REALISTIC DYNAMIC SIMULATION OF AN INDUSTRIAL ROBOT WITH JOINT FRICTION

Ronald G.K.M. Aarts, Ben J.B. Jonker
University of Twente, Faculty of Engineering Technology
Enschede, The Netherlands

Rob R. Waiboer
Netherlands Institute for Metals Research
Delft, The Netherlands

Keywords: Realistic closed-loop trajectory simulation, Industrial robot, Perturbation method, Friction modelling.

Abstract: This paper presents a realistic dynamic simulation of the closed-loop tip motion of a rigid-link manipulator with joint friction. The results from two simulation techniques are compared with experimental results. A six-axis industrial Stäubli robot is modelled. The LuGre friction model is used to account both for the sliding and pre-sliding regime. The manipulation task implies transferring a laser spot along a straight line with a trapezoidal velocity profile.
Firstly, a non-linear finite element method is used to formulate the dynamic equations of the manipulator mechanism. In a closed-loop simulation the driving torques are generated by the control system. The computed trajectory tracking errors agree well with the experimental results. Unfortunately, the simulation is very time-consuming due to the small time step of the discrete-time controller.
Secondly, a perturbation method has been applied. In this method the perturbed motion of the manipulator is modelled as a first-order perturbation of the nominal manipulator motion. Friction torques at the actuator joints are introduced at the stage of perturbed dynamics. A substantial reduction of the computer time is achieved without loss of accuracy.

1 INTRODUCTION

Robotic manipulators for laser welding must provide accurate path tracking performance of 0.1 mm and less at relatively high tracking speed exceeding 100 mm/s. High speed manipulator motions are accompanied by high-frequency vibrations of small magnitudes whereas velocity reversals in the joints lead to complicated joint friction effects. These dynamic phenomena may significantly effect the weld quality, since laser welding is sensitive to small variations in the processing conditions like welding speed as well as to disturbances of the position of the laser spot with respect to the seam to be welded. To study the applicability of industrial robotic manipulators as in Fig. 1 for laser welding tasks, a framework for realistic dynamic simulations of the robot motion is being developed. Such a simulation should predict the path tracking accuracies, thereby taking into account the dynamic limitations of the robotic manipulator such as joint flexibility, friction and limits of the drive system like the maximum actuator torques. Obviously the accuracy of the simulation has to be sufficiently high with respect to the allowable path deviations.

Unfortunately, closed-loop dynamic simulations with a sufficient degree of accuracy are computationally expensive and time consuming especially when static friction, high-frequency elastic modes and a digital controller operating with small discrete time steps are involved. In an earlier paper we discussed a so-called perturbation method (Jonker and Aarts, 2001) which proved to be accurate and computationally efficient for simulations of robotic manipulators with flexible links. In this paper a similar perturbation scheme is presented for the closed-loop dynamic simulation of a rigid-link manipulator with joint friction. The extension to an industrial robot with elastic joints is still a topic of ongoing research.

We consider the motion of a six axes industrial robot (Stäubli RX90B), Fig. 1. The unknown robot manipulator parameters, such as inertias of the manipulator arms, the stiffness and the pretension of a gravity compensation spring are determined by means of parameter identification techniques. The friction characteristics are identified separately. The controller model is based on manufacturer's data and has been verified experimentally using system identification techniques. Details of the identification tech-

Aarts R., Jonker B. and Waiboer R. (2004).
REALISTIC DYNAMIC SIMULATION OF AN INDUSTRIAL ROBOT WITH JOINT FRICTION.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 48-55
DOI: 10.5220/0001143500480055

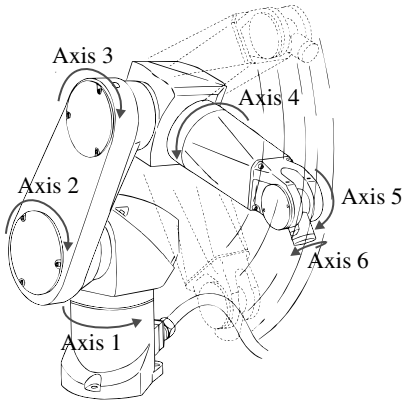


Figure 1: The six axes Stäubli RX90B Industrial robot

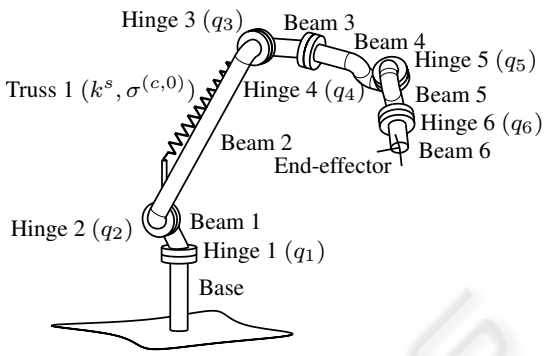


Figure 2: The 6 DOF Finite Element Model.

niques are outside the scope of this paper and will be outlined elsewhere (Waiboer, 2004).

The present paper will focus on two simulation techniques and a comparison of numerical and experimental results. At first, a non-linear finite element method (Jonker, 1990) is used to formulate the dynamic equations of the manipulator mechanism, section 2. The finite element formulation and the proposed solution method are implemented in the program SPACAR (Jonker and Meijaard, 1990). An interface to MATLAB is available and the closed-loop simulations are carried out using SIMULINK's graphical user interface. A driving system is added to the manipulator model, section 3, and the influence of joint friction is taken into account, section 4. The LuGre friction model (Wit et al., 1995) has been used as it accounts for both the sliding and pre-sliding regimes. Finally the closed-loop model is assembled by including the control system, section 5.

The second approach is the so-called perturbation method, section 6. The differences between the actual manipulator motion and the nominal (desired) motion are modelled as first order perturbations of that nominal motion. The computation of the perturbed motion is carried out in two steps. In the first step, the finite element method permits the generation of locally linearised models (Jonker and Aarts, 2001). In the second step, the linearised model can simulate the perturbed motion of the manipulator accurately as long as it remains sufficiently close to the nominal path. The perturbation method is computationally more efficient as the non-linear model only has to solved during the first step in a limited number of points along the nominal trajectory. This is the solution of an inverse dynamic analysis along a known path, so with only algebraic equations. The friction torques at the actuator joints and the control system are introduced at the stage of perturbed dynamics in the second step. In section 7, the simulation results will be compared with the experimental results. Also results acquired with the perturbation method will be compared with those obtained from a full non-linear dynamic simulation.

2 FINITE ELEMENT REPRESENTATION OF THE MANIPULATOR

In the non-linear finite element method, a manipulator mechanism is modelled as an assembly of finite elements interconnected by joint elements such as hinge elements and (slider) truss elements. This is illustrated in Fig. 2, where the Stäubli robot with six degrees of freedom is modelled by three different types of elements. The gravity compensation spring is modelled as a slider-truss element. The manipulator arms are modelled by beam elements. Finally, the joints are represented by six cylindrical hinge elements, which are actuated by torque servos. The manipulator mechanism is assembled by allowing the elements to have nodal points in common. The configuration of the manipulator is then described by the vector of nodal coordinates $\boldsymbol{x}$, some of which may be the Cartesian coordinates, while others describe the orientation of orthogonal triads rigidly attached at the element nodes. The motion of the manipulator mechanism is described by relative degrees of freedom which are the actuator joint angles denoted by the vector $\boldsymbol{q}$. By means of the geometric transfer functions $\boldsymbol{\mathcal{F}}^{(x)}$ and $\mathcal{F}^{(e,c)}$, the nodal coordinates $\boldsymbol{x}$ and the elongation of the gravity compensating spring $e^{(c)}$ are expressed as functions of the joint angles $\boldsymbol{q}$

$$\boldsymbol{x} = \boldsymbol{\mathcal{F}}^{(x)}(\boldsymbol{q}) \tag{1}$$

and

$$e^{(c)} = \mathcal{F}^{(e,c)}(\boldsymbol{q}). \tag{2}$$

Differentiating the transfer functions with respect to time gives

$$\dot{\boldsymbol{x}} = \mathrm{D}\boldsymbol{\mathcal{F}}^{(x)}\dot{\boldsymbol{q}} \tag{3}$$

and

$$\dot{e}^{(c)} = \mathrm{D}\boldsymbol{\mathcal{F}}^{(e,c)}\dot{\boldsymbol{q}}, \tag{4}$$

where the differentiation operator D represents partial differentiation with respect to the degrees of freedom. The acceleration vector $\ddot{\boldsymbol{x}}$ is obtained by differentiating (3) again with respect to time,

$$\ddot{\boldsymbol{x}} = \mathrm{D}\boldsymbol{\mathcal{F}}^{(x)}\ddot{\boldsymbol{q}} + (\mathrm{D}^2\boldsymbol{\mathcal{F}}^{(x)}\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}. \tag{5}$$

The geometric transfer functions $\boldsymbol{\mathcal{F}}^{(x)}$ and $\mathcal{F}^{(e,c)}$ are determined numerically in an iterative way (Jonker, 1990).

The inertia properties of the concentrated and distributed mass of the elements are described with the aid of lumped mass matrices. Let $\mathbf{M}(\boldsymbol{x})$ be the global mass matrix, obtained by assembling the mass matrices of the individual elements and let $\boldsymbol{f}(\boldsymbol{x}, \dot{\boldsymbol{x}}, t)$ be the global force vector including gravity forces and the velocity dependent inertia forces. Then the equations of motion described in the dynamic degrees of freedom are given by

$$\begin{aligned}\bar{\mathbf{M}}\ddot{\boldsymbol{q}} + \mathrm{D}\boldsymbol{\mathcal{F}}^{(x)T}\left[\mathbf{M}(\mathrm{D}^2\boldsymbol{\mathcal{F}}^{(x)}\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{f}\right] \\ + \mathrm{D}\boldsymbol{\mathcal{F}}^{(e,c)T}\sigma^{(c)} = \boldsymbol{\tau},\end{aligned} \tag{6}$$

where $\bar{\mathbf{M}} = \mathrm{D}\boldsymbol{\mathcal{F}}^{(x)T}\mathbf{M}\mathrm{D}\boldsymbol{\mathcal{F}}^{(x)}$ is the reduced mass matrix and $\sigma^{(c)}$ is the total stress in the gravity compensating spring. The vector $\boldsymbol{\tau}$ represents the joint driving torques. The constitutive relation for the gravity compensating spring is described by

$$\sigma^{(c)} = \sigma^{(c,0)} + k^s\, e^{(c)}, \tag{7}$$

where $\sigma^{c,0}$ and k^s denote the pre-stress and the stiffness coefficients of the spring, respectively.

This finite element method has been implemented in the SPACAR software program (Jonker and Meijaard, 1990).

3 THE DRIVING SYSTEM

The robot is driven by servo motors connected via gear boxes to the robot joints. The inputs for the drive system are the servo currents i_j and the outputs are the joint torques τ_j, where the subscript $j = 1..6$ denotes either the motor or the joint number. A schematic model of the drives of joints 1 to 4 is shown in Fig. 3. The relations between the vector of motor angles $\boldsymbol{\varphi}$, the vector of joint angles $\boldsymbol{q}$ and their time derivatives are given by

$$\boldsymbol{\varphi} = \mathbf{T}\boldsymbol{q}, \tag{8}$$

$$\dot{\boldsymbol{\varphi}} = \mathbf{T}\dot{\boldsymbol{q}}, \tag{9}$$

$$\ddot{\boldsymbol{\varphi}} = \mathbf{T}\ddot{\boldsymbol{q}}, \tag{10}$$

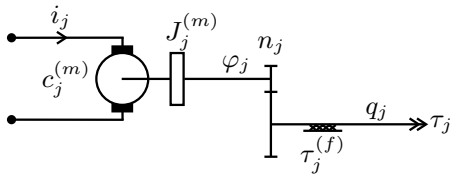


Figure 3: Schematic representation of the drives for joints 1 to 4.

where $\mathbf{T}$ is the transmission model for the joints. For joints $j = 1..5$ only the respective gear ratio n_j of the joint plays a role. The drives for the wrist of the robot, drives 5 and 6, are coupled. This causes an extra term leading to

$$\mathbf{T} = \begin{bmatrix} n_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & n_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & n_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & n_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & n_5 & 0 \\ 0 & 0 & 0 & 0 & n_6 & n_6 \end{bmatrix}. \tag{11}$$

The friction torque in each joint is denoted $\tau_j^{(f)}$ with $j = 1..6$. Due to the coupling in the wrist, an additional friction torque $\tau_7^{(f)}$ is identified on the axis of motor 6. The vector of joint friction torques is then defined as

$$\begin{aligned}\boldsymbol{\tau}^{(f)} = \Big[\tau_1^{(f)},\, \tau_2^{(f)},\, \tau_3^{(f)},\, \tau_4^{(f)},\, \cdots \\ \tau_5^{(f)}+\tau_7^{(f)},\, \tau_6^{(f)}+\tau_7^{(f)}\Big]^T.\end{aligned} \tag{12}$$

The joint friction modelling is continued in section 4.

The drives are equipped with permanent magnet, three-phase synchronous motors, yielding a linear relation between motor currents and torque. The vector of joint driving torques $\boldsymbol{\tau}$ is then given by

$$\boldsymbol{\tau} = \mathbf{T}^T\mathbf{C}^{(m)}\boldsymbol{i} - \mathbf{T}^T\mathbf{J}^{(m)}\ddot{\boldsymbol{\varphi}} - \boldsymbol{\tau}^{(f)}, \tag{13}$$

where the matrices $\mathbf{C}^{(m)}$ and $\mathbf{J}^{(m)}$ are diagonal matrices with the motor constants $c_j^{(m)}$ and rotor inertias $J_j^{(m)}$, respectively. Substitution of (10) and (13) into (6) and some rearranging yields

$$\begin{aligned}\bar{\mathbf{M}}^{(n)}\ddot{\boldsymbol{q}} + \mathrm{D}\boldsymbol{\mathcal{F}}^{(x)T}\left[\mathbf{M}(\mathrm{D}^2\boldsymbol{\mathcal{F}}^{(x)}\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{f}\right] \\ + \mathrm{D}\boldsymbol{\mathcal{F}}^{(e,c)T}\sigma^{(c)} = \boldsymbol{\tau}^{(n)},\end{aligned} \tag{14}$$

where the mass matrix $\bar{\mathbf{M}}^{(n)}$ is defined by

$$\bar{\mathbf{M}}^{(n)} = \bar{\mathbf{M}} + \mathbf{T}^T\mathbf{J}^{(m)}\mathbf{T}, \tag{15}$$

as the rotor inertias $\mathbf{T}^T\mathbf{J}^{(m)}\mathbf{T}$ obviously have to be added to the reduced mass matrix $\bar{\mathbf{M}}$ in the equations of motion (6). Furthermore, the vector of net joint torques is defined as

$$\boldsymbol{\tau}^{(n)} = \mathbf{T}^T\mathbf{C}^{(m)}\boldsymbol{i} - \boldsymbol{\tau}^{(f)}. \tag{16}$$

The inertia properties and spring coefficients have been found by means of parameter identification techniques (Waiboer, 2004).

4 JOINT FRICTION MODEL

For feed-forward dynamic compensation purposes and robot inertia identification techniques it is common to model friction in robot joints as a torque $\tau^{(f)}$ which consists of a Coulomb friction torque and an additional viscous friction torque for non-zero velocities (Swevers et al., 1996; Calafiore et al., 2001). These so-called static or kinematic friction models are valid only at sufficiently high velocities because they ignore the presliding regime. At zero velocity they show a discontinuity in the friction torque which gives rise to numerical integration problems in a forward dynamic simulation. To avoid this problem we apply the LuGre friction model (Wit et al., 1995), that accounts for both the friction in the sliding and in the presliding regime.

In the LuGre friction model there is an internal state z that describes the average presliding displacement, as introduced by Heassig et al. (Haessig and Friedland, 1991). The state equations with a differential equation for the state z and an output equation for the friction moment $\tau^{(f)}$ are

$$\dot{z} = \dot{q} - \frac{|\,\dot{q}\,|}{g(\dot{q})} z, \tag{17}$$

$$\tau^{(f)} = c^{(0)}\, z + c^{(1)}\, \dot{z} + c^{(2)}\, \dot{q}\,. \tag{18}$$

Note that a subscript $j = 1 \ldots 7$ should be added to all variables to distinguish between the separate friction torques in the robot model, but it is omitted here for better readability. For each joint friction model with $j = 1..6$ the input velocity equals the joint velocity $\dot{q}_j$. For the extra friction model $\tau_7^{(f)}$ the input velocity is defined as the sum of the joint velocities of joints 5 and 6.

In general, the friction torque in the pre-sliding regime is described by a non-linear spring-damper system that is modelled with an equivalent stiffness $c^{(0)}$ for the position–torque relationship at velocity reversal and a micro-viscous damping coefficient $c^{(1)}$. At zero velocity, the deformation of the non-linear spring torque is related to the joint (micro) rotation q.

The viscous friction torque in the sliding regime is modelled by $c^{(2)}\,\dot{q}$, where $c^{(2)}$ is the viscous friction coefficient. In addition, at a non-zero constant velocity $\dot{q}$, the internal state z, so the average deformation, will approach a steady state value equal to $c^{(0)}\, g(\dot{q})$. The function $g(\dot{q})$ can be any function that represents the constant velocity behaviour in the sliding regime. In this paper the Stribeck model will be used, which models the development of a lubricating film between the contact surfaces as the relative velocity increases from zero. The Stribeck model is given by

$$c^{(0)}\, g(\dot{q}) = \tau^{(c)} + (\tau^{(s)} - \tau^{(c)}) e^{-(|\dot{q}|/\omega_s)^{\delta}}, \tag{19}$$

where $\tau^{(c)}$ is known as Coulomb friction torque, $\tau^{(s)}$ is the static friction torque and ω_s and δ are shaping parameters. The values for the static friction torque $\tau^{(s)}$ and Coulomb friction torque $\tau^{(c)}$ may be different for positive and negative velocities and are therefore distinguished by the subscripts $+$ and $-$, respectively.

For each friction torque in the robot model, the parameters describing the sliding regime of the LuGre friction model are estimated separately using dedicated joint torque measurements combined with non-linear parameter optimisation techniques (Waiboer, 2004). The parameters describing the presliding regime are approximated by comparing the pre-sliding behaviour in simulation with measurements.

5 CLOSED-LOOP ROBOT MODEL

The SPACAR model of the manipulator mechanism, the controller, the friction models and the robot drives are assembled into a complete model of the closed-loop robot system, Fig. 4.

The CS7B controller used in the Stäubli RX90B is an industrial PID controller based on the Adept V+ operating- and motion control system. The controller includes six SISO controllers, one for each servo-motor. A trajectory generator computes the set-points $\ddot{\boldsymbol{q}}^{(r)}$, $\dot{\boldsymbol{q}}^{(r)}$ and $\boldsymbol{q}^{(r)}$ at a rate of 1 kHz. The actual joint position $\boldsymbol{q}$ is compared with the set-point $\boldsymbol{q}^{(r)}$ and the error $\boldsymbol{\epsilon}$ is fed into the PI-block of the controller, which includes the proportional part and the integrator part. The acceleration set-points $\ddot{\boldsymbol{q}}^{(r)}$ and velocity set-points $\dot{\boldsymbol{q}}^{(r)}$ are used for acceleration feed-forward and friction compensation, both included in the FF-block. The friction compensation includes both Coulomb and viscous friction. The velocity feedback $\dot{\boldsymbol{q}}$ is fed into the D-block, representing the differential part of the control scheme. The block C represents the current loop, including the power amplifier. Note that the position loop runs at $f_s^{(1)} = 1$ kHz and the velocity loop and current loop run at $f_s^{(2)} = 4$ kHz. The transfer functions of the different blocks have been identified and implemented in a SIMULINK block scheme of the robot controller (Waiboer, 2004).

The current vector $\boldsymbol{i}$ is fed into the drive model where the joint torques are computed. The LuGre friction models compute the friction torques from the joint velocities $\dot{\boldsymbol{q}}$. The net joint torque $\boldsymbol{\tau}^{(n)}$ in (14) is the input for the non-linear manipulator model (SPACAR). The output of the manipulator model contains the joint positions, and velocities, denoted by $\boldsymbol{q}$

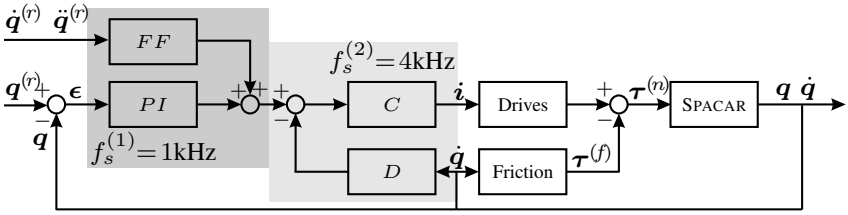


Figure 4: Assembly of the closed-loop robot model for a simulation with the non-linear finite element method (SPACAR block).

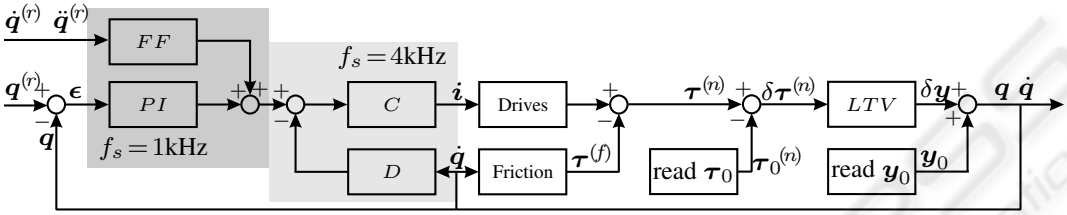


Figure 5: Perturbation scheme with the linearised equations of motion (LTV block).

and $\dot{\boldsymbol{q}}$, respectively.

6 PERTURBATION METHOD

In the presented closed-loop model, both the SPACAR model and the friction model are continuous in time, while the controller sections are discrete and running at 1 kHz and 4 kHz, respectively. In SIMULINK this implies that the SPACAR model is simulated at a time step equal to 0.25 ms or even smaller if this is required by SIMULINK's integrator. Due to the fact that the position, computed within the SPACAR simulation model, is obtained in an iterative way, the computation effort is quite elaborate. This results in long simulation times. In order to overcome this drawback, the perturbation method (Jonker and Aarts, 2001) is applied.

In the perturbation method, deviations from the (nominal) desired motion $(\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0, \ddot{\boldsymbol{q}}_0)$ due to joint friction and errors of the control system are modelled as first-order perturbations $(\delta\boldsymbol{q}, \delta\dot{\boldsymbol{q}}, \delta\ddot{\boldsymbol{q}})$ of the nominal motion, such that the actual variables are of the form:

$$\boldsymbol{q} = \boldsymbol{q}_0 + \delta\boldsymbol{q}, \tag{20}$$

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_0 + \delta\dot{\boldsymbol{q}}, \tag{21}$$

$$\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_0 + \delta\ddot{\boldsymbol{q}}, \tag{22}$$

where the prefix δ denotes a perturbation. Expanding (1) and (3) in their Taylor series expansion and disregarding second and higher order terms results in the linear approximations

$$\delta\boldsymbol{x} = \mathrm{D}\boldsymbol{\mathcal{F}}_0 \cdot \delta\boldsymbol{q}, \tag{23}$$

$$\delta\dot{\boldsymbol{x}} = \mathrm{D}\boldsymbol{\mathcal{F}}_0 \cdot \delta\dot{\boldsymbol{q}} + (\mathrm{D}^2\boldsymbol{\mathcal{F}}_0 \cdot \dot{\boldsymbol{q}}_0) \cdot \delta\boldsymbol{q}. \tag{24}$$

The subscript 0 refers to the nominal trajectory.

The nominal motion is computed first and is described by the non-linear manipulator model. The non-linear equations of motion (14) for the nominal motion become

$$\begin{aligned} &\bar{\mathbf{M}}_0^{(n)} \ddot{\boldsymbol{q}}_0 \\ &+ \mathrm{D}_q \boldsymbol{\mathcal{F}}_0^T \left[\mathbf{M}_0 (\mathrm{D}^2\boldsymbol{\mathcal{F}}_0 \cdot (\dot{\boldsymbol{q}}_0)) \cdot (\dot{\boldsymbol{q}}_0) - \boldsymbol{f} \right] \\ &\qquad + \mathrm{D}\boldsymbol{\mathcal{F}}_0^{(e,c)} \sigma_0^{(c)} = \boldsymbol{\tau}_0^{(n)}, \end{aligned} \tag{25}$$

where $\sigma_0^{(c)} = \sigma^{(c,0)} + k^s\, e_0^{(c)}$. The right hand side vector $\boldsymbol{\tau}_0^{(n)}$ represents the nominal net input torque necessary to move the manipulator along the nominal (desired) trajectory. It is determined from an inverse dynamic analysis based on (25). Note that $\boldsymbol{\tau}_0^{(n)}$ is computed without the presence of friction. Obviously, the desired nominal motion $(\ddot{\boldsymbol{q}}_0, \dot{\boldsymbol{q}}_0, \boldsymbol{q}_0)$ equals the reference motion $(\ddot{\boldsymbol{q}}^{(r)}, \dot{\boldsymbol{q}}^{(r)}, \boldsymbol{q}^{(r)})$ of the closed loop system.

Next the perturbed motion is described by a set of linear time-varying equations of motion, which are obtained by linearising the equations of motion around a number of points on the nominal trajectory. The resulting equations of motion for the perturbations of the degrees of freedom $\delta\boldsymbol{q}$ are

$$\bar{\mathbf{M}}_0^{(n)} \delta\ddot{\boldsymbol{q}} + \mathbf{C}_0 \delta\dot{\boldsymbol{q}} + \bar{\mathbf{K}}_0 \delta\boldsymbol{q} = \delta\boldsymbol{\tau}^{(n)}, \tag{26}$$

where $\bar{\mathbf{M}}_0^{(n)}$ is the reduced system mass matrix as in (14), $\mathbf{C}_0$ is the velocity sensitivity matrix. Matrix $\bar{\mathbf{K}}_0$ is a shorthand notation for the combined stiffness matrices

$$\bar{\mathbf{K}}_0 = \mathbf{K}_0 + \mathbf{N}_0 + \mathbf{G}_0, \tag{27}$$

where $\mathbf{K}_0$ denotes the structural stiffness matrix, $\mathbf{N}_0$ and $\mathbf{G}_0$ are the dynamic stiffening matrix and the geometric stiffening matrix, respectively. The matrices $\bar{\mathbf{M}}_0^{(n)}$, $\mathbf{K}_0$ and $\mathbf{G}_0$ are symmetric, but $\mathbf{C}_0$ and $\mathbf{N}_0$ need not to be symmetrical. Expressions for these matrices are given in (Jonker and Aarts, 2001).

Fig. 5 shows the block scheme of the closed-loop perturbation model. Note that the non-linear manipulator model (SPACAR) is replaced by the LTV-block which solves (26). Its input $\delta\boldsymbol{\tau}^{(n)}$ is computed according to

$$\delta\boldsymbol{\tau}^{(n)} = \boldsymbol{\tau}^{(n)} - \boldsymbol{\tau}_0^{(n)}, \tag{28}$$

where $\boldsymbol{\tau}^{(n)}$ represents the vector of the net joint torques as defined by (16). The perturbed output vector $\delta\boldsymbol{y}$ is added to the output vector $\boldsymbol{y}_0$ of the nominal model, yielding the actual output $\boldsymbol{y}$. This output vector will be discussed below.

Within the SIMULINK framework the LTV block represents a linear time-varying state space representation of a system

$$\begin{aligned} \dot{\boldsymbol{x}}_{\text{ss}} &= \mathbf{A}_{\text{ss}}\boldsymbol{x}_{\text{ss}} + \mathbf{B}_{\text{ss}}\boldsymbol{u}_{\text{ss}} \\ \boldsymbol{y}_{\text{ss}} &= \mathbf{C}_{\text{ss}}\boldsymbol{x}_{\text{ss}} \end{aligned} \tag{29}$$

where $\boldsymbol{u}_{\text{ss}}$, $\boldsymbol{y}_{\text{ss}}$ and $\boldsymbol{x}_{\text{ss}}$ are the input, output and state vectors, respectively, and $\mathbf{A}_{\text{ss}}$, $\mathbf{B}_{\text{ss}}$ and $\mathbf{C}_{\text{ss}}$ are the time-varying state space matrices. Equation (26) is written in this representation by defining the state and input vectors

$$\begin{aligned} \boldsymbol{x}_{\text{ss}} &= [\delta\dot{\boldsymbol{q}}, \delta\boldsymbol{q}]^T, \\ \boldsymbol{u}_{\text{ss}} &= \delta\boldsymbol{\tau}^{(n)}. \end{aligned} \tag{30}$$

The state space matrices $\mathbf{A}_{\text{ss}}$ and $\mathbf{B}_{\text{ss}}$ are then computed from the matrices in (26) using a straightforward procedure :

$$\begin{aligned} \mathbf{A}_{\text{ss}} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\bar{\mathbf{M}}_0^{(n)\,-1}\bar{\mathbf{K}}_0 & -\bar{\mathbf{M}}_0^{(n)\,-1}\mathbf{C}_0 \end{bmatrix}, \\ \mathbf{B}_{\text{ss}} &= \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{M}}_0^{(n)\,-1} \end{bmatrix}. \end{aligned} \tag{31}$$

The output matrix $\mathbf{C}_{\text{ss}}$ depends on the output vector $\boldsymbol{y}_{\text{ss}}$ that is defined by the user. It may consist of (first derivatives of) the degrees of freedom $\delta\boldsymbol{q}$ present in $\boldsymbol{x}_{\text{ss}}$ or (first derivatives of) nodal coordinates $\delta\boldsymbol{x}$ that are computed from $\boldsymbol{x}_{\text{ss}}$ using (23) and (24). E.g. the coordinates $\delta\boldsymbol{x}$ in $\boldsymbol{y}_{\text{ss}}$ are computed using

$$\mathbf{C}_{\text{ss}} = [\ \mathrm{D}\boldsymbol{\mathcal{F}}_0 \quad \mathbf{0}\ ]. \tag{32}$$

For a practical implementation of the perturbation method the total trajectory time T is divided into intervals. The state space matrices and the vectors $\boldsymbol{\tau}_0$ and $\boldsymbol{y}_0$ are computed during a preprocessing run at a number N of discrete time steps $t = t_i$ $(i = 0, 1, 2, ..., N)$ and stored in a file. During the simulation runs the matrices and vectors are then read back from this file. Between two time steps the coefficients of the state space matrices are interpolated linearly. Cubic interpolation is used for the vectors $\boldsymbol{\tau}_0$ and $\boldsymbol{y}_0$.

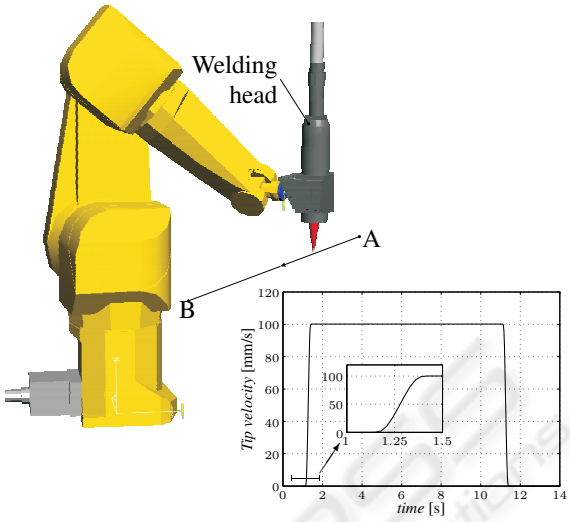


Figure 6: The simulated straight line path and the path velocity.

7 SIMULATION RESULTS

In Fig. 6 an experiment is defined in which a straight line motion of the robot tip has to be performed in the horizontal plane with a velocity profile as presented in the insert. The welding head remains in the vertical position as shown in Fig. 6. The joint set-points are computed with the inverse kinematics module of the SPACAR software. The joint velocities are shown in Fig. 7. The figures show clearly the non-linear behaviour of the robot. Note the velocity reversal of joints 2 and 3 nearly halfway the trajectory.

First, the trajectory will be simulated with the non-linear model. The simulation has been performed in SIMULINK using the `ode23tb` integration scheme, which is one of the stiff integrators in SIMULINK. The stiff integration scheme is needed due to the high equivalent stiffness in the pre-sliding part of the friction model (18).

The path performance as predicted by the model will be compared to measurements carried out on the actual robot. In both the measurements and simulations, the path position is based on joint axis positions and the nominal kinematic model. In Fig. 8, the horizontal and vertical path deviations are shown. The results show that the agreement between the simulated prediction and the measurement is within 20 μm

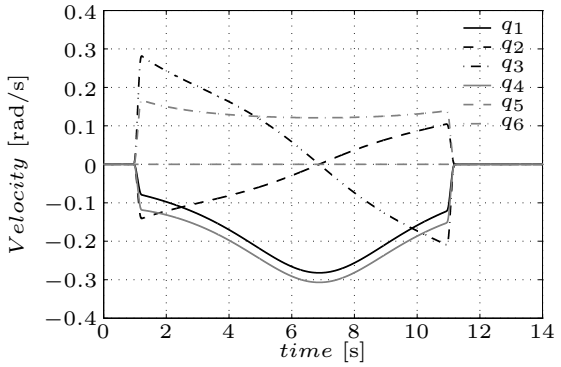


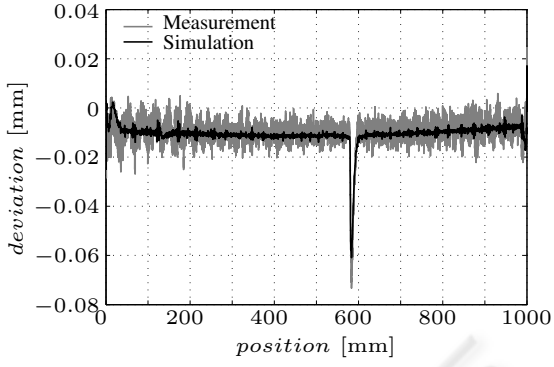
Figure 7: The joint velocity.

which is well within the range of accuracy required for laser welding. At about 600 mm on the trajectory, at the velocity reversal in joints 2 and 3, there is a jump of the friction force, resulting in a relatively large path error.

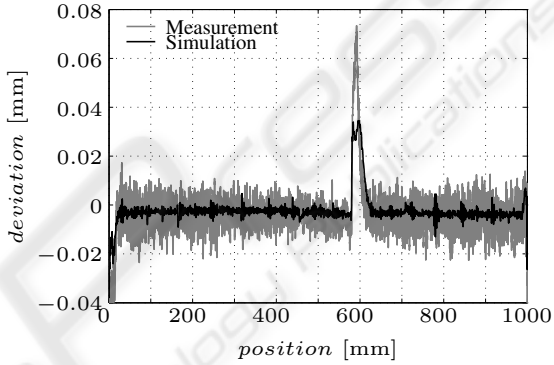
During the experiment the motor currents were recorded. In Fig. 9 the measured motor currents of joint 3 is compared with the simulated motor current as an example. The simulation shows good agreement with the measurements. It can be observed that the motor currents are predicted quite accurately in the sliding regime and near the velocity reversal. However, the model is unable to predict the stick-torques with a high level of accuracy. This can be explained by the fact that the stiction torque can be anywhere between $\tau_-^{(s)}$ and $\tau_+^{(s)}$ over a very small position range of tenths of μrad, due to the high equivalent stiffness $c^{(0)}$ in (18).

The required simulation time with the non-linear simulation model was about 70 minutes on a 2.4 GHz Pentium 4 PC, which indicates that it is computationally quite intensive. This is caused by the small time steps at which the manipulator configuration, see (1) and (2), needs to be solved by an iterative method. To overcome this drawback, the perturbation approach will be used next.

For the perturbation method the number of points along the trajectory N at which the system is linearised is taken to be 400. The simulation results of the perturbation method and the non-linear simulation have been compared. Fig 10 shows the path accuracy along the trajectory. There is no noticeable difference between the results at this scale. The error stays well below 10 μm and is mostly less than 2 μm. Also the agreement between the simulated motor currents for both the non-linear simulation and the perturbation method is good, especially when the manipulator is in motion. At rest, the torques in the stick or pre-sliding regime shows some differences as before. A significant reduction of simulation time by about a factor 10



a. Horizontal deviation



b. Vertical deviation

Figure 8: Simulated and measured path deviation.

has been achieved.

8 CONCLUSIONS

In this paper a realistic dynamic simulation of a rigid industrial robot has been presented. The following components are essential for the closed-loop simulation:

- A finite element-based robot model has been used to model the mechanical part of the robot. Robot identification has been applied to find the model parameters accurately.
- Furthermore, joint friction has been modelled using the LuGre friction model.
- Finally, a controller model has been identified and implemented.

The complete closed-loop model has been used for the simulation of the motion of the robot tip along a prescribed trajectory. The simulation results show good agreement with the measurements. It was found that a closed-loop simulation with the non-linear robot model was very time consuming due

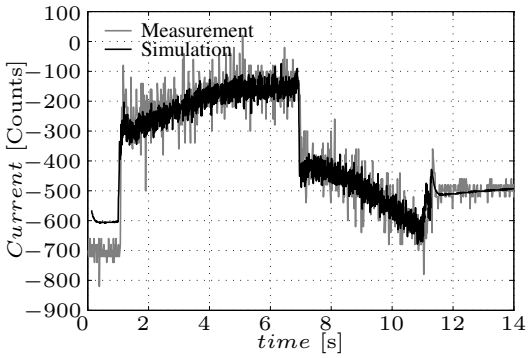


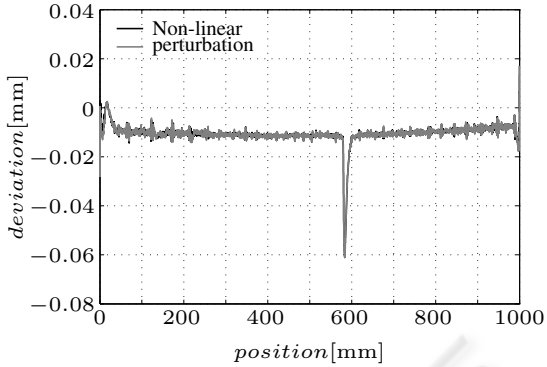
Figure 9: Measured and simulated motor current for joint 3.

to the small time step imposed by the discrete controller. Application of the perturbation method with the linearised model yields a substantial reduction in computer time without any loss of accuracy. The non-linear finite element model and the perturbation method can relatively easy be extended to account for flexibilities in the robot.

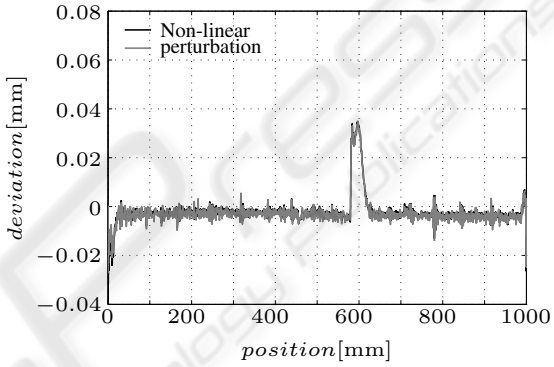
ACKNOWLEDGMENT

This work was carried out under the project number MC8.00073 in the framework of the Strategic Research programme of the Netherlands Institute for Metals Research in the Netherlands (http://www.nimr.nl/).

The authors acknowledge the work carried out on the identification of the controller, inertia parameters and friction parameters by W. Holterman, T. Hardeman and A. Kool, respectively.



a. Horizontal deviation



b. Vertical deviation

Figure 10: Simulated path deviation (non-linear versus perturbation method).

REFERENCES

Calafiore, G., Indri, M., and Bona, B. (2001). Robot dynamic calibration: Optimal excitation trajectories and experimental parameter estimation. *Journal of Robotic Systems*, 18(2):55–68.

Haessig, D. and Friedland, B. (1991). On the modeling and simulation of friction. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 113(3):354–362.

Jonker, J. (1990). A finite element dynamic analysis of flexible manipulators. *International Journal of Robotics Research*, 9(4):59–74.

Jonker, J. and Aarts, R. (2001). A perturbation method for dynamic analysis and simulation of flexible manipulators. *Multibody System Dynamics*, 6(3):245–266.

Jonker, J. and Meijaard, J. (1990). Spacar-computer program for dynamic analysis of flexible spatial mechanisms and manipulators. In Schiehlen, W., editor, *Multibody systems handbook*, pages 123–143. Springer-Verlag, Berlin.

Swevers, J., Ganseman, C., Schutter, J. D., and Brussel, H. V. (1996). Experimental robot identification using optimised periodic trajectories. *Mechanical Systems and Signal Processing*, 10(5):561–577.

Waiboer, R. R. (to be published, 2004). *Dynamic robot simulation and identification – for Off-Line Programming in robotised laser welding*. PhD thesis, University of Twente, The Netherlands.

Wit, C. d., Olsson, H., Åström, K., and Lischinsky, P. (1995). A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425.