Automated Design of Simple and Robust Manipulators for Dexterous In-Hand Manipulation Tasks using Evolutionary Strategies

Andre Meixner[1], Christopher Hazard[2], Nancy Pollard[2]

Abstract— **In spite of substantial progress, robust and dexterous in-hand manipulation remains a robotics grand challenge. Recent research has shown that optimization of robot hand morphology for specific tasks can result in custom hand designs that are low-cost, easy to maintain, and highly capable. However, the resulting manipulation strategies may not be very robust or generalizable in real-world situations. This paper shows that robustness can be improved dramatically by optimizing controls instead of contact force / trajectories and by considering uncertainty explicitly during the optimization process. We present a evolutionary algorithm based pipeline for co-optimizing hand morphology and control strategy over families of problems and initial states in order to achieve robust in-hand manipulation. We demonstrate that this approach produces robust results which utilize all surfaces of the hand and surprising dynamic motions. We showcase the advantage of optimizing joint limit values to create robust designs. Furthermore, we demonstrate that our approach is complementary to trajectory optimization based approaches and can be utilized to improve robustness of such results as well as to create custom hand designs from scratch. Results are shown for repositioning and reorienting diverse objects relative to the palm of the hand.**

I. Introduction

Dexterous manipulators can achieve manipulation goals much more efficiently than non-dexterous manipulators [1]. Many strategies involve building dexterous manipulators by imitating the kinematics of human hands which are versatile on a wide variety of tasks [2][3]. These human-like hands provide a high degree of freedom to increase the workspace of the manipulator, but are still difficult and expensive to build, control and maintain. In contrast, low cost hands with fewer degrees of freedom offer similar capabilities for specialized tasks [4][5] and are significantly less expensive. Since manually designing these manipulators for each task requires much effort, automated strategies for hand design optimization are preferred.

Trajectory optimization methods demonstrate exceptional capabilities to synthesize complex dexterous motions especially when using contact invariance [6]. Hazard et al. [7] shows that this concept can also be applied to optimize the robot hand morphology. As this trajectory optimization approach relies on approximations that guide the optimizer to a good result, the robustness of generated designs or motion plans is not ensured. To address the challenge of creating robust motions, we take a different approach, optimizing for control strategies which produce success over families of object geometries and initial states (i.e., utilizing domain randomization as in [3]). Evolutionary algorithms [8] allow the direct search for optimal morphology and control parameters in such a simulation environment. While this is obviously a more difficult approach due to the high-dimensional search space that must be explored, these evolved robots generalize much better to unseen environments [9].

This paper extends the previous work of Hazard et al. [7], which introduces a trajectory optimization pipeline to automatically generate low-DoF hand designs capable of executing given object manipulation tasks. We demonstrate the value of evolutionary algorithms to co-optimize the hand morphology and controller in a physics simulation, utilizing domain randomization for robustness. We similarly take a sequence of goal poses for the manipulated object as input task description. Consequently, the evolutionary strategy must be able to achieve multiple objectives containing object positions and orientations at specified timestamps simultaneously. We demonstrate the value of optimizing for joint limit locations, and demonstrate that our approach can both design custom hands and control strategies from scratch as well as improve the robustness of an existing hand and strategy created as described in the research of Hazard and his colleagues [7]. Co-optimizing the hand and control strategy as described in this paper results in simple hands and interesting dynamic motions which can robustly achieve desired in-hand manipulations in the presence of uncertainty.

II. Related Work

Evolutionary optimization strategies are scalable methods for optimizing control policies in physics simulation [10]. In particular, these strategies are well suited to evolve both the policy and the structure [8]. Recent work already provides different algorithms to handle the evolutionary optimization for multiple objectives. Single-objective evolutionary algorithms such as CMA-ES [11] can be used by minimizing the weighted sum of the objectives [12], but this trade-off needs to be chosen a-priori. The results of actual multi-objective algorithms are given as pareto front and allow the trade-off to be selected a-posteriori. Simultaneously optimizing multiple dimensions may require more generations and therefore more computational time to match the quality if good weights are already known. Multi-objective algorithms are the favored choice in literature for more complex approaches such as optimizing robot designs and controllers [9][13]. Well

*This work was supported in part by the National Robotics Initiative grant NSF IIS-1637853

[1]Andre Meixner is with the High Performance Humanoid Technologies Lab, Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany andre.meixner@student.kit.edu

[2]Christopher Hazard and Nancy Pollard are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 chazard@andrew.cmu.edu, nsp@cs.cmu.edu

known algorithms are MO-CMA [14][15], a multi-objective extension of CMA-ES, NSGA-III [16] and MOEA/D [17]. Many authors also suggest further improvements to balance convergence and diversity [18][19][20] but initially this paper relies on the basic algorithms.

Multiple works discuss the use of evolutionary algorithms in a physics simulation to co-optimize robot morphology and controller. Only Bongard [9] optimizes the hand morphology and controller for object manipulation. In his paper three-to-five fingered hands attached to a shoulder are optimized to simultaneously grasp, lift and actively perceive the category of different objects in reach. This hand-arm model contains a proprioceptive sensor in the shoulder and range/tactile sensors in each phalange. Retrieved sensor values are translated to desired target joint angles using a continuous-time recurrent neuronal network as controller. The weights and other parameters of the neuronal network controller, as well as the finger segments lengths, phalange radii and spacing between fingers are evolved by a multi-objective evolutionary algorithm. The author concludes that evolved robots generalize better in unseen environmental conditions and also that evolving more aspects of robot increases the probability of discovering successful solutions as the complexity of the task increases. Other papers contain improvements for the co-optimization process. Lehman et al. [21] suggest including a novelty metric as additional objective which measures the morphological difference to other individuals to reward diverging solutions and mitigate early convergence to a local minimum. The same author also proposes a different approach completely abandoning other objectives such as fitness and only optimizing for behavioral diversity [22]. Furthermore, Cheney et al. [23] recommend including how long a morphology remained unvaried as objective to keep possible good morphologies with poorly adapted controllers in the population. Nygaard et al. [13] propose a two-phases approach to better improve the robot's controller by locking the morphology in a second optimization phase and only adapting the control parameters. A different approach on co-optimization uses Compositional Pattern Producing Networks (CPPNs) to indirectly encode whole robots including physical topologies, sensor placements and embedded closed-loop neuronal network policies [24][25]. Auerbach et al. [25] evolve simple rigid body robots able to sense and navigate towards target objects. Cheney et al. [26] optimize two separate networks for morphology and controller of virtual creatures based on 3D voxel-based soft robots. The applied optimization algorithm gradually increases the complexity of the network by adding new nodes and links. Therefore, this method can dynamically adjust the resolution to explore a greater variety of possible solutions. Even though Auerbach et al. [24] state that this method may be useful for object manipulation, no further research on this problem is present.

The alternative strategy to evolutionary algorithms is based on a trajectory optimization pipeline [7]. This method designs the hand morphology and a physically plausible joint angle trajectory which exerts specific forces at determined contact point locations on the object. The proposed pipeline allows the generation of feasible mechanisms on a variety of in-hand manipulation tasks. However, this approach does not address the building of robust manipulators. The required forces to move the object to its desired pose are interpolated between selected waypoints and not calculated. Furthermore, this framework does not incorporate planning under uncertainty. Executing obtained result in a physics simulation will most likely lead to failure as even slight differences in exerted forces, contact points or the environment perturb the motion. Domain randomization can be used generate more robust control policies in simulation that are able to transfer to similar environments and also generalize better to reality [3]. As an example, observation noise can be added or physical parameters of the manipulated object or robot such as surface friction coefficients, object dimensions and masses may be randomized in a scene.

This paper presents an approach for generating hands for any desired object manipulation based on a high-level task description. Our goal is to create manipulators which already incorporate robustness in their structure and do not rely on complex controllers. Therefore, we choose to optimize simple torque-based policies on sensor-less hands so that the hand morphology reduces uncertainty as much as possible. An important aspect of our designs is the optimization of joint limit parameters to increase robustness. We are unaware of any case in literature attempting to do this.

III. Methods

This section covers the structure of the optimization algorithm, parameters of the hand morphology and controller to be optimized as well as tested object manipulation tasks and conducted experiments.

A. Task environment

The tasks considered in this paper are manipulations of a single object, namely a sphere, box or capsule. This manipulation is specified as re-positioning or re-orienting the object with respect to the palm. Therefore, the task complexity can be defined as the number of different position and orientation goals that need to be achieved by the robotic hand, thereby minimizing the distance and angle to the desired object poses. Initially, the object can either be located on a flat plane surface or in midair. Domain randomization is incorporated to encourage more robust hand designs and motion plans. For a given task a fixed number of world states are created by uniformly varying the initial state of the manipulated object. An interval of position, orientation and friction deltas, as well as minimum and maximum multipliers for mass and scale can be specified depending on the task.

B. Optimization process

As the goal of this paper is to find the simplest hand design able to execute the task as accurately as possible, multiple optimizations with different basic hand structures are evaluated in a physics simulation following a bottom-up approach along a hierarchy. This hierarchy as seen in

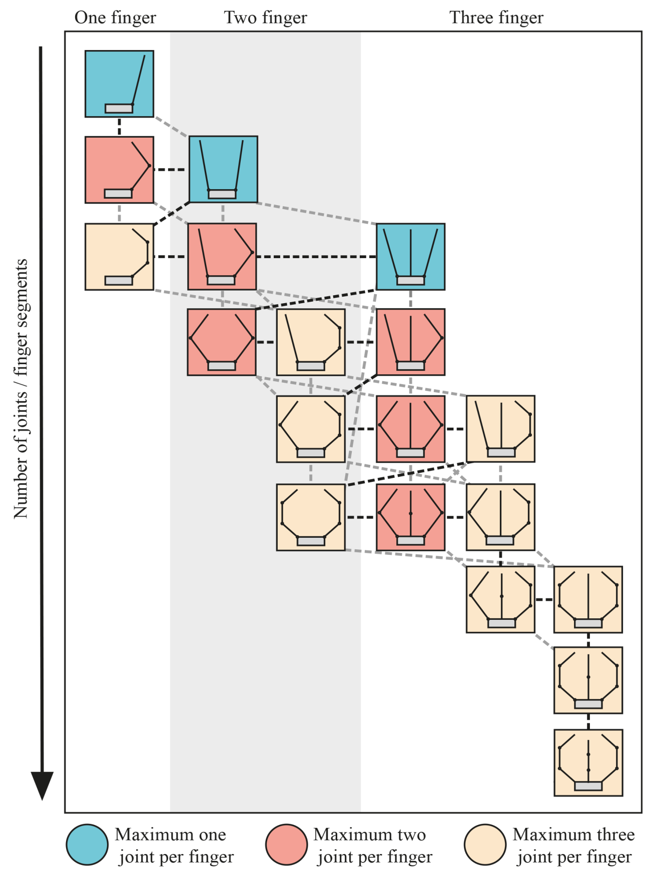
One finger
Two finger
Three finger
Number of joints / finger segments
Maximum one joint per finger
Maximum two joint per finger
Maximum three joint per finger

Fig. 1. This figure displays the hand structure hierarchy. A transition between the hand structures either adds/removes one joint/segment or rearranges them. The highlighted transitions model the bottom-up approach when starting with a one-fingered hand and choosing the next hand design such that rearrangement of fewer joints/segments is preferred over adding new ones. The number of degrees of freedom and the number of optimization parameter are always increasing.

figure 1 is based on the number of fingers and joints per finger. Only hinge joints having one degree of freedom are considered. To achieve a comparable amount of dexterity as hands with multi-DoF joints, the joint axes are allowed to differ inside a finger in contrast to human-like hands. As some tasks may not be achieved by the simplest hand structure, different starting points in the hierarchy can be chosen to save computational time. After each run the best result is evaluated on a specified number of different world states stopping when reaching a small predefined error value for all position and orientation goals.

While the discrete hand structure stays fixed during a run, the finger positions on the palm, segment lengths, friction coefficient, initial joint angles, axes and limits can be individually optimized in between specified minimum and maximum values depending on the task. The control policy is jointly optimized with the hand morphology. This paper primarily focuses on a simple control policy where piecewise constant joint torques are applied over defined time intervals. As the policy may have difficulty coping with gravity, compensating torques can also be calculated and added per time step. This policy, which will be referred to as a ”torque policy” throughout the paper, requires minimal calculation in simulation and therefore allows for faster optimization. In another approach desired target joint poses and coefficients of a proportional plus derivative (PD) controller with feed forward term are optimized [27]. The corresponding torques applied at the joints are computed with equation (1).

$$\begin{aligned}\tau = M(\theta)(\ddot{\theta}_d - K_v\dot{e} - K_p e) + C(\theta,\dot{\theta})\dot{\theta}_d + N(\theta,\dot{\theta}),\\ \text{with } e = \theta - \theta_d \wedge \dot{e} = \dot{\theta} - \dot{\theta}_d\end{aligned} \tag{1}$$

$M(\theta)$ denotes the inertia matrix of the manipulator, $C(\theta,\dot{\theta})\dot{\theta}_d$ contains Coriolis and centrifugal forces and $N(\theta,\dot{\theta})$ comprises a gravity compensation term. K_v and K_p are selected as diagonal positive definite gain matrices. The diagonal entries are called p coefficients for K_p and d coefficients for K_v. In contrast to the original formula in [27], the matrices are multiplied with $M(\theta)$ to make the p and d coefficients less dependent on changes of the morphology during optimization. θ denote the actual and θ_d the desired joint angle values.

C. Evolutionary Optimization

This hand generation process uses an evolutionary algorithm to simultaneously co-optimize the morphology and control parameters. CMA-ES [11] and MO-CMA [15] can be selected. As both covariance matrix adaption strategies can exceed set optimization parameter boundaries provided parameters are mapped to a continuous function in the set range [28]. Equation (2) displays a weighted penalty term which is used to punish high values τ_i of the torque policy greater than a term τ_p.

$$\begin{aligned}p_\tau = \begin{cases} w\,(\tau^* - \tau_p)\,, & \tau^* > \tau_p \\ 0, & \text{otherwise.}\end{cases}\,,\\ \text{with } w \in (0,1), \quad \tau^* = \max_i |\tau_i|\end{aligned} \tag{2}$$

This penalty is required because the torques are non-restricted. Also, moderate torques are preferred.

Let $p^{(i)} = (p_d^{(i)}, t_p^{(i)}), i \in (1,...,n)$ describe a specified position goal i of n containing its desired position $p_d^{(i)}$ at time step $t_p^{(i)} < t_p^{(i+1)}$. Equation (3) maps this position goal to the distance (either Manhattan or Euclidean) in space according to the actual position $p_{t_p}^{(k)}$ at time step t_p on world state k.

$$f_p^{(k)} : (\mathbb{R}^3, \mathbb{R}_{\geq 0}) \rightarrow \mathbb{R}_{\geq 0}, (p_d, t_p) \mapsto d(p_d, p_{t_p}^{(k)}) \tag{3}$$

Further, $p^{(0)} = (p_d^{(0)}, t_p^{(0)}) = (p_{init}, 0)$ is a position goal containing the initial position p_{init}. The corresponding position objective $o_{p^{(i)}}$ can contain either a single position goal $\{p^{(i)}\}$ or a trajectory of goal positions $\{\dots, \tilde{p}_{t_j}^{(i)}, \dots, p^{(i)}\}$ where $\tilde{p}_{t_j}^{(i)}$ is interpolated based on $p_d^{(i-1)}$ and $p_d^{(i)}$ at time step $t_j \in (t_p^{(i-1)}, t_p^{(i)})$. Let $T(o_{p^{(i)}}) = t_p^{(i)}$ be a function which is returning the time step of the current and $t(o_{p^{(i)}}) = t_p^{(i-1)}$ is returning the time step of the preceding goal $p_d^{(i-1)}$.

The orientation/rotation objective $o_{r^{(i)}}$ is constructed similarly where $r^{(i)} = (q_d^{(i)}, t_r^{(i)}), i \in (1, ..., m)$ is an orientation goal i of m containing the desired orientation $q_d^{(i)}$ as quaternion at time step $t_r^{(i)} < t_r^{(i+1)}$. Equation (4) maps the orientation goal to an angle between the desired q_d and actual orientation $q_{t_r}^{(k)}$ at time step t_r on world state k. $< q_{t_r}^{(k)}, q_d >$ denotes to the inner product of corresponding unit quaternion.

$$\begin{aligned} f_r^{(k)} &: (\mathbb{H}, \mathbb{R}_{\geq 0}) \rightarrow \mathbb{R}_{\geq 0}, \\ (q_d, t_r) &\mapsto cos^{-1}(2 < q_{t_r}^{(k)}, q_d >^2 -1) \end{aligned} \tag{4}$$

The corresponding objective function value results from equation (5) where o denotes either a position objective $o_{p^{(i)}}$ with $f_o^{(k)} = f_p^{(k)}$ or an orientation objective $o_{r^{(i)}}$ with $f_o^{(k)} = f_r^{(k)}$.

$$f_o = \begin{cases} 10.000, & \text{if COLLISION} \\ p_\tau + \frac{1}{|W|} \sum\limits_{k \in W} \underbrace{\frac{1}{|o|} \sum_{g \in o} f_o^{(k)}(g)}_{:= h_o^{(k)}}, & \text{otherwise.} \end{cases} \tag{5}$$

Each objective goal is averaged over all domain randomized world states W. Changing the hand morphology can cause initial collisions of the hand with itself or the environment. These collisions are therefore checked before the simulation and all objective values are potentially set to an unreachable constraint term preventing the optimization to return a non-collision-free result. This paper further introduces a more complex variant dividing the optimization into ten sub-problems per second using early termination to save computational time and potentially guide the simulation to relevant results. The individual optimizations are conducted in consequent order where at each end the position and orientation goals are evaluated for the current time step. The simulation is terminated if these values exceed a predefined margin s_o, either $s_o = d_s$ as maximal possible distance between actual and desired position or $s_o = \theta_s$ as maximal possible angle between actual and desired orientation. Accordingly, $h_o^{(k)}$ in eq. (5) is replaced with ${h'}_o^{(k)}$ displayed in eq. (6) where t_{stop} denotes the time step where the simulation is terminated. The remaining not evaluated goals are set to the maximum possible value based on a computed proportion a_o.

$${h'}_o^{(k)} = \begin{cases} h_o^{(k)}, \quad \text{if } t_o < t_{stop} \\ \frac{1}{|o|}(\underbrace{\frac{T(o) - t_{stop}}{T(o) - t(o)}}_{:= a_o} s_o + \sum\limits_{g_{t_i} \in o}^{t_i < t_{stop}} f_o^{(k)}(g_{t_i})), & \text{otherwise.} \end{cases} \tag{6}$$

As already stated, the single-objective CMA-ES requires a weighted sum of all objectives as fitness function to be used with multiple goals. Throughout this paper $w_{o_p} = 1$ and $w_{o_r} = 0.1$ are selected as respective goal trade-off. In comparison to single-objective, the results of the multi-objective algorithms are provided as pareto front containing the best individual for each goal and corresponding pareto-optimal trade-offs. Therefore, the best performing manipulator is determined by normalizing all position and orientation objectives between occurring minimum and maximum values in the optimal pareto front and selecting the solution with the smallest distance to zero.

D. Experiments

In this section multiple experiments are described to address different questions. The first sections focus on how the trajectory optimization [7] and the evolutionary approach relate. The following parts contains experiments to demonstrate the opportunities of the evolutionary approach to achieve robust manipulators, in particular the optimization of joint limits. The hand morphology/control and domain randomization parameter ranges as well as object and optimization specifications for all optimized manipulation tasks are displayed in Table I. On all experiments an initial standard deviation of 0.1 is specified for the covariance matrix adaption strategy and the mean vector computed based on the parameter interval boundaries. Furthermore, 0.005 is used as physics simulation step size.

Various whole hand results and motion plans obtained by the trajectory optimization [7] are directly transferred to this paper's physics simulation without applying domain randomization. The object manipulation tasks comprise horizontally/vertically rotating a capsule 90 degree in midair, rotating a sphere 180 degree clock-/counterclockwise and drawing a box with a capsule shaped pen. Since the motion plan is given as sequence of target joint angles at specific timestamps a PD controller with feed forward term and gravity compensation is used to follow this generated trajectory. This experiment is conducted to see if the trajectory optimization is able to produce results that can be reconstructed in a physics simulation.

The next experiment elaborates if the evolutionary approach can be used to further improve the trajectory optimization results by co-optimizing the morphology and controller. Only small deltas from the initial hand parameters and joint angles values for the PD controller are selected during optimization as the optimal design and motion is expected to be close the initial result. Rotating a capsule 90 degrees in midair *(Task 1)* is selected as primary example in this paper. The first experiment optimizes morphology along with the target joint angles from the already used PD controller. In the second experiment a simple torque policy replaces this controller. The torques are interpolated based on three optimized parameters. The trajectory optimization results already contain a desired object trajectory consisting of multiple position and orientation goals. As the number of goals is too large for multi-objective algorithms, CMA-ES is used for optimization. This single-objective algorithm is expected to have relevant results as the search space is significantly reduced which decreases the probability to converge to a local optimum. Applying domain randomization

TABLE I

PARAMETER AND PARAMETER RANGES FOR OPTIMIZATION EXPERIMENTS ON OBJECT MANIPULATION TASKS

	Parameter	Task 1	Task 2	Task 3	Task 4		Task 5	Task 6
Morphology	Initial joint angle (in rad)	[-0.5, 0.5]*		$[-\frac{\pi}{2}, \frac{\pi}{2}]$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$	$[-\pi, \pi]$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$	[-1, 1]
	Finger position palm[1](in cm)	[-1, 1]*		[-3, 3]	[-3, 3]	[-4, 4]	-	-
	Finger segment length (in cm)	[-5, 5]*		[1, 6]	[3, 10]^6	[1, 20]^6	[4, 9]	[4, 8]
	Joint axis angle (in rad)	[-0.5, 0.5]*		$[-\pi, \pi]$	$[-\pi, \pi]$		-	-
	Joint limits (in rad)	-		$[-\frac{\pi}{2}, \frac{\pi}{2}]$	-	$[-\pi, \pi]$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$
	Friction coefficient	[-0.3, 0.1]*		[0.2, 0.8]	[0.3, 1.0]		-	-
Control	torque (in Nm)	[-0.5, 0.5]^3	-	[-0.05, 0.05]	[-0.05, 0.05]		[-0.01, 0.01]^3	[-0.01, 0.01]
	Target joint angles (in rad)	[-0.3, 0.3]*^{3}	[-0.3, 0.3]*	-	-		$[-\frac{\pi}{2}, \frac{\pi}{2}]^3$	-
	P coefficients	[100, 5000]^3	[100, 5000]	-	-		[1, 100]^3	-
	D coefficients	[12.5, 625]^3	[12.5, 625]	-	-		[0.1, 10]^3	-
Domain R.	Object position* (in cm)	[-3, 3]^4		[-0.3, 0.3]^4	[-0.3, 0.3]^5		[-0.5, 0.5]^5	[-0.5, 0.5]^5
	Object orientation*^{2}(in rad)	[-1.5, 1.5]		[-0.15, 0.15]	[-0.35, 0.35]		[-0.4, 0.4]	[-0.3, 0.3]
	Object friction coefficient*	[-0.05, 0.05]		[-0.05, 0.05]	[-0.1, 0.1]		[-0.1, 0.1]	[-0.05, 0.05]
	Object scale**	[0.909, 1.1]		[0.909, 1.1]	[0.95, 1.05]		[0.66, 1.50]	[0.5, 2.00]
	Object mass**	[0.66, 1.50]		[0.66, 1.50]	[0.66, 1.50]		[0.66, 1.50]	[0.66, 1.50]
Object	Shape	Capsule	Sphere	Capsule	Box		Box	Box
	Measurement (in cm)	1.5L x 0.2r	0.35r	0.15L x 0.02r	0.04		0.04	0.04
	Mass (in kg)	0.2	0.1	0.02	0.03		0.03	0.03
	Friction coefficient	1.0	1.0	1.0	0.3		0.3	0.3
Evo. Alg.	Algorithm	CMA		MO-CMA	MO-CMA		CMA/MO-CMA	CMA
	Iterations	1000		1000	200	1000	100	100
	Population size	250		250	150	250	80	80
	World states	16		16	64	128	8, 64, 128	128
	Goal	Pose trajectory		Pose trajectory	Pose		Alignment	Orientation

[1] in x- and z-direction of hand-base-coordinate system [2] angle around defined axis [3] dependent on the applied control policy
[4] in x-, y- and z-direction of global coordinate system [5] in x- and z-direction of global coordinate system [6] interval divided by the number of joints belonging to the same finger * delta added to initial value ** scaling factor multiplied with initial value

to alter the object likely leads to initial collisions between object and robotic hand. Therefore, two optimizations are conducted with and without domain randomization where the optimization with uses the best performing result without as starting point. As second trajectory optimization result, the re-orientation of a sphere clock- and counter-clockwise around 180 degree is evolved (*Task 2*). In addition, early termination with $\theta_s = 0.5$ and $d_s = 0.5$ is applied for this optimization. It is notable that the trajectory optimization results and corresponding manipulated objects are designed ten times bigger than actual hand sizes. Other parameters are accordingly adapted to match this larger size.

In a subsequent attempt it is investigated if similar solutions to the trajectory optimization can also be obtained in a completely evolutionary based approach. Therefore the same task of rotating a capsule is similarly specified and optimized on an identical hand structure in a much bigger search space *(Task 3)*. The controller comprises a torque policy which applies interpolated torques based on three optimized parameters over a two second time frame. Early termination with $\theta_s = 0.3$ and $d_s = 0.02$ is applied for this optimization.

The next experiment examines if suitable results can be obtained when not provided with an initial hand structure. This approach follows the introduced hand hierarchy. The optimization starts with the simplest design of a finger with only one joint and stops when reaching a two-fingered hand with three joints instead of using a penalty term. The motion task evaluated is a rotation of a box by 45 degrees around its vertical axis while the object remains at its initial position on a flat plane *(Task 4)*. One constant torque is applied per joint for a duration of one second. This task is evaluated on a large deviation of the initial rotation to see if a robust manipulator can be built with this simple control strategy. Accordingly, the distribution of the initial and final object pose error is calculated on the same 15625 domain randomization samples to ensure the comparability of different hand structures. The optimization pipeline is performed twice on this task to exclude that some successful or poor performing manipulators are only found by coincidence. The two best hand structures are again optimized on a much larger iteration and population size as well as using a broader range of morphology parameters and a control policy with two interpolated torques over 1.5 seconds to see if the results can be improved if the search space as well as the optimization duration is increased.

The last experiment explores if optimizing for specific joint limits can effect the robustness of designed manipulators on the following two tasks. The goal of the first task involves the alignment of a box at a specified position by applying two different constant torques per 1.5 seconds while the object is lying on a flat surface *(Task 5)*. The objective of the second task is a 45 degree rotation of the same object while the final position is irrelevant *(Task 6)*. For both tasks the hand structure is fixed having two fingers with two hinge joints each. The design space of the hand morphology considered is two-dimensional and only segment lengths, initial joint angles and limits are optimized. Multiple experiments are performed on different numbers of randomized world states and both covariance matrix adaption strategies are used to assess their quality in finding joint limits. Furthermore, target joint angles for a PD controller with feed forward term are evolved for the first task to compare its results to the simpler torque policy.

IV. Results and Discussion

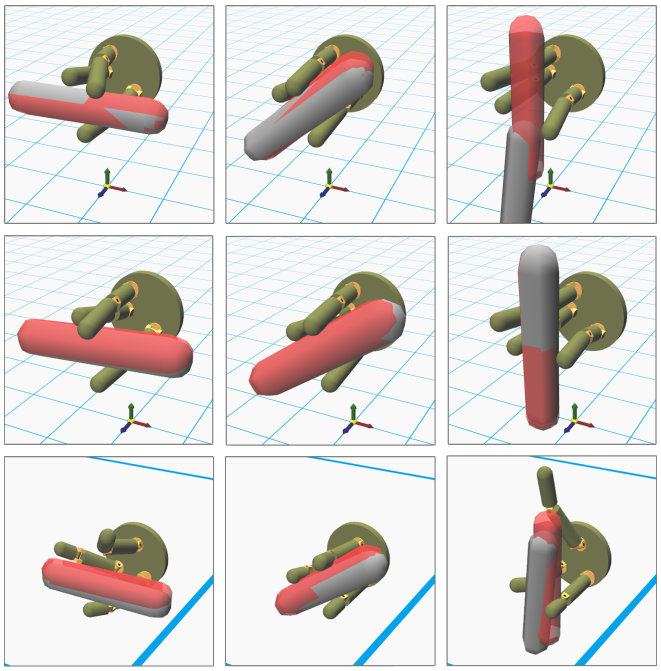

Fig. 2. This figure shows different optimization results to rotate a capsule 90 degrees in a physics simulation. The first motion *(top)* contains the hand design and motion plan received from a trajectory optimization [7]. As in many other cases, this direct transfer leads to failure as required forces can not be matched and the object is slipping out of the hand. The following motion *(mid)* shows that further optimizing this result within a simulation using an evolutionary algorithm has excellent results. A similar robust and successful hand design and motion is not as easily obtained in a completely evolutionary based approach *(bottom)*.

This section demonstrates the weakness of the current trajectory optimization [7] and the advantage of using evolutionary algorithms to optimize a robust hand morphology on object manipulation tasks. All results from the different experiment setups are displayed in motion in the attached video.

Directly transferring the hand design and joint angle based motion plan from the trajectory optimization pipeline [7] into a physics simulation does not work in most cases. Utilizing a hybrid approach to further optimize both morphology and the PD controller leads to more successful results. Figure 2 provides both outcomes for an example task. Another experiment replacing the PD controller with a simple torque policy provides a comparable result. Therefore, our approach to optimize simpler policies seems like a solid plan and may also work on more complex tasks. Figure 2 also contains the best result for the evolutionary approach from scratch. The final state is not robust but the manipulator is able to mostly follow the desired trajectory. It is noted that of five optimizations only this produces such a good result. Three other manipulators just use two of the three available fingers and therefore are not able to closely perform the manipulation. Therefore, future work has to improve the exploration of the search space. In contrast, when utilizing an initial good seed for the robot hand and control policy a much lower search space needs to be addressed and therefore fewer iterations in a time consuming physics simulation are needed to obtain significant results. Furthermore, using a PD controller leads to better motions than just applying optimized torques but as further experiments show optimizing this control policy in the whole joint value space is much more complex.

The performance of solutions for the hand hierarchy optimization are visualized in figure 3. One-finger-hands perform worst on the specified task as they are only able to change the orientation while also altering the position of the box. Potentially applying multiple torques can improve their results. Two-finger-hands with at least three joints on one finger are able to achieve significant results which are close to the initial deviation of position and orientation. Surprisingly, a non-symmetric hand containing a finger with one and a finger with three joints drastically outperforms a symmetric two-joint-fingered-hand on this task even though the degrees of freedom are the same. Adding another finger to the same hand yields worse result which justifies this papers bottom up approach when performing tasks which may not need additional fingers to be robust. Overall, hands with three fingers do not provide successful results on this task. The reason is not only the increased complexity when adding more degrees of freedom. The additional finger also hinders the other two fingers from performing well. In most solutions the extra finger is chosen to be as small as possible and lifted immediately. Having to deal with an additional obstacle in the beginning may lead to convergence to a local minimum. As each optimization for a hand structure is executed twice and most results do not diverge by much, there is a slightly higher probability that these outcomes are representative for the given specified manipulator and controller settings. Applying a more extensive evolutionary optimization to the two-fingered one-three- and two-three-joint hand structure yields more robust results that are able to nearly reduce the optimization error to zero. However, the position error does not improve as decreasing the rotation error is achieved by aligning the object with both fingers on two sides still leaving two possible movement directions. Probably more finger or a better exploited morphology are required to also improve the position error. Overall, this reveals that doing a more extensive search on the optimization pipeline is required to find more robust solutions as the convergence rate to achieve these results is slow. This decrease the performance of this approach by a lot and therefore future work has to improve the current optimization pipeline. Varying the number of iterations/populations based on the degree of freedom is a possibility.

Figure 4 and 5 show the simulation results for the joint

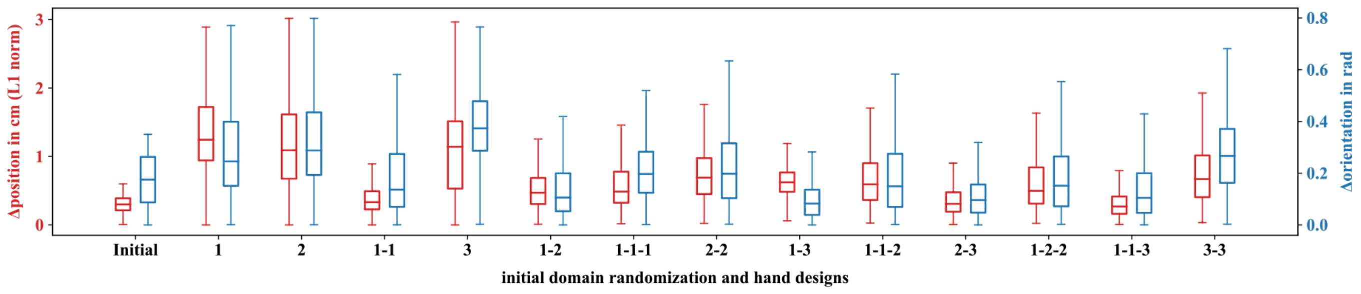

Fig. 3. This figure displays the distribution of the initial object pose and corresponding final object poses from optimized manipulators executed on task 4. The hand structures are ordered in accordance with the proposed hierarchy and the digits denote the number of joints per finger.

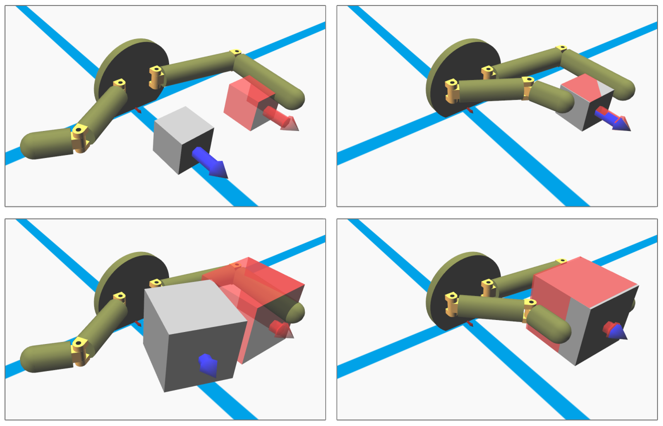

Fig. 4. This figure displays two example motions results *(right)* for different domain randomization *(left)* of task one. The two optimized joint limits in the left finger cooperate to align with the side of the box. The red cube with the arrow symbolizes the desired goal pose.

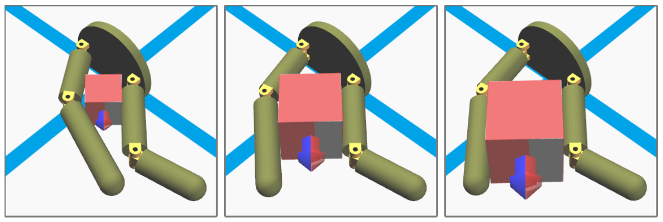

Fig. 5. Three different final results for task two. The optimized joint limit of the first segment in the left finger aligns with the desired orientation of the box.

limit optimization experiments. In both tasks specific joint limits are exploited to align the finger with the desired object pose to be more robust on different initial object sizes and poses. These results may not look interesting at first as they are only built for a specific goal. However, if the resulting manipulators and motion plans are very robust on randomized domains, moving or rotating the base of the hand allows to accomplish any desired goal pose robustly with the same optimized joint limits. The joint limits form an obstacle which stops the object motion in its tracks. Therefore, no opposing force from the finger is needed to granular position an object. This is significant because a simple controller such as constant torque is not able to do this very well. The joints limits are only found in about 50% of the trials when optimizing on a larger number of simulated domains. Results simulated on boundary points of the domain randomization intervals especially for scaling and orientation are rather poor. Thus, more domain randomization samples may lead to convergence to local minimum because the mean fitness on the uniformly distributed samples is optimized. Evaluating other strategies to combine these fitness values such as weighting them based on their difference to the initial values to improve overall result is part of future work. Further experiments on the same tasks also show mostly poor outcomes when optimizing a PD controller in comparison to a torque policy. Finding optimal target joint angles in the whole space while also modifying the morphology is very difficult and this problem needs to be addressed. Another deficit of the PD controller is the number of calculations required per time step which lead to at least three times slower computation scaling with degrees of freedom.

V. Conclusion

This paper provides a basis for future work to create simple and robust task-specific hand designs for object manipulation. A general framework is introduced applying evolutionary strategies in a physics simulation to co-optimize hand morphology and controller based on a high-level task description. The optimization is performed simultaneously on varying world states using domain randomization to account for uncertainty and improve robustness. Our approach further shows that optimizing joint limits can support a simple control policy to be very robust to deviations on specific tasks. Evolutionary algorithms are able to find arguably good results, but are still restricted because of necessary exploration of a large search space and a small number of relevant solutions. The next steps involve optimizing robust manipulators for more complex manipulation tasks. Increasing the number of iterations may not automatically lead to better results as evolution selects the best performing individuals on the short term and therefore potentially converges to a local optimum [8]. Therefore, acquiring a higher diversity in the population is an important future goal. Literature provides potential approaches adapting basic evolutionary algorithms or adding additional objectives to balance convergence and diversity.

The fact that our optimization takes place in a simulated environment makes planning extremely difficult and in many cases prohibits us from effectively optimizing target joint positions for a PD controller to follow. We believe that a hybrid approach that fuses the trajectory optimization [7] with the simulation based optimization of this work can make this possible. This framework will essentially use a trajectory optimization based motion planner and morphology optimizer to provide good initial seeds for the physics based optimization. The physics simulation will further fine-tune the hand design and control policy to accomplish the desired task in a variety of different domains.

Future work also includes building some of the optimized robot hands and testing them to evaluate whether the current approach is sufficient to create robust manipulators that generalize well. In addition, further deviations can be introduced as the current model of the world is still imperfect. Adding more types of perturbations may encourage robustness to some variations and errors that are not considered.

VI. Acknowledgements

I would like to offer my special thanks to Prof. Tamim Asfour at KIT who encouraged me to apply for the exchange program with CMU. He provided valuable insights and supported me during my semester abroad.

Further thanks goes to the Continuous Learning in International Collaborative Studies (CLICS) exchange program allowing me to spent half a year at the Robotics Institute at Carnegie Mellon University.

VII. Appendix: Implementation and Hardware Details

This framework is implemented in C++ using the Open Dynamics Engine to simulate physics and applying evolutionary algorithms from the open-source shark-ml library [29]. Simulations with about 300 iterations, a population size of 150, 128 domain randomization and a physics simulation step size of 0.005 using the simple torque based control policy take about 60-120 minutes when executed on all eight cores of a laptop with a i7-6700HQ CPU with 2.6GHz and 16GB RAM.

References

[1] R. R. Ma and A. M. Dollar, "On dexterity and dexterous manipulation," in *2011 15th International Conference on Advanced Robotics (ICAR)*, pp. 1–7, June 2011.

[2] Y. Bai and C. K. Liu, "Dexterous manipulation using both palm and fingers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1560–1565, May 2014.

[3] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. W. Pachocki, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *CoRR*, vol. abs/1808.00177, 2018.

[4] L. U. Odhner and A. M. Dollar, "Dexterous manipulation with underactuated elastic hands," in *2011 IEEE International Conference on Robotics and Automation*, pp. 5254–5260, May 2011.

[5] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.

[6] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, (Goslar Germany, Germany), pp. 137–144, Eurographics Association, 2012.

[7] C. Hazard, N. Pollard, and S. Coros, "Automated design of manipulators for in-hand tasks," pp. 1–8, 11 2018.

[8] S. Doncieux, N. Bredèche, J.-B. Mouret, and A. E. Eiben, "Evolutionary robotics: What, why, and where to," *Front. Robotics and AI*, vol. 2015, 2015.

[9] J. Bongard, "The utility of evolving simulated robot morphology increases with task complexity for object manipulation," *Artificial Life*, vol. 16, no. 3, pp. 201–223, 2010.

[10] T. Salimans, J. Ho, X. Chen, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *ArXiv*, vol. abs/1703.03864, 2017.

[11] N. Hansen, "The CMA evolution strategy: A tutorial," *CoRR*, vol. abs/1604.00772, 2016.

[12] V. Trianni and M. Lpez-Ibez, "Advantages of task-specific multi-objective optimisation in evolutionary robotics," *PLOS ONE*, vol. 10, pp. 1–27, 08 2015.

[13] T. F. Nygaard, E. Samuelsen, and K. Glette, "Overcoming initial convergence in multi-objective evolution of robot control and morphology using a two-phase approach," in *Applications of Evolutionary Computation* (G. Squillero and K. Sim, eds.), (Cham), pp. 825–836, Springer International Publishing, 2017.

[14] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, pp. 1–28, March 2007.

[15] T. Vo, N. Hansen, and C. Igel, "Improved step size adaptation for the mo-cma-es," *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 07 2010.

[16] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, pp. 577–601, Aug 2014.

[17] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 712–731, Dec 2007.

[18] S. Rodrigues, P. Bauer, and P. A. Bosman, "A novel population-based multi-objective cma-es and the impact of different constraint handling techniques," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, (New York, NY, USA), pp. 991–998, ACM, 2014.

[19] O. R. Castro, R. Santana, J. A. Lozano, and A. Pozo, "Combining cma-es and moea/dd for many-objective optimization," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1451–1458, June 2017.

[20] Z. Cui, Y. Chang, J. Zhang, X. Cai, and W. Zhang, "Improved nsga-iii with selection-and-elimination operator," *Swarm and Evolutionary Computation*, vol. 49, pp. 23 – 33, 2019.

[21] J. Lehman and K. O. Stanley, "Evolving a diversity of virtual creatures through novelty search and local competition," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, (New York, NY, USA), pp. 211–218, ACM, 2011.

[22] J. Lehman and K. O. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary Computation*, vol. 19, no. 2, pp. 189–223, 2011. PMID: 20868264.

[23] N. Cheney, J. Bongard, V. SunSpiral, and H. Lipson, "Scalable co-optimization of morphology and control in embodied machines," *CoRR*, vol. abs/1706.06133, 2017.

[24] J. Auerbach and J. Bongard, "Dynamic resolution in the co-evolution of morphology and control," 06 2010.

[25] J. Auerbach and J. Bongard, "Evolving complete robots with cppn-neat: The utility of recurrent connections," pp. 1475–1482, 01 2011.

[26] H. Lipson, V. Sunspiral, J. Bongard, and N. Cheney, "On the difficulty of co-optimizing morphology and control in evolved virtual creatures," *The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, no. 28, pp. 226–233, 2016.

[27] Z. L. R. Murray and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. 1994.

[28] M. Rommerman, D. Kuhn, and F. Kirchner, "Robot design for space missions using evolutionary computation," in *2009 IEEE Congress on Evolutionary Computation*, pp. 2098–2105, May 2009.

[29] C. Igel, V. Heidrich-Meisner, and T. Glasmachers, "Shark," *Journal of Machine Learning Research*, vol. 9, pp. 993–996, 2008.