

Simulation Model of a 2 Degrees of Freedom Industrial Manipulator

Patrik Axelsson

Division of Automatic Control

E-mail: axelsson@isy.liu.se

17th June 2011

Report no.: LiTH-ISY-R-3020

Address:

Department of Electrical Engineering

Linköpings universitet

SE-581 83 Linköping, Sweden

WWW: <http://www.control.isy.liu.se>

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET



Abstract

A simulation model for a two degrees of freedom industrial manipulator where an accelerometer is attached to the robot arm is presented. An overview of the kinematic and dynamic models as well as a thorough description of the accelerometer model are given. The simulation model can be run with different types of properties, e.g. model errors and disturbances. Different types of suggested simulation setups are also presented in the paper.

Keywords: Industrial manipulator, kinematic models, dynamic models, accelerometer

Simulation Model of a 2 Degrees of Freedom Industrial Manipulator

Patrik Axelsson

2011-06-17

Abstract

A simulation model for a two degrees of freedom industrial manipulator where an accelerometer is attached to the robot arm is presented. An overview of the kinematic and dynamic models as well as a thorough description of the accelerometer model are given. The simulation model can be run with different types of properties, e.g. model errors and disturbances. Different types of suggested simulation setups are also presented in the paper.

1 Robot Model

This section describes the kinematic and dynamic models that are used. The robot model corresponds to joint 2 and 3 for a serial 6 degrees of freedom (DOF) industrial robot.

Here, the forward kinematic equations and dynamic equations are given without any derivations. Methods to derive these equations can be found in e.g. [2] and [3]. The inverse kinematic model and the accelerometer model are explained in more details.

1.1 Forward Kinematic Model

The forward kinematic model describes the geometrical relation between the joint angles q_a and the Cartesian coordinates of the tool center point (TCP) on the robot. In this section, the forward kinematic model for the robot in Figure 1.1 will be derived. The kinematic model can be expressed as

$$P = \begin{pmatrix} x \\ z \end{pmatrix} = \Gamma(q_a), \quad (1.1)$$

where P is the Cartesian coordinates for TCP, expressed in the base coordinate frame $(x_b \ z_b)^T$, denoted by $\{b\}$, Γ is a nonlinear function and $q_a = (q_{a1} \ q_{a2})^T$ are the joint angles. The forward kinematic has a unique solution for a serial robot whereas there exist several solutions for a parallel arm robot such as ABB IRB360. There are different ways to derive the nonlinear function Γ . For this model it is fairly easy to get the expressions by considering directly the geometry in Figure 1.1. For more complex cases it is preferable to use homogeneous

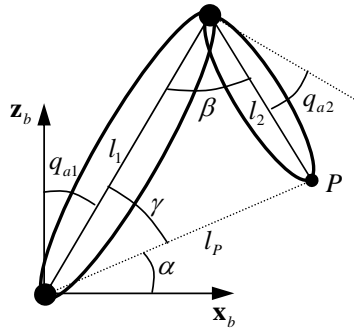


Figure 1.1: 2 DOF robot model showing the kinematic properties of the structure.

transformations with or without the Denavit-Hartenberg convention, see e.g. [2] and [3]. Using Figure 1.1 we can see that

$$\begin{aligned}
 P = \Gamma(q_a) &= \begin{pmatrix} l_1 \sin q_{a1} + l_2 \sin \left(\frac{\pi}{2} + q_{a1} + q_{a2} \right) \\ l_1 \cos q_{a1} + l_2 \cos \left(\frac{\pi}{2} + q_{a1} + q_{a2} \right) \end{pmatrix} \\
 &= \begin{pmatrix} l_1 \sin q_{a1} + l_2 \cos (q_{a1} + q_{a2}) \\ l_1 \cos q_{a1} - l_2 \sin (q_{a1} + q_{a2}) \end{pmatrix} \quad (1.2)
 \end{aligned}$$

describes the relation between the joint angles q_a and the Cartesian coordinates x and z .

1.2 Inverse Kinematic Model

In practice, often the position P for TCP is given, and we are interested in the corresponding joint angles. The joint angles are given by the inverse of (1.1), i.e.,

$$q_a = \Gamma^{-1}(P). \quad (1.3)$$

This problem is more difficult than the forward kinematic. The forward kinematic has a unique solution for a serial robot as was mentioned in Section 1.1, however, the inverse kinematic has several solutions or no solution at all. The joint angles q_a can be computed either using an analytical expression or as a result of a numerical solver. An analytical solution to the inverse problem for the robot in Figure 1.1 can be derived using trigonometric identities. The solution is however not unique. Two different sets of joint angles will give the same position. These solutions are known as elbow-up and elbow-down.

The law of cosine gives

$$\cos \gamma = \frac{l_1^2 + l_P^2 - l_2^2}{2l_1 l_P}, \quad (1.4a)$$

$$\cos \beta = \frac{l_1^2 + l_2^2 - l_P^2}{2l_1 l_2} = \sin q_{a2}, \quad (1.4b)$$

where $l_P^2 = x^2 + z^2$. The joint angle q_{a2} can now be obtained directly from (1.4b). However, the function atan2^1 will be used for numerical reasons. The angle for

¹ atan2 is a MATLAB-function for the four quadrant arctangent

joint two can therefore be calculated according to

$$q_{a2} = \text{atan2} \left(\sin q_{a2}, \pm \sqrt{1 - \sin^2 q_{a2}} \right). \quad (1.5)$$

The angle γ can be obtained in the same way,

$$\gamma = \text{atan2} \left(\pm \sqrt{1 - \cos^2 \gamma}, \cos \gamma \right). \quad (1.6)$$

The angle for joint one can now be calculated according to

$$q_{a1} = \frac{\pi}{2} - \gamma - \alpha, \quad (1.7)$$

where $\alpha = \text{atan2}(z, x)$. The elbow-up solution is obtained if the plus sign in (1.5) and (1.6) is chosen, otherwise the solution will correspond to the elbow-down solution.

1.3 Dynamic Model

The dynamic robot model is a joint flexible two-axes model from [1], see Figure 1.2. Each link is modeled as a rigid-body and described by mass m_i , length l_i , center of mass ξ_i and inertia j_i with respect to the center of mass. The joints are modeled as a spring damping pair with nonlinear spring torque τ_{si} and linear damping d_i . The deflection in each joint is given by the arm angle q_{ai} and the motor angle q_{mi} . The motor characteristics are given by the inertia j_{mi} and a nonlinear friction torque f_i . Let

$$q = (q_{a1} \quad q_{a2} \quad q_{m1}/\eta_1 \quad q_{m2}/\eta_2)^T, \quad (1.8a)$$

$$u = (0 \quad 0 \quad u_{m1}\eta_1 \quad u_{m2}\eta_2)^T, \quad (1.8b)$$

where u_{mi} is the motor torque and $\eta_i = q_{mi}/q_{ai} > 1$ is the gear ratio. The gear ratio is used to transform the motor angles and motor torques from the motor side of the gear box to the arm side. A dynamic model can be derived as

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + T(q) + D\dot{q} + F(\dot{q}) = u. \quad (1.9)$$

using Lagrange's equation [2]. Here $M(q)$ is the inertia matrix, $C(q, \dot{q})$ is the Coriolis- and centrifugal terms, $G(q)$ is the gravitational torque, $T(q)$ is the nonlinear stiffness torque, $D\dot{q}$ is the linear damping torque and $F(\dot{q})$ is the nonlinear friction torque. The complete expressions for (1.9) can be found in Appendix A.

2 Accelerometer Model

The accelerometer attached to the robot measures the acceleration due to the motion the robot performs, the gravity component and in addition some measurement noise is introduced. When modelling the accelerometer it is also important to include a drift term. The acceleration is measured in a frame $\{s\}$ fixed to the accelerometer relative an inertial frame. Note that $\{s\}$ is also fixed

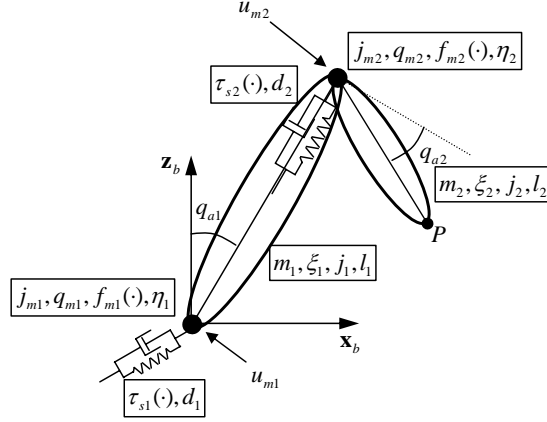


Figure 1.2: Serial robot with two degrees of freedom. The joints are modeled by nonlinear springs and linear dampers. Nonlinear friction torques are also included.

to arm two. The inertial frame is here chosen to be the world fixed base frame $\{b\}$. The acceleration in $\{s\}$ can be expressed as

$$\ddot{\rho}_s = R_s^b(q_a) (\ddot{\rho}_b + G_b) + \delta_s, \quad (2.1)$$

where $\ddot{\rho}_b$ is the acceleration due to the motion and $G_b = (0 \quad -g)^T$ models the gravitation, both expressed in the base frame $\{b\}$. The drift term is denoted by δ_s and is expressed in $\{s\}$. $R_s^b(q_a)$ is a rotation matrix that represents the rotation from frame $\{b\}$ to frame $\{s\}$. The rotation matrix $R_s^b(q_a)$ can be obtained from Figure 2.1 according to

$$R_s^b(q_a) = \begin{pmatrix} \cos(q_{a1} + q_{a2}) & -\sin(q_{a1} + q_{a2}) \\ \sin(q_{a1} + q_{a2}) & \cos(q_{a1} + q_{a2}) \end{pmatrix}. \quad (2.2)$$

The vector $\ddot{\rho}_b$ can be calculated as the second derivative of ρ_b which is shown in Figure 2.1. Note that the sensor is attached in TCP, hence

$$\rho_b = \Gamma(q_a). \quad (2.3)$$

Taking the derivative of (2.3) with respect to time twice gives

$$\dot{\rho}_b = \dot{\Gamma}(q_a) = J(q_a)\dot{q}_a, \quad (2.4a)$$

$$\ddot{\rho}_b = \ddot{\Gamma}(q_a) = J(q_a)\ddot{q}_a + \dot{J}(q_a)\dot{q}_a, \quad (2.4b)$$

where $J(q_a) = \frac{\partial \Gamma}{\partial q_a}$ is the Jacobian matrix. The Jacobian can be calculated as

$$J(q_a) = \begin{pmatrix} l_1 \cos q_{a1} - l_2 \sin(q_{a1} + q_{a2}) & -l_2 \sin(q_{a1} + q_{a2}) \\ -l_2 \cos(q_{a1} + q_{a2}) - l_1 \sin q_{a1} & -l_2 \cos(q_{a1} + q_{a2}) \end{pmatrix}, \quad (2.5)$$

using (1.2). The time derivative of the Jacobian matrix is given by

$$\dot{J}(q_a) = \sum_{i=1}^2 \frac{\partial J(q_a)}{\partial q_{ai}} \dot{q}_{ai}. \quad (2.6)$$

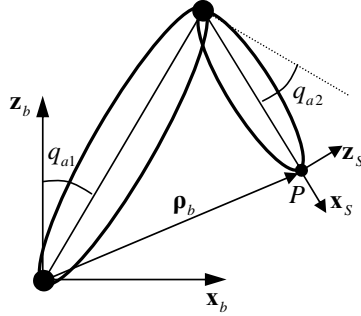


Figure 2.1: The vector ρ_b from the origin of frame $\{b\}$ to the origin of frame $\{s\}$ is used to calculate the acceleration of the point P , i.e., the acceleration measured by the accelerometer which originates from the motion.

The final expression for the acceleration given by the accelerometer can now be written as

$$\ddot{\rho}_s^M = R_s^b(q_a) \left(J(q_a) \ddot{q}_a + \left(\sum_{i=1}^2 \frac{\partial J(q_a)}{\partial q_{ai}} \dot{q}_{ai} \right) \dot{q}_a + G_b \right) + \delta_s, \quad (2.7)$$

where $R_s^b(q_a)$ and $J(q_a)$ are given by (2.2) and (2.5), respectively.

3 Overview of the Simulation Model

The simulation model is implemented in MATLAB Simulink and a block diagram of the model can be seen in Figure 3.1. The block *Path Generator* generates the desired arm angles q_a^{ref} from a set of points in the joint space or in the Cartesian space. It is also possible to use a predefined path for TCP. The desired arm angles are then used for calculating the reference trajectory for TCP, i.e., P^{ref} , using the *Forward Kinematics* in (1.2). The desired arm angles are also used to calculate references for the motor angles q_m^{ref} and a feed forward torque τ_m^{ffw} . This is done in the block *Motor Reference and Feed Forward* using an inverse dynamical model. The feedback controller is a diagonal PID controller which uses the measured motor position q_{mi}^M and the reference q_{mi}^{ref} to calculate a torque for motor i . The feed forward torque τ_m^{ffw} is added to the calculated torque from the PID controllers before it enters the robot. The block *Robot* only simulates the robot models, described in Section 1, with or without disturbances W . The output from the *Robot* block are the true position P of the TCP, true motor angles q_m and true position, velocity and acceleration for the arm angles, i.e., $Q_a = (q_a^T \quad \dot{q}_a^T \quad \ddot{q}_a^T)^T$. The block *Accelerometer* uses Q_a to calculate the acceleration of TCP according to (2.7).

Different options, listed below, are available for adjusting the realism and complexity of the simulation model.

Calibration Error The position and orientation of the accelerometer can be set to deviate from the nominal values.

Accelerometer Offset The value of the parameter δ_s in (2.1).

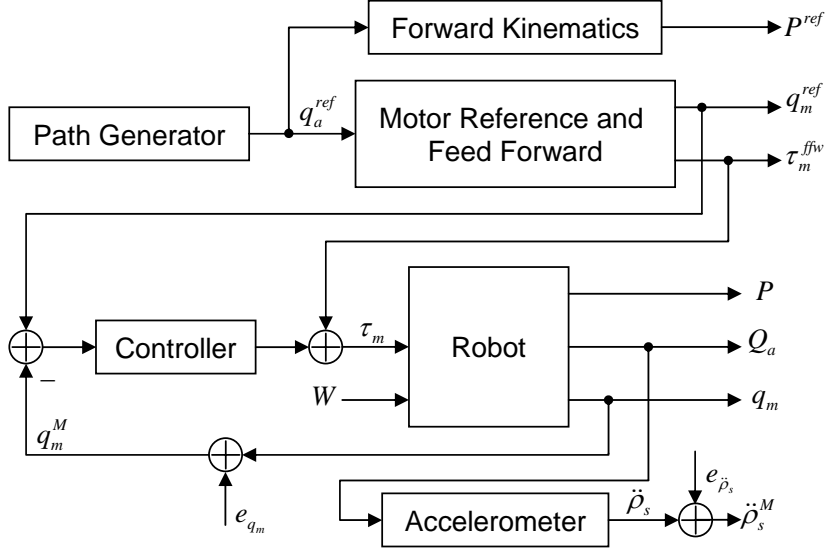


Figure 3.1: Overview of the simulation model.

Model complexity The complexity of the dynamic equation (1.9) can be changed. It is possible to turn on or off the gravity component $G(q)$ and the friction component $F(\dot{q})$. It is also possible to choose between linear and nonlinear spring torque $T(q)$. For the linear case is $k_i^{low} = k_i^{high}$ in (A.7).

Disturbances Two types of disturbances can be used. The first one is disturbances on the motor torques, which is modeled as a chirp signal. The disturbance τ_{md} is simply added to the applied torque τ_m , i.e., $u_m = \tau_m + \tau_{md}$. If no motor disturbance is present, then $u_m = \tau_m$. The second type is a force acting on TCP. The force is described by an amplitude and an angle relative the base frame $\{b\}$.

Ripple Torque ripple τ_{ri} can be added to the applied motor torque τ_{mi} , i.e., $u_{mi} = \tau_{mi} + \tau_{ri}$. The torque ripple is dependent of the motor angle q_{mi} and the applied torque τ_{mi} . The model is given by

$$\tau_{ri} = A_{c1} \sin(C_1 q_{mi} + \phi_{c1}) \tau_{mi} + \sum_{j=1}^3 A_{tj} \sin(T_j q_{mi} + \phi_{tj}). \quad (3.1)$$

If no torque ripple is present, then $u_m = \tau_m$. Ripple can also be added to the resolvers measuring the motor angles, i.e., $q_{mi}^M = q_{mi} + R_{ri}$, where the resolver ripple R_{ri} is modeled as

$$R_{ri} = A_{r1} \sin q_{mi} + A_{r2} \sin(2q_{mi} + \phi_{r2}). \quad (3.2)$$

If no resolver ripple is present, then $q_{mi}^M = q_{mi}$.

Model Errors The model parameters in (A.5), (A.7) and (A.9) and the masses m_1 and m_2 can be given uncertainty values. In this case, the feed forward

Name	Description
Sim1	Without calibration errors, drift and model errors.
Sim2	With calibration errors (4 mm in x-direction, -5 mm in z-direction and 2° in orientation), drift ($\delta_s = 0.1 \text{ m/s}^2$ in both directions) and model errors (20% in stiffness parameters and 50% in friction parameters).
Sim3	With calibration errors (4 mm in x-direction, -5 mm in z-direction and 2° in orientation), drift ($\delta_s = 0.1 \text{ m/s}^2$ in both directions) and without model errors.
Sim4	With calibration errors (4 mm in x-direction, -5 mm in z-direction and 2° in orientation), drift ($\delta_s = 0.2 \text{ m/s}^2$ in both directions) and without model errors.

Table 4.1: Four different simulation scenarios.

block in Figure 3.1 does not use the nominal values as is the case for the robot model.

Path The path can be set either as interpolation in joint space between two sets of arm angles, or linear interpolation in the Cartesian space. For more complicated paths for the TCP it is possible to create them off-line before the simulation starts.

Measurement Noise Normal distributed white measurement noise with zero mean and variance $\sigma_{q_m}^2$ and $\sigma_{\rho_s}^2$ can be added to the motor angles and/or the acceleration of TCP, respectively.

4 Suggested Simulation Setups

Here, four different paths are suggested together with different configurations from the list in Section 3. These paths and configurations are used during later work.

The suggested paths can be seen in Figure 4.1. The paths start at the star and goes clockwise and the circle indicates TCP when $q_a = (0 \ 0)^T$. These four paths are generated using a standard ABB controller.

Four different configurations of the simulation model according to Table 4.1 are suggested to cover model errors in the dynamical model parameters, as well as drift and calibration errors for the accelerometer. All four configurations use full model complexity, i.e., gravity, friction and nonlinear spring torque are present. Motor torque ripple, resolver ripple and measurement noise on the motor angles and the acceleration of TCP are also present, but no disturbances. The model errors can be seen as the worst case and are chosen based on suggestions from the authors of [1]. Note that the true trajectory for TCP will be the same for all suggested configurations except for Sim2. The reason is that the feed forward controller changes, i.e., the control performance is changed, when model errors are present.

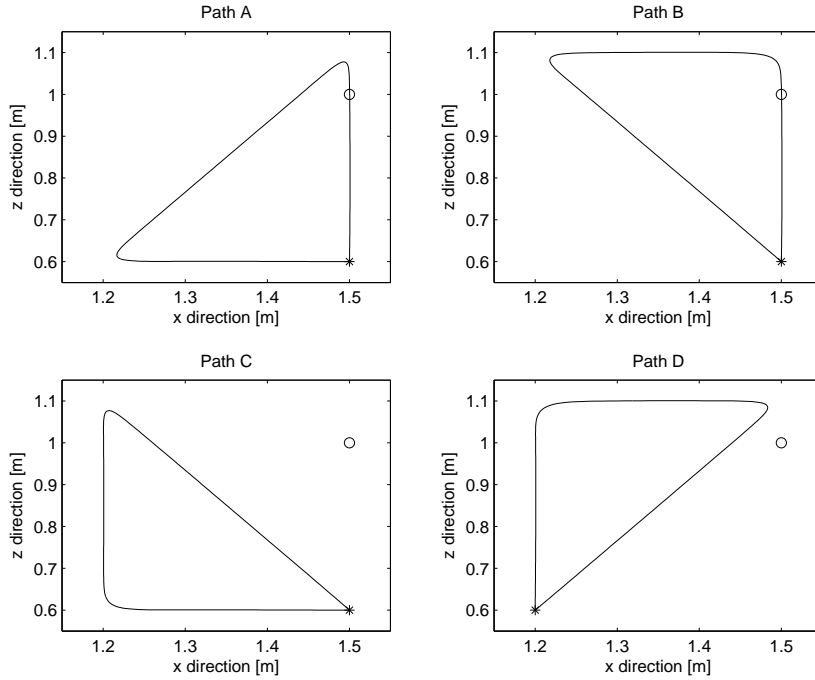


Figure 4.1: Suggested paths for simulation. The path starts at the star and goes clockwise. The circle indicates the tool position when $q_a = (0 \ 0)^T$.

References

- [1] Stig Moberg, Jonas Öhr, and Svante Gunnarsson. A benchmark problem for robust control of a multivariable nonlinear flexible manipulator. In *Proceedings of 17th IFAC World Congress*, pages 1206–1211, Seoul, Korea, July 2008. URL: <http://www.robustcontrol.org>.
- [2] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and Control of Robot Manipulators*. Springer, London, UK, second edition, 2000.
- [3] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, 2005.

A Dynamic Model Equations

The expressions for the dynamic model (1.9) are given here. More information about the model and numerical values for the parameters can be found in [1].

The inertia matrix $M(q)$ is partitioned as

$$M(q) = \begin{pmatrix} M_{11}(q) & M_{12}(q) & 0 & 0 \\ M_{21}(q) & M_{22}(q) & 0 & 0 \\ 0 & 0 & j_{m1}\eta_1^2 & 0 \\ 0 & 0 & 0 & j_{m2}\eta_2^2 \end{pmatrix}, \quad (\text{A.1})$$

where

$$M_{11}(q) = j_1 + m_1 \xi_1^2 + j_2 + m_2 (l_1^2 + \xi_2^2 - 2l_1 \xi_2 \sin q_{a2}), \quad (\text{A.2a})$$

$$M_{12}(q) = M_{21}(q) = j_2 + m_2 (\xi_2^2 - l_1 \xi_2 \sin q_{a2}), \quad (\text{A.2b})$$

$$M_{22}(q) = j_2 + m_2 \xi_2^2. \quad (\text{A.2c})$$

The Coriolis and centripetal terms are described by

$$C(q, \dot{q}) = \begin{pmatrix} -m_2 l_1 \xi_2 (2\dot{q}_{a1} \dot{q}_{a2} + \dot{q}_{a2}^2) \cos q_{a2} \\ m_2 l_1 \xi_2 \dot{q}_{a1}^2 \cos q_{a2} \\ 0 \\ 0 \end{pmatrix}, \quad (\text{A.3})$$

the gravity component is

$$G(q) = \begin{pmatrix} -g (m_1 \xi_1 \sin q_{a1} + m_2 (l_1 \sin q_{a1} + \xi_2 \cos(q_{a1} + q_{a2}))) \\ -m_2 \xi_2 g \cos(q_{a1} + q_{a2}) \\ 0 \\ 0 \end{pmatrix}, \quad (\text{A.4})$$

and the linear damping matrix is given by

$$D = \begin{pmatrix} d_1 & 0 & -d_1 & 0 \\ 0 & d_2 & 0 & -d_2 \\ -d_1 & 0 & d_1 & 0 \\ 0 & -d_2 & 0 & d_2 \end{pmatrix}. \quad (\text{A.5})$$

The nonlinear spring torque is described by

$$T(q) = \begin{pmatrix} \tau_{s1}(\Delta_{q1}) \\ \tau_{s2}(\Delta_{q2}) \\ \tau_{s1}(-\Delta_{q1}) \\ \tau_{s2}(-\Delta_{q2}) \end{pmatrix}, \quad (\text{A.6})$$

where

$$\Delta_{q_i} = q_{ai} - q_{mi} / \eta_i \quad (\text{A.7a})$$

$$\tau_{si} = \begin{cases} k_i^{low} \Delta_{q_i} + k_{i3} \Delta_{q_i}^3, & |\Delta_{q_i}| \leq \psi_i \\ \text{sign}(\Delta_{q_i}) (m_{i0} + k_i^{high} (|\Delta_{q_i}| - \psi_i)), & |\Delta_{q_i}| > \psi_i \end{cases} \quad (\text{A.7b})$$

$$k_{i3} = \frac{k_i^{high} - k_i^{low}}{3\psi_i^2} \quad (\text{A.7c})$$


$$m_{i0} = k_i^{low} \psi_i + k_{i3} \psi_i^3 \quad (\text{A.7d})$$

Finally, the nonlinear friction torque is given by

$$F(\dot{q}) = \begin{pmatrix} 0 \\ 0 \\ f_{m1}(\dot{q}) \\ f_{m2}(\dot{q}) \end{pmatrix}, \quad (\text{A.8})$$

where

$$f_{mi}(\dot{q}) = \eta_i (f_{di} \dot{q}_{mi} + f_{ci} (\mu_{ki} + (1 - \mu_{ki}) \cosh^{-1}(\beta_i \dot{q}_{mi})) \tanh(\alpha_i \dot{q}_{mi})). \quad (\text{A.9})$$

	Avdelning, Institution Division, Department Division of Automatic Control Department of Electrical Engineering	Datum Date 2011-06-17
	Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input checked="" type="checkbox"/> Övrig rapport <input type="checkbox"/> _____
URL för elektronisk version http://www.control.isy.liu.se		LiTH-ISY-R-3020
Titel Simulation Model of a 2 Degrees of Freedom Industrial Manipulator Title		
Författare Patrik Axelsson Author		
Sammanfattning Abstract <p>A simulation model for a two degrees of freedom industrial manipulator where an accelerometer is attached to the robot arm is presented. An overview of the kinematic and dynamic models as well as a thorough description of the accelerometer model are given. The simulation model can be run with different types of properties, e.g. model errors and disturbances. Different types of suggested simulation setups are also presented in the paper.</p>		
Nyckelord Keywords Industrial manipulator, kinematic models, dynamic models, accelerometer		