Control- & Task-Aware Optimal Design of Actuation System for Legged Robots using Binary Integer Linear Programming

Youngwoo Sim, Guillermo Colin and Joao Ramos

***Abstract*— Athletic robots demand a whole-body actuation system design that utilizes motors up to the boundaries of their performance. However, creating such robots poses challenges of integrating design principles and reasoning of practical design choices. This paper presents a design framework that guides designers to find optimal design choices to create an actuation system that can rapidly generate torques and velocities required to achieve a given set of tasks, by minimizing inertia and leveraging cooperation between actuators. The framework serves as an interactive tool for designers who are in charge of providing design rules and candidate components such as motors, reduction mechanism, and coupling mechanisms between actuators and joints. A binary integer linear optimization explores design combinations to find optimal components that can achieve a set of tasks. The framework is demonstrated with 200 optimal design studies of a biped with 5-degree-of-freedom (DoF) legs, focusing on the effect of achieving multiple tasks (walking, lifting), constraining the mass budget of all motors in the system and the use of coupling mechanisms. The result provides a comprehensive view of how design choices and rules affect reflected inertia, copper loss of motors, and force capability of optimal actuation systems.**

I. Introduction

Athletic robots such as the MIT cheetah and Cassie have shown agile movements, such as utilizing momentum to jump or push heavy objects. Achieving such athletic motions requires a comprehensive approach to whole-body control [1], [2] and systematic actuation system design [3]–[5]. While the literature has made significant strides in controls of humanoids, the recipe to create high-DoF actuation systems for legged robot is still an open question.

The ability to generate large forces and accelerations stems from principles at two distinct layers of the actuation system. At the actuator level, the proprioceptive actuation [6] suggests employing torque-dense motors and smaller gear ratio. This helps improving contact force control and allows agile motions. In cases where actuator torques are insufficient, this can be addressed at the system level through parallel actuation [7], [8], where coupled actuators may combine torques to amplify joint torques. In comparison to serial actuation, parallel actuation offers larger joint torques without substantially increasing reflected inertia [9].

However, as the number of actuators grows, designers encounter challenges when applying design principles. Firstly, these principles offer general guidelines but leave specific design choices to the discretion of the designers. Consequently, conducting a multi-dimensional analysis of trade-offs between a wide range of component choices and their impact on performance metrics (e.g., the generalized inertia matrix [10] or the force capability polytope [11]) becomes a daunting task. Taking the example of parallel actuation, in which actuators are coupled, even a slight modification to a single gear ratio can affect the torque and velocity capabilities of other actuators, thereby complicating the design analysis. Secondly, discrete leaps in available component choices (e.g., motors or reduction mechanisms) often necessitate additional design iterations and engineering work.

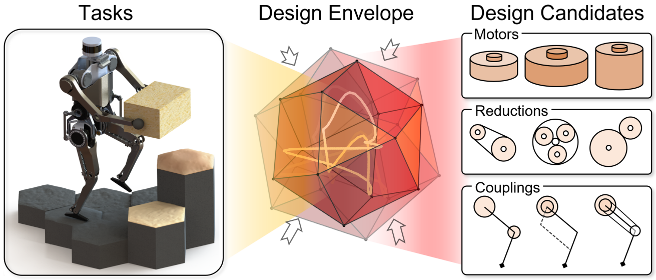


Fig. 1. Conceptual overview of the proposed design optimization of multi-actuator system. Video: `https://youtu.be/DEXYp0TsqRI`

To circumvent the design challenges, robot designers took inspirations from nature. Evolutionary algorithms have been employed to concurrently optimize morphology design and motion controls [12], [13]. In particular, the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) has gained widespread usage [14], [15]. Another approach involves optimizing mechanical component selections to enhance high-level system performance for predefined tasks [16], [17].

Nonetheless, most existing legged robot design methods require additional efforts to improve practicality. First, robot models are often simplified for efficient numerical optimization. As a result, designers must iteratively address design considerations (e.g., self-collision [18], packaging, or component availability) that were not accounted for during optimization. The component availability issue may be avoided by limiting design candidates to a few safe options as in [19]. Second, designers typically rely on design tools to gain a comprehensive view of multiple design scenarios, rather than seeking a singular solution. Moreover, during iterative processes of making necessary modifications, design tools assist in reasoning and justifying these modifications. Hence, it is desirable for the design tool to run fast to incorporate gained intuitions. However, existing methods

This work is supported by the National Science Foundation via grant CMMI-2043339 and CMMI-2220924.

The authors are with the Department of Mechanical Science and Engineering at the University of Illinois at Urbana - Champaign, USA. (`sim17@illinois.edu`)

require or use computationally heavy algorithms such as inverse kinematics [13], [19], evolutionary algorithms [13], [14] or mixed-integer programming [16].

This paper proposes a *design tool* that assists designers in creating high-DoF actuation systems, addressing the aforementioned issues. First, we want a design tool that guarantees a system design that is capable of various humanoid tasks such as carrying objects, weight-lifting, and/or running. We assume that such motions can be synthesized from a reduced-order model (ROM) of a robot consisting of high-level design parameters such as overall mass, inertia, and center of mass (CoM) height with fixed morphology. These assumptions are consistent with existing motion planners based on single rigid body dynamics (SRBD) [20] or centroidal dynamics [1], allowing us to use similar ROM-based motion planners. Second, designers are encouraged to use their expert knowledge to curate libraries of design candidates such as motors, gear ratios, and coupling mechanisms between actuators and joints. Moreover, the high-level intuition of designers is translated into design rules, which designers can experiment with to observe how optimal solutions behave under these rules. Third, our objective is set to minimizing reflected inertia of actuation system to create dynamic humanoids capable of rapid limb acceleration and impact mitigation [6]. Lastly, the optimal design problem is translated into a binary integer linear program and solved by Gurobi [21] which ensures global optimum for mixed-integer programs.

This framework is demonstrated by 200 optimal design studies of a biped with 5-DoF legs focusing on the effects of 1) achieving a set of tasks, 2) employing actuator-joint couplings (parallel mechanism) and 3) limiting the mass budget of all motors in a system. The optimization study was given two tasks that involve different joint usage patterns: walking and weight-lifting in a snatch style. Next, design libraries of motors, gear ratio, and coupling mechanisms were compiled from existing humanoids and unprecedented-but-practical designs for comparison.

The contribution of the proposed framework is as follows. First, this tool allows designers to quickly navigate vast design spaces and reveals a comprehensive overview of optimal multi-actuator system designs for legged robots. Second, the framework serves as a reasoning tool for high-DoF system design. Designers can quickly run a large number of optimal studies to observe how optimal solutions respond to modifying tasks, design libraries, or design rules. For example, providing a larger set of tasks would result in general-purpose humanoid design whereas providing a single task would yield a more dedicated system design. Moreover, novice engineers can update components available in the market to create better designs, while experts can explore combinations of novel mechanisms for comparison.

II. Background: Representation of Design Library & Combinations

The binary variables are efficient tools for handling two essential aspects of design: 1) determining whether to select or exclude component candidates from libraries and 2) expressing combinations of components.

Definition 1: *(Binary Combination)* Let $\mathbf{p}=\{p_1,\cdots,p_n\}$ be a library of n objects, $\mathbf{x}=(x_1,\cdots,x_n)\in\{0,1\}^n$ be a vector of binary variables. An arbitrary singular choice of an object $p(\mathbf{x})_B$ out of the library $\mathbf{p}$ is obtained by a *binary combination of* $\mathbf{p}$ *and* $\mathbf{x}$ as follows,

$$p(\mathbf{x})_B \coloneqq \textstyle\sum_{i=1}^{n} p_i x_i \qquad (1)$$

$$\implies \textstyle\sum_{i=1}^{n} x_i = 1 \quad \text{(exclusivity)}. \qquad (2)$$

For instance, a selection of p_j is achieved by assigning 1 to the corresponding variable $(x_j=1)$ and 0 to the rest $(x_k=0, k\neq j, k\in[n])$. The definition itself implies an exclusivity constraint that enforces a singular choice. It should be noted that the objects p_i can be scalars such as maximum torque or matrices like the Jacobian of coupling mechanism that maps actuator velocities to joint velocities.

Remark 1: *(Linearization of Multilinear Polynomials)* Products of binary variables commonly arise when combining components; e.g., an actuator is a combination of a motor and a reduction mechanism. A useful property of the product of binary variables is that it can be perfectly linearized (as opposed to linearly approximated). The product of binary variables x and y, denoted as xy, is linearized by introducing a new binary variable z in place of xy and adding the following linear constraints:

$$z \leq x, \quad z \leq y, \quad z \geq x + y - 1. \qquad (3)$$

For linearization of general multilinear forms, refer to [22].

III. Optimal Design of Transmission System for Humanoid Robots

A. Overview

We introduce a design optimization tool that finds mechanical components for the actuation system of athletic humanoids. Most importantly, this framework is composed as an interactive tool for designers and addresses difficulties in designing a high-DoF actuation system. Hence, the framework expects designers to provide their knowledge and intuition and run several iterations to gain an understanding of optimal designs of humanoids.

The workflow of the proposed tool is divided into three steps. (Fig. 2). The first step is to prepare key ingredients for design optimization; 1) ROMs for the generation of task trajectories, 2) a set of task trajectories generated by motion planning software using ROMs, 3) libraries of candidate components to create the robot, 4) relevant design rules, and 5) design goals. As to the preparation of task trajectories, first, ROMs are created from known kinematics and parameters of high-level design envelopes and behavior targets set by designers (e.g., overall mass, inertia, CoM height or desired velocity) depending on the tasks. Then, using ROM-based motion planning approaches (e.g., SRBD-based trajectory optimization [23] or kino-dynamic planning [1]), task-space trajectories of desired tasks are obtained.

Finally, the task-space trajectories are converted to joint-space trajectories using inverse kinematics, inverse dynamics, or whole-body controllers. It is important to note that the ROM assumes fixed limb lengths and joint ordering. This is because humanoids are typically designed to have similar proportions and kinematics to humans. Also, as there exist only a few practical joint orderings that mimic human limb kinematics, design studies for every joint ordering can be carried out. It is also assumed that the actuators are placed very closely to the center of the robot (proximal actuation) and actuator dynamics is neglected. Hence, the actuator placements and their inertial torques do not affect the dynamics of the robot. Lastly, the framework does not take into account the structural design of limbs because we assume that the apparent inertia and mass could be significantly minimized compared to the reflected inertia of the actuation system.

The second step of the framework is to integrate tasks, design library, and design rules and translate them to linear binary integer programming. The overarching concept is expressed as follows with $\mathbf{x}$ as design choices of mechanical components,

$$\begin{aligned} \min_{\mathbf{x}} \quad & \text{ReflectedInertia}(\mathbf{x}), && \text{(4a)} \\ \text{s.t.} \quad & \mathbf{x} \in \text{DesignLibrary}, && \text{(4b)} \\ & \text{TaskTrajectory} \subseteq \text{OperationDomain}(\mathbf{x}), && \text{(4c)} \\ & \text{DesignRules}(\mathbf{x}). && \text{(4d)} \end{aligned}$$

The contents of (4) are briefed as follows.

Control-Awareness (4a): The objective is to minimize the reflected inertia of the entire actuation system perceived at the joint-level (joint-space reflected inertia). This objective aims to improve force control performance, enhance the system's ability to regulate external impacts and enable agile limb movements [6]. The objective aligns with the QDD paradigm, although designers have the flexibility to modify it to prioritize energy efficiency or minimum torque expenditure, depending on their interest.

Discrete Design Space (4b): In this framework, designers are responsible for providing and modifying libraries of candidate motors, reduction mechanisms, and actuator-joint couplings. Then, an optimizer searches for the optimal design by exploring all possible combinations of provided components. Designers are encouraged to use actuator-joint coupling mechanisms from existing robots, utilize their expert knowledge to curate practical component candidates, or include novel components whose utility is in question.

Task-Awareness (4c): Robots are expected to achieve multiple tasks. This framework ensures the feasibility of these tasks within the capability of motors. In other words, the given tasks are mapped to trajectories of motor torques and velocities. Then, these trajectories are constrained to remain within the motor limits such as voltage and thermal limit (operation domain) as in Fig. 3(b). The safety of a controller can be tuned by parameters that define allowable closeness between extremes of task trajectory and the boundary of the operating domain.

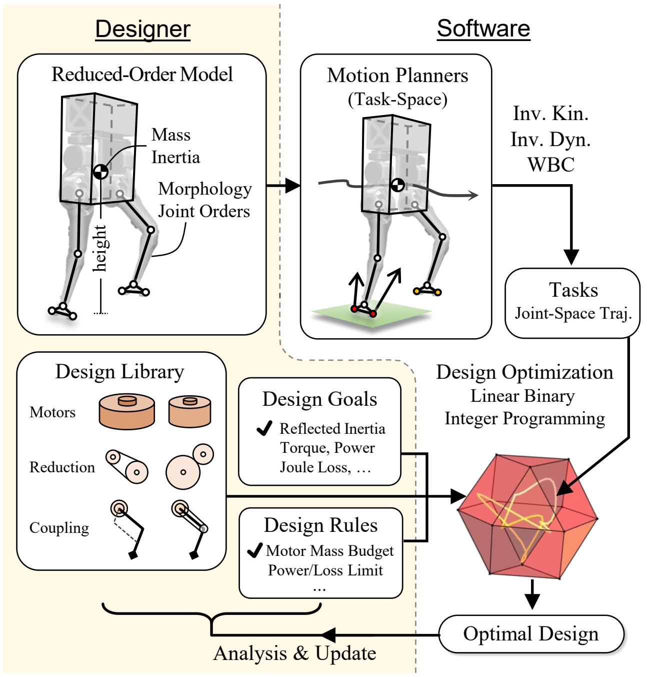


Fig. 2. Overview of design optimization of high-DoF actuation system for humanoid robots

Design Rules (4d): Additional constraints can be incorporated into the optimization process, such as limiting the sum of motor mass for a subset of joints or limiting the copper loss on motors for a certain duration. As long as the constraints are multilinear polynomial, they can be perfectly linearized (see Sec. II). Although not presented in this paper, logical constraints can be imposed for compact arrangement of large components such as motors.

As the last step of the workflow, designers need to analyze the resulting optimal solutions and modify the aforementioned ingredients for design optimization. For instance, if the optimization is infeasible due to high torque requirements, one can expand the design library by adding more torque-dense motors, relax the design rules, or relax the task requirements (e.g., smaller total mass, smaller desired CoM velocity of the ROM).

B. Maximizing Force Control Bandwidth

Smaller reflected inertia of actuators contributes to faster limb swing or acceleration and improves the ability to mitigate impacts from contact by quickly retracting from contacts (e.g., ground contact). Both capabilities are critical requirements for dynamic humanoids.

Hereafter, the goal of minimizing reflected inertia of the actuation system is reconnected with the earlier optimization sketch (4) to formulate objective functions using binary variables. Although [6] suggests minimizing reflected inertia in task-space, obtaining such expression using binary variables in linearizable form is prohibitive due to rational forms arising from matrix inversions [10, Eqn. (17)]. As a proxy to task-space inertia, a joint-space reflected inertia matrix is used. The downside of using this proxy is that it neglects

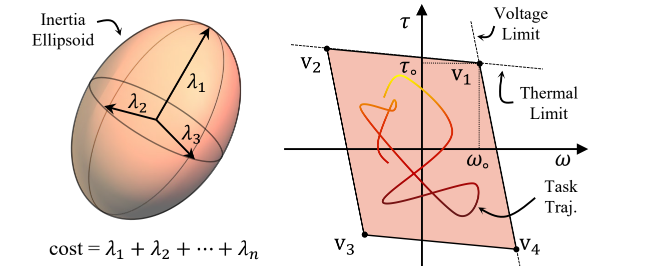


Fig. 3. (a) The size of reflected inertia matrix is measured as the sum of semi-axes lengths which is obtained by trace operation (b) Task trajectory projected on to motor-level is constrained to remain within the operation domain of a motor defined by vertices $\mathbf{v}_i$.

the lever effect of limb lengths to joints which is beyond the scope of this paper. Nevertheless, some relevant observations are stated in Sec. V-C.

Let us build the joint-space reflected inertia matrix of an actuation system consisting of n_d joints (or actuators) from libraries of motors, gearboxes, and actuator-joint couplings with associated binary variables $\mathbf{x}_i \in \mathbb{B}^{n_{m_i}}$, $\mathbf{y}_i \in \mathbb{B}^{n_{g_i}}$, $i \in [n_d]$, and $\mathbf{z} \in \mathbb{B}^{n_c}$, where $n_{(\cdot)}$ is the number of candidates in the respective libraries. The reflected inertia of an i'th actuator is the product of binary combinations of rotor inertia $I_i^r(\mathbf{x}_i)_B$ and gear ratio squared $N_i^2(\mathbf{y}_i)_B$,

$$I_i^a(\mathbf{x}_i, \mathbf{y}_i) = I_i^r(\mathbf{x}_i)_B N_i^2(\mathbf{y}_i)_B. \tag{5}$$

Next, the reflected inertia of each actuator forms a diagonal matrix of actuator-space reflected inertia $\mathbf{H}_\mathrm{r}^a(\mathbf{x}, \mathbf{y}) = \mathrm{diag}(I_1^a, \ldots, I_{n_d}^a)$. This inertia matrix is projected on to joint-space by a binary combination of actuator-joint coupling Jacobian $\mathbf{C}(\mathbf{z})_B$ which represents a mechanical connection between actuators and joints. The coupling Jacobian maps actuator velocities $\dot{\boldsymbol{\psi}} \in \mathbb{R}^{n_a}$ to joint velocities $\dot{\mathbf{q}} \in \mathbb{R}^{n_a}$ as $\mathbf{C} : \dot{\boldsymbol{\psi}} \rightarrow \dot{\mathbf{q}}$. The joint-space reflected inertia is as follows,

$$\overline{\mathbf{H}}_r^\mathrm{j}(\mathbf{x}, \mathbf{y}, \mathbf{z}, k) = \mathbf{C}^{-\top} \mathbf{H}_\mathrm{r}^a \mathbf{C}^{-1}, \tag{6}$$

where k denotes the time instance at which the inertia is evaluated along the given tasks and $\mathbf{x}$, $\mathbf{y}$ are lifted binary variables of $\mathbf{x}_i$, $\mathbf{y}_i$.

Lastly, the *size of the inertia matrix* is measured using the $\mathrm{trace}()$ operator. The trace of an inertia matrix is the sum of its eigenvalues which relate to the lengths of semi-axes of the inertia ellipsoid (Fig. 3(a)). Hence, the trace operator allows the optimizer to minimize reflected inertia on every dimension until motor mass becomes too small and, subsequently, a motor does not have sufficient torque to achieve given tasks.

C. Task Feasibility

Task feasibility assures that the desired task trajectories are accommodated within the capabilities of the motors. The torque and velocity capability or an operation domain of a motor is approximated as a convex polygon $\mathrm{convhull}(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ (colored polygon in Fig. 3(b)). This polygon comprises vertices $\mathbf{v}_i = (\alpha_i \omega_\circ, \beta_i \tau_\circ)$, whose coordinates are defined by motor parameters $\omega_\circ, \tau_\circ$ such as rated velocity and torque with coefficients α_i, β_i. Next, the vertices are associated with the motor library through a binary combination, $\mathbf{v}_i(\mathbf{x}) = (\alpha_i \omega_\circ(\mathbf{x})_B, \beta_i \tau_\circ(\mathbf{x})_B)$. Consequently, the operation domain is expressed as follows,

$$P(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathrm{convhull}\left(\mathbf{v}_1(\mathbf{x})_B, \ldots, \mathbf{v}_k(\mathbf{x})_B\right), \tag{7}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are vectors of coefficients $\alpha_{(\cdot)}$ and $\beta_{(\cdot)}$.

Next, the joint-space trajectories of given tasks are concatenated and projected to motor-space. The joint-space trajectory of length T, $U^\mathrm{j} = \{(\boldsymbol{\omega}, \boldsymbol{\tau})_k \,|\, \boldsymbol{\omega}, \boldsymbol{\tau} \in \mathbb{R}^m, k \in [T]\}$, is projected to actuator-space using actuator-joint coupling Jacobian $\mathbf{C}(\mathbf{z})_B$, then projected to motor-space using gear ratio Jacobian $\mathbf{N}(\mathbf{y}) = \mathrm{diag}(N_1(\mathbf{y})_B, \ldots, N_m(\mathbf{y})_B)$. The tasks trajectory on motor-space $U^\mathrm{m}(\mathbf{y}, \mathbf{z})$ is obtained as a function of choices of gearboxes and couplings,

$$\begin{aligned} U^\mathrm{m}(\mathbf{y}, \mathbf{z}) = \{(\boldsymbol{\omega}, \boldsymbol{\tau}) \mid \boldsymbol{\omega} = \mathbf{N}\mathbf{C}^{-1}\boldsymbol{\omega}_\mathrm{j}, \\ \boldsymbol{\tau} = \mathbf{N}^{-\top}\mathbf{C}^\top \boldsymbol{\tau}_\mathrm{j}, (\boldsymbol{\omega}_\mathrm{j}, \boldsymbol{\tau}_\mathrm{j}) \in U^\mathrm{j}\}. \end{aligned} \tag{8}$$

Finally, the following task feasibility constraint ensures the task trajectory to be contained within the operation domain

$$U^m(\mathbf{y}, \mathbf{z}) \subseteq P(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{9}$$

which easily converts to a set of linear constraints.

D. Design Rules

In this study, we bound the motor mass budget. If the motor mass budget is left unbounded, the optimizer will always prefer heavier, larger, and more torque-dense motors. However, there is a limit to the diameter of the motor that fits inside the robot's body. Moreover, the sum of motor mass may take up to 40% of the entire mass of a robot [24]. Hence, a smaller motor mass budget may leave significant room for a larger payload. The motor mass budget constraint is enforced as follows,

$$m_t(\mathbf{x})_B = \textstyle\sum_{i=1}^m m_i x_i \leq m_t^\mathrm{UB}, \tag{10}$$

where m_t is total mass of a design case, m_i is mass of candidate motors and m_t^UB is the total mass budget.

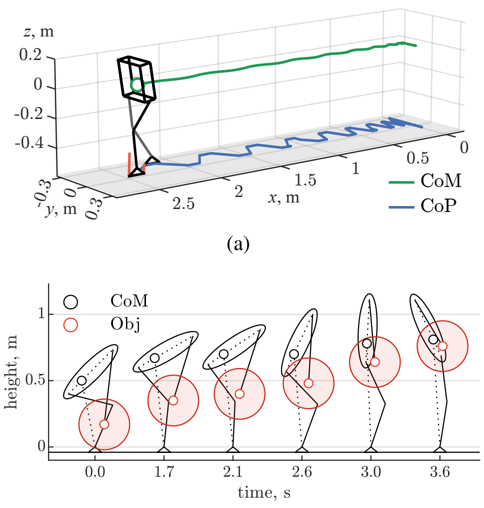
Designers can further employ various design rules such as limits on torque, power, etc. at various levels such as motor, actuator, and joint-levels. It is because system capability and task trajectories can be explicitly expressed using binary combinations. Second, logical constraints are also allowed. A statement, 'If a component A_1 is used, then A_2 also is used', is translated as $x_2 - x_1 \geq 0$ where x_1, x_2 are binary variables that associate components A_1, A_2. For more logical constraints, refer to [25].

IV. Humanoid Design Study

A. Robot Model and Tasks

A humanoid robot is modeled as a single rigid body with massless limbs and weight 20 kg that closely resembles Tello in [9]. The joint ordering is hip yaw, hip roll, hip pitch, and knee pitch followed by ankle pitch.

Next, based on the ROM, two task trajectories were generated (Fig. 4). The first task corresponds to moderately



(a)

(b)

Fig. 4. Sketches of tasks used in humanoid design study. (a) walking forward at moderately fast speed and (b) lifting an object in snatch style.

fast walking, with a peak speed of 1.3 m/s, a step frequency of 0.3 s. Joint torques and velocities of single support phase (0.15 s) are used. The walking trajectories are generated by simulating the SRBD of Tello. A step placement feedback controller that tracks a predefined walking speed is implemented [26]. An optimization-based balance controller, similar to in [27], computes the ground reaction forces required to realize the desired center-of-mass walking dynamics. The SRBD task-space trajectories are converted to joint-space trajectories using inverse kinematics. The resulting joint velocities, which contain high-frequency signals, are low-pass filtered at a cutoff frequency of 30 Hz to obtain meaningful solutions. The second task trajectory is to lift a point mass of 15 kg in a snatch style over 3.6 s. The load weight is set relatively large to excite motor torques and velocities in a noticeably different pattern than the first task. This trajectory is generated by extracting the joint trajectory from a video of an athlete performing a snatch. The trajectory is scaled down to match the size of the robot, and joint torques are obtained using inverse dynamics. The lifting task only recruits the pitch axes of the hip, knee, and ankle joints.

It is important to note that designers may choose other tasks and this study is merely showcasing an example. Depending on the choice of task, optimal designs may vary significantly.

B. Design Library and Rules

The design space is formulated by combinations of the following component libraries. The motor library contains $n_m = 10$ motors from CubeMars (AK, R, RI series [28]). The reduction library consists of ideal transmission without frictional loss and gear ratio ranges from 1 to 12 at whole number interval ($n_r = 12$). The vertices of the operation domains of the motors are defined by peak torque and rated velocity from datasheets and vectors of tuning coefficients $\boldsymbol{\alpha} = (1.5, -2.5, -2.5, 1.5)$, $\boldsymbol{\beta} = (1.2, 1.2, -1.2, -1.2)$.

TABLE I
List of Actuator-Joint Couplings

Name	Jacobian	Name	Jacobian
serial	$\mathrm{diag}(1,1,1,1,1)$	par-45	$\mathrm{diag}(1,1,1,\mathbf{C}_d)$
par-12	$\mathrm{diag}(\mathbf{C}_d,1,1,1)$	par-12-34	$\mathrm{diag}(\mathbf{C}_d,\mathbf{C}_d,1)$
par-23	$\mathrm{diag}(1,\mathbf{C}_d,1,1)$	par-12-45	$\mathrm{diag}(\mathbf{C}_d,1,\mathbf{C}_d)$
par-34	$\mathrm{diag}(1,1,\mathbf{C}_d,1)$	par-23-45	$\mathrm{diag}(1,\mathbf{C}_d,\mathbf{C}_d)$

There are 8 couplings studied in this paper including 1 serial and 7 parallel couplings. The parallel couplings are based on modules of 2-DoF differential coupling whose Jacobian is

$$\mathbf{C}_d = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}. \tag{11}$$

The couplings are labeled as 'serial', 'par-(n_1m_1)' and 'par-(n_1m_1)-(n_2m_2)' where n_i, m_i denotes the index of joints involved in i'th differential coupling (see Table. I). Since a mechanical coupling between faraway joints is typically impractical, the couplings are limited to consecutive joints. The hardware realization of Tello [9] utilizes 'par-23-45'.

The total mass of 5 motors is subject to a constraint (motor mass budget), with 25 values ranging from 2.2 kg to 4.0 kg at intervals of 0.075 kg. After a few runs of modifying the values of the motor mass budget, it was found that the optimization was infeasible below 2.2 kg and solutions did not vary beyond 4.0 kg.

C. Design Optimization

The optimization problem is detailed as follows. Inclusion of library data is implied by the use of binary optimization variables $\mathbf{x}, \mathbf{y}, \mathbf{z}$.

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y},\mathbf{z}} \quad & \textstyle\sum_{k=1}^{T} \mathrm{trace}(\overline{\mathbf{H}}^{\mathrm{j}}_r(\mathbf{x},\mathbf{y},\mathbf{z},k)) \\ \text{s.t.} \quad & U^m(\mathbf{y},\mathbf{z}) \subseteq P(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}), \\ & m_t(\mathbf{x})_B \le m_t^{\mathrm{UB}}, \\ & \mathbf{x} \in \mathbb{B}^{n_a \times n_m}, \mathbf{y} \in \mathbb{B}^{n_a \times n_r}, \mathbf{z} \in \mathbb{B}^{n_c}, \\ & \mathrm{exclusivity}(\mathbf{x},\mathbf{y},\mathbf{z}) \end{aligned} \tag{12}$$

To explicitly study the effect of constraining total mass budget for all 5 motors and to show rich details of how optimal solution behaves per every coupling Jacobian, the optimization was run for all combination of mass budget constraints and coupling Jacobians (total $25 \times 8 = 200$ runs). After optimal solutions are obtained, system characteristics such as inertia matrix and force/torque capability polytopes (FCP/TCP) in joint and task-space and copper loss on actuators are retrieved. The FCP/TCP represents the maximum force/torque that can be applied for a given configuration, based on the limits of actuators [11].

The optimization problem (12) is linearized using techniques introduced in Sec. II. The design optimization is written in MATLAB and calls an integer linear programming solver (`intlinprog`, Gurobi). The solver was simultaneously run on CPU (20 cores, Intel i7-12700H) with default

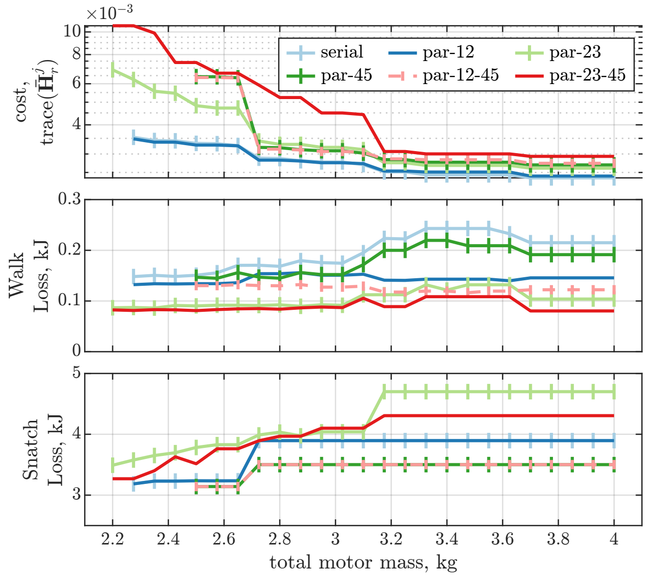


Fig. 5. Optimal costs (trace of joint-space reflected inertia) vs. total mass budget for 5 motors for every coupling case. Total copper losses per walking and snatch lifting task

solver parameters. Computation of total 200 runs took less than 100 s.

V. Results and Discussion

A. *Motor Mass Budget vs. Reflected Inertia vs. Copper Loss*

The first row of Fig. 5(a) gives an overview of optimal designs obtained with varied motor mass budgets and use of different couplings. First, as the motor mass budget decreases, availability of larger and more torque-dense motors decreases. Consequently, the optimal solutions leans towards smaller and weaker motors, necessitating the usage of elevated gear ratios. In the end, the use of higher gear ratios inflates reflected inertia or contributes to higher cost (optimal objective function value).

As to the effect of couplings, a few coupling cases were found infeasible (par-34, par-12-34) across all motor mass budgets, whereas other couplings become infeasible as the motor mass budget decreases. When the motor mass budget was limited to extremely small values, only coupled mechanisms provided sufficient torques and velocities (feasible). The serial actuation (no coupling) demonstrated the least reflected inertia at the cost of high copper loss for walking task across all cases of motor mass budget. In conclusion, this result suggests that 1) the framework can filter out couplings that cannot generate sufficient torque or speed, 2) at extremely low motor mass budget, actuators need to cooperate so that they can amplify joint torques to achieve desired tasks, and 3) use of coupling increases the reflected inertia for designs with a large motor mass budget.

The second and third row of Fig. 5 shows the combined copper loss of all motors of each task. As the motor mass budget decreases, the copper loss also decreases for most couplings. This tendency is opposite to that of reflected inertia because larger reflected inertia implies higher gear ratios, which again implies less motor torque and motor current draw. Two optimal designs following this trend are compared in Table. III. However, these implications are not strict: in the walking task, a few coupling cases did not follow this tendency (Fig. 5, second row, serial, par-45, par-23-45, around 3.2∼3.7 kg). This is because different choices of gear ratio and motor can emerge as long as they are optimal and satisfy design rules.

TABLE II
Design Comparison: Tello vs Optimal Solution

	Tello	Optimal Design
Coupling	par-23-45	par-23-45
Total motor mass [kg]	2.2	2.3
Gear Ratio	(6, 6, 6, 6, 6)	(5, 9, 7, 3, 10)
Motors	Steadywin V3	(R60, R80, RI100, RI100, RI100)
Obj. fcn. value [kgm^2]	0.0207	0.0099

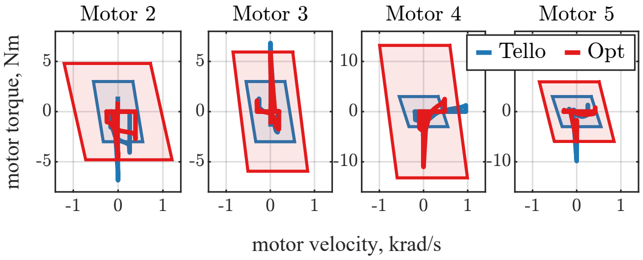


Fig. 6. Motor level trajectories of Tello vs optimized design.

Lastly, results from the optimization study is compared to a biped Tello [9]. Tello leg was designed to use identical actuators (Steadywin V3, gear ratio 6, peak torque 15 Nm, reflected inertia 0.0023 kgm^2). Hence, differential couplings were employed to amplify joint torque capabilities. In Table. II, the Tello is compared to a design optimized with similar motor mass budget and identical coupling. Most notably, the choices of motor and gear ratio differ across all actuators. Next, Tello design shows not only twice or larger objective function value, but also violation of motor operation boundary as in Fig. 6.

B. *Effect of Actuator-Joint Coupling*

The influence of the usage of couplings on task-space FCP and motor-level trajectory is visualized in Fig. 7 which presents a comparative analysis between two design cases, specifically, 'par-12' and 'par-23-45'. The latter utilizes more a higher degree of coupling than the former while both are designed with an identical motor mass budget.

A notable point in this comparison involves the velocity-dependent nature of FCP, attributed to the contour of the operational domain of motors. When motor velocities are high, the available torque (T1) near the boundary of the operation domain is reduced due to high motor velocity (Fig. 7(d1), (d2)).

The latter case, with additional degrees of coupling, has more chances for cooperative actuation which means that the load on a joint can be shared among the coupled

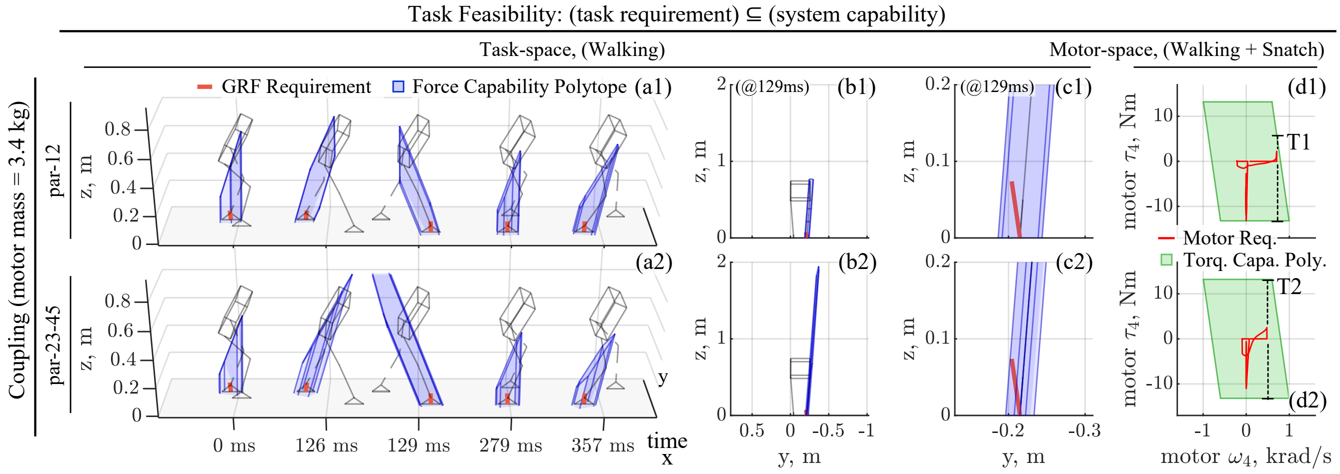


Fig. 7. Comparison of two optimal designs under identical motor mass budget and with different usage of coupling mechanism. (a1), (a2) FCP of contact leg in contact along walking. (b1), (b2) Frontal views show significant difference in size the FCP. (c1), (c2) FCPs of both designs are large enough to contain ground reaction forces to achieve moderately fast walking. (d1), (d2) Couplings affect how task trajectories are projected onto motor-level and motor torque capability margins (T1 vs. T2).

actuators. Thanks to the load sharing, the individual actuators experience lower torque requirements. This allows employing smaller gear ratios which renders smaller motor velocities. Hence, employing more degrees of coupling has a benefit of keeping the motor-level trajectory (velocity) at a safe distance from the boundary of the operational domain. The degradation of motor torque capability in design case 'par-12' manifests as a significantly smaller FCP at 129 ms (Fig. 7(a1), (b1)), in contrast to a larger FCP in (Fig. 7(a2), (b2)).

The state-dependent nature of the FCP could be related to the margin of admissible error of feedback controllers. For instance, if joint velocity deviates from a (pre)planned trajectory towards the boundary of the operation domain of motors, the torque capability diminishes. Simultaneously, a feedback controller will attempt to bring the state back to reference. At that moment, if the torque margin is insufficient, the torque commands from the feedback controller could suffer from saturation which may lead to failure. Designers can prevent this by modifying the approximated operation domain to be smaller than the actual operation domain of motors by tuning the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

C. Characterization of Inertia and Capability Polytopes

The relation between reflected inertia and FCP/TCP in both joint- and task-space is analyzed to address two possible claims; 1) minimization of reflected inertia should take place at task-space level, not joint-space and 2) maximization of TCP/FCP could be also taken into account as well as minimization of the reflected inertia. Unfortunately, the authors could not obtain direct quantification of reflected inertia in task-space and the volume of TCP/FCP using binary integer linear programming. Still, a few trends were observed by plotting the relation between the size of reflected inertia versus volume of TCP/FCP in joint-space and task-space as in Fig. 8. Optimal designs with smaller motor mass budgets are plotted with more opacity. The TCP/FCP were calculated without considering the degradation in torque margin due to high velocity. Since FCP varies as configuration (joint angles) changes it is difficult to plot all instances of tasks on a single plot. Instead, the distribution of FCP volume vs. trace of task space inertia is represented as a covariance ellipse whose origin is at the mean of data points. In each space, two similar trends were observed; 1) serial actuation shows more cohesiveness in the relation between the size of reflected inertia and FCP/TCP compared to coupled ones, 2) a combination of coupled actuation and smaller mass budget leads to larger size and variation of reflected inertia, and 3) larger motor mass budget allows larger FCP/TCP.

TABLE III
Comparison of Two Optimal Design Cases

	Joint (Axis)				
	Hip(z)	Hip(x)	Hip(y)	Knee(y)	Ankle(y)
Design Case A *(lightest, coupling* = (par-12), m_t = 2.28 kg*)*					
Inertia [gm^2]	(0.31, 0.31)		1.06	1.06	0.75
Peak Torq. [Nm]	(25.7, 25.7)		92.4	92.4	28.9
Motor	(RI70, RI70)		R100	R100	RI70
Gear Ratio	(3, 5)		7	7	9
Design Case B *(heaviest, coupling* = (par-23-45), m_t = 3.7 kg*)*					
Inertia [gm^2]	0.068	(0.69, 0.69)		(0.73, 0.73)	
Peak Torq. [Nm]	26.4	(105.6, 105.6)		(105.6, 105.6)	
Motor	R100	(R100, R100)		(R100, R100)	
Gear Ratio	2	(4, 4)		(3, 5)	

Note: Two entities within a parenthesis denotes two joints involved in 2-DoF actuator-joint coupling.

VI. Conclusion

The existing design principles and optimal design tools for the design of multi-DoF actuation systems often fall short in providing practical designs or understanding of optimal

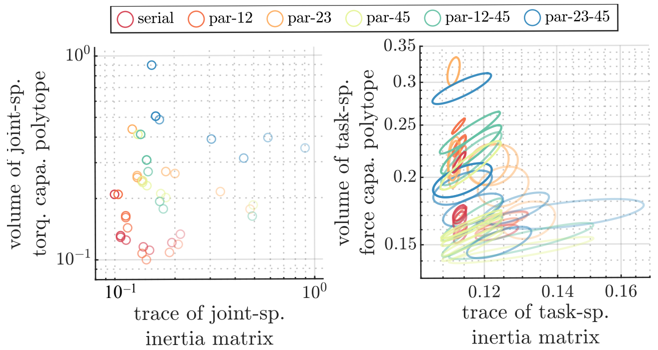


Fig. 8. Normalized optimal objective function value (trace of reflected inertia matrix) vs. normalized volume of FCP/TCP in joint-space (left) and task-space (right).

designs. To resolve this issue, an interactive optimal design tool for actuator system for humanoids is introduced. This tool is designed to incorporate expert designers' knowledge and rapidly solve large numbers of design studies to expand understanding of optimal designs.

This paper closes with several limitations of the present study and suggestions for future research. First, ROM employed in this study assumes massless limbs and actuator dynamics is neglected. Hence, the inertial forces exerted on the limbs and rotors within the actuation system are not accounted for. Future work may incorporate inertial forces associated with the limbs' apparent inertia and the reflected inertia of the actuators into the optimization process. Second, the applicability of optimal design solutions produced by the proposed tool remains unclear. An essential next step would be to integrate this tool with simulation software for the validation of proposed designs.

ACKNOWLEDGMENT

To research advisors Patrick Wensing and Donghyun Kim, Johannes Englsberger, and my colleagues Donghoon Baek and Seunghyun Bang for guidance and fruitful discussion.

REFERENCES

[1] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, "The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 1–8.

[2] M. S. Ahn, *Development and Real-Time Optimization-based Control of a Full-sized Humanoid for Dynamic Walking and Running*. University of California, Los Angeles, 2023.

[3] J. Hurst, B. Morris, H.-W. Park, and K. Sreenath, "Mabel, a new robotic bipedal walker and runner," in *2009 American Control Conference*. IEEE, 2009, pp. 2030–2036.

[4] N. G. Tsagarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, J. Santos-Victor, A. J. Ijspeert, M. C. Carrozza *et al.*, "icub: the design and realization of an open humanoid platform for cognitive and neuroscience research," *Advanced Robotics*, vol. 21, no. 10, pp. 1151–1175, 2007.

[5] G. Ficht and S. Behnke, "Bipedal humanoid hardware design: A technology review," *Current Robotics Reports*, vol. 2, pp. 201–210, 2021.

[6] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[7] K. G. Gim, J. Kim, and K. Yamane, "Design of a serial-parallel hybrid leg for a humanoid robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6076–6081.

[8] Z. Pandilov and V. Dukovski, "Comparison of the characteristics between serial and parallel robots." *Acta Technica Corviniensis-Bulletin of Engineering*, vol. 7, no. 1, 2014.

[9] Y. Sim and J. Ramos, "Tello leg: The study of design principles and metrics for dynamic humanoid robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9318–9325, 2022.

[10] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *The international journal of robotics research*, vol. 14, no. 1, pp. 19–36, 1995.

[11] P. Chiacchio, Y. Bouffard-Vercelli, and F. Pierrot, "Force polytope and force ellipsoid for redundant manipulators," *Journal of Robotic Systems*, vol. 14, no. 8, pp. 613–620, 1997.

[12] M. Garcia-Sanz, "Control co-design: An engineering game changer," *Advanced Control for Applications*, vol. 1, no. 1, p. e18, 2019, e18 ADC2-19-0039. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/adc2.18

[13] 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.

[14] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Joint optimization of robot design and motion parameters using the implicit function theorem." in *Robotics: Science and systems*, vol. 8, 2017.

[15] K. Wampler and Z. Popović, "Optimal gait and form for animal locomotion," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 1–8, 2009.

[16] Y.-H. Shin, S. Hong, S. Woo, J. Choe, H. Son, G. Kim, J.-H. Kim, K. Lee, J. Hwangbo, and H.-W. Park, "Design of kaist hound, a quadruped robot platform for fast and efficient locomotion with mixed-integer nonlinear optimization of a gear train," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6614–6620.

[17] W. Pawlus, G. Hovland, M. Choux, D. Frick, and M. Morari, "Drivetrain design optimization for electrically actuated systems via mixed integer programing," in *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2015, pp. 001 465–001 470.

[18] J. S. Ketchel and P. M. Larochelle, "Self-Collision Detection in Spatial Closed Chains," *Journal of Mechanical Design*, vol. 130, no. 9, p. 092305, 08 2008. [Online]. Available: https://doi.org/10.1115/1.2965363

[19] H. Ding, Z. Shi, Y. Hu, J. Li, B. Yu, and P. Zhang, "Lightweight design optimization for legs of bipedal humanoid robot," *Structural and Multidisciplinary Optimization*, vol. 64, pp. 2749–2762, 2021.

[20] R. Batke, F. Yu, J. Dao, J. Hurst, R. L. Hatton, A. Fern, and K. Green, "Optimizing bipedal maneuvers of single rigid-body models for reinforcement learning," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 714–721.

[21] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com

[22] C.-T. Chang and C.-C. Chang, "A linearization method for mixed 0–1 polynomial programs," *Computers & Operations Research*, vol. 27, no. 10, pp. 1005–1016, 2000.

[23] J. Ahn, S. J. Jorgensen, S. H. Bang, and L. Sentis, "Versatile locomotion planning and control for humanoid robots," *Frontiers in Robotics and AI*, p. 257, 2021.

[24] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, 2016.

[25] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[26] Y. Gong and J. Grizzle, "One-step ahead prediction of angular momentum about the contact point for control of bipedal locomotion: Validation in a lip-inspired controller," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2832–2838.

[27] M. Chignoli and P. M. Wensing, "Variational-based optimal control of underactuated balancing for dynamic quadrupeds," *IEEE Access*, vol. 8, pp. 49 785–49 797, 2020.

[28] "Cubemars," accessed: 12-Jul.-2023. [Online]. Available: https://store.cubemars.com/category-116-Exoskeleton+Robot.html