2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)
October 23-27, 2022, Kyoto, Japan

Co-optimization of Acrobot Design and Controller for Increased Certifiable Stability

Lasse Jenning Maywald[1,*], Felix Wiebe[1,*], Shivesh Kumar[1], Mahdi Javadi[1], and Frank Kirchner[1,2]

***Abstract*— Unlike fully actuated systems, the control of underactuated robots necessitates the use of passive dynamics to fulfill control objectives. Hence, there is an increased interdependence between their design parameters and the closed loop performance. This paper proposes a novel approach for co-optimization of robot design and controller parameters for increased certifiable stability obtained with means of region of attraction analysis and gradient free optimization. In particular, it discusses the co-optimization problem of a gymnastic acrobot robot where the design and the controller are optimized to have a large region of attraction (ROA) taking into account the closed loop dynamics of the non-linear system stabilized by a linear quadratic regulator (LQR) controller. The results are validated by extensive simulation of the acrobot's closed loop dynamics.**

I. Introduction

The human sensorimotor system is a result of various mechanisms working together on different time scales. More specifically, behavioral performance emerges from processes of evolution, development, learning and adaptation [1]. Analogous to the synergy of body and behavior in living beings, optimal behavioral performance of robots could be attributed to a shared optimality between three different domains: (i) open loop trajectory optimization, (ii) closed loop trajectory stabilization via feedback control and (iii) design optimization. The coupling between these domains (Fig. 1) renders itself even stronger when dealing with underactuated robots where the controller necessitates the use of passive dynamics to fulfill control objectives.

In recent robotics research, the interdependence between the three domains is being increasingly exploited in both model-free as well as model-based control domains. Boston Dynamics' recent successful advancement of the 34-DOF humanoid robot, Atlas, in performing highly dynamic tasks, such as backflips, parkour, and dance, clearly shows the significance of kinodynamic whole-body approaches to robotic motion [2]. An approach for increasing the robustness in the trajectory optimization step by considering ellipsoidal disturbances and LQR feedback is discussed in [3]. A co-optimization that optimizes the design and motion parameter adjustment in the form of trajectories or actuator force specification simultaneously was proposed in [4]. The validation of the framework is done through hardware implementation of the manipulator and a legged robot. Furthermore, a co-design scheme for the hardware optimization and DDP based energy-efficient optimal hopping trajectory optimization was introduced in [5]. An open-source framework for co-design optimization of legged robots through a user-defined metric was presented in [6]. A three-step approach was used: (i) trajectory generation for a baseline robot using TOWR (Trajectory Optimization for Walking Robots [7]), (ii) motion analysis, and (iii) design optimization using a genetic algorithm. The pipeline was validated on a quadruped robot by considering the energy minimization metric. A bi-level optimization approach for the co-design of a quadruped was proposed in [8]. It considers motion planning and design optimization as a lower and upper-level optimization, respectively. Another bi-level framework was introduced in [9]. Based on stochastic programming, it was able to produce mechanical structures and control parameters that performed well for a broad range of nominal trajectories.

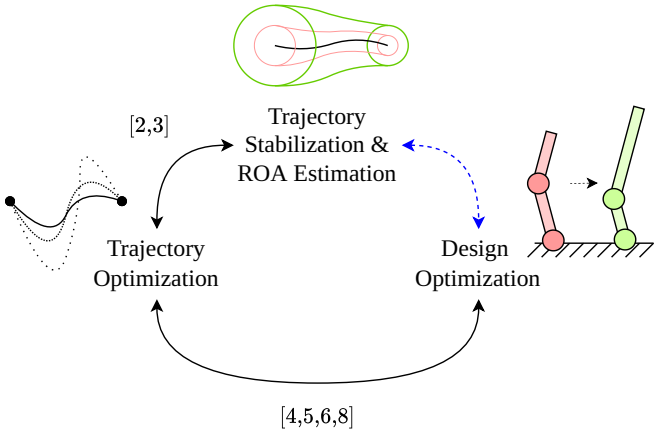


Fig. 1. Interplay between trajectory optimization, stabilization and design optimization for optimal behavioral performance.

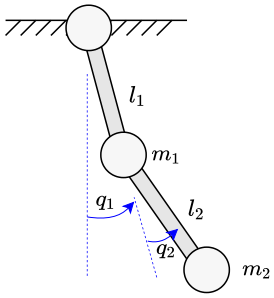
The Region of Attraction (ROA) associated to a fixed point of a nonlinear dynamical system can be used to certify stability. For closed loop dynamics, a large ROA around the desired state is favorable. Closed loop dynamics and ROA are shaped by both, the design- as well as control parameters. The concepts of ROA and funnels have gained popularity in the optimal control community, especially in the study of underactuated systems. While ROA estimation has been used for controller verification and policy synthesis [10],

This work has been performed in the M-RoCK project funded by the German Aerospace Center (DLR) with federal funds (Grant Number: 01IW21002) from the Federal Ministry of Education and Research (BMBF) and is supported with project funds from the federal state of Bremen for setting up the Underactuated Robotics Lab (Grant Number: 201-342-04-2/2021-4-1) and the KiMMI-SF project funded by the German Aerospace Center (DLR) with federal funds (Grant Number: 50RA2021) from the German Federal Ministry of Economic Affairs and Climate Action (BMWK).

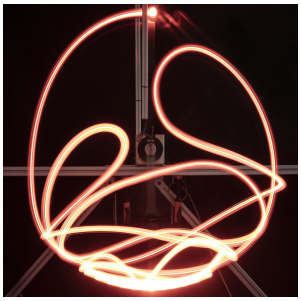
*Both authors equally contributed to this work.

[1]Robotics Innovation Center, DFKI GmbH, Bremen, Germany firstname.lastname@dfki.de

[2]AG Robotics, Department of Mathematics and Computer Science, University of Bremen, Germany frank.kirchner@uni-bremen.de



(a) Acrobot kinematic chain



(b) Chaotic free fall

Fig. 2. (a) Acrobot kinematic chain and (b) long exposure shot of chaotic free-fall.

the link to design optimization is yet to be made. To the best knowledge of the authors, the co-optimization of the feedback controller and robot design with the objective of maximizing the ROA has not been discussed in the literature (depicted by the dotted blue line in Fig. 1). This work proposes a novel approach for co-optimization of robot design and controller for maximizing its certifiable stability obtained via the means of ROA analysis. In order to demonstrate our approach, we perform a case study for the stabilization of the acrobot system (Fig. 2) around its unstable fixed point associated to the upright posture using an infinite horizon LQR policy. The optimization strategy exploits two popular gradient free optimization algorithms namely Nelder-Mead and CMA-ES [11], [12].

This paper is organized as follows: Section II introduces the mathematical preliminaries for establishing the system dynamics, derivation of LQR controller for the acrobot and ROA estimation. The overall methodology where the objective function maximizing the stability is introduced along with methods used for co-optimization of controller and design is covered in Section III. Section IV presents the results and discussion and finally Section V concludes the paper and motivates future research.

II. Mathematical Preliminaries

A. System Dynamics

An acrobot is a planar, two-link chain with a single actuator at the middle joint [13]. Non-actuated double pendulums are known to be chaotic systems [14]. With only one actuated joint, the acrobot is an underactuated system for which simple control techniques such as PD position control are not sufficient [15].

We model the acrobot with weightless links with point masses at the end. Thus, our dynamic model of the acrobot contains five parameters. These are the masses m_1 and m_2, the link lengths l_1 and l_2 as well as gravity g (Fig. 2(a)). The generalized coordinates describing the system configuration are the joint angles $\boldsymbol{q} = (q_1, q_2)^T$. q_1 is measured from the free hanging position whereas q_2 is defined relative to q_1. The full state vector of the system contains the coordinates and the angular velocities: $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})$, where a dot above the vector denotes its time derivative. The torque applied by the actuator is $\boldsymbol{u} = (u)$. The equations of motion for the dynamics of such a dynamical system can be written as [13]:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{M}^{-1}(\boldsymbol{B}_{sel}\boldsymbol{u} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q})) \end{bmatrix} \tag{1}$$

The equation in the bottom half of the vector is also known as manipulator equation. For the acrobot, the entities in the manipulator equation are the mass matrix (with $s_i = \sin(q_i), c_i = \cos(q_i)$)

$$\boldsymbol{M} = \begin{bmatrix} m_1 l_1^2 + m_2 l_2^2 + l_1^2 m_2 + 2 m_2 l_1 l_2 c_2 & m_2 l_2^2 + m_2 l_1 l_2 c_2 \\ m_2 l_2^2 + m_2 l_1 l_2 c_2 & m_2 l_2^2 \end{bmatrix},$$

the coriolis matrix

$$\boldsymbol{C} = \begin{bmatrix} -2\dot{q}_2 m_2 l_1 l_2 s_2 & -\dot{q}_2 m_2 l_1 l_2 s_2 \\ \dot{q}_1 m_2 l_1 l_2 s_2 & 0 \end{bmatrix}, \tag{2}$$

the gravity vector

$$\boldsymbol{G} = \begin{bmatrix} -g m_1 l_1 s_1 - g m_2 \left(l_1 s_1 + l_2 s_{1+2}\right) \\ -g m_2 l_2 s_{1+2} \end{bmatrix}, \tag{3}$$

and the actuator selection matrix $\boldsymbol{B}_{sel} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$.

B. LQR Controller

The linear quadratic regulator (LQR) controller is a well established and widespread controller from optimal control theory. As the name suggests the controller acts on a linear system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{4}$$

and an objective which is specified by a quadratic, instantaneous cost function

$$J = \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u} \tag{5}$$

with the symmetric and positive definite matrices $\boldsymbol{Q} = \boldsymbol{Q}^T \succeq 0$ and $\boldsymbol{R} = \boldsymbol{R}^T \succ 0$. This allows for reducing the Hamilton-Jacobi-Bellman equation, whose solution is the optimal cost -to-go, from which the optimal policy can be inferred, to the algebraic Riccati equation

$$\boldsymbol{S}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{S} - \boldsymbol{S}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S} + \boldsymbol{Q} = 0 \tag{6}$$

for which there exist performant numerical solvers for finding the optimal cost-to-go matrix $\boldsymbol{S}$. The optimal policy obtained this way is

$$\boldsymbol{u}(\boldsymbol{x}) = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x} = -\boldsymbol{K}\boldsymbol{x}. \tag{7}$$

Such an infinite-horizon LQR controller will always apply torques to drive the system state towards the origin of the linear system. An LQR controller can be used for stabilizing the acrobot on the top by linearizing the dynamics around the top position $\boldsymbol{x}_0 = (\pi, 0, 0, 0)$ and $\boldsymbol{u}_0 = (0)$ with a Taylor expansion so that:

$$\boldsymbol{A} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_0, \boldsymbol{u}=\boldsymbol{u}_0}, \tag{8}$$

$$\boldsymbol{B} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\boldsymbol{x}_0, \boldsymbol{u}=\boldsymbol{u}_0} \tag{9}$$

and expressing the state and actuation in relative error coordinates $\bar{\boldsymbol{x}} = (\boldsymbol{x} - \boldsymbol{x}_0)$, $\bar{\boldsymbol{u}} = (\boldsymbol{u} - \boldsymbol{u}_0)$. In the following, let $\boldsymbol{f}_{cl}$ denote the closed loop dynamics obtained by inserting (7) into (1).

C. Region of Attraction Estimation

The region of attraction of a nonlinear system describes the set, $\mathcal{R}_a$, of initial states around a fixed point $\boldsymbol{x}_0$ for which the system evolves towards said fixed point as $t \to \infty$. For sufficiently complex systems, $\mathcal{R}_a$ generally can not be computed directly but only be estimated. The simplest such estimate considers sublevel sets of a known Lyapunov function $V(\boldsymbol{x})$ that is limited by a scalar ρ [16]:

$$\mathcal{B} = \{\boldsymbol{x} | V(\boldsymbol{x}) < \rho\}. \tag{10}$$

We seek for the greatest ρ such that the Lyapunov conditions are satisfied, i.e., where $V(\boldsymbol{x}) > 0$ and $\dot{V}(\boldsymbol{x}) = \nabla V \boldsymbol{f}(\boldsymbol{x}) < 0$ for $x \in \mathcal{B}$. Here, ∇V denotes the gradient of V. The linear optimal cost-to-go, $J^\star$, that is obtained during LQR synthesis, locally qualifies as a quadratic Lyapunov function [17]:

$$V(\boldsymbol{x}) = J^\star(\boldsymbol{x}) = \bar{\boldsymbol{x}}^{\mathrm{T}} \boldsymbol{S} \bar{\boldsymbol{x}}. \tag{11}$$

For the infinite horizon LQR, $\partial \boldsymbol{S}/\partial t = 0$ and $\partial \boldsymbol{x}_0/\partial t = 0$, hence, $\dot{V}$ is:

$$\dot{V}(\boldsymbol{x}) = 2\bar{\boldsymbol{x}}^{\mathrm{T}} \boldsymbol{S} \dot{\boldsymbol{x}}. \tag{12}$$

For time invariant dynamics, $\mathcal{R}_a$ can be estimated by evaluating the Lyapunov conditions for randomly chosen initial states sampled from a successively shrinking estimate $\mathcal{B}$. We use a slightly modified version of the memory-less algorithm introduced in [18] listed in Algorithm 1 to obtain an inner approximation of the ROA. First ρ is initialized with a reasonable first ρ_0, such that the first estimate of the ROA is guaranteed to be an overestimate. Then a random state $\tilde{\bar{\boldsymbol{x}}}$ is sampled directly from the current estimate of $\mathcal{R}_a$ and $\dot{\boldsymbol{x}}$ is evaluated by considering the nonlinear closed loop dynamics $\boldsymbol{f}_{cl}$. We calculate $\dot{V}$ and check for $\dot{V} < 0$. In case this condition is not satisfied, we set $\rho = V$. Note, that we do not have to check for $V(\boldsymbol{x}) > \rho$, since $\tilde{\bar{\boldsymbol{x}}}$ is sampled directly from within the current estimate of $\mathcal{B}$. This can be achieved by sampling from the d dimensional unit sphere using the *direct-sphere* algorithm in [19] followed by a mapping to the hyperellipsoid defined by (11) using a linear transformation introduced in the following.

It is well known, that a linear transform can be used to map the unit ball in $\mathbb{R}^n$ to an ellipsoid in $\mathbb{R}^m$ and vice versa. Consider the n-ball $\mathcal{S}_n = \left\{\boldsymbol{y} | \boldsymbol{y}^{\mathrm{T}} \boldsymbol{y} \leq 1\right\}$ and a transform $\boldsymbol{T} \in \mathbb{R}^{n \times m}$, such that $\boldsymbol{y} = \boldsymbol{T}\boldsymbol{x}$. Expressing $\mathcal{S}_n$ in terms of $\boldsymbol{x}$ yields:

$$\mathcal{S}_n = \left\{\boldsymbol{x} | \boldsymbol{x}^{\mathrm{T}} \boldsymbol{T}^{\mathrm{T}} \boldsymbol{T} \boldsymbol{x} \leq 1\right\}. \tag{13}$$

This describes an ellipse in $\boldsymbol{x}$, since $\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}$ is symmetric and by definition positive definite. A diagonalization of $\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}$ yields:

$$\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T} = \boldsymbol{W}^{\mathrm{T}} \boldsymbol{\Lambda} \boldsymbol{W} \tag{14}$$

Algorithm 1 Sampling Based RoA Estimation for infinite horizon LQR based on [18]

Require: $\boldsymbol{S}$,N,ρ_0
  $\rho \leftarrow \rho_0$
  for $i \leftarrow 0$ to N **do**
    $\tilde{\bar{\boldsymbol{x}}} \leftarrow$ sample directly from $\bar{\boldsymbol{x}}^{\mathrm{T}} \boldsymbol{S} \bar{\boldsymbol{x}} < \rho$
    $\dot{\boldsymbol{x}} \leftarrow \boldsymbol{f}_{cl}(\boldsymbol{x}_0 + \tilde{\bar{\boldsymbol{x}}})$
    $\dot{V} \leftarrow 2\tilde{\bar{\boldsymbol{x}}}^{\mathrm{T}} \boldsymbol{S} \dot{\boldsymbol{x}}$
    if $\dot{V} > 0$ **then**
      $\rho \leftarrow \tilde{\bar{\boldsymbol{x}}}^{\mathrm{T}} \boldsymbol{S} \tilde{\bar{\boldsymbol{x}}}$
    end if
  end for
  return ρ

where $\boldsymbol{W}$ and $\boldsymbol{\Lambda}$ are matrices of eigenvectors and eigenvalues, respectively. Therefore, the transform $\boldsymbol{T}$ can be written as:

$$\boldsymbol{T} = \sqrt{\boldsymbol{\Lambda}} \boldsymbol{W}. \tag{15}$$

Using $\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T} = \rho^{-1}\boldsymbol{S}$, the following transform allows us to sample randomly from a sublevel set of a quadratic Lyapunov function such as e.g. (11):

$$\tilde{\bar{\boldsymbol{x}}} = \left(\sqrt{\boldsymbol{\Lambda}} \boldsymbol{W}\right)^{-1} \boldsymbol{y}. \tag{16}$$

where $\boldsymbol{y}$ is the state sampled directly from the unit ball.

III. Methodology

The problem of finding optimal design and controller parameters, such that the ROA around a fixed point to be stabilized is maximized, is formulated as a gradient free optimization problem. The volume of the estimated ROA is used as the objective and the design and control parameters are the decision variables. The approach presented here is applicable to problems that involve stabilizing a system around one of its fixed points using LQR.

A. Objective Function

Recall, that in ROA estimation, one seeks to find the greatest sublevel set of a Lyapunov function V limited by a scalar ρ. Because a given ρ only has expressive power for a specific V, the state space volume of the estimated ROA, $V_{\mathrm{ROA}}(\mathcal{B})$, is used as an objective function. $V_{\mathrm{ROA}}(\mathcal{B})$ can be calculated by multiplying the volume of the n ball with the determinant of the mapping given in (15), which is the transform that maps n ball to the hyperellipsoid that describes the estimated ROA.

$$V_{\mathrm{ROA}} = |det(\sqrt{\boldsymbol{\Lambda}} \boldsymbol{W})|^{-1} \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2} + 1)} \tag{17}$$

where Γ denotes Euler's Gamma function. Note, that the fraction in the end of (17) is a constant and could be omitted during optimization.

B. Optimization Methods

1) Nelder-Mead: The Nelder-Mead method [11] is a commonly used numerical gradient free optimization method. For the optimization of an n-dimensional objective function, an $n+1$ simplex is maintained during the optimization process. The objective function is evaluated at the simplex's corner points and the behavior of the objective function is extrapolated to find a new, better performing point in the decision variable space to replace an existing simplex corner point. For low dimensional problems, it outperformed the other optimization problems in a survey of black box optimizations [20]. This method has been exploited in robotics research for example in kinematic optimization of parallel mechanisms [21]. For our analysis we used the Nelder-Mead implementation included in the scipy python library [22].

2) CMA-ES: The Covariance Matrix Adaption Evolution Strategy (CMA-ES) is a numerical optimization method belonging to the class of evolutionary algorithms [12]. It is a stochastic and gradient-free method and is particularly useful for non-convex or ill-conditioned objective functions [20]. The algorithm is motivated by the rules of biological evolution. For the optimization, a population of candidate solutions (individuals) is maintained. In each optimization iteration, new candidate solutions are sampled by variation of the current population, where better performing individuals are preferred for producing new solutions. Variants of the CMA-ES optimizer outperformed 31 other optimization algorithms for difficult functions in the aforementioned survey of black box optimizations [20]. For our analysis we used the implementation of the CMA-ES algorithm from [23].

C. Co-optimization of Controller and Design

For controller and design optimization, we assume an acrobot model as described in Section II-A. An infinite horizon LQR policy is obtained by considering the linearized dynamics at the unstable fixed point at $\boldsymbol{x}_0 = (\pi, 0, 0, 0)^{\mathrm{T}}$. The overarching goal of our co-optimization is to find a physical design and controller parameters, such that the volume V_{ROA} of the ROA of $\boldsymbol{x}_0$ is maximized. The decision variables include m_2, the point mass at the tip of the second link, the link lengths l_1 and l_2. Furthermore, we consider the diagonal elements of $\boldsymbol{Q}$ and $\boldsymbol{R}$, which are denoted by q_{ii} and r, respectively. The mass of the first link $m_1 = 0.608\,kg$ is kept constant because in the considered hardware design a motor is mounted at the tip of the first link. Gravity is set to $g = 9.81\,\mathrm{m\,s^{-2}}$ in all simulations. The complete optimization problem can be formulated as:

$$\max_{m_2, l_1, l_2, q_{ii}, r} \quad V_{\mathrm{ROA}} \tag{18a}$$

$$\text{subject to} \quad m_{2,\min} \leq m_2 \leq m_{2,\max}, \tag{18b}$$

$$l_{1,\min} \leq l_1 \leq l_{1,\max}, \tag{18c}$$

$$l_{2,\min} \leq l_2 \leq l_{2,\max}, \tag{18d}$$

$$q_{ii,\min} \leq q_{ii} \leq q_{ii,\max}, i \in \{1 \dots 4\}, \tag{18e}$$

$$r_{\min} \leq r \leq r_{\max}. \tag{18f}$$

Where, in practice, the bounds for m_2 are $[0.1, 1.0]\,\mathrm{kg}$, for l_1 and l_2 are $[0.2, 0.4]\,\mathrm{m}$, for q_{11}, q_{22} are $[0.1, 10.0]$, and q_{33}, q_{44} as well as r are $[0.1, 1.0]$. Note, that off-diagonal elements were not considered in our work for the sake of keeping the dimensionality of the search space low. We do not expect qualitatively different outcomes when including them. We employ different optimization strategies in order to solve (18) either in an alternating or a combined fashion. In the combined approach, (18) is solved at once. For the alternating approach, we try to find the optimal design first (*design first*), and then seek for the optimal control parameters, or vice versa (*controller first*), which requires splitting (18) into two sub-problems. For the design optimization sub-problem, we only consider m_2, l_1 and l_2 as decision variables and (18b) to (18d) as constraints. Similarly, only q_{ii} and r as well as (18e) and (18f) are considered in the control parameter optimization sub-problem. For each evaluation of either one of the optimization problems, first an LQR feedback policy is synthesized for a given set of design and control parameters. Subsequently, by using the optimal cost-to-go as a Lyapunov function (11), a ROA estimation is conducted according to Algorithm 1. In practice, we found $N = 10^5$ for a good compromise of accuracy of the estimate on the one hand and computational feasibility on the other.

Note, that for the closed loop dynamics $\boldsymbol{f}_{cl}$, the ROA always exists even for poorly chosen parameters. Hence, any initial design can be considered feasible, albeit often times suboptimal. However, due to nonlinear effects such as e.g. coulomb-friction, that have not been considered in this work, this does not translate to the real world acrobot plant.

IV. Results and Discussion

We conducted five different kinds of optimizations for the design and controller parameters (source code available online[1]). We performed *design first* and *controller first* with both, Nelder-Mead and CMA-ES, optimizers. Additionally, we performed *simultaneous* optimization of all parameters with CMA-ES. Nelder-Mead optimizations were executed with a maximum number of $N_{max} = 200$ function evaluations and the CMA-ES optimizations with $N_{max} = 1000$ function evaluations. All optimizations were initialized with the initial parameters: $m_2 = 0.63\,\mathrm{kg}$, $l_1 = 0.3\,\mathrm{m}$, $l_2 = 0.2\,\mathrm{m}$ and identity matrices for $\boldsymbol{Q}$ and $\boldsymbol{R}$. These parameters are identical to those of the real-world acrobot depicted in Fig. 2(b). The CMA-ES optimizations used a population size of 29 (design optimization), 35 (controller optimization) or 40 (*simultaneous*) with an initial standard deviation of $\sigma_0 = 0.4$ in the normalized parameter space. Nelder-Mead optimizations were executed on a single *i9* core, whereas the CMA-ES optimizations were run on an Intel Xeon Gold 6140 CPU in parallel with one core for every individual in every generation. The runtimes T_{opt} are noted in Table I.

In early experiments with the Nelder-Mead optimizer no convergence could be observed for the controller optimization. For this reason we assumed $q_{11} = q_{22}$ and $q_{33} = q_{44}$ to reduce the dimensionality of the optimization problem. The progression of the optimizations in terms of the largest

[1] https://github.com/dfki-ric-underactuated-lab/design_controller_cooptimization_acrobot

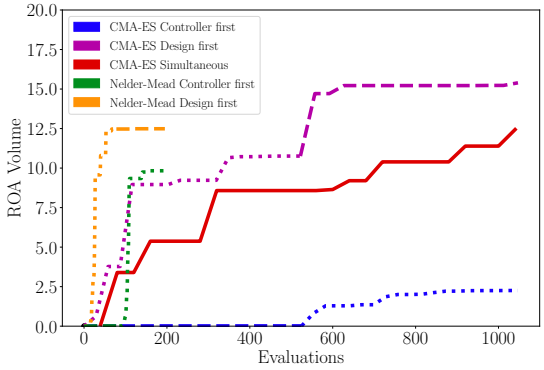
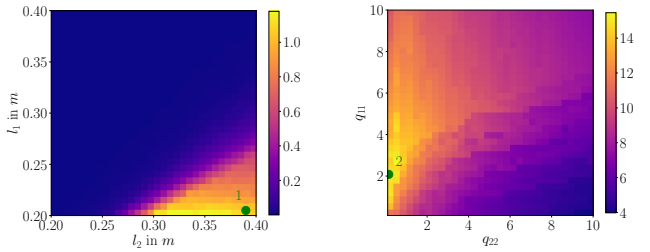


Fig. 3. Progression of ROA volume during optimization for the different optimization strategies. The design optimization phases are indicated by dotted lines and the controller parameter optimization phases by dashed lines. The *simultaneous* CMA-ES optimization is the solid line.



(a) l_1-l_2 plane (b) q_{11}-q_{22} plane

Fig. 4. Estimated ROA in the (a) l_1 vs. l_2 dimensions for the initial controller parameters, and (b) q_{11} vs. q_{22} dimensions for the optimal design parameters. The green dots mark the optimal solution found by the CMA-ES *design first* optimization strategy.

ROA volume versus number of ROA estimations used for the optimization can be found in Fig. 3.

The *design first* strategy performs considerably better than the *controller first* strategy for both Nelder-Mead and CMA-ES. The increase in the objective function is steeper for Nelder-Mead in the design optimization phase, i.e. Nelder-Mead needs fewer function calls to find a decent region of attraction. However, Nelder-Mead seems to be over-challenged by the controller optimization with its five decision variables and the dimensionality reduction from setting $q_{11} = q_{22}$ and $q_{33} = q_{44}$ also prevents a good convergence. There are only minor improvements during these phases. CMA-ES *controller first* does perform the worst of all tested strategies. On the other hand, CMA-ES *design first* has found the largest ROA after 1000 evaluations. CMA-ES *simultaneous* performs reasonably well, even though in the optimization window $N_{max} \leq 1000$ it has not yet converged and is less sample efficient than CMA-ES *design first*. This can be explained with the higher dimensional search space. With more evaluations, we expect CMA-ES *simultaneous* to reach at least the level of CMA-ES *design first*, if it does not get stuck in a local minimum. In our optimizations, CMA-ES takes more evaluations than Nelder-Mead. The CMA-ES hyper parameters could be tuned for a faster convergence, however, as we did not want to lose the global optimality by converging too fast to a local optimum, we sticked with the default hyper parameters for a parameter space of our size. The optimized parameters for design and controller, the ROA volumes and the computation times are listed in Table I.

Fig. 4 depicts the ROA volume values in slices of the decision variable space. The slices are in the l_1-l_2 plane as well as the q_{11}-q_{22} plane. For the slice in the l_1-l_2 plane, the other parameters are fixed at the initial parameters. The heatmap therefore shows the sliced objective function for the *design first* optimization strategies. In the q_{11}-q_{22} plane, the design parameters are set to the optimal parameters which were found in the CMA-ES *design first* optimization. The other controller parameters are kept constant at the final optimal parameters found by the same strategy. The l_1-l_2 slice plot shows a clear objective function gradient towards shorter first links and longer second links. The gradient in l_2 direction is steeper than in l_2 direction. The objective function in the q_{11}-q_{22} plane shows a more complex structure. Clearly, smaller q_{22} values are preferable over larger ones. For small q_{22} the influence of q_{11} is small. The global maximum in this plane is at approximately $(2.0, 0.1)$ which is found by CMA-ES *design first*.

Fig. 5 shows the approximated ROAs for the initial parameters as well as for the optimized parameters that were found using the CMA-ES *design first* optimization strategy. The initial ROA volume is $V_{ROA,0} = 0.0013$ and the optimized volume is $V_{ROA,opt} = 16.18$, which represents an increase of roughly four orders of magnitude. The increase in volume leads to a larger certifiable set of stabilizable initial states, hence, the optimized design is more robust than the initial design. Final ROA estimates have been verified by performing extensive simulations ($n = 1000$) of the forward dynamics for initial conditions sampled randomly from the estimated ROA. For each of the designs listed in Table I, no initial conditions were found, for which the top position

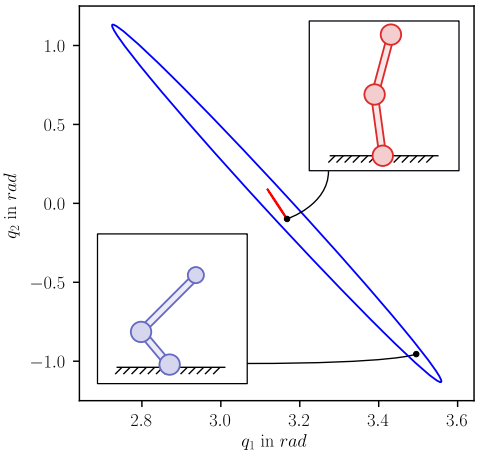


Fig. 5. ROA estimates projected into joint space for the initial design (blue) and for the improved design (red), assuming that $\dot{q}_1 = \dot{q}_2 = 0$. The optimized design was found using the CMA-ES *design first* optimization strategy and exhibits a significantly larger ROA, which in practice means, that larger deviations wrt. $\boldsymbol{x}_0$ can still be stabilized by the controller (as shown by the pictograms).

TABLE I

RESULTS OF VARIOUS OPTIMIZATION STRATEGIES.

Method	Strategy	Design			Controller						
		m_2 in kg	l_1 in m	l_2 in m	q_{11}	q_{22}	q_{33}	q_{44}	r_{11}	V_{ROA}	T_{opt} in min
Nelder Mead	Design First	0.20	0.20	0.40	1.00	1.00	1.00	1.00	0.95	12.49	67
	Controller First	0.17	0.20	0.40	10.00	10.00	0.10	0.10	0.66	9.83	61
CMA-ES	Design First	0.22	0.20	0.40	2.08	0.15	0.99	0.99	0.62	16.18	116
	Controller First	0.10	0.20	0.39	0.11	8.00	0.19	0.15	0.74	2.18	112
	Simultaneous	0.25	0.20	0.39	9.11	2.16	0.61	0.88	0.83	15.50	59

could not be stabilized, which probabilistically verifies the ROA estimate.

V. CONCLUSIONS

A design and controller co-optimization was conducted with the goal of maximizing the ROA of the closed loop dynamics of an acrobot using LQR control. Mechanism design greatly contributes to achieving robust dynamical control. This work demonstrated that ROA estimation can in fact be employed to find design and control parameters that lead to a larger closed loop ROA. In the presented case study of the acrobot, finding optimal design parameters first and control weights thereafter using a CMA-ES optimizer yielded the best overall results. On the other hand, Nelder-Mead required fewer calls of the objective function and showed faster convergence, albeit to a local minimum. The proposed method is applicable only to a limited range of scenarios, in which a single fixed point is being stabilized by an infinite horizon LQR controller. Furthermore, it might not scale well for higher dimensional problems. Building upon this work, further research is needed in order to be able to solve problems that involve optimization of parameters for trajectory stabilization scenarios. Because the nonlinear dynamics of a robot are shaped not only by control parameters and policies, but also heavily by its mechanical structure, we believe that more holistic design methods can greatly improve the design of future robots.

REFERENCES

[1] E. Todorov, "Optimality principles in sensorimotor control," *Nature neuroscience*, vol. 7, no. 9, pp. 907–915, 2004.

[2] S. Kuindersma, "Recent progress on atlas, the world's most dynamic humanoid robot," 2020, Robotics Today - A series of technical talks. [Online]. Available: https://www.youtube.com/watch?v=EGABAx52GKI

[3] Z. Manchester and S. Kuindersma, "Dirtrel: Robust trajectory optimization with ellipsoidal disturbances and lqr feedback." in *Robotics: Science and Systems*. Cambridge, MA, USA, 2017.

[4] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1521–1536, 2018.

[5] 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)*. IEEE, 2021, pp. 9898–9904.

[6] M. Chadwick, H. Kolvenbach, F. Dubois, H. F. Lau, and M. Hutter, "Vitruvio: An open-source leg design optimization toolbox for walking robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6318–6325, 2020.

[7] A. W. Winkler, D. C. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, pp. 1560–1567, July 2018.

[8] T. Dinev, C. Mastalli, V. Ivan, S. Tonneau, and S. Vijayakumar, "Co-designing robots by differentiating motion solvers," *arXiv preprint arXiv:2103.04660*, 2021.

[9] G. Bravo-Palacios, A. D. Prete, and P. M. Wensing, "One Robot for Many Tasks: Versatile Co-Design Through Stochastic Programming," vol. 5, no. 2, pp. 1680–1687. [Online]. Available: https://ieeexplore.ieee.org/document/8972465/

[10] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," vol. 36, no. 8, pp. 947–982. [Online]. Available: https://doi.org/10.1177/0278364917712421

[11] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 01 1965. [Online]. Available: https://doi.org/10.1093/comjnl/7.4.308

[12] N. Hansen, *The CMA Evolution Strategy: A Comparing Review*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 75–102. [Online]. Available: https://doi.org/10.1007/3-540-32494-1_4

[13] R. Tedrake, *Underactuated Robotics*, 2022. [Online]. Available: http://underactuated.mit.edu

[14] R. Olfati-Saber and A. Megretski, "Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles," Ph.D. dissertation, USA, 2001, aAI0803036.

[15] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering (2nd ed.)*. CrC Press, 2015.

[16] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2002.

[17] 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, July 2010.

[18] E. Najafi, R. Babuška, and G. A. D. Lopes, "A fast sampling method for estimating the domain of attraction," *Nonlinear Dynamics*, vol. 86, no. 2, pp. 823–834, Oct. 2016.

[19] W. Krauth, *Statistical Mechanics: Algorithms and Computations*, ser. Oxford Master Series in Physics. Oxford: Oxford University Press, 2006, no. 13.

[20] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Posik, "Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking BBOB-2009," in *ACM-GECCO Genetic and Evolutionary Computation Conference*, Portland, United States, July 2010, pp. 1689-1696. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00545727

[21] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094114X22000684

[22] F. Gao and L. Han, "Implementing the Nelder-Mead simplex algorithm with adaptive parameters," *Computational Optimization and Applications*, vol. 51, no. 1, pp. 259–277, Jan. 2012.

[23] 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