

## MODELING AND IDENTIFICATION OF A MECHANICAL INDUSTRIAL MANIPULATOR <sup>1</sup>

M. Norrlöf \* F. Tjärnström \* M. Östring \* M. Aberger \*\*

\* Department of Electrical Engineering, Linköpings universitet,  
SE-581 83 Linköping, Sweden  
Fax: +46-13-282622 Email: {mino, fredrikt, mans}@isy.liu.se

\*\* Department of Design and Control of Mechatronic Systems, Johannes Kepler  
University Linz  
A-4040 Linz, Austria  
Email: martin.aberger@students.uni-linz.ac.at

**Abstract:** This paper covers modeling and identification of one joint of an industrial robot manipulator including flexibilities. It is shown how models can be built in the Modelica graphical environment and how these models can be transformed into a mathematical state space description which directly can be used for identification. A motivation to use linear models for modeling of the robot arm is given. This includes an analysis of the nonlinearities in the input/output data from the actual robot using a special kind of input signal. Identification and validation of the physically parameterized models are also covered.

**Keywords:** Robot dynamics, system identification, mechanical systems, modeling, industrial robots, closed-loop identification

### 1. INTRODUCTION

In this paper a number of approaches to modeling and identification of one joint of a classical robot manipulator is presented. There are two major contributions in the paper. The first is to use special odd multi-sine signals as input signals to the system. This gives the possibility to analyze the linearity of the system. With this input signal, dynamics are also found that were not found using a chirp input signal, compare e.g., (Östring *et al.*, 2001). The second contribution is to show how modern mathematical tools can be used to easily construct a physical model, e.g., in a graphical environment. From this model a mathematical description can be found and as a last step this description is transformed into a form that can be used for identification. The tools that will be used are, apart from classical mechanics, the System Identification Toolbox (Ljung, 2000) for Matlab and the newly developed MathModelica software tool (Jirstrand, 2000)

from MathCore. MathModelica is an extension to Mathematica (Wolfram, 1999) and implements the Modelica (Modelica Association, 2000; Tiller, 2001) standard for modeling and it also gives the possibility to use the powerful mathematical capabilities of Mathematica to derive, for example, a state space description for a mechanical system from a Modelica model.

From an industrial perspective, modeling and identification of industrial manipulators is becoming more and more important. In the competition of consumers, that is present today on the robot market, the price and performance issues are essential. In order to reduce price but still keep (or increase) the performance, a more light robot structure is a tempting possibility. Making this step will also reduce the rigidity of the structure, which means that good models for control will be even more important.

---

<sup>1</sup> This work was supported by ISIS and ERNSI at Linköpings universitet.

## 2. THE PROCESS

This section gives an overview of the physical system used in the identification experiments. A picture of the manipulator is shown in Figure 1. It has 6 degrees of freedom, but we will in this paper only model the motion at axis 1.

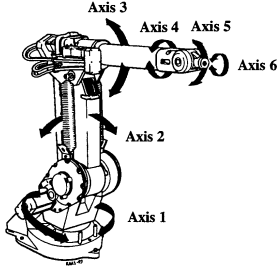


Fig. 1. The IRB1400 industrial robot from ABB used in the identification experiments.

The robot is a commercial robot with a commercial control system. This means, among other things, that the controller cannot be altered in any major way and the actual implementation cannot be assumed to be known. The structure of the robot system is depicted in Figure 2. It is only possible to directly affect the reference signal,  $\theta_{ref}$ . The torque,  $u$ , generated by the electrical motor is affected only indirectly by the feedback control system.

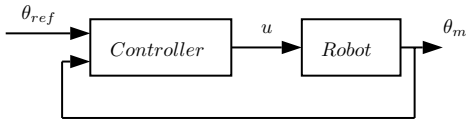


Fig. 2. Block diagram of the robot.

In the experiments the reference signal,  $\theta_{ref}$ , and the torque reference from the controller,  $u$ , are measured together with the angle of the motor driving the robot arm,  $\theta_m$ . The sampling interval is 0.5 ms. An electrical motor drive the joint and the gear-box. The motor contains in itself a fast controller loop to maintain the desired torque. The goal is to identify a model of the robot by using  $u$  as input (the torque reference), and  $\theta_m$  as output (measured motor angle).

## 3. MOTIVATION OF A LINEAR MODEL

Although considering only one joint, the manipulator clearly contains quite a few nonlinear elements, e.g., backlash, stiction, and saturation. The different nonlinearities can roughly be divided into two main groups; those whose influence decrease when the input amplitude decreases (e.g., saturations and polynomial gains), and those whose nonlinear influence increase when the input amplitude decreases (e.g., relays, dead zones, backlashes). The dominating nonlinearities in the manipulator, backlash and stiction,

are of the second kind. It is possible to show that the nonlinearities that decrease with decreasing input amplitude actually are negligible using the methods described in (Pintelon and Schoukens, 2001, Chapter 3). Let us consider nonlinearities that can be described by a series expansion around the operating region. When the input amplitude decreases, the linear term will be dominating over the higher order terms.

To actually show that the effect of this class of nonlinearities is small in the manipulator an input consisting of a sum of sinusoids is constructed,

$$u(t) = \sum_{k=1}^n A_k \sin(2\pi f_k t + \phi_k). \quad (1)$$

If this input is fed through a nonlinear element, say  $(\cdot)^m$ , the output will consist of frequencies

$$\tilde{f}_k = f_{i_1} \pm f_{i_2} \pm \dots \pm f_{i_m}, \quad (2)$$

where

$$i_j \in \{1, 2, \dots, n\}, \quad j = 1, \dots, m. \quad (3)$$

Now the frequencies  $f_1, \dots, f_n$  can be chosen in many different ways. One popular choice is the so called *special odd multisine* (Pintelon and Schoukens, 2001, Chapter 3)

$$f_1 = f_0, f_2 = 3f_0, f_3 = 9f_0, \dots, f_n = Ff_0, \quad (4)$$

where  $f_n = Ff_0$  is the highest frequency chosen. (In Matlab-like notation this would be  $[1 : 8 : F] \cup [3 : 8 : F]$ ). The frequencies showing up at even multiples of  $f_0$  and of the non-used odd multiples of  $f_0$  would originate from the type of nonlinearities considered here. The choice of phases,  $\phi_k$ , is of less importance for this application, but should be chosen with care in other situations, see (Pintelon and Schoukens, 2001, Chapter 3).

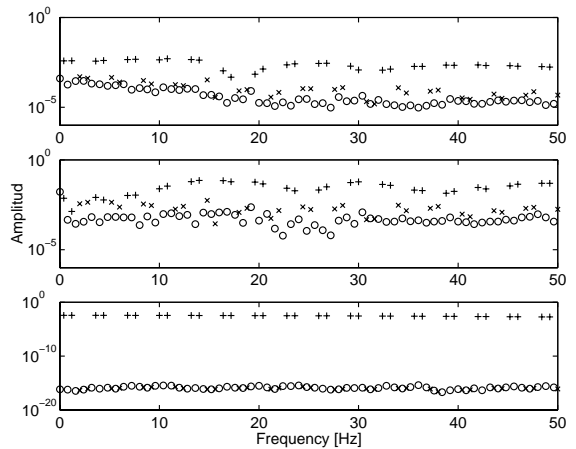


Fig. 3. Results from an experiment with a sum of sinusoids as reference. Bottom - reference ( $\theta_{ref}$ ) amplitudes. Middle - input ( $u$ ) amplitudes. Top - output ( $\theta_m$ ) amplitudes. + - excited frequencies. x - odd frequencies. o - even frequencies.

In the experiment performed on the manipulator a reference signal consisting of a sum of 76 sinusoids,

with  $f_0 = 0.4\text{Hz}$  and  $f_n = 121.6\text{Hz}$ , is chosen. The amplitudes are picked so that they follow the amplitudes of a Butterworth filter of order 1 with cut off frequency at  $50\text{Hz}$ . The phases are uniformly distributed between 0 and  $2\pi$  (which is a typical choice). The length of the experiment was 9.5 s, where the first 2 s were assumed to be transients and the remaining 7.5 s were used to see the results from 3 periods of the sinusoids (3 full periods of the lowest frequency). The sampling interval was 4 ms.

The resulting amplitudes of the reference, the input, and the output are shown in Figure 3. The amplitudes are averaged over the 3 periods, to reduce the influence of noise. From these plots it follows that the amplitudes of the overtones in  $u$  and  $\theta_m$  are very low, actually about a factor  $1/100$  compared to the excited frequencies. This confirms the statement that these type of nonlinearities have very little influence in the robot system. Also note that, due to the regulator, very little input energy is found below  $10\text{Hz}$ . This will cause large uncertainties in the estimated models in this frequency range.

## 4. MODELING

### 4.1 *MathModelica*

Modelica is a standardized modeling language (see (Modelica Association, 2000)) but it does not include analysis and simulation capabilities. For this purpose MathModelica (Jirstrand, 2000) has been created. MathModelica is an add-on to Mathematica (Wolfram, 1999). It implements the Modelica standard and gives the possibility to manipulate the equations in the model using Mathematica. This will be explored later on in the paper. Using the Dymola engine, MathModelica also provides simulation capabilities. Within Modelica a number of standard model components exist. This includes components for modeling of mechanical systems as well as electrical systems. In this paper the former libraries are used.

### 4.2 *Physical model*

The work in this paper is restricted to modeling of one joint of the industrial robot. The model includes the gear box and the robot arm. The model explains how the torque applied to the gear box affects the angle of the motor and the arm. The controller of current in the electrical motor is neglected and the torque reference to the motor is considered to be the actual torque applied to the system.

The robot arm can be modeled using different levels of approximation. A first approximation could be to assume that the arm is stiff and acts as a rigid structure. In a robot having gear boxes this is a rough approximation. Modeling the gear box as a spring coupling

two masses gives a more accurate model. If the robot is moving fast the robot arm cannot be assumed to be stiff. This makes it reasonable to use at least a three-mass flexible model to describe the system.

The notations regarding the physical modeling of the three-mass model in this paper can be seen in Table 1. The friction is modeled as a viscous friction acting on the motor of the robot.

Table 1. Notations.

$\theta_m$	motor angle
$\theta_{a1}, \theta_{a2}$	arm angle
$f_m$	friction coefficient of the motor
$k_g, k_a$	spring constants
$d_g, d_a$	damping coefficients in the springs
$J_m$	moment of inertia of the motor
$J_{a1}, J_{a2}$	moment of inertia of the first and the second part of the arm
$\tau$	motor torque
$r$	gear box ratio ( $\frac{1}{118}$ )

### 4.3 *Generation of the physically parameterized models*

In Figure 4 the Modelica implementation of the three mass and the four mass model is shown. This section describes the generation of the equations for the three mass model from the graphical representation. The four mass model is generated in a similar way.

From the graphical description of the system in Figure 4 it is possible to make a simulation using the MathModelica (Jirstrand, 2000) environment. This gives a simulated model with 6 continuous states. By directly taking the number of states from the Modelica model, using the MathModelica command `GetFlatStateVariables[ThreeMM]`, a total of 9 states are found. The difference in the number of states comes from the fact that the different sub-models are modeled individually and when they are interconnected it is possible to reduce the total number of states. In Mathematica, using MathModelica, the equations describing the three mass system can be found using `GetFlatEquations[ThreeMM]`. It results in 55 equations. Many are trivial, e.g., saying that the angular velocity on one edge of a component equals the angular velocity of the connecting edge of the next component.

The trivial equations can be removed easily in Mathematica using `Eliminate[eqs,list]` where `eqs` are the equations coming from the Modelica description and `list` is a list of the variables that should be eliminated. The result from this step is that the number of equations are reduced from 55 to 6. In the current version of MathModelica the list of variables to eliminate must be found by hand.

After this step it is also necessary to find and replace some state variables that could be expressed as functions of other state variables. For the three mass model this includes the spring damper components where the relative angular position between the two connections

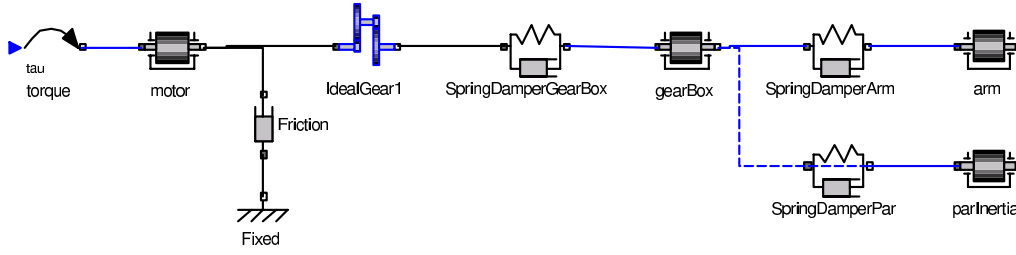


Fig. 4. The Modelica model used to derive the state space models of the mechanical system. The extension from the three mass model to the four mass model is shown as a dashed connection.

becomes a state variable. These two variables are replaced by  $\theta_{a1} - \theta_m r$  and  $\theta_{a2} - \theta_{a1}$ , respectively.

The next step is to introduce the state variables,  $x_1$  to  $x_5$ . For the three mass model described here they are given by

$$\begin{aligned} x_1 &= \theta_m r - \theta_{a1}, & x_2 &= \theta_{a1} - \theta_{a2}, \\ x_3 &= \dot{\theta}_m, & x_4 &= \dot{\theta}_{a1}, & x_5 &= \dot{\theta}_{a2} \end{aligned} \quad (5)$$

For Mathematica to be able to solve the equations it is also necessary to include the time derivative of the state variables among the equations added to the 6 equations found above. This step is done with the command `Join[list1,list2]` in Mathematica. `Solve[eqs, {x1', x2', x3', x4', x5'}]` finally gives the state equations for the three mass model in a closed form.

The state space description of the three mass model from Mathematica becomes

$$\begin{aligned} A &= \begin{pmatrix} 0 & 0 & r & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -\frac{k_g r}{J_m} & 0 & -\frac{f_m + d_g r^2}{J_m} & \frac{d_g r}{J_m} & 0 \\ \frac{k_g}{J_{a1}} & -\frac{k_a}{J_{a1}} & \frac{d_g r}{J_{a1}} & -\frac{d_g + d_a}{J_{a1}} & \frac{d_a}{J_{a1}} \\ 0 & \frac{k_a}{J_{a2}} & 0 & \frac{d_a}{J_{a2}} & -\frac{d_a}{J_{a2}} \end{pmatrix} \\ B &= (0 \ 0 \ \frac{1}{J_m} \ 0 \ 0)^T \\ C &= (0 \ 0 \ 1 \ 0 \ 0) \end{aligned} \quad (6)$$

Output from this model is the angular velocity of the motor. If the angular position is chosen as output the number of states becomes 6 since an extra integrator has to be added. This equals the number of states in the simulation model in MathModelica.

The four mass model with the extra parallel mass according to Figure 4 is found in a similar way. In addition to parameters and variables in Table 1, the extra parameters are  $J_p$ ,  $k_p$ , and  $d_p$  and the extra variable is  $\theta_p$ . The state space model from Mathematica, where  $a_{44} = -\frac{f_m + d_g r^2}{J_m}$  and  $a_{55} = -\frac{d_g + d_a + d_p}{J_{a1}}$ , becomes

$$\begin{aligned} A &= \begin{pmatrix} 0 & 0 & 0 & r & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ -\frac{k_g r}{J_m} & 0 & 0 & a_{44} & \frac{d_g r}{J_m} & 0 & 0 \\ \frac{k_g}{J_{a1}} & -\frac{k_a}{J_{a1}} & -\frac{k_p}{J_{a1}} & \frac{d_g r}{J_{a1}} & a_{55} & \frac{d_a}{J_{a1}} & \frac{d_p}{J_{a1}} \\ 0 & \frac{k_a}{J_{a2}} & 0 & 0 & \frac{d_a}{J_{a2}} & -\frac{d_a}{J_{a2}} & 0 \\ 0 & 0 & \frac{k_p}{J_p} & 0 & \frac{d_p}{J_p} & 0 & -\frac{d_p}{J_p} \end{pmatrix} \\ B &= (0 \ 0 \ 0 \ \frac{1}{J_m} \ 0 \ 0 \ 0)^T \\ C &= (0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0) \end{aligned} \quad (7)$$

The states are chosen as

$$\begin{aligned} x_1 &= \theta_m r - \theta_{a1}, & x_2 &= \theta_{a1} - \theta_{a2}, \\ x_3 &= \theta_{a1} - \theta_p, & x_4 &= \dot{\theta}_m, \\ x_5 &= \dot{\theta}_{a1}, & x_6 &= \dot{\theta}_{a2}, & x_7 &= \dot{\theta}_p \end{aligned} \quad (8)$$

Here  $\dot{\theta}_m$  is picked as output. With the angular position as output,  $\theta_m$ , the number of states becomes 8.

The models in (6) and (7) are symbolically represented in Mathematica and this makes it very easy to reparameterize and scale the parameters. After this step the model description is saved in a text file. This file can then be transformed, using a simple Matlab program, into an m-file (Matlab script-file) that can be used directly by the System Identification Toolbox. Using these tools the modeling to identification process becomes straight-forward and nearly automatic.

## 5. SYSTEM IDENTIFICATION

The aim of this section is not to give an extensive introduction to system identification, the interested reader is instead referred to, for example, (Ljung, 1999). Some important features will however be discussed.

In this application the identification of the flexible robot is performed under strong feedback. The main problem with identification from closed loop data is that the data contain less information about the open loop system. The reason for this is that the purpose of the feedback is to make the closed loop system less sensitive to changes in the open loop system. Another problem that can occur is bias due to inaccurate noise models, see below. The identification is made using the System Identification Toolbox (Ljung, 2000) in Matlab.

In this contribution the focus is on state space models (with state vector  $x(t)$ ), parameterized as either pure black-box models in discrete time, i.e.,

$$\begin{aligned} x(t+1) &= A(\theta)x(t) + B(\theta)u(t) + K(\theta)e(t) \\ y(t) &= C(\theta)x(t) + e(t), \end{aligned} \quad (9)$$

or as physically parameterized models in continuous time, i.e.,

$$\begin{aligned} \dot{x}(t) &= F(\theta)x(t) + G(\theta)u(t) \\ y(t) &= C(\theta)x(t) + v(t). \end{aligned} \quad (10)$$

The conversion from continuous time to discrete time is typically performed using the assumption that the input is constant over the sampling interval  $T_s$ , see (Ljung, 2000, page 94)

$$\begin{aligned} x(t+1) &= A_T(\theta)x(t) + B_T(\theta)u(t) \\ y(t) &= C(\theta)x(t) + v(t), \end{aligned} \quad (11)$$

$$A_T(\theta) = e^{F(\theta)T_s}, \quad (12)$$

$$B_T(\theta) = \int_{t=0}^{T_s} e^{F(\theta)\tau} G(\theta) d\tau \quad (13)$$

These models are estimated by minimizing the sum of the squared prediction errors. For (11) this becomes

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \theta) \quad (14)$$

$$\begin{aligned} \varepsilon(t, \theta) &= y(t) - \hat{y}(t|\theta) \\ &= y(t) - C(\theta)(qI - A_T(\theta))^{-1} B_T(\theta)u(t) \end{aligned} \quad (15)$$

Here  $q$  is the discrete time shift operator, i.e.,  $qu(t) = u(t+1)$ . As the number of data increases the estimate will converge to the true system, assuming that the model class can describe the underlying system and that the noise model is correct. If the noise model is incorrect a slight bias will be achieved (depending on the mismatch). If, however, the noise to signal ratio is small, the bias will also be small. As a matter of fact this is the case for the studied manipulator. This is somewhat implicitly stated in (Norrlöf, 2000, Chapter 7). For the exact expression of the limiting estimate, see for example (Forssell and Ljung, 1999).

One important question to answer when modeling and performing identification is whether or not the model describes the true system in a good way. To answer this question one typically does a number of validation tests, see (Ljung, 2000, Chapter 16). Two types of validation tests will be used in this paper. The first test is based on the fact that a high order model is able to describe any linear system arbitrarily well, if the model order is chosen high enough, see (Ljung, 2000, Chapter 10). This means that the estimated physically parameterized models can be validated against a high order model in the frequency domain. In this way it is possible to judge whether or not the model has picked up some important aspects of the system dynamics. The second test that will be used is cross validation, i.e., the model is simulated on new data and the simulated output,  $y_{sim}(t)$  is compared with the output from

the system,  $y(t)$ . A measure of the cross validation performance is how many percent of the output that is explained by the model, i.e.,

$$FIT = 100 \cdot \frac{\text{Var}(y(t) - y_{sim}(t))}{\text{Var} y(t)}. \quad (16)$$

## 6. EXPERIMENTS

This section describes the identification experiments performed on the manipulator. The parameters of the models (6) and (7) are estimated together with a high order state space model. The excitation of the robot is performed according to the description in Section 2.

For identification and cross validation, two different reference signals were used. The identification was performed using one period of a special odd multisine and the validation was performed using a Gaussian white noise sequence, both having the same variance and frequency content, cf. Section 3. The length of the signals was 7.5 s and the sampling interval was 0.5 ms. Since the sampling interval is rather short compared to the bandwidth of the system a down-sampling of a factor 10 was performed. From the estimation data an 11th order black-box state space model was estimated together with the physically parameterized three and four mass models. The results are depicted in Figures 5 and 6. Clearly the three mass model does not give a full description of the manipulator. However, by increasing the complexity of the model, going to a four mass model structure, the two notches in the region 15-20 Hz can be picked up reasonably well. The 11th order black-box model is the lowest order black-box model that finds the double notch around 17 Hz and this is the reason why this model has been chosen for comparison.

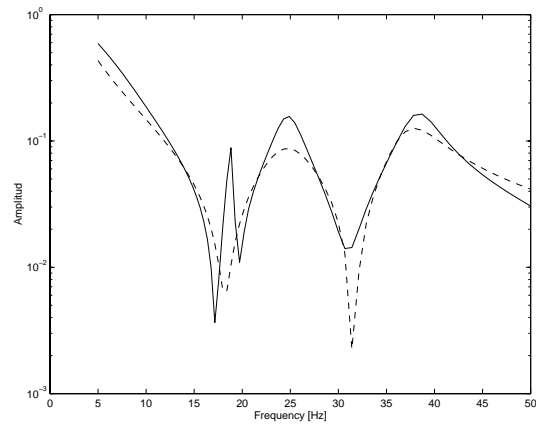


Fig. 5. Solid line - 11th order black box state space model. Dashed line - physically parameterized 3 mass model (6 states).

The fit for these three models on estimation and validation data is summarized in Table 2. Clearly the physically parameterized models do not perform as good as the 11th order black-box model in simulation.

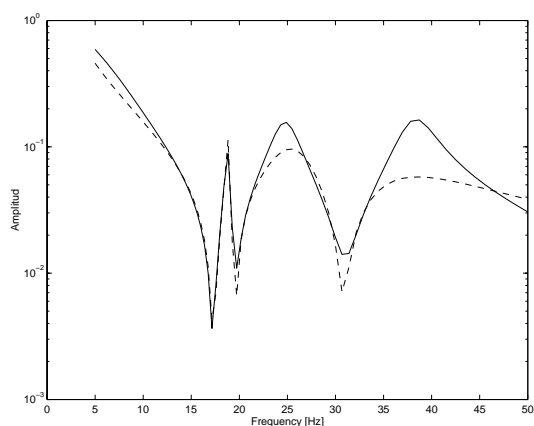


Fig. 6. Solid line - 11th order black box state space model. Dashed line - physically parameterized 4 mass model (8 states).

However, the physically parameterized models contribute with more knowledge about the manipulator, since each parameter and state space variable is interpretable as some (in most cases) known physical fact about the system. To get a even better description, the manipulator must be modeled further, maybe by including yet another mass which should describe better the physical link structure of the arm. The main disadvantage in using a physical description of the model is that the initialization of totally unknown parameters may be very difficult and one might therefore get stuck in a local minima in the search of the optimal estimate.

Table 2. Fit for the three estimated models on estimation and validation data.

Data Set	11th order	3 mass	4 mass
Est. data	74.8	57.8	58.0
Val. data	73.0	56.2	53.6

## 7. CONCLUSIONS

The paper gives a full treatment of the procedure to go from a physical system, via a physical model, to a mathematical model that can be transformed into a form that can be used for identification. The choice of linear models to describe the dynamics of the system is also motivated by doing an experiment using an input signal of special odd multisine type. The process of going from modeling to parameter estimation is highly automated and it is shown how modern mathematical tools such as Modelica, MathModelica, Mathematica, and Matlab can be used to aid the process. A comparison between a high order black-box model and different physically parameterized models is performed. One problem with the latter models is the choice of initial values for the parameters that might be completely unknown.

## REFERENCES

- Forsell, U. and L. Ljung (1999). Closed-loop identification revisited. *Automatica* **35**, 1215–1241.
- Jirstrand, M. (2000). Mathmodelica – a full system simulation tool. In: *Product Models 2000, The Sixth Conference on Product Models, Global Product Development*. Linköping, Sweden.
- Ljung, L. (1999). *System Identification: Theory for the User*. second ed.. Prentice-Hall.
- Ljung, L. (2000). *System Identification Toolbox - For Use with Matlab*. The MathWorks Inc.
- Modelica Association (2000). Modelica – a unified object-oriented language for physical systems modeling: Language specification, version 1.4. <http://www.modelica.org>.
- Norrlöf, M. (2000). Iterative Learning Control: Analysis, Design, and Experiments. PhD thesis. Linköpings universitet. Linköping, Sweden. Linköping Studies in Science and Technology. Dissertations; 653. Download from <http://www.control.isy.liu.se/publications/>.
- Pintelon, R. and J. Schoukens (2001). *System Identification: A Frequency Domain Approach*. IEEE Press and John Wiley & Sons, Inc.
- Östring, M., S. Gunnarsson and M. Norrlöf (2001). Closed loop identification of the physical parameters of an industrial robot. In: *Proceedings of the 32nd ISR(International Symposium on Robotics)*. Seoul, Korea.
- Tiller, Michael M. (2001). *Physical Modeling with Modelica*. Kluwer Academic Publishers.
- Wolfram, S. (1999). *The Mathematica Book*. fourth ed.. Cambridge University Press.