MATLAB DESIGN ENVIRONMENT FOR ROBOTIC MANIPULATORS

Alexander Breijs, Ben Klaassens, Robert Babuška

Delft Center for Systems and Control, Delft University of Technology
Mekelweg 2, 2628 CD Delft, The Netherlands
e-mail:r.babuska@dcsc.tudelft.nl

Abstract: An automated modelling and control design environment for serial manipulators has been implemented in Matlab/Simulink. This development was motivated by the need for a fast and insightful modelling tool, given that currently available modelling environments are not well suited for control design. The manipulator configuration is defined within a graphical user interface and the corresponding mathematical model is automatically generated. The model is exported to Matlab for analysis and control design, as well as to Simulink for simulation and verification purposes. Friction and stiction phenomena are included in the model. The simulation results can be visualized by standard Matlab means as well as through virtual reality animations. The modelling environment has been used in the design of a control system for a seven-degree-of-freedom manipulator in a tunnel-boring machine. *Copyright © 2005 IFAC*

Keywords: Rapid prototyping, Serial manipulator, Graphical user interface, Virtual reality, Matlab, Simulink, Robot.

1. INTRODUCTION

The design of hardware and the corresponding control system of robotic manipulators is often done simultaneously. Control requirements typically influence the manipulator structure and vice versa, thereby subjecting the model to frequent changes. This implies the need for a modelling environment in which the manipulator configuration and the corresponding mathematical model can easily be adapted. Such an environment has been developed at the Delft Center for Systems and Control in cooperation with a Dutch company IHC Systems. It is primarily designed for serial manipulators, powered by hydraulic or ideal torque actuators. The functions have been implemented in Matlab/Simulink and are seamlessly integrated with the standard tools such as the Control Systems Toolbox or Stateflow, see Fig. 1.

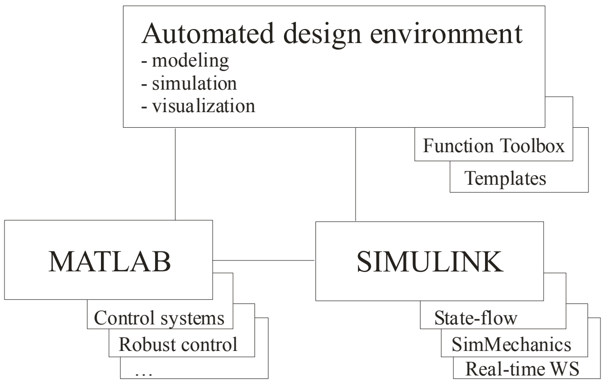


Figure 1. The design environment.

To our knowledge, the only advanced robot modelling environment available under Matlab/Simulink is the SimMechanics toolbox. It provides a wide variety of general mechanical structures, configurable joints, bodies and actuator, sensor and constraint blocks. The constructed model is simulated by means of a special solver and the results can be visualised by using signal plotting or virtual reality (VR) visualization. The main disadvantage of SimMechanics is that it cannot simulate closed-loops control schemes based on inverse dynamic and kinematic models. It was mainly this drawback that motivated the development of the new modelling environment described in this article. Table 1 gives a brief comparison of the most important features of the constructed GUI environment and SimMechanics.

Table 1. A comparison of SimMechanics and the GUI environment described in this article.

	SimMechanics	**GUI environment**
1	Special numerical solver	Standard Simulink solvers
2	Serial and parallel manipulator modelling capabilities	Serial manipulator modelling capability
3	Graphical model representation	Graphical and analytical model representation
4	Signal plotting and VR visualization of the results.	Comparing signal plotting and VR visualization of the results.

2. ROBOT CONFIGURATION

The class of considered manipulators consists of a base attached to the fixed world and an end-effector connected to the base through a number of joints and body combinations. Fig. 2 shows an example of two bodies connected by a rotational joint.

The reference frame $O_0(xyz)$ is located in the base. A joint can be of a rotational or prismatic type with its degree of freedom along the x, y or z axis of the base reference frame. The geometrical centre of joint q_j is the origin of reference frame $O_{qj}(xyz)$. Body B_j holds a reference frame $O_{bj}(xyz)$, located in its centre of gravity (COG). All geometric properties are stored in a Matlab structure containing two 4-dimensional homogenous matrices $\mathbf{H}_{q_j}^{q_{j+1}}$ and $\mathbf{H}_{q_j}^{b_j}$ defined by:

$$\mathbf{H}_{q_j}^{q_{j+1}} = \begin{bmatrix} \mathbf{R}_{q_j}^{q_{j+1}} & \mathbf{c}_{q_j}^{q_{j+1}} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} 1 & l_{b_j}\mathbf{e} \\ \mathbf{0} & 1 \end{bmatrix}$$
$$\mathbf{H}_{q_j}^{b_j} = \begin{bmatrix} \mathbf{R}_{q_j}^{q_{j+1}} & \mathbf{c}_{q_j}^{q_{j+1}} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} 1 & \mathbf{c}_{q_j}^{b_j} \\ \mathbf{0} & 1 \end{bmatrix} \qquad (1)$$

where $\mathbf{c}_{q_j}^{q_{j+1}}$ and $\mathbf{R}_{q_j}^{q_{j+1}}$ define the prismatic or rotational joint geometry between q_j and q_{j+1}. The binary value l_{b_j} times the unit vector $\mathbf{e}$ (defined in the base frame) specifies whether a connecting body B_j is present or absent. This gives the freedom to construct a multiple DOF joint with prismatic and/or rotational features.

The dynamics of the manipulator are modelled according to the standard equation:

$$\mathbf{T} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{F}_v\dot{\mathbf{q}} + e\mathbf{M}_D(\mathbf{q})\ddot{\mathbf{q}} \qquad (2)$$

Where $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$, $\mathbf{F}_v$ and $\mathbf{q}$ are the Coriolis/centrifugal matrix, joint viscous friction and joint position vectors, respectively. Vector $\mathbf{G}(\mathbf{q})$ accounts for the gravitational acceleration and $\mathbf{M}(\mathbf{q})$ is the mass/inertia matrix defined by the following matrix mapping between the base frame and end-effector

$$\mathbf{M} = \left[m_{b_1}\mathbf{J}_{b_1}^0\mathbf{J}_{b_1}^{0\,T} + \cdots + m_{b_E}\mathbf{J}_{b_E}^0\mathbf{J}_{b_E}^{0\,T}\right]$$
$$+\left[\mathbf{J}_{b_1}^{0\,T}\mathbf{R}_{b_1}^0\mathbf{I}\ \mathbf{R}_0^{b_1}\mathbf{J}_{b_1}^0 + \cdots + \mathbf{J}_{b_E}^{0\,T}\mathbf{R}_{b_E}^0\mathbf{I}\ \mathbf{R}_0^{b_E}\mathbf{J}_{b_E}^0\right] \qquad (3)$$

where the binary value m_{b_j} determines the presence or absence of a connecting body mass, thereby giving the freedom to reduce the model complexity. It should be noted that m_b is not related to l_b defined in (1). The Jacobian between the base frame and connecting body B_j

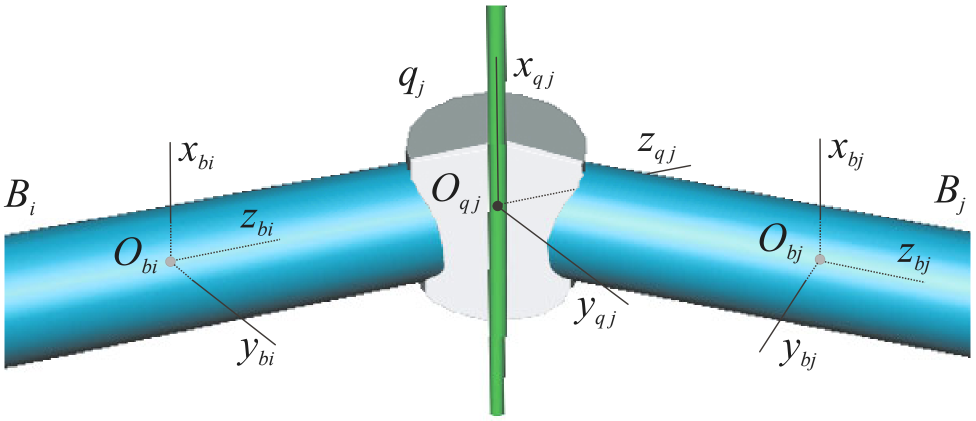


Figure 2. A virtual-reality configuration of two bodies connected by a rotational joint.

is given by $\mathbf{J}_{b_j}^0$ and $\mathbf{I}_{b_j}$ is the inertia with respect to the frame $O_{bj}(xyz)$. The term $e\mathbf{M}_D(\mathbf{q})\ddot{\mathbf{q}}$ accounts for the coupling effects in the presence of friction. The results of equations (1)-(3) are automatically generated as symbolic matrices and vectors stored in a workspace, such that they can be accessed by the GUI, other functions in Matlab and from Simulink blocks.

3. ACTUATOR AND FRICTION MODELLING

The environment has two actuator models which can power the manipulator joints. Depending on the joint type, a rotational or translational torque generator can be chosen, which outputs the demanded torque or force onto the joint.
As most heavy-duty industrial manipulators are 'powered by hydraulics, the other choice is a linear hydraulic actuator augmented with a friction model. This hydraulic actuator applies a force, resulting from the differential pressure ΔP described by expressions for the oil flow Q

$$\begin{aligned} &\frac{V}{E}\Delta\dot{P} + L_e\Delta P + A\dot{x} = Q \\ &Q + \tau\dot{Q} = i_v C\sqrt{1-\frac{|\Delta P|}{P_s}} \end{aligned} \tag{4}$$

ΔP is a result of the pressures difference in the two cylinder chambers of an asymmetric hydraulic actuator with an asymmetric valve. In equation (4) the general piston area A times the piston speed $\dot{x}$ defines the oil flow as a result of piston movement. Furthermore, the leakage flow is given by the leakage coefficient L_e. With P_s, C, i_v and τ as the pump pressure, valve gain, valve current and valve time constant. It should again be noted that all results are also available in Matlab/Simulink, therefore giving the possibility to implement any actuator model.

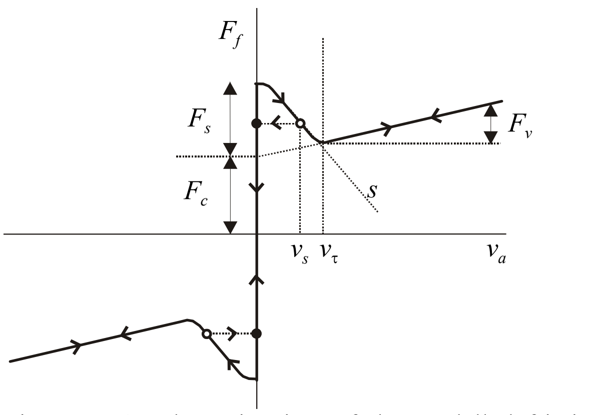


Figure 3. A schematic view of the modelled friction force F_f.

The Lund-Grenoble dynamical friction model (the LuGre model) (Canudas *et al.* 1995) is used. It includes effects like stiction, viscous and Coulomb friction, quantified by means of damping and stiffness coefficients. As these

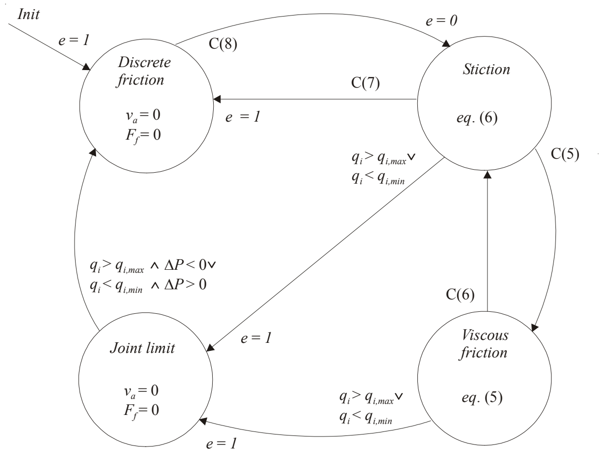


Figure 4. Hybrid friction automaton describing the function visualized in Fig. 3.

coefficients can substantially differ in their relative values, a computationally demanding computation of stiff differential equation is generally required.
Therefore, a model the form of a hybrid automaton is proposed to overcome the computational issues. This is accomplished by the introduction of discrete signals and states. Furthermore a hybrid automaton gives a clear visual insight into the friction model. The division between coulomb friction F_c and a discrete part of stiction F_s opposite to the continuous viscous friction F_v and Stribeck effect (Fig. 3) results in the hybrid approach.

The friction function shown in Fig. 3 can be described according to the following equations, describing the three friction areas.

$$F_f = F_v v_a \quad v_a \geq v_\tau \quad \vee \quad v_a \leq -v_\tau \tag{5}$$

$$F_f = e^{-\left(\frac{v_a}{v_s}\right)^2} \quad 0 \leq v_a < v_\tau \quad \vee \quad -v_\tau < v_a \leq 0 \tag{6}$$

$$\begin{aligned} v_k = 0 \quad &\left(0 \leq v_a < v_s \wedge \dot{v}_a \leq 0\right) \\ &\vee\left(-v_s < v_a \leq 0 \wedge \dot{v}_a \geq 0\right) \end{aligned} \tag{7}$$

$$v_k = v_a \quad F_f > F_c + F_s \vee F_f < -F_c - F_s \tag{8}$$

Equation (5) defines the viscous friction F_v in Fig. 3, where v_a is the actuator speed. Equation (6) describes the continuous part of stiction, defined in (Canudas *et al.* 1995)] as the Stribeck effect, with v_s as the Stribeck speed. The speed defines slope s in Fig. 3. Equations (7) and (8) describe the discrete part of friction in the form of coulomb friction F_c and the discrete part of stiction F_s.
The above division has a side effect in the sense that the joint position limitations can be easily added. The introduction of an additional discrete state results into position limit functionality. As the joint limits are not the main focus, the dynamical properties are neglected. A hybrid automaton can now be constructed (see Fig. 4).

$$\begin{tikzcd} c_i \arrow[r] & \boxed{\begin{matrix}\text{Actuator } A_i \\ Eq.(4)\end{matrix}} \arrow[r, "F_i", "v_i"'] & \boxed{\begin{matrix}\text{Domain} \\ \text{transformation } T_i\end{matrix}} \arrow[r, "T_i", "\omega_i"'] & \boxed{\text{Structure } S_i} \end{tikzcd}$$

Figure 5. Schematic template SIMULINK plant model used in the automated modeling environment.

In this figure, e equals the discrete actuator speed reset signal, with $e = 0$ resulting in $v_a = 0$ and $\dot{v}_a = 0$ and $e = 1$ leading to $v_a = v_a$ and $\dot{v}_a = \dot{v}_a$. The conditions of equations (5), (6), (7) and (8) are given by C(5), C(6), C(7) and C(8). The maximum and minimum joint position of joint q_i equal $q_{i,max}$ and $q_{i,min}$, with ΔP as the differential actuator pressure described in equation (4) (Johansson *et al.* 2000)

4. MODEL SIMULATION AND VISUALIZATION

The functions of the environment are controlled via a Graphical User Interface (GUI) consistsing of three main parts: manipulator configuration, simulation and visualization. All computational results are transformed into Simulink s-functions. These results are then used in a template Simulink plant model, schematically shown in Fig. 5.

The parameters are listed in a Matlab file with accompanying default values, with the exception of the end-effector mass M_e, the hydraulic leakage L_e and the joint viscous friction F_{vj}. Values for these three parameters can be chosen in the GUI as they greatly define the overall system behaviour. Furthermore the actuator joint friction can be defined.
Together with the definable elementary step, impulse

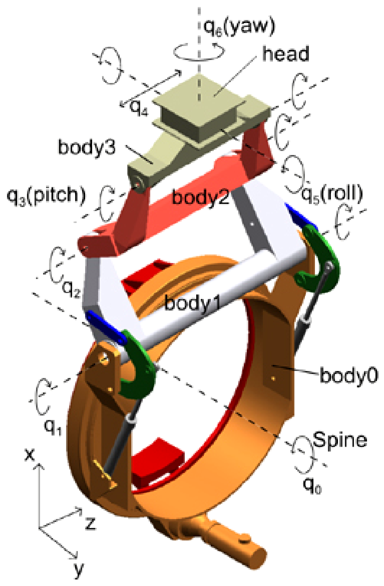
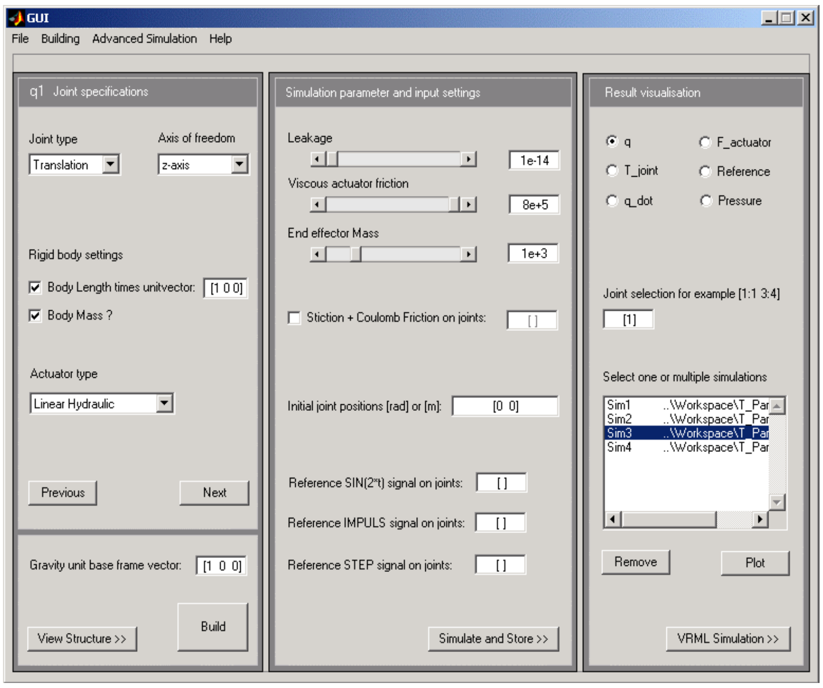


Figure 7. Schematic view of the 7 DOF erector.

and sine control signals c_i, a Simulink simulation model is created. Multiple simulation results with different parameter settings can be stored in the environmental workspace. This gives the ability to easily compare the results and obtain a quick insight in the model. As all results are also available in Matlab/Simulink, the model can be easily expanded or the default parameter settings


GUI

File Building Advanced Simulation Help

q1 Joint specifications

Joint type: Translation
Axis of freedom: z-axis

Rigid body settings
☑ Body Length times unitvector: [1 0 0]
☑ Body Mass ?

Actuator type: Linear Hydraulic

Previous | Next

Gravity unit base frame vector: [1 0 0]

View Structure >> | Build

Simulation parameter and input settings

Leakage: 1e-14
Viscous actuator friction: 8e+5
End effector Mass: 1e+3

☐ Stiction + Coulomb Friction on joints: [ ]

Initial joint positions [rad] or [m]: [0 0]

Reference SIN(2*t) signal on joints: [ ]
Reference IMPULS signal on joints: [ ]
Reference STEP signal on joints: [ ]

Simulate and Store >>

Result visualisation

(●) q ( ) F_actuator
( ) T_joint ( ) Reference
( ) q_dot ( ) Pressure

Joint selection for example [1:1 3:4]: [1]

Select one or multiple simulations
Sim1 ..\Workspace\T_Par
Sim2 ..\Workspace\T_Par
Sim3 ..\Workspace\T_Par
Sim4 ..\Workspace\T_Par

Remove | Plot

VRML Simulation >>

Figure 6. Screen-shot of the implemented GUI.

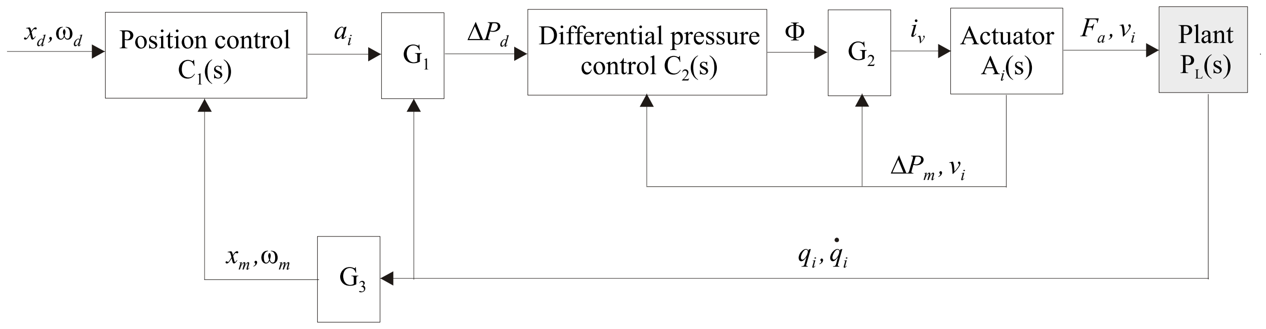


Figure 8. Closed-loop block scheme of cascaded delta P controller in combination with GUI plant model of figure 5.

can be altered thereby matching a particularly situation more closely.
The final step in the GUI contains the simulation result visualization abilities. Next to the possibility to plot numerous signals of different simulations into one plot figure, a Virtual Reality (VR) option exists. Each building block available in the modelling part is defined in the VR template file. On demand a VR simulation is created of the modelled robotic structure with accompanying transient position data.

5. MANIPULATOR CONTROL DESIGN

This section describes a modelling example that uses the automated modelling environment, in order to reach the control design for a manipulator.

5.1 Manipulator modelling for a TBM manipulator

The proposed modelling environment has been used in the design of a manipulator (erector) for a novel shield tunnel-boring machine (TBM), see Fig. 7. The task of the erector is to place steel segments such that they form the tunnel lining (Braaksma *et al.* 2004). The erector has in total seven DOFs (q_0 to q_6). The first DOF is neglected, as it is fixed during the placement. Body 3 (the last body in the chain) holds a complex joint which consist of a roll, yaw and translation in the *z*-direction (see Fig. 7). All joints are hydraulically actuated. The GUI is shown in Fig. 6.

Table 2. GUI erector specification, with R meaning rotation, Tr translation, Nn not needed.

GUI model option	Erector value sets
DOF	{z, z, z, z, y, x}-axis
Joint Type	{R, R, R, Tr, R, R}
Body length l_b	{1, 1, 0, 0, 0, 1}
Unit vector e	{[1 0 0], [1 0 0], Nn, Nn, Nn, [1 0 0]}
Body mass m_b	{1, 1, 0, 0, 0, 1}

The first step is to specify the above-described example in the GUI. As can be seen from Fig. 6 the GUI consists of three main windows: matching modelling, simulation and visualization. Table 2 shows the results according to the options listed in the GUI of the example. With the specification of the gravitational acceleration vector *g,* a plant model is constructed, which will be expanded with a Simulink control scheme. The resulting closed-loop model will then again be simulated and visualized in the GUI of Fig. 6.

5.2 Control design

The control scheme holds three main blocks: Cartesian position control, joint space pressure control and feedback linearisation. The expansion of the plant model shown in Fig. 5 with the above control scheme is shown in Fig. 8.
Both position and pressure control are based on a Proportional-Integral-Differential control, law (PID):

$$a_x(s) = K_{po}x_e + K_{do}x_e s + K_{io}x_e / s \tag{9}$$

$$\Phi(s) = K_{pi}\Delta P_e + K_{di}v_e + K_{ii}\Delta P_e / \mathrm{s} \tag{10}$$

with a_x as the position control acceleration and Φ as the pressure control oil flow. Gains K_{po}, K_{do} and K_{io} equal the proportional, differential and integral outer loop position control values, with x_e as the position error. Furthermore K_{pi}, K_{di} and K_{ii} define the inner loop pressure PID control values with ΔP_e as the differential pressure error and v_e as the speed error.

Block G_3 holds the kinematics to transform the joint space sensor signals into their Cartesian counterparts. The feedback linearisation is defined in G_1 and G_2. G_1 equals the matrices in (2) together with the domain transformation, which result in a desired differential pressure ΔP_d. G_2 holds the non-linearity of the valve and flow compensations (4) and transforms the control oil flow into a control valve current i_v.

6. CONCLUSIONS

An automated modelling environment has been implemented to facilitate the simultaneous design of the configuration and the corresponding control system of robotic manipulators. Via a user-friendly graphical interface, one can easily define physical parameters of this manipulator. Detailed knowledge of the modelling formalism is not required. Insight into the static and dynamic properties of the model can be obtained through the inspection of simulation results (using an automatically generated Simulink model) as well as by analysing the model in Matlab.

REFERENCES

Braaksma, J., C. de Keizer, J.B. Klaassens, R. Babuška (2004). *Hybrid Control Design for a Manipulator in a Shield Tunneling Machine*. Proceedings ICINCO 2004 IFAC conference. Setubal, Portugal, pp. 185-192.

Canudas de Wit, C., H. Olsson, K.J. Astrom, P. Lischinksy (1995). *A new model for control of systems with friction*. IEEE Transcript Automatic Control, Volume 40, No 3, pp.419-425.

Heintze, H. (1997), *Design and Control of a Hydraulically Actuated Industrial Brick Laying Robot*. PhD-thesis, Delft University of Technology, Dept. of Mechanical Engineering. ISBN 90-370-0156-4.

Johansson, K.H., J. Lygeros, J. Zhang, S. Sastry (2000). *Hybrid automata: A Formal Paradigm for Heterogeneous Modelling*, Proceedings of the 2000 IEEE International Symposium on Computer Aided Control System Design, Anchorage, Alaska..