# Closed-form kinematic and dynamic models of an industrial-like RRR robot

Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

# Closed-form Kinematic and Dynamic Models of an Industrial-like RRR Robot

Dragan Kostić[1], *Student Member, IEEE*, Ron Hensen[2], Bram de Jager[3], and Maarten Steinbuch[4], *Member, IEEE*
Dynamics and Control Technology Group, Department of Mechanical Engineering,
Eindhoven University of Technology (TU/e), P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Phone: +31 40 247 5730, Fax: +31 40 246 1418
[1]D.Kostic@tue.nl, [2]R.H.A.Hensen@tue.nl, [3]A.G.de.Jager@wfw.wtb.tue.nl, and [4]M.Steinbuch@tue.nl

***Abstract*—This paper presents closed-form kinematic and dynamic models of a robot with three rotational degrees of freedom. The derivation of the models and estimation of their parameters are explained. Relevancy of the models is investigated with a writing task. Validation results, obtained by simulation and experiment, establish correctness of the models and illuminate their practical benefits.**

***Key Words*—Kinematics, Dynamics, Modeling.**

## I. INTRODUCTION

THE essential purpose of industrial robots is execution of prescribed motions in 3-dimensional space. Fast and accurate movements require appropriate models of both robot kinematics [1] and dynamics [2]. A kinematic model enables the calculation of joint motions corresponding to a given trajectory of the robot's end-effector. A dynamic model provides information about the control inputs that should be applied to the joint actuators in order to achieve the desired motions of the robot. It also shows how particular dynamical phenomena, such as inertial, Coriolis/centripetal, gravitational and friction effects, influence the robot behavior in a given motion task [3].

An industrial-like robot with three revolute joints, an RRR-robot, is used as a test bed for a variety of non-linear control laws [4,5]. In particular, attention is focused on control laws based on a computed torque method [6]. It is used to resolve non-linear plant dynamics and to enable a simplified design of motion controllers. Recent advances in control of underactuated robotic systems [7] additionally stimulate use of the computed torque method. The basic prerequisite of this method is a good knowledge of the dynamical model and its parameters. Therefore, modeling of the RRR-robot dynamics and estimation of model parameters become vital tasks to be solved. To facilitate the definition of motion tasks, a kinematic model of the RRR-robot should be provided, too.

The goal is to derive both kinematic and dynamic models in closed-form and thus make the computation of kinematics and dynamics more efficient. A given accuracy of both forward and inverse kinematics solutions [1] is achieved much faster using the closed-form representation but using common numerical techniques [3]. As such, closed-form solutions of the kinematics are preferable for real time control. A closed-form representation of the dynamics enables an explicit analysis of each dynamical effect, and a direct evaluation of its influence on robot behavior. Direct insight into the model structure makes adequate compensation for each nonlinear aspect of the dynamics easier. For example, a closed-form model of gravitational effects can be used in the compensation of nonlinear gravitational loads. Although an implementation of the dynamics in explicit closed-form form requires more computational efforts compared with commonly used recursive techniques [1,2], its use for motion control is possible due to the computational power of modern digital controllers. Moreover, it ensures accurate calculations and thus it is suitable for practical use. Unfortunately, the derivation of closed-form models in compact form, particularly dynamic models, usually is not a straightforward task, but it demands a series of operations accompanied by a permanent check for errors and simplification of intermediate results. Hence, it is valuable to report any contribution in this area.

When a model is available, the next step is estimation of its parameters. Kinematic parameters, such as, link lengths, twist angles and link offsets, are usually known with better accuracy than inertial ones: link masses, location of respective centers of gravity and moments of inertia [8]. The kinematic parameters are either directly provided by the manufacturer, or can be achieved by direct measurements on the robot. Correct values of the dynamic parameters are rarely provided directly by a manufacturer, but they should be estimated from applied excitations (control torques) and measured responses (joint positions, velocities and accelerations) [9,10]. For the RRR robot, the kinematic parameters are derived from manufacturer data, while the inertial ones are estimated using experimental data. A detailed description of the estimation procedure is given in [11]. Here we only report results using a neural-network friction model [12] rather than using the LuGre model [13], as it has been done in [11].

In this paper we do not treat the estimation, but we focus on model validation. It should show the relevancy of the models and estimated parameters. A writing task [14] is particularly suitable for that purpose. It features non-uniform joint motions and imposes significant demands on the dynamics, such as high accelerations and inertial loads [15]. Hence, a proper behavior of the models in this task improves our confidence in their correctness.

The experimental set-up is described in Section II. A closed-form kinematic model is given in Section III. Section IV presents a closed-form formulation of the robot dynamics. The model validation with a writing task is described in Section V. Final remarks are given at the end.

0-7803-7272-7/02/$17.00 © 2002 IEEE

## II. Experimental Set-up

Our robot, shown in Fig. 1, has three revolute degrees of freedom (dof): waist, shoulder and elbow. They are implemented in an anthropomorphic manner, which is a usual kinematic architecture for industrial robots [3]. Each joint has an infinite range of motion thanks to the use of slip-rings for the transfer of power and sensor signals. Joints are actuated by direct-drive brushless DC motors with electronic commutation. Lack of gear-heads and counter-weights complicates motion control, as they may reduce nonlinear effects [4]. The motion controllers are implemented on a PC-based hardware/software platform that enables high flexibility in evaluation of a diversity of control laws. Current invertors are used to amplify control signals before they are applied to the motors.

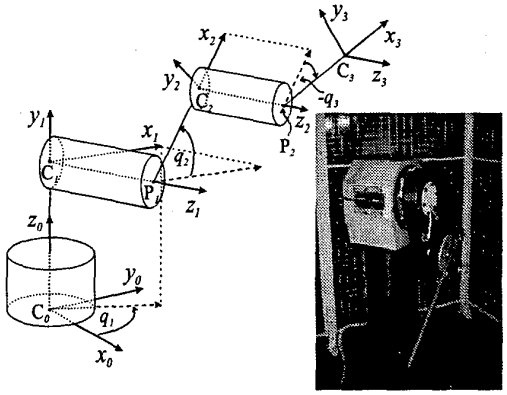


Fig. 1. RRR robot experimental set-up

## III. Robot Kinematics in Closed-form

To model the robot kinematics we make use of the kinematic diagram also presented in Fig. 1. It helps us to determine a minimal kinematic parameterization of the robot, according to a well-known Denavits-Hartenberg's (DH) notation [1]. Using it we establish Table 1, that contains DH parameters: twist angles, $\alpha_i$, link lengths, $a_i$, joint displacements, $q_i$, and link offsets, $d_i$.

TABLE I: DH Parameters of RRR-robot

| dof (=*i*) | Link twist ($=\alpha_i$) | Link length ($=a_i$) | Joint angle ($=q_i$) | Link offset ($=d_i$) |
|---|---|---|---|---|
| 1 | $\pi/2$ | 0 | $q_1$ | $C_0C_1$ |
| 2 | 0 | $P_1C_2$ | $q_2$ | $C_1P_1$ |
| 3 | 0 | $P_2C_3$ | $q_3$ | $C_2P_2$ |

In the following, we consider two spaces:

- *operational*, induced by tip Cartesian coordinates:

$$\mathbf{x} = [x \; y \; z]^T, \tag{1}$$

- *configuration*, induced by joint motions:

$$\mathbf{q} = [q_1 \; q_2 \; q_3]^T. \tag{2}$$

The homogenous transformation defining position and orientation of the end-effector, given the configuration variables $q_1$, $q_2$ and $q_3$, has a standard form [1]:

$$\mathbf{T} = \begin{bmatrix} \mathbf{O(q)} & \mathbf{x(q)} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}. \tag{3}$$

Elements of the orientation matrix $\mathbf{O}$ are as follows:

$$o_{11} = \cos q_1 \cos(q_2 + q_3), \; o_{12} = -\cos q_1 \sin(q_2 + q_3),$$
$$o_{13} = \sin q_1, \; o_{21} = \sin q_1 \cos(q_2 + q_3),$$
$$o_{22} = -\sin q_1 \sin(q_2 + q_3), \; o_{23} = -\cos q_1,$$
$$o_{31} = \sin(q_2 + q_3), \; o_{32} = \cos(q_2 + q_3), \; o_{33} = 0. \tag{4}$$

Cartesian coordinates of the robot's tip are:

$$x = \cos q_1 (a_3 \cos(q_2 + q_3) + a_2 \cos q_2) + (d_2 + d_3) \sin q_1,$$
$$y = \sin q_1 (a_3 \cos(q_2 + q_3) + a_2 \cos q_2) - (d_2 + d_3) \cos q_1,$$
$$z = a_3 \sin(q_2 + q_3) + a_2 \sin(q_2) + d_1. \tag{5}$$

The inverse kinematics (mapping from the operational to configuration space) is formulated in closed-form:

$$q_1 = \operatorname{asin} \frac{x(d_2 + d_3) + y\sqrt{x^2 + y^2 - (d_2 + d_3)^2}}{x^2 + y^2},$$

$$p_{wh} = \sqrt{(x - (d_2 + d_3)\sin q_1)^2 + (y + (d_2 + d_3)\cos q_1)^2},$$

$$p_{wv} = z - d_1,$$

$$q_3 = \operatorname{atan} \frac{\pm\sqrt{1 - \left(\dfrac{p_{wh}^2 + p_{wv}^2 - a_2^2 - a_3^2}{2a_2a_3}\right)^2}}{\dfrac{p_{wh}^2 + p_{wv}^2 - a_2^2 - a_3^2}{2a_2a_3}},$$

$$q_2 = \operatorname{atan} \frac{(a_2 + a_3 \cos q_3) p_{wv} - a_3 \sin q_3 p_{wh}}{(a_2 + a_3 \cos q_3) p_{wh} + a_3 \sin q_3 p_{wv}}. \tag{6}$$

The closed-form representation of the inverse kinematics has at least three advantages: (*i*) it enables direct recognition of "irregular" robot configurations, such as kinematic singularities; (*ii*) by specifying a sign in the fourth formula of (6) we may explicitly control the actual posture [3] – the robot can reach each point either with the elbow up (- sign) or with the elbow-down (+ sign); (*iii*) it requires far less computational efforts with respect to equivalent numerical methods [1,3]. The last property will be verified by an example presented in Section V.

## IV. Robot Dynamics in Closed-form

A dynamic model of the robot is derived using the Lagrange-Euler method [1], and has a standard form:

$$\mathbf{D(q)}\ddot{\mathbf{q}} + \mathbf{c(q,\dot{q})} + \mathbf{h(q)} = \boldsymbol{\tau}, \tag{7}$$

where $\mathbf{q}$ is defined by (2), $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are vectors of joint velocities and accelerations, respectively, $\mathbf{D}$ is a 3×3 inertia matrix, $\mathbf{c}$ and $\mathbf{g}$ are 3×1 vectors of Corio-

lis/centripetal and gravitational effects, and $\boldsymbol{\tau}$ is a 3×1 vector of joint torques. The current invertors provide a linear static relation between input voltages $u_i^n$ (excitations) and output torques of motors. This means:

$$u_i^n = (\tau_i + J_{m,i}\ddot{q}_i)/k_{t,i} + u_i^f,\ (i=1,2,3), \tag{8}$$

where $u_i^f$, $k_{t,i}$ and $J_{m,i}$ denote friction voltage, torque constant and a total rotor moment of inertia of the $i$th motor, respectively. Total moment of inertia is equal to the moment of inertia of a rotor itself, plus moments of inertia of all attached items, such as encoders, slip-rings, etc.

A number of possibilities is available for modeling the friction [16]. Traditionally, only Coulomb and a viscous friction effects are taken into account [9,10], but they are not sufficient to describe the friction. Therefore, more sophisticated friction models are required. For the RRR robot satisfactory results are achieved by adopting the LuGre friction model [13]. It offers a nice compromise between mathematical complexity and a number of friction phenomena (such as Stibeck effect, frictional lag, etc.). Unfortunately, the estimation of its parameters is not time efficient, since it demands either a series of individual experiments accompanied with intermediate processing of measured signals [16], or a two-step procedure that combines time-domain with frequency domain data processing [11]. Although a properly estimated LuGre model can be used for high quality friction compensation [17], a long estimation time might be a problem for its frequent practical use, particularly when regular calibration of a dynamic model is required. An elegant way to improve time efficiency, maintaining a reasonable quality of the friction model, is to use a neural-network friction model [12]:

$$u_i^f = \sum_{k=1}^{3} f_{ik}\left(1 - \frac{2}{e^{2w_{ik}\dot{q}_i}+1}\right) + b_i\dot{q}_i. \tag{9}$$

This model enables appropriate reconstruction of friction effects and has as simple implementation. The coefficients $f_{ik}$, $w_{ik}$ and $b_i$ can be easily estimated using the Extended Kalman Filter technique. If the coefficients $w_{ik}$ are fixed, then the neural-network model allows a parameterization linear in the remaining friction coefficients, enabling use of much faster least-squares estimation techniques [9,10]. These techniques are common for estimation of robot inertial parameters.

A dynamic model can be represented linearly in a minimum set of identifiable dynamic parameters, referred to as a base parameter set (BPS) [8]. Elements of the BPS are combinations of inertial parameters: link masses ($m_i$), Cartesian coordinates of link gravity centers ($x_i, y_i, z_i$), principle moments of inertias ($I_{xx,i}, I_{yy,i}, I_{zz,i}$), and products of inertias ($I_{xy,i}, I_{yz,i}, I_{xz,i}$).

For the RRR robot, a linear parameterization of a dynamic model has the form:

$$\mathbf{u}^n = \mathbf{k}_t^{-1}\mathbf{R}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\mathbf{p} + \mathbf{u}^f, \tag{10}$$

where

$$\mathbf{k}_t = \mathrm{diag}[k_{t,1}\ k_{t,2}\ k_{t,3}], \tag{11}$$

$\mathbf{R}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) \in \mathbb{R}^{3\times 16}$ is a regression matrix and $\mathbf{p} \in \mathbb{R}^{16\times 1}$ is a vector of BPS elements. The elements of $\mathbf{p}$ and $\mathbf{R}$ are presented in closed-form in (12) and (13), respectively.

These expressions reveal the complexity of the robot dynamics, but also enable independent analysis of particular dynamical effects. With elements of (13) containing the gravitational constant $g$ we may assemble gravitational terms, with elements containing joint accelerations we may recover inertial terms, while the remaining elements define Coriolis/centripetal terms. These terms can also be used for on-line compensation of the corresponding dynamic phenomena. A prerequisite for such use of a dynamic model is the knowledge of correct values of the model parameters, including friction. In the Introduction it has been already pointed out that proper parameter values are primarily achieved by estimation. A survey of estimation techniques is given in [9,10]. A systematic estimation procedure of both friction and BPS elements is suggested in [11]. In this paper, a procedure from [11] is made even faster, since here we consider the neural-network friction model (9) instead of the LuGre model. Estimation of the friction coefficients is carried out for each joint independently, following the procedure suggested in [12]. After that, optimal excitations $\mathbf{u}^n$ determined in [11] are applied to the joint actuators, and the resulting joint positions, velocities and accelerations are recorded. Friction effects $\mathbf{u}^f$ are reconstructed by (9), and then subtracted from $\mathbf{u}^n$. The term $\mathbf{u}^n - \mathbf{u}^f$ is linear in $\mathbf{p}$, as is obvious from (10). This linear relationship is exploited for estimation of $\mathbf{p}$ using the least-squares method. The quality of the model should be verified in simulations and experimentally. In the next section we do this with a writing task.

## V. Model Verification

Writing and drawing tasks commonly require complex motions and accordingly they could be very demanding from the viewpoint of imposed dynamic loads. An extensive survey of research related with human writing is given in [14]. In the same reference a kinematic analysis of one representative writing task is presented. It consists of writing a sequence of letters presented in Fig. 2. This sequence can be mathematically described in closed-form, allowing specification of an arbitrary velocity profile. Freedom in defining velocity profiles enables quite demanding dynamic tasks, which is confirmed by analysis carried out in [15]. These properties can be exploited for verification of the kinematic and dynamic models. If the

$$p_1 = a_2^2 m_3 + a_2 m_2 (a_2 + 2x_2) - I_{xx,2} + I_{yy,2},\ p_2 = -2(a_2 m_2 y_2 + I_{xy,2}),\ p_3 = m_3(a_3 + x_3),\ p_4 = m_3 y_3,$$
$$p_5 = a_3 m_3 (a_3 + 2x_3) - I_{xx,3} + I_{yy,3},\ p_6 = 2(a_3 m_3 y_3 + I_{xy,3}),$$
$$p_7 = 2d_2 m_2 z_2 + d_2^2 m_2 + 2(d_2 + d_3) m_3 z_3 + (d_2 + d_3)^2 m_3 + I_{yy,1} + I_{xx,2} + I_{xx,3} + J_{m,1}\ p_8 = -(d_2 + d_3) m_3 y_3 - I_{yz,3},$$
$$p_9 = -a_3 m_3 (z_3 + d_2 + d_3) - (d_2 + d_3) m_3 x_3 - I_{xz,3},\ p_{10} = -d_2 m_2 y_2 - I_{yz,2},$$
$$p_{11} = -a_2 (m_3 (z_3 + d_2 + d_3) + m_2 (d_2 + z_2)) - d_2 m_2 x_2 - I_{xz,2},\ p_{12} = (a_2^2 + a_3^2 + 2a_3 x_3) m_3 + a_2 m_2 (a_2 + 2x_2) + I_{zz,2} + I_{zz,3} + J_{m,2},$$
$$p_{13} = a_3 m_3 (a_3 + 2x_3) + I_{zz,3},\ p_{14} = p_{13} + J_{m,3},\ p_{15} = m_2 (x_2 + a_2) + m_3 a_2,\ p_{16} = m_2 y_2. \quad (12)$$

$$r_{1,1} = \ddot{q}_1 \cos^2 q_2 - \dot{q}_1 \dot{q}_2 \sin(2q_2),\ r_{1,2} = 0.5\ddot{q}_1 \sin(2q_2) + \dot{q}_1 \dot{q}_2 \cos(2q_2),$$
$$r_{1,3} = 2a_2 [(\ddot{q}_1 \cos(q_2 + q_3) - \dot{q}_1 \dot{q}_3 \sin(q_2 + q_3)) \cos q_2 - \dot{q}_1 \dot{q}_2 \sin(2q_2 + q_3)],$$
$$r_{1,4} = -2a_2 [(\ddot{q}_1 \sin(q_2 + q_3) + \dot{q}_1 \dot{q}_3 \cos(q_2 + q_3)) \cos q_2 + \dot{q}_1 \dot{q}_2 \cos(2q_2 + q_3)],\ r_{1,5} = \ddot{q}_1 \cos^2(q_2 + q_3) - \dot{q}_1 (\dot{q}_2 + \dot{q}_3) \sin(2q_2 + 2q_3),$$
$$r_{1,6} = -0.5\ddot{q}_1 \sin(2q_2 + 2q_3) - \dot{q}_1 (\dot{q}_2 + \dot{q}_3) \cos(2q_2 + 2q_3),\ r_{1,7} = \ddot{q}_1,\ r_{1,8} = (\ddot{q}_2 + \ddot{q}_3) \cos(q_2 + q_3) - (\dot{q}_2 + \dot{q}_3)^2 \sin(q_2 + q_3),$$
$$r_{1,9} = (\ddot{q}_2 + \ddot{q}_3) \sin(q_2 + q_3) + (\dot{q}_2 + \dot{q}_3)^2 \cos(q_2 + q_3),\ r_{1,10} = -\dot{q}_2^2 \sin q_2 + \ddot{q}_2 \cos q_2,\ r_{1,11} = \dot{q}_2^2 \cos q_2 + \ddot{q}_2 \sin q_2,$$
$$r_{2,1} = 0.5\dot{q}_1^2 \sin(2q_2),\ r_{2,2} = -0.5\dot{q}_1^2 \cos(2q_2),$$
$$r_{2,3} = 2a_2 [(\ddot{q}_2 + 0.5\ddot{q}_3) \cos q_3 - (0.5\dot{q}_3^2 + \dot{q}_2 \dot{q}_3) \sin q_3 + 0.5\dot{q}_1^2 \sin(2q_2 + q_3)] + g\cos(q_2 + q_3),$$
$$r_{2,4} = 2a_2 [-(\ddot{q}_2 + 0.5\ddot{q}_3) \sin q_3 - (0.5\dot{q}_3^2 + \dot{q}_2 \dot{q}_3) \cos q_3 + 0.5\dot{q}_1^2 \cos(2q_2 + q_3)] - g\sin(q_2 + q_3),\ r_{2,5} = 0.5\dot{q}_1^2 \sin(2q_2 + 2q_3),$$
$$r_{2,6} = 0.5\dot{q}_1^2 \cos(2q_2 + 2q_3),\ r_{2,8} = \ddot{q}_1 \cos(q_2 + q_3),\ r_{2,9} = \ddot{q}_1 \sin(q_2 + q_3),\ r_{2,10} = \ddot{q}_1 \cos q_2,\ r_{2,11} = \ddot{q}_1 \sin q_2,\ r_{2,12} = \ddot{q}_2,$$
$$r_{2,13} = \ddot{q}_3,\ r_{2,15} = g\cos q_2,\ r_{2,16} = -g\sin q_2,\ r_{3,3} = a_2 [(0.5\dot{q}_1^2 + \dot{q}_2^2) \sin q_3 + \cos q_3 \ddot{q}_2 + 0.5\dot{q}_1^2 \sin(2q_2 + q_3)] + g\cos(q_2 + q_3),$$
$$r_{3,4} = a_2 [(0.5\dot{q}_1^2 + \dot{q}_2^2) \cos q_3 - \ddot{q}_2 \sin q_3 + 0.5\dot{q}_1^2 \cos(2q_2 + q_3)] - g\sin(q_2 + q_3),\ r_{3,5} = 0.5\dot{q}_1^2 \sin(2q_2 + 2q_3),$$
$$r_{3,6} = 0.5\dot{q}_1^2 \cos(2q_2 + 2q_3),\ r_{3,8} = \ddot{q}_1 \cos(q_2 + q_3),\ r_{3,9} = \ddot{q}_1 \sin(q_2 + q_3),\ r_{3,13} = \ddot{q}_2,\ r_{3,14} = \ddot{q}_3,$$
$$r_{1,12} = r_{1,13} = r_{1,14} = r_{1,15} = r_{1,16} = r_{2,7} = r_{2,14} = r_{3,1} = r_{3,2} = r_{3,7} = r_{3,10} = r_{3,11} = r_{3,12} = r_{3,15} = r_{3,16} = 0. \quad (13)$$

models behave properly in the writing task, we may expect equivalent behavior for an arbitrary motion tasks.

Here, the tip of the robot should move along the planar path shown on the left hand side in Fig. 2 ($y$ coordinate is fixed), with a continuos velocity profile

$$v(t) = \sqrt{\dot{x}^2(t) + \dot{z}^2(t)} \quad (14)$$

presented on the right hand side in Fig. 2. The profile has fast and slow phases.

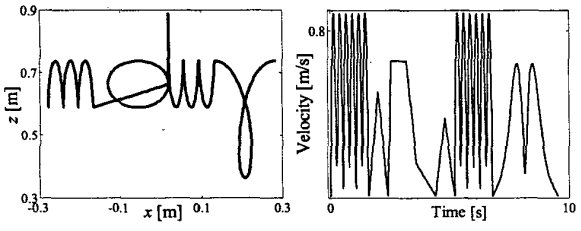


Fig. 2. Reference operational tip motion (left) and velocity (right)

For this reference trajectory, the inverse kinematics solution determines joint motions shown in Fig. 3. The joint velocities are presented in Fig. 4. These figures confirm our expectation that the specified writing task imposes non-uniform joint motions, with significant velocity and acceleration levels. These joint motions are compared with an inverse kinematics solutions calculated using a numeric method, implemented in the Robotics Toolbox for Matlab [18]. Guaranteed accuracy of the numeric solutions is below $10^{-10}$ rad. Differences between the closed-form and the Robotics Toolbox solutions are within the accuracy of the latter ones, as by the error diagrams in Fig. 5. This verifies the correctness of the equations (6). Moreover, the routine of the Robotics Toolbox requires 2783 Matlab's floating-point operations (flops) for each position of the considered motion, while the closed-form equations require only 102 flops. This confirms a well-known property that computations of inverse kinematics using the model in closed-form provide given accuracy in the most time-efficient way [3]. This is why the closed-form kinematic models are preferable for real-time applications.

We verify the dynamic model using simulations and experiments. Control torques corresponding to the reference configuration motions and velocities (Figs. 3,4) are calculated using the model in closed-form, and then compared with torques generated using a numeric routine implemented in the Robotics Toolbox. The differences are caused by round-off effects, as shown in Fig. 6. This verifies correctness of the dynamic equations in closed-form. Moreover, due to non-optimal implementation of the Robotics Toolbox routine, the closed-form model also

requires lower computation effort, as it takes 393 flops for each position of the considered motion, while the Robotics Toolbox requires 1050 flops. It means that simulations of the robot dynamics can be carried out much faster using the model in closed-form. Finally, the closed-form formulation of the robot dynamics admits a real-time experimental implementation, as verified next.

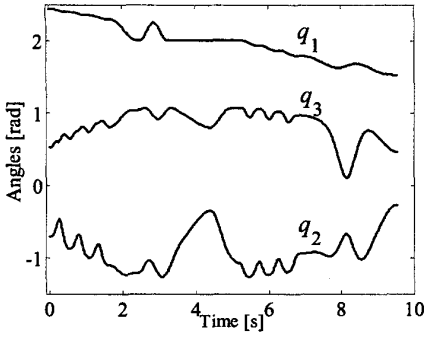


Fig. 3. Reference positions in configuration space

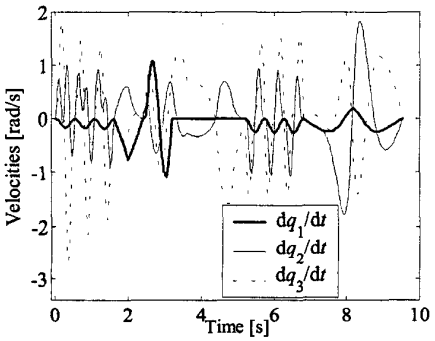


Fig. 4. Reference velocities in configuration space

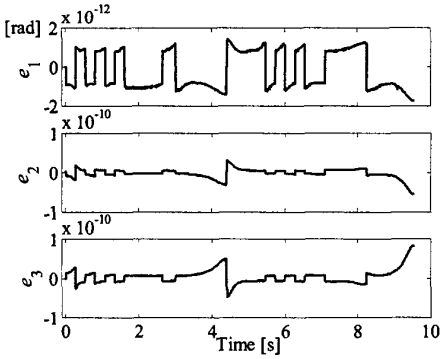


Fig. 5. Differences between closed-form and numeric inverse kinematics solutions

Consider a computed torque controller [19]:

$$u_i(t) = u_i^n(t) + u_i^c(t) = u_i^n(\mathbf{q}_{ref}(t), \dot{\mathbf{q}}_{ref}(t), \ddot{\mathbf{q}}_{ref}(t)) + \\ + k_{p,i}(q_{ref,i}(t) - q_i(t)) + k_{d,i}(\dot{q}_{ref,i}(t) - \dot{q}_i(t)) \quad (15)$$

($i = 1,2,3$). Adopt for reference motions $q_{ref,i}$ and velocities $\dot{q}_{ref,i}$ those shown in Figs. 3,4. The term $u_i^n$ is nominal (feedforward) control component, used for compensation of nonlinear dynamics. The term $u_i^c$ is produced by a PD controller. In the first experiment we exclude $u_i^n$ and control the joint actuators using the PD controller only. Then we tune gains $k_{p,i}$ and $k_{d,i}$ to ensure stability, but allow a visible discrepancy between the desired and measured operational coordinates, as obvious from Fig. 7.

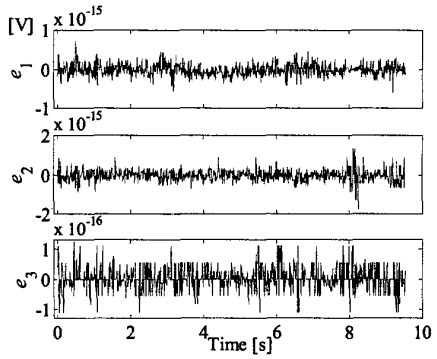


Fig. 6. Differences between closed-form and numeric inverse dynamics solutions

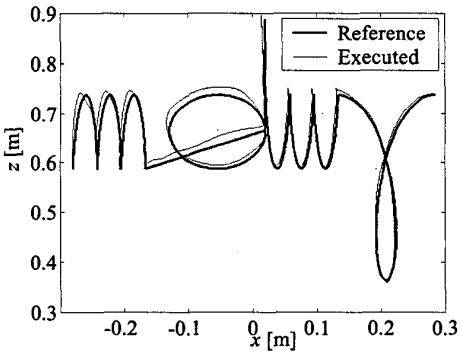


Fig. 7. Reference sequence of letters and a sequence executed using a PD controller only

Enabling $u_i^n$ in (15) significantly reduces the difference between the desired and executed sequence of letters, as obvious from Fig. 8. The signals $u_2^c$ and $u_2^n$, shown in Fig. 9, reveal a dominant role of the nominal component $u_2^n$ determined from the model. The PD controller serves just for error correction, that inevitable occurs in real-life applications. As a similar effect is observed in the other two joints, we may conclude that the model (10)-(13) is a relevant representation of the RRR-robot's dynamics. Dominant contribution of the nominal controls is also confirmed by their higher RMS levels (in V) than the RMS levels of the PD controls:

$$\mathbf{u}^n = [0.25\ 1.98\ 0.78]^T \text{ vs. } \mathbf{u}^c = [0.11\ 0.24\ 0.17]^T. \quad (16)$$

In the plot of the PD control component, presented in Fig. 9, we may notice several peaks. They indicate that the estimated model still cannot accurately predict rapid changes of the reference motions. The rapid changes occur when the letters 'ω', 'm' and 'γ' are written. For their accurate writing, a suitable contribution of the PD control component is required. This fact, however, does not restrict practical applicability of the estimated model. It certainly dominantly contributes to the total control effort, as confirmed by the RMS values given in (16).

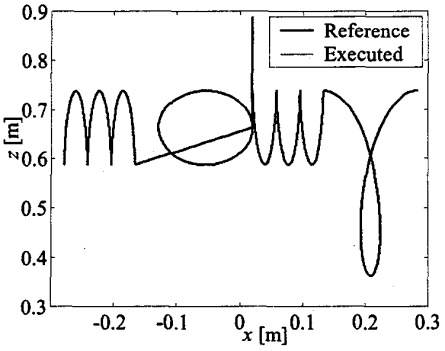


Fig. 8. Reference sequence of letters and a sequence executed using PD and nominal control inputs together

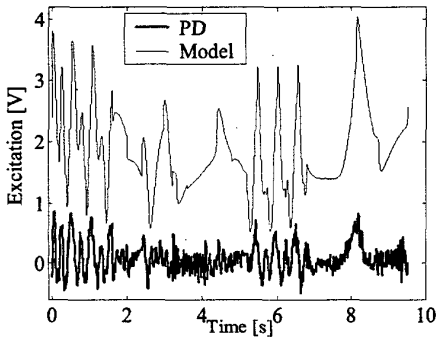


Fig. 9. Model and PD components of the control input applied to the second joint

## VI. Conclusion

Kinematics and dynamics of an industrial-like robot with three rotational degrees of freedom are considered in the paper. Models of the robot kinematics and dynamics are derived in closed-form, and are presented in full detail. We verified several advantageous properties of these models: (*i*) computation efficiency, which is convenient for on-line applications, (*ii*) numeric accuracy, (*iii*) direct recognition of "irregular" robot configurations (e.g., kinematic singularities), (*iv*) possibility to control the operational motions of the robot's tip, which increases flexibility in defining motion tasks, (*v*) straightforward linear parameterization of a dynamic model, which simplifies estimation of inertial parameters, (*vi*) use of the dynamic model for compensation of dynamic effects. We also verified that a writing task could be used for firm verification of the quality of the models.

## References


[1] K.S. Fu, R.C. Gonzales, C.S.G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, London, 1987.

[2] M. Vukobratović, V. Potkonjak, *Dynamics of Manipulation Robots: Theory and Application*, Springer, Berlin, 1982.

[3] L. Sciavicco, B. Siciliano, *Modeling and Control of Robot Manipulators*, McGraw-Hill, London, 1996.

[4] B. van Beek, B. de Jager, "RRR-Robot Design: Basic Outlines, Servo Sizing, and Control," *Proc. IEEE Int. Conf. Control Applications*, pp. 36-41, Hartford, USA, 1997.

[5] B. van Beek, B. de Jager, "An Experimental Facility for Nonlinear Robot Control," *Proc. IEEE Int. Conf. Control Applications*, pp. 668-673, Hawai'i, USA, 1999.

[6] J.J.E. Slotine, W. Li, *Applied Nonlinear Control*, Prentice Hall, London, 1991.

[7] H. Arai, K. Tanie, N. Shiroma, "Nonholonomic Control of a Three-DOF Planar Underactuated manipulator," *IEEE Trans. on Rob. Autom.*, Vol. 14, No. 5, pp. 681-695, 1998.

[8] H. Mayeda, K. Yoshida, K. Osuka, "Base Parameters of Manipulator Dynamic Models," *IEEE Trans. Rob. Autom.*, Vol. 6, No. 3, pp. 312-321, 1990.

[9] J. Swevers, C. Ganseman, D.B. Tukel, J. de Schutter, H. van Brussel, "Optimal Robot Excitation and Identification," *IEEE Trans. Rob. Autom.*, Vol. 13, No. 5, pp. 730-740, 1997.

[10] G. Calafiore, M. Indri, B. Bona, "Robot Dynamic Calibration: Optimal Excitation Trajectories and Experimental Parameter Estimation," *J. of Rob. Sys.*, Vol. 18, No. 2, pp. 55-68, 2001.

[11] D. Kostić, R. Hensen, B. de Jager, M. Steinbuch, "Modeling and Identification of an RRR-robot," *Proc. IEEE Conf. Dec. Control*, pp. 1144-1149, Orlando, FL, December, 2001.

[12] R.H.A. Hensen, G.Z. Angelis, M.J.G. van de Molengraft, A.G. de Jager, J.J. Kok, "Grey-box Modelling of Friction: An Experimental Case Study," *Europ. J. Contr.*, Vol. 6, No. 3, pp. 258-267, 2000.

[13] C. Canudas de Wit, H. Olsson, K.J. Åström, P. Lischinsky, "A New Model for Control of Systems With Friction," *IEEE Trans. on Autom. Control.*, Vol. 40, No. 3, pp. 419-425, 1995.

[14] V. Potkonjak, M. Popović, M. Lazarević, J. Sinanović, "Redundancy Problem in Writing: From Human to Anthropomorphic Robot Arm," *IEEE Trans. Sys., Man, Cyb., Part B*, Vol. 28, No. 6, pp. 790-805, 1998.

[15] V. Potkonjak, S. Tzafestas, D. Kostić, "Concerning the Primary and Secondary Objectives in Robot Task Definition – The "Learn From Humans" Principle," *Math. and Comp. in Simul.*, Vol. 54, No. 1-3, pp. 145-157, 2000.

[16] H. Olsson, K.J. Åström, C.C. de Wit, M. Gäfvert, P. Lischinsky, "Friction Models and Friction Compensation," *Eur. J. Contr.*, Vol. 4, No. 3, pp. 176-195, 1998.

[17] J. Swevers, F. Al-Bender, C.G. Ganseman, T. Projogo, "An Integrated Friction Model Structure With Improved Presliding Behavior for Accurate Friction Compensation," *IEEE Trans. Autom. Control.*, Vol. 45, No. 4, pp. 675-686, 2000.

[18] P.I. Corke, "A Robotics Toolbox for Matlab," *IEEE Robot. Autom. Mag.*, Vol. 3, No. 1, pp. 24-32, 1996.

[19] I.M.M. Lammerts, Adaptive Computed Reference Computed Torque Control of Flexible Manipulators, *Ph.D. Thesis*, Eindhoven University of Technology, 1993.