Robust Co-Design of Canonical Underactuated Systems for Increased Certifiable Stability

Federico Girlanda[1,2], Lasse Shala[1], Shivesh Kumar[1,3] and Frank Kirchner[1,4]

Abstract—Optimal behaviours of a system to perform a specific task can be achieved by leveraging the coupling between trajectory optimization, stabilization, and design optimization. This approach is particularly advantageous for underactuated systems, which are systems that have fewer actuators than degrees of freedom and thus require for more elaborate control systems. This paper proposes a novel co-design algorithm, namely Robust Trajectory Control with Design optimization (RTC-D). An inner optimization layer (RTC) simultaneously performs direct transcription (DIRTRAN) to find a nominal trajectory while computing optimal hyperparameters for a stabilizing time-varying linear quadratic regulator (TVLQR). RTC-D augments RTC with a design optimization layer, maximizing the system's robustness through a time-varying Lyapunov-based region of attraction (ROA) analysis. This analysis provides a formal guarantee of stability for a set of off-nominal states. The proposed algorithm has been tested on two different underactuated systems: the torque-limited simple pendulum and the cart-pole. Extensive simulations of off-nominal initial conditions demonstrate improved robustness, while real-system experiments show increased insensitivity to torque disturbances.

I. INTRODUCTION

Different living organisms have evolved diverse locomotion strategies and physical characteristics to adapt to their unique environments. Similarly, for underactuated systems a combined optimization of structural parameters and motion control is important to effectively accomplish the desired tasks. A design optimization tunes the system's hardware parameters, such as link lengths or the center of mass position, of a system to allow for executing the described task with the desired behaviour. For trajectory tracking control, reaching the goal requires to search for a motion trajectory and to compute the control policy that permits the closed-loop trajectory following. Typically, the best trajectory is obtained via a trajectory optimization step, while the control input is computed by a specifically defined controller. A traditional approach to find the best trade-off between mechanical design and motion planning is to iterate between

[1]Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI GmbH), Bremen, Germany.

[2]Department of Information Engineering, University of Padova, Italy. federico.girlanda@outlook.com

[3]Dynamics Division, Department of Mechanics & Maritime Sciences, Chalmers University of Technology, Gotheburg, Sweden.

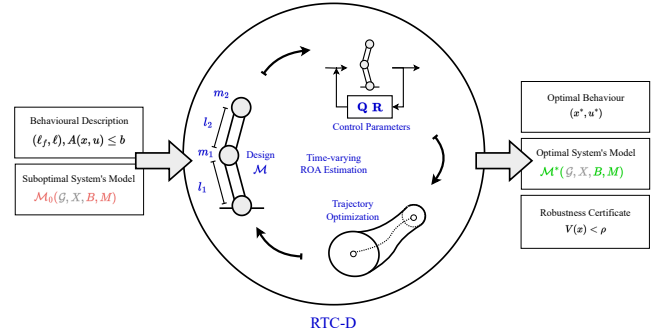[4]Faculty of Mathematics and Computer Science, University of Bremen, Bremen, Germany.

Fig. 1. Robust co-optimization for the optimal fitness of the desired motion.

the two processes [1]. Instead, concurrent design (co-design [2]) aims to automate this process by numerically optimizing both the motion and design parameters, a strategy that has demonstrated superior results [3]–[9]. An example of *gradient-based co-design* methods have been proposed in [9]. Here, the optimization for robot design and motion planning is handled on two levels. On the lower level, an efficient state-of-the-art constrained motion planner is implemented, which is continuously differentiable. This last property is then exploited on the higher level where the derivative of the motion is embedded into a nonlinear program. Another approach is the *gradient-free co-design*. For instance, Ha et. al. [7] proposed a framework that had successfully optimized designs for legged robots performing tasks that include jumping, walking, and climbing up a step. Similarly, Fadini et al. [8] introduced a bi-level optimization scheme that finds the optimal actuator properties of a monoped in order to improve energy efficiency. Both approaches are two-staged and make use of variants of Monte-Carlo sampling to find candidate robot designs, which are subsequently evaluated through a motion planning stage. The covariance matrix adaption evolution strategy (CMA-ES) [10] is a commonly employed optimization strategy in this scenario. It uses a Gaussian prior on candidate design parameters and estimates a covariance matrix needed for the following sampling steps. Another popular gradient free algorithm is Nelder-Mead which has been used in the context of co-design of surgical robots [11]. Recent studies in the literature are incorporating non-idealities, such as measurement noise and parameter variations, into co-optimization processes to reduce discrepancies between theoretical predictions and real-world performance [12]–[15]. For instance, Manchester and Kuindersma [13] derived a tractable robust optimization algorithm, namely DIRTREL, that combines direct transcription

with linear-quadratic control design to reason about closed-loop responses to disturbances. Fadini [14] introduced a simulation-based cost in a bi-level co-optimization algorithm that aims to take into account a given set of disturbances. The concept of robustness has also been explored through ROA analysis [16], enabling the reasoning about stabilizable state sets. A formal mathematical certificate of stabilizability can also be provided. This methodology has mostly been applied for stability analysis but it's use in space-filling algorithms like LQR-trees has obtained promising results [17]–[19]. ROA estimation has already proven valuable in co-design frameworks. In [15], where the volume of the estimated region was used as an optimization cost for the up-right stabilization of an Acrobot system, resulting in a more robust closed-loop system against off-nominal initial states. See [20] for an extension of [15] to pendubot system. Additionaly, a bi-level optimization scheme was introduced by [14] to compute a robust energy efficient design and trajectory for a jumping monoped, considering a simulation-based robustness cost to enhance result's robustness.

This paper proposes a novel two layered *gradient-free co-design* algorithm, namely RTC-D. The process involves co-optimization among the cost matrices of a TVLQR controller, the desired trajectory provided by DIRTRAN, and the system's design parameters. It aims to improve the real-world applicability of the combined optimization result by leveraging the volume of an estimated time-varying ROA. To the best knowledge of the authors this metric has not been studied in this context. We employ the CMA-ES optimization strategy to determine the optimal decision variables. Our approach has been rigorously tested on two different canonical underactuated systems: the torque-limited simple pendulum and a cart-pole. These systems are inherently underactuated, demanding a robust control approach. The specific case study involves the swing-up problem, consisting in stabilizing the pole in it's up-right position. We validate our results through extensive simulations of the system's closed loop dynamics, assessing the stabilizability of off-nominal initial conditions. Furthermore, a real-world experiment has been implemented to test a scenario subject to input torque noise.

Organization: Section II introduces a mathematical description of the co-design problem and its components including trajectory optimization, TVLQR control and ROA estimation. Following this, Section III presents our proposed robust co-design methodology for canonical underactuated systems. Subsequently, in Section IV we present results from applying this methodology to two different canonical systems namely torque limited pendulum and cart-pole system. Section V concludes the paper.

II. MATHEMATICAL BACKGROUND

A. Co-Design Problem Description

A mechanical system with n indepedent degrees of freedom can be defined by a system's description $\mathcal{M}(\mathcal{G}, \mathsf{X}, \mathsf{B}, \mathsf{M})$ where $\mathcal{G}$ is its topological graph, $\mathsf{X} = \text{diag}(\mathbf{X}_1, \ldots, \mathbf{X}_n) \in \mathbb{R}^{6n \times n}$ is the system matrix of body-fixed joint axis screw coordinates with $\mathbf{X}_i \in \mathbb{R}^6$, $\mathsf{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_n\}$

is the set of body frames with $\mathbf{B}_i \in SE(3)$ and $\mathsf{M} = \text{diag}(\mathbf{M}_1, \ldots, \mathbf{M}_n) \in \mathbb{R}^{6n \times 6n}$ is the matrix of mass-inertia matrices of every moving body ($\mathbf{M}_i \in \mathbb{R}^{6 \times 6}$) in $\mathcal{G}$. Given $\mathcal{M}(\mathcal{G}, \mathsf{X}, \mathsf{B}, \mathsf{M})$, it is straight forward to develop equations of motion (EOM) for any mechanical system in closed-form [21], [22]. The EOM can be used to develop a 1st order ODE description of dynamics in terms of state-space $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{2n}$ (with $\mathbf{q}$ and $\dot{\mathbf{q}}$ denoting the generalized position and velocity coordinates respectively) and action-space $\mathbf{u} \in \mathbb{R}^p$ in the form $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$. Given an initial state description, $\mathbf{x}_0$, and the mechanical system's model $\mathcal{M}$, a desired behavior can be achieved by optimizing its corresponding cost model, which is composed of a final cost model $l_f(\mathbf{x}_f)$ and running cost model $l(\mathbf{x}, \mathbf{u})$, under a given set of constraints imposed by the robot and the environment. This typically entails solving an optimal control problem (OCP). In a typical OCP, the design parameters of $\mathcal{M}(\mathcal{G}, \mathsf{X}, \mathsf{B}, \mathsf{M})$ are assumed to be constant.

We formulate co-design as a mathematical optimization problem with the decision variables including the minimal independent set of design space variables in $\mathcal{M}$ (while fixing the robot topology $\mathcal{G}$ and joint axes screws X), state trajectory $\mathbf{x}(t)$, and control policy $\mathbf{u}(\mathbf{x}(t))$. The overall cost function of the co-design problem (ℓ_c, ℓ_{cf}) may include the behavior cost functions l and l_f as well as robustness of the control policy (e.g. volume of the region of attraction $\mathcal{B}$ of the controller). The problem can be mathematically written as the following.

$$
\begin{aligned}
\min_{\mathcal{M}, \mathbf{x}(\cdot), \mathbf{u}(\cdot)} \quad & \ell_{cf}(\mathcal{B}_f, l_f) + \int_{t_0}^{t_f} \ell_c(\mathcal{B}, l) dt \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = f(\mathcal{M}, \mathbf{x}(t), \mathbf{u}(t)), \quad \forall t \in [t_0, t_f] \\
& \mathbf{x}(t_0) = \mathbf{x}_0, \mathbf{x}(t_0) = \mathbf{x}_N \\
& \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} \\
& \mathcal{M}_{\min} \leq \mathcal{M} \leq \mathcal{M}_{\max}
\end{aligned}
\tag{1}
$$

As expected, this is a complex mathematical optimization problem with a very high dimensional space of decision variables. Hence, deliberation should go into decomposing the problem in a computationally tractable way.

B. Trajectory Optimization

Trajectory optimization is a way of solving OCP where one solves for an open loop state and effort trajectory while optimizing the behavior cost model (l_f, l). Direct methods discretize the trajectory optimization problem directly, converting it into a constrained parameter optimization problem [23]. This discretization is also referred as *transcription*. Transcription and the use of $\mathbf{x}[\cdot], \mathbf{u}[\cdot]$ as decision variables leads to the so-called direct transcription (DIRTRAN). The time discretization is done for N points, the *knot-points*.

$$
\begin{aligned}
\min_{\mathbf{x}[\cdot], \mathbf{u}[\cdot]} \quad & l_f(\mathbf{x}[N]) + \sum_{k=0}^{N-1} l(\mathbf{x}[k], \mathbf{u}[k]) \\
\text{subject to} \quad & \mathbf{x}[k+1] = f_d(\mathbf{x}[k], \mathbf{u}[k]), \quad \forall k \in [0, N-1] \\
& \mathbf{x}[0] = \mathbf{x}_0 \\
& \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}
\end{aligned}
$$

The resulting open-loop trajectory $(\mathbf{x}^\star, \mathbf{u}^\star)$, usually referred as *nominal trajectory*, has to satisfy the discrete dynamic's f_d constraints while minimizing the standard additive-cost optimal control objective $l = (\mathbf{x}[k] - \mathbf{x}_N)^T \mathbf{Q}_T (\mathbf{x}[k] - \mathbf{x}_N) + \mathbf{u}^T \mathbf{R}_T \mathbf{u}$ and final cost $l_f = (\mathbf{x}[k] - \mathbf{x}_N)^T \mathbf{Q}_{Tf}(\mathbf{x}[k] - \mathbf{x}_N)$ where $(\mathbf{Q}_T, \mathbf{Q}_{Tf})$ denote the running and final state regularization cost matrices respectively and $\mathbf{R}_T$ denotes the effort minimization cost matrix to reach the goal state $\mathbf{x}_N \in \mathbb{R}^{2n}$. This optimization problem can be solved using commercial sequential-quadratic programming (SQP) solvers, such as SNOPT [24]. The set of hyperparameters of DIRTRAN is defined as the set $\mathcal{H}_T = \{\mathbf{Q}_T, \mathbf{R}_T, \mathbf{Q}_{Tf}\}$.

C. Trajectory Stabilization

The stabilization of a nominal trajectory defined in a finite time interval $t \in [t_0, t_f]$ can be achieved via *Time-Varying LQR (TVLQR)* [23]. TVLQR aims to minimize the error coordinates $\bar{\mathbf{x}} = (\mathbf{x} - \mathbf{x}^*)$ and $\bar{\mathbf{u}} = (\mathbf{u} - \mathbf{u}^*)$, where * denote states of the nominal trajectory obtained from trajectory optimization step. For this, a time-varying linearization using a Taylor series approximation is performed, resulting in a time-varying linear system in the error coordinates:

$$\dot{\mathbf{x}} = \mathbf{A}(t)\bar{\mathbf{x}}(t) - \mathbf{B}(t)\bar{\mathbf{u}}(t) \qquad (2)$$

The quadratic cost function is defined as:

$$J = \bar{\mathbf{x}}^T(t)\mathbf{Q}_{Cf}\bar{\mathbf{x}}(t) + \int_0^{t_f} \left(\bar{\mathbf{x}}^T(t)\mathbf{Q}_C\bar{\mathbf{x}}(t) + \bar{\mathbf{u}}^T(t)\mathbf{R}_C\bar{\mathbf{u}}(t) \right) dt$$

where $\mathbf{Q}_C = \mathbf{Q}_C^T \succeq 0$, $\mathbf{Q}_{Cf} = \mathbf{Q}_{Cf}^T \succeq 0$ and $\mathbf{R}_C = \mathbf{R}_C^T \succ 0$. The optimal cost-to-go can be written as a time-varying quadratic term and the controller gain $\mathbf{K}(t)$ be found by solving the differential Riccati Equation (DRE). The optimal tracking control policy is given by:

$$\begin{aligned}
\pi(\mathbf{x}, \mathbf{u}) &= \mathbf{u}^\star(t) - \mathbf{K}(t)\bar{\mathbf{x}} \\
&= \mathbf{u}^\star(t) - \mathbf{R}_C^{-1}\mathbf{B}^T\mathbf{S}(t)(\mathbf{x} - \mathbf{x}^\star(t))
\end{aligned}$$

The solution of the DRE provides the optimal cost-to-go $J^\star(\mathbf{x}, t) = \bar{\mathbf{x}}^T\mathbf{S}(t)\bar{\mathbf{x}}$, a function that returns the accumulated cost when running the optimal controller from any initial state to the goal. The set of hyperparameters of TVLQR controller is defined as the set $\mathcal{H}_C = \{\mathbf{Q}_C, \mathbf{R}_C, \mathbf{Q}_{Cf}\}$.

D. Region of Attraction Estimation

The region of attraction of a closed-loop system is defined as the set of states around a fixed point that are guaranteed to be stabilized by the controller. Most often, it is difficult, if not impossible, to determine analytically and an estimation has to be considered. The estimation methods which are available in the literature can be classified in two main categories: Lyapunov-based and Non-Lyapunov methods [25]. Lyapunov analysis can be employed to obtain a formal guarantee of stability for an inner estimate of the ROA. The Lyapunov function V is a generic function of the state such that

$$V(\mathbf{x}^\star) = 0 \quad \text{and} \quad V(\mathbf{x}) > 0, \ \dot{V}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \in \mathcal{X}\backslash\{\mathbf{x}^\star\}$$

If the above conditions hold for the fixed point $\mathbf{x}^\star$, then it is globally asymptotically stable. Sublevel sets $\mathcal{B} = \{\mathbf{x} | V(\mathbf{x}) < \rho\}$, $\rho > 0$, of a Lyapunov function are then used as

approximations of the region of attraction [26]. For closed loop dynamics under LQR control, the cost-to-go matrix $\mathbf{S}$ can be used to construct a suitable quadratic Lyapunov function. Setting $V = \bar{\mathbf{x}}^T\mathbf{S}\bar{\mathbf{x}}$ makes the estimation to be a matter of finding a suitable ρ. In the time-varying case each knot point of the nominal trajectory is associated to a ROA, hence a vector $\boldsymbol{\rho} \in \mathcal{R}_+^N$ has to be estimated. In this case, the resulting ROA is also called *funnel* [16]. Two methods for computing the funnel are described below:

1) Sum of Squares (SOS): In the case of polynomial dynamics, SOS optimization can be used to express the estimation task as an optimization problem. Polynomial, non-linear dynamics can always be obtained via Taylor approximation. The idea is that every initial state state taken in the estimated ROA of the knot point k must end up, following the system's dynamics, inside the ROA estimate associated to $k+1$. An estimation procedure for each couple of knot points is then considered. Each estimation process is a bilinear alternation between two optimizations [27]:

Multiplier Step:

$$\begin{aligned}
&\max_{\gamma_i, \lambda_i} && \gamma_i \\
&\text{subject to} && -(\dot{V} - \dot{\boldsymbol{\rho}}_i) + \lambda_i(V - \boldsymbol{\rho}_i) - \gamma_i \quad is \ SOS \\
& && \lambda_i \quad is \ SOS \\
& && \gamma_i > 0
\end{aligned}$$

Rho Step:

$$\begin{aligned}
&\max_{\boldsymbol{\rho}_i} && \boldsymbol{\rho}_i \\
&\text{subject to} && -(\dot{V} - \dot{\boldsymbol{\rho}}_i) + \lambda_i(V - \boldsymbol{\rho}_i) \quad is \ SOS \\
& && \lambda_i \quad is \ SOS \\
& && \boldsymbol{\rho}_i > 0
\end{aligned}$$

Given a meaningful initial guess of $\boldsymbol{\rho}$, these two steps have to be alternatively solved until a convergence condition. This condition can be, for instance, related to the overall magnitude of $\boldsymbol{\rho}$. After a time-invariant $\boldsymbol{\rho}_N$ has been determined, a bilinear alternation for each couple of knot points has to be solved going backwards for $k \in [0, N-1]$. By means of the S-procedure [28], we are constraining the Lyapunov function to change slower than $\boldsymbol{\rho}$ for all the states at the boundary of the ROA. This is sufficient for the dynamics to completely stay inside the funnel [23].

2) Simulation based: The methodology proposed in [19] is based on simulations and falsifications. Initially, a funnel hypothesis is proposed, then each trajectory knot point is verified through sampling and simulations. Given a desired trajectory $(\mathbf{x}^*, \mathbf{u}^*)$ defined for N knot points, the initial ROA guess k is set as an open ball with radius $\rho_k > 0$, centered at $\bar{\mathbf{x}}_k$:

$$\mathcal{B}(\bar{\mathbf{x}}_k, \rho_k) := \{\mathbf{x}_k : V(\bar{\mathbf{x}}_n) < \rho_n\}.$$

Assume that a stabilizable region $\mathcal{B}_N$ around the goal state is given. For each node a set of samples is obtained and simulated through the end of the nominal trajectory. A simulation is considered successful if the sampled initial condition has been stabilized to the goal region, $\mathbf{x}_N \in \mathcal{B}_N$.

If so, the verified ROA remains valid. If the simulation fails the initial region guess and the following ones are shrunk accordingly to the following rule:

$$\rho_{i,new} = \min(V(\bar{\mathbf{x}}_i), \rho_{i,old}), \ \forall i \in [k, N-1].$$

III. PROPOSED CO-DESIGN FRAMEWORK

In order to solve the problem described in Section II-A, a bi-level co-optimization algorithm has been conceived which aims to seek shared optimality between design optimization, trajectory optimization and trajectory stabilization with the aim of optimizing the final robustness of the control scheme. The trajectory optimization and stabilization tasks have been solved through DIRTRAN and TVLQR control, respectively. This allows the introduction of explicit constraints on the motion variables, which is not possible in Differential Dynamic Programming (DDP) where the constraints are introduced as weighted costs in the optimization.
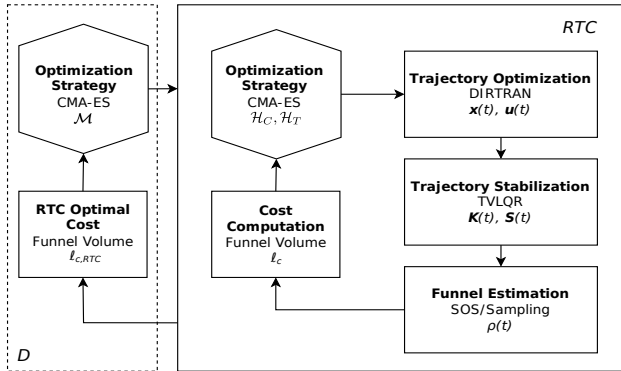

Fig. 2. RTC-D algorithm scheme.

These two choices are also compatible with the funnel estimation process. The volume of the time-varying ROA was chosen as a metric for the system's robustness and calculated as the sum of the volumes of the ROAs at each knot point. The gradient-free CMA-ES method has been chosen to optimize the decision variables. This method operates by adapting the covariance matrix of the search distribution for a population of candidate solutions, which are referred to as individuals. CMA-ES does this efficiently with limited evaluations, giving preference to individuals that perform better during the optimization process.

A. RTC

The inner-layer of the proposed algorithm, namely RTC, is concurrently optimizing the nominal trajectory and the stabilizing controller, for fixed design parameters $\mathcal{M}$. This has been implemented by considering the hyperparameters of the trajectory optimization $\mathcal{H}_T$ and stabilizing controller $\mathcal{H}_C$ as decision variables for the optimization problem. We have assumed the decision variables to stay inside some reasonable bounds to restrict the search space. At each iteration, the optimizer proposes a new set of hyperparameter cost matrices. They are used to compute a new nominal

trajectory $(\mathbf{x}^*, \mathbf{u}^*)$ via DIRTRAN method as explained in Section II-B and a stabilizing control policy π^* via TVLQR method as explained in Section II-C. These quantities are necessary to compute the ROA of the stabilizing controller. Finally, the volume ℓ_c of the estimated funnel is used by the CMA-ES strategy to weigh each initially proposed set of costs and provide better $\mathcal{H}_T, \mathcal{H}_C$ such that the volume of the resulting ROA is maximized. We iterate this process until a predefined maximum number of evaluations of the objective function.

B. RTC-D

To complete the solution of our co-design problem, an outer CMA-ES optimization layer, namely D, with the capability to vary the design parameters in $\mathcal{M}$ has been added. Specifically, we consider the minimal independent set of design space variables in $\mathcal{M}$ (while fixing the robot topology $\mathcal{G}$ and joint axes screws X). Also, in this case the decision variables are constrained inside some reasonable bounds which may be inspired from manufacturing constraints or availability of off-the-shelf parts. Compared to RTC, the resulting process is more computationally expensive due to the increased number of optimization layers. On the other hand, the solver has now more power to improve the objective function. The optimization strategy proposes a new set of design parameters for each iteration. The RTC layer fixes them and then determines an optimal stabilized trajectory along with the related funnel volume. A different set of design parameters is then provided by the D CMA-ES optimizer to perform the same operation. Eventually, designs that maximize the ROA volume computed in the inner layer will be preferred by the solver. Similar to RTC, we iterate this process until a predefined maximum number of objective function evaluations.

IV. RESULTS AND DISCUSSION

In this section the optimization results and their verification is presented[1]. Two underactuated systems have been considered for simulated and experimental verification: torque-limited simple pendulum [29] and cart-pole [30] (Figure 3). The common task is to solve the well-known swing-up problem. In particular, the optimization parameters were inspired by the real hardware available in the *Underactuated Robotics Lab* of the DFKI Robotics Innovation Center. The computations related to the simple pendulum have been handled by a having 2x 8-core Xeon E5-2630 v3 (Haswell) machine with 128GB of ECC RAM. The computations for the cart-pole have been performed by a 2-core 4-thread 2.70 GHz Intel(R) Core(TM) i7-7500U computer with 8 Gb of RAM. A maximum of 3 parallelized cost computations have been considered for both the systems.

The controller and trajectory cost matrices have been assumed to be the same i.e. $\mathbf{Q}_T = \mathbf{Q}_C = \mathbf{Q} = \mathrm{diag}(Q_{11}, \ldots, Q_{2n2n})$ and $\mathbf{R}_T = \mathbf{R}_C = \mathbf{R} =$
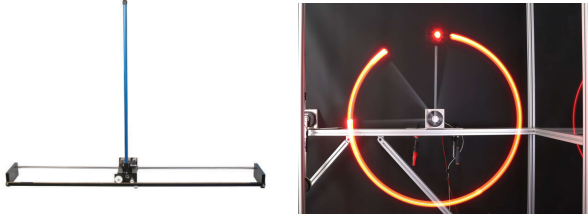
Fig. 3. Experimental Systems: Cart Pole (left) and Simple Pendulum (right)

$\mathrm{diag}(R_{11}, \ldots, R_{pp})$ during the optimization in order to decrease the overall number of decision variables and hence to reduce the computational complexity. For simplicity, the final cost matrices for the trajectory optimization and TVLQR controller were fixed as $\mathbf{Q}_{Tf} = \mathbf{Q}_{Cf} = 100\mathbf{I}_{2n \times 2n}$.

A. Linear Inverted Pendulum

We consider an off-the-shelf Quanser linear inverted pendulum (cart-pole) with $n = 2$ DOFs and $p = 1$ actuated DOF for testing the RTC algorithm. Since, it is a commercial system, we are only interested in optimizing the swing-up controller. We additionally fix velocity related cost terms $Q_{33} = Q_{44} = 1$ to reduced the decision variables. We let a subset of hyperparameters including Q_{11}, Q_{22} and R_{11} to vary during the co-design process. As can be seen in Table I, the implemented algorithm resulted in an improvement of the funnel volume by a factor of 2.17. In Figure 4 (left), a comparison between initial and final optimized funnels has been shown. A projection of the estimated ROA onto reduced state-spaces have been considered. Each ellipse represents the ROA associated with a knot point along the related optimal trajectory. State-space coverage improvement is evident from the overlap of the initial and final funnels. The ROA estimation has been implemented through the simulation-based method introduced in Section II-D with a minor modification in the shrinking policy. In our implementation, just one region is shrunk at a time. The final estimated region is associated with a certificate of stability that depends on the number of simulations performed during the estimation. The verification of such a guarantee has been implemented via extensive sampling and simulation. Off-nominal initial states inside the RTC funnels are almost always stabilized. Also an experimental verification on the real system has been implemented. An impulsive torque disturbance has been introduced to check the controller's robustness. As shown in Figure 5 (left), the optimized set-up is able to reject the disturbance that cause the initial controller, referred to as DIRTRAN, to fail.

B. Simple pendulum

We consider a torque-limited simple pendulum with $n = p = 1$ for testing the full RTC-D algorithm. The limit in the input torque renders it as one of the simplest underactuated system. Firstly, the RTC optimization is implemented with fixed design parameters, resulting in a funnel volume increase of 3.18 times as noted in Table I and visually shown in Figure 4 (right). Then, the mass m and the link length l have

been chosen as decision variables for the design optimization in the outer loop. As we are aware that, in the case of this system, reducing m and l makes the torque limit less penalizing for the swing-up, we are expecting our algorithm to push these decision variables towards their lower limit. Applying RTC-D resulted in an optimized ROA volume that has increased to almost 4 times the initial value as reported in Table I. This improvement is shown in the funnel plots of Figure 4 (right). RTC-D execution was stopped after almost 1500 cost function evaluations with a final execution time of 3 hours. Almost 90% of the computational cost can be attributed to the ROA estimation phase. As expected, the best design is the one that exhibits the least underactuation. Both values of m and l tend to decrease, which convince us about the sanity of the choices made by the outer loop of the codesign process. For this system, the ROA was computed via SOS, and the resulting region is coming with a formal guarantee of stability. Firstly, the verification of such a certificate was implemented in simulation. The simulated system's closed-loop dynamics has been tested with 1000 off-nominal initial states sampled inside the ROA. The controller was able to stabilize all of the unexpected initial states by bringing the state evolution to join the nominal trajectory. To assess the real world applicability of RTC-D, an experiment has also been implemented on the physical system. The initial and the optimized versions have been tested in accomplishing the swing-up task while being affected by an impulsive torque disturbance. The experimental trajectories have been plotted with respect to the RTC-D funnel in Figure 5 (right). Only the initial trajectory is not able to recover the desired behaviour, i.e. to come back inside the funnel.

V. CONCLUSION AND OUTLOOK

This paper presented a novel gradient free co-design algorithm, namely RTC-D, which provides optimal design parameters, nominal trajectory and stabilizing controller by taking into account their mutual dependencies. Our approach has been tested on two canonical underactuated systems: a torque-limited simple pendulum and a cart-pole. The results demonstrated an increased volume of ROA, and were validated both in simulation and through real-world experiments. In contrast to the approach described in [14], the inner optimization layer (RTC) is implemented through DIRTRAN which allows for the introduction of explicit state and input constraints and TVLQR which allows for the incorporation of robustness analysis through time-varying ROA estimation. This approach offers an intuitive perspective and can potentially provide a stability guarantee for off-nominal states around the optimal trajectory, especially when employing the SOS-based estimation method. It's worth noting that this estimation method was not implemented in [15], where the focus was on considering the volume of a sample-based region around the desired fixed point. During the empirical analysis of the optimization process, we observed that design optimization consistently mitigates underactuation. However, when dealing with the Cart-pole system, we encountered challenges due to the increased
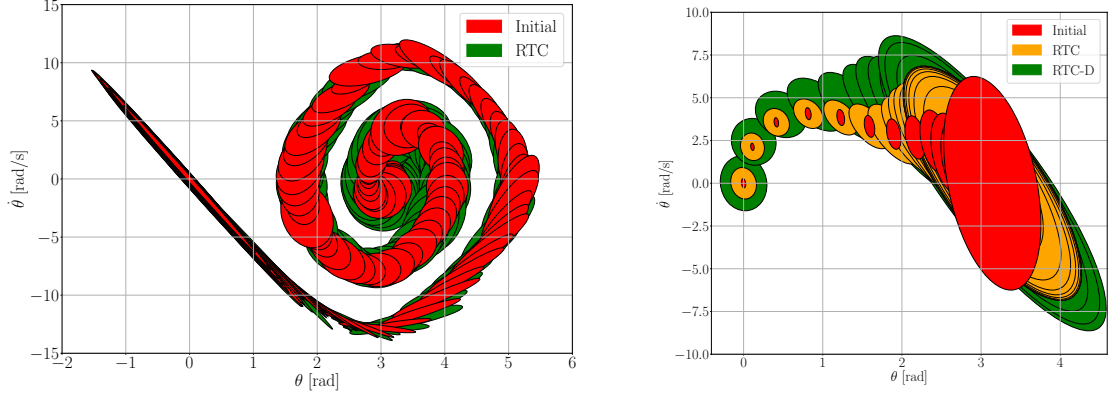
Fig. 4. Funnel volume increasing due to RTC for Cart-pole (left) and comparison between RTC and RTC-D optimization for Simple pendulum (right).
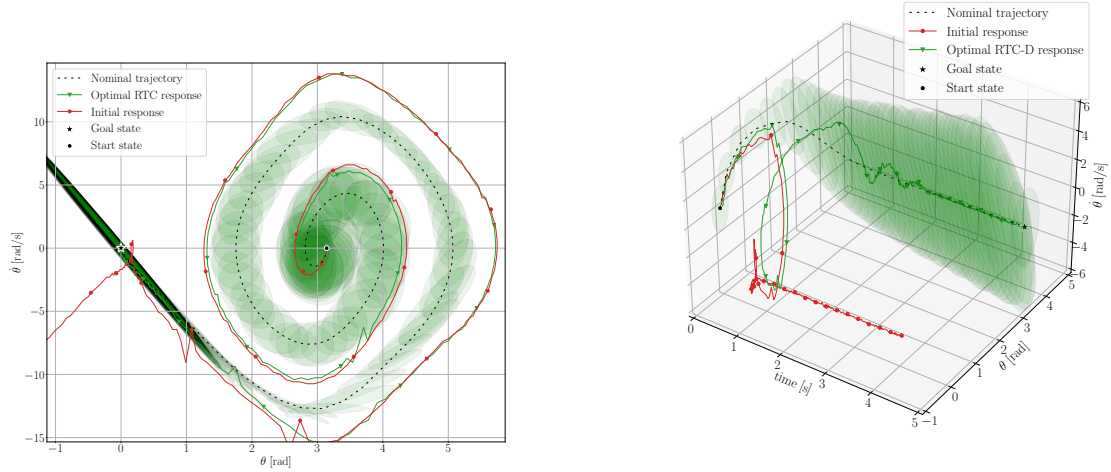

Fig. 5. Experimental verification of stability guarantee (green funnel) given by the RTC for Cart-pole (left) and RTC-D for Simple pendulum (right). The optimal configuration (RTC and RTC-D) manages to achieve the desired final stabilization where the initial one does not.

TABLE I
RTC-D OPTIMIZATION RESULTS

System	Initial Decision Variables					Optimal Decision Variables					Volume	
	m (kg)	l (m)	Q_{11}	Q_{22}	R_{11}	m (kg)	l (m)	Q_{11}	Q_{22}	R_{11}	ℓ_c^*/ℓ_c^0	Time (h)
Cart-pole (RTC)	0.23	0.18	10.00	10.00	10.00	-	-	10.91	12.57	6.09	2.17	3.55
Pendulum (RTC)	0.70	0.40	10.00	1.00	0.10	-	-	9.50	1.20	1.64	3.18	0.40
Pendulum (RTC-D)	0.70	0.40	10.00	1.00	0.10	0.61	0.40	9.98	1.00	3.39	3.90	3.00

system complexity. The dimensional increase affects the computational cost of ROA estimation and, in the case of SOS-based estimation, its functionality. In our opinion, a deeper analysis on handling this problem should follow this work. As an alternative to SOS-based estimation, we used a simulation-based method. While it doesn't provide a strict formal stabilizability guarantee, it yields a viable cost for the co-optimization process. Another alternative would be to use analytical region of attraction estimation as proposed in [31],

whenever available. We then would approach the increased computational complexity with a C++ implementation and an increased use of parallelization. We also plan to extend this formulation to address the co-design of some simple hybrid dynamical systems such as AcroMonk [32], hopping leg [33] and RicMonk [34]. As a future development, extending the RTC-D scheme to accommodate multiple tasks simultaneously could offer a powerful solution for more complex robotic systems such as humanoids [35].

References

[1] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq - a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011. [Online]. Available: https://doi.org/10.1177/0959651811402275

[2] Q. Li, W. Zhang, and L. Chen, "Design for control-a concurrent engineering approach for mechatronic systems design," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 2, pp. 161–169, 2001.

[3] J. Allison and S. Nazari, "Combined plant and controller design using decomposition-based design optimization and the minimum principle," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 1, 01 2010.

[4] J. T. Allison, T. Guo, and Z. Han, "Co-Design of an Active Suspension Using Simultaneous Dynamic Optimization," *Journal of Mechanical Design*, vol. 136, no. 8, 06 2014, 081003. [Online]. Available: https://doi.org/10.1115/1.4027335

[5] G. Bravo-Palacios, A. D. Prete, and P. M. Wensing, "One robot for many tasks: Versatile co-design through stochastic programming," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1680–1687, 2020.

[6] G. Grandesso, G. Bravo-Palacios, P. Wensing, M. Fontana, and A. del Prete, "Exploring the limits of a hybrid actuation system through Co-Design - Technical Report," University of Trento," Technical Report, June 2020. [Online]. Available: https://hal.science/hal-02737086

[7] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Task-based limb optimization for legged robots," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2062–2068.

[8] G. Fadini, T. Flayols, A. Del Prete, N. Mansard, and P. Souères, "Computational design of energy-efficient legged robots: Optimizing for size and actuators," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9898–9904.

[9] T. Dinev, C. Mastalli, V. Ivan, S. Tonneau, and S. Vijayakumar, "A versatile co-design approach for dynamic legged robots," 2022.

[10] N. Hansen, yoshihikoueno, ARF1, K. Nozawa, L. Rolshoven, M. Chan, Y. Akimoto, brieglhostis, and D. Brockhoff, "Cma-es/pycma: r3.2.0," Feb. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6300858

[11] D. H. Salunkhe, G. Michel, S. Kumar, M. Sanguineti, and D. Chablat, "An efficient combined local and global search strategy for optimization of parallel kinematic mechanisms with joint limits and collision constraints," *Mechanism and Machine Theory*, vol. 173, p. 104796, 2022.

[12] A. E. Gkikakis and R. Featherstone, "Robust analysis for mechanism and behavior co-optimization of high-performance legged robots," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, 2022, pp. 752–758.

[13] Z. Manchester and S. Kuindersma, "Dirtrel: Robust trajectory optimization with ellipsoidal disturbances and lqr feedback," in *Robotics: Science and Systems (RSS)*, 2017. [Online]. Available: https://github.com/HarvardAgileRoboticsLab/drake/tree/dirtrel

[14] G. Fadini, T. Flayols, A. D. Prete, and P. Souères, "Simulation aided co-design for robust robot optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11306–11313, 2022.

[15] L. J. Maywald, F. Wiebe, S. Kumar, M. Javadi, and F. Kirchner, "Co-optimization of acrobot design and controller for increased certifiable stability," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2636–2641.

[16] B. R. R., R. A. A., and K. D. E., "Sequential composition of dynamically dexterous robot behaviors," *International Journal of Robotics Research*, vol. 8, no. 18, pp. 534–555, 1999.

[17] J. Moore, R. Cory, and R. Tedrake, "Robust post-stall perching with a simple fixed-wing glider using lqr-trees," *Bioinspiration and Biomimetics*, vol. 9, no. 2, p. 025013, may 2014. [Online]. Available: https://dx.doi.org/10.1088/1748-3182/9/2/025013

[18] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010. [Online]. Available: https://doi.org/10.1177/0278364910369189

[19] P. Reist, P. Preiswerk, and R. Tedrake, "Feedback-motion-planning with simulation-based lqr-trees," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1393–1416, 2016. [Online]. Available: https://doi.org/10.1177/0278364916647192

[20] F. Wiebe, S. Kumar, L. J. Shala, S. Vyas, M. Javadi, and F. Kirchner, "Open source dual-purpose acrobot and pendubot platform: Benchmarking control algorithms for underactuated robotics," *IEEE Robotics and Automation Magazine*, pp. 2–13, 2023.

[21] F. C. Park, B. Kim, C. Jang, and J. Hong, "Geometric Algorithms for Robot Dynamics: A Tutorial Review," *Applied Mechanics Reviews*, vol. 70, no. 1, p. 010803, 02 2018. [Online]. Available: https://doi.org/10.1115/1.4039078

[22] A. Mueller and S. Kumar, "Closed-form time derivatives of the equations of motion of rigid body systems," *Multibody System Dynamics*, vol. 53, no. 3, pp. 257–273, 2021.

[23] R. Tedrake, *Underactuated Robotics*. Course Notes for MIT 6.832, 2023. [Online]. Available: https://underactuated.csail.mit.edu

[24] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM Review*, no. 47(1):99-131, 2005.

[25] E. Najafi, R. Babuška, and G. A. D. Lopes, "A fast sampling method for estimating the domain of attraction," *Nonlinear Dynamics*, no. 86, p. 823–834, 2016.

[26] H. K. Khalil, *Non linear Systems*. Prentice Hall, Upper Saddle River, NJ, 1996.

[27] J. Moore, R. Cory, and R. Tedrake, "Robust post-stall perching with a simple fixed-wing glider using lqr-trees," *Bioinspiration & Biomimetics*, vol. 9, no. 2, p. 025013, may 2014. [Online]. Available: https://dx.doi.org/10.1088/1748-3182/9/2/025013

[28] S. Boyd and L. Vandenberghe, "Convex optimization," cambridge University Press, p. 655.

[29] F. Wiebe, J. Babel, S. Kumar, S. Vyas, D. Harnack, M. Boukheddimi, M. Popescu, and F. Kirchner, "Torque-limited simple pendulum: A toolkit for getting familiar with control algorithms in underactuated robotics," *Journal of Open Source Software*, vol. 7, no. 74, p. 3884, 2022. [Online]. Available: https://doi.org/10.21105/joss.03884

[30] Quanser, "Linear servo base unit with inverted pendulum," simulink Courseware. [Online]. Available: https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum/

[31] L. Gross, L. Maywald, S. Kumar, F. Kirchner, and C. Lüth, "Analytic estimation of region of attraction of an lqr controller for torque limited simple pendulum," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 2695–2701.

[32] M. Javadi, D. Harnack, P. Stocco, S. Kumar, S. Vyas, D. Pizzutilo, and F. Kirchner, "AcroMonk: A minimalist underactuated brachiating robot," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3637–3644, jun 2023. [Online]. Available: https://doi.org/10.1109%2Flra.2023.3269296

[33] R. Soni, D. Harnack, H. Isermann, S. Fushimi, S. Kumar, and F. Kirchner, "End-to-end reinforcement learning for torque based variable height hopping," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7531–7538.

[34] G. S. Shourie, M. Javadi, S. Kumar, H. Z. Boroujeni, and F. Kirchner, "Ricmonk: A three-link brachiation robot with passive grippers for energy-efficient brachiation," in *2024 International Conference on Robotics and Automation (ICRA)*, 2024.

[35] M. Boukheddimi, S. Kumar, H. Peters, D. Mronga, R. Budhiraja, and F. Kirchner, "Introducing rh5 manus: A powerful humanoid upper body design for dynamic movements," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 01–07.