

Design optimisation of industrial robots using the Modelica multi-physics modeling language

A. Kazi,

(ArifKazi, GuentherMerk)@kuka-roboter.de
KUKA Roboter GmbH
Research & Development
Blücherstr. 144, 86165 Augsburg
Germany

G. Merk,

M. Otter,

(Martin.Otter, Hui.Fan)@dlr.de
German Aerospace Center
Institute of Robotics and Mechatronics
82234 Weßling
Germany

H. Fan,

ABSTRACT

Superior performance of industrial robots can only be achieved if the interaction of mechanics, electronics and software is taken into account already in the design phase. Conventional approaches to modelling, however, cannot cope with such multi-physics systems. The new Modelica modelling language has been specifically developed for this purpose. It also significantly increases re-usability of model components, since components adapt to the connection structure in which they are used.

In the European research project “Real-time Simulation for Design of Multi-physics Systems” (RealSim), efficient tools for modelling, simulating and optimising industrial robots are developed. A general Modelica MultiBody library (freely available from [1] or [2]) formed the basis for the development of a specific robot component library. A development environment for creating robot models out of the available components was provided, in which initial calculations to verify the first rough design can also be performed. Tools for automated optimisation were developed that take the initial design as the starting point. Afterwards, the performance of a design can than be verified in a real-time simulation before investing in building a real prototype. These powerful development tools provide the basis for an even more efficient development process and superior robot performance in the future.

Keywords Industrial robots, robot development, performance optimisation, multi-physics systems, Modelica language

1 INTRODUCTION

Industrial robots are truly “mechatronic” devices: their performance is governed by a close interaction of mechanics, electronics and software. When developing a new type of robot, it is therefore not sufficient to consider the different physical domains individually. Superior performance can only be achieved by taking a holistic view that covers all aspects.

A large amount of simulation software is commercially available, and it is already being used extensively in the design of industrial robots. Current methods are, however, usually strong in one physical domain only. For example, the 3D mechanical

construction can be simulated efficiently with a multi-body program, yet not the electric drives. Block diagram-based simulation software requires that the user performs the most difficult part of the modelling process manually: he has to transform the natural non-causal description of physical systems into a block diagram form.

Modelica, on the other hand, is a new, free modelling language that can cope with large scale, multi-physics systems [1]. It features non-causal, object oriented modelling: direct equations are used instead of assignment statements. This significantly increases re-usability of model components, since components adapt to the connection structure in which they are used. The Modelica simulation environment Dymola from Dynasim [3] provides the means for real-time simulation of large Modelica models with short deadlines. In the European research project “Real-time Simulation for Design of Multi-physics Systems” (RealSim, IST-Project 1999-11979 [2]), the Modelica language is applied for optimising the design of industrial robots.

With the tools developed in the RealSim project, the design of a new industrial robot will follow a certain typical sequence (see fig. 1): After the initial specification of the robot (desired payload, work space, etc.), a first model of the robot is created. In the robot component library, the data of older designs are available, which will serve as a reference. The initial design is carried out following heuristic rules. “Static” calculations are performed to check the joint torques in different arm configurations and adjust the kinematic parameters.

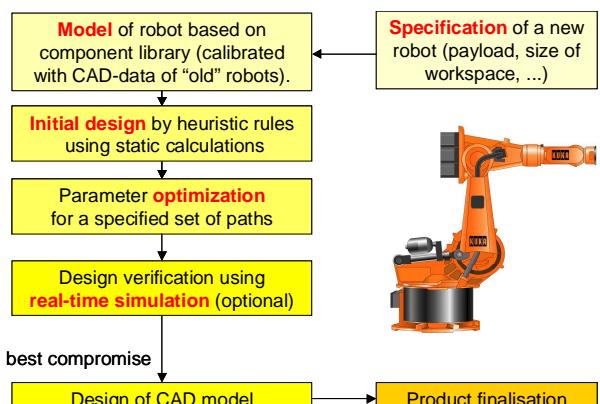


Fig. 1 – Design process for industrial robots.

The initial design is followed by a min/max parameter optimisation phase in which a good compromise candidate is determined by minimising the maximum of a set of (scaled) criteria.

Finally, the design can be verified in a real-time simulation using the actual robot control hardware before resources are allocated to building a prototype. The following sections describe the individual design steps and the corresponding RealSim tools in the sequence they have been mentioned above.

2 DESIGN SPECIFICATION

The design and optimisation of a robot can only have a successful outcome if the specification is sufficiently exact. As a starting point, the following parameters are of utmost importance:

- Robot payload - masses, moments of inertia, and load center distances
- Robot dynamics - final velocity and acceleration of the robot
- Robot work envelope

At first glance, it would appear desirable to always keep payload, dynamics and work envelope as high as possible. However, the space required for the robot may then become prohibitively large for certain applications. Secondly, performance which is not required causes the cost of the robot to rise massively. A good specification is therefore essential for an economically sound result.

3 MODEL DEVELOPMENT

For the simulation and optimisation of a robot design, a mathematical model has to be available. For reasons of efficiency, this model will be created from a library of components, in which also old designs are available which can serve as a reference.

3.1 Modelica MultiBody library

As a basis for modelling robots of all kinematics, a Modelica library was developed to model arbitrary 3-dimensional mechanical systems. The Modelica MultiBody library can be combined with the existing 1-dimensional mechanical Modelica libraries. It is also possible to define multibody systems that contain kinematic loops. A unique feature is the efficient treatment of joint locking and unlocking. This allows, for example, easy modeling of friction or brakes in the joints. This library is freely available and can be either downloaded from the Modelica home page [1] as part of the ModelicaAdditions library or from the RealSim project home page [2].

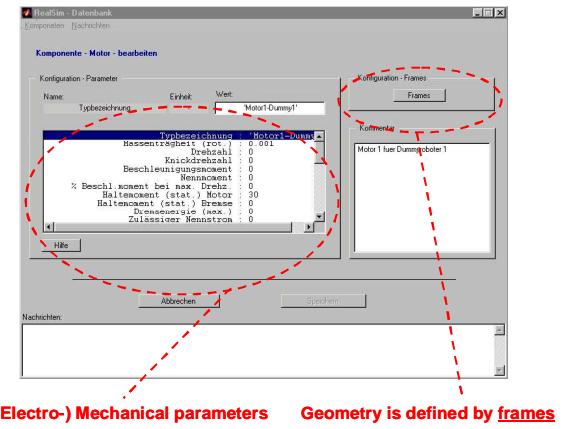
3.2 Robot component library and data base

In the next step, a proprietary Modelica robot component library and a corresponding data base were developed. The robot component library contains the general equations, while the data itself is kept in the data base. Component library and data base are usable for any kind of kinematic system, with an arbitrary number of axis and axis coupling.

3.3 Defining the model

The robot components and the designed robots are stored in a data base. To be open for all possible future extensions, the data base is not strictly designed for 6-axes robots, but to store any kind of kinematic system. Matlab is used as a front-end to the data base system to enter new parts, modify existing ones and to define robots (components and how components are connected together). In the data base, individual parts and complex structures built from the parts are stored separately.

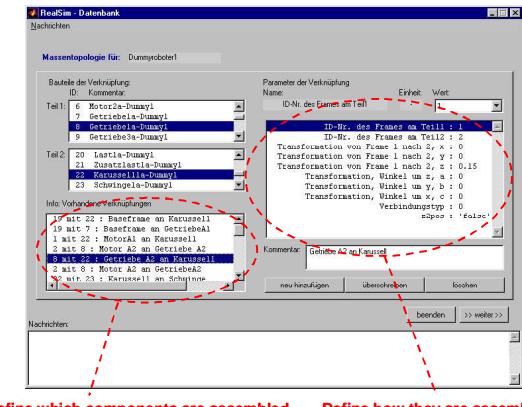
The graphical user interface implemented in Matlab provides the user with an intuitive means for defining new parts or complete robots. A screen shot of the interface for defining a new part is shown in fig. 2. Means are provided to enter the (electro-) mechanical parameters of the part as well as the kinematic frames which define the geometry for connecting the part.



(Electro-) Mechanical parameters Geometry is defined by frames

Fig. 2 - GUI for the definition of a part.

Fig. 3 shows the equivalent screen shot for entering a new robot. In the user interface, it is defined which parts are connected as well as how they are assembled with respect to their kinematics (definition of rotational or translational degrees of freedom, etc.)



Define which components are assembled Define how they are assembled

Fig. 3 - GUI for the definition of a kinematic structure.

4 INITIAL DESIGN

Before an optimisation can start, a reasonable first initial design has to be available. This is a difficult task by itself since all components – such as motors, gear units, counterbalancing system, structural parts

etc. – need to be designed in such a way that the essential requirements are already fulfilled and the robot is functional, i.e., simulations can be carried out. Otherwise, the optimisation task is too difficult to be solved with today's optimisers. In order to verify that the initial design meets the requirements, some heuristic tools for “static calculations” are provided.

4.1 System overview

For performing static calculations, the system described in section 3 was extended. The software prototype now consists of three parts: The data base with the component library, the Matlab user interface and a calculation kernel.

- As before, the ordinary user only has to deal with the user interface programmed in Matlab. All pre-processing and post-processing features are also integrated, so that all general conditions for any calculation are defined here.
- The calculation kernel runs under the Modelica simulation environment Dymola [3]. Dymola accesses the robot component library, in which the equations for performing the computations are stored.

The computational process itself runs as follows (see fig. 4). After the complete robot has been created in the user interface, the Matlab data structure is stored as a Modelica model in a file using the actual parameters from the data base. Afterwards, Dymola is started and reads this file, performs the required calculation on this model and stores the simulation result back in a result file. Finally, the result file is passed to Matlab for post-processing.

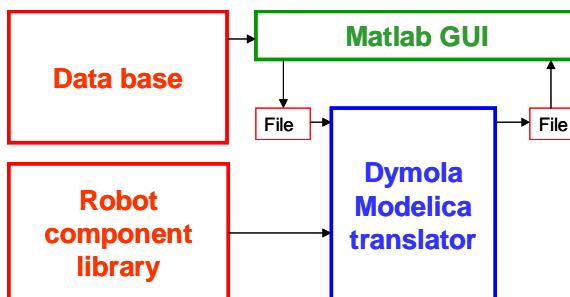


Fig.4 – Architecture of the software prototype.

4.2 Performing calculations

In the window of this calculation tool, one can manually enter the joint angles of the robot. The tool will then calculate the corresponding motor and gear torques in the robot in this position. The results are displayed in a graphical format (see fig. 5).

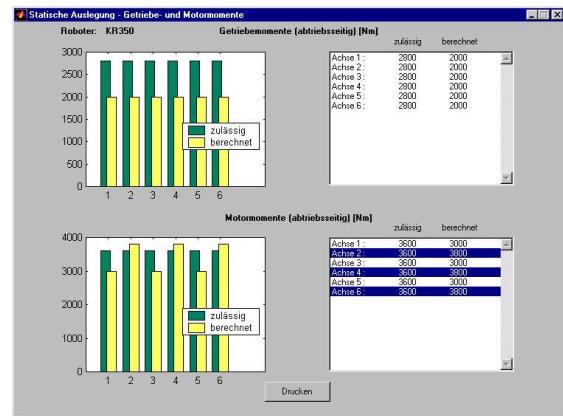


Fig. 5 – Results of the static calculation

5 OPTIMISATION

In the preceding sections, a method has been devised for simulating industrial robots. The aim now is to expand this capability to allow for automatic optimisation. This optimisation procedure is relatively complex and it is not always clear which parameters are to be optimised. In order to illustrate the complexity of the task and the interaction of the elements to be optimised, an overview of the aspects to be considered is given.

5.1 Optimisation criteria and parameters

In most cases, the optimisation criterion with the highest priority is robot cycle time for particular tasks, taking into account the limits of the robot (such as maximum motor speeds). Also energy consumptions of the motors can be used as additional lower priority criteria. All parameters in the robot model, such as gear ratio or link length, can be utilised in the optimisation process.

5.2 Selection of the robot path

The layout of the path also has a major influence on the optimisation of the robot. A path with long motions and few intermediate points is generally optimal in terms of cycle times, with as high a final velocity as possible maintained for as long as possible. Such paths are encountered in certain press linking or handling applications. In spot welding programs, on the other hand, paths have a large number of points with relatively short gaps between them. The robot rarely reaches its final speed, but high acceleration is nonetheless very important.

The user must, therefore, have a clear idea of the planned application profile of the robot and choose the paths used for optimisation purposes accordingly. In many cases, robots are built to cover as wide a spectrum of applications as possible. In this case, a compromise must be found between final speed, acceleration and energy consumption. This factor represents a major potential source of errors and cannot be automated, but depends strongly on the experience of the user.

5.3 System overview

MOPS (Multi-Objective Parameter Synthesis), which has been developed in DLR during the past years, is a modular Matlab-based software approach for practical design applications based on min-max parameter optimisation [4]. The main feature of the MOPS methodology is that the various kinds of design objectives can be taken into account in their most natural form and that design alternatives can be assessed most visibly with respect to given requirements. Multi-objective synthesis tuning by min-max parameter optimisation allows interactive compromising in the set of what can be best-possibly achieved within a chosen model structure.

The optimisation process is shown in fig. 6.

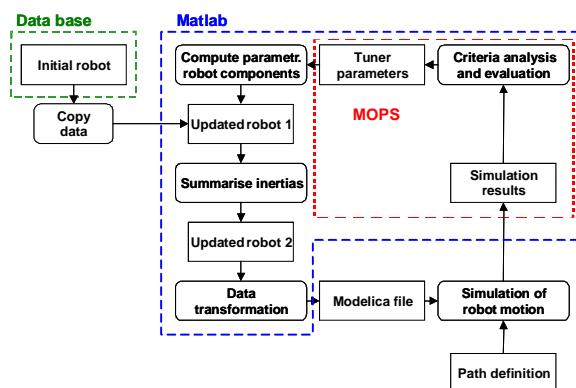


Fig. 6 – Optimisation of the robot design

During the MOPS optimisation iterations, a copy of the initial robot data structure ("updated robot 1") is used and the parameters in this data structure, e.g., gear ratios, are overwritten with new values proposed from the optimiser. In order to speed up the model simulation, the many bodies and drive inertias present in the "updated robot 1" data structure are summarised whenever possible (e.g. the mass properties of the gear units and motors attached to a structure part are summarised into one body). The result is an "updated robot 2" data structure.

The "updated robot 2" data structure is written into a data file in Modelica format ("Modelica-File") and used as input for robot motion simulation, together with the reference path on which the robot should move ("path definitions"). It serves as a basis for a simulation of the robot which produces the desired criteria variables, such as cycle time etc. The tuner parameters for the next simulation run are generated by MOPS. In every optimisation iteration, several simulations may be performed in order to calculate the criteria values for different reference paths to take into account the wide spectrum of usage of the robot.

6 REAL-TIME SIMULATION

The possibility to perform real-time simulations becomes particularly important in the later stages of the design process. The final design can be verified before one embarks on the costly and time consuming process of building a prototype. A detailed model of the mechanics and the drive system of a six-axis industrial robot with about 1000 algebraic equations

and 80 states was used as benchmark. Using a new technique for real-time simulation of stiff systems developed in the RealSim project by Dynasim and DLR [5,6], this model was then semi-automatically partitioned into the "slow" mechanical part and the "fast" drive system part involving much smaller time constants. This mixed-mode integration approach resulted in a performance improvement of more than 15 with respect to the high quality, variable step-size integrator DASSL and to the fixed step-size methods of explicit and of implicit Euler. The simulation model run on a 650 MHz desktop PC easily in real-time, while receiving input from a KUKA office PC running the original KUKA control software. The set-up of the complete real-time simulation system is shown in fig. 7.

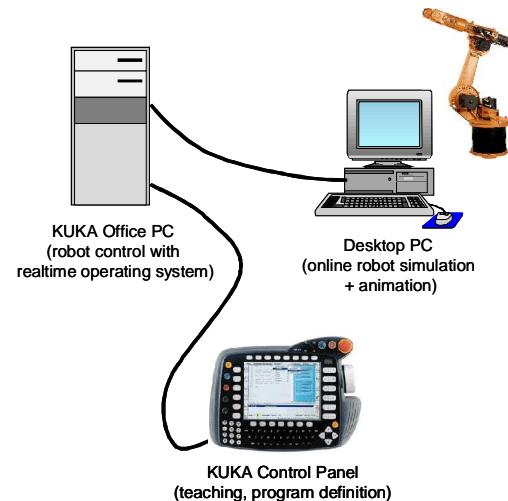


Fig. 7 - Set-up of the real-time simulation system.

In the simulation, the deviation between the commanded path of the robot and its real path was evaluated. During the motion of the robot, an online animation with robot CAD data, where the size and direction of the deviation was visualised on the desktop PC, gave immediate visual feedback (see fig. 8).

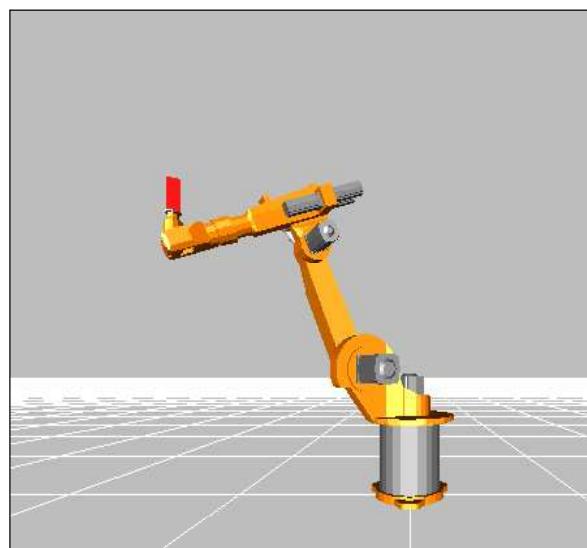


Fig. 8 - Visualisation of the deviation between commanded and real path in the simulated robot.

The simulation displayed a high correlation with the performance of a real robot of this type. A picture from the online teaching with the KUKA control panel and the online visualisation of the robot is shown in fig. 9.



*Fig. 9 - Interaction and visualisation
of robot real-time simulation.*

7 CONCLUSIONS

Industrial robots are complex systems involving mechanics, electronics and software. For an optimised performance, the development environment needs to offer means to consider the close interaction between all three physical domains. The new Modelica modelling language offers a unique platform for the simulation of such multi-physics systems. The robot component library presented in this paper provides the means to design and optimise new robots of almost arbitrary kinematics very efficiently. After the initial design and the optimisation phase, the robot model developed can even be used in real-time simulations in order to verify the design.

8 REFERENCES

- (1). Modelica homepage at www.modelica.org.
- (2). RealSim project information available at www.ida.liu.se/~pelab/realsim.
- (3). Dymola homepage at www.dynasim.org.
- (4). Joos H.-D.: "A methodology for multi-objective design assessment and flight control synthesis tuning". Aerospace Science and Technology, 1999, no. 3, pp. 161-176.
- (5). Schiela A., Olsson H.: "Mixed-mode Integration for Real-Time Simulation". Modelica'2000, Oct. 23-24, Lund, Sweden, 2000. Downloadable from <http://www.Modelica.org/workshop2000/proceedings/Schiela.pdf>.
- (6). Elmqvist H., Mattsson S.E., Olsson H.: "New Methods for Hardware-in-the-Loop Simulation of Stiff Models". Modelica'2002, March 18-19, 2002. Downloadable from: http://www.Modelica.org/Conference2002/papers/p08_Elmqvist.pdf.

9 ACKNOWLEDGEMENTS

This work was funded in parts by the European Commission within the Information Societies Technology (IST) programme under contract number IST 1999-11979.