# Humanoid Robot Design Assistant – Requirements from Human Motion

Velin Kossev*, Cornelius Klas* and Tamim Asfour

***Abstract*— Humanoid robots are expected to interact with humans in built-for-human environments and perform human-like actions. This makes the design and optimization of humanoid robots challenging, in part because of the complexity of human motions. In our previous work, we introduced an approach that automatically calculates the necessary actuator requirements for a given upper-body humanoid robot kinematic performing motions retargeted from human motion data. In this paper, we propose a humanoid robot design framework, which encompasses robot kinematic arrangement selection and actuator optimization based on the actuator requirements data with a focus on the robot upper-body. We also develop a novel actuator optimization index, based on the speed, acceleration, and torque requirements, to help evaluate possible actuator configurations. The potential of the framework is illustrated through a theoretical optimization analysis of the actuator specifications of the humanoid robots ARMAR-6 and ARMAR-7, in which the optimal gear ratios of the arm joint actuators are determined based on a novel actuator optimization index for a specific set of human motions.**

## I. Introduction

Robots have the potential to be an indispensable part of our daily live in the near future by providing immense help in fields like healthcare, transportation, entertainment, etc. [1]. This implies that new robots will have to be designed to successfully complete tasks in different areas. Since robots usually consist of multiple electronic, computational, and mechanical sub-components , robot design remains a challenging task requiring high degree of understanding of mechatronic systems ([2], [3]). Computer tools are often employed by engineers to find solutions to the complex design problems, which readily present themselves in the field of robotics. Expert systems [4], simulation tools [5], genetic algorithms [6], custom software [7], implementations of optimization algorithms [8] among others, have been investigated by the robotics research community in terms of their impact on robot design.

One integral aspect of robot design is the choice of suitable actuators [9]. The selection of an actuator, consisting of a motor and a transmission (also called gearbox), has a direct effect on the dynamic capabilities of the robot [10]. Even though a considerable amount of research effort has gone into developing optimization techniques for the optimal selection of robot actuators, there is a severe lack of literature focusing on the application of actuator optimization algorithms to humanoid robots. Since humanoid robots are expected to accomplish complex tasks (e.g. bi-manual manipulation) the requirements on the actuation and the proper selection of motor-transmission combination need to be examined. In this work, we focus on the upper body, as it is essential for complex manipulation tasks, and most of our robots are positioned on a platform, reducing the need to address lower-body dynamics.

The Master Motor Map (MMM) framework[1] [11] unifies the representation of human motion by decoupling the human motion capture process from further post-processing tasks. Human motion data can be normalized uniformly to the MMM human reference body model and easily transferred to different robot kinematics. In [12], we developed a framework to calculate the actuator requirements for human-like motions, by retargeting motion data from uni- and bimanual demonstrations to normalized robot kinematics, to compute joint positions, velocities, and accelerations, as well as joint torques using the normalized dynamic properties of the MMM reference model. In this paper, we extend our work by introducing a framework for humanoid robot design based on human motion data.

**Contribution:** *(i)* a humanoid robot design framework[2] with multiple customization options and evaluation criteria, both visual and numeric; *(ii)* a novel actuator optimization index based on the speed, acceleration, and torque requirements derived from the motion data; *(iii)* implementation of the framework in a publicly available custom-built computer program. The usefulness of the framework is demonstrated by the theoretical optimization of the actuators for our robots ARMAR-6 and ARMAR-7.

## II. Related Work

In the following, we review the state of the art in computer-aided robot design and actuator optimization.

### A. Computer Aided Robot Design

Robot design is a complicated engineering task often involving the consideration of many design variables and subject to a wide set of constraints [2]. Furthermore, the presence of many non-linear relationships between the design variables and the properties of the final result makes the search for an analytical solution extremely difficult [13]. To alleviate some of the complexities of robot design engineers use computer tools (e.g. implemented solutions, simulation software, advanced equations solvers, etc.) ([3], [14]).

*The authors contributed equally to the paper

This work has been supported by the Carl Zeiss Foundation through the JuBot project and the German Federal Ministry of Education and Research (BMBF) under the Robotics Institute Germany (RIG). The authors are with the High Performance Humanoid Technologies Lab, Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany. {klas,asfour}@kit.edu

[1] https://git.h2t.iar.kit.edu/sw/mmm

[2] https://git.h2t.iar.kit.edu/sw/computer-aided-robot-engineering

Computer-aided robot design can take one (or a combination) of the following forms: expert system, evolutionary algorithm, optimization problem, or design system. In this literature review, we focus on task-based design, meaning optimizing the robot for executing a certain set of predefined actions, as opposed to optimizing the structural integrity of the robot components, for example.

Expert systems aim to encapsulate a large body of knowledge about a certain topic which would normally be spread into many different sources [15]. They operate on the basis of a set of logical rules which is used to determine the output of the system based on the input [16]. More recently, [4] introduced an expert system, that can perform multi-stage reasoning on an ontological knowledge base, to help the design of humanoid robot components. This system executes a systematic search within the solution space, which is composed of catalog components, previous solutions, and possible combinations between those. Although expert systems have the advantage of being able to search a vast knowledge base quickly and can often perform many important design parameter calculations automatically, they have the drawbacks of only being able to reason within the knowledge that they possess.

Evolutionary algorithms (also called genetic algorithms) are a class of search methods, suited for solving complex optimization problems, by imitating natural evolution. Instead of evaluating one possible solution at a time, they rely on a population of solutions, thus executing a search in multiple directions at once. They use a fitness function to decide which individuals from a population to keep or to discard. A new population is created from the individuals kept over from the previous population by means of unary and binary operators [17]. To sum up, genetic algorithms are able to find very good solutions for optimization problems, which are extremely hard to analyze analytically. However, evaluation of the fitness of each population can be computationally demanding, since the chance of finding a good solution increases with the size of the population [17] and evolutionary algorithms are known to be inconsistent [18].

Formulating the design task as an optimization problem requires synthesising an optimization function, dependent on a finite set of optimization variables, subject to a known set of equality and/or inequality constraints. This is advantageous when it comes to task-based robot design since the optimization function can be specifically tailored to the movement which is required from the robot and the environment, in which the robot should operate [19]. Representing the task of robot design as an optimization problem allows the engineer a large degree of control over the design via the formulation of the objective function. Further investigating robot design as an optimization problem could be a viable way of developing highly skilled robots for specific tasks ([1], [18]).

Design systems (also design frameworks) present a simplified overview of the design process with a high level of abstraction. They aid engineers by splitting the design task into several steps starting from the system requirements and ending with a complete design concept [20]. A framework for interactive robot design was proposed by [21], which investigates the robot's operational space for task execution. Instead of optimizing an objective function, the algorithm iteratively translates the region of operational spaces accessible to the robot and informs the designer how to adjust robot parameters and avoid breaking constraints. This human-in-the-loop optimization allows the engineer to incorporate their domain-specific knowledge in terms of both the robot morphological and dynamic properties. Design systems allow the user a great deal of control over the final solution of the design problem, while still providing the necessary aid in terms of computational power.

### *B. Actuator Optimization*

Actuator selection is one vital aspect of robot design since the motor-gear combination has an enormous effect on the robot system's dynamic properties [10]. The actuator needs to output the proper torque, speed, and acceleration in order to drive the load in a way permitting the robot to achieve the desired performance ([22], [6]). Although designing application-specific components (such as motors and transmissions) could lead to optimal results for the specific engineering task, this is usually not a viable option for most robot designers, and therefore, a large part of the research is focused on finding the most optimal actuator from a database of motors and gears ([23], [24]). In that case, the engineer has no control over the structure of the components, so electromagnetic, power-loss, or temperature effects of the components are not considered in this paper.

The idea of maximum system acceleration was introduced by [25] as one of the earliest actuator optimization heuristics. They analytically calculated the optimal gear ratio for the highest acceleration to $i* = \sqrt{J_M/J_L}$. This is known as the principle of inertia match since the gear ratio is chosen such that the input inertia reflects the output inertia.

It can be observed that, the properties of an actuator are closely related to the choice of gear ratio [26], and consequently, many researchers choose to model the actuator selection as an optimization problem focused on finding the optimal gear ratio of the motor-gear system. This is further advantageous, since the actuator parameters could be represented in terms of the gear ratio (in the case where the motor is treated as fixed) and interpolated, where the exact values are unknown.

In the literature, there is a lack of publications which: a) investigate the actuators of humanoid robots, which present novel design challenges compared to industrial robots; and b) consider the requirements for human-like arm motions. The current work seeks to fill this gap in the literature by introducing an innovative framework for the design of humanoid robots with a focus on actuator selection and optimization based on requirements derived from human motion data.

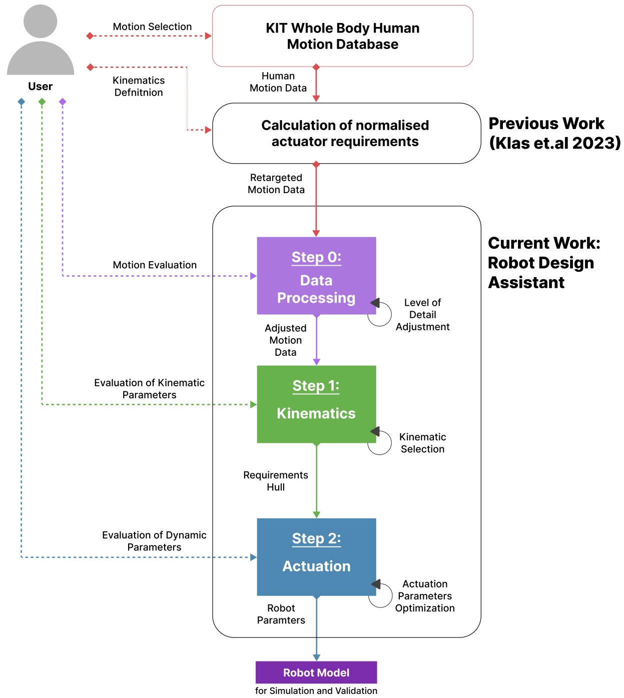


Fig. 1: Humandoid Robot Design Assistant - Framework Overview

## III. Approach

We propose a novel robot design framework for the engineering of humanoid robots based on requirements from human motion. As previously stated, motions from the KIT Whole-Body Human Motion Database [27] were selected and the normalized requirements for the actuators of humanoid kinematics were calculated in [12] by simulating how these kinematics would perform those movements. The current work can be classified as a design system, but it also includes an optimization step.

### A. Framework Overview

The principal structure of the system can be seen in Fig. 1. It consists out of three main components required for the data processing, the kinematics selection, and the actuator optimization. These three components are bound together by the start-up screen, which allows the user to quickly switch from window to window, based on which step they have completed. The robot designer has to provide human motion data and sequentially carry out each step in the given order. At each step the user can adjust the system settings until a desirable configuration is reached, before proceeding with the following step, allowing a high degree of customization and control over the robot design process. After the actuator optimization step, the user has the optimal actuator parameters and a suitable robot kinematic, which can be used for the purposes of testing and creation of higher-level robot model.

The MATLAB simulation data of the retargeted motions from the KIT Whole Body Human Motion Database [12], [27] are the source of the system presented here.

Furthermore, the system works with a database of predefined motor-gear configurations including KIT SAC units, which can be easily augmented if needed.

### B. Data Processing

The Data Processing step contains the mechanisms to reduce the level of detail of the motion data to a desired state and preview the changes made for each motion parameter: joint position, speed, acceleration, and torque over the whole motion. The motion data can be analyzed on a frame-by-frame basis for maximum precision. Since the focus here is the motion itself, all the data presented in this step is for the MMM reference model. The data and videos for all other robot kinematics can be viewed in the next step of the process. During this step the user can adjust the level of detail to make the following steps easier, as well as, preview how the motions would be executed by the MMM model and observe the required speed, acceleration, and torque for each joint, allowing them to make judgments about the complexity of the design task.

The motion data dimensions are in the form $\mathbf{R}^{n\times m\times p}$ where $n = 22 + 1$ is the number of joints (1 additional dimension is added to save the time variable), $m = 4$ for each position, speed, acceleration, and torque, and $p$ is equal to the number of time-steps of the simulation. One object with these dimensions is loaded for each available robot kinematic. Since each large movement type is made up of smaller movements, MATLAB interpolates between those for the sake of continuity. These interpolations, however, do not correspond to the actual movement gathered from the human motion data, and therefore should not be considered further, thus they are set to an empty data type. The data for the object torques is loaded in the same manner.

The MATLAB simulation output files have a very high level of detail, in that there are multiple joint position, speed, acceleration, and torque data points for each second. This exceeds the amount of information needed to derive meaningful information about the actuator requirements for the whole movement. In order to reduce the data to a more easily manageable amount, the user can perform a "Delete Small Time-Steps" operation , to achieve a desired level of detail. Two modes were implemented to achieve this: mode time-step deletes data points on an interval, specified by the user; mode norm deletes data points, for which the difference in magnitude of the value change from one point to the next is smaller than the user-specified value.

### C. Kinematic Arrangement Selection

In the Kinematic Selection window, the user can view the different motion requirements for all robot kinematics (for which the human motion data was retargeted) and compare them based on their actuator requirements hulls and the kinematic performance index. The user can also specify a subset of all available motions, which they wish to consider. The kinematics comparison is facilitated by the following information presented to the designer: joint position throughout the whole movement, normalized requirements hulls for each joint based on the required speed, acceleration, and torque to perform the movement, and the Kinematic Performance Index [12]. The requirement hulls are calculated with the help of alphashapes [28] to be more exact than the convex hulls of the data. The visual and quantitative modes of comparison give information about the trade-offs between the requirements for different robot kinematics and should allow the user to choose the best robot kinematic for the selected motions.

*1) Normalised Requirements Hull:* To find the alphashape, the values for speed, acceleration, and torque need to be normalized, in order for the calculation to be sufficiently accurate. To achieve this, the value of speed, acceleration, and torque for each time-step is divided by the respective limit (chosen based on all available robot kinematics and across all motions). This can be expressed mathematically for one speed data point as follows: $s^* = s_i/s_{max}$, where $s^*$ is the normalized speed value, $s_i$ the original data point speed value for the time-step $i$, and $s_{max}$ is the maximum speed value across all robot kinematics and loaded motions. In order to avoid short peak values, the data is limited to the maximum values for each specific robot kinematic. This is done by finding the ratio between the limit of the chosen kinematic and the highest limit across all robots in the form: $r_l = s^r_{max}/s_{max}$, where $r_l$ is the desired ratio and $s^r_{max}$ is the speed limit for the chosen robot kinematic $r$. Then, each value from the data points can be set to $r_l$, if it was previously larger than that.

In order to calculate the requirements hull we need to take into account the object torque, as well as the robot joint speed, acceleration, and torque. Even though the 3D hull in the speed-acceleration-torque is being displayed, it could be the case that some points with relevant object torque are left out, if only the robot data is used for the calculation. To prevent that, the following method is employed: Firstly, the 4D alphashape hull in the speed-acceleration-robot-torque-object-torque space is calculated. To get the array of points over which the 3D hull should be calculated, all unique points are extracted from the output of the 4D hull.

*2) Power Requirement Index:* The kinematic score (introduced in [12]) is a vital part of kinematic arrangement comparison, alongside the visual representation of the requirements hull. To calculate the kinematic score of a kinematic arrangement the maximum torque and speed values of each joint are needed. Because of symmetry constraints, the highest value for the limit from the left and right respective joints is considered. Additionally, the kinematic score is split into two values: one for the robot kinematic itself, and one for the object torque. The object torque score can be calculated as follows:

$$P = \sum_{n=0}^{N} j_{n\omega} \cdot j_{n\tau,o} \tag{1}$$

$$PRI_O = \frac{P_{robot,o}}{P_{MMM,o}} \tag{2}$$

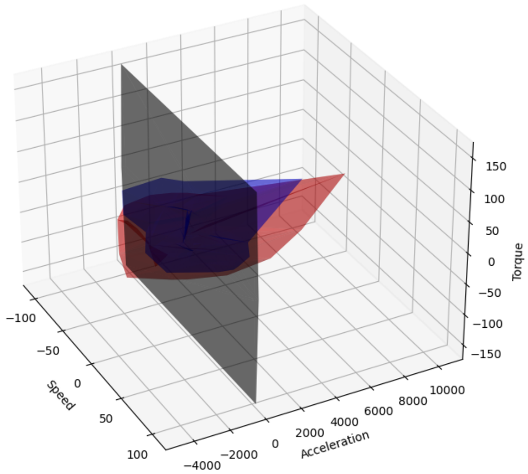


Fig. 2: Requirements hull of a robot with ARMAR-7 kinematics performing entertainment motions for the j1a joint (left: blue; right: red), compared with the capability hull (gray) of the U2 SAC unit (gear ratio: 120). In this case possible acceleration is insufficient for the rapid entertainment motions.

where $j_{n\tau,o}$ is the robot object torque, $P_{robot,o}$ is the object torque score for the selected robot kinematic, and $P_{MMM,o}$ is the object torque score for the MMM model. It should be noted, that the object torque score is only then non-zero, when at least one of the loaded motions is performed with an object.

The index is used to compare different dynamic, human-like, uni- and bimanual manipulation movements for 11 humanoid robot kinematics: LIMS2 [29], CENTAURO [30], iCub [31], David [32], Justin [33], HRP-4 [34], Valkyrie [35], ARMAR-III [36], ARMAR-4 [37], ARMAR-6 [38], ARMAR-7) and the New Design robot kinematics. The specifications of the MMM model are also given as reference (Note: the MMM kinematic has no mechanical realization as of the writing of this work).

The motion categories used for the comparison are the same as those used in our previous work [12].

## D. Actuator Selection and Optimization

To choose the optimal actuator configuration the user can select different motor-gear configurations for each joint in the actuator optimization window and observe how well they cover the requirements in the 3D speed-acceleration-torque plot not only visually (Fig. 2), but also based on the actuator optimization index. For a higher level of precision in the optimization, the user can use the automatic gear ratio optimization to find the gear ratio, for which the current actuator configuration will be most optimal in terms of satisfying the requirements. To accommodate for a wide range of robot designs, the user can freely select the desired robot height and arm weight. There is the option to specify if the actuator is within the arm or in the robot body, with the arm and robot weights being adjusted accordingly. This step allows the exploration of different actuator configurations, where the visual and quantitative feedback modes can provide valuable information about the viability of each configuration.

*1) Data Scaling:* The robot weight is a primary robot parameter. The robot torque data from the MATLAB simulation is calculated based on a 1kg reference so that it can later be scaled by the actual robot weight. To give the user maximum flexibility over the robot design, the input parameter for determining the final robot weight is the arm weight. The weight of all actuators, which lie within the arm, is added to the arm weight for the calculation of the resulting total robot weight. The arm segments of the MMM reference model account for 7% (14% for both arms) of the total robot mass. This value is used to calculate the resulting total robot weight from the total resulting arm weight.

The data scaling is performed based on the Froude number $Fr = \dfrac{v^2}{g \cdot l} = \dfrac{l}{g \cdot t^2}$, given by the ratio of the inertial force to the force of gravity [12], using the following scaling factors:

$$\lambda = \frac{1}{l_{user}}; \quad \mu = \frac{1}{w_{user}} \tag{3}$$

where $l_{user}$ is the robot height specified by the user; 1 is the height of the MMM robot model in m, and $w_{user}$ is the robot weight calculated as previously described.

$$\omega_t = \frac{1}{\sqrt{\lambda}} \cdot \omega_s; \quad \dot{\omega}_t = \frac{1}{\lambda} \cdot \dot{\omega}_s \tag{4}$$

where $\omega_t$ is the required speed; $\omega_s$ is the speed from the motion data; $\dot{\omega}_t$ is the required acceleration, and $\dot{\omega}_s$ is the acceleration from the motion data. The scaling factors for speed ($\omega$) and acceleration ($\dot{\omega}$) were derived from the scaling factor for time.

$$\tau_t = \frac{1}{\mu \cdot \lambda} \cdot \tau_s; \quad \tau_{o,t} = i_u \cdot \tau_{o,s} \tag{5}$$

where $\tau_t$ is the required torque; $\tau_s$ is the torque from the motion data; $\tau_{o,t}$ is the required object torque, $\tau_{o,s}$ is the object torque from the motion data, and $i_u$ is the scaling factor from the user input.

These factors are used over all data points to arrive at the real actuator requirements data.

*2) Requirements & Motor Hulls:* The requirements hull is derived based on the scaled requirements data and used for the calculation of the actuator optimization index. One last step for the acquisition of the requirements data is the addition of the robot and object torques for each time-step. In order to allow the user to separately scale those, it is impossible to consolidate them until right before the calculation of the requirements hull. Consequently the 3D array for a time-step $i$ in the data looks like this: $[S_i, A_i, T_i, r + T_i, o]$, where $S_i$ is the speed value, $A_i$ the acceleration value, $T_i, r$ the robot torque value, and $T_i, o$ the object torque value.

For the calculation of the actuator optimization index the motor parameters hull is also needed. The motor hull is constructed based on the speed, acceleration, and torque values of the motor-gear combination. The motor is treated as ideal, in the sense that it can produce the same negative values as positive. Therefore, for the motor hull calculation, the 3D vertices given in Table I are used, where $S$ is the speed parameter, $A$ is the acceleration parameter, and $T$ is the torque parameter of the motor.

TABLE I: Motor Hull Vertices

| $\omega$ | $\dot{\omega}$ | $\tau$ |
|---|---|---|
| $S$ | $0$ | $T$ |
| $S$ | $0$ | $-T$ |
| $S$ | $A$ | $0$ |
| $S$ | $-A$ | $0$ |
| $-S$ | $-A$ | $0$ |
| $-S$ | $A$ | $0$ |
| $-S$ | $0$ | $-T$ |
| $-S$ | $0$ | $T$ |

These parameters are dependent on the gear ratio for a chosen motor-gear configuration. Gear ratio values that are not found in the manufacturer's catalog are interpolated based on the known values, to allow finding new optimal gear ratio values. It was taken into account that it is not uncommon for gear manufacturers to be able to create gears with user-specified gear ratios.

*3) Actuator Optimization Index:* The Actuator Optimization Index is based on finding the intersection between the motor hull and the requirements hull in the 3D speed, acceleration, and torque space. Mathematically this can be expressed as follows:

Let $\mathbf{M}$ be the convex hull (alphashape hull with $\alpha = \infty$) of the motor parameters and $\mathbf{R}$ be the alphashape hull (with $0 < \alpha < \infty$ ) of the motion requirements both in $\mathrm{R}^{\omega \times \dot{\omega} \times \tau}$ space where $V_{\mathbf{I}}$ is the volume of the intersection and $V_{\mathbf{R}}$ is the volume of the requirements hull. Then the intersection of those two volumes is a new volume: $V_{\mathbf{I}} = V_{\mathbf{R}} \cap V_{\mathbf{M}}$. The actuator optimization index ($A.O.I.$) is calculated to:

$$A.O.I. = \frac{V_{\mathbf{I}}}{V_{\mathbf{R}}} \tag{6}$$

In effect, the ratio describes what part of the requirements hull is covered by the selected actuator configuration.

In the automatic sequence for the optimization of the gear ratio the speed, acceleration, and torque interpolation functions are calculated based on the data from the actuators table in the form: $f_1(i) = \omega_p$, $f_2(i) = \dot{\omega}_p$, and $f_3(i) = \tau_p$. Afterward, a SciPy implementation of the Powell's method [39] is used to arrive at a new optimal gear ratio ($i$) candidate. In short, Powell's method iteratively searches down directions, which are linearly independent, based on the best-known approximation of the minimum. Powell's method does not require the calculation of derivatives, which makes it a suitable candidate for our use-case. The new speed, acceleration, and torque parameters of a motor with the candidate gear ratio are calculated with the help of the interpolation functions. Based on those, a new motor hull volume ($V_{\mathbf{M}}$) is calculated, which is used to calculate the new intersection ($V_{\mathbf{I}}$) with the requirements hull volume ($V_{\mathbf{R}}$). Finally, the ratio of the volumes of the new intersection and the requirements hull is computed.

After an optimal value has been produced a value correction follows to accommodate for the physical realization of the transmission. The value correction consists of rounding up the result of the automatic optimization to a number with one decimal place after the comma and calculating the new resulting actuator optimization index. The bound of the optimization algorithm are set to $[i_{min}; i_{max}]$, where $i_{min}$ is the minimum value of the gear ratio for the chosen actuator configuration from the database, and $i_{max}$ is the maximum.

The gear ratios of all upper-body limb joints can be selected manually or optimized, according to the described procedure. The result of the final step is a robot actuator configuration, which can be used for further analysis and simulation.

## IV. Analysis

The framework was used for a theoretical optimization of the ARMAR-6 and ARMAR-7 upper-body actuators. The data level of detail adjustment was performed with the use of the mode norm Delete Small Time-steps algorithm with the norm value 0.25. The object torque scaling was set to 100% and the robot weight and height were set to the ARMAR-6 and ARMAR-7 robot specifications. The specifications of the robot kinematics are shown in Table II, where $\lambda$ is the scaling factor based on the MMM reference model; and the MMM model specifications are included for comparison. Two sets of evaluations were performed for each robot: one, for the motions Entertainment (E), Household (H), Factory (F), and Conversation (C); and another one, for the motions Entertainment and Household (for a full breakdown of the movements see [12]). Each step of the framework was performed for each robot and the gear ratios for the actuators of each upper-limb joint were optimised. This was done to showcase how the framework aids task-based design by providing meaningfully different results, based on the selected motions.

TABLE II: Specifications of Robot Kinematic Examples Used for Evaluation

| Name | jR | j1 | j2 | j3 | Torso | $\lambda$ |
|---|---|---|---|---|---|---|
| ARMAR-7 | jRC, jR1 | xy | xy | quat | Pz | 0.41 |
| ARMAR-6 | jRC, jR1 | xy | xz | xy | Pz | 0.31 |
| MMM | jR0 | xy | xy | xy | Rx, Ry, Pz | 1.00 |

The results of the automatic gear ratio optimization can be seen in Table III, Table IV, Table V, and Table VI, where "SAC" is the joint specific KIT SAC Unit, $i_C$ is the current gear ratio (for the selected SAC Unit), $A.O.I._C$ is the actuator optimization index for the currently selected gear ratio, $i_{O*}$ is the value-corrected gear ratio calculated by the automatic optimization algorithm, $A.O.I._{O*}$ is resulting A.O.I. for the optimal value-corrected gear ratio, and $\%*$ is the change from the selected gear ratio to the optimal one.

As can be observed, the motor parameters calculated based on the gear ratio results from the automatic optimization procedure produce a higher overlap between the motor hull and the requirements hull. It can be seen, that the optimal gear ratios are mostly smaller than the currently selected ones. This is explained by the fact that the motion data sometimes includes large accelerations and lower transmission ratios

TABLE III: ARMAR-6 Upper-body Joint Actuators Optimization (E), (F), (C), (H)

| Joint | SAC | $i_C$ | $A.O.I._C$ | $i_{O*}$ | $A.O.I._{O*}$ | %* |
|---|---|---|---|---|---|---|
| $J_{1a}$ | L | 160 | 0.035 | 44.1 | 0.110 | 214% |
| $J_{1b}$ | L | 160 | 0.048 | 50.2 | 0.124 | 158% |
| $J_{2a}$ | M | 160 | 0.072 | 50.2 | 0.165 | 129% |
| $J_{2b}$ | M | 160 | 0.050 | 50.0 | 0.146 | 192% |
| $J_{3a}$ | S | 160 | 0.149 | 50.0 | 0.333 | 123% |
| $J_{3b}$ | S | 160 | 0.103 | 67.0 | 0.147 | 43% |
| $J_R$ | M | 160 | 0.027 | 50.0 | 0.068 | 152% |
| $J_C$ | L | 160 | 0.060 | 54.7 | 0.137 | 128% |
| | | | | | Avg. | +142% |

TABLE IV: ARMAR-6 Upper-body Joint Actuators Optimization (E), (H)

| Joint | SAC | $i_C$ | $A.O.I._C$ | $i_{O*}$ | $A.O.I._{O*}$ | %* |
|---|---|---|---|---|---|---|
| $J_{1a}$ | L | 160 | 0.104 | 45.1 | 0.108 | 3% |
| $J_{1b}$ | L | 160 | 0.036 | 49.5 | 0.096 | 167% |
| $J_{2a}$ | M | 160 | 0.029 | 50.5 | 0.065 | 124% |
| $J_{2b}$ | M | 160 | 0.055 | 50.0 | 0.164 | 198% |
| $J_{3a}$ | S | 160 | 0.120 | 51.4 | 0.232 | 93% |
| $J_{3b}$ | S | 160 | 0.142 | 62.8 | 0.228 | 61% |
| $J_R$ | M | 160 | 0.034 | 50.0 | 0.093 | 174% |
| $J_C$ | L | 160 | 0.071 | 56.7 | 0.155 | 118% |
| | | | | | Avg. | +117% |

TABLE V: ARMAR-7 Upper-body Joint Actuators Optimization (E), (F), (C), (H)

| Joint | SAC | $i_C$ | $A.O.I._C$ | $i_{O*}$ | $A.O.I._{O*}$ | %* |
|---|---|---|---|---|---|---|
| $J_{1a}$ | U1 | 160 | 0.077 | 65.6 | 0.140 | 82% |
| $J_{1b}$ | U1 | 160 | 0.104 | 68.5 | 0.176 | 69% |
| $J_{2a}$ | U1 | 100 | 0.074 | 62.1 | 0.095 | 28% |
| $J_{2b}$ | HD8 | 166 | 0.204 | 58.1 | 0.339 | 66% |
| $J_{3a}$ | A7W | 94 | 0.278 | 92.0 | 1.000 | 260% |
| $J_{3b}$ | A7W | 94 | 0.492 | 81.3 | 0.500 | 2% |
| $J_R$ | U1 | 100 | 0.076 | 56.3 | 0.107 | 41% |
| $J_C$ | U1 | 160 | 0.068 | 72.6 | 0.109 | 60% |
| | | | | | Avg. | +76% |

TABLE VI: ARMAR-7 Upper-body Joint Actuators Optimization (E), (H)

| Joint | SAC | $i_C$ | $A.O.I._C$ | $i_{O*}$ | $A.O.I._{O*}$ | %* |
|---|---|---|---|---|---|---|
| $J_{1a}$ | U1 | 160 | 0.081 | 62.7 | 0.153 | 89% |
| $J_{1b}$ | U1 | 160 | 0.131 | 69.6 | 0.221 | 69% |
| $J_{2a}$ | U1 | 100 | 0.260 | 59.4 | 0.348 | 34% |
| $J_{2b}$ | HD8 | 166 | 0.365 | 60 | 0.568 | 56% |
| $J_{3a}$ | A7W | 94 | 0.711 | 91.1 | 0.712 | 0.1% |
| $J_{3b}$ | A7W | 94 | 0.650 | 89.8 | 0.651 | 0.1% |
| $J_R$ | U1 | 100 | 0.080 | 57.4 | 0.111 | 29% |
| $J_C$ | U1 | 160 | 0.177 | 73.5 | 0.278 | 57% |
| | | | | | Avg. | +42% |

produce higher accelerations. It can be seen that ARMAR-6 was specifically designed for industrial applications, where slower movements and higher forces are needed. Hence, lower transmission ratios are reasonable in the case of fast conversation motions.

This example application of the proposed framework shows that it is a useful tool to derive optimal actuator parameters based on motion requirements from a set of motions. It also displays the practicality of the proposed actuator optimization index in comparing different actuator configurations and highlights the usefulness of an automatic gear ratio optimization, which can find new possible solutions.

## V. Discussion

In conclusion, this paper presents a novel framework for the kinematic selection and comparison and actuator selection and optimization for the design of humanoid robots based on actuator requirements, calculated from human motion data. The proposed framework provides the robot designer with a significant range of creative freedom, compared to other computer aided methods.

Furthermore, the engineer receives immediate feedback based on the parameters they alter, which can save a lot of time, especially in the early stages of the design process. This allows the user to quickly remove sub-optimal solutions and focus on those with more potential. Similarly to an expert system, this framework works with a database of human motion data and a table of actuator configurations (our SAC units). However, the KIT Whole Body Human Motion database [27] is constantly being extended with new motions and objects. Additionally the system can be applied to totally new kinematics and actuators, making it much more flexible.

The proposed Actuator optimization Index, in contrast to other actuator optimization approaches, includes the required acceleration for the movement (in addition to the speed and torque), which could bridge the gap between robot and human trajectories. It is also supplemented by a visual representation of intersection of the requirements hull with the motor hull in the speed-acceleration-torque space. Allowing comparison in multiple ways (in this case: numerically and visually), enables discovering meaningful differences between the inspected configurations.

Outside the purview of this framework are design problems regarding topics such as robot control, trajectory optimization, and structural integrity. However, the robot model derived from the optimal parameters, which this framework can deliver, should serve as a valuable blueprint for investigating the optimal solutions for other engineering problems concerning humanoid robot design.

One significant improvement to the proposed framework would be the addition of an optional kinematics optimization step. It would involve a new program window, which allows the user to specify the desired robot joints, their position along the arm, and the joint types. After deciding on the kinematic arrangement, the user would export it to MATLAB, where the necessary actuator requirements for the selected motions would be calculated. Then, following the steps outlined in this work, the user would be able to compare the kinematic indices of the newly designed kinematic arrangement against all other known ones and decide if it is worth pursuing its design further. This would allow for the co-optimization of kinematic and actuator configurations, thereby increasing the design space for new humanoid robots significantly.

Additionally, since the output of the framework is the optimal robot parameter set for the kinematic and actuator configurations, the possibility to export the robot model could be added. The robot model could then be imported into the MATLAB simulation for the design to be inspected further. This simulation could inform other design decisions (such as actuator placement, robot weight distribution, etc.) on a more practical level, further supporting the design process and making the design of humanoid robots a less daunting endeavor. In the future, the system could be extended to the lower body if required, but this brings additional complexity due to the interaction forces with the environment that are not currently available in the database. Linear actuation of the arms is not yet integrated but could be added in the future, as is already the case with the mobile base. Future work should include the application of the framework to real robots to validate its effectiveness and solve practical implementation problems.

## References


[1] 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 XIII*, ser. RSS2017. Robotics: Science and Systems Foundation, July 2017.

[2] A. Schulz, C. Sung, A. Spielberg, W. Zhao, R. Cheng, E. Grinspun, D. Rus, and W. Matusik, "Interactive robogami: An end-to-end system for design of robots with ground locomotion," *The International Journal of Robotics Research*, vol. 36, no. 10, p. 11311147, Aug. 2017.

[3] J. Ziglar, R. K. Williams, and A. Wicks, "Context-aware system synthesis, task assignment, and routing," *Autonomous Robots*, vol. 47, no. 2, p. 193210, Dec. 2022.

[4] O. Karrenbauer, S. Rader, and T. Asfour, "An ontology-based expert system to support the design of humanoid robot components," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, Nov. 2018.

[5] A. Spielberg, B. Araki, C. Sung, R. Tedrake, and D. Rus, "Functional co-optimization of articulated robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2017.

[6] S. Rezazadeh and J. W. Hurst, "On the optimal selection of motors and transmissions for electromechanical and robotic systems," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sept. 2014.

[7] V. Vodovozov and J. Laugis, "Object-oriented electric drive development technology," in *2007 IEEE International Electric Machines I&; Drives Conference*. IEEE, May 2007.

[8] E. S. Barjuei, M. M. G. Ardakani, D. G. Caldwell, M. Sanguineti, and J. Ortiz, "Optimal selection of motors and transmissions in back-support exoskeleton applications," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 3, p. 320330, Aug. 2020.

[9] L. Zhou, S. Bai, and M. R. Hansen, "Design optimization on the drive train of a light-weight robotic arm," *Mechatronics*, vol. 21, no. 3, p. 560569, Apr. 2011.

[10] A. Bowling and O. Khatib, "Actuator selection for desired dynamic performance," in *IEEE/RSJ International Conference on Intelligent Robots and System*, ser. IROS-02. IEEE, 2002.

[11] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, "Unifying representations and large-scale whole-body motion databases for studying human motion," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 796–809, 2016.

[12] C. Klas, A. Meixner, D. Ruffler, and T. Asfour, "On the actuator requirements for human-like execution of retargeted human motion on humanoid robots," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, Dec. 2023.

[13] A. Sathuluri, A. V. Sureshbabu, and M. Zimmermann, "Robust co-design of robots via cascaded optimisation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2023.

[14] C. Pupăză, G. Constantin, and S. NEGRILĂ, "Computer aided engineering of industrial robots," *Proceedings in Manufacturing Systems*, vol. 9, no. 2, pp. 87–92, 2014.

[15] E. Olier, "Conceptual design for space robots using expert systems," *IFAC Proceedings Volumes*, vol. 18, no. 16, p. 145150, Nov. 1985.

[16] P. Bhatia, J. Thirunarayanan, and N. Dave, "An expert system-based design of scara robot," *Expert Systems with Applications*, vol. 15, no. 1, p. 99109, July 1998.

[17] G. Renner and A. Ekrt, "Genetic algorithms in computer aided design," *Computer-Aided Design*, vol. 35, no. 8, p. 709726, July 2003.

[18] J. Whitman and H. Choset, "Task-specific manipulator design and trajectory synthesis," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, p. 301308, Apr. 2019.

[19] J. Kim, "Task based kinematic design of a two dof manipulator with a parallelogram five-bar link mechanism," *Mechatronics*, vol. 16, no. 6, p. 323329, July 2006.

[20] H. Komoto and T. Tomiyama, "A framework for computer-aided conceptual design and its application to system architecting of mechatronics products," *Computer-Aided Design*, vol. 44, no. 10, p. 931946, Oct. 2012.

[21] B. Canaday, S. Zapolsky, and E. Drumwright, "Interactive, iterative robot design," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2017.

[22] G. Cusimano, "A procedure for a suitable selection of laws of motion and electric drive systems under inertial loads," *Mechanism and Machine Theory*, vol. 38, no. 6, p. 519533, June 2003.

[23] P. Chedmail and M. Gautier, "Optimum choice of robot actuators," *Journal of Engineering for Industry*, vol. 112, no. 4, p. 361367, Nov. 1990.

[24] F. Roos, H. Johansson, and J. Wikander, "Optimal selection of motor and gearhead in mechatronic applications," *Mechatronics*, vol. 16, no. 1, p. 6372, Feb. 2006.

[25] K. A. Pasch and W. P. Seering, "On the drive systems for high-performance machines," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 106, no. 1, p. 102108, Mar. 1984.

[26] H. Giberti, S. Cinquemani, and G. Legnani, "A practical approach to the selection of the motor-reducer unit in electric drive systems," *Mechanics Based Design of Structures and Machines*, vol. 39, no. 3, p. 303319, July 2011.

[27] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, "The kit whole-body human motion database," in *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, July 2015.

[28] H. Edelsbrunner and E. P. Mcke, "Three-dimensional alpha shapes," in *Proceedings of the 1992 workshop on Volume visualization - VVS 92*, ser. VVS 92. ACM Press, 1992.

[29] H. Song, Y.-S. Kim, J. Yoon, S.-H. Yun, J. Seo, and Y.-J. Kim, "Development of low-inertia high-stiffness manipulator lims2 for high-speed manipulation of foldable objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2018.

[30] N. Kashiri, S. Cordasco, P. Guria, A. Margan, N. G. Tsagarakis, L. Baccelliere, L. Muratore, A. Laurenzi, Z. Ren, E. M. Hoffman, M. Kamedula, G. F. Rigano, and J. Malzahn, "Centauro: A hybrid locomotion and high power resilient manipulation platform," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, p. 15951602, Apr. 2019.

[31] D. Shah, Y. Wu, A. Scalzo, G. Metta, and A. Parmiggiani, "A comparison of robot wrist implementations for the icub humanoid," *Robotics*, vol. 8, no. 1, p. 11, Feb. 2019.

[32] M. Grebenstein, A. Albu-Schaffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger, H. Hppner, S. Jrg, M. Nickl, A. Nothhelfer, F. Petit, J. Reill, N. Seitz, T. Wimbck, S. Wolf, and G. Hirzinger, "The dlr hand arm system," 06 2011, pp. 3175 – 3182.

[33] C. Ott, M. A. Roa, F. Schmidt, W. Friedl, J. Englsberger, R. Burger, A. Werner, A. Dietrich, D. Leidner, B. Henze, O. Eiberger, A. Beyer, B. Buml, C. Borst, and A. Albu-Schffer, *Mechanisms and Design of DLR Humanoid Robots*. Springer Netherlands, 2017, p. 126.

[34] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi, and N. Kanehira, "Humanoid robot hrp-4 - humanoid robotics platform with lightweight and slim body -," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sept. 2011.

[35] N. Radford, P. Strawser, K. Hambuchen, J. Mehling, W. Verdeyen, A. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, R. Berka, R. Ambrose, M. Markee, N. Fraser-Chanpong, C. Mcquin, J. Yamokoski, S. Hart, R. Guo, A. Parsons, and J. Akinyode, "Valkyrie: Nasa's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, pp. 397–419, 05 2015.

[36] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-iii: An integrated humanoid platform for sensory-motor control," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Dec. 2006.

[37] T. Asfour, J. Schill, H. Peters, C. Klas, J. Bucker, C. Sander, S. Schulz, A. Kargov, T. Werner, and V. Bartenbach, "Armar-4: A 63 dof torque controlled humanoid robot," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, Oct. 2013.

[38] T. Asfour, M. Waechter, L. Kaul, S. Rader, P. Weiner, S. Ottenhaus, R. Grimm, Y. Zhou, M. Grotz, and F. Paus, "Armar-6: A high-performance humanoid for human-robot collaboration in real-world scenarios," *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 108–121, 2019.

[39] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The Computer Journal*, vol. 7, no. 2, p. 155162, Feb. 1964.