EVALUATION OF AN AUTOMATED DESIGN AND OPTIMIZATION FRAMEWORK FOR MODULAR ROBOTS USING A PHYSICAL PROTOTYPE

Vaheed Nezhadali[1], Omer Khaleeq Kayani[1], Hannan Razzaq[1] and Mehdi Tarkian[1]
(1) Linköping University, Sweden

ABSTRACT
This paper presents an automated design and evaluation framework, by integrating design tools from various engineering domains for rapid evaluation of design alternatives. The presented framework enables engineers to perform simulation based optimizations. As a proof of concept a seven degree of freedom modular robot is designed and optimized using the automated framework. The designed robot is then manufactured to evaluate the framework using preliminary tests.

Keywords: Automated design, simulation-based optimization, multidisciplinary design, modular robot, CAD automation

1 INTRODUCTION
The first generation of industrial robots was introduced in 1950s, and at that time it was expected that in a near future they will be widely used. However, due to long development times, high initial costs and complexity involved in the design process, use of robots has been mainly limited to specialized industrial task. Conventionally robot design is application-specific, implying that a given robot cannot be modified for diverse applications, reducing its reusability and forcing manufactures to incur high initial costs for new applications.
Evidently, a new design methodology is required to overcome the mentioned problems faced during robot design process. A modular design approach addresses the shortcomings of conventional design method by sharing and reusing modules, leading to a higher customization level.
Modular robots, like conventional robots, typically consist of subsystems which belong to multiple engineering domains, such as mechanics, electronics, and control. These subsystems are designed and evaluated using various engineering tools. Design modification of a component belonging to a certain domain will consequently require an update in other engineering domains. Propagation of such modifications across the engineering tools is generally done manually, which is a time-consuming process. To propagate these modifications in a time efficient manner, an automated interaction between these engineering tools is preferred. An automated design framework integrates the engineering tools and eliminates the need for manual modifications. Moreover, in contrast to a manual design evaluation process, more design iterations can be performed for a given development time. As a larger design space can be explored, the likelihood of finding an improved design increases. Additionally, an automated design framework would lead to reduction in time to market of the product.

1.1 Related Work
A robot design process mainly involves geometry generation, kinematic analysis and preferably optimization. The benefits of generating flexible and robust geometries for automated design have been demonstrated in various research groups and disciplines. The aircraft research domain has made efforts to describe methods for the automatic generation of geometries. This has been effectively demonstrated by Jouannet et al. [1], for micro-UAV design, Tarkian et al. [2] for civil aircraft design and La Rocca et al. [3] in the analysis for specific aircraft feature. The advantages of automated geometric modeling have also been illustrated for industrial robots, Tarkian et al. [4], as well as modeling airfoil shapes for use in wind turbine design, as presented by Cooper et al. [5]. Tarkian et al. in [6] discuss the dynamic models for industrial robots. Dynamic models for aircraft design analysis have also been effectively utilized by Johanson et al. [7]. Ölvander et al. have presented an optimization framework for aircraft design in [8], and a study of modular robots drive train

optimization is carried out by Pettersson and Ölvander [9]. In [10] an optimization procedure for the selection of actuators and mechatronic components is presented.

An automated design framework for industrial robots has been presented in [4] and [6]; however, no validation has been performed for this framework. These works also lack an automatic selection of optimized actuators which is an important part of the robot design process.

In this paper a new automated design and evaluation framework for modular robots is presented. The proposed framework enables the designer to create and evaluate different concepts rapidly, and facilitates the implementation of a simulation-based optimization for the selection of actuators. The presented optimization procedure, however, can also be extended to other robot design parameters. Additionally, a physical prototype, designed using this framework, is manufactured to ensure the reliability of the presented framework.

1.2 Modular Design

Modular design is an approach that subdivides a system into smaller parts (modules) which can be independently created and then used in different systems to drive multiple functionalities [4]. Modular robots refer to the idea of a family of robots which share modules within the family. Figure 1 shows an example of a modular robot family sharing modules among each other. The concept of modular robots has been discussed intensely since late 80s, e.g. Krenn et al. [11] and Paredis et al. [12]. Recently these robots are becoming increasingly attractive for robot manufacturers, such as Motoman [13] and Nachi [14]. Studies regarding some aspects involved in the design process of modular industrial robots have been carried out at Linköping University; Petterson et al. [15] and Safavi et al. [16].

The paper is divided into four sections. After this first introduction, the second section presents different components of the automated design framework in detail. In the third section, a design example is presented in which the automated design framework is utilized to design a seven degree of freedom (DOF) modular robot. An evaluation of the framework is also presented in this section. Finally section four comprises the conclusions and way forward.

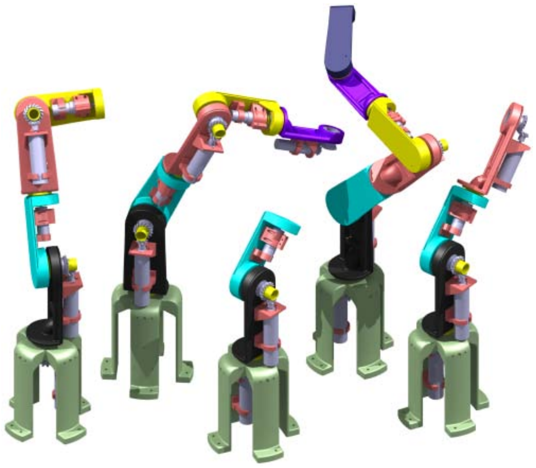

Figure 1. Example of a modular robot family

2 AUTOMATED DESIGN AND EVALUATION FRAMEWORK

The automated design and evaluation framework is implemented by integrating design tools as shown in Figure 2. In the framework, Microsoft Excel serves as common platform for data exchange between various tools. It also acts as an interface through which users can interact with the framework. The framework can be broken down into an automated CAD framework, automated dynamic simulation framework and an optimization routine.

2.1 Automated CAD Framework

For simulation, evaluation and verification of the properties of any given product, a geometric CAD model is preferred [17]. In the beginning of the geometric modeling phase, simplified geometries are used which can result in inaccurate representation of the product. However, in order to define a sufficiently accurate geometric CAD model, frequent re-modeling has to be performed to introduce more details in the model.

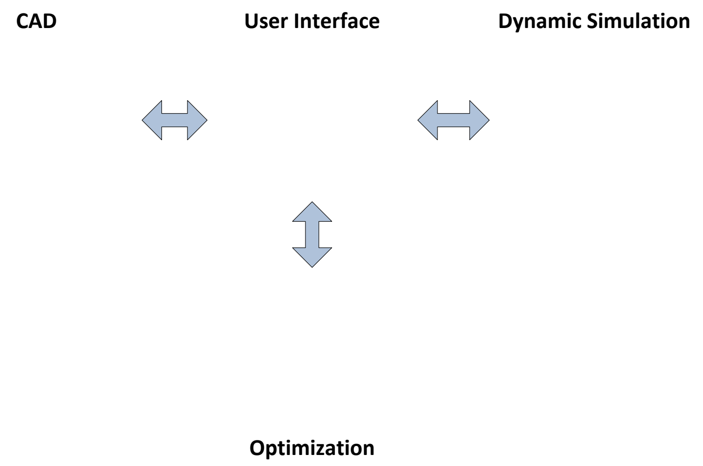

Figure 2. Design tool integration

Updating and generating geometric CAD models is a time demanding process. An automated CAD framework (ACF) allows for the rapid generation of geometric CAD model, hence, reducing the time to reach a sufficiently accurate geometric representation of the product.

Commercial CAD tools available in the market are becoming increasingly suitable to generate automated geometries for multi-disciplinary optimization (MDO) and design. CAD tools such as CATIA, Solid Works, Pro Engineer and NX6 all offer parametric design with varying functionalities. Geometric models created using these CAD tools can be very flexible and robust in the sense that both shape and number of geometric objects of the model can be parametrically defined. This parameterization is accomplished by defining templates and context manuals. These manuals contain complete construction procedures of the template objects. The template objects can be parametrically modified to obtain different sizes and shapes, and these construction procedures enable the template objects to be instantiated into different contexts, consequently, increasing reusability of created geometries in the CAD model. The geometric complexity of the CAD model dictates the accuracy of the model, whereas, degree of parameterization defines its flexibility. In general, the geometry should be flexible enough to allow generation of all conceivable configurations or concepts of interest. However, to define a generic geometry a large number of parameters have to be introduced, so a compromise has to be made between the two.

2.1.1 Automated generation of geometric CAD model for modular robots

The mechanical structure of a modular robot consists of a base followed by a series of modules. Each module mainly consists of a link, actuator, shaft and bearings. The choices made while designing these modules to make them modular are not trivial. Considering the fact that in the beginning, limited knowledge is at hand about the properties of the finished product, some design parameters should be kept flexible to be easily modified during the entire design phase. Hence, the possibility to remedy the shortcomings of certain choices made early in the design cycle is vital. For example, the choice of actuator dictates the shape of the module and in particular the link length. Changing the actuator type can result in re-modeling of entire geometry of the robot. Such changes can be verified rapidly using an automated CAD framework. Moreover, properties like mass, centre of gravity and moment of inertia; required for dynamic simulation and optimization can also be obtained.

The CAD tool selected for automated geometry generation of the modular robot is CATIA V5 which offers great flexibility for automated CAD generation. The parameters for changing the shape and number of different geometric parts in the robot CAD model are controlled from a user interface.

The main parameter which defines the number of instances of the modules in the robot CAD model is the DOF. The CAD model hierarchy in CATIA consists of a main assembly containing sub-assemblies. The first sub-assembly represents the base, and the remaining sub-assemblies, depending on DOF, correspond to the modules of the robot.

As mentioned earlier, templates are essential for automatic generation of CAD models. These templates are parametrically modeled using CATIA and stored in a library folder. Depending on the geometry of the CAD model being generated, CAD files are copied from the library folder to a working folder. The parts from the working directory are instantiated and constrained to each other in their respective sub-assemblies. Once all the sub-assemblies or modules are ready, they are inserted into the main assembly of the modular robot. All the instantiations are parametrically done using Visual Basic Script (VB Script). The above discussed approach forms the automated CAD framework for the modular robots which is shown in Figure 3.

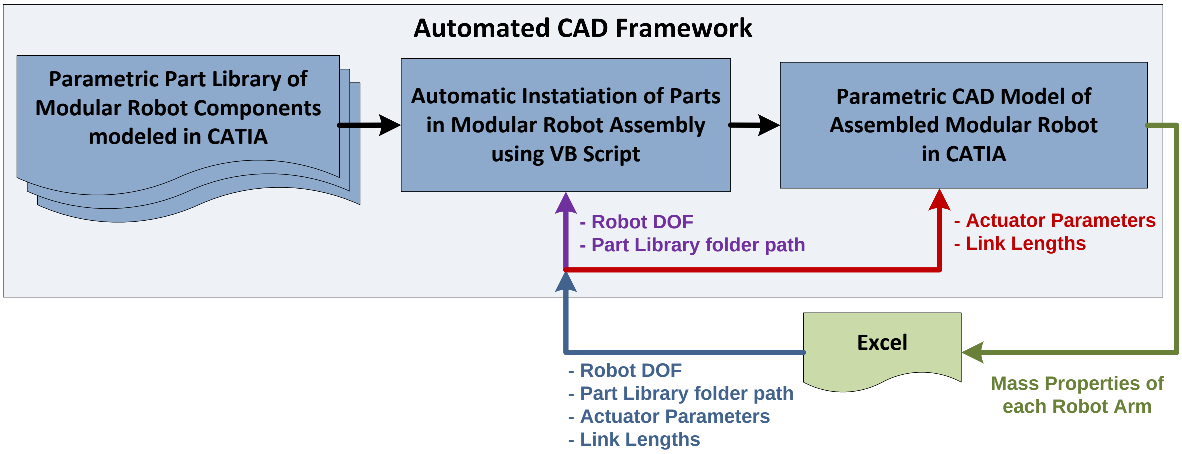

Figure 3. Automated CAD framework for modular robots

2.2 Automated Dynamic Simulation Framework

To investigate the dynamic aspects of the geometric CAD model created by the ACF, a dynamic model is required. These models are based on differential equations and algebraic equations. In the process of designing a robot, such models are often used to select suitable joint actuators by calculating the maximum torque required to rotate a joint at a certain rotational speed. Moreover, such models are also used to study the workspace of a robot and can provide a life time estimate of different components making up the drive train as demonstrated in [18].

To perform these analyses rapidly, an automated dynamic simulation framework, similar to ACF, is developed with a dynamic model at its core which is shown in Figure 4. This framework automatically retrieves mass and geometric properties of the current geometric CAD model, applies them to the dynamic model, performs a dynamic simulation and finally outputs the results of the simulation to the user interface for evaluation by a user or an optimization routine.

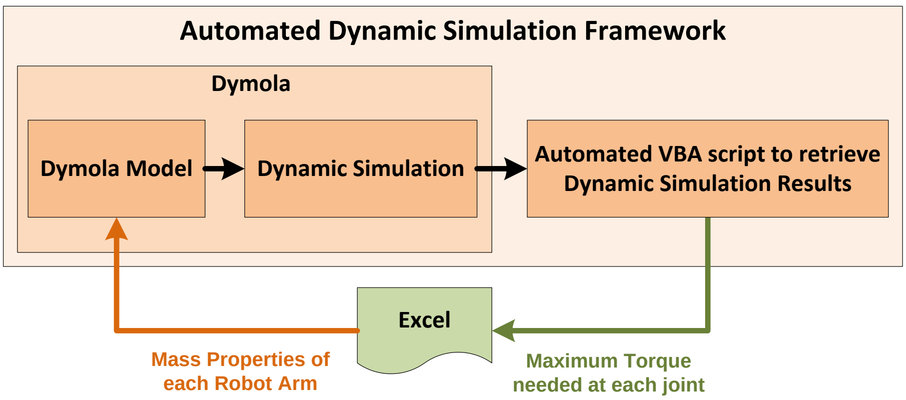

Figure 4. Automated dynamic simulation framework for modular robots

The degree of flexibility of an automated design framework is defined by the flexibility of its subcomponents. Introducing a non-parametric dynamic model in the framework would make the whole framework rigid. On the other hand a dynamic model offering more flexibility than the geometric model is also redundant. Commercially available softwares like Visual Nastran, ADAMS and Dymola offer the possibility of developing a parametric dynamic model, to different extents. For the presented work the dynamic model is developed using Modelica in Dymola.

Modelica is an object-oriented modeling language used largely for physical modeling of interconnected multidisciplinary systems. The hierarchical and object-oriented nature of Modelica makes it a prime candidate for modular design. Component models from the Modelica standard library [19] are used to develop the dynamic model.

2.3 Simulation-Based Optimization

In serial robots, the mass properties of each actuator affect the required torque in all previous actuators; therefore, a sequential approach is not suitable for selection of the actuators. Moreover, actuators of any industrial robot can highly affect weight, performance and cost of the robot. Therefore, it is important to ensure that a reliable procedure is implemented to select the actuators.

In this work, an optimization routine is utilized to choose the best suited actuators among available choices from a library which is stored in a Microsoft Excel sheet. During the optimization process, values of the various objectives are calculated by the use of different softwares and transferred to Microsoft Excel which acts as an interface between all framework components.

3 FRAMEWORK VALIDATION

To validate the automated framework, first, the CAD model of a seven DOF modular robot is generated. Afterwards, maximum torques required at each joint of the robot is calculated through a dynamic simulation of the most torque-demanding trajectories. Using an optimization process, the best suited actuators are selected based on the torques calculated in the previous step. Finally, a physical prototype of the designed modular robot is manufactured to evaluate the framework.

3.1 Automated CAD Model Generation

Figure 5 shows the procedure followed by the ACF to generate a seven DOF modular robot. As it is shown, first, components of modules are assembled and then the modules are placed in the robot structure. At the end of robot generation process in CATIA, the mass properties of each module which would be needed in dynamic simulation are transferred to the Excel interface.

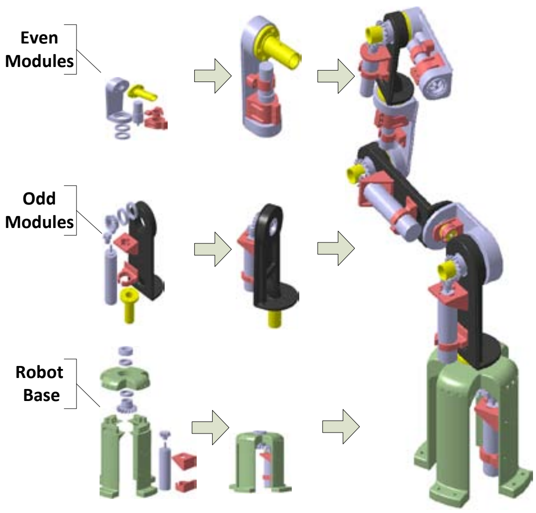

Figure 5. CAD model generation procedure for a 7 DOF modular robot

3.2 Automated Dynamic Simulation

To evaluate the dynamic properties of the generated CAD model a dynamic simulation is performed by following a predefined trajectory. The maximum required torque at each joint obtained from the dynamic simulation is passed to the Excel interface automatically. The torque curves obtained from the dynamic simulation for each joint are shown in Figure 6.

3.3 Optimization: Selection of Joint Actuators

When design problems are characterized by discontinuous and non-convex design spaces, nonlinear programming techniques are inefficient, computationally expensive, and, in most cases, find the relative optimum which is closest to the starting point. Genetic algorithm (GA) is applicable for the solution of such problems. GA is based on the principles of natural genetics, and is a stochastic method that can find the global minimum of problems with a high probability [20]. In this work, GA is used as the optimization algorithm.

The CAD model shown in Figure 5 contains randomly selected actuators from the actuator's library. In the following section, it is illustrated how the optimized actuator at each joint of a 7 DOF modular robot is selected by the optimization algorithm.

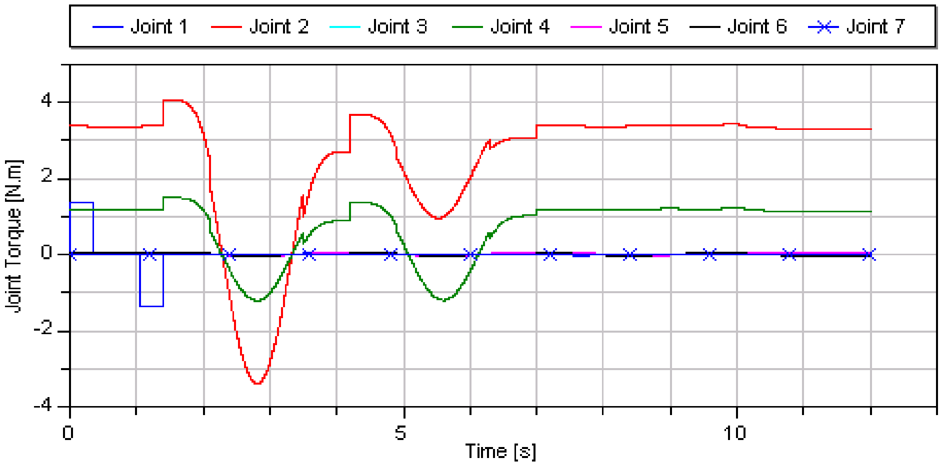

Figure 6. Torque requirement at different joints

3.3.1 Problem Formulation

In an optimization problem, the problem formulation is of a high importance as it would directly affect the results of the optimization. Choosing the lightest actuator for each joint in a serial robot would minimize the total weight of the structure and is an objective which is subjected to some constraints. The length of actuators is the most important constraint since it should be smaller than the link length so that the actuator would be mountable on the structure. The next constraint is the delivered torque by each actuator which should obviously be more than the required torque calculated in the dynamic simulation.

Briefly, the problem of selecting the best actuators for each joint can be summarized as follows:

Objective (F): Minimizing the weight of robot structure by selecting the lightest actuators

Subjected to:

Constraint 1 (g_1): Length of the actuators should be less than the link length

Constraint 2 (g_2): Required torque at each joint should be delivered by the actuator

Weighted sum method is chosen to formulate the abovementioned problem. The constraints are reformulated as penalty functions and multiplied by suitable weight factors according to their level of importance. Subsequently, the penalty functions are added to the objective function according.

The predefined reach of the robot limits the length of each link in the structure. Thus, as mentioned before, the length of each actuator could not exceed a certain value. The formulation of the length constraint as a penalty function, $p_1(x)$, would look like:

$$L_i = \begin{cases} q_1, & \text{Length of Link (i)} - \text{Length of Actuator (i)} < 40\,mm \\ 0, & \text{Otherwise} \end{cases} \quad (1)$$

$$p_1(x) = \sum L_i \,, \; i = 1, 2, \dots, DOF$$

Reformulation of the second constraint as a penalty function, $p_2(x)$, is:

$$T_i = \begin{cases} 0, & \text{Calculated torque} < \text{Torque delivered by the actuator} \\ q_2, & \text{Otherwise} \end{cases} \quad (2)$$

$$p_2(x) = \sum T_i \,, \; i = 1, 2, \dots, DOF$$

Large values are selected for q_1 and q_2 in (2) and (3), to avoid the selection of unsuitable actuators. The formulation of f(x) is according to (3):

$$M_i = \text{Mass of each module containing all component}$$
$$M(x) = \sum M_i \,, \; i = 1, 2, \dots, DOF \quad (3)$$

As a result, the primary optimization problem which had two constraints can now be rewritten as the following optimization problem without any constraint.

$$Min\ f(x) = M(x) + \sum w_j p_j(x_i) \quad (4)$$

$$\sum w_j = 1$$

$$g_k(x) = \sum w_j p_j(x_i)$$

$$x_i \in \{1, 2, 3, \dots, 30\}$$

$$i = 1, 2, \dots, DOF;\ j = 1, 2;\ k = 1, 2$$

According to the importance of the constraints, the weight factors are selected to be $w_1 = 0.6$, $w_2 = 0.4$.

3.3.2 Optimization workflow

Actuators are represented as parametric CAD models. Depending on the DOF, corresponding number of actuator models are instantiated in the robot structure. In each iteration, the dimensions and mass of all selected actuators are sent to CATIA. After updating the CAD models, the mass properties of each module are sent back to the Excel interface. Figure 7 shows the workflow of the optimization process.

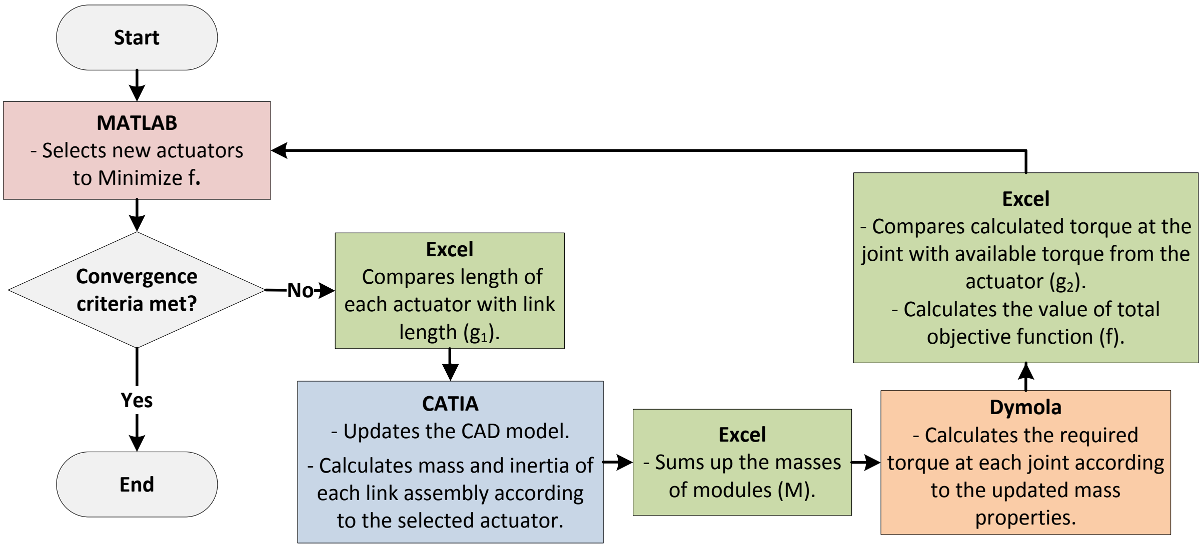

Figure 7. Optimization workflow

3.3.3 Optimization results

The optimization process took nearly 13 hours on a computer with 3.2 GHz microprocessor and 2 GB of RAM. Initial settings for the GA are as follows:

Number of generations = 150
Number of individuals = 20
Generation Gap = 0.9
Number of design variables = 7 (The following results are for a 7 DOF robot)

Each individual x is an array of 7 integers which correspond to the number of actuators. Since the problem is formulated as a minimization problem, the better individuals are the ones with lower objective values. According to Figure 8 and Figure 9, one can see how the objective values of individuals and the average of objective function values have changed during the optimization process.

3.4 Framework Evaluation Using Physical Prototype

Finally, a physical prototype is manufactured using a 3D printer to further evaluate the framework results using some preliminary tests. The physical prototype of the robot along with its CAD model is shown in Figure 10.

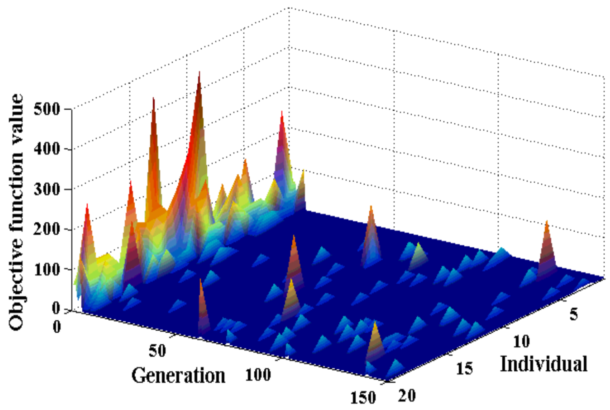

Figure 8. Objective function values for all individuals during the optimization process

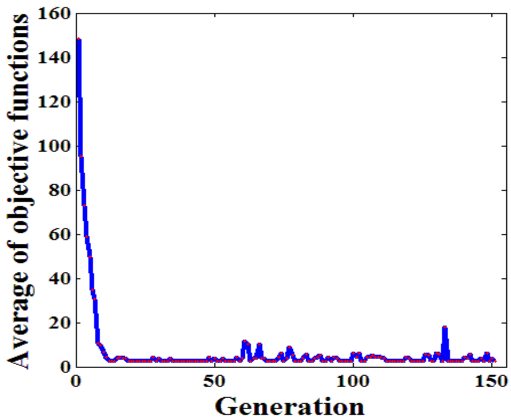

Figure 9. Average objective function value in each generation

Figure 10. Prototype vs. CAD model

In Dymola, the torque calculated at each joint is dependent on the mass properties calculated in CATIA. To ensure the validity of the calculated torques, a comparison between the mass of manufactured components and the calculated mass of corresponding CAD model should be made. This comparison is shown in Table 1.

Table 1. Mass comparison between CAD model and physical prototype of a 7 DOF modular robot

Mass	Module 1	Module 2	Module 3	Module 4	Module 5	Module 6	Total
CAD Model [kg]	0.954	0.535	0.935	0.441	0.506	0.427	3.798
Prototype [kg]	0.869	0.503	0.886	0.399	0.465	0.389	3.511
Difference (%)	8.9	5.9	5.2	9.5	8.1	8.9	7.6

According to the comparison, shown in Table 1, the calculated mass of each module is higher than the measured mass of the respective module in the prototype. This is the result of assigning higher density values to the CAD models than that of the material used during manufacturing. This ensures that the torque calculation at each joint has a certain amount of safety factor.

As mentioned in Section 3.3, an optimization routine is employed to select the actuators. To evaluate this selection of actuators, the physical prototype is set to follow different trajectories including the one used in dynamic simulation during optimization. These trajectories for the physical prototype are generated in joint space using a trajectory planner implemented in Dymola, which are passed to the corresponding joint actuator in real-time as shown in Figure 11. The trajectory generated in Dymola is transferred to LabVIEW using a MATLAB (Simulink) interface. A motion controller and a servo amplifier are used to generate and amplify the required command signals for each actuator. A control system is also implemented by using position feedback from an encoder mounted on the actuator and velocity feed-forward.

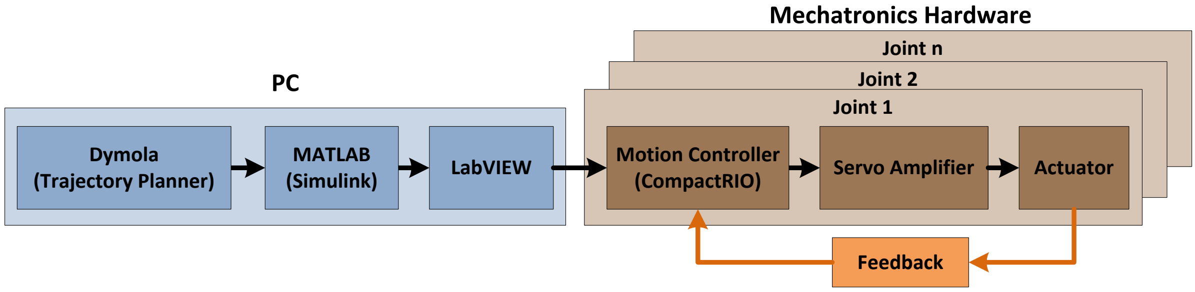

Figure 11. Robot motion-control procedure

The actuators were able to follow their target trajectories with an acceptable amount of positional error. As an example, the performance of an actuator whilst following one of the trajectories is shown in Figure 12. This test confirms that the selection of actuators by the optimization routine is valid. These tests, though preliminary in nature, help in increasing the confidence of the designer in the presented automated framework.

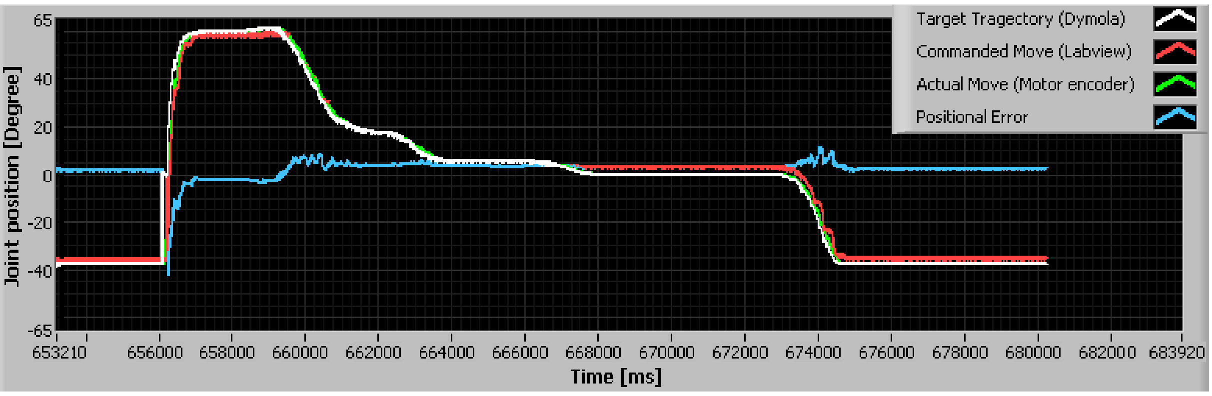

Figure 12. Trajectory tracking performance of an actuator

4 CONCLUSIONS, DISCUSSION AND FUTURE WORK

In this paper an automated framework for design and evaluation of modular robots is presented. A design example focusing on the design of a seven DOF robot using the presented framework is also discussed. An optimization is performed to select the best suited actuators to the robot structure. Finally, a prototype is manufactured to validate the results of the framework.

During the design process, using the automated framework the changes were incorporated in the robot structure rapidly. Using an automated dynamic simulation framework, the dynamic analysis of the robot was done effortlessly, which enabled for a simulation-based optimization to be carried out. Moreover, the framework proved to be useful during the prototyping/manufacturing stage as last-minute changes could not only be incorporated but also were evaluated before production which would have been impossible to perform manually in the same time.

As a future work, optimization of other design variables like link lengths, payload, sharing of modules within a robot family, etc can be studied. For the optimization process, GA was used. As a future work, other algorithms, like Complex algorithm, can be implemented to confirm the optimality of the optimization results. Furthermore, a finite element analysis of the robot structure will be included in the framework to further enhance its functionality. This will allow the designer to optimize the morphology of the robot module for a given payload. For further evaluation of the framework, a full-scale robot based on the manufactured prototype, using industry standard components can be designed and manufactured.

ACKNOWLEDGMENTS

The authors wish to thank Professor Johan Ölvander and Edris Safavi of Linköping University and Dr. Xialong Feng of ABB Corporate Research for their valuable guidance and constructive comments. We would also like to express our gratitude to National Instruments for their sponsorship and technical support.

REFERENCES

[1] Jouannet, C., Lundström, D., Amadori, K. and Berry, P. Design of a Very Light Jet and a Dynamically Scaled Demonstrator. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, January 2008.

[2] Tarkian, M. and Zaldivar, F. Aircraft Parametric 3D Modeling and Panel Code Analysis for Conceptual Design. In *26th ICAS*, Anchorage, USA, September 2008.

[3] La Rocca, G., van Tooren, M.J.L. Enabling distributed multi-disciplinary design of complex products: a knowledge based engineering approach. *Journal of Design Research*, Vol. 5, No. 3, 2007, pp.333-352.

[4] Tarkian, M. Ölvander, J., Feng X. and Pettersson M. Design Automation of Modular Industrial Robots. In *ASME CIE'09*, San Diego, USA, September 2009.

[5] Cooper D. and La Rocca G. Knowledge-based Techniques for Developing Engineering Applications in the 21st Century. In *Proceedings of 7th AIAA Aviation Technology, Integration and Operations Conference*, Belfast, Northern Ireland, September 2007.

[6] Takian M., Lunden B. and Ölvander J. Integration of parametric CAD and dynamic models for industrial robot design and optimization. In *ASME CIE'08*, New York, USA, August 2008.

[7] Johanson B., Jouannet C. and Krus P. Distributed aircraft analysis using web service technology. In *Proceedings of World Aviation Congress*, Montreal, Canada, September 2003.

[8] Ölvander J., Lunden B. and Gavel H. A Computerized Optimization Framework for Morphological Matrix Applied to Aircraft Conceptual Design. *Computer-Aided Design*, Vol. 41, No. 3, 2009, pp.187-196.

[9] Pettersson M. and Ölvander J. Drive train optimization of industrial robots. *IEEE Transactions on Robotics*, Vol. 25, No. 6, 2009, pp.2047-2052.

[10] Roos F.*Towards a methodology for integrated design of mechatronic servo systems,* Doctoral dissertation, Dept. Machine Design, Royal Institute of Technology, Stockholm, Sweden, 2007.

[11] Krenn, R., Schäfer, B. and Hirzinger G. Dynamics Simulation and Assembly Environment for Rapid Manipulator Design. In *7th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, ESTEC, Noordwijk, Netherlands, November 2002.

[12] Paredis, C.J.J., Brown, H.B. and Khosla, P.K. A rapidly deployable manipulator system. In *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 2, Minneapolis, MN, USA, April 1996, pp. 1434-1439.

[13] Motoman SIA20, http://www.motoman.com

[14] Nachi MR20, http://www.nachirobotics.com

[15] Petterson M., Andersson J. and Krus P. Methods for Discrete Design Optimization. In *Proceedings of ASME DETC'05, Design Automation Conference*, Long Beach, California, USA, September 2005.

[16] Safavi E., Tarkian M. and Ölvander J. Rapid Concept Realization for Conceptual Design of Modular Industrial Robots. In *NordDesign 2010*, Göteborg, Sweden, August 2010.

[17] Mäntylä M. A modeling system for top-down design of assembled products. *IBM Journal of Research and Development*. Vol. 34, No. 5, 1990, pp.636-659.

[18] Tarkian M, Ölvander J and Lundén B. Integration of Parametric CAD and Dynamic Models for Industrial Robot Design and Optimization. In *2008 ASME International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE)*, New York City, New York, USA, August 2008, pp. 761-769.

[19] Modelica Standard Library, https://www.modelica.org/libraries/Modelica/

[20] Rao S.S. *Engineering Optimization Theory and Practice, 4th Edition: Modern Methods of Optimization,* 2009 (John Wiley & Sons).