An Ontology-Based Expert System to Support the Design of Humanoid Robot Components

Oliver Karrenbauer*, Samuel Rader* and Tamim Asfour

***Abstract*— The design of humanoid robots is a complex, challenging and time-consuming task. Due to conflicting requirements, such as human-like capabilities within human dimensions, the design of humanoid robots relies highly on the experience and expert knowledge of the engineers. This paper presents an expert system framework that allows to store this knowledge in order to reuse it for the systematic design of humanoid robot components. Based on user requirements, the system executes a multi-stage reasoning on an ontological knowledge base: Partial solutions are generated by integrating existing catalog components into potential concept solutions. After checking logical and physical constraints as well as calculating properties, these partial solutions are either discarded or combined in a bottom-up way to generate valid solutions that are then visualized by a user interface. We evaluate the developed system in terms of its capability to reproduce available solutions for state-of-the-art sensor-actuator units used in several robots as well as its capability to optimize the design of such units.**

I. Introduction

The design of humanoid robots remains a complex, time-consuming and challenging task. The design of versatile humanoid robots does not only require the realization of intelligent behavior, but also the development of suitable hardware components. One of the main challenges are conflicting requirements, such as the high integration of numerous mechatronic subcomponents into human-like dimensions. Taking into account the interfaces and physical constraints of each subcomponent, the complexity increases further since selection and arrangement can not be made separately. Consequently, humanoid robot design relies highly on the experience and expert knowledge of robot engineers, who have to find trade-offs to fulfill different requirements.

We propose a novel expert system framework that supports the design process of humanoid robots through systematic search within the solution space. This solution space consists of previous solutions, catalog components (motors, gears, sensors, etc.) and their possible combinations to fulfill given user requirements. Based on these user requirements, the expert system starts a multi-stage reasoning process that executes rules in a bottom-up graph search by generating, combining and discarding partial solutions until complete, valid design solutions are found (Figure 1). The rule set is not limited to logical and physical constraints, but also supports different concept solutions. Consequently, existing design solutions, which are already available in the system, can be combined in a novel way to create new solutions. By the use of an ontological knowledge base for subcomponents and rules, the expert system can be easily extended. As a validation for our approach, we present results regarding the design of sensor-actuator units (SA units) for humanoid robot joints. Since SA units usually include many of the most important subcomponents of the whole robot within human-like dimensions, they have proven to be one of the most challenging components in the development of our own humanoid robots, see [1] and [2]. The experience gained in these developments, especially the knowledge about suitable catalog components and a rule set describing possibilities for their integration into an overall system, is stored in the knowledge base of the expert system.

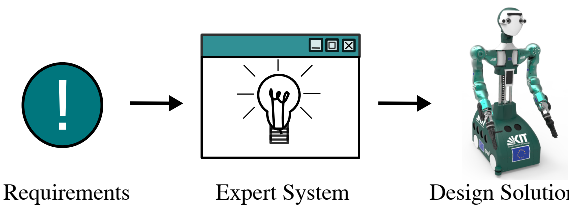


Fig. 1: Expert system to support the design of humanoid robots

In summary, the main contributions are (1) an expert system framework that supports the systematic design of humanoid components by using a novel ontological multistage-reasoning approach that integrates existing catalog components into an overall system and (2) a detailed expert system for the design of SA units, which serves as validation for a complex humanoid robot design problem. The full source code of the expert system is publicly available.[1]

The remainder of this paper is organized as follows. In Section II we present related work in the field of expert systems with a focus on the support of robotic and mechatronic design. Section III describes the expert system's architecture and the reasoning process. To evaluate the framework in Section IV, we present a SA unit expert system, which is used to reproduce and optimize existing SA units. Section V concludes the paper and presents ideas for future work.

II. Related Work

Expert systems are located in the field of artificial intelligence. Core of expert systems is the representation of the knowledge base which has to be machine readable for automated reasoning. The reasoning is done by an inference

The research leading to these results has received funding from the European Unions Horizon 2020 Research and Innovation Programme under grant agreement 643950 (SecondHands).

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany {rader, asfour}@kit.edu

*The first two authors contributed equally to this work.

[1] github.com/OliverKrr/ES-Robot-Component-Design

engine which is, beside the knowledge base and user interface, one typical component of an expert system [3].

There are some early expert systems that allow an effective selection of robotic systems, such as [4] [5] for common types of industrial robots and [6] for robot grippers. They use technical requirements with multiple attribute decision making to choose the best fitting robot. Selection can be seen as a subproblem of design, by evaluating existing individuals of one component rather than constructing and evaluating the combination of multiple subcomponents.

Automated support of design is done on a conceptual and detailed level. On a conceptual design level, given requirements are used to infer potential design solutions. Actual physical realizability is evaluated on an abstract level mostly on the basis of abstract geometric models of the subcomponents, which are typically modeled as a hierarchical tree. Different systems have been developed for the conceptual design, like a system for the design of SCARA robots [7], a process model to support innovation in conceptual design [8], an evaluator for abstract geometric models [9], automated mechanism design based on kinematic building blocks [10] and an evolutionary approach which simulates kinematics to realize a rapid dynamic locomotion on a legged robot [11]. Conceptual design can lead to problems in the realization of highly integrated systems, as they occur in humanoid robotics, due to a vague estimation of the construction space.

On a detailed level, existing subcomponents with their dimensions and interfaces are used to evaluate if their combination is feasible. Based on exact models of the subcomponents, realistic dimensions are calculated. Myung et al. [12] built an expert system that interacts with a CAD model. The user can configure partial subcomponents, but their selection and combination are made by the user rather than automatically based on requirements. Other computational approaches realize robotic kinematics based on a limited selection of modular elements and simple 3D printed parts [13] [14].

To implement a detailed design approach, the required formulas for the humanoid robot component have to be formalized. Formulas based on technical requirements were modeled for a motor-gearbox combination [15] and a spindle box system [16]. They combine the physical constraints between the subcomponents, the requirements influencing the design and the expert knowledge to build such systems. However, the formulas have still to be applied by an engineer for different configurations. An automated evaluation of the formulas by a rule-based system could reduce his effort.

The models for conceptual and detailed design are mostly based on trees [8], [9], [12]–[14], representing the hierarchy of properties and subcomponents. Bastinos et al. [17] modeled a multi-criteria decision making process, but only for selection of one individual, through a decision tree in an ontology. Ontologies formally model concepts and their relation in a domain of discourse. Building the knowledge base with an ontology allows the direct integration into other systems, e. g. an ontology for a conceptual robot design based on robot actions and requirements to select fitting structural robot parts and generate controllers [18], a high-level model of robotic embodiments [19] and an ontology of mechatronic systems components with their influencing parameters required during the design [20].

III. Approach

The related work shows that hierarchical tree-like decomposition of components and criteria lead to effective design solutions on a conceptual design level. Ontologies are used as knowledge representation for modeling different areas in the field of robotics. Our approach is to build an ontology-based expert system for the design of humanoid robot components on a detailed level (Figure 2). The user selects the robot component to be designed in a user interface. Depending on the selection, different sets of requirements can be specified. Chosen requirements can be prioritized by weight and allowed deviations can be set. The reasoning is controlled by an inference engine based on the OWL API and using the Pellet Reasoner [21]. It executes the multi-stage reasoning on an ontological knowledge base that includes expert knowledge about requirements and subcomponents which are required to infer design solutions. At the end of a reasoning procedure, the resulting design solutions are visualized by the user interface, including subcomponents, structural options and calculated properties.

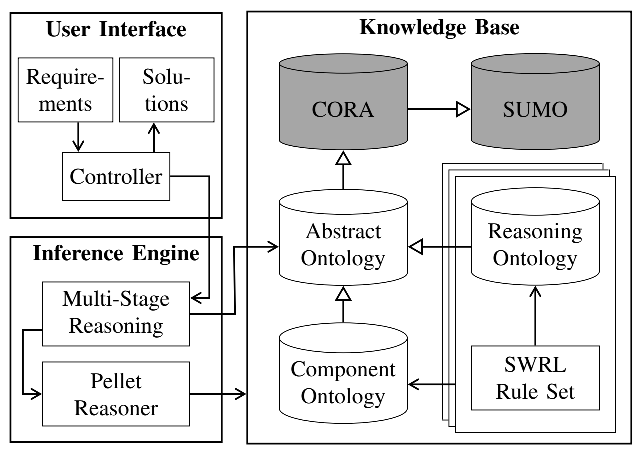


Fig. 2: UML architecture diagram of the expert system

A. *Ontological Knowledge Base*

The knowledge base contains expert knowledge to design humanoid robot components with their relations to requirements and subcomponents. Ontologies provide a common language for the developers of the expert system and the robot engineers providing the domain knowledge. They allow for easy changes and interoperability in an easy-to-use development environment like Protégé editor, but are still formal for automated reasoning. The knowledge base can be constantly extended, e. g. by adding new components and structural options. Reasoned design solutions of humanoid robot components can reflow back in the system as a subcomponent for higher-level components. Thereby, ontologies are a good choice for modeling the complex and dynamic field of humanoid robot design. Our knowledge base encompasses different ontology types (Table I) formalized

TABLE I: Examples of classes of the different ontologies

Ontology	Example (OWL 2)
Abstract	*SubClassOf (AbstractNode, SUMO:Abstract)*
	SubClassOf (AbstractNode,
	restriction (hasChild, someValuesFrom (SatisfiedNode))
Component	*SubClassOf (ILM85x13, Motor)*
	SubClassOf (ILM85x13,
	restriction (has_n_M_max, hasValue(2900))
Reasoning	*SubClassOf (MotorGearboxMatch, StageNode)*
	SubClassOf (MotorGearboxMatch,
	restriction (hasMotor, someValuesFrom (SatisfiedMotor))

with OWL 2 [22]. During runtime the ontologies will be composed to one working ontology to infer design solutions by the reasoner. The abstract ontology describes the concept of humanoid robot components, their subcomponents and the multi-stage reasoning process (Figure 3). It is based on the Suggested Upper Merged Ontology (SUMO) and Core Ontology for Robotics and Automation (CORA) [23] and acts as an interface to the inference engine for minimal dependencies which allows exchangeability and encapsulation to the humanoid robot component to be designed. The component ontology models existing subcomponents whose properties are taken from manufacturer catalogs. The ontology can be matched against existing ontologies where subcomponents are modeled. Furthermore, the component ontology can be used as standalone in other applications. Each reasoning ontology models the actual multi-stage reasoning tree and requirements affecting a specific humanoid robot component. The actual multi-stage reasoning is extended by a SWRL [24] rule set. The rules reflect the physical constraints, user requirements and expert knowledge for feasible design arrangements. In each step, calculations, evaluations and discarding of partial results are performed.

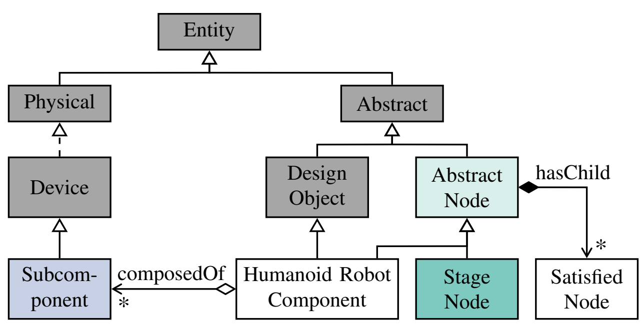


Fig. 3: UML class diagram of the abstract ontology which is based on the upper ontologies SUMO and CORA (dark gray)

B. Multi-Stage Reasoning

A humanoid robot component is composed of multiple subcomponents which can be arranged in different structural options. Depending on the physical constraints and the requirements of the user, not all possible design solutions are realizable. A brute force approach would be to generate each permutation of subcomponents and evaluate its feasibility. However, this is inefficient and leads to a combinatorial explosion. Instead of generating each permutation at once, it is more effective to generate permutations only for a subset of the subcomponents. This subset is based on the physical constraints, requirements and the expert knowledge in order to discard subcombinations at an early stage. The resulting subsets of subcomponents are ordered as a graph-based tree and can be traversed in a bottom-up approach. Consequently, the humanoid robot component is build up incrementally. The reasoning tree is a directed acyclic tree $G = (V, E)$.

$$V = \{(v_1, .., v_k), (v_{k+1}, .., v_{n-1}), v_n\} \tag{1}$$

$$E = \{\mathit{hasChild}\} \tag{2}$$

The vertexes $(v_1, .., v_n)$ represent different entities of the reasoning process (Figure 3) with the root v_n representing the $\mathit{HumanoidRobotComponent}$. The existing $\mathit{Subcomponents}$ are the leafs of G represented as $(v_1, .., v_k)$. Each other $\mathit{StageNode}$ $(v_{k+1}, .., v_{n-1})$ represents the dependencies between the subcomponents and other nodes. The edges E are composed of concrete subclasses of the abstract $\mathit{hasChild}$ object property between the nodes V. The multi-stage reasoning (Algorithm 1) is executed for a specific graph G depending on the humanoid robot component v_n. G is bottom-up sorted. At the beginning, an individual for each subcomponent node $(v_1, .., v_k)$ is created. In the next step, $\mathit{getIndividuals}(v_j)$ is called for a given node v and each subnode v_j. This function uses the reasoner to evaluate which individuals of v_j have satisfied the rule set. The rule set is modeled in a way, that for each v_j, calculations and evaluation of constraints are executed. When v_j satisfies the constraints it will be assigned to a respective $\mathit{SatisfiedNode}$, for which $\mathit{getIndividuals}(v_j)$ is called. Only if each v_j has satisfied individuals ($I \neq \emptyset$), the node v will be further processed. A set of permutations P is constructed with $\mathit{generatePermutations}(I[\,])$ for each individual over the set of $\{v_j\}$, e. g. a motor-gearbox combination has two $\mathit{hasChild}$ dependencies to a motor and a gearbox. Each individual of a motor is then combined with each individual of a gearbox. For each permutation p a new individual (i_{parent}) is created with the type of v (the motor-gearbox combination). The edges $\{e = (v, v_j)\}$ are mapped to a new

Algorithm 1 Multi-Stage Reasoning

procedure REASONING($G = (V, E)$) ▷ G sorted
  for all $\{v \mid v \in \{v_1, .., v_k\}\}$ **do**
    $\mathit{createIndividuals}(v.type)$
  for all $\{v \mid v \in \{v_{k+1}, .., v_n\}\}$ **do**
    for all $\{v_j \mid v_j \in V, e \in E, e = (v, v_j)\}$ **do**
      $I[v_j] := \mathit{getIndividuals}(v_j)$
    if $\forall I \in I[\,] \mid I \neq \emptyset$ **then**
      $P := \mathit{generatePermutations}(I[\,])$
      for all $\{p \mid p \in P\}$ **do**
        $i_{parent} := \mathit{createIndividual}(v.type)$
        for all $\{i_{child} \in p.I\}$ **do**
          $\mathit{createObjectPropertyAssertion}(e.type, i_{parent}, i_{child})$

Abbreviations: I set of individuals i, $I[v_j]$ map of individuals I per v_j, $P = (I)$ set of permutations object property assertion like $hasMotor$ ($e.type$) from the motor-gearbox combination (i_{parent}) to a motor (i_{child}). The algorithm terminates when all nodes V are traversed. Afterwards, the satisfied individuals of the component v_n can be inferred along with the $composedOf$ subcomponents and the calculated properties, which can be partially mapped to the requirements. Before presenting the solutions to the user, the generated solutions are rated (Section III-C). This determines the order in which the solutions are presented. In a further reasoning step, the inferred components could be used as subcomponents in a higher-level component.

C. Rating functions

The user is able to influence the rating of the design solutions by specifying an allowed deviation and a weight w_i for a chosen requirement $u_{Req,i} \in U = U_{min} \cup U_{max}$, which can either be $u_{Req,i,min} \in U_{min}$ or $u_{Req,i,max} \in U_{max}$. We define a positive and negative deviation. The deviation is negative, if $u_{Sol,i} < u_{Req,i,min}$ or $u_{Req,i,max} < u_{Sol,i}$, with $u_{Sol,i}$ the solution value for requirement $u_{Req,i}$. The positive deviation is defined inverse and represents a $u_{Sol,i}$ which is better than required. The relative error e_i is:

$$e_i = \begin{cases} (u_{Req,i})^{-1} \, (u_{Req,i} - u_{Sol,i}) & , u_{Req,i} \in U_{min} \\ (u_{Req,i})^{-1} \, (u_{Sol,i} - u_{Req,i}) & , u_{Req,i} \in U_{max} \end{cases} \quad (3)$$

As error function for the negative deviation we use a weighted variation of the normalized root-mean-square deviation ($NRMSD$). The error e_i is normalized between 0 and 1 to compare requirements with different ranges.

$$NRMSD = \sqrt{\left(\sum_{i=1}^{\#U} w_i\right)^{-1} \sum_{i=1}^{\#U} w_i \ \max(0, e_i)^2} \quad (4)$$

Beside the $NRMSD$ we calculate a performance index (PX) which takes also the positive deviation into account to represent an overall performance of a solution:

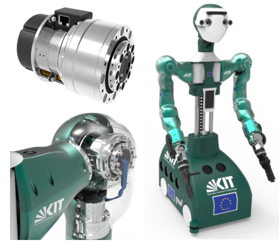
$$PX = \left(\sum_{i=1}^{\#U} w_i\right)^{-1} \sum_{i=1}^{\#U} w_i \ (1 - e_i) \quad (5)$$

IV. Evaluation

To evaluate the framework, we choose sensor-actuator units (SA units) as a case study, which are mid-level components in the hierarchy of humanoid robot components (Figure 4). Situated in humanoid segments like arms or legs, they confront robot engineers with typical challenges in humanoid robot design, in particular a high functional integration in a limited construction space. Therefore, we present an expert rule set that focuses on the choice and arrangement of the SA unit's subcomponents. Finally, we demonstrate the expert system's capabilities to reproduce and optimize existing SA units.

A. Case Study: Sensor-Actuator Units (SA Units)

SA units usually include several subcomponents such as the drive train, sensors and other mechatronic parts. Some designs like the KIT sensor-actuator-controller units (SAC units) [2], also include control and communication electronics. To achieve a high integration degree (Figure 5), the construction space, interfaces and other physical requirements of each subcomponent have to be taken into account. Many humanoid robots use the combination of a frameless, brushless DC (BLDC) motor and a Harmonic Drive reduction gearbox, enabling a compact high-torque design with a through bore for cabling. Besides the motor-gearbox combination, the arrangement of the subcomponents has a great influence on the construction space. Therefore, we identified different structural options which have different advantages and result in different dimensions (Figure 6). For example, the drive bearings can be placed under (D2) instead of beside the motor (D1), resulting in a shorter length, but a smaller through bore for cabling. Furthermore, SA units differ in their subcomponent setting, which is used for further classification (Table II). In the following, we use the presented abbreviations: The KIT SAC unit M (Figure 5) is described by "D1 O1 M1 C1" and "MF1 AE1 TS1 IM1 BR0".



Level (Lvl)	Example	
System	Humanoid Robot	
Comp. Lvl 1	Manipulation System	
Comp. Lvl 2	Robot Arm	
Comp. Lvl 3	Shoulder	
Comp. Lvl 4	**SA Unit**	Case Study
Comp. Lvl 5	Drive Train	Case Study
Comp. Lvl N	Catalog Motor	Case Study

Fig. 4: Modular SA units (left) for humanoid robots (middle) and hierarchical model of humanoid robot components (right)

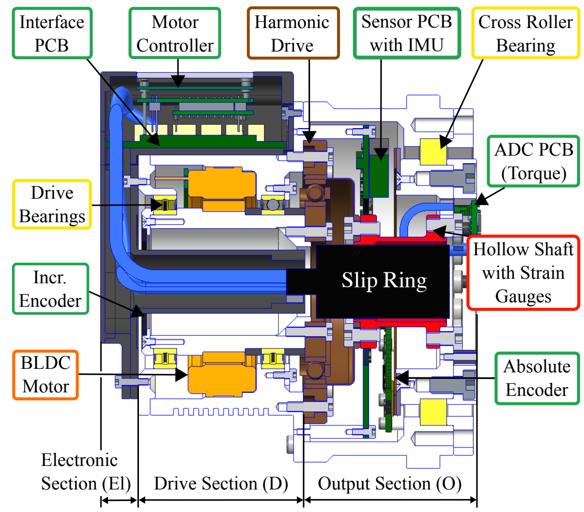


Fig. 5: Labeled cross section of a KIT SAC unit (Size M)

B. Design Rules for SA Units

The expert system uses a set of engineering design rules, which are based on the experience we gained during the design phase of the KIT SAC units. Taking requirements as dimensions, performance data and necessary functions as input, the rules describe the calculation of the resulting characteristics as well as the choice of possible subcompo-

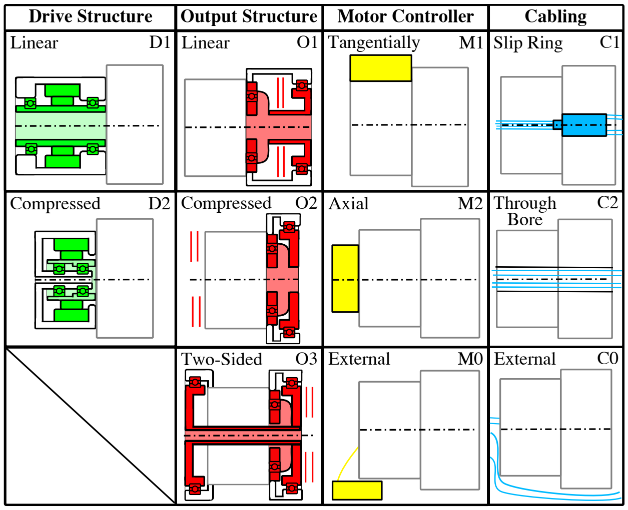


Fig. 6: Classification for structural options of SA units

TABLE II: Classification for subcomponent setup of SA units

	Motor Feedback (MF)	**Absolute Encoder (AE)**	**Torque Sensor (TS)**	**IMU (IM)**	**Brake (BR)**
0	None	None	None	None	None
1	Incremental Encoder	One	Strain-Gauge-Based	Yes	Yes
2	Hall Sensor	Two	Encoder-Based		
3	Both		Both		

nents. As we identified different structural options for SA units (Figure 6), the rules distinguish between these options.

1) Motor-Gearbox Matching: In the first place, the goal of this step is to find motor-gearbox combinations, which fulfill the performance requirements, i. e. the maximum speed $n_{Req,max}$ and the peak torque $T_{Req,p}$. At first, both devices, the $Motors(M)$ and the $Gearboxes(G)$, are filtered separately and assigned to their satisfied counterpart $SatisfiedM$ and $SatisfiedG$ (Figure 7). Therefore, the mechanical limits of the gearbox, i.e. the maximum output speed $n_{G,max}$ and the peak torque $T_{G,p}$, have to satisfy the requirements:

$$n_{Req,max} \leq n_{G,max} \tag{6}$$
$$T_{Req,p} \leq T_{G,p} \tag{7}$$

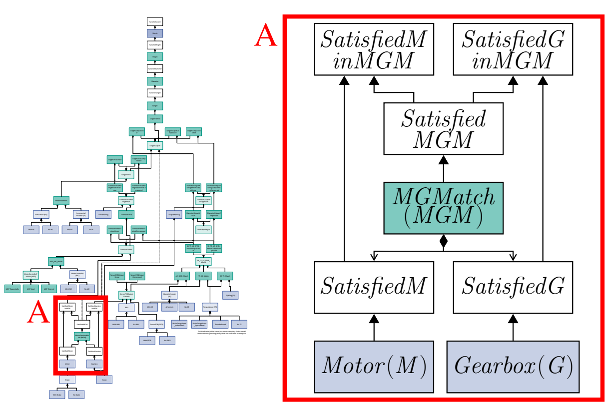


Fig. 7: Motor-gearbox matching (A) in the multi-stage reasoning

The filtering of the motors with a maximum speed $n_{M,max}$ and a peak torque $T_{M,p}$ can be described as follows:

$$n_{Req,max} \leq \frac{n_{M,max}}{i} = n_{Mi,max} \tag{8}$$
$$T_{Req,p} \leq T_{M,p}\ \eta_G\ \eta_{other}\ i = T_{Mi,p} \tag{9}$$

The motor filter considers all possible gearbox reductions i at an efficiency of the gearbox $\eta_G = 1$. For this purpose, the output speed $n_{Mi,max}$ and the peak torque $T_{Mi,p}$ of the motor at gear ratio i are calculated.

After filtering both devices separately, the expert system checks each $MotorGearboxMatch(MGM)$:

$$n_{Req,max} \leq n_{Sol,max} = \min(n_{G,max}, n_{Mi,max}) \tag{10}$$
$$T_{Req,p} \leq T_{Sol,p} = \min(T_{G,p}, T_{Mi,p}) \tag{11}$$

In contrast to the motor filter, the efficiency $\eta_G < 1$ and ratio i of the respective gearbox are used. The result of the motor-gearbox matching is a set of *SatisfiedMGMs* with performance parameters ($n_{Sol,max}, T_{Sol,p}$), which fulfill or surpass the requirements ($n_{Req,max}, T_{Req,p}$).

All described formulas are modeled as a SWRL rule set in the reasoning ontology. The SWRL example rules in Table III correspond to the calculation of $n_{Mi,max}$ (Equation 8) and the requirement $n_{Req,max} \leq n_{Sol,max}$ (Equation 10).

2) Choice of Other Subcomponents: Similar to the described motor-gearbox matching, the remaining subcomponents as sensors and bearings are chosen in different nodes of the reasoning tree, based on rules described by mathematical expressions. Based on the requirements, for each motor-gearbox matching and each structural option a subcomponent setup is determined. For example, for each motor-gearbox matching the system chooses two different drive bearing setups as two drive structure options are implemented (D1, D2). Besides construction space and performance requirements, electrical requirements as the cabling of sensors are considered. For example, slip rings, mechatronic devices, which allow continuous cable rotation, are chosen based on electrical requirements of the rotating sensors and following SA units, i. e. the necessary power supply and data lines for a bus or emergency stop. However, as a result of the slip ring choice, the gearboxes and structural options are filtered based on the necessary construction space.

TABLE III: Examples of formulas modeled as SWRL rules

Formula	**SWRL Rule**
$n_{Mi,max} = \frac{n_{M,max}}{i}$	$MGM(?mgm)$ˆ$hasM(?mgm,\ ?m)$ˆ $hasG(?mgm,\ ?g)$ˆ$has_n_M_max(?m,$ $?n_M_max)$ˆ$has_i(?g,\ ?i)$ˆ$swrlb:$ $divide(?n\_Mi\_max,\ ?n\_M\_max,\ ?i)$ $\rightarrow has\_n\_Mi\_max(?mgm, ?n\_Mi\_max)$
$n_{Req,max} \leq n_{Sol,max}$	$MGM(?mgm)$ˆ$Requirement(?req)$ˆ $has_n_Sol_max(?mgm,\ ?n_Sol_max)$ˆ $has_n_Req_max(?req,\ ?n_Req_max)$ˆ $swrlb:lessThanOrEqual($ $?n\_Req\_max,\ ?n\_Sol\_max)$ $\rightarrow Satisfied\_n\_max(?mgm)$

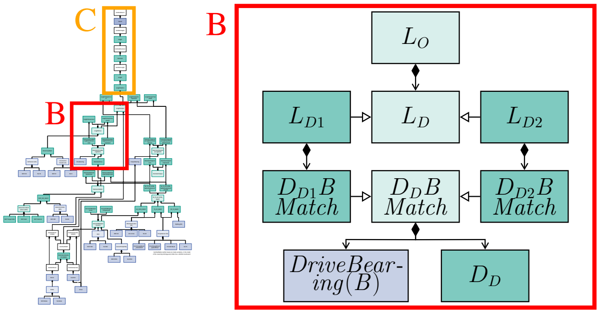


Fig. 8: Calculation of L_D in the multi-stage reasoning tree (B) and the final dimensions L, D, H of the SA unit (C)

3) Determination of Physical Dimensions: Based on the motor-gearbox matches and subcomponent choice, for each structural option the dimensions of the three cylindrical sections of the SA unit (Figure 5) can be determined: drive (D_D, L_D), output (D_O, L_O) and electronics (D_{El}, L_{El}). Each section dimension is modeled as an $AbstractNode$ with $StageNode$ representing the actual options (Section III-B).

$$L_{D1} = L_{MHB} + L_{LB} + L_{FB} + \epsilon_1 \quad (12)$$

$$L_{D2} = \max(L_{MHB} + \epsilon_2,\ L_{LB} + L_{FB} + \epsilon_3) \quad (13)$$

Equations 12 and 13 present the calculation of the drive section length L_D for both drive structure options simplified for a better understanding by summarizing comparatively short lengths with $\epsilon_1, \epsilon_2, \epsilon_3$. As L_D mostly depends on the lengths of the motor-encoder-brake unit (L_{MHB}) and the drive bearings $(L_{LB},\ L_{FB})$, the parallel arrangement of D2 results in a shorter value, which is represented by the maximum. The rules based on the equations are evaluated in the correspondent nodes (Figure 8). L_D is based on $D_{D1}BMatch$, in which the best fitting $DriveBearing(B)$ is chosen. The $AbstractNodes$ (e. g. L_D) are not represented by an individual in the ontology, but rather their structural options (e. g. L_{D1} and L_{D2}).

Analogous to L_D, the remaining diameters and lengths of the three sections are determined for all structural options. Thereafter, the system determines the total length L and maximum diameter D of each possible SA unit combination:

$$L = L_D + L_O + L_{El} \quad (14)$$

$$D = \max(D_D, D_O, D_{El}) \quad (15)$$

Finally, the height H is calculated to describe non-cylindrical designs, e.g. with tangentially placed motor controller (M1).

C. Scope of the System

Table IV lists all catalog subcomponents which are implemented in the expert system's ontological knowledge base in its current state. Subcomponents with currently only one option (e. g. IMU) are suitable for the whole torque capacity range of SA units from 1.8 to 823 Nm. During the reasoning, the system is able to consider all possible combinations of subcomponents. It is possible to combine every motor with every gearbox. However, to avoid a combinatorial explosion during the search, we apply effective pruning. For example, only a single drive bearing type is chosen to match a given combination of a motor and drive structure option.

TABLE IV: Subcomponent list

Subcomponent	**No.**	**Subcomponent**	**No.**
BLDC Motor (opt. hall sensor)	14 (x2)	Gearbox (Harmonic Drive)	67
Brake	4	Slip Ring	20
Drive Bearing	23	Output Bearing	23
Incremental Encoder	1	Absolute Encoder	3
Motor Controller	1	IMU	1

D. Reproduction and Optimization of SA Units

The evaluation of the system is conducted in three steps:

1. Verification: Reproduction of the KIT SAC units which can be considered as training data
2. Generalization: Reproduction of state-of-the-art SA units for humanoid robotics
3. Optimization: Building optimized KIT SAC units

For each SA unit, the requirements (1st row) and the best solution (2nd row) suggested by the expert system are presented (Table V). Structural and subcomponent options are fixed requirements, which have to be fulfilled by the solutions unless they are unknown ("∗"). In contrast, dimensional and performance requirements are treated more flexibly and used to optimize the solution. The first criterion for rating is the $NRMSD$, which takes negative deviations from the required dimensions and performance into account. Ideally, the solution's dimensions (L, D, H) have to be fulfilled or fall below the requirements whereas the performance parameters (T_p, n_{max}) have to be fulfilled or surpassed. Fulfilling and positive deviations are rated same (0). A second criterion, the performance index (PX), is used to rank solutions with the same $NRMSD$ (Section III).

1) Reproduction of the KIT SA/SAC Units: To verify the correct implementation of the expert system, we reproduced the KIT SAC units and the SA unit of the humanoid ARMAR-4. They can be considered as training data as the rule set is based on the experience acquired during their design and their subcomponents are part of the ontology. The proposed design solutions (Table V) correspond to the realized SA units with regard to subcomponents, performance and, for the most part, dimensions. As we found optimization possibilities, some dimensions are even smaller. This is reflected by the rating $(NRMSD \approx 0,\ PX \geq 1)$.

2) Reproduction of State-of-the-Art SA Units: Besides the KIT units, we reproduced 11 state-of-the-art SA units for humanoid robots with different subcomponents and structures of which most specifications are known (Table V). The results show that 10 out of 11 solutions are very close to the real SA units $(NRMSD < 0.1)$. The only negative exception is the reproduction of WALK-MAN C $(NRMSD = 0.247)$, as the dimensions of the solution are approximately 30% larger than the requirements. Whereas the required diameter could be realized with a new structural option, the greater length might be a consequence of wrong assumptions about the requirements (e. g. the motor controller of the real unit could be placed differently). The solutions for the NREC, ETH and other two WALK-MAN units show comparatively small deviations, which can be explained by the use of

TABLE V: Evaluation of the expert system through reproduction and optimization of state-of-the-art SA units for humanoid robotics

SA unit			Structural and subcomponent options		Dimensions and performance					Rating	
Source	**Size**		**Structure**	**Subcomponent setup**	L [mm]	D [mm]	H [mm]	T_p [Nm]	n_{max} [°/s]	**NRMSD**	**PX**
KIT SAC Units [2]	L	Req	D1 O1 M1 C1	MF1 AE1 TS1 IM1 BR0	159.0	112.0	118.0	176	78.75		
		Sol	D1 O1 M1 C1	MF1 AE1 TS1 IM1 BR0	153.4	106.0	114.5	176	78.75	0.000	(1.02)
	M	Req	D1 O1 M1 C1	MF1 AE1 TS1 IM1 BR0	113.0	112.0	118.0	123	131.25		
		Sol	D1 O1 M1 C1	MF1 AE1 TS1 IM1 BR0	113.7	103.0	113.0	123	131.25	0.003	(1.02)
	S	Req	D1 O1 M2 C1	MF1 AE1 TS1 IM1 BR0	117.0	85.0	85.0	56	206.25		
		Sol	D1 O1 M2 C1	MF1 AE1 TS1 IM1 BR0	110.5	84.0	84.0	56	206.25	0.000	(1.02)
KIT SA Units ARMAR-4 [1]	Leg	Req	D2 O2 M0 C2	MF1 AE1 TS1 IM0 BR0	84.0	112.0	112.0	157	174.00		
		Sol	D2 O2 M0 C2	MF1 AE1 TS1 IM0 BR0	82.6	113.0	113.0	157	174.00	0.006	(1.00)
IIT WALK-MAN † [25]	A	Req	D∗ O3 M2 C2	MF2 AE2 TS2 IM0 BR0	150.0	110.0	110.0	270	84.00		
		Sol	D2 O3 M2 C2	MF2 AE2 TS2 IM0 BR0	146.6	113.0	113.0	299	**108.75**	0.017	(1.07)
	B	Req	D∗ O3 M2 C2	MF2 AE2 TS2 IM0 BR0	140.0	100.0	100.0	140	100.00		
		Sol	D2 O3 M2 C2	MF2 AE2 TS2 IM0 BR0	135.6	103.0	103.0	147	**131.25**	0.019	(1.07)
	C	Req	D∗ O3 M2 C2	MF2 AE2 TS2 IM0 BR0	100.0	60.0	60.0	56	67.80		
		Sol	D2 O3 M2 C2	MF2 AE2 TS2 IM0 BR0	**128.6**	**80.0**	**80.0**	54	**210.00**	**0.247**	(**1.22**)
NREC Drive Joint † [26]	NGT 200	Req	D∗ O3 M0 C1	MF1 AE2 TS2 IM0 BR1	135.0	140.0	140.0	660	65.40		
		Sol	D2 O3 M0 C1	MF1 AE2 TS2 IM0 BR1	156.7	133.0	133.0	630	65.00	0.075	(0.98)
	NGT 100	Req	D∗ O3 M0 C1	MF1 AE2 TS2 IM0 BR1	130.5	111.5	111.5	360	88.80		
		Sol	D2 O3 M0 C1	MF1 AE2 TS2 IM0 BR1	137.3	113.0	113.0	304	85.00	0.076	(0.94)
	NGT 50	Req	D∗ O3 M0 C1	MF1 AE2 TS2 IM0 BR1	113.5	94.5	94.5	175	162.60		
		Sol	D2 O3 M0 C1	MF1 AE2 TS2 IM0 BR1	120.9	103.0	103.0	147	174.00	0.096	(0.93)
	NGT 20	Req	D∗ O3 M0 C1	MF1 AE2 TS2 IM0 BR1	90.5	77.0	77.0	50	182.40		
		Sol	D2 O3 M0 C1	MF1 AE2 TS2 IM0 BR1	107.5	80.0	80.0	54	210.00	0.088	(0.99)
ETH ANYdrive [27]	-	Req	D∗ O∗ M1 C2	MF∗ AE∗ TS∗ IM0 BR0	95.0	90.0	120.0	40	684.00		
		Sol	D2 O2 M1 C2	MF1 AE1 TS1 IM0 BR0	89.2	93.0	116.0	41	547.50	0.091	(0.98)
RoboDrive RD-HD [28]	50 x08	Req	D∗ O∗ M0 C2	MF∗ AE1 TS0 IM0 BR0	77.6	97.0	97.0	28	330.00		
		Sol	D2 O2 M0 C2	MF2 AE1 TS0 IM0 BR0	68.1	93.0	93.0	**52**	**438.00**	0.000	(**1.33**)
	70 x10	Req	D∗ O∗ M0 C2	MF∗ AE1 TS0 IM0 BR0	99.8	128.0	128.0	92	132.00		
		Sol	D2 O2 M0 C2	MF2 AE1 TS0 IM0 BR0	82.6	113.0	113.0	96	**348.00**	0.000	(**1.49**)
	85 x13	Req	D∗ O∗ M0 C2	MF∗ AE1 TS0 IM0 BR0	111.3	141.0	141.0	176	108.00		
		Sol	D2 O2 M0 C2	MF2 AE1 TS0 IM0 BR0	**72.5**	133.0	133.0	183	**174.00**	0.000	(**1.26**)
KIT SAC Units Optimization	L	Req	D∗ O∗ M∗ C1	MF∗ AE1 TS1 IM1 BR0	159.0	112.0	118.0	176	78.75	± 10%	
	L+	Sol	D2 O1 M2 C1	MF1 AE1 TS1 IM1 BR0	149.1	106.0	106.0	167	**145.00**	(0.023)	**1.20**
	M	Req	D∗ O∗ M∗ C1	MF∗ AE1 TS1 IM1 BR0	113.0	112.0	118.0	123	131.25	± 10%	
	M+	Sol	D2 O2 M1 C1	MF1 AE1 TS1 IM1 BR0	99.1	113.0	126.0	137	**217.50**	(0.031)	1.16
	S	Req	D∗ O∗ M∗ C1	MF∗ AE1 TS1 IM1 BR0	117.0	85.0	85.0	56	206.25	± 10%	
	S+	Sol	D2 O1 M2 C1	MF2 AE1 TS1 IM1 BR0	119.9	91.0	91.0	52	**390.00**	(0.058)	1.13

Abbreviations: Req = Requirements, Sol = Best solution, "∗"= Any option except from "none"; see Section IV-A for abbreviations
Rating ($NRMSD$, PX): L, D, H, T_p, n_{max} are weighted equally ($w_i = 1$); **Positive deviations $\geq 20\%$ green**; **Negative deviations $\geq 20\%$ red**

subcomponents from other manufacturers. For example, the NREC Drive Joints use smaller brakes, which results in a reduced total length. The solutions for the RoboDrive units surpass the requirements ($NRMSD = 0$, $PX > 1$). The system uses the construction space to integrate motor-gearbox combinations with higher performance.

3) Optimization of the KIT SAC Units: The possibility for optimization of existing SA units is shown using the example of the KIT SAC units. In contrast to the reproduction, we use less constraints ("∗") and a maximum deviation of 10% from the requirements, which results in more solutions (L:89/M:84/S:54). For each solution the UI visualizes all catalog subcomponents besides the rating, structural options and calculated properties. This time, the solution with the highest PX is considered best. The optimized solution L+ for SAC unit L shows the value of the system as optimization tool: By the use of a different motor-gearbox combination and other structural options, L+ nearly doubles its speed with reduced dimensions at the cost of $9\,\mathrm{Nm}$ less peak torque.

E. Discussion

The evaluation of the SA unit expert system showed the possibilities of our approach regarding the reproduction and optimization of robotic components. Thereby, we demonstrated the system's capabilities to work with looser constraints by generating all possibilities and choosing the best solution based on the given requirements. One limitation of the expert system is that it can only reason within the given knowledge base. Therefore, only solutions can be found that can be derived from the stored knowledge. However, the generated solutions have (1) a high probability of being physically realizable. They are based on existing catalog components and tested designs, which were analyzed and generalized in detail. Furthermore, (2) the expert system only needs a few minutes to find design solutions for SA units, including all necessary catalog components and their arrangement. By comparison, engineers usually spend hours or even days doing the same job. The expert system can also be used to (3) quickly check the impact of optional subcomponents such as slip rings or new catalog components on the total design. At the moment we are taking advantage of (4) the easy expandability of the knowledge base to add new catalog motors as well as costs and weight as new requirements. The current system already considers thermal conduction in the selection and arrangement of motors, but we are planning to consider different thermal installation conditions. In order to expand the proposed framework to higher-level components such a robot arms, we are planning to combine our presented bottom-up approach with more top-down elements that translate user requirements into subcomponent constraints.

V. Conclusion and Future Work

This work introduced an expert system framework, which significantly simplifies the design process of humanoid robot components. Based on user requirements, the expert system is able to find the best solutions within a large solution space consisting of potential concept solutions and existing catalog components. The complexity in the development of humanoid robot components, arising in particular from the integration of multiple mechatronic subcomponents, is handled by multi-stage reasoning on an ontological knowledge base. This novel approach executes rules in a bottom-up graph search by generating, combining and discarding partial solutions until complete solutions are found. The expert system is made accessible via a user interface that allows easy definition and weighting of requirements. By using this automated approach, the time to evaluate different design solutions in the large solution space can be significantly reduced. Thereby, it can be used by experts and novices alike for finding adequate solutions during the design of humanoid robot components. To evaluate this framework, we implemented a detailed expert system for sensor-actuator units (SA units) for humanoid robots. The system was able to accurately reproduce 10 of 11 tested state-of-the-art SA units as well as our 4 KIT SA/SAC units. Furthermore, it was able to optimize our existing KIT SAC units. In future work, we want to extend our approach to be applicable to higher-level components and complete humanoid robots.

References

[1] T. Asfour, J. Schill, H. Peters, C. Klas, J. Bücker, C. Sander, S. Schulz, A. Kargov, T. Werner, and V. Bartenbach, "ARMAR-4: A 63 DOF Torque Controlled Humanoid Robot," in *IEEE/RAS Int. Conf. Humanoid Robots (Humanoids)*, 2013, pp. 390–396.

[2] S. Rader, L. Kaul, P. Weiner, and T. Asfour, "Highly integrated sensor-actuator-controller units for modular robot design," in *IEEE Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, 2017, pp. 1160–1166.

[3] W. P. Wagner, "Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies," *Expert systems with applications*, vol. 76, pp. 85–96, 2017.

[4] V. P. Agrawal, S. Gupta, and V. Kohli, "Computer aided robot selection: The 'multiple attribute decision making' approach," *Int. Journal of Production Research*, vol. 29, no. 8, pp. 1629–1644, 1991.

[5] N. Boubekri, M. Sahoui, and C. Lakrib, "Development of an expert system for industrial robot selection," *Computers & Industrial Engineering*, vol. 20, no. 1, pp. 119–127, 1991.

[6] D. T. Pham and E. Tacgin, "An expert system for selection of robot grippers," *Expert systems with applications*, vol. 5, no. 3-4, pp. 289–300, 1992.

[7] P. Bhatia, J. Thirunarayanan, and N. Dave, "An expert system-based design of scara robot," *Expert systems with applications*, vol. 15, no. 1, pp. 99–109, 1998.

[8] W. Li, Y. Li, J. Wang, and X. Liu, "The process model to aid innovation of products conceptual design," *Expert systems with applications*, vol. 37, no. 5, pp. 3574–3587, 2010.

[9] 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, pp. 931–946, 2012.

[10] S.-J. Chiou and K. Sridhar, "Automated conceptual design of mechanisms," *Mechanism and machine theory*, vol. 34, no. 3, pp. 467–495, 1999.

[11] D. R. Frutiger, J. C. Bongard, and F. Iida, "Iterative product engineering: Evolutionary robot design," in *Proceedings of the fifth international conference on climbing and walking robots*. Professional Engineering Publishing, 2002, pp. 619–629.

[12] S. Myung and S. Han, "Knowledge-based parametric design of mechanical products based on configuration design method," *Expert systems with applications*, vol. 21, no. 2, pp. 99–107, 2001.

[13] R. Desai, Y. Yuan, and S. Coros, "Computational abstractions for interactive design of robotic devices," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1196–1203, 2017.

[14] S. Ha, S. Coros, A. Alspach, J. M. Bern, J. Kim, and K. Yamane, "Computational design of robotic devices from high-level motion specifications," *IEEE Transactions on Robotics*, no. 99, pp. 1–12, 2018.

[15] J. Laugis and V. Vodovozov, "Expert system for electric drive design," in *IEEE Int. Power Electronics and Motion Control Conf.*, 2008, pp. 1017–1019.

[16] Q. Wang, X. Chen, Y. Yin, and J. Lu, "Ontology-based coupled optimization design method using state-space analysis for the spindle box system of large ultra-precision optical grinding machine," in *IEEE 2nd Int. Conf. on Enterprise Systems*, 2014, pp. 273–278.

[17] A. Šaša Bastinos and M. Krisper, "Multi-criteria decision making in ontologies," *Information Sciences*, vol. 222, pp. 593–610, 2013.

[18] F. Ramos, A. S. Vázquez, R. Fernández, A. Olivares-Alarcos, C. Schlenoff, S. Balakirsky, and H. Christensen, "Ontology based design, control and programming of modular robots," *Integrated Computer-Aided Engineering*, vol. 25, no. 2, pp. 173–192, 2018.

[19] A. Juarez, C. Bartneck, and L. Feijs, "Using semantic technologies to describe robotic embodiments," in *Proceedings of the 6th Int. Conf. on Human-robot interaction*, 2011, pp. 425–432.

[20] X. Liu, D. Rosen, and Z. Yu, "Ontology based knowledge modeling and reuse approach in product redesign," in *IEEE Int. Conf. Information Reuse and Integration (IRI)*, 2010, pp. 270–273.

[21] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Web Semantics*, vol. 5, no. 2, pp. 51–53, 2007.

[22] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, "Owl 2: The next step for owl," *Web Semantics*, vol. 6, no. 4, pp. 309–322, 2008.

[23] S. R. Fiorini, J. L. Carbonera, P. Gonçalves, V. A. Jorge, V. F. Rey, T. Haidegger, M. Abel, S. A. Redfield, S. Balakirsky, V. Ragavan, H. Li, C. Schlenoff, and E. Prestes, "Extensions to the core ontology for robotics and automation," *Robotics and Computer-Integrated Manufacturing*, vol. 33, pp. 3–11, 2015.

[24] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean *et al.*, "Swrl: A semantic web rule language combining owl and ruleml," *W3C Member submission*, vol. 21, p. 79, 2004.

[25] F. Negrello, M. Garabini, M. G. Catalano, J. Malzahn, D. G. Caldwell, A. Bicchi, and N. G. Tsagarakis, "A Modular Compliant Actuator for Emerging High Performance and Fall-Resilient Humanoids," in *IEEE/RAS Int. Conf. Humanoid Robots (Humanoids)*, 2015, pp. 414–420.

[26] A. Stentz *et al.*, "Chimp, the cmu highly intelligent mobile platform," *Journal of Field Robotics*, vol. 32, no. 2, pp. 209–228, 2015.

[27] "ANYbotics". [Online]. Available: https://www.anybotics.com

[28] "TQ-Systems GmbH". [Online]. Available: http://www.robodrive.com