Robot off-line programming and simulation as a true CIME-subsystem

Nielsen, L.F; Trostmann, S; Trostmann, Erik; Conrad, Finn

Published in:
Proceedings of the IEEE International Conference on Robotics and Automation

Link to article, DOI:
10.1109/ROBOT.1992.220203

Publication date:
1992

Document Version
Publisher's PDF, also known as Version of record

Proceedings of the 1992 IEEE
International Conference on Robotics and Automation
Nice, France - May 1992

Robot Off-line Programming and Simulation As a True CIME-Subsystem.

L.F. Nielsen, M.Sc. S. Trostmann, M.Sc.
E. Trostmann, Professor, Ph.D. F. Conrad, Ass. Professor, M.Sc.

Control Engineering Institute, Technical University of Denmark
DK-2800 Lyngby, Denmark.

Abstract. *An implementation of a new robot off-line programming and real-time simulation system, ROPSIM, based on a generic architecture is described. As a new feature ROPSIM is using a neutral interface communication at input/output level based upon using the ISO standard proposals STEP and ICR. This ensures a straight forward and well defined procedure for the integration of the ROPSIM system in a largerCIME system architecture. In order to avoid dependency of dedicated robot models ROPSIM is based on genericmodels describing the robot controller (incl. external sensor feed-back), robot arm geometry and the robot arm kinetics. The software is developed using the C++ programming language and key modules like robot program interpretation (ICR), simulation algorithm, calculation of direct and inverse dynamics (for open as well as closed kinematic chains), and visualization (animation) of results are presented. Also results from simulations are illustrated.*

Keywords. *Robots, control, off-line programming, simulation, animation, modeling, CIME system architecture, integration, data transfer, neutral interfaces, ICR, STEP.*

1 Introduction

Robots offer a high degree of automation, programability, and repeatability. This makes robots suitable as production components in a Computer Integrated Manufacturing and Engineering (CIME) system. The programability is useful for making required adjustments or changes in a production. The repeatability contributes to a higher quality of the produced products.

It is a general trend that many companies want to produce more and more complex, and frequently changing products in small batches at a *high speed* and with a *high repeatability* by using robots. This has brought up problems in the control of the dynamic behaviour of the robot, and raised the issue of robustness of the control system. Also vibrations and collision avoidance become significant problems. In order to overcome these problems the robots are provided with the possibility to measure, via external sensors, both the state of the robot system itself and the state of the environment. The sensor signals are feed back to the control system and used for generation of new control input.

In order to benefit from the programability and repeatability of the robot the selection of the programming method becomes an important issue, because it effects both the cost effectiveness of the programming phase and the reliability of the developed programs.

Today the task of generating programs is a bottle-neck for obtaining better performance and more effective use of robots. When the programming tools become more efficient, then robots will become suitable in many more production systems.

In order to develop reliable and time optimal programs and as a result of the ambition to produce at a high speed and with high repeatability, a real-time simulation, including the true dynamic behaviour of the total robot system and the use of external sensors is required.

In the simulation and programming of the robots the modeling of the robot system is a critical issue. In fact the accuracy and effectiveness of the programming, simulation, and control of the robot depend on the model of the robot. The situation today is that the modeling facilities of most programming and simulation systems are not capable of describing either the full dynamic behaviour of the total robot system nor the use of external sensor feed-back in the generation of control data.

In conclusion there is a need for a programming and simulation system for robot driven workcells that illustrates the true real-time behaviour of the total robot system. Such a system should, in order to be a true CIME subsystem, facilitate the exchange and reuse of robot model definition data and robot program definition data with systems of other origin or different functionality.

This paper describes in the following paragraphs such a programming and simulation system currently under development at the Control Engineering Institute, Technical University of Denmark.

2 The neutral interface concept

Many companies have experienced a pressing need for the integration of all the levels of manufacturing from the identification of business objectives through design, planning, and to real-time control of machining and assembly at the shop floor and further to the shipping of finished products.

A rational way of achieving this integration is the concept of neutral interfaces. Integration in this context is defined as the process of linking different system components into a complete system which fulfils the objectives and specifications of the integrated manufacturing system. The neutral interface concept leads to a conceptual schema for a CIME system architecture as shown in Fig. 2.1.

Three levels of system categories are identified. Design systems, planning systems, and real-time pro-

0-8186-2720-4/92 $3.00 ©1992 IEEE

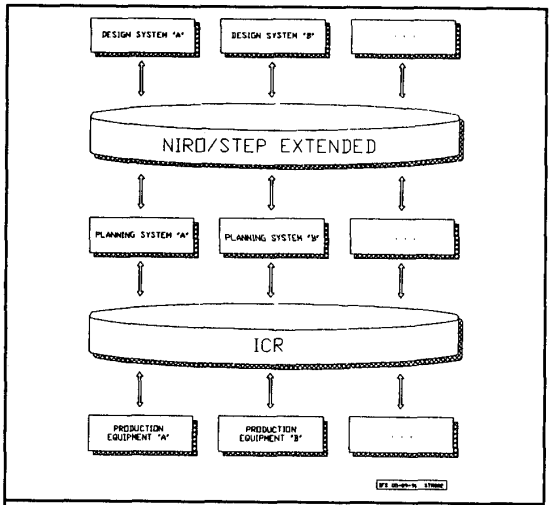

Figure 2.1. *Conceptual schema for a CIME system architecture using the neutral interface concept [9].*

duction systems. Systems at adjacent levels are integrated through neutral interfaces such that any system in principle can exchange pertinent data with other systems. One neutral interface NIRO/STEP EXTENDED [6] links the design domain with the planning domain. NIRO/STEP EXTENDED holds information about geometry and kinematics. Another neutral interface ICR [4] links the planning level with the real-time control of the production system. ICR holds a description of the task to be executed.

The conceptual schema shown in Fig. 2.1 has lead to an implementation of a CIME system for off-line programming and control of robots based upon neutral interfaces at the Control Engineering Institute, Technical University of Denmark. The detailed architecture of this system is shown in Fig. 2.2.

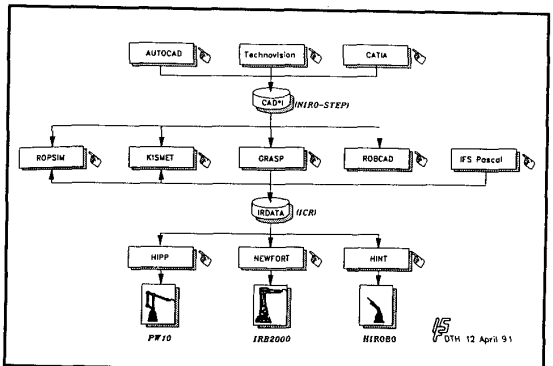

Figure 2.2. *CIME system implementation at the Control Engineering Institute.*

The actual implementation includes the following sub systems.

* Three commercial CAD-systemsi : CATIA, AUTOCAD and Technovision.
* Five off-line programming systems : ROPSIM, KISMET, GRASP, ROPCAD and IFS PASCAL.
* Three industrial robots : Hitachi PW10, ABB IRB 2000, HIROBO and MULTIDOS test robot.

The main results from this implementation have demonstrated that existing subsystems, commercial available or inhouse made, can be integrated though a systematic use of neutral interfaces so they can operate with a better overall performance. It has also been demonstrated that this concept allows for building individually tailored CIME systems that later easily can be rearranged or expanded in accordance with the users needs and opportunities.

3 Generic architecture for off-line programming and real-time simulation systems

This section introduces a generic architecture for a robot off-line programming and real-time simulation system. The architecture identifies the necessary components of such a system, see Fig. 3.1. Each of the identified components may be found in various commercial simulation systems at different level of implementation. The generic architecture can be used to evaluate and compare different systems for off-line programming and simulation.

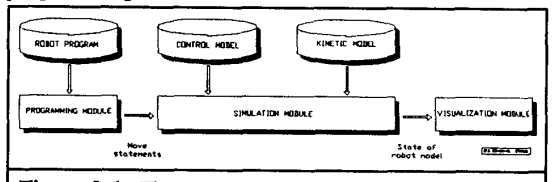

Figure 3.1. *The generic architecture of robot off-line and real-time simulation systems.*

The components of the architecture can be grouped into modules and models. In Fig. 3.1 the interface between the modules and the relationships of the models are shown.

The information content of the individual models depend on the complexity of the system. The models must carry the information needed in the modules. The control model holds a description of the controller which include the structure and the functions (algorithms) of the controller. As kinetics is the study of all aspects of motion, the kinetic model of the arm system and must include the necessary information for simulating the kinetics. The kinetic model may include kinematics, dynamics and geometry. The robot program model carries a description of the how to perform a given task. The three modules is described more detailed in the following.

3.1 Programming module

The programming module can be viewed as a function. The input is a description in a relevant programming language of the actions the robot shall execute to perform a given task. The module (function) carries out an interpretation of the input on basis of a model of the input language. The output is a flow of move instructions to the controller of the robot.

3.2 Simulation module

The simulation module can be split into two parts. One part concerning simulation of the behaviour of the controller and another concerning the behaviour of the robot arm kinetics. The controller simulation takes the move instructions from the programming module as input and generates, on basis of the controller model, control vectors (control signals to the actuators) as output. The simulation of the robot arm takes control vectors as input and calculates, on basis of the kinetic model, the internal states of the robot arm. The final result of the simulation module is a flow of states of the whole robot model (controller and robot arm).

3.3 Visualization module

It is the purpose of the visualization module to illustrate the simulation results generated in the simulation module. The visualization module receives a flow of states of the robot model and may on this basis generate any kind of presentation of any part of the robot model. This could be a 3D animation of the robot movements, diagrams presenting the state of the servo loops, etc.

4 ROPSIM

The following sections focus on the development and implementation of a new **R**obot **O**ff-line **P**rogramming and real-time **SIM**ulation system, ROPSIM.

ROPSIM offers two new important facilities compared to existing commercial systems. Firstly, ROPSIM is designed on basis of the neutral interface concept. Two different neutral interfaces are used in ROPSIM, ICR for the communication of robot programs and NIRO/STEP EXTENDED for communication of robot arm kinetic models.

Secondly it is of great importance that a simulation system can handle models of the necessary complexity. Using a simple model may deteriorate the correctness of the simulation. To overcome this ROPSIM has been designed to simulate the static and dynamic behaviour of both the robot controller (incl. external sensor feed-back) and the robot arm.

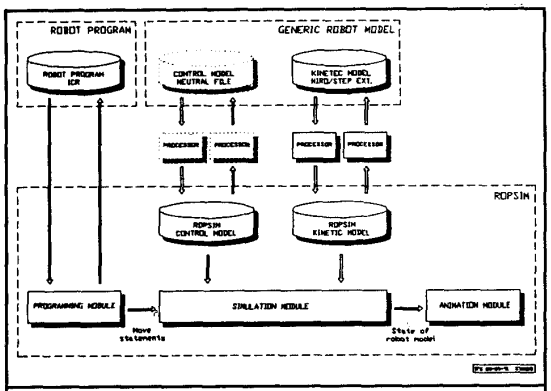

Figure 4.1. *The implemented architecture of ROPSIM.*

In the next sections the neutral interfaces ICR for robot program representation, the generic robot model, and the specific module algorithms in ROPSIM are described.

4.1 Intermediate Code for Robots, ICR

The information in an ICR [4] robot program can be grouped into two categories. One category dealing with data manipulation instructions and another category dealing with robot related instructions.

The data manipulation instructions correspond to the intermediate code level of computer programming. This means that the data manipulation instructions of ICR is a simple hardware independent stack-machine. This permits translation of the information content of a high level programming language like Pascal into ICR. Also programming languages of other programming systems like ROBCAD, GRASP ,etc. can be translated into ICR.

The robot related instructions describe the explicit movements of the robot. The important aspect here is that the instructions in ICR dealing with robot movement is robot independent in the sense that they are the same for all robots. Therefore programs expressed in ICR is not dedicated to a specific robot and can in principle be executed on different robots. The robot task described in the ICR program may implicitly require a robot with a special kinematic structure, e.g. 6 degrees of freedom, in order to achieve a successful execution of the task. In that case the ICR program can be executed on a family of robots with the proper kinematic structure.

The application of the ICR neutral programming interface offers the following advantages :

* Exchange of robot programs between different programming systems and robots.
* The exact same robot program is used for the simulation and for real-time execution on the robot.
* Communication of the verified robot programs to the real-time equipment.

4.2 Generic robot model

The generic robot model is used as the fundamental information carrier between ROPSIM and other software systems when a specific robot model is communicated, ref. Fig 4.1. The objective of the generic model is to ensure that ROPSIM is independent of which system (software) is used for the design of the robot model. The structure of the generic robot model is based upon the neutral interface concept and as a result the generic robot model is an extremely useful tool for the sharing of robot model data within software systems of different origin and functionality.

The generic model of robot systems consists of a model of the robot controller and a model of the robot arm kinetics.

The generic controller model: The generic controller model must be capable of describing a wide range of different controllers (motion control, force control, adaptive control, etc.). Here an example of a generic model for motion control is given.

The generic model for motion control defines a fundamental structure of the robot controller. The fundamental structure covers most robots which deals with motion control. Also the integration of external sensors with the robot controller is covered. Sensors in the servo loops are considered as internal sensors. The structure of the generic controller model is illustrated in Fig. 4.2.

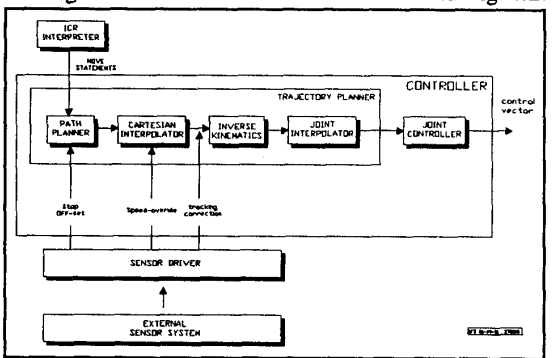

Figure 4.2. *The generic robot controller entities and their structure.*

The generic model for motion control classifies six entities. These are : Path planner, interpolator, inverse kinematic transformation, joint interpolation, joint controller, and an interface for external sensors.

Also the generic model for motion control specifies a specific configuration or interconnection of these six entities. This configuration is illustrated in Fig. 4.2.The only freedom in evaluating the model lies in the actual implementation of the individual functions or control algorithms (the boxes on the figure). This makes sense because the entities and the interconnection of the entities specified in the generic model for motion control are found in almost all industrial robot controllers.

The application of the generic model for motion control offers the following advantages :

* Enables modelling of different algorithms for trajectory generation and joint control.
* Enables modelling of external sensor feed-back.

It is obvious that a neutral file format for the representation of the generic control model would be useful. Such a neutral file format should cover all applicable control schemas (motion control, force control, adaptive control, etc.). The placement of this neutral file format in the ROPSIM architecture is illustrated with dashed lines in fig. 4.1. The neutral interface for controller models would enable a sharing of controller models with software systems developed especially for controller design (MATLAB, ACSL, SIMULAB etc.). This would acquire appropriate pre and post processors translating to and from the ROPSIM controller model.

The generic model of manipulator kinetics: The generic model of robot arm kinetics is developed on basis of the ISO proposal STEP [2,3,6]for neutral representation of product model data. The STEP approach for developing neutral file formats for product data representation is to formulate all the entities of the product on a general information level using the information modeling language EXPRESS [2](Topical Information Models). Such a general information model is a convenient tool in the development of a physical file format for the representation of the generic model. Once the information model describing the essential information has been defined, the information can be represented in a neutral file format. A neutral file format is a handy way of communicating an information model between different systems. The neutral file format can be thought of as a language with syntax and semantic which can express the identified information.

New entities concerning mass properties of the links, friction in the joints and description of servo motor drives (DC motors, hydraulic motors, gears etc.) has been developed. The definition of new entities was carried out using formal statements in EXPRESS. In this way the information model keeps it's formal framework. On basis of the resulting information model, a physical file format has been defined. An example of a robot model expressed in this format, called NIRO/STEP EXTENDED, is shown in fig 4.6.

The generic model of robot arm kinetics consists of four sub-models : a kinematic model, a geometry model, a dynamic model and a model of the servo motor drives.

Kinematics is the study of motion exclusive the influence of masses of the involved objects and exclusive the forces coursing the motion. The kinematic model of the robot arm carries information of the link shapes, the joint types, and their configuration in the robot arm. The model and algorithms support only lower pairs, but could easily be expanded to higher pairs. The geometric model describes the physical dimensions of all the parts in the robot arm. Dynamics is the study of the relationship between the motion and forces affecting the motion. As a consequence the dynamic model must describe the mass properties of the moving objects (the links) in the robot arm. The model of the servo motor drives includes information of the motors and gears in the robot arm.

```
  STEP; HEADER;
  FILE_IDENTIFIER('niro test robot ',
      '1991-04-09T07:48:16',...
      ('IfS - Control Engineering Institute', ...
      'CATIA (AIX) version 2.3');
  FILE_DESCRIPTION(('Test file'),...);
ENDSEC; DATA;...
#151 = &SCOPE
  #152 = CRTPNT(+2.00E+00,+1.00E+00,-1.00E-01);
  #165 = PLYLOP((#156,#159,#158,#157));
  #166 = FACE($,(#160),$); ...
  #172 = CLSSHL((#166,#167,#168,#169,#170,#171));
ENDSCOPE
FCTBRP(#172,$); ...
#217 = &SCOPE
  #218 = &SCOPE
    #219 = &SCOPE
      #220 = DIRCTN(+1.00E+00,+0.00E+00,+0.00E+00);
      #223 = CRTPNT(+0.00E+00,+0.00E+00,+0.00E+00);
    ENDSCOPE
    AXSPLZ(#223,#222,#220);
    #224 = REVOLUTE_PAIR(0.00,-1.57,0.00); ...
    #230 = PAIR_PLACEMENT_STRUCTURE(#224); ...
    #253 = KINEMATIC_JOINT(#250,#251,#224);
    #255 = KINEMATIC_STRUCTURE((#253,#254),$);
    #262 = !INERTIA((1.0,0.0,0.0),(0.0,2.0,0.0),(0.0,0.0,2.0));
    #263 = !MASS_PROPERTIES(#253,1.0,#244,#262);
  ENDSCOPE
  MECHANISM(#255,#250,#219);
  #256 = &SCOPE ...
  ENDSCOPE
  GROUND(#257,(#1,#151));
ENDSCOPE
KINEMATIC_MODEL(#256,(#218),$,$);
ENDSEC; ENDSTEP;
```

Figure 4.6. *Example of a model of a robot arm expressed in NIRO/STEP Extended.*

Together the kinematic, geometric and dynamic model defines the kinetic model of the robot arm.

The application of the generic model of robot arm kinetics offers the following advantages :

* Enables modelling of the robot arm kinematics, open as well as closed kinematic chains.
* Enables modelling of the robot arm geometry.
* Enables modelling of the robot arm dynamics.
* Enables sharing of kinetic robot arm model data with other software systems.

4.3 Modules

It is in the modules that the robot program interpretation, simulation, and visualization takes place. The different algorithms in the modules work on basis of the models.

Programming module: The programming module is in principle an interpreter for the ICR code. It can be viewed as a function which takes an ICR program as argument and produces a flow of move instructions as output.

An ICR interpreter can be split into two domains. The fist describes the actual state of the interpreter and the second incorporate instructions for state changes. A program written in ICR is a list of statements in which one can jump. When staring an execution of a program one starts out from an initial state. Stepping through the program one statement is executing at a time and each of these statements result in a state change.

The simulation module: As described above the simulation module can be split in two. The simulation of the controller is at the moment based on the programming language C++ with a library for matrix algebra. This makes it possible to represent every controller which can be expressed in matrix algebra. A neutral interface for controller models would enable a sharing of controller models with software systems developed especially for controller design (MATLAB, ACSL, SIMULAB etc.). Such a sharing of controller models is unfortunately not possible today.

The simulation algorithms concerning the physical system is based on a kinetic model of the robot arm. The kinetic model includes all necessary information for simulation of kinematics, dynamics and geometry. The simulation module includes real-time algorithms which works on these models. The simulation of the robot arm dynamics is based on an extended Newton-Euler algorithm [5,12]. This algorithm supports robot arms with open as well as closed kinematic chains. Also selected different actuators (DC, hydraulic, etc.) and gear transmission is supported.

The visualization module: The visualization module works on the internal robot model and is able to visualise every state in the robot model. The visualization can take many forms. The module supports visualization of robot model states as numbers, diagrams, sliders, etc. A high performance 3D animation of the robot movements is also provided by using the robot programming and animation system KISMET. The 3D animation utilizes gouraud shading, transparency, lighting of geometry objects, on-line placement of viewpoint and simultaneously viewing from different viewpoints (multiple windows). In order to utilize the high graphic performance of the animation the software is implemented on a Silicon Graphics work-station (IRIS 80GTB).

5 Results

The main results achieved by development of ROPSIM are the following:

- *ROPSIM is a true CIME component.* ROPSIM is prepared for integration in the CIME concept due to the NIRO/STEP EXTENDED ext. interface for the arm model and the ICR programming interface.
- *The ICR programming interface.* With ICR as the programming interface ROPSIM can execute very advanced robot programs (e.g. translated PASCAL programs) and exchange programs with other programming systems or robots.
- *Portable arm models NIRO/STEP EXTENDED ext.* Arm models can be exchanged and shared with other systems in the CIME system.
- *Generic motion controller structure.* Within the generic motion controller structure a wide range of motion controllers (PID, Independent joint, Multivariabel, Adaptive controllers) can be used and evaluated.
- *Simulation of dynamic behaviour of the total robot system.* Due to the model driven simulation of the arm dynamics, different arm models can easily be simulated and evaluated. The simulation module generates and updates all the state variables in the total robot model.
- *Visualization.* The visualization module can present any state in the total robot model. Transient response, dynamic errors, tracking errors and TCP trajectories, animation of the arm model are all visualization features.

To illustrate one of ROPSIM's functionalities a simulation example of the dynamic behaviour of a two link hydraulic test robot [13] is given. The robot is

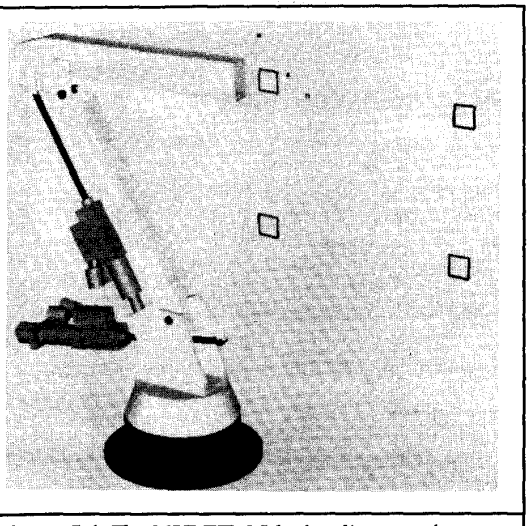

Figure 5.1. *The MULTIDOS hydraulic test robot.*

approximately man size. The robot has been developed at the Control Engineering Institute, Denmark, see Fig 5.1.

A kinematic and geometric model of this MULTIDOS robot has been designed in CATIA and transferred into ROPSIM via the NIRO/STEP EXTENDED format. The controller model is developed in MATLAB and transferred by hand into ROPSIM.

In the example the robot is programmed in ICR to follow a square path at four different locations in the robot work space, see Fig.5.1.The horizontal displacement between the squares is 1.0 meter and the vertical displacement is 0.6 meter. The controller used in this example is simple and consists of two parallel PI-regulators. No optimization of the controller has been carried out. The example has been chosen to illustrate characteristics of ROPSIM rather than the behaviour of a specific controller. The path to be followed by the robot TCP is a 10cm by 10cm square. The programmed velocity is 0.3 m/s. The desired and simulated TCP paths are shown in Fig.5.2.

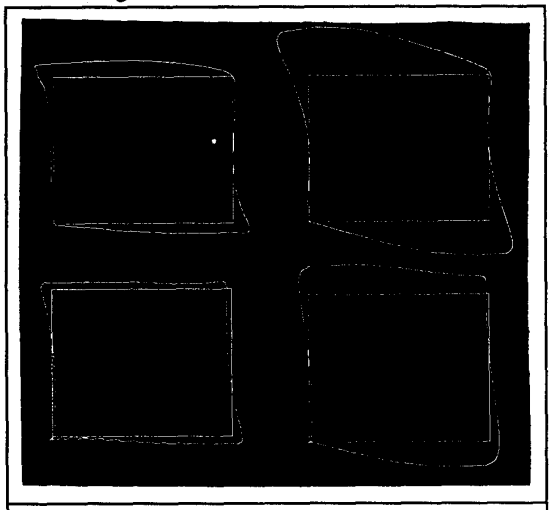

Figure 5.2. *The simulation TCP path errors along four squares.*

As expected the TCP errors vary with the position of the path. Such a simulation can be used to evaluate the total or parts of robot model. On basis of the simulation the different parts of the robot model can be optimized. This can be the controller or the arm mechanism. Also the ICR program can be optimized on basis of the simulation.

6 Conclusion and outlook

A new Robot Off-line Programming and real-time SIMulation system, ROPSIM, which in based on the neutral interface concept and features simulation of dynamics of both controller and robot arm has been developed and implemented at IFS, DTH. Interfaces for programs and kinetic information has been developed.

As the example displays, ROPSIM facilitates new off-line programming opportunities leading to a more accurate simulation.

Also ROPSIM can be used in the design of new robots since both the controller and the physical mechanism can be evaluated and changed. This enables an integrated design phase involving both the robot arm and the controller.

Future development and research willbe concentrated on the following subjects.

First of all a neutral interface for controller models will be developed. Also simulation of the use of external sensor feed-back in the generation of new control data is urgent. Today the internal controller in ROPSIM is prepared for integration of external sensor feed-back. The problem lies in the generation of the correct feed-back signals in a simulation. Also the generic model of the robot arm kinetics and the simulation algorithms can be enlarged in order to provide simulation of elasticity in the links and friction in the joints.

7 Acknowledgement

The development and implementation of the Robot Off-line Programming and SIMualation system ROPSIM at the Control Engineering Institute at the Technical University of Denmark is financed primarily by the Danish Technical Research Council (STVF) which is gratefully acknowledged. Many people have supported this development, mainly project partners and staff members in the EEC ESPRIT project NIRO Project No. 5109. Also achievements [10] of this project have been fundamental to ROPSIM.

8 References

[1] Fu,K.S., Gonzalez,R.C., And Lee,C.S.G. Robotics. Control, Sensing, Vision and Intelligence. McGraw-Hill, 1987.

[2] ISO 10301-11. Industrial Automation Systems. Product Model Data Representation and Exchange, Part 11 : The EXPRESS Language, 1990.

[3] ISO 10301-105. Industrial Automation Systems. Product Model Data Representation and Exchange, Part 105 : Kinematics, St. Louis 1990.

[4] ISO 10562-1. ICR - Intermediate Code for Robots, 1989.

[5] Kleinfinger,J.F, Khalil,W. Dynamic Modelling of Closed-Loop Robots. Proc. 16th Int. Symp. Industrial Robots (ISIR), pp 401-412, Brussels, Belgium, 1986.

[6] Krozynski,U.I, Sorensen,T., Schlechtendahl,E.G. NIRO-STEP Specification for the Transfer of Robotic Models, Ver. L01V03 February 91. Technical University of Denmark and Kernforschungszentrum Karlsruhe (KfK).

[7] Krozynski,U.I, Sørensen,T., Clausen,T.G., Trostmann,E. Driving Robots via Neutral Interfaces. Presented at the ESPRIT Conference, Nov 25-29,1991, Brussels, Belgium.

[8] Paul,R.P. Robot Manipulators. Mathematics, Programming and Control. The MIT Press, 1981.

[9] Trostmann,E. CAD Data Interfaces for Robot Control. Proc. of 2. Duisburger Kolloquium. Automation und Robotic, pp 73-96, 1987.

[10] Trostmann,E. Robotics Research Within ESPRIT : NIRO. Proc. 21st International Symposium on Industrial Robots (ISIR), pp 377-382, Copenhagen, Denmark, 1991.

[11] Trostmann,E. Intelligent Interfaces in Robotics. Proc. of 1st EURO-CIM-Conference "Mechatronics and Robots", pp 231-239 Aachen, Germany, 1991.

[12] Walker,M.W, Orin,D.E. Efficient Dynamic Computer Simulation of Robot Mechanisms. Journal of Dynamic Systems, Measurement, and Control. Vol. 104, pp.205-211, 1982.

[13] Conrad F., Sørensen P. H., Trostmann E., and Zhou,J.J. On Mechanical Design and Digital Adaptive Control of the Fast TUD-Hydraulic Robot Manipulator. Proc. of the 1991 ASME Winter Annual Meeting, Fluid Power Symposium, Atlanta, U.S.A.,1-6 December.