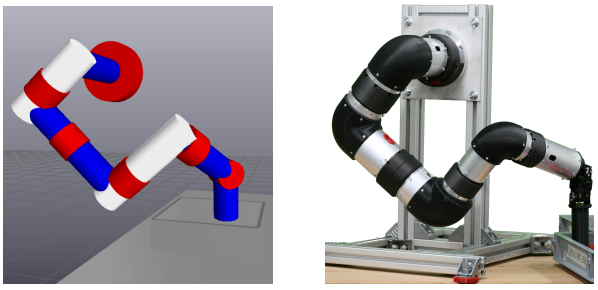
2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Robust co-design of robots via cascaded optimisation

Akhil Sathuluri, Anand Vazhapilli Sureshbabu and Markus Zimmermann

***Abstract*— Optimising mechanical, control and actuator design variables together as a co-design problem enables identifying novel and better-performing robot architectures. Typically, solving such problems using conventional optimisation methods yields a single, point-based solution. Deviating from the computed optima may be necessary to ensure physical feasibility, typically associated with a performance loss. In this work, we present a two-step cascaded optimisation approach to identify non-intuitive designs and recover the loss in performance by constructing a *solution space*. The solution space provides robustness in the form of permissible ranges of design variable values and enables the selection of a physically feasible design. In our study, we observe (1) up to 20% of the lost performance is recovered and (2) an improvement of 30% on the task metric in comparison to an existing robot and (3) designs with cost savings of up to 10% can be identified.**



(a) (b)

Fig. 1: (a) Simulation model of the robot design resulting from the cascaded optimisation. (b) Realised prototype of the physically feasible design.

I. Introduction

Robots are often needed to meet domain-specific requirements while not compromising performance. Traditionally, such robots are designed using a *bottom-up* approach. Bottom-up design generally involves expensive iterations between different sub-systems like mechanical design, actuator selection and control synthesis [1]. Moreover, it is primarily experience-driven, potentially limiting the opportunity to explore novel designs due to inherent bias. Physically realising a robot introduces challenges in terms of manufacturing tolerances and availability of components. Due to the non-linear interaction between various sub-systems, understanding trade-offs in the design process is non-intuitive. Attributes like energy efficiency and task performance of a robot are highly dependent on the choice of actuators and the kinematics and dynamics of the robot. This highlights the necessity for a systematic design approach to identify physically feasible designs robust to variations in the realised design variable values. A *top-down* design procedure takes into account interaction between design variables (DVs) of various sub-systems and their performance metrics or quantities of interest (QoIs) with respect to a given high-level task. As demonstrated in [2]–[5], co-design optimisation yields better-performing results as compared to conventional robots. Classical optimisation techniques such as gradient-based optimisers used in [6] or evolutionary algorithms in [7] are employed to solve such co-design problems. These optimisations result in identifying a single optimal design, i.e., a point in the design space. Deviating from the computed optimal point design may have a significant depreciating effect on the robot's performance. In such cases, designs are not robust to variations in the realised design variable values.

In this work, we propose a cascaded optimisation scheme to identify optimal designs and improve their robustness. This alleviates the problem of physical feasibility of point-based designs and simplifies the prototyping of physical robots as shown in Fig. 1b. Physical feasibility implies the existence of components like motor, transmission and manufacturing links.

The approach consists of two sequential optimisation steps summarised in Fig. 2. The first step is similar to the previously discussed co-design approaches involving the combined optimisation of mechanical, actuator and control parameters given a high-level task, which results in a point-based design $\boldsymbol{x}^*$. Next, a physically feasible design $\boldsymbol{x}_p$ *close* to the obtained optimal design is chosen. A second optimisation step is then performed to maximise the permissible ranges of all the design variables without compromising the performance of the design $\boldsymbol{x}_p$, resulting in a so-called *solution space*.

The robot chosen in this study is motivated by the increasing demand for using robots in industrial applications like bin-picking, assembly, production lines, logistics etc. A custom humanoid arm, including its mechanics, actuator, and controller subsystems, is designed for a particular task ensemble while optimising on metrics like time and energy efficiency. The task ensemble in the current example is a set of five sampled, independent pick-and-place sub-tasks with their associated constraints described in Section III.

Our core contribution is the cascaded optimisation scheme to identify optimal designs and compute solution spaces, which improves the computed design's robustness against variations in the realised DV values. In this sense, the current

The authors are with the Robot Systems group, Laboratory for Product Development and Lightweight Design, TUM School of Engineering and Design, Technical University of Munich (TUM), Germany akhil.sathuluri@tum.de

method is compatible and augments all the point-based co-design approaches proposed in the literature. As a further contribution, we provide the *X-Ray toolbox*[1] to interpret the non-linear design space of robots. The toolbox allows the designer to manipulate DV ranges, allowing a trade-off between different designs in the solution space resulting from the cascaded optimisation scheme. Implementation details and further information regarding the X-Ray toolbox can be found in [8].

To demonstrate the advantages of the top-down procedure, we compare the resulting robot with the Blue robot [9] as a reference design. The latter is developed using traditional bottom-up design with off-the-shelf components, using expert knowledge and hence would be a suitable comparison to a robot resulting from the proposed optimisation procedure. As shown in Fig. 3, similar to the robot in the current study, the Blue robot has 7-DoF robot, unlike our previous generation 5-DoF robot.

II. Related Work

1) Co-design of robots: The combined optimisation of mechanical design, motion planning and actuator placement for quadrupeds presented in [10], [11] resulted in identifying efficient robot motions. Further, the work exhibited in [12], [13] demonstrates hardware-focused co-design of robots for a higher-level task. The study in [14] explored robust co-design in the context of control system formulation. The parametrisation of the robot in the current work is related to the work done in [15], where a bi-level optimisation scheme is employed to achieve robot designs resulting in significant energy savings for a monoped jumping task. Moreover, the task definition considered in the current work is close to the one in [16], where the robot, environment and tool design were optimised, resulting in a non-intuitive design with low torque requirements for a set of pick and place tasks.

2) Solution space computation: The concept of solution space computation has numerous benefits while dealing with a system consisting of several coupled sub-systems. The work done in [17] deals with simplifying a complex problem by breaking down the system-level requirements into component-level requirements using solution spaces. Further, in [18], steering control of an autonomous vehicle is developed by constructing the solution spaces using requirements on the closed-loop system expressed as boundaries of the stability regions. Various system elements were decoupled in [19], allowing for their concurrent design. To the authors' best knowledge, this paper is the first to present a solution space approach in the context of co-design of robots using a cascaded optimisation approach.

III. Methodology

A. Problem overview

The first step of the cascaded optimisation scheme is formulated as a concurrent optimisation problem involving the following DVs from mechanical, actuator and control sub-systems:

- Length of the links l_i, offsets d_i and joint masses m_i, as, $\boldsymbol{x}_M = [\mathbf{l}^\top, \mathbf{d}^\top, \mathbf{m}^\top]^\top$
- Maximum torque, angular velocity and the motor constant, grouped as, $\boldsymbol{x}_A = [\boldsymbol{\tau}^\top, \boldsymbol{\omega}^\top, \boldsymbol{K}_m^\top]^\top$
- Gear ratio, $\boldsymbol{x}_T = [\mathbf{N}]$
- Control gains, $\boldsymbol{x}_C = [\boldsymbol{K}_p^\top, \boldsymbol{K}_d^\top]^\top$

An offset d_i between two adjacent links is the distance between their corresponding link axes. To capture the explicit effects of the controller on the design space, an inverse dynamics controller parametrised in terms of the controller gains, $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$, is chosen. However, one could employ a bi-level optimisation scheme to formulate the co-design problem as an optimal control problem as previously mentioned in Section II. The first step, or the co-design step, involves minimising the *objective meta function* described in problem (1) to identify optimal design variable values for a

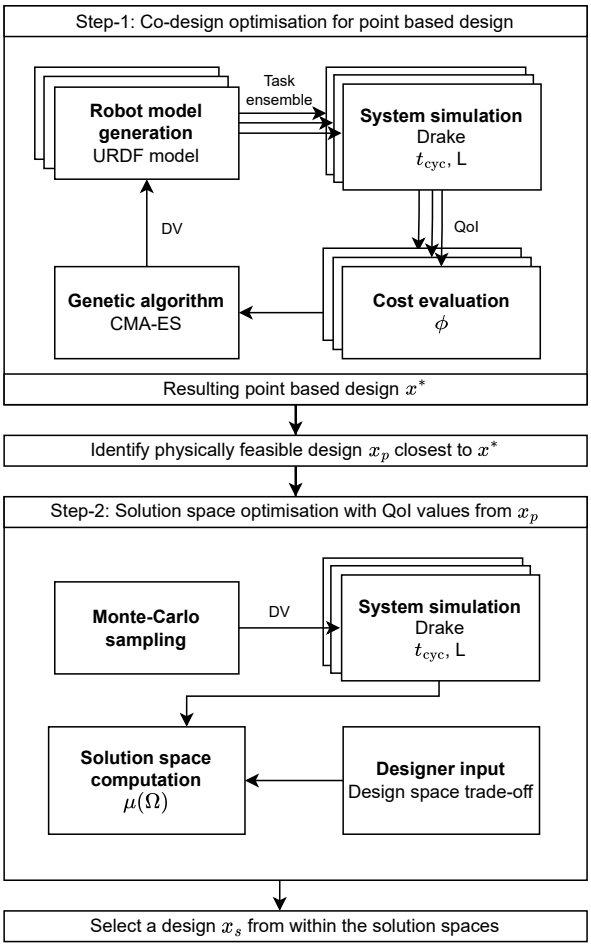
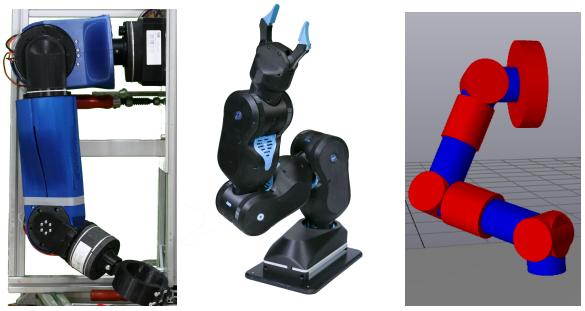


Fig. 2: Flowchart of the cascaded optimisation approach. Step-1 or the co-design step, identifies an optimal point solution followed by step-2 or the solution space optimisation step to compute the maximum permissible range of DVs.

[1]https://github.com/akhilsathuluri/x-ray-tool



(a) (b) (c)

Fig. 3: Typical kinematics of a humanoid robot arm with collinear consecutive link axes. (a) Our previous generation robot (5 DoF), and (b) the Blue robot [9]. (c) Equivalent robot model in Drake [20] simulation environment (7 DoF).

particular task ensemble.

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \phi(\mathrm{t_{cyc}}, \mathrm{L}), \\ \text{subject to,} \ & h(\boldsymbol{x}) = 0, \\ & \boldsymbol{x}_l \leq \boldsymbol{x} \leq \boldsymbol{x}_u \end{aligned} \tag{1}$$

where $\boldsymbol{x}_l$ and $\boldsymbol{x}_u$ are the considered upper and lower bounds of the design space. The equality function $h(\cdot)$ is explained in Section III-C. The objective function ϕ accounts for two QoIs, i.e., the total energy consumed L and the cycle-time $\mathrm{t_{cyc}}$ taken to complete a given task ensemble. Where $\boldsymbol{x}$ is the set of all the design variables $[\boldsymbol{x_M}^\top, \boldsymbol{x_A}^\top, \boldsymbol{x_T}^\top, \boldsymbol{x_C}^\top]^\top$. An evolutionary algorithm is used to solve the optimisation problem formulated in (1), resulting in a point-based optima $\boldsymbol{x}^*$.

One might not realise the computed specifications of design $\boldsymbol{x}^*$ due to the availability of transmission, motors etc. Therefore, the *closest* physically feasible design $\boldsymbol{x}_p$ is chosen, resulting in a compromise in performance as compared to design $\boldsymbol{x}^*$. Note that this compromise is *necessary* to be able to build a physical robot. As mentioned in [9], typically, components are selected with specifications closest (in an $L2$-norm sense) to the computed DV values. Additionally, a weighted norm accounting for the relative importance of the components can also be employed.

As a second optimisation step, we exploit the coupled design space of the co-design problem to identify the largest solution space containing variations in all DVs that are at least as performant as the chosen physically feasible design $\boldsymbol{x}_p$. This can be mathematically formulated as,

$$\begin{aligned} \max_{\boldsymbol{\zeta}} \ & \mu(\Omega(\boldsymbol{\zeta})), \\ \text{such that,} \ & \mathrm{t_{cyc}} \leq \mathrm{t_{cyc}}(\boldsymbol{x}_p), \\ & \mathrm{L} \leq \mathrm{L}(\boldsymbol{x}_p) \end{aligned} \tag{2}$$

where $\boldsymbol{\zeta} = (x_{1,l}, x_{1,u}, \ldots, x_{d,l}, x_{d,u})$, contains permissible upper and lower bounds of all DVs, $1 \ldots d$. Further, $\Omega = [x_{1,l}, x_{1,u}] \times \cdots \times [x_{d,l}, x_{d,u}]$, represents the solution space as a Cartesian product of individual DV ranges and $\mu(\Omega)$ is the size of the corresponding solution space. This problem results in the largest range of permissible design variable values. All designs within the obtained range are at least as good as $\boldsymbol{x}_p$ in terms of their QoIs, so a selected design $\boldsymbol{x}_s$ from this range is robust to any variations. An illustration of the proposed cascaded optimisation scheme is given in Fig. 2. Statistical analysis and a discussion on the connectedness of such a solution spaces is presented in [21].

B. Quantities of interest

The objective function in (1) is chosen to include two opposing quantities of interest, i.e., the energy consumed and the time taken to complete a given task, k.

1) Energy losses: The energy expended by the motor as heat to generate the required torque is computed using the associated motor constant, K_m. The total energy expended by the robot at all its joints is given by,

$$\mathrm{L}^k = \int \sum_i (\frac{\tau_i}{K_{m,i}})^2 dt,$$

where τ_i and $K_{m,i}$ are the motor torque and motor constant of joint i. Since energy consumed is related to the realised torque, this QoI penalises designs which need high joint torque to complete a task.

2) Cycle-time: The cycle-time is defined as the time taken by the robot to reach a desired joint pose,

$$\begin{aligned} & t^k_{\mathrm{cyc}} = \min \zeta, \\ \text{such that,} \ \forall \ & t > \zeta, \\ & \| \frac{\boldsymbol{q}_d - \boldsymbol{q}_r(t)}{\boldsymbol{d}_{ref}} \| < e_c \end{aligned}$$

where $\boldsymbol{q}_d$ and $\boldsymbol{q}_r$ is the desired and realised poses of the end-effector, $\boldsymbol{d}_{ref}$ is a reference distance used as a normalising factor, and e_c is a user chosen distance threshold. This QoI forces the robot to complete the task in a finite time.

C. Modeling and parametrisation

1) Robot system parametrisation: Starting from a typical architecture of a humanoid robot arm shown in Fig. 3, a parametrised simulation model is generated as shown in Fig. 3c. The blue, white and red elements represent links, offsets and joints. Unlike in [22], we assume fixed geometry for the links throughout the study. From the chosen mechanical design variable values $\boldsymbol{x}_M$, the inertial data is computed from the assumed cylindrical link geometry. Additionally, the localised masses of the joints are considered explicit design variables.

2) Actuator model: To account for the interdependence between different attributes of the actuator, we follow the methodology proposed in [23]. We use the heuristics constructed in [15], relating the motor mass and its motor constant and torque output. Further, motor mass is considered half of the sampled joint mass value similar to [14].

$$\tau_{max} = 5.48 \left(\frac{m}{2}\right)^{0.97},$$
$$K_m = 0.15 \left(\frac{m}{2}\right)^{1.39}.$$

These equations are imposed as equality constraints $h(\boldsymbol{x})$ for problem (1). The imperfect nature of such heuristics means that the resulting optima cannot be exactly realised as discussed in Section I,

To realise an accurately sized, physically feasible actuator, it is essential to model the velocity-dependent behaviour of the motor. One could model such an effect using non-linear transformations, as done in [24] or by adding it as a constraint during the motion planning phase as done in [25]. In the current work, this behavior is captured via linear mapping between the output torque and motor velocity to derive `safety_limits`[2] as used in PR2 control manager [26]. Due to imposed state dependant actuator limits, the cycle-time computation is non-trivial and no longer corresponds to the settling time of a second-order system.

D. Task definition

The task is an ensemble of five pick and place movements representing an object re-sorting scenario. The robot is required to transfer randomly orientated objects between two random points corresponding to a pick and a drop bin, placed over a table as shown in Fig. 1b. For each task, the following conditions are imposed on the end-effector (eef) as equality constraints, $h(\boldsymbol{x})$ stated in problem (1).

1) Orientation is constrained to match the orientation of the manipulated object by aligning their respective normals at both the pick and place locations.

$$\boldsymbol{z}_{\text{eef}}^{\text{pick}} - \boldsymbol{z}_{\text{object}}^{\text{pick}} = \mathbf{0},$$
$$\boldsymbol{z}_{\text{eef}}^{\text{place}} - \boldsymbol{z}_{\text{object}}^{\text{place}} = \mathbf{0},$$

   where $\boldsymbol{z} \in \mathbb{R}^3$ corresponds to the local z-axis of the eef and the object in the global coordinate system.
2) Position is constrained to reach both the pick and place locations to complete the task.

$$\boldsymbol{p}_{\text{eef}}^{\text{pick}} - \boldsymbol{p}_{\text{object}}^{\text{pick}} = \mathbf{0},$$
$$\boldsymbol{p}_{\text{eef}}^{\text{place}} - \boldsymbol{p}_{\text{object}}^{\text{place}} = \mathbf{0},$$

   where $\boldsymbol{p} \in \mathbb{R}^3$ corresponds to the end-effector position in the global coordinate system.
3) Velocities of the end-effector at the pick and place locations are,

$$\dot{\boldsymbol{p}}_{\text{eef}}^{\text{pick}} = 0,$$
$$\dot{\boldsymbol{p}}_{\text{eef}}^{\text{place}} = 0.$$

The imposed constraints only admit robot designs that can achieve the required end-effector poses. The root mean square of the individual cycle-times $\mathrm{t}_{\text{cyc}}^k$ and energy losses L^k are used to compute the total cycle-time t_{cyc} and energy loss L corresponding to the task ensemble.

[2]http://wiki.ros.org/pr2_controller_manager/safety_limits

IV. Results

A. Numerical results

1) Co-design optimisation step: The objective function for the first optimisation step maps the total cycle-time and energy loss to a scalar quantity as

$$\phi = [\mathrm{t}_{\text{cyc}}, \mathrm{L}]^\top \; K_w \; [\mathrm{t}_{\text{cyc}}, \mathrm{L}],$$

where K_w is a diagonal matrix with the weighting factors for t_{cyc} and L as $1e6$ (J/s) and 1 respectively.

The CMA-ES optimiser developed in [27], is used to solve the optimisation problem (1), by setting the number of iterations as 1000 and a population size of 50. As illustrated in Fig. 2, a URDF file corresponding to the sampled design variable values is constructed programmatically using the `Odio URDF` library [28]. Following this, the robot model is simulated using Drake toolbox [20] to compute the objective function value.

Optimisation problem (1), results in a point-based design, $\boldsymbol{x}^*$ shown in Fig. 6a. As mentioned in Section I, the Blue robot developed in [9], is chosen as a representative of the bottom-up, experience-driven design process. The Blue robot is chosen only to contrast the design strategy and is not designed for the same QoIs as in the current study. A quantitative comparison between the robots is presented for the root mean squared torque required to complete the task ensemble described in Section III-D. Simulations were run with 15 different seeds to generate random pick, place locations and orientations. The corresponding mean and standard deviation of QoI, L of different robots is presented in Fig. 4. The lower mean torque of design $\boldsymbol{x}^*$ can be attributed to its unique kinematics resulting from optimising over the specified tasks ensemble. This can be explained qualitatively by the fact that the robot can *counterbalancing* itself. The resulting architecture allows the robot to partially balance its weight with respect to the first joint, which moves most of the robot's weight as shown in Fig. 3c. This starkly differs from a conventional robot arm, where the robot's architecture resembles a cantilever beam, causing the first joint to support large loads in fully stretched poses.

As suggested in [15], one could custom-build components for the robot that satisfy the required specification. However, custom development of motors and gearboxes is expensive and requires expertise, precision and manufacturing capabilities. Alternatively, one could choose off-the-shelf components resulting in design $\boldsymbol{x}_p$, close to the specification of the optimal design; but with reduced performance as compared to $\boldsymbol{x}^*$. This compromise in the robot's performance is necessary to ensure the physical feasibility of the robot. Interestingly, since we deal with a co-design problem, this necessary compromise could be exploited by identifying what this compromise means with respect to the other design variables.

2) Solution space construction: Specifically, the motor mass, torque and gearbox combinations required by design $\boldsymbol{x}^*$, depicted in Fig. 5 as ●, cannot be realised by any available off-the-shelf components. Therefore, an actuator

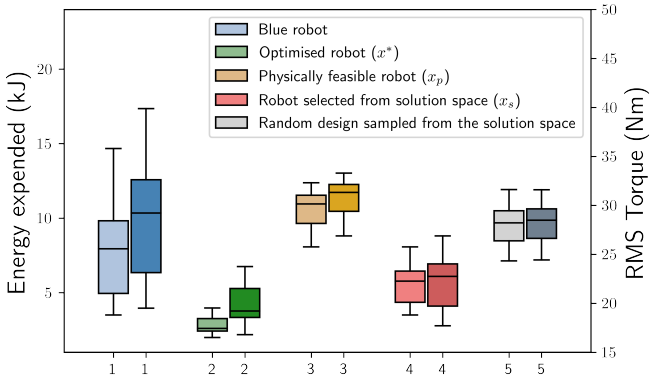


Fig. 4: A comparison between energy expended QoI (left boxes) and RMS torque required (right boxes) to perform 15 randomly sampled pick and place tasks.

closest to this specification is chosen. In Fig. 5, chosen motor (T-motor, AK10-9v2.0[3]) is marked as ▲. Along with the motor, the designer's input is incorporated to modify certain mechanical DVs resulting in design $\boldsymbol{x}_p$, shown in Fig. 6b. The depreciation in performance of design $\boldsymbol{x}_p$ is tabulated in Table I. Since step-1 deals with combined design space, this deviation from the optima can translate to performance loss due to changes in other design variable values. Therefore, several other designs could correspond to the same performance reduction. For example, choosing a low-performing motor could be equivalent to modifying certain link lengths or changing the motor placement within the robot.

Computing the maximum number of such equivalent or better designs can thus be posed as an optimisation problem of finding the extensive set of designs that are at least as performant as $\boldsymbol{x}_p$ as described in problem (2). This results in identifying a range of DV values instead of a point-based design resulting from conventional optimisation. Problem (2) can be solved using methods like the corner tracking algorithm, described in [29] or via the stochastic iteration algorithm presented in [21]. The result would then be the largest hypercube, i.e., the solution space containing all designs at least as performant as $\boldsymbol{x}^*$. The solution space can be visualised as design space sections projected in 2D as illustrated in Fig. 5. Any design within the solution space would be robust to variations concerning the realised design variable values. Sometimes, computing the solution spaces for some variables might not be necessary. For example, controller parameters, K_p, K_d, are purely logical and can be modified without any cost. Implying that range in such variables can be traded for an increased range of other DVs of interest. For this, we provide the X-Ray toolbox as an interface a designer can use to explore the trade-off between different DVs. The tool lets a designer modify the ranges of all design variables while visualising the design space as 2D sections.

Such projected solution space for a set of DVs is illustrated

[3]https://store.tmotor.com/goods.php?id=1188

TABLE I: Comparison of the QoI values realised by various designs from the solution space

Design	**Cycle time t_{cyc} (s)**	Energy loss L (J)	Cost C ($)
$\boldsymbol{x}^*$(●)	0.96	$0.5e3$	-
$\boldsymbol{x}_p$ (▲)	3.05	$6.8e3$	798.9
$\boldsymbol{x}_s$ (★)	2.30	$1.6e3$	698.6

in Fig. 5. The optimal design $\boldsymbol{x}^*$ is marked as ●, while the closest possible physically feasible design is marked as ▲. All the designs equivalent to or better than $\boldsymbol{x}_p$ are denoted by ●. Designs that violate cycle time or energy requirements are depicted as ● and ● respectively. These design sections allow one to comprehend the coupled design space. For example, the non-linear interface between good and bad designs between τ and m can be visually identified in Fig. 5. Similarly, the section containing l_7 vs l_4/l_5, i.e., the length of the last link vs the ratio of two links, shows that the performance of the robot is independent of the link length ratio, i.e., the placement of the motor between link-4 and link-5 does not affect the performance of the robot. Such insights are beneficial and can help realise informed designs.

3) Discussion: Of all possible alternatives from the constructed solution space, a design $\boldsymbol{x}_s$ is selected, accounting for physical feasibility and the designer's choice shown in Fig. 3c. Further, to show that the performance of any design within the solution space is at least as good as $\boldsymbol{x}_p$, a random design shown in Fig. 6c is sampled. The performance comparison between all the designs with respect to the QoI, L and their corresponding RMS joint torques is presented in Fig. 4. Firstly, it is shown that the optimal design performs better than the Blue robot with respect to the QoI, L. As expected, the mean torque required by the optimal design x^* is the lowest compared to the other robots. Further, the selected robot $\boldsymbol{x}_s$ performs better than $\boldsymbol{x}_p$ and any random robot, as shown in Fig. 6c, sampled from the solution space performs at least as good as the robot $\boldsymbol{x}_p$ owing to the requirements imposed in problem (2). Not only is the mean torque required lower than that of the Blue robot, but since the robot designs are optimised over the task ensemble, the variance of the QoI and torque is significantly lower. This also means that the necessary joint torque of the optimised robots is less sensitive to the variation in payload mass.

As another advantage, the iterative process involving repetitive checking of the feasibility of available motors can be eliminated, and one can directly choose an off-the-shelf motor with specifications within the ranges obtained from the solution space. In the current case, an alternative motor AK10-9v1.1, which costs 12% lower than the initially chosen motor, also falls within the solution space and is selected as design $\boldsymbol{x}_s$. It is to be noted that this motor is not an obvious initial choice since its mass and torque are lower as compared to $\boldsymbol{x}^*$. The performance comparison of the three designs, $\boldsymbol{x}^*$, $\boldsymbol{x}_p$ and $\boldsymbol{x}_s$ is shown in Table I. As shown, the QoI L of $\boldsymbol{x}_s$ is around 30% better than the Blue blue robot. Further, up to 20% of the lost per-

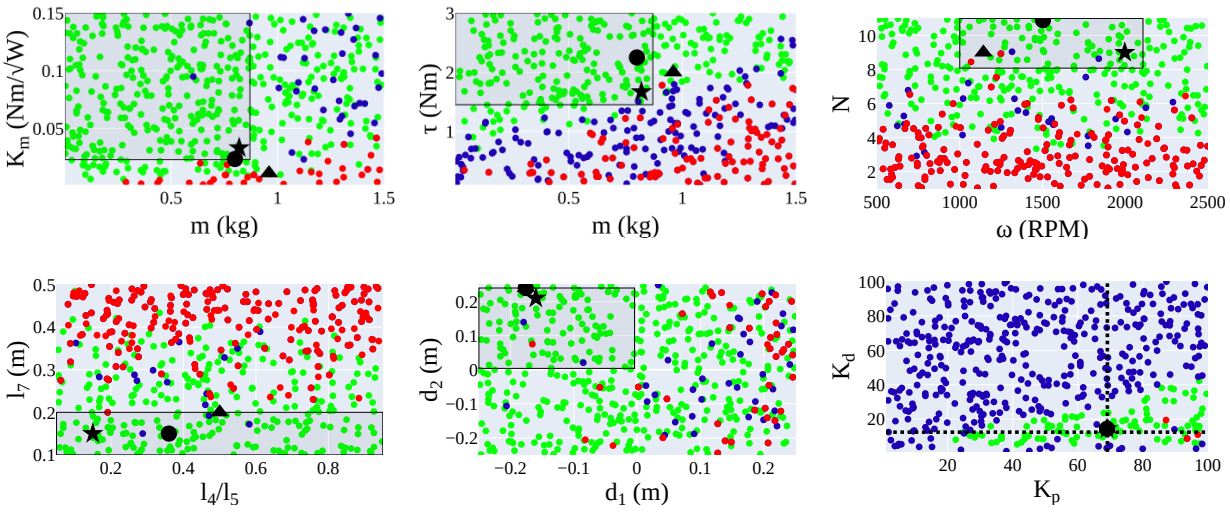
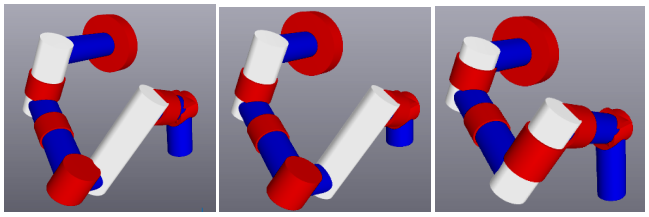


Fig. 5: The rectangular boxes overlaid on each plot are the projections of the solution space. ●, ●, ● represent designs that are good and that violating the t_{cyc} and L requirements. ●, ▲, ★ represent the optimal design $\boldsymbol{x}^*$, physically feasible design $\boldsymbol{x}_p$ and the selected design $\boldsymbol{x}_s$. Note that in each diagram, values for those DVs not shown on the axes are chosen randomly from within the intervals defined by the box edges shown in other diagrams as discussed in [8]. In the plots, DVs from $\boldsymbol{x}_A, \boldsymbol{x}_T, \boldsymbol{x}_C$ are variables associated only with the first joint of the robot. l_7 is the length of the last link, and l_4/l_5 is the ratio of two links, affecting the motor's location. d_1 and d_2 are offsets shown as white cylinders in Fig. 6.



(a) (b) (c)

Fig. 6: (a) The design $\boldsymbol{x}^*$ obtained from step-1. (b) Chosen physically feasible design $\boldsymbol{x}_p$. (c) Randomly chosen robot from within the solution spaces.

formance between $\boldsymbol{x}^*$,and $\boldsymbol{x}_p$ is recovered by the selected design $\boldsymbol{x}_s$. This is made possible by choice of DV ranges of mechanical, actuator and control variables as shown in Fig. 5. Furthermore, this method allows designers to investigate alternate designs when chosen solutions are not feasible, which is practical during component and electronics scarcity and supply chain issues.

The upper and lower bounds produced on the mechanical DVs give flexibility to the designer and provide tolerance over variations. Further, this allows the designer to bridge the gap between the simulated model and the physically realisable design. Moreover, since all the designs within the solution spaces are equivalent, component selection of motors and gearbox can now be made independent of each other, parallelising the design process. Moreover, due to the tolerance obtained as bounds on different design variables, the system level performance is robust to manufacturing imperfections, leading to a robust design. The real-world prototype shown in Fig. 1b, constructed using the solution spaces, highlights the capabilities of the proposed method.

Based on their performance, a method for clustering robot designs in combined mechanics and control space is presented in [30]. Alternatively, in the current technique, the set of designs within the solution spaces are equivalent with respect to the system-level performance. In the current example, this means that the solution spaces containing only the mechanical DVs, $\boldsymbol{x}_M$ is an equivalent cluster of designs independent of the chosen controller and vice versa. Therefore, the constructed solution spaces enable the controller's decoupling from the robot's morphology. This decoupling is unique to the proposed approach and can be exploited to identify controllers capable of generating a required behaviour on a wide range of robot morphologies. However, it is limited to the parametrisation of the controller and further investigation is needed for a more general case.

V. Conclusion and Future Work

A method for robust co-design of robots via cascaded optimisation is presented. The advantage of concurrent design optimisation is illustrated via a quantitative comparison showing significant improvement in the performance metrics. The benefits of constructing solution spaces to identify better designs, provide tolerance on DVs and realise physically feasible designs are discussed. Further, the unique ability to decouple morphology and control of the robot via solution spaces is noted. To emphasise the value of the proposed approach, a tangible prototype constructed using the proposed approach is presented. Finally, the X-Ray toolbox is offered as an interface for designers to comprehend and manipulate solution spaces. The utility of the proposed method with different control strategies and robot morphologies will be explored in the future.

REFERENCES

[1] A. Parmiggiani, L. Fiorio, A. Scalzo, A. V. Sureshbabu, M. Randazzo, M. Maggiali, U. Pattacini, H. Lehmann, V. Tikhanoff, D. Domenichelli *et al.*, "The design and validation of the r1 personal humanoid," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 674–680.

[2] J. T. Allison and D. R. Herber, "Multidisciplinary design optimization of dynamic engineering systems," *AIAA journal*, vol. 52, no. 4, pp. 691–710, 2014.

[3] D. R. Herber and J. T. Allison, "Nested and Simultaneous Solution Strategies for General Combined Plant and Control Design Problems," *Journal of Mechanical Design*, vol. 141, no. 1, 2018. [Online]. Available: https://doi.org/10.1115/1.4040705

[4] C. Schaff, D. Yunis, A. Chakrabarti, and M. R. Walter, "Jointly learning to construct and control agents using deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9798–9805.

[5] C. Hazard, N. Pollard, and S. Coros, "Automated design of robotic hands for in-hand manipulation tasks," *International Journal of Humanoid Robotics*, vol. 17, no. 01, p. 1950029, 2020. [Online]. Available: https://doi.org/10.1142/S0219843619500294

[6] F. De Vincenti, D. Kang, and S. Coros, "Control-aware design optimization for bio-inspired quadruped robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1354–1361.

[7] E. Zardini, D. Zappetti, D. Zambrano, G. Iacca, and D. Floreano, "Seeking quality diversity in evolutionary co-design of morphology and control of soft tensegrity modular robots," ser. GECCO '21. Association for Computing Machinery, 2021, pp. 189–197. [Online]. Available: https://doi.org/10.1145/3449639.3459311

[8] M. Zimmermann, S. Knigs, C. Niemeyer, J. Fender, C. Zeherbauer, R. Vitale, and M. Wahle, "On the design of large systems subject to uncertainty," *Journal of Engineering Design*, vol. 28, no. 4, pp. 233–254, 2017. [Online]. Available: https://doi.org/10.1080/09544828.2017.1303664

[9] D. V. Gealy, S. McKinley, B. Yi, P. Wu, P. R. Downey, G. Balke, A. Zhao, M. Guo, R. Thomasson, A. Sinclair, P. Cuellar, Z. McCarthy, and P. Abbeel, "Quasi-direct drive for low-cost compliant robotic manipulation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 437–443.

[10] T. Dinev, C. Mastalli, V. Ivan, S. Tonneau, and S. Vijayakumar, "A versatile co-design approach for dynamic legged robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)*, 2022.

[11] 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.

[12] T. Chen, Z. He, and M. Ciocarlie, "Hardware as policy: Mechanical and computational co-optimization using deep reinforcement learning," in *Proceedings of the 2020 Conference on Robot Learning*, vol. 155. PMLR, 2021, pp. 1158–1173. [Online]. Available: https://proceedings.mlr.press/v155/chen21a.html

[13] A. Mehta, J. DelPreto, and D. Rus, "Integrated Codesign of Printable Robots," *Journal of Mechanisms and Robotics*, vol. 7, no. 2, 2015. [Online]. Available: https://doi.org/10.1115/1.4029496

[14] G. Bravo-Palacios, G. Grandesso, A. D. Prete, and P. M. Wensing, "Robust co-design: Coupling morphology and feedback design through stochastic programming," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 2, 2022.

[15] G. Fadini, T. Flayols, A. Del Prete, N. Mansard, and P. Soures, "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.

[16] M. Toussaint, J.-S. Ha, and O. S. Oguz, "Co-optimizing robot, environment, and tool design via joint manipulation planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6600–6606.

[17] M. Daub, F. Duddeck, and M. Zimmermann, "Optimizing component solution spaces for systems design," *Structural and Multidisciplinary Optimization*, vol. 61, no. 5, pp. 2097–2109, 2020.

[18] J.-D. Korus, P. G. Ramos, C. Schütz, M. Zimmermann, and S. Müller, "Top-down development of controllers for highly automated driving using solution spaces," in *9th International Munich Chassis Symposium 2018*. Springer, 2019, pp. 325–342.

[19] M. Münster, M. Lehner, D. Rixen, and M. Zimmermann, "Vehicle steering design using solution spaces for decoupled dynamical subsystems," in *26th Conference on Noise and Vibration Engineering ISMA 2014, Proceedings*, vol. 26, 2014, pp. 279–288.

[20] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu

[21] M. Zimmermann and J. E. von Hoessle, "Computing solution spaces for robust design," *International Journal for Numerical Methods in Engineering*, vol. 94, no. 3, pp. 290–307, 2013.

[22] P. Ma, T. Du, J. Z. Zhang, K. Wu, A. Spielberg, R. K. Katzschmann, and W. Matusik, "Diffaqua: A differentiable computational design pipeline for soft underwater swimmers with shape interpolation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.

[23] Y. Yesilevskiy, Z. Gan, and C. David Remy, "Energy-optimal hopping in parallel and series elastic one-dimensional monopeds," *Journal of Mechanisms and Robotics*, vol. 10, no. 3, p. 031008, 2018.

[24] M. Charbonneau, F. Nori, and D. Pucci, "On-line joint limit avoidance for torque controlled robots by joint space parametrization," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 899–904.

[25] V. Modugno, U. Chervet, G. Oriolo, and S. Ivaldi, "Learning soft task priorities for safe control of humanoid robots with constrained stochastic optimization," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 101–108.

[26] W. Garage, "PR2 user manual," 2012.

[27] N. Hansen, Y. Akimoto, and P. Baudis, "CMA-ES/pycma: r3.2.2," 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6370326

[28] Hauptmech, "Odio_URDF," 2021. [Online]. Available: https://github.com/hauptmech/odio_urdf

[29] S. Erschen, F. Duddeck, M. Gerdts, and M. Zimmermann, "On the Optimal Decomposition of High-Dimensional Solution Spaces of Complex Systems," *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, vol. 4, no. 2, 2017. [Online]. Available: https://doi.org/10.1115/1.4037485

[30] K. Ghazi-Zahedi, R. Deimel, G. Montúfar, V. Wall, and O. Brock, "Morphological computation: the good, the bad, and the ugly," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 464–469.