$ is compact, because we are able to obtain the hull of the set as shown in Corollary 1. Next, we prove the reachable set $\mathcal{R}_y^{\mathcal{B}}(T_d^{\Delta t}, x_0)$ is connected.

Corollary 3 *Suppose that $x_0 \in \mathcal{R}_x(t_0, x_0)$ and f_y is continuous. Then, $\mathcal{R}_y^{\mathcal{B}}(T_d^{\Delta t}, x_0)$ is connected.*

Proof The proof is similar to that of Corollary 2 and therefore is omitted. See Appendix C.

In linear systems, it is straightforward to recursively propagate boundary states by, for instance, using Proposition 6.5 in Blanchini and Miani [2008] and to over-approximate the reachable set using Zonotopes Girard et al. [2006], Hänsch et al. [2013]. In addition, propagating boundary states is valid for obtaining conservative reachable sets for manipulators Lee et al. [2019] and legged robots Lee et al. [2020] if their kinematic workspaces are connected, the reason being that the reachable set is a subset of the kinematic workspace. The above proposition can be extended for the class of nonlinear robotic systems with connected workspaces.

6.4 Computational Complexity Analysis

We analyze the computational complexity to compare the efficiency of the propagation of full states and that of boundary states. There exists many algorithms to obtain the concave hull from the set of data Galton and Duckham [2006], Duckham et al. [2008], Moreira and Santos [2007]. They have $\mathcal{O}(n^3)$ or $\mathcal{O}(n \log n)$ time complexity with n data in 2-D space. General QPs are non-deterministic polynomial-time hard, which means the algorithms are more complex than the polynomial time complexity to be solved. In the case that the QP is convex, it is widely known that the time complexity of the QP is $\mathcal{O}(m^3)$ where m is the number of decision variables.

Based on the aforementioned discussion, we can compare the computational complexity of two cases: propagation of full states and propagation of boundary states. Let us consider N_t steps over the time interval $[t_0, t_k]_d^{\Delta t}$ where $\Delta t = (t_k - t_0)/N_t$, and N_u is the number of input samples. For each propagation method, the computational complexity can be represented as

$$\begin{aligned} \mathbf{C}_f \sim & \mathcal{O}(\sum_{i=1}^{N_t} N_u^i (n_c + n_x)^3) \approx \mathcal{O}\left(N_u^{N_t}(n_c + n_x)^3\right) \\ \mathbf{C}_b \sim & \mathcal{O}(\sum_{i=1}^{N_t} N_b (n_c + n_x)^3 + (iN_b)^3) \\ & \approx \mathcal{O}(N_t N_b (n_c + n_x)^3 + N_t^4 N_b^3) \end{aligned} \tag{33}$$

where $\mathbf{C}_b$, and N_b denote the complexity of full state propagation, that of boundary state propagation, and the number of boundary samples. Normally, a set of boundary samples contains much smaller samples than a set of entire states, that is, $N_b \ll N_u$. The effect of the boundary sampling on computational complexity becomes significantly advantageous in terms of the number of time steps. We will show the comparison of the computational complexity using an example in the simulation section.

Algorithm 2: Sequential Optimal Control

Data: q_0, $\dot{q}_0$, $\mathbf{Q}_c$, $\mathbf{Q}_x$, $\mathbf{Q}_y$, y^g, $\mathbf{Y}_\star$
Result: $\mathbf{\Psi}$, $\mathbf{U}$
if y^g *is reachable* **then**
  $\mathbf{\Psi} = \emptyset$, $\mathbf{U} = \emptyset$;
  $x_0 \leftarrow [q_0^\top, \dot{q}_0^\top]^\top$;
  for $k \leftarrow 1$ **to** $n_\pi - 1$ **do**
    $y^f \leftarrow y_{\star c}^{(k)}$;
    $(\mathbf{\Psi}_\star^{(k)}, \mathbf{U}_\star^{(k)}) \leftarrow$ solve NLP problem in (36) ;
    $\mathbf{\Psi} \leftarrow \text{vertcat}(\mathbf{\Psi}, \mathbf{\Psi}_\star^{(k)})$;
    $\mathbf{U} \leftarrow \text{vertcat}(\mathbf{U}, \mathbf{U}_\star^{(k)})$;
    $x_0 \leftarrow \xi_f$ in $\mathbf{\Psi}_\star^{(k)}$;
  end
end
else
  Terminate due to the infeasible goal output
end
;

6.5 Sequential Reachability Analysis

Given an optimal sequence of subregions in the output space, $\mathbf{Y}_\star = \{Y_\star^{(1)}, \cdots, Y_\star^{(n_\pi)}\}$, we check it is possible to successively move from one subregion to the next one via reachability analysis as described in Subproblem 3.

7 Nonlinear Constrained Optimal Control

In this work, we describe the use of sequential optimal control for nonlinear programs without constraint softening. Instead of considering end-to-end trajectory generation, we focus on finding a trajectory connecting two subregions obtained by the POMDP process described earlier. By iterating this process for connecting subregions, the optimal control process is able to attain the desired output with reduced computational cost.

7.1 Nonlinear Programming

In order to formulate the optimal control problem solved by NLP, a performance measure is defined in the discrete time and state space, that is the sum of running cost, ℓ, and, final cost, ℓ_f:

$$\mathcal{J}(\mathbf{U}, \mathbf{F}_c, N) \coloneqq \sum_{k=0}^{N-1} \ell(x_k, u_k, F_{c,k}) + \ell_f(x_f, F_{c,f}) \quad (34)$$

where $\mathbf{U} \coloneqq \{u_0, \ldots, u_{N-1}\}$ and $\mathbf{F}_c \coloneqq \{F_{c,0}, \cdots, F_{c,N}\}$. The running and final costs are formulated by using quadratic terms as follows:

$$\begin{aligned} \ell_f(x_f, F_{c,f}) =& \mathbf{e}_y^f(x_f)^\top \mathbf{Q}_y \mathbf{e}_y^f(x_f) + F_{c,f}^\top \mathbf{Q}_c F_{c,f}, \\ \ell(x_k, u_k, F_{c,k}) =& \mathbf{e}_y^f(x_k)^\top \mathbf{Q}_y \mathbf{e}_y^f(x_k) + F_{c,k}^\top \mathbf{Q}_c F_{c,k} \\ &+ \mathbf{e}_x^0(x_k)^\top \mathbf{Q}_x \mathbf{e}_x^0(x_k) \end{aligned} \quad (35)$$

where $\mathbf{e}_y^f(x) = y^f - f_y(x)$ and $\mathbf{e}_x^0(x) = x_0 - x$. Additionally, $\mathbf{Q}_x \in \mathbb{S}_{>0}^{n_x}$, $\mathbf{Q}_c \in \mathbb{S}_{>0}^{n_c}$, and $\mathbf{Q}_y \in \mathbb{S}_{>0}^{n_y}$ denote the weighting matrices for the state, the contact force, and the output, respectively. $\xi(t) \in \mathbb{R}^{n_x}$ and $y_d \in \mathbb{R}^{n_y}$ denote the trajectory of the state and the desired goal of the output of the NLP problem, respectively. The NLP problem is defined to obtain the end-to-end trajectory as follows:

$$\begin{aligned} \min_{\mathbf{\Psi}, \mathbf{U}} \quad & \mathcal{J}(\mathbf{U}, \mathbf{F}_c, N) \\ \text{s.t.} \quad & \xi_{k+1} = f_D(\xi_k, u_k, F_{c,k}), \\ & h_e(\xi_k) = 0,\ h_i(\xi_k) \leq 0, \\ & h_u(u_k) \leq 0,\ h_c(\xi_k, F_{c,k}) \leq 0. \end{aligned} \quad (36)$$

In order to reduce the computational time, we sequentially solve the optimal control problem by referring the results of POMDP problem. Given $\mathbf{Y}_\star$, we solve $n_\pi - 1$ optimal control problems to reach the goal output. Algorithm 2 shows the way to sequentially obtain the optimal trajectory and the corresponding input torque.

8 Numerical Simulations

In this section, we validate the proposed approach by demonstrating numerical simulations using two robotic systems which are called *Draco* and *Valkyrie*. *Draco* is developed as a test platform for efficient and dynamic locomotion using liquid-cooled series elastic actuators Kim et al. [2018] and *Valkyrie* is a biped full-body humanoid robot operated for executing locomanipulation missions by NASA Radford et al. [2015]. For the basic software setup, the dynamic simulation is implemented by DART Lee et al. [2018]. We utilize two optimizers: Goldfarb for QP and IPOPT implementing a primal-dual interior point method Wächter and Biegler [2006]. In addition, we extract analytic expressions of both kinematic and dynamic properties to deploy them for nonlinear optimization process by using Mathematica[1], FROST Hereid and Ames [2017], and MATLAB[2]. The simulation is executed on a laptop with a Core i7-8650U CPU and 16.0 GB RAM.

8.1 One-sided Leg *Draco*

A simulation model of *Draco* consists of three virtual joints for its floating base (q_1, q_2, q_3) $\in SE(2)$, i.e., q_1, q_2, and q_3 refer to virtual joints for the floating base positions and pitch orientation, and three actuated

[1] https://www.wolfram.com/mathematica/ (ver.12)
[2] https://www.mathworks.com/ (R2019b)

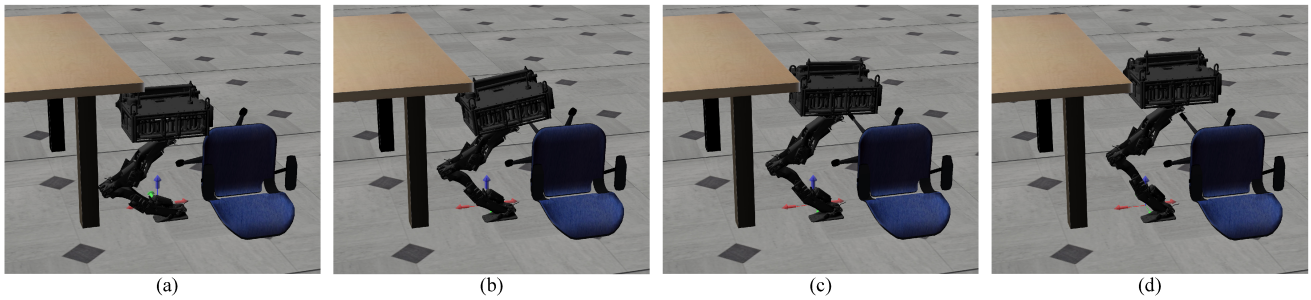

Fig. 3 Snapshots of a Numerical Simulation in the Presence of Two obstacles: a table and a chair: (a) Initial configuration ($t = 0$ s), (b) Avoiding the obstacles ($t = 3$ s), (c) Moving toward the goal position ($t = 4$ s), (d) Final configuration ($t = 5.5$ s).

joints, i.e. hip, knee and ankle joints (q_4, q_5, q_6). In this simulation, we assume that the floating base position in the y direction, and its roll and yaw orientations are fixed. The joint position constraints for three actuated joints are defined as follows:

$$q_{LB} = [-1.2,\ 0.5,\ -1.5]\,, \quad q_{UB} = [-0.2,\ 2.6,\ -0.5]$$

and each joint has ± 1000 rad/s velocity limits. In addition, the position, orientation, and velocity of the foot should satisfy the kinematic constraints for the rigid surface contact. We consider 2-dimensional motion with a surface contact having rectangular support polygon on the foot so that the friction cone constraints can be characterized as

$$|f_x| \leq k_\mu f_z, \quad f_z > 0, \quad |\tau_y| \leq d_x f_z \tag{37}$$

where d_x denotes the distance between the center of the polygon and the vertex in the local frame of the foot and k_μ represents the friction coefficient. $F_c := [f_x^\top, f_z^\top, \tau_y^\top]^\top$ is the contact wrench, which is a resultant contact force at the center of the support polygon. Based on the inequality constraints of (37) and the coordinate transformation from local frame on the foot to the global frame, we can represent the friction cone constraints in the form $\mathbf{W}_{local}\mathbf{R}_c(q)F_c = \mathbf{W}_c(q)F_c \leq 0$ where $\mathbf{W}_{local}$ is a coefficient matrix derived from (37) and $\mathbf{R}_c(q)$ is a rotational matrix from global to foot frames. In $\mathbf{W}_{local}$, we set the friction coefficient $k_\mu = 0.4$. Considering all constraints, we will control the robot's motion while maintaining the contact.

To start the reachability analysis, we generate 10^6 state samples and gather the states fulfilling the constraints. The initial configuration of the robot is $q_{init} = [0.352,\ 0.348,\ 0.0,\ -0.95,\ 2.2,\ -1.25]^\top$. Using the initial configuration, we set the mean and covariance of the Gaussian distribution for sampling: $\mu_x = [q_{init}^\top,\ \mathbf{0}_{1\times 6}]^\top$ and $\mathbf{\Sigma}_x = \mathrm{diag}([\boldsymbol{\pi}^\top, 2\boldsymbol{\pi}^\top]^\top)$ where $\boldsymbol{\pi} = [\pi,\ \pi,\ \pi,\ \pi,\ \pi,\ \pi]^\top$. We define a threshold for numerical convergence of the optimization with value 1.0×10^{-7} and the maximum number of iterations is 1.0×10^6. After implementing the sampling-based approach described in Section 4, we obtain 3.47×10^5 states among 10^6 state samples. For formulating the POMDP problem in 2D space, we set 40 nodes defined by $\mathcal{S} = \{s_i\}$ where $i \in [1, 40]_\mathbb{N}$ and 8 actions $\mathcal{A} = \{a_j\}$ where $j \in [1, 8]_\mathbb{N}$, and each action consists of moving up, down, right, left, up-right, up-left, down-right, and down-left in the grid world, respectively. We consider two static obstacles for the robot to avoid in the output space. The objective of our numerical simulation is to obtain an optimized trajectory to reach the goal output while avoiding the obstacles and fulfilling all constraints. In addition, we generate 1.0×10^5 input samples for propagating the states.

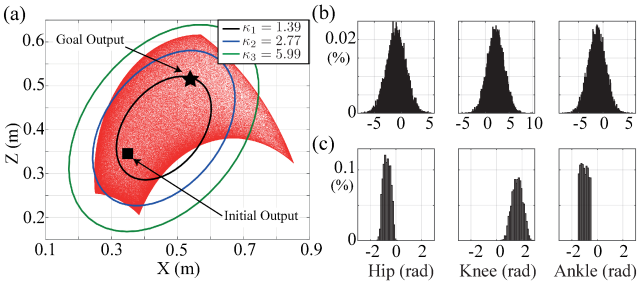

Fig. 4 Output and State Samples: (a) Output samples computed by the feasible state set method and approximated by ellipsoids, (b) State sample distributions before implementing the proposed optimization process, (c) State sample distributions in SRS $\mathcal{X}$. Each histogram is normalized to show its distribution.

First, the results of our sample-based optimization is shown in Fig. 4(a). The set with red dots contains the outputs associated with the states fulfilling the constraints given by the optimization process described in Section 4. As shown in Fig. 4(a), both the initial [0.384, 0.352] and goal [0.51, 0.52] outputs are located at the interior of the feasible set. Then, we solve the POMDP problem to find a sequence of nodes, which result in 12 of them, avoiding the obstacles as shown in Fig. 4(b). After obtaining the sequence of nodes, we

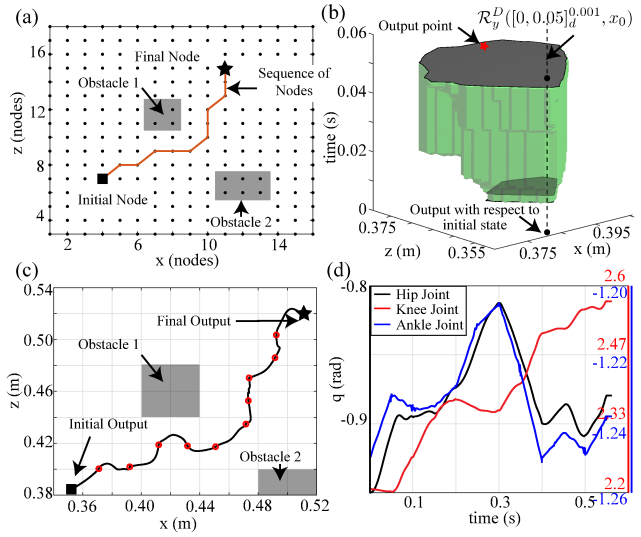

Fig. 5 Simulation results of subproblems: (a) Optimal sequence of the defined POMDP, (b) Propagated reachable volume over the time horizon, (c) Output trajectory by solving optimal control problem, (d) Joint position trajectory by solving optimal control problem.

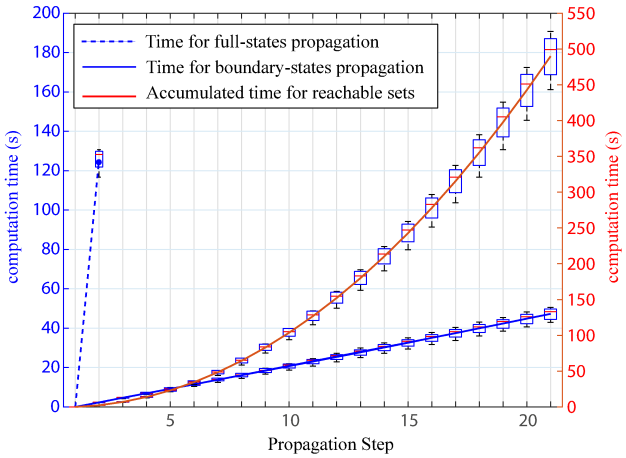

Fig. 6 Computational time for obtaining reachable sets with discretization step $\Delta t = 0.01$ and 10 simulations.

Algorithms	CPU Time (s)	Iter.	Total
Obtaining SRS $\mathcal{X}$	14.133	1	839.024
POMDP	30.218	1	
Reachability	21.091 (average)	11	
Optimal Control	51.152 (average)	11	
Single NLP	7867.151	1	7887.154

Table 1 Computation time for executing two algorithms. Our method reduces the computation time by 10× compared to the baseline (10.66 % of the computation time taken by solving a single large NLP for the entire horizon.)

obtain the reachable sets as shown in Fig. 4(c). The reachable set $\mathcal{R}_y^D([0,0.05]_d^{0.001}, x_0)$ contains the desired output associated with the first node. Based on this result, we solve the NLP (36) to find a trajectory to reach the desired output from the initial configuration.

Fig. 4(d), (e), and (f) show the results of trajectory optimization to reach the goal position with respect to a given initial state. The optimization result includes both joint position and velocity trajectories fulfilling kinodynamic constraints, e.g. joint position, velocity, and torque limits and contact kinematic and force constraints. As shown in Fig 4(d), the generated trajectories pass through the subregions in an optimized sequence obtained by solving the POMPD problem and reaching the final output goal position.Furthermore, the optimization results for the actuated joints in the phase space have stabilizable end points in Fig. 4(f). Fig. 3 shows sequential snapshots of the numerical simulation. In the simulations environment, a pair of table and chair is modeled around the robot and we consider them as obstacles in the output space. As presented in Fig. 3 (d), the robot reaches the goal position avoiding the collision to the obstacles.

The computational complexity is analyzed by measuring the execution time of the algorithm for computing the reachable sets. We repeat 10 simulations to measure the computation time for both the full-state and boundary-state propagation methods and display the results in Fig. 6. The algorithm cannot compute the reachable sets via the full-state propagation method for more than two steps due to the lack of memory capacity, as shown in the blue dotted line in Fig 6. As we predicted in the complexity analysis of Section 6.4, the boundary-states propagation method significantly reduces the computational time for computing the reachable sets. Table 1 shows the computation time for our algorithm compared with the computation time for solving a single large nonlinear optimization problem to reach the goal output. Our method takes 839.024 s to solve the trajectory, which accounts for 10.66 % of the computation time taken by the single large NLP, e.g., 7887.154 s. This result demonstrates that our method has a significant advantage for improving computational efficiency.

8.2 Humanoid Robot *Valkyrie*

We also test the proposed approach using full-body humanoid robot, Valkyrie, having 34 degrees of freedom. The initial configuration and the state and input constraints are specified in Appendix E. The sampling conditions and the threshold for the optimizations are the same as those for the *Draco* simulation in Section 8.1. In this simulation, the state is a 69-dimensional vector consisting of the joint position, velocity, and running

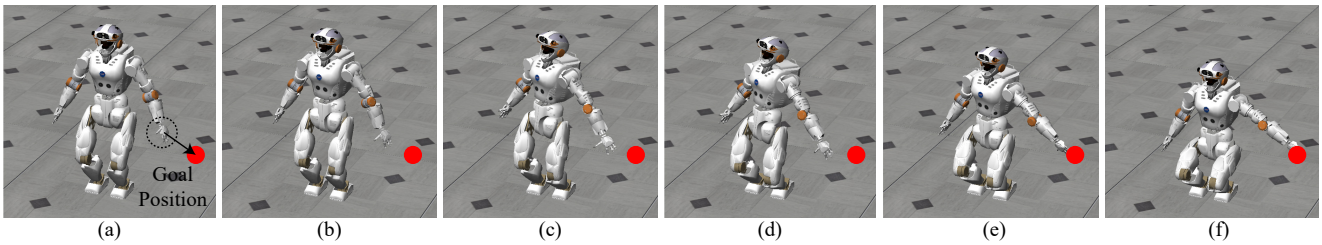

Fig. 7 Snapshots for Valkyrie simulation: red dots indicate the final goal position in Cartesian space. (a) Initial configuration at $t = 0$ s, (b) configuration at $t = 0.1$ s, (c) configuration at $t = 0.2$ s, (d) configuration at $t = 0.3$ s, (e) configuration at $t = 0.4$ s, (f) final configuration at $t = 0.5$ s and the left hand reaches to the desired goal position.

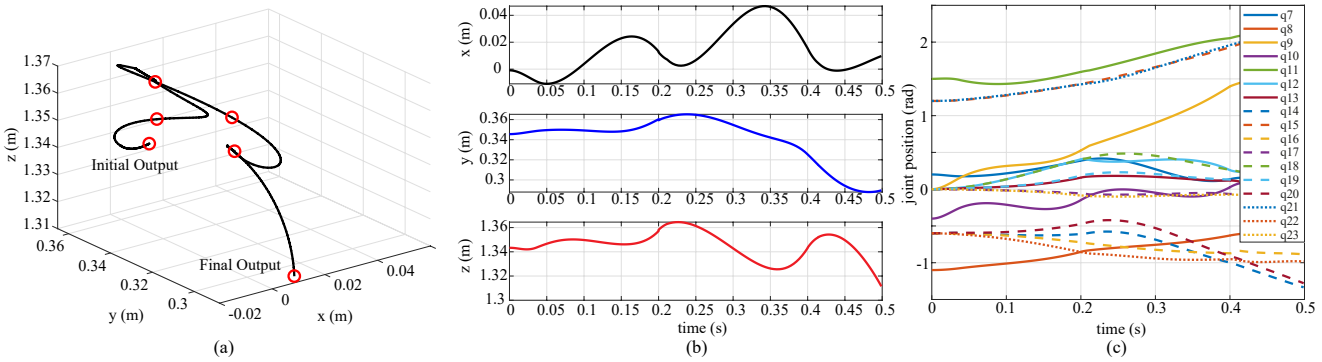

Fig. 8 Simulation Results using Valkyrie: (a) the hand position in Cartesian space with red circles which mean way-points produced by POMDP process, (b) the hand position in the time domain, (c) the joint configuration in the time domain.

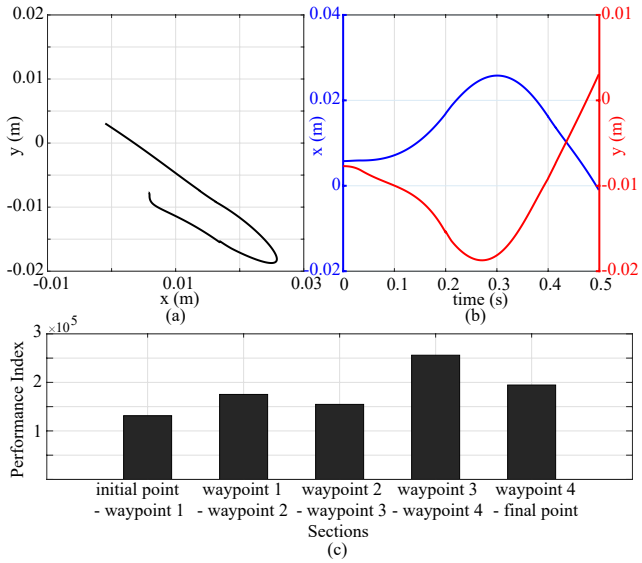

Fig. 9 CoM behavior while controlling the hand position and Performance indices: (a) CoM in the x-y plane, (b) CoM in the time domain (c) performance index values for each section in the output space

cost so that it takes significant time to solve the optimal control problem using NLP. Therefore, we reduce the dimension of the states to 47 by making one of the arms and the neck completely rigid. To start the nonlinear dynamic optimization process, we compute an initial set of initial inputs by using a simple whole-body controller as follows:

$$u_{init} = (\mathbf{M}^{-1}\mathbf{N}_c^\top \mathbf{S}^\top (\mathbf{S}\mathbf{N}_c\mathbf{M}^{-1}\mathbf{S}^\top)^\dagger)^\top b(0, q_{init}) \qquad (38)$$

where $\mathbf{N}_c = \mathbf{I} - \mathbf{M}^{-1}\mathbf{J}_c^\top(\mathbf{J}_c\mathbf{M}^{-1}\mathbf{J}_c^\top)^\dagger\mathbf{J}_c$, $\mathbf{J}_c$ refers the contact Jacobian, and $b(.,.)$ stands for the vector of gravitational, centrifugal and coriolis forces. The goal of this simulation is to move the robot's left hand toward a goal position defined in the operational space while fulfilling contacts on the feet. In this numerical simulation, we consider a 3 dimensional output space (Cartesian coordinates of the left hand) and two surface contacts on the feet. The foot size is 0.135×0.08 and we set the friction coefficient to the value 0.7 in the simulation environment.

The initial and goal positions of the left hand are $[-0.0009, 0.3455, 1.3433]$ and $[0.0134, 0.2849, 1.3085]$, respectively. As shown in Fig. 7, the robot maintains its contacts correctly while changing its whole body configuration to reach the goal output. The proposed process based on POMDP produces 5 waypoints, which are specified by the red circles shown in Fig. 8 (a). The joint position trajectories and the corresponding hand trajectories are generated while satisfying all constraints via the reachability analysis, as shown in Fig. 8 (b) and (c). The robot's Center of Mass (CoM) is one of the interesting results in this simulation as shown in Fig.

9 (a) and (b). Even-though we do not directly control the CoM dynamics, they do not fluctuate much and remains inside the support polygon. The reason is that we fulfill friction constraints and while minimizing kinetic energy: we formulate the quadratic running cost using the state error so that the optimization prevents the robot from moving fast and jerky. We also show in Fig. 9 (c) the performance index values to reach each waypoint from the previous one and the resulting trajectories corresponding to locally optimal solutions.

9 Conclusion

This paper proposes a method to generate trajectories for complex robotic systems subject to contact constraints. We formulate the problem as an optimal control problem, which we subsequently convert to an NLP. Our approach focuses on efficiently solving the NLP problem so that we can scale the method to many types of complex robotic systems. We devise a new approach to obtain discrete-time reachable sets for trajectory generation and solve the nonlinear optimization problem. Although the computational cost is significantly reduced, it is still challenging to employ this approach to real-time control. We demonstrate two numerical simulations using *Draco* and *Valkyrie*. These simulations show that the proposed method is effective in real robotics applications and scalable for highly articulated robots such as full-body humanoid robots. Soon, we will investigate ways to combine this approach with feedback controllers and extend the proposed method for hybrid dynamical systems (e.g., walking or jumping robots and dual-arm manipulation). We will also study the relationship between the optimal conditions of POMDP and the sampling distribution for more efficient motion planning.

Acknowledgements The authors would like to thank the members of the Human Centered Robotics Laboratory at The University of Texas at Austin for their great help and support. This work was supported by an NSF Grant# 1724360 and partially supported by an ONR Grant# N000141512507. The second author acknowledges partial support by NSF under Grant #1924790.

Appendices

A Proof of Proposition 1

Proof Since the samples are uniformly distributed, it is possible to select any unit vector in $\mathbb{R}^{n_y}$ as the PSV of s_i, that is, $\mathbf{R}_a\mathcal{V}(s_i)$ where $\mathbf{R}_a \in \mathrm{SO}(3)$ is a rotation matrix. If we select the rotation matrix $\mathbf{R}_a$ such that $d^\perp = \mathbf{R}_a\mathcal{V}(s_i)$, which is orthogonal to $\mathcal{V}(s_i)$, it follows that $\mathcal{T}(s', s, a) = \langle \mathcal{V}(s_i), d\rangle = \langle d^\perp, d\rangle = 0$ for all $a \in \mathcal{A}$.

B Proof of Corollary 1

Proof Let us consider a general hull of the set $\mathcal{R}_y(T_d^{\Delta t}, x_0)$, that is ghull$(\mathcal{R}_y^D(T_d^{\Delta t}, x_0))$, being compact. By the Heine−Borel theorem, all closed subsets of a compact set are also compact. Since $\mathcal{R}_y(T_d^{\Delta t}, x_0) \subset$ ghull$(\mathcal{R}_y(T_d^{\Delta t}, x_0))$, the reachable set $\mathcal{R}_y(T_d^{\Delta t}, x_0)$ is compact.

C Proof of Corollary 2

Proof Consider three sets: $\mathcal{F}_1 = \mathcal{R}_x(T_d^{\Delta t}, x_0)$, $\mathcal{F}_2 = \mathcal{R}_x(t_k, x_0)$, and $\mathcal{F}_3 = \mathcal{F}_2 \cup \mathcal{F}_1'$, where $\mathcal{F}_1'$ is the collection of states $x \in \mathcal{F}_1$ producing the next feasible state by Definition 5 with respect to $x \in \mathcal{F}_1$. It is true that $\mathcal{R}_x(T_d^{\Delta t}, x_0) = \mathcal{F}_1 \cup \mathcal{F}_2 = \mathcal{F}_2 \cup \mathcal{F}_3$. Let us consider arbitrary two sets $\mathcal{H}_1$ and $\mathcal{H}_2$ satisfying $\mathcal{R}_x(T_d^{\Delta t}, x_0) = \mathcal{H}_1 \cup \mathcal{H}_2$ with $\mathcal{H}_1 \cap \mathcal{H}_2 = \emptyset$. Let $x \in F_1'$ and suppose $x \in \mathcal{H}_1$. Then, $\mathcal{H}_1 \cap \mathcal{F}_1 \neq \emptyset$ and $\mathcal{H}_1 \cap \mathcal{F}_3 \neq \emptyset$. This implies that $\mathcal{F}_1 \subseteq \mathcal{H}_1$ and $\mathcal{F}_3 \subseteq \mathcal{H}_1$, hence, $\mathcal{H}_2 = \emptyset$. This proves that $\mathcal{R}_x(T_d^{\Delta t}, x_0)$ is connected. Since the mapping f_y is continuous, we also conclude that the set $\mathcal{R}_y(T_d^{\Delta t}, x_0)$ is connected.

D Proof of Theorem 1

Proof Since $\mathcal{R}_y(T_d^{\Delta t}, x_0)$ is compact and $\hat{f}_y$ is continuous, $\mathcal{R}_x(T_d^{\Delta t}, x_0)$ is closed and $\hat{f}_y^{-1}$ is also continuous. Then, $\mathcal{R}_x(T_d^{\Delta t}, x_0)$ is connected because $\mathcal{R}_y(T_d^{\Delta t}, x_0)$ is connected and $\hat{f}_y^{-1}$ is continuous. Therefore, there exists at least one trajectory connecting x_0 to $x(\tau)$ satisfying $f_y(x(\tau)) = y^g$ in $\mathcal{R}_x(T_d^{\Delta t}, x_0)$.

E Specifications of Valkyrie

We consider the following joint position/velocity/torque constraints. Excluding the virtual joints for the floating base, the actuated joints ($\mathbb{R}^{28}$) are specified such as

q_{init} =[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.2, 1.1, 0.0, 0.4, 1.5,
0.0, 0.0, −0.6, 1.2, 0.6, 0.0,
0.0, 0.0, −0.6, 1.2, 0.6, 0.0],

q_{UB} =[1.181, 0.666, 0.255, 1.162, 1.047, 0.0, 2.0, 1.519,
2.18, 2.174, 3.14, 2.0, 1.266, 2.18, 0.12, 3.14,
0.4141, 0.467, 1.619, 2.057, 0.875, 0.348, 1.1,
0.5515, 1.619, 2.057, 0.875, 0.348],

q_{LB} =[0.0, −1.047, −0.872, −2.85, −1.266, −3.1, −0.12,
− 2.019, −2.85, −1.519, −3.1, −2.174, −2.019, −1.1,
− 0.5515, −2.42, −0.083, −0.8644, −0.349, −0.4141,
− 0.467, −2.42, −0.083, −0.8644, −0.349],

$\dot{q}_B$ =[5.29, 9.0, 9.0, 5.0, 5.0, 5.0, 5.89, 5.89, 11.5, 11.5,
5.0, 5.89, 5.89, 11.5, 11.5, 5.0, 5.89, 7.0, 6.11, 6.11,
11.0, 11.0, 5.89, 7.0, 6.11, 6.11, 11.0, 11.0],

u_B =[150.0, 150.0, 150.0, 26.0, 26.0, 26.0, 190.0, 190.0,
65.0, 65.0, 14.0, 190.0, 190.0, 65.0, 65.0, 14.0, 190.0,
350.0, 350.0, 350.0, 205.0, 205.0, 190.0,
350.0, 350.0, 350.0, 205.0, 205.0];

where $\dot{q}_{UB} = +\dot{q}_B$, $\dot{q}_{LB} = -\dot{q}_B$, $u_{UB} = +u_B$, and $u_{LB} = -u_B$, respectively.

References

M. Arcak and J. Maidens. Simulation-based reachability analysis for nonlinear systems using componentwise contraction properties. *arXiv preprint arXiv:1709.06661*, 2017.

E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 20–31. Springer, 2000.

F. Blanchini and S. Miani. *Set-theoretic methods in control*. Springer, 2008.

R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard. Differential dynamic programming for multi-phase rigid contact dynamics. In *Proceeding of the IEEE/RSJ International Conference on Humanoid Robots*, pages 1–9. IEEE, 2018.

F. Burget and M. Bennewitz. Stance selection for humanoid grasping tasks by inverse reachability maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5669–5674. IEEE, 2015.

J. Carius, R. Ranftl, V. Koltun, and M. Hutter. Trajectory optimization with implicit hard contacts. *IEEE Robotics and Automation Letters*, 3(4):3316–3323, 2018.

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 *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5107–5112, 2015.

J. Carpentier and N. Mansard. Multicontact locomotion of legged robots. *IEEE Transactions on Robotics*, 34(6):1441–1460, 2018.

M. Duckham, L. Kulik, M. Worboys, and A. Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern recognition*, 41(10):3224–3236, 2008.

P. Fernbach, S. Tonneau, O. Stasse, J. Carpentier, and M. Taïx. C-croc: Continuous and convex resolution of centroidal dynamic trajectories for legged robots in multicontact scenarios. *IEEE Transactions on Robotics*, 36(3):676–691, 2020.

A. Galton and M. Duckham. What is the region occupied by a set of points? In *International Conference on Geographic Information Science*, pages 81–98. Springer, 2006.

P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

A. Girard. Reachability of uncertain linear systems using zonotopes. In *International Workshop on Hybrid Systems: Computation and Control*, pages 291–305. Springer, 2005.

A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *International Workshop on Hybrid Systems: Computation and Control*, pages 257–271. Springer, 2006.

Y. Guan, K. Yokoi, and X. Zhang. Numerical methods for reachable space generation of humanoid robots. *International Journal of Robotics Research*, 27(8): 935–950, 2008.

A. Hamadeh and J. Goncalves. Reachability analysis of continuous-time piecewise affine systems. *Automatica*, 44(12):3189–3194, 2008.

P. Hänsch, H. Diab, I. B. Makhlouf, and S. Kowalewski. Reachability analysis of linear systems with stepwise constant inputs. *Electronic Notes in Theoretical Computer Science*, 297:61–74, 2013.

G. Hendeby and F. Gustafsson. On nonlinear transformations of gaussian distributions. *Technical Report from Automatic Control at Link? pings Universitet*, 2007.

A. Hereid and A. D. Ames. Frost fast robot optimization and simulation toolkit. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 719–726. IEEE, 2017.

A. Herzog, N. Rotella, S. Schaal, and L. Righetti. Trajectory generation for multi-contact momentum control. In *Proceeding of the IEEE/RSJ International Conference on Humanoid Robots*, pages 874–880. IEEE, 2015.

F. Islam, J. Nasir, U. Malik, Y. Ayaz, and O. Hasan. RRT*-smart: Rapid convergence implementation of RRT* towards optimal solution. In *2012 IEEE International Conference on Mechatronics and Automation*, pages 1651–1656. IEEE, 2012.

S. J. Jorgensen, M. Vedantam, R. Gupta, H. Cappel, and L. Sentis. Finding locomanipulation plans quickly in the locomotion constrained manifold. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020.

S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1620–1626, 2003.

S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime motion planning using the rrt*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1478–1483. IEEE,

2011.

N. Kariotoglou, S. Summers, T. Summers, M. Kamgarpour, and J. Lygeros. Approximate dynamic programming for stochastic reachability. In *Proceedings of European Control Conference*, pages 584–589, 2013.

L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *International Journal of Robotics and Automation*, 3(1):43–53, 1987.

D. Kim, J. Ahn, O. Campbell, N. Paine, and L. Sentis. Investigations of a robotic testbed with viscoelastic liquid cooled actuators. *IEEE/ASME Transactions on Mechatronics*, 2018.

D. Kim, S. J. Jorgensen, J. Lee, J. Ahn, J. Luo, and L. Sentis. Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control. *International Journal of Robotics Research*, page 0278364920918014, 2019.

J.-H. Kim. Improved ellipsoidal bound of reachable sets for time-delayed linear systems with disturbances. *Automatica*, 44(11):2940–2943, 2008.

J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 995–1001. IEEE, 2000.

C. Le Guernic and A. Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250–262, 2010.

I. Lee and J.-H. Oh. Humanoid posture selection for reaching motion and a cooperative balancing controller. *Journal of Intelligent & Robotic Systems*, 81 (3-4):301–316, 2016.

J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. S. Srinivasa, M. Stilman, and C. K. Liu. Dart: Dynamic animation and robotics toolkit. *The Journal of Open Source Software*, 3(22):500, 2018.

J. Lee, E. Bakolas, and L. Sentis. Trajectory generation for robotic systems with contact force constraints. In *Proceedings of American Control Conference*, pages 671–678. IEEE, 2019.

J. Lee, J. Ahn, E. Bakolas, and L. Sentis. Reachability-based trajectory optimization for robotic systems given sequences of rigid contacts. In *Proceedings of American Control Conference*, pages 2158–2165. IEEE, 2020.

L. Liebenwein, C. Baykal, I. Gilitschenski, S. Karaman, and D. Rus. Sampling-based approximation algorithms for reachability analysis with provable guarantees. In *Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.014.

Y. Liu, P. M. Wensing, D. E. Orin, and Y. F. Zheng. Trajectory generation for dynamic walking in a humanoid over uneven terrain using a 3d-actuated dual-slip model. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 374–380. IEEE, 2015.

D. M. Lofaro, R. Ellenberg, P. Oh, and J.-H. Oh. Humanoid throwing: Design of collision-free trajectories with sparse reachable maps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1519–1524. IEEE, 2012.

J. Maidens and M. Arcak. Reachability analysis of nonlinear systems using matrix measures. *IEEE Transactions on Automatic Control*, 60(1):265–270, 2015.

Z. Manchester and S. Kuindersma. Robust direct trajectory optimization using approximate invariant funnels. *Autonomous Robots*, 43(2):375–387, 2019.

N. Mansard, O. Khatib, and A. Kheddar. A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transactions on Robotics*, 25(3):670–685, 2009.

C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, S. Vijayakumar, and N. Mansard. Crocoddyl: An efficient and versatile framework for multi-contact optimal control. *arXiv preprint arXiv:1909.04947*, 2019.

C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini. Motion planning for quadrupedal locomotion: coupled planning, terrain mapping and whole-body control. *arXiv preprint arXiv:2003.05481*, 2020.

I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

A. Moreira and M. Y. Santos. Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. 2007.

Q. Nguyen and K. Sreenath. Optimal robust control for bipedal robots through control lyapunov function based quadratic programs. In *Robotics: Science and Systems*. Rome, Italy, 2015.

Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath. 3d dynamic walking on stepping stones with control barrier functions. In *Proceedings of the IEEE Conference on Decision and Control*, pages 827–834. IEEE, 2016.

M. Posa, C. Cantu, and R. Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *International Journal of Robotics Research*, 33(1):69–81, 2014.

M. Posa, S. Kuindersma, and R. Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1366–1373. IEEE, 2016.

N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, et al. Valkyrie: Nasa's first bipedal humanoid robot. *Journal of Field Robotics*, 32(3):397–419, 2015.

L. Righetti, J. Buchli, M. Mistry, and S. Schaal. Inverse dynamics control of floating-base robots with external constraints: A unified view. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1085–1090. IEEE, 2011.

M. Rungger and M. Zamani. Accurate reachability analysis of uncertain nonlinear systems. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*, pages 61–70. ACM, 2018.

B. Sakcak, L. Bascetta, G. Ferretti, and M. Prandini. Sampling-based optimal kinodynamic planning with motion primitives. *Autonomous Robots*, pages 1–18, 2019.

L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(04):505–518, 2005.

A. Sintov. Goal state driven trajectory optimization. *Autonomous Robots*, 43(3):631–648, 2019.

B. J. Stephens and C. G. Atkeson. Dynamic balance force control for compliant humanoid robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1248–1255, 2010.

Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012.

N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann. Humanoid motion planning for dual-arm manipulation and re-grasping tasks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2464–2470. IEEE, 2009.

N. Vahrenkamp, T. Asfour, and R. Dillmann. Robot placement based on reachability inversion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1970–1975. IEEE, 2013.

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

P. M. Wensing and D. E. Orin. Improved computation of the humanoid centroidal dynamics and application for whole-body control. *International Journal of Humanoid Robotics*, 13(01):1550039, 2016.

P.-B. Wieber. Holonomy and nonholonomy in the dynamics of articulated motion. In *Fast motions in biomechanics and robotics*, pages 411–425. Springer, 2006.

Y. Yang, W. Merkt, H. Ferrolho, V. Ivan, and S. Vijayakumar. Efficient humanoid motion planning on uneven terrain using paired forward-inverse dynamic reachability maps. *IEEE Robotics and Automation Letters*, 2(4):2279–2286, 2017.