Hierarchical Task-Space Optimal Covariance Control with Chance Constraints

Jaemin Lee, Efstathios Bakolas, *Member, IEEE* and Luis Sentis, *Member, IEEE*

***Abstract*— This letter presents a new control paradigm applicable to nonlinear systems such as robots subject to chance and covariance assignment constraints which we refer to as hierarchical optimal covariance control. To the best of our knowledge, this is the first study to formulate the hierarchical optimal covariance control problem involving multiple operational tasks. The framework is defined as a multi-stage optimization problem considering multiple hierarchical tasks specified in lexicographic order. Towards this goal, we first approximate the nonlinear dynamic model of a robot into multiple linear stochastic systems by linearizing the model along given trajectories. We then project these stochastic models onto the null-space of the previous task models for efficiently solving lexicographical optimization. In addition, we specify probability functions to account for chance constraints using the Whittacker M function. We formulate the chance constraints as a positive semi-definite matrix constraint and solve the hierarchical optimal covariance control problem using sequential semi-definite programming. We demonstrate that this procedure yields higher accuracy for multiple hierarchical tasks than employing deterministic operational space control models.**

***Index Terms*— Stochastic Control, Hierarchical Task-Space Control, Robotics**

I. Introduction

THIS letter presents a control paradigm, which we refer to as hierarchical covariance control (HCC), to control highly articulated nonlinear robotic systems when executing multiple operational tasks [1]. We formulate HCC using lexicographic optimization to solve for the task hierarchy in redundant robots [2]. The main purpose of this work is to steer all tasks' states to the desired Gaussian distributions, while optimizing the objective functions of the tasks in their order of importance. In addition to the mean and covariance constraints associated with the terminal states, we consider equality constraints enforcing the task hierarchy and chance constraints restricting the probability that the state exceeds the tunnel defined by a Euclidean distance on the states.

State-of-the-art control of redundant robotic systems uses task hierarchies to synthesize complex task behaviors [2]–[5]. For instance, Operational Space Control (OSC) [1], and Task-Space Inverse Dynamics Control (TSIDC) [6] have been developed to track accurately given task trajectories based on deterministic models. However, most robotic systems such as legged robots, aerial manipulators, and underwater robots contain high degree of uncertainty due to sensor noise, structural deformations, complex mechanical transmissions, and external perturbations [7]. Tracking performance of robots with uncertain kinematics, dynamics and sensing can be improved using adaptive control [8], [9]. At the joint torque level, Gaussian stochastic models are employed to enhance the robustness of task-space controllers [10]. Although the above studies have shown improved performance under uncertainty, the use of covariance constraints on the terminal states has not been previously studied as a way to substantially decrease task errors when uncertainty is high.

Optimal covariance control (OCC) has been broadly studied to control stochastic systems subject to covariance constraints. Recently, linear stochastic systems have been controlled over a finite-time horizon considering not only covariance constraints but also state means and control input constraints [11]–[13]. In addition to covariance constraints, chance constraints can be incorporated to restrict the probability of constraint violation [14], [15]. OCC researchers have extended their studies to nonlinear dynamical systems by linearizing the system dynamics along a reference trajectory [16], or at the current state of the system [17]. An iterative covariance control problem has been proposed to steer nonlinear uncertainty by iteratively solving an approximated, and linearized problem [16]. Also, stochastic differential dynamic programming has been used to solve the nonlinear OCC problem [18].

The above OCC studies only consider the problem of steering the state of a single task or output. For instance, an autonomous vehicle is controlled using OCC using the location of the vehicle as the state [19], [20]. Our work, in contrast, deals with multiple prioritized tasks defined by using the nonlinear kinematics and dynamics of a robot. The authors in [21] formulate an OCC problem for multi-agent systems. Although the inter-agent constraints are considered, the approach mentioned above does not consider shared dynamic properties or a hierarchy among the agents. To the best of our knowledge, OCC has not been studied for nonlinear systems executing multiple hierarchical tasks on a shared dynamical model to date.

The main contribution of this study is to formulate and solve an HCC problem for nonlinear robotic systems with uncertainty represented as Gaussian white noise. We deal with the task hierarchy by solving a lexicographic optimization problem in a similar way to Hierarchical Quadratic Programming (HQP) in [2]. HQP solves an instantaneous-time optimization problem. In contrast, our formulation provides the means to control multiple tasks enforcing a hierarchy over a finite horizon. Our solution efficiently provides for an operational task hierarchy for stochastic systems. In addition, we incorporate chance constraints thus determining the probability that the states remain in the trust tubes. Our task hierarchies impose that the tubes of higher prioritized

This work was supported by NSF Grants# 1724360 and #1924790, and ONR Grant# N000141512507.

J. Lee is with the Department of Mechanical Engineering, The University of Texas at Austin, TX, 78712, USA (e-mail: jmlee87@utexas.edu).

E. Bakolas and L. Sentis are with the Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, TX, 78712, USA (email: bakolas@utexas.edu; lsentis@utexas.edu).

tasks are narrower than those of lower prioritized ones. Our formulation also incorporates task space equality constraints. These techniques reduce the computation burden of the optimization. Chance constraints are described explicitly using the Whittacker M function however they are nonlinear. If we directly employ Nonlinear Programming (NLP) to solve the proposed problem, a significant amount of computation time will be required and the final result will be highly sensitive to initial guesses. Recently, constrained nonlinear optimization problems can be solved using collocation methods. However, these methods are still costly because they formulate large optimization problems over all control variables. By using convex relaxation techniques [12], we solve the proposed HCC problem via sequential Semi-Definite Programming (SDP). SDP is capable of solving the problem much faster than NLP problems. Numerical simulations clearly show that the proposed approach improves the control accuracy compared to operational space control.

II. Problem Statement

A. Preliminaries

In general, with a configuration space $\mathcal{Q} \subset \mathbb{R}^n$ and an input space $\mathcal{U} \subset \mathbb{R}^n$, the rigid-body dynamics of a robotic system is described as

$$M(q)\ddot{q} + b(\dot{q}, q) = u \tag{1}$$

where $q \in \mathcal{Q}$ and $u \in \mathcal{U}$ denote the joint variable and control torque input, respectively. $M(q) \in \mathbb{S}^n_{++}$ and $b : \mathcal{Q} \times \mathbb{R}^n \mapsto \mathbb{R}^n$ represent the mass/inertia matrix and the functional mapping for the sum of Coriolis/centrifugal and gravitational forces. Given m hierarchical tasks, we transform the above dynamic equation into the operational space dynamic equation for the i-th task, $\vartheta_i = g_i(q) \in \mathbb{R}^{n_i}$ where $g_i : \mathcal{Q} \mapsto \mathbb{R}^{n_i}$ is a C^1 function, as follows:

$$\ddot{\vartheta}_i + p_i(\dot{q}, q) = J_i(q)M(q)^{-1}u \tag{2}$$

where $J_i(q) = \frac{\partial \vartheta_i}{\partial q}$ and $p_i(\dot{q}, q) = J_i(q)M(q)^{-1}b(\dot{q}, q) - \dot{J}_i(\dot{q}, q)\dot{q}$. In turn we represent the i-th task-space subsystems in state-space form with $x_i = [\vartheta_i^\top, \dot{\vartheta}_i^\top]^\top \in \mathbb{R}^{2n_i}$:

$$\mathrm{d}x_i = f_i(x_i, u)\mathrm{d}t + \mathcal{C}(t)\mathrm{d}w \tag{3}$$

where $w(t)$ is a standard Brownian motion, $\mathcal{C}(t)$ is a C^1 function, and

$$f_i(x_i, u) = \begin{bmatrix} \dot{\vartheta}_i \\ J_i(q)M(q)^{-1}u - p_i(\dot{q}, q) \end{bmatrix}. \tag{4}$$

Using the discretization and linearization techniques described in, for instance, Section II of [16], we can obtain the following discrete-time linear stochastic system:

$$x_{i,\tau+1} = A_{i,\tau}x_{i,\tau} + B_{i,\tau}u_{i,\tau} + r_{i,\tau} + C_{i,\tau}w_\tau \tag{5}$$

where $\tau \in [0, N-1]_d$ and N is a positive integer. $A_{i,\tau}$, $B_{i,\tau}$, $C_{i,\tau}$, and $r_{i,\tau}$ are defined as in [16]. In addition, $\mathbb{E}\left[w_\tau w_s^\top\right] = \delta(\tau, s)W_\tau$, and $W_\tau \in \mathbb{S}^{2n}_+$. δ denotes the Kronecker function computed as follows: $\delta(\tau, s) = 1$ when $\tau = s$ and $\delta(\tau, s) = 0$, otherwise. Furthermore, we assume that $\mathbb{E}\left[x_{i,\tau}w_s^\top\right] = \mathbf{0}$ and $\mathbb{E}\left[w_s x_{i,\tau}^\top\right] = \mathbf{0}$ where $s \in [\tau, N-1]_d$. We will consider this state-space model in (5) to formulate the HCC problem.

B. Hierarchical Covariance Control Problem

The objective of HCC is to steer all task trajectories of the robotic system to a desired terminal (N-th) Gaussian distribution. We denote by $\mu_{i,\tau} \in \mathbb{R}^{2n_i}$ and $\Sigma_{i,\tau} \in \mathbb{S}^{2n_i}_+$ the mean vector and covariance matrix for the i-th task's Gaussian distribution at the τ-th stage and by $\mu^d_{i,N}$ and $\Sigma^d_{i,N}$ the corresponding desired quantities at the N-th stage, while optimizing multiple performance indices in a lexicographic order

$$\operatorname{lexmin} \quad \mathcal{I}(\mathcal{J}_1, \cdots, \mathcal{J}_i, \cdots \mathcal{J}_m) \tag{6}$$

where $\mathcal{J}_i$ denotes the performance index for the i-th task. In addition, we incorporate chance constraints which require that the probability of the state remaining within a given trust region is equal or greater than a pre-defined threshold. Since in most cases task-space (operational space) is defined as a subspace of the Euclidean space to control the positions of robot body parts [1]–[5], we use the Euclidean distance between task states, $x_{i,\tau}$, and their state mean to formulate the chance constraint, i.e.:

$$\mathbb{P}[d_{i,\tau} \le \varepsilon_i] \ge \eta, \quad d^2_{i,\tau} = \|x_{i,\tau} - \mu_{i,\tau}\|^2 \tag{7}$$

with $\varepsilon_i > 0$, $0 \le \eta \le 1$, $i \in \{1, \cdots, m\}$, and $\tau \in \{0, \cdots, N\}$. We employ chance constraints to control the effect of uncertainty while steering the robotic system. The chance constraints ensure that the probability of keeping the Euclidean distance between the task states small is above a certain level over the interval $[0, N]$. Based on these descriptions, we formulate the following optimal covariance control problem to satisfy the given terminal conditions.

Problem 1. *Let $N \in \mathbb{Z}_{++}$, initial conditions, $\mu_{i,0}$, $\Sigma_{i,0}$, and terminal conditions, $\mu^d_{i,N}$, $\Sigma^d_{i,N}$ be known a priori. The optimal covariance control problem for a linear stochastic system is formulated as follows:*

$$\min \quad \mathcal{J}_i := \mathbb{E}\left[\sum_{\tau=0}^{N-1} x_{i,\tau}^\top R_{x_i} x_{i,\tau} + u_{i,\tau}^\top R_{u_i} u_{i,\tau}\right]$$

$$\text{s.t.} \quad x_{i,\tau+1} = A_{i,\tau}x_{i,\tau} + B_{i,\tau}u_{i,\tau} + r_{i,\tau} + C_{i,\tau}w_\tau \tag{8a}$$

$$\mu_{i,N} = \mathbb{E}[x_{i,N}] = \mu^d_{i,N}, \tag{8b}$$

$$\Sigma_{i,N} = \Sigma^d_{i,N}, \tag{8c}$$

$$\mathbb{P}\left[d_{i,\tau} \le \varepsilon_i\right] \ge \eta_i, \tag{8d}$$

$$\mathbb{E}\left[B_{k,\tau}u_{i,\tau}\right] = \mathbf{0}, \quad \forall k \in \{1, \cdots, i-1\} \tag{8e}$$

where $R_{x_i} \in \mathbb{S}^{2n_i}_+$, and $R_{u_i} \in \mathbb{S}^{n_i}_+$ are given and $\varepsilon_k(\tau) \le \varepsilon_i(\tau)$ where $\tau \in [0, N]$. The constraint (8e) enforces the hierarchy and vanishes when $i = 1$.

The constraint (8e) aims to perturb less the optimal trajectory of the k-th task due to the control command $u_{i,\tau}$. Therefore, the constraint (8e) ensures that the k-th task error stays smaller than the error of the i-th task. The initial state $x_{q,0} = [q_0^\top, \dot{q}_0^\top]^\top$ is a random vector; in particular, $q_0 \sim \mathcal{N}(\mu_{q,0}, \Sigma_{q,0})$ and $\dot{q}_0 = \mathbf{0}$, which corresponds to a static configuration. Then, it is possible to transform $x_{q,0}$, $\mu_{q,0}$, and $\Sigma_{q,0}$ into $x_{i,0}$, $\mu_{i,0}$, and $\Sigma_{i,0}$ for the i-th task space using the functional mapping g_i. To solve the lexicographic optimization in (6), we sequentially solve Problem 1 for all $i \in \{1, \cdots, m\}$.

III. The Proposed Method

In this section, we properly modify the cost function and define the constraints which include the task-space dynamics, the task hierarchy, and the mean and covariance terminal conditions. The resulting problem is turned into a Semi-Definite Program (SDP).

A. Linear Model with a Decision Variable

We define a finite-horizon covariance control problem for discrete-time stochastic linear systems [12]. Consider the concatenated state and control input vectors: $\boldsymbol{x_i} := [x_{i,0}^\top, \cdots, x_{i,N}^\top]^\top$, $\boldsymbol{u_i} := [u_{i,0}^\top, \cdots, u_{i,N-1}^\top]^\top$, $\mathbf{r}_i := [r_{i,0}^\top, \cdots, r_{i,N-1}^\top]^\top$, and $\boldsymbol{w} := [w_{i,0}^\top, \cdots, w_{i,N-1}^\top]$. In addition, the control torque input is computed as $u_\tau = \sum_{i=1}^m u_{i,\tau}$, or $\boldsymbol{u} = \sum_{i=1}^m \boldsymbol{u}_i$ when considering the concatenated vectors. We formulate a linear stochastic system on the concatenated vectors in a similar form to [12]:

$$\boldsymbol{x}_i = \boldsymbol{\mathcal{A}}_i x_{i,0} + \boldsymbol{\mathcal{B}}_i \boldsymbol{u} + \boldsymbol{\mathcal{C}}_i \boldsymbol{w} + \boldsymbol{\mathcal{D}}_i \mathbf{r}_i \tag{9}$$

where $\boldsymbol{\mathcal{A}}_i = \boldsymbol{\Omega}_i(0)$ and

$$\begin{aligned} \boldsymbol{\mathcal{B}}_i =& [\boldsymbol{\Omega}_i(1)B_{i,0}, \ \cdots, \ \boldsymbol{\Omega}_i(N)B_{i,N-1}], \\ \boldsymbol{\mathcal{C}}_i =& [\boldsymbol{\Omega}_i(1)C_{i,0}, \ \cdots, \ \boldsymbol{\Omega}_i(N)C_{i,N-1}], \\ \boldsymbol{\mathcal{D}}_i =& [\boldsymbol{\Omega}_i(1), \ \cdots, \ \boldsymbol{\Omega}_i(N)], \\ \boldsymbol{\Omega}_i(s) =& [\boldsymbol{\Phi}_i(0,s)^\top, \ \cdots, \ \boldsymbol{\Phi}_i(N,s)^\top]^\top. \end{aligned} \tag{10}$$

In addition, $\boldsymbol{\Phi}_i(\tau, s) = A_{i,\tau-1} \cdots A_{i,s}$ and $\boldsymbol{\Phi}_i(s,s) = \mathbf{I}$ when $\tau \geq s$ and $\boldsymbol{\Phi}_i(\tau, s) = \mathbf{0}$, otherwise.

Let us formulate an admissible control policy. In principle, the control policy consists of feedback and feedforward terms. However, when the terminal state mean is close to zero or the feedforward terms are considered in the joint space dynamics, it suffices to consider control policies that can be represented as follows:

$$\boldsymbol{\pi}_i(x_{i,0}, \cdots, x_{i,\tau}; \tau) = \sum_{s=0}^{\tau} K_i(\tau, s) x_{i,s} \tag{11}$$

where $K_i(\tau, s) \in \mathbb{R}^{n_i \times 2n_i}$ for all $\tau \in [0, N-1]_d$ with $s \leq \tau$. In the general case, an additional feedforward term should be included in (11) (in other words, the control policy should be an affine function of past and current states rather than a linear one as in (11)). The loss of generality is, however, minimal and one can easily re-derive the subsequent formulas for the more general case after making the necessary minor changes. The control input can therefore be computed using $\boldsymbol{u}_i = \boldsymbol{\mathcal{K}}_i \boldsymbol{x}_i$ where

$$\boldsymbol{\mathcal{K}}_i := \begin{bmatrix} K_i(0,0) & \mathbf{0} & \cdots & \mathbf{0} \\ K_i(1,0) & K_i(1,1) & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ K_i(N-1,0) & K_i(N-1,1) & \cdots & \mathbf{0} \end{bmatrix}. \tag{12}$$

Then, we obtain the expression of the closed loop dynamics for each state as follows:

$$\boldsymbol{x}_i = (\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)^{-1} (\mathbf{X}_i + \boldsymbol{\mathcal{B}}_i \overline{\boldsymbol{u}}_{i-1}^\star) \tag{13}$$

where $(\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)$ is invertible. In addition, $\mathbf{X}_i = \boldsymbol{\mathcal{A}}_i x_{i,0} + \boldsymbol{\mathcal{C}}_i \boldsymbol{w} + \boldsymbol{\mathcal{D}}_i \mathbf{r}_i$ and $\overline{\boldsymbol{u}}_{i-1}^\star = \sum_{k=1}^{i-1} \boldsymbol{u}_k^\star$ where $\boldsymbol{u}_k^\star$ is the optimal control input for the k-th operational task and $\overline{\boldsymbol{u}}_0^\star = \mathbf{0}$. It is noted that $(.)^\star$ denotes the optimal properties. Let us define the decision variable, $\boldsymbol{\Psi}_i := \boldsymbol{\mathcal{K}}_i (\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)^{-1}$. Therefore we can express $\boldsymbol{x}_i$ with respect to $\boldsymbol{\Psi}_i$ as follows:

$$\boldsymbol{x}_i = (\mathbf{I} + \boldsymbol{\mathcal{B}}_i \boldsymbol{\Psi}_i)(\mathbf{X}_i + \boldsymbol{\mathcal{B}}_i \overline{\boldsymbol{u}}_{i-1}^\star). \tag{14}$$

Proposition 1. *$(\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)$ is invertible and its inverse matrix is $\mathbf{I} + \boldsymbol{\mathcal{B}}_i \boldsymbol{\Psi}_i$.*

Proof. Since $\boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i$ is a strict block lower triangular matrix, which is nilpotent: i.e., $(\boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)^n = \mathbf{0}$ for some $n \in \mathbb{N}$:

$$(\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)(\mathbf{I} + \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i + \cdots + (\boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)^{n-1}) = \mathbf{I} - (\boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)^n = \mathbf{I}$$

After simple algebraic manipulation, it follows

$$\begin{aligned} (\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)^{-1} =& (\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i + \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)(\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)^{-1} \\ =& \mathbf{I} + \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i (\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)^{-1} = \mathbf{I} + \boldsymbol{\mathcal{B}}_i \boldsymbol{\Psi}_i. \end{aligned}$$

It is noted that $\boldsymbol{\Psi}_i$ is a block lower triangular matrix, since $\boldsymbol{\mathcal{B}}_i$ and $\boldsymbol{\mathcal{K}}_i$ are block lower triangular matrices. □

B. Constraints Determining the Task Hierarchy

By using the concatenated model in (9), we next show that the optimal values of the objective functions of previous tasks $\mathcal{J}_1^\star, \cdots, \mathcal{J}_{i-1}^\star$ will not be affected by the control input $\boldsymbol{u}_i^\star$ minimizing the objective function $\mathcal{J}_i$.

Proposition 2. *If $\mathbb{E}\left[B_{k,\tau} u_{i,\tau}\right] = \mathbf{0}$ for all $k \in \{1, \cdots, i-1\}$, $\boldsymbol{u}_i^\star$ does not affect $\mathcal{J}_1^\star, \cdots, \mathcal{J}_{i-1}^\star$.*

Proof. Assume that R_{x_k} and R_{u_k} are specified a priori. Also assume that $\boldsymbol{u}_k^\star$ are computed by solving prior optimization problems defined in Problem 1 for $i = \{1, \cdots, k\}$. Next, we will show that $\mathbb{E}\left[\mathrm{trace}(\boldsymbol{x}_k \boldsymbol{x}_k^\top)\right] = \mathbb{E}\left[\mathrm{trace}(\boldsymbol{x}_k^\star \boldsymbol{x}_k^{\star\top})\right]$. Since adding an additional task $\boldsymbol{u}_i$ is equivalent to adding a perturbation to the optimal trajectory, i.e. $\boldsymbol{x}_k = \boldsymbol{x}_k^\star + \boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i$, we have

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{x}_k \boldsymbol{x}_k^\top\right] =& \mathbb{E}\left[(\boldsymbol{x}_k^\star + \boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i)(\boldsymbol{x}_k^\star + \boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i)^\top\right] \\ =& \mathbb{E}\left[\boldsymbol{x}_k^\star \boldsymbol{x}_k^{\star\top}\right] + \mathbb{E}\left[\boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i \boldsymbol{x}_k^{\star\top}\right] + \mathbb{E}\left[\boldsymbol{x}_k^\star \boldsymbol{u}_i^\top \mathcal{B}_k^\top\right] \\ &+ \mathbb{E}\left[\boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i \boldsymbol{u}_i^\top \boldsymbol{\mathcal{B}}_k^\top\right] = \mathbb{E}\left[\boldsymbol{x}_k^\star \boldsymbol{x}_k^{\star\top}\right] \end{aligned}$$

where $\mathbb{E}\left[\boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i\right] = \mathbb{E}\left[\sum_{j=0}^{N-1} \boldsymbol{\Omega}_k(j+1) B_{k,j} u_{i,j}\right] = \mathbf{0}$. In addition, $\mathbb{E}\left[\boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i \boldsymbol{x}_k^{\star\top}\right] = \mathbb{E}\left[\sum_{j=0}^{N-1} \boldsymbol{\Omega}_k(j+1) B_{k,j} u_{i,j} x_{k,j}^{\star\top}\right] = \mathbf{0}$ implies that $\mathbb{E}[\boldsymbol{x}_k^\star \boldsymbol{u}_i^\top \mathcal{B}_k^\top] = \mathbf{0}$. Therefore, $\boldsymbol{u}_i$ does not affect $\mathcal{J}_1^\star, \cdots, \mathcal{J}_{i-1}^\star$. □

The above term $\mathbb{E}\left[B_{k,\tau} u_{i,\tau}\right]$ is expressed in terms of $\boldsymbol{\Psi}_i$ as follows:

$$\begin{aligned} \mathbb{E}\left[B_{k,\tau} u_{i,\tau}\right] =& B_{k,\tau} \mathbf{S}_\tau \boldsymbol{\Psi}_i \left(\mathbb{E}[\mathbf{X}_i] + \boldsymbol{\mathcal{B}}_i \mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]\right) & (15) \\ \mathbb{E}[\mathbf{X}_i] =& \boldsymbol{\mathcal{A}}_i \mu_{i,0} + \boldsymbol{\mathcal{D}}_i \mathbf{r}_i & (16) \end{aligned}$$

where $\mathbf{S}_\tau = [\mathbf{0}, \cdots, \mathbf{I}, \cdots, \mathbf{0}] \in \mathbb{R}^{n \times (N+1)n}$ is a block matrix whose $\tau + 1$-th sub-matrix is identity. For instance, $\mathbf{S}_0 = [\mathbf{I}, \cdots, \mathbf{0}]$ and $\mathbf{S}_N = [\mathbf{0}, \cdots, \mathbf{I}]$. We concatenate these constraints to enforce the task hierarchy for the i-th task in the following manner:

$$\begin{bmatrix} B_{1,N}^{\mathrm{bdiag}} \\ \vdots \\ B_{i-1,N}^{\mathrm{bdiag}} \end{bmatrix} \boldsymbol{\Psi}_i (\mathbb{E}[\mathbf{X}_i] + \boldsymbol{\mathcal{B}}_i \mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]) = \mathbf{0} \tag{17}$$

where $B_{k,N}^{\mathrm{bdiag}} = \mathrm{bdiag}(B_{k,1}, \cdots, B_{k,N})$. Lower priority tasks contain more constraints in (17) since the i-th index is higher, which means that the size of the constraint matrix becomes larger, (size = $\sum_{k=1}^{i-1} 2n_k N \times nN$) resulting in a significant increase of the computational time.

Instead of explicitly considering the above equality constraint matrix, we employ a null-space projection matrix onto higher prioritized tasks as follows:

$$B_{k,\tau}^{\text{null}} = \left[\begin{array}{c} \mathbf{0} \\ J_{k|\text{null}}(q_\tau^d) M(q_\tau^d)^{-1} \end{array} \right], \quad (18)$$

where $J_{k|\text{null}}(q) = J_k(q)\mathbf{N}_{k-1}(q)$, $J_{1|\text{null}}(q) = J_1(q)$, and $\mathbf{N}_k(q) = \mathbf{I} - J_{k|\text{null}}(q)^\dagger J_{k|\text{null}}(q)$. Since $\mathbf{N}_k(q)M(q)^{-1} = M(q)^{-1}\mathbf{N}_k(q)^\top$, we can write that $B_{k,\tau}^{\text{null}} = B_{k,\tau}\mathbf{N}_{k-1}(q_\tau^d)^\top$. Then, the perturbed task-space dynamics fulfilling the constraint (17) becomes

$$\boldsymbol{x}_i = \boldsymbol{\mathcal{A}}_i x_{i,0} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}(\boldsymbol{u}_i + \overline{\boldsymbol{u}}_{i-1}^\star) + \boldsymbol{\mathcal{C}}_i \boldsymbol{w} + \boldsymbol{\mathcal{D}}_i \mathbf{r}_i \quad (19)$$

where $\boldsymbol{\mathcal{B}}_{\mathbf{N}_i} = [\boldsymbol{\Omega}_i(1)B_{i,0}^{\text{null}}, \cdots, \boldsymbol{\Omega}_i(N)B_{i,N-1}^{\text{null}}]$. We now express the decision variable $\boldsymbol{\Psi}_i$ with the respect to the constrained matrix $\boldsymbol{\mathcal{B}}_{\mathbf{N}_i}$: $\boldsymbol{\Psi}_i = \boldsymbol{\mathcal{K}}_i(\mathbf{I} - \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\mathcal{K}}_i)^{-1}$. Since $\boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\mathcal{K}}_i$ is a strict block lower triangular matrix, $(\mathbf{I} - \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\mathcal{K}}_i)$ is invertible and $(\mathbf{I} - \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\mathcal{K}}_i)^{-1} = \mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i$. The detailed proof is identical to Proposition 1 and therefore is omitted. We modify $\boldsymbol{x}_i$ using the decision variable $\boldsymbol{\Psi}_i$ as follows:

$$\boldsymbol{x}_i = (\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)(\mathbf{X}_i + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\overline{\boldsymbol{u}}_{i-1}^\star) \quad (20)$$

where $\boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i = \mathbf{0}$ for all $k < i$ as is shown in the proposition that follows.

Proposition 3. *$\boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i = \mathbf{0}$ where $\boldsymbol{u}_i$ is obtained via equation (20) for all $k < i$.*

Proof. The matrix $\boldsymbol{\mathcal{B}}_{\mathbf{N}_i}$ can be expressed as $\boldsymbol{\mathcal{B}}_{\mathbf{N}_i} = \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{N}}_i^\top$ where $\boldsymbol{\mathcal{N}}_i = \text{bdiag}(\mathbf{N}_{i-1,0}, \cdots, \mathbf{N}_{i-1,N-1})$ and the simplified notation $\mathbf{N}_{i-1,\tau} = \mathbf{N}_{i-1}(q_\tau^d)$ has been used. Then, we can express $\boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i$ as follows:

$$\begin{aligned} \boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i &= \boldsymbol{\mathcal{B}}_k \boldsymbol{\mathcal{K}}_i \boldsymbol{x}_i = \boldsymbol{\mathcal{B}}_k (\mathbf{I} + \boldsymbol{\Psi}_i \boldsymbol{\mathcal{B}}_{\mathbf{N}_i})^{-1} \boldsymbol{\Psi}_i \boldsymbol{x}_i \\ &= \boldsymbol{\mathcal{B}}_k (\boldsymbol{\Psi}_i - \boldsymbol{\Psi}_i \boldsymbol{\mathcal{B}}_{\mathbf{N}_i} \boldsymbol{\mathcal{K}}_i) \boldsymbol{x}_i = \boldsymbol{\mathcal{B}}_k (\boldsymbol{\Psi}_i - \boldsymbol{\Psi}_i \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i^{\mathbf{N}}) \boldsymbol{x}_i \end{aligned}$$

where $\boldsymbol{\mathcal{K}}_i^{\mathbf{N}} = \boldsymbol{\mathcal{N}}_{i-1}^\top \boldsymbol{\mathcal{K}}_i$. Since $\boldsymbol{\Psi}_i \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i^{\mathbf{N}} = \boldsymbol{\mathcal{K}}_i^{\mathbf{N}} \boldsymbol{\mathcal{B}}_i \boldsymbol{\Psi}_i$ and $\boldsymbol{\mathcal{B}}_k \boldsymbol{\mathcal{N}}_{i-1}^\top = \mathbf{0}$, we can write that $\boldsymbol{\mathcal{B}}_k \boldsymbol{\mathcal{K}}_i \boldsymbol{x}_i = \boldsymbol{\mathcal{B}}_k \boldsymbol{\Psi}_i \boldsymbol{x}_i$. Because $\boldsymbol{\mathcal{K}}_i$ and $\boldsymbol{\Psi}_i$ are block lower triangular matrices, we determine that either $\boldsymbol{\mathcal{K}}_i = \boldsymbol{\Psi}_i = \boldsymbol{\mathcal{K}}_i(\mathbf{I} - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i)^{-1}$ or $\boldsymbol{\mathcal{B}}_k \boldsymbol{\mathcal{K}}_i = \boldsymbol{\mathcal{B}}_k \boldsymbol{\Psi}_i = \mathbf{0}$ to satisfy the above equation. When $\boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{K}}_i \neq \mathbf{0}$, i.e., $\boldsymbol{u}_i \neq \mathbf{0}$, $\boldsymbol{\mathcal{K}}_i$ cannot be identical to $\boldsymbol{\Psi}_i$. In that case $\boldsymbol{\mathcal{B}}_k \boldsymbol{\mathcal{K}}_i = \boldsymbol{\mathcal{B}}_k \boldsymbol{\Psi}_i = \mathbf{0}$. Therefore, $\boldsymbol{\mathcal{B}}_k \boldsymbol{u}_i$ is zero for all $k < i$. □

Using the formulation in (20), avoids the need to include the matrix equality constraint (17). As such, the computational complexity of the SDP problem does not change significantly when incorporating task hierarchy constraints.

C. Performance Index

In this section, we express the performance index defined in Problem 1 in terms of the decision variable $\boldsymbol{\Psi}_i$. Based on the constrained state-space model defined in (20), the performance index takes the following form:

$$\mathcal{J}_i(\boldsymbol{\Psi}_i) = \mathbb{E}\left[\text{trace}\left(\boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{\mathcal{R}}_{\boldsymbol{x}_i} + \boldsymbol{u}_i \boldsymbol{u}_i^\top \boldsymbol{\mathcal{R}}_{\boldsymbol{u}_i}\right)\right] \quad (21)$$

with $\boldsymbol{\mathcal{R}}_{\boldsymbol{x}_i} := \text{bdiag}(R_{x_i}, \cdots, R_{x_i}) \in \mathbb{S}_+^{2(N+1)n_i}$ and $\boldsymbol{\mathcal{R}}_{\boldsymbol{u}_i} := \text{bdiag}(R_{u_i}, \cdots, R_{u_i}) \in \mathbb{S}_+^{Nn}$. We express the following expectations to represent the performance index in terms of $\boldsymbol{\Psi}_i$.

$$\begin{aligned} \mathbb{E}[\boldsymbol{x}_i \boldsymbol{x}_i^\top] &= (\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)\mathbf{E}_i(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)^\top, \quad (22) \\ \mathbb{E}[\boldsymbol{u}_i \boldsymbol{u}_i^\top] &= \boldsymbol{\Psi}_i \mathbf{E}_i \boldsymbol{\Psi}_i^\top \quad (23) \end{aligned}$$

where $\mathbf{E}_i = \boldsymbol{\mathcal{A}}_i(\Sigma_{i,0} + \mu_{i,0}\mu_{i,0}^\top)\boldsymbol{\mathcal{A}}_i^\top + \boldsymbol{\mathcal{C}}_i \mathbb{E}\left[\boldsymbol{w}\boldsymbol{w}^\top\right]\boldsymbol{\mathcal{C}}_i^\top + \boldsymbol{\mathcal{D}}_i \mathbf{r}_i \mathbf{r}_i^\top \boldsymbol{\mathcal{D}}_i^\top + \mathbb{E}[\mathbf{X}_i]\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]^\top \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}^\top + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]\mathbb{E}[\mathbf{X}_i]^\top + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]^\top \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}^\top + \boldsymbol{\mathcal{A}}_i \mu_{i,0} \mathbf{r}_i^\top \boldsymbol{\mathcal{D}}_i^\top + \boldsymbol{\mathcal{D}}_i \mathbf{r}_i \mu_{i,0}^\top \boldsymbol{\mathcal{A}}_i^\top$. The above performance index, $\mathcal{J}_i(\boldsymbol{\Psi}_i)$, is convex. The proof is similar to that of Proposition 1 in [12] and therefore is omitted. Using the optimal decision variables of the previous task problems, we compute $\mathbb{E}[\boldsymbol{u}_i^\star]$. It is possible to generalize the formulation of the mean as follows:

$$\mathbb{E}\left[\boldsymbol{u}_i^\star\right] = \boldsymbol{\Psi}_i^\star(\mathbb{E}[\mathbf{X}_i] + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]) \quad (24)$$

where $i \geq 1$ and $\mathbb{E}[\overline{\boldsymbol{u}}_0^\star] = \mathbf{0}$. Since $\mathbf{E}_i \in \mathbb{S}_+^{2n_i(N+1)}$, we calculate the value $\boldsymbol{\mathcal{E}}_i$ using the following decomposition, $\boldsymbol{\mathcal{E}}_i \boldsymbol{\mathcal{E}}_i^\top = \mathbf{E}_i$. Finally, we express the performance index as follows:

$$\mathcal{J}_i(\boldsymbol{\Psi}_i) = \|\boldsymbol{\mathcal{R}}_{\boldsymbol{x}_i}^{1/2}(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)\boldsymbol{\mathcal{E}}_i\|_F^2 + \|\boldsymbol{\mathcal{R}}_{\boldsymbol{u}_i}^{1/2}\boldsymbol{\Psi}_i\boldsymbol{\mathcal{E}}_i\|_F^2 \quad (25)$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

D. Mean, Covariance, and Chance Constraints

We represent the constraints in terms of the variable $\boldsymbol{\Psi}_i$. To do that, we express $\mathbb{E}\left[\boldsymbol{x}_i\right]$ and $\boldsymbol{\Sigma}_i$ using $\boldsymbol{\Phi}_i$ as follows:

$$\begin{aligned} \mathbb{E}[\boldsymbol{x}_i] &= (\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)(\mathbb{E}[\mathbf{X}_i] + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]), \quad (26) \\ \boldsymbol{\Sigma}_i &= (\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)\boldsymbol{\Phi}_i(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)^\top, \quad (27) \\ \boldsymbol{\Phi}_i &= \mathbf{E}_i + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]\mathbb{E}[\mathbf{X}_i]^\top + \mathbb{E}[\mathbf{X}_i]\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]^\top \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}^\top \\ &\quad + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]^\top \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}^\top + \mathbb{E}[\mathbf{X}_i]\mathbb{E}[\mathbf{X}_i]^\top. \quad (28) \end{aligned}$$

First, the terminal mean constraint (8b) is written as the equality constraint using (26): $\boldsymbol{\Pi}_i^{\text{eq}}(\boldsymbol{\Psi}_i) = \mathbf{0}$ where

$$\begin{aligned} \boldsymbol{\Pi}_i^{\text{eq}}(\boldsymbol{\Psi}_i) &= \mathbf{S}_N \mathbb{E}\left[\boldsymbol{x}_i\right] - \mu_{i,N}^d \quad (29) \\ &= \mathbf{S}_N(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)(\mathbb{E}[\mathbf{X}_i] + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]) - \mu_{i,N}^d. \end{aligned}$$

Second, the terminal covariance condition (8c) can be formulated as a Positive Semi-Definite (PSD) matrix constraint. In particular, equation (8c) corresponds to the following matrix equality constraint in terms of $\boldsymbol{\Psi}_i$ using (27):

$$\begin{aligned} \Sigma_{i,N}^d &= \mathbf{S}_N \boldsymbol{\Sigma}_i \mathbf{S}_N^\top \\ &= \mathbf{S}_N(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)\boldsymbol{\Phi}_i(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)^\top \mathbf{S}_N^\top. \quad (30) \end{aligned}$$

However, to avoid solving a complicated NLP, we employ a simple convex relaxation technique by replacing the matrix equality constraint with a PSD matrix inequality constraint: $\mathbf{P}_i(\boldsymbol{\Psi}_i) \succeq 0$ where

$$\mathbf{P}_i(\boldsymbol{\Psi}_i) = \left[\begin{array}{cc} \Sigma_{i,N}^d & \boldsymbol{\mathcal{O}} \\ \boldsymbol{\mathcal{O}}^\top & \mathbf{I} \end{array} \right]. \quad (31)$$

and $\boldsymbol{\mathcal{O}}\boldsymbol{\mathcal{O}}^\top = \mathbf{S}_N(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)\boldsymbol{\Phi}_i(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)^\top \mathbf{S}_N^\top$.

We consider the chance constraints in Problem 1 using the Euclidean distance of a point $x_{i,\tau}$ from $\mu_{i,\tau}$. The chance constraints correspond to the following inequalities [22]: $\mathbb{P}(d_i \leq \varepsilon_i) = \mathcal{H}(\varepsilon_i)$ where

$$\begin{aligned} \mathcal{H}(\varepsilon_i) &= \frac{1}{2^{m-1}(m-1)!} \int_0^{\varepsilon_i} y^{2m-1} e^{-\frac{y^2}{2}} dy \\ &= \frac{\varepsilon_i^{2m} e^{-\frac{\varepsilon_i^2}{4}} M_{\frac{(1-m)}{2}, \frac{m}{2}}(-\frac{\varepsilon_i^2}{2})}{2^{m-1}(m-1)!(2m(-\frac{\varepsilon_i^2}{2})^{\frac{(m+1)}{2}})} \end{aligned} \quad (32)$$

an $M_{a,b}(z)$ denotes the Whittacker M function defined as follows:

$$M_{a,b}(z) = e^{-\frac{z}{2}} z^{\frac{b+1}{2}} M\left(b - a + \frac{1}{2}, 1 + 2b, z\right) \tag{33}$$

with the Kummer's confluent hypergeometric function M explained in [23]. Given η_i, we define the upper bound of the Euclidean distance as $\varepsilon_i = \mathcal{H}^{-1}(\eta_i)$, which can be numerically obtained. We change the bound of the Euclidean distance, ε_i, to make the optimization problem feasible as follows: $\varepsilon_i = \max\left(\mathcal{H}^{-1}(\eta_i),\ \varepsilon_{i-1}\right)$. The Euclidean distance of $\boldsymbol{x}_i$ to the mean at time step τ is rewritten as follows:

$$d_{i,\tau}^2 = (x_{i,\tau} - \mu_{i,\tau})^\top (x_{i,\tau} - \mu_{i,\tau}) \leq \varepsilon_i^2. \tag{34}$$

The above inequality constraint is considered as the PSD constraint based on the following Proposition 4.

Proposition 4. *The chance constraints (8d) can be formulated as the following PSD constraint:* $\boldsymbol{\mathcal{L}}_{i,\tau} \succeq 0$, *where*

$$\boldsymbol{\mathcal{L}}_{i,\tau} := \begin{bmatrix} \varepsilon_i^2 & \mathbf{v}_{i,\tau}^\top \\ \mathbf{v}_{i,\tau} & \mathbf{I} \end{bmatrix}$$

where $\mathbf{v}_{i,\tau} = \mathbf{S}_\tau(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)(\boldsymbol{\mathcal{A}}_i(x_{i,0} - \mu_{i,0}) + \boldsymbol{\mathcal{C}}_i\boldsymbol{w} + \boldsymbol{\mathcal{B}}_i^{\mathbf{N}}(\overline{\boldsymbol{u}}_{i-1}^\star - \mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star]))$.

Proof. This proof uses $d_{i,\tau}^2$ in (34), $\boldsymbol{x}_i$ in (14), and $\mathbb{E}[\boldsymbol{x}_i]$ in (26). We express $x_{i,\tau} - \mu_{i,\tau}$ as follows:

$$\begin{aligned} x_{i,\tau} - \mu_{i,\tau} = &\mathbf{S}_\tau(\mathbf{I} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\boldsymbol{\Psi}_i)(\boldsymbol{\mathcal{A}}_i(x_{i,0} - \mu_{i,0}) \\ &+ \boldsymbol{\mathcal{C}}_i\boldsymbol{w} + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}(\overline{\boldsymbol{u}}_{i-1}^\star - \mathbb{E}[\overline{\boldsymbol{u}}_{i-1}^\star])) \end{aligned} \tag{35}$$

Then the inequality constraints become $d_{i,\tau}^2 = \mathbf{v}_{i,\tau}^\top \mathbf{v}_{i,\tau} \leq \varepsilon_i^2$. Therefore, we can express the above inequality constraint as a PSD constraint using Schur complements. □

Considering the above matrix constraints for all time steps $\tau \in [1,\ N-1]$, we define the following matrix including the terminal covariance constraint in (31) as follows:

$$\boldsymbol{\Pi}_i^{\text{PSD}}(\boldsymbol{\Psi}_i) := \text{bdiag}(\mathbf{P}_i(\boldsymbol{\Psi}_i), \boldsymbol{\mathcal{L}}_{i,1}, \cdots, \boldsymbol{\mathcal{L}}_{i,N-1}) \tag{36}$$

where $\boldsymbol{\Pi}_i^{\text{PSD}}(\boldsymbol{\Psi}_i) \succeq 0$ is equivalent to holdng $\mathbf{P}_i(\boldsymbol{\Psi}_i) \succeq 0$ and $\boldsymbol{\mathcal{L}}_{i,\tau} \succeq 0$ for all $\tau \in [1,\ N-1]$ by Theorem 4.3 in [24]. Finally, we formulate Problem 1 as a semi-definite program:

$$\begin{aligned} \min_{\boldsymbol{\Psi}_i} \quad & \|\boldsymbol{\mathcal{R}}_{\boldsymbol{x}_i}^{1/2}(\mathbf{I} + \boldsymbol{\mathcal{B}}_i^{\mathbf{N}}\boldsymbol{\Psi}_i)\boldsymbol{\mathcal{E}}_i\|_F^2 + \|\boldsymbol{\mathcal{R}}_{\boldsymbol{u}_i}^{1/2}\boldsymbol{\Psi}_i\boldsymbol{\mathcal{E}}_i\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\Pi}_i^{\text{PSD}}(\boldsymbol{\Psi}_i) \succeq 0, \\ & \boldsymbol{\Pi}_i^{\text{eq}}(\boldsymbol{\Psi}_i) = \mathbf{0}, \end{aligned} \tag{37}$$

where $\boldsymbol{\Pi}_i^{\text{PSD}}(\boldsymbol{\Psi}_i)$ and $\boldsymbol{\Pi}_i^{\text{eq}}(\boldsymbol{\Psi}_i)$ are defined in (36) and (29), respectively. After obtaining the optimal decision variables for all tasks, we can compute the optimal control input as

$$\boldsymbol{u}^\star = \sum_{i=1}^{m} \boldsymbol{\Psi}_i^\star \left(\mathbf{X}_i + \boldsymbol{\mathcal{B}}_{\mathbf{N}_i}\overline{\boldsymbol{u}}_{i-1}^\star\right). \tag{38}$$

IV. Numerical Simulation

In this section, we show numerical simulation results to validate the proposed hierarchical optimal covariance control approach. A simple mobile manipulator model, i.e., double inverted pendulum on a cart (DIPC), is used to demonstrate the simulations. Also, we run all simulations and computations on a laptop with 3.4 GHz Intel Core i7 and utilize software such as MATLAB, and CVX [25] with Mosek. We compare the behavior of the DIPC model using the proposed optimal covariance control approach against a control method using operational space control. The mass and length of each link are 1 kg and 1 m. The mass of the cart is 5 kg.

TABLE I
Results of the Numerical Simulations

SDP	Iterations	Time (seconds)	Optimal Cost
1st Task	12	809.4	6557.8
2nd Task	10	188.4	14099.5

Results			OSC w/o	OSC w/	HCC w/	Ratio
1st Task	mean	x	4.919	6.261	4.925	
		y	-0.003	-0.033	-0.011	
	error	L2	0.419	1.761	0.425	
	variance	x	-	1.236	0.021	1.70%
		y	-	0.708	0.008	1.13%
2nd Task	mean	x	6.551	7.326	6.580	
	error	L2	0.449	0.326	0.420	
	variance	x	-	0.632	0.019	3.01%

We define two hierarchical tasks. The higher and lower prioritized tasks are defined to control the position of the end-effector ($\boldsymbol{\vartheta}_1 \in \mathbb{R}^2$) and the position of the cart ($\boldsymbol{\vartheta}_2 \in \mathbb{R}$), respectively. The mean and variance at the initial configuration q_0 are $[0,\ 1.0472,\ -0.5236]$ and 0.1. The goal positions of the end effector and cart are defined as $[4.5,\ 0.0]$ m and 7.0 m, respectively. The terminal covariance constraints for these two tasks are specified a priori as $\Sigma_{1,N}^d = \text{diag}(0.02, 0.01)$ and $\Sigma_{2,N}^d = 0.02$ with horizon $N = 30$ and $\Delta t = 0.033$ s. We also introduce a white noise process with $W_\tau = \text{diag}(0.03, 0.03, 0.03)$ that adds to the two pendulum joint positions and the cart position. The reference trajectory for HCC is generated based on the OSC method without noise. We generate references using cubic splines in the task space. We then obtain the joint position and velocity by updating the robot's dynamics with the command computed via OSC as described in [3] (which we call OSC w/o). Also, we employ the same OSC with the noise process (which we call OSC w/) to compare results with our proposed approach. The terminal mean constraints for the two tasks are obtained by running an OSC controller without noise and turn out to be: $\mu_{1,N} = [4.9193, -0.0034]^\top$ and $\mu_{2,N} = 6.5506$. The probabilistic threshold of the chance constraint is specified as $\eta_2 = 0.99$. The weight matrices for the objective functions are $R_{x_1} = \text{diag}(20, 20, 1, 1)$, $R_{x_2} = \text{diag}(20, 1)$, and $R_{u_1} = R_{u_2} = 0.1\mathbf{I}$.

Two sequential SDPs for the defined tasks are successfully solved, with the results shown in Table I. The computation time for solving the secondary SDP is significantly reduced because the dimension of the secondary task state is smaller than the first task. Figure 1 represents the simulation results of OSC and HCC over the defined time horizon. Compared with OSC, HCC steers the DIPC model more accurately towards the goal and with reduced distance from the mean trajectories (i.e. narrower tubes). In addition, the $\mathcal{L}_2$-norm for each task error is given in Table I. The results show that the 1st task error is larger than the 2nd task error when controlling the system using OSC with uncertainty, which means the task hierarchy is violated. By contrast, our proposed HCC attains the task hierarchy, i.e., the 1st task error is less than the 2nd task error as shown in Table I. The terminal states of the tasks are depicted in Figure 2. The terminal states controlled by OSC are considerably scattered from the desired means. However, the final states of the tasks steered by HCC are much closer to the desired mean with smaller variance. We verify

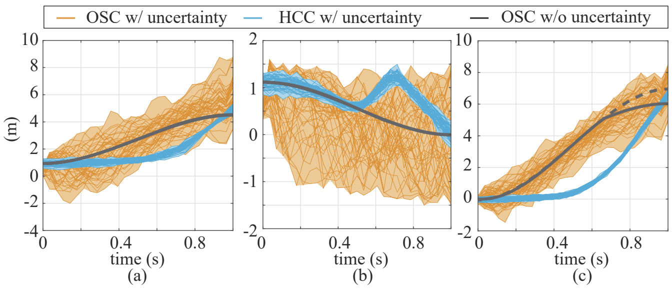

Fig. 1. Execution results of the tasks: (a) x position of the end-effector (b) y position of the end-effector (c) x position of the cart. The dotted lines are the desired values generated by cubic splines. Notice that the terminal error is much smaller for HCC which is the main goal of the method.

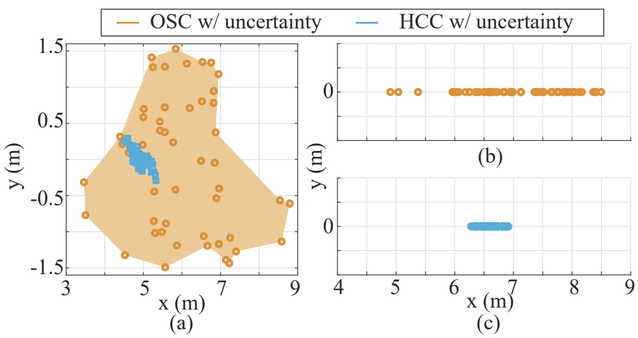

Fig. 2. Terminal states of the tasks: (a) end-effector positions of the terminal states, (b) cart x position of the terminal states controlled by OSC (c) cart x position of the terminal states driven by the proposed covariance controller.

the effectiveness of the proposed approach by comparing the numerical results in Table I. The ratios of the results in Table I are computed as Ratio $= \frac{|\text{variance of HCC w/}|}{|\text{variance of OSC w/}|} \times 100(\%)$. We perform simulations with significant noise levels to make obvious and visible the noise levels in the result figures. In more practical applications, the contributions of this paper still remain by significantly reducing the tracking noise levels.

V. Conclusion

This letter proposes a novel hierarchical optimal covariance controller that executes multiple tasks for robotic systems to ensure accuracy of their terminal states. The proposed approach is used as an alternative to operational space control for robotic systems operating in the presence of uncertainties and noise. One significant limitation of operational space control is that it cannot control the state covariance of the system. Consequently, its performance in steering systems to their desired goals with high accuracy is significantly worsened due to noise and uncertainties. By contrast, our method can directly control the state covariance while considering a task hierarchy often used in high dimensional robots and thus achieve much better performance on achieving the terminal state. In the future, we will extend the proposed approach to nonlinear robotic systems without linearization while enforcing probabilistic constraints using stochastic Differential Dynamic Programming. In addition, HCC will be extended for floating-based and contact-constrained robotic systems with complex nonlinear constraints such as humanoid robots.

References

[1] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[2] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[3] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 505–518, 2005.

[4] F. Romano, A. Del Prete, N. Mansard, and F. Nori, "Prioritized optimal control: A hierarchical differential dynamic programming approach," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2015, pp. 3590–3595.

[5] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1260–1277, 2012.

[6] 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*, 2011, pp. 1085–1090.

[7] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[8] H. Wang, "Adaptive control of robot manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 948–954, 2016.

[9] J. Lee, H. Dallali, M. Jin, D. G. Caldwell, and N. G. Tsagarakis, "Robust and adaptive dynamic controller for fully-actuated robots in operational space under uncertainties," *Autonomous Robots*, vol. 43, no. 4, pp. 1023–1040, 2019.

[10] A. Del Prete and N. Mansard, "Robustness to joint-torque-tracking errors in task-space inverse dynamics," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1091–1105, 2016.

[11] E. Bakolas, "Optimal covariance control for discrete-time stochastic linear systems subject to constraints," in *Proceedings of the IEEE Conference on Decision and Control*, 2016, pp. 1153–1158.

[12] ——, "Finite-horizon covariance control for discrete-time stochastic linear systems subject to input constraints," *Automatica*, vol. 91, pp. 61–68, 2018.

[13] K. Okamoto and P. Tsiotras, "Input hard constrained optimal covariance steering," in *Proceedings of the IEEE Conference on Decision and Control*, 2019, pp. 3497–3502.

[14] K. Okamoto, M. Goldshtein, and P. Tsiotras, "Optimal covariance control for stochastic systems under chance constraints," *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 266–271, 2018.

[15] J. A. Paulson and A. Mesbah, "An efficient method for stochastic optimal control with joint chance constraints for nonlinear systems," *Int. J. Robust Nonlinear Control*, vol. 29, no. 15, pp. 5017–5037, 2019.

[16] J. Ridderhof, K. Okamoto, and P. Tsiotras, "Nonlinear uncertainty control with iterative covariance steering," in *Proceedings of the IEEE Conference on Decision and Control*, 2019, pp. 3484–3490.

[17] E. Bakolas and A. Tsolovikos, "Greedy finite-horizon covariance steering for discrete-time stochastic nonlinear systems based on the unscented transform," in *Proceedings of the American Control Conference*, 2020, pp. 3595–3600.

[18] Z. Yi, Z. Cao, E. Theodorou, and Y. Chen, "Nonlinear covariance control via differential dynamic programming," in *Proceedings of the American Control Conference*, 2020, pp. 3571–3576.

[19] K. Okamoto and P. Tsiotras, "Optimal stochastic vehicle path planning using covariance steering," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2276–2281, 2019.

[20] N. Chohan, M. A. Nazari, H. Wymeersch, and T. Charalambous, "Robust trajectory planning of autonomous vehicles at intersections with communication impairments," in *Annual Allerton Conference*, 2019, pp. 832–839.

[21] A. D. Saravanos, A. G. Tsolovikos, E. Bakolas, and E. A. Theodorou, "Distributed covariance steering with consensus admm for stochastic multi-agent systems," in *Robotics: Science and Systems*, 2021.

[22] M. Bensimhoun, "N-dimensional cumulative function, and other useful facts about gaussians and normal densities," *Jerusalem, Israel, Tech. Rep*, pp. 1–8, 2009.

[23] M. Abramowitz, I. A. Stegun, and R. H. Romer, "Handbook of mathematical functions with formulas, graphs, and mathematical tables," 1988.

[24] J. Gallier *et al.*, "The Schur complement and symmetric positive semidefinite (and definite) matrices," *Penn Engineering*, pp. 1–12, 2010.

[25] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.