



TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 64, Issue IV, November, 2021

THE KINEMATIC MODEL OF FANUC LR Mate 100iB CONSIDERING THE GEOMETRIC ERRORS

Ovidiu-Aurelian DETEȘAN

Abstract: The methods of kinematic modeling of industrial robots, regularly used in the literature, are strictly related to robot's nominal geometry and kinematics, the robot being considered as a system of rigid bodies, dimensionally and positionally perfect. This paper outlines the influence of dimensional and generalized coordinates errors on the operational velocities and accelerations, angular and linear, corresponding to FANUC LR Mate 100iB industrial robot. The dimensional-constructive errors and the errors of generalized coordinates are included in the equations of the direct geometric model, determined by the method of location matrices. The obtained data will be used as input data into the iterative algorithm of kinematic modeling. By substituting numerical values into the kinematic equations, the influence of these errors on the operational kinematic parameters is outlined.

Key words: articulated robot, geometric errors, kinematic errors, symbolic computation.

1. INTRODUCTION

The methods applied when modeling a robot, usually do not consider the geometric, kinematic, and dynamic errors [1]. Some simulators, as SimMEcROb [2] are complex applications that analyze various parameters of serial industrial robots, including precision, but they are difficult to be used because of the software requirements such as: operating system and other related applications that are required to be installed. A major drawback of pure numerical applications is that the results are numeric only and the algebraic form of the geometric, kinematic or dynamic model of the robot put into analysis cannot be determined.

The aim of this paper is to establish the equations of the kinematic model of an articulated industrial robot, by symbolic computations, considering the dimensional errors and generalized coordinates errors.

2. FANUC LR Mate 100iB

This study will focus on the robot FANUC LR Mate 100iB [3], used with didactic and research purpose at the Laboratory of Applied Mechanics in Robotics, Faculty of Industrial

Engineering, Robotics and Production Management, Technical University of Cluj-Napoca, Romania. It is an articulated robot of a small to medium size, having five degrees of freedom (fig. 1). The kinematic scheme is presented in fig. 2 and it will be one source of input data in the algorithm of kinematic modeling of the mechanical structure.

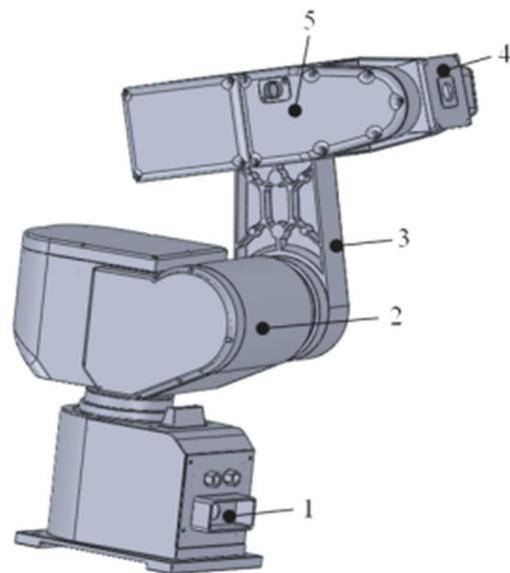


Fig. 1. FANUC LR Mate 100iB [4]

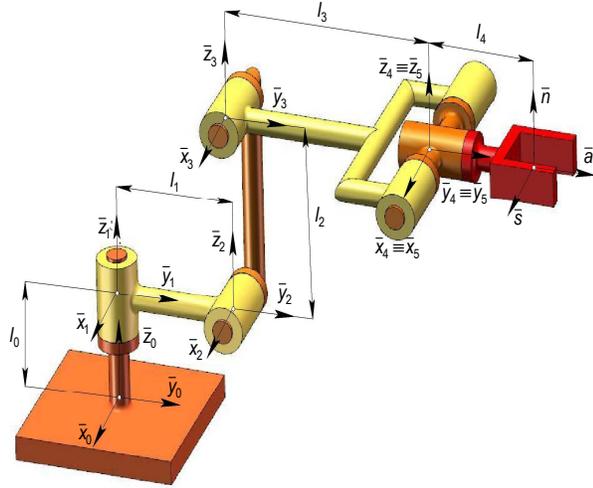


Fig. 2. FANUC LR Mate 100iB. Kinematic scheme [2]

3. THE ITERATIVE METHOD OF KINEMATIC MODELING, CONSIDERING THE GEOMETRIC ERRORS

One of the commonly used methods in kinematic modeling is the iterative method [2]. It is based on the use of position vectors, rotation matrices and the kinematic parameters corresponding to the previous link. The iterative method consists in traversing the kinematic chain of the robot from the fixed base {0} to the final effector (n) and determining by iterations the following kinematic parameters:

$$\{ {}^i \bar{k}_i, {}^i \bar{\omega}_i, {}^i \dot{\bar{\omega}}_i, {}^i \bar{v}_i, {}^i \dot{\bar{v}}_i, i = \overline{1, n} \}. \quad (1)$$

They characterize the motion of each link ($i = \overline{1, n}$) with respect to the fixed frame {0}. They are expressed in the frame { i } and have the following meanings:

${}^i \bar{k}_i$ – the unit vector of the kinematic axis (i);

${}^i \bar{\omega}_i, {}^i \dot{\bar{\omega}}_i$ – the angular velocity and acceleration of the link (i) with respect to the fixed frame {0}, expressed in the frame { i };

${}^i \bar{v}_i, {}^i \dot{\bar{v}}_i$ – the velocity and acceleration of the origin of the frame { i } with respect to the fixed frame {0}, expressed in the frame { i }.

The kinematic parameters related to the fixed base {0} are also considered, which have the following form:

$$\begin{aligned} {}^0 \bar{\omega}_0 &= [0 \ 0 \ 0]^T; & {}^0 \bar{v}_0 &= [0 \ 0 \ 0]^T; \\ {}^0 \dot{\bar{\omega}}_0 &= [0 \ 0 \ 0]^T; & {}^0 \dot{\bar{v}}_0 &= [0 \ 0 \ g]^T, \end{aligned} \quad (2)$$

Since this method will be used with the sole purpose of kinematic modeling, the last term (g) of the above equation will be equal to zero.

To determine the kinematic parameters (1), the mechanical structure of the robot is geometrically modeled, considering the constructive parameters and the generalized coordinate errors and the homogeneous transformation matrices: ${}^{i-1} [T]$, ${}^{i-1} [T]^{-1}$ and ${}^0 [T]$, for ($i = \overline{1, n+1}$) are determined. The calculation algorithm comprises, according to [2], the following matrix equations:

$${}^i \bar{\omega}_i = {}^{i-1} [R] {}^{i-1} \bar{\omega}_{i-1} + \begin{cases} \dot{q}_i \cdot {}^i \bar{k}_i, & \text{if } i = R \\ 0, & \text{if } i = T \end{cases} \quad (3)$$

$$\begin{aligned} {}^i \bar{v}_i &= {}^{i-1} [R] \{ {}^{i-1} \bar{v}_{i-1} + {}^{i-1} \bar{\omega}_{i-1} \times {}^{i-1} \bar{r}_i \} + \\ &+ \begin{cases} 0, & \text{if } i = R \\ \dot{q}_i \cdot {}^i \bar{k}_i, & \text{if } i = T \end{cases} \end{aligned} \quad (4)$$

$$\begin{aligned} {}^i \dot{\bar{\omega}}_i &= {}^{i-1} [R] {}^{i-1} \dot{\bar{\omega}}_{i-1} + \\ &+ \begin{cases} {}^{i-1} [R] {}^{i-1} \dot{\bar{\omega}}_{i-1} \times \dot{q}_i \cdot {}^i \bar{k}_i + \ddot{q}_i \cdot {}^i \bar{k}_i, & \text{if } i = R \\ 0, & \text{if } i = T \end{cases} \end{aligned} \quad (5)$$

$$\begin{aligned} {}^i \dot{\bar{v}}_i &= {}^{i-1} [R] \{ {}^{i-1} \dot{\bar{v}}_{i-1} + {}^{i-1} \dot{\bar{\omega}}_{i-1} \times {}^{i-1} \bar{r}_i + \\ &+ {}^{i-1} \bar{\omega}_{i-1} \times {}^{i-1} \bar{\omega}_{i-1} \times {}^{i-1} \bar{r}_i \} + \\ &+ \begin{cases} 0, & \text{if } i = R \\ 2 \cdot {}^i \bar{\omega}_i \times \dot{q}_i \cdot {}^i \bar{k}_i + \ddot{q}_i \cdot {}^i \bar{k}_i, & \text{if } i = T \end{cases} \end{aligned} \quad (6)$$

The skew-symmetric matrix associated to the vector $\bar{\omega}$ appears in the expressions (4)-(6), and it is expressed as:

$$(\bar{\omega} \times) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (7)$$

It will be computed by defining a function, described further.

If the kinematic parameters (3)-(6) are to be expressed in the fixed frame {0}, then the following transformation equations are used:

$$\begin{aligned} {}^0 \bar{v}_n &= {}^0 [R] \cdot {}^n \bar{v}_n; & {}^0 \dot{\bar{v}}_n &= {}^0 [R] \cdot {}^n \dot{\bar{v}}_n; \\ {}^0 \bar{\omega}_n &= {}^0 [R] \cdot {}^n \bar{\omega}_n; & {}^0 \dot{\bar{\omega}}_n &= {}^0 [R] \cdot {}^n \dot{\bar{\omega}}_n. \end{aligned} \quad (8)$$

The equations (8) represent the direct kinematic model (DKM), according to which the

operational kinematic parameters (linear and angular velocities and accelerations) were determined, which express the motion of the end effector with respect to the fixed frame {0}. The equations of DKM, being determined based on the rotation matrices and position vectors containing the geometric errors, will characterize the kinematic behavior of the real mechanical structure of the robot.

4. THE FUNCTION skew()

To define the skew-symmetric matrix of a vector $\bar{\omega}$ (7), the following MATLAB function will be defined:

skew(v) – returns the skew-symmetric matrix associated to the vector *v*.

Function file skew.m

```
function [ M ] = skew( v )
% M = skew(v)
% Returns the skew-symmetric matrix
% associated to the vector v
M = sym(zeros(3));
M(3,2) = v(1);
M(1,3) = v(2);
M(2,1) = v(3);
M(2,3) = -v(1);
M(3,1) = -v(2);
M(1,2) = -v(3);
end
```

5. PREPARING THE ENVIRONMENT. USING THE INPUT DATA

The script file *FANUC_kin_err.mlx* was created with the purpose of determining the equations of the direct kinematic model of FANUC robot, considering the geometric errors. The first section of the script handles the working environment preparation. With this regard, all the variables existing in the current MATLAB workspace are cleared. Since a *Live Script* is used here, there is no need to record the content of the *Command Window* associated to the current MATLAB session, as in [5]. The declaration of the symbolic variables *q1...q5* follows, representing the generalized coordinates, as well as the variables *l0...l4*, which are the geometric-constructive parameters.

Along them, the symbolic variables *dq1...dq5* are defined, representing the errors of the generalized parameters, also *dl0...dl4*, the dimensional parameters errors. The generalized velocities *q_dot_1, q_dot_2, q_dot_3, q_dot_4* and *q_dot_5* are then declared, as well as the generalized accelerations *q_ddot_1, q_ddot_2, q_ddot_3, q_ddot_4* and *q_ddot_5*.

All the results from the geometric model [5] are loaded from the data file *FANUC_geo_err.mat*.

The axes' unit vectors are then defined, according to (1) and the kinematic scheme (Fig. 2). The kinematic parameters related to the fixed base {0} are also defined with (2). Now, from the location matrices determined in [5], the rotation matrices and the position vectors are extracted, being used in the equations (3)-(6). The matrices of the inverse rotation are then obtained, considering that the inverse of a rotation matrix equals its transpose. The corresponding MATLAB code is reproduced below:

```
clear variables
syms q1 q2 q3 q4 q5 real
syms l0 l1 l2 l3 l4 real
syms dq1 dq2 dq3 dq4 dq5 real
syms dl0 dl1 dl2 dl3 dl4 real
syms q_dot_1 q_dot_2 q_dot_3...
      q_dot_4 q_dot_5 real
syms q_ddot_1 q_ddot_2 q_ddot_3...
      q_ddot_4 q_ddot_5 real
load FANUC_geo_err
% axes' unit vectors
k_bar_11 = sym([0; 0; 1])
k_bar_22 = sym([1; 0; 0])
k_bar_33 = sym([1; 0; 0])
k_bar_44 = sym([1; 0; 0])
k_bar_55 = sym([0; 1; 0])
% kinematic parameters of the
% fixed base {0} (2)
omega_bar_00 = sym([0; 0; 0])
v_bar_00 = sym([0; 0; 0])
omega_bar_dot_00 = sym([0; 0; 0])
v_bar_dot_00 = sym([0; 0; 0])
% rotation matrices extraction
R10 = T10(1:3,1:3)
R21 = T21(1:3,1:3)
R32 = T32(1:3,1:3)
R43 = T43(1:3,1:3)
R54 = T54(1:3,1:3)
R65 = T65(1:3,1:3)
```

```

R50 = T50(1:3,1:3)
R60 = T60(1:3,1:3)
% position vectors extraction
r10 = T10(1:3,4)
r21 = T21(1:3,4)
r32 = T32(1:3,4)
r43 = T43(1:3,4)
r54 = T54(1:3,4)
r65 = T65(1:3,4)
% matrices of inverse rotation: inv(R)=R'
R01 = R10'
R12 = R21'
R23 = R32'
R34 = R43'
R45 = R54'
R56 = R65'

```

6. THE KINEMATIC PARAMETERS IN THE FRAME {i}

By applying the iterative relations (3)-(6), the script determines the angular and linear operational velocities, as well as the angular and linear operational accelerations. The function *skew()*, defined above, is used in applying the equations (4)-(6).

```

% angular operational velocities (3)
omega_bar_dot_11 = R01*omega_bar_dot_00+...
    q_dot_1*k_bar_11
omega_bar_dot_22 = R12*omega_bar_dot_11+...
    q_dot_2*k_bar_22
omega_bar_dot_33 = R23*omega_bar_dot_22+...
    q_dot_3*k_bar_33
omega_bar_dot_44 = R34*omega_bar_dot_33+...
    q_dot_4*k_bar_44
omega_bar_dot_55 = R45*omega_bar_dot_44+...
    q_dot_5*k_bar_55
omega_bar_dot_66 = R56*omega_bar_dot_55
% linear operational velocities (4)
v_bar_11 = R01*(v_bar_dot_00+...
    skew(omega_bar_dot_00)*r10)
v_bar_22 = R12*(v_bar_dot_11+...
    skew(omega_bar_dot_11)*r21)
v_bar_33 = R23*(v_bar_dot_22+...
    skew(omega_bar_dot_22)*r32)
v_bar_44 = R34*(v_bar_dot_33+...
    skew(omega_bar_dot_33)*r43)
v_bar_55 = R45*(v_bar_dot_44+...
    skew(omega_bar_dot_44)*r54)
v_bar_66 = R56*(v_bar_dot_55+...
    skew(omega_bar_dot_55)*r65)
% angular operational accelerations(5)

```

```

omega_bar_dot_11 = R01*...
    omega_bar_dot_00+R01*skew(...
    omega_bar_dot_00)*q_dot_1*k_bar_11...
    +q_ddot_1*k_bar_11
omega_bar_dot_22 = R12*...
    omega_bar_dot_11+R12*skew(...
    omega_bar_dot_11)*q_dot_2*k_bar_22...
    +q_ddot_2*k_bar_22
omega_bar_dot_33 = R23*...
    omega_bar_dot_22+R23*skew(...
    omega_bar_dot_22)*q_dot_3*k_bar_33...
    +q_ddot_3*k_bar_33
omega_bar_dot_44 = R34*...
    omega_bar_dot_33+R34*skew(...
    omega_bar_dot_33)*q_dot_4*k_bar_44...
    +q_ddot_4*k_bar_44
omega_bar_dot_55 = R45*...
    omega_bar_dot_44+R45*skew(...
    omega_bar_dot_44)*q_dot_5*k_bar_55...
    +q_ddot_5*k_bar_55
omega_bar_dot_66 = R56*omega_bar_dot_55
% linear operational accelerations (6)
v_bar_dot_11 = R01*(v_bar_dot_00+...
    skew(omega_bar_dot_00)*r10+...
    skew(omega_bar_dot_00)^2*r10)
v_bar_dot_22 = R12*(v_bar_dot_11+...
    skew(omega_bar_dot_11)*r21+...
    skew(omega_bar_dot_11)^2*r21)
v_bar_dot_33 = R23*(v_bar_dot_22+...
    skew(omega_bar_dot_22)*r32+...
    skew(omega_bar_dot_22)^2*r32)
v_bar_dot_44 = R34*(v_bar_dot_33+...
    skew(omega_bar_dot_33)*r43+...
    skew(omega_bar_dot_33)^2*r43)
v_bar_dot_55 = R45*(v_bar_dot_44+...
    skew(omega_bar_dot_44)*r54+...
    skew(omega_bar_dot_44)^2*r54)
v_bar_dot_66 = R56*(v_bar_dot_55+...
    skew(omega_bar_dot_55)*r65+...
    skew(omega_bar_dot_55)^2*r65)

```

7. THE KINEMATIC PARAMETERS IN THE FRAME {0}

The most important results of this iterative algorithm are the absolute operational kinematic parameters, expressed in the frame {0}. They are computed using the equations (8). In the following code sequence, the LaTeX syntax is used to display the proper notations in the output section of the *Live Code* window.

```

% kinematic parameters in frame {0} (8)

```

```

v_bar_50 = R50*v_bar_55
omega_bar_50 = R50*omega_bar_55
v_bar_dot_50 = R50*v_bar_dot_55
omega_bar_dot_50 = R50*omega_bar_dot_55
disp(sym('v_bar_60'))
v_bar_60 = R60*v_bar_66;
disp(v_bar_60)
disp(sym('omega_bar_60'))
omega_bar_60 = R60*omega_bar_66;
disp(omega_bar_60)
disp(sym('v_bar_dot_60'))

```

```

v_bar_dot_60 = R60*v_bar_dot_66;
disp(v_bar_dot_60)
disp(sym('omega_bar_dot_60'))
omega_bar_dot_60 = R60*omega_bar_dot_66;
disp(omega_bar_dot_60)

```

Because of the complexity of the results, the intermediary ones are omitted in this paper and the kinematic parameters corresponding to the end effector, expressed in the frame {0} are the only presented results. They are the following:

The linear velocity vector is:

$$\bar{v}_{60} = \begin{pmatrix} -\left(c(dq_1 + q_1)c(dq_5 + q_5) - s(\sigma_1)s(dq_1 + q_1)s(dq_5 + q_5)\right)\sigma_3 - \\ -\left(c(dq_1 + q_1)s(dq_5 + q_5) + s(\sigma_1)c(dq_5 + q_5)s(dq_1 + q_1)\right)\sigma_2 - c(\sigma_1)s(dq_1 + q_1)\sigma_4 \\ c(\sigma_1)c(dq_1 + q_1)\sigma_4 - \left(s(dq_1 + q_1)s(dq_5 + q_5) - s(\sigma_1)c(dq_1 + q_1)c(dq_5 + q_5)\right)\sigma_2 - \\ -\left(c(dq_5 + q_5)s(dq_1 + q_1) + s(\sigma_1)c(dq_1 + q_1)s(dq_5 + q_5)\right)\sigma_3 \\ s(\sigma_1)\sigma_4 - c(\sigma_1)c(dq_5 + q_5)\sigma_2 + c(\sigma_1)s(dq_5 + q_5)\sigma_3 \end{pmatrix}, \quad (9)$$

where:

$$\sigma_1 = dq_2 + dq_3 + dq_4 + q_2 + q_3 + q_4 \quad (10)$$

$$\sigma_2 = (dl_4 + l_4)(s(dq_5 + q_5)\sigma_5 - c(dq_5 + q_5)(\dot{q}_2 + \dot{q}_3 + \dot{q}_4)) + s(dq_5 + q_5)\sigma_6 - c(dq_5 + q_5)\sigma_7 \quad (11)$$

$$\sigma_3 = (dl_4 + l_4)(c(dq_5 + q_5)\sigma_5 + s(dq_5 + q_5)(\dot{q}_2 + \dot{q}_3 + \dot{q}_4)) + c(dq_5 + q_5)\sigma_6 + s(dq_5 + q_5)\sigma_7 \quad (12)$$

$$\sigma_4 = s(dq_4 + q_4)\sigma_9 - \dot{q}_2 c(dq_3 + q_3)c(dq_4 + q_4)(dl_2 + l_2) \quad (13)$$

$$\sigma_5 = c(dq_4 + q_4)\sigma_8 - s(dq_4 + q_4)(\dot{q}_1 c(dq_2 + q_2)s(dq_3 + q_3) + \dot{q}_1 c(dq_3 + q_3)s(dq_2 + q_2)) \quad (14)$$

$$\sigma_6 = \dot{q}_1(dl_1 + l_1) + (dl_3 + l_3)\sigma_8 - \dot{q}_1 s(dq_2 + q_2)(dl_2 + l_2) \quad (15)$$

$$\sigma_7 = c(dq_4 + q_4)\sigma_9 + \dot{q}_2 c(dq_3 + q_3)s(dq_4 + q_4)(dl_2 + l_2) \quad (16)$$

$$\sigma_8 = \dot{q}_1 c(dq_2 + q_2)c(dq_3 + q_3) - \dot{q}_1 s(dq_2 + q_2)s(dq_3 + q_3) \quad (17)$$

$$\sigma_9 = (dl_3 + l_3)(\dot{q}_2 + \dot{q}_3) + \dot{q}_2 s(dq_3 + q_3)(dl_2 + l_2). \quad (18)$$

The angular velocity vector has the form:

$$\bar{\omega}_{60} = \left(\begin{array}{c} \sigma_3(c(dq_1 + q_1)s(dq_5 + q_5) + s(\sigma_1)c(dq_5 + q_5)s(dq_1 + q_1)) - \\ -\sigma_2(c(dq_1 + q_1)c(dq_5 + q_5) - s(\sigma_1)s(dq_1 + q_1)s(dq_5 + q_5)) - c(\sigma_1)s(dq_1 + q_1)\sigma_4 \\ \hline \sigma_3(s(dq_1 + q_1)s(dq_5 + q_5) - s(\sigma_1)c(dq_1 + q_1)c(dq_5 + q_5)) - \\ -\sigma_2(c(dq_5 + q_5)s(dq_1 + q_1) + s(\sigma_1)c(dq_1 + q_1)s(dq_5 + q_5)) + c(\sigma_1)c(dq_1 + q_1)\sigma_4 \\ \hline s(\sigma_1)\sigma_4 + c(\sigma_1)c(dq_5 + q_5)\sigma_3 + c(\sigma_1)s(dq_5 + q_5)\sigma_2 \end{array} \right), \quad (19)$$

where:

σ_1 - has the expression (10)

$$\sigma_2 = s(dq_5 + q_5)\sigma_5 - c(dq_5 + q_5)(\dot{q}_2 + \dot{q}_3 + \dot{q}_4) \quad (20)$$

$$\sigma_3 = c(dq_5 + q_5)\sigma_5 + s(dq_5 + q_5)(\dot{q}_2 + \dot{q}_3 + \dot{q}_4) \quad (21)$$

$$\sigma_4 = \dot{q}_5 + c(dq_4 + q_4)\sigma_7 + s(dq_4 + q_4)\sigma_6 \quad (22)$$

$$\sigma_5 = c(dq_4 + q_4)\sigma_6 - s(dq_4 + q_4)\sigma_7 \quad (23)$$

$$\sigma_6 = \dot{q}_1 c(dq_2 + q_2) c(dq_3 + q_3) - \dot{q}_1 s(dq_2 + q_2) s(dq_3 + q_3) \quad (24)$$

$$\sigma_7 = \dot{q}_1 c(dq_2 + q_2) s(dq_3 + q_3) + \dot{q}_1 c(dq_3 + q_3) s(dq_2 + q_2). \quad (25)$$

The linear acceleration vector is the following:

$$\dot{v}_{60} = \left(\begin{array}{c} (c(dq_1 + q_1)s(dq_5 + q_5) + s(\sigma_1)c(dq_5 + q_5)s(dq_1 + q_1))\sigma_3 + \\ + (c(dq_1 + q_1)c(dq_5 + q_5) - s(\sigma_1)s(dq_1 + q_1)s(dq_5 + q_5))\sigma_2 + c(\sigma_1)s(dq_1 + q_1)\sigma_4 \\ \hline (s(dq_1 + q_1)s(dq_5 + q_5) - s(\sigma_1)c(dq_1 + q_1)c(dq_5 + q_5))\sigma_3 + \\ + (c(dq_5 + q_5)s(dq_1 + q_1) + s(\sigma_1)c(dq_1 + q_1)s(dq_5 + q_5))\sigma_2 - c(\sigma_1)c(dq_1 + q_1)\sigma_4 \\ \hline c(\sigma_1)c(dq_5 + q_5)\sigma_3 - s(\sigma_1)\sigma_4 - c(\sigma_1)s(dq_5 + q_5)\sigma_2 \end{array} \right), \quad (26)$$

where:

σ_1 - has the expression (10)

$$\sigma_2 = c(dq_5 + q_5)\sigma_7 + (dl_4 + l_4)(\dot{q}_5\sigma_8 + c(dq_5 + q_5)\sigma_6 - s(dq_5 + q_5)(\ddot{q}_2 + \ddot{q}_3 + \ddot{q}_4)) \\ - s(dq_5 + q_5)\sigma_5 - (dl_4 + l_4)\sigma_8(\dot{q}_5 + \sigma_{13} + \sigma_{12}) \quad (27)$$

$$\sigma_3 = c(dq_5 + q_5)\sigma_5 + (dl_4 + l_4)(c(dq_5 + q_5)(\ddot{q}_2 + \ddot{q}_3 + \ddot{q}_4) - \dot{q}_5\sigma_9 + s(dq_5 + q_5)\sigma_6) + \\ + s(dq_5 + q_5)\sigma_7 + (dl_4 + l_4)\sigma_9(\dot{q}_5 + \sigma_{13} + \sigma_{12}) \quad (28)$$

$$\sigma_4 = c(dq_4 + q_4)\sigma_{11} - s(dq_4 + q_4)\sigma_{10} + (\sigma_9^2 + \sigma_8^2)(dl_4 + l_4) \quad (29)$$

$$\sigma_5 = c(dq_4 + q_4)\sigma_{10} + s(dq_4 + q_4)\sigma_{11} \quad (30)$$

$$\sigma_6 = c(dq_4 + q_4)\sigma_{14} + s(dq_4 + q_4)(\dot{q}_3\sigma_{20} + c(dq_3 + q_3)\sigma_{18} + s(dq_3 + q_3)\sigma_{19}) + \dot{q}_4(\sigma_{13} + \sigma_{12}) \quad (31)$$

$$\sigma_7 = (dl_2 + l_2)\sigma_{18} - \ddot{q}_1(dl_1 + l_1) + (dl_3 + l_3)\sigma_{14} + (dl_3 + l_3)(\dot{q}_2 + \dot{q}_3)\sigma_{21} + \dot{q}_1\dot{q}_2c(dq_2 + q_2)(dl_2 + l_2) \quad (32)$$

$$\sigma_8 = s(dq_5 + q_5)\sigma_{15} - c(dq_5 + q_5)(\dot{q}_2 + \dot{q}_3 + \dot{q}_4) \quad (33)$$

$$\sigma_9 = c(dq_5 + q_5)\sigma_{15} + s(dq_5 + q_5)(\dot{q}_2 + \dot{q}_3 + \dot{q}_4) \quad (34)$$

$$\sigma_{10} = s(dq_3 + q_3)\sigma_{16} - c(dq_3 + q_3)\sigma_{17} + (dl_3 + l_3)(\ddot{q}_2 + \ddot{q}_3) + (dl_3 + l_3)\sigma_{21}\sigma_{20} \quad (35)$$

$$\sigma_{11} = s(dq_3 + q_3)\sigma_{17} + (dl_3 + l_3)((\dot{q}_2 + \dot{q}_3)^2 + \sigma_{20}^2) + c(dq_3 + q_3)\sigma_{16} \quad (36)$$

$$\sigma_{12} = s(dq_4 + q_4)\sigma_{20} \quad (37)$$

$$\sigma_{13} = c(dq_4 + q_4)\sigma_{21} \quad (38)$$

$$\sigma_{14} = \dot{q}_3\sigma_{21} - c(dq_3 + q_3)\sigma_{19} + s(dq_3 + q_3)\sigma_{18} \quad (39)$$

$$\sigma_{15} = c(dq_4 + q_4)\sigma_{20} - s(dq_4 + q_4)\sigma_{21} \quad (40)$$

$$\sigma_{16} = \ddot{q}_2(dl_2 + l_2) + \dot{q}_1^2c(dq_2 + q_2)(dl_1 + l_1) - \dot{q}_1^2c(dq_2 + q_2)s(dq_2 + q_2)(dl_2 + l_2) \quad (41)$$

$$\sigma_{17} = (dl_2 + l_2)(\dot{q}_1^2s(dq_2 + q_2)^2 + \dot{q}_2^2) - \dot{q}_1^2s(dq_2 + q_2)(dl_1 + l_1) \quad (42)$$

$$\sigma_{18} = \ddot{q}_1s(dq_2 + q_2) + \dot{q}_1\dot{q}_2c(dq_2 + q_2) \quad (43)$$

$$\sigma_{19} = \ddot{q}_1c(dq_2 + q_2) - \dot{q}_1\dot{q}_2s(dq_2 + q_2) \quad (44)$$

$$\sigma_{20} = \dot{q}_1c(dq_2 + q_2)c(dq_3 + q_3) - \dot{q}_1s(dq_2 + q_2)s(dq_3 + q_3) \quad (45)$$

$$\sigma_{21} = \dot{q}_1c(dq_2 + q_2)s(dq_3 + q_3) + \dot{q}_1c(dq_3 + q_3)s(dq_2 + q_2). \quad (46)$$

The angular acceleration vector is:

$$\dot{\vec{\omega}}_{60} = \left(\begin{array}{c} \frac{(c(dq_1 + q_1)c(dq_5 + q_5) - s(\sigma_1)s(dq_1 + q_1)s(dq_5 + q_5))\sigma_2 - (c(dq_1 + q_1)s(dq_5 + q_5) + s(\sigma_1)c(dq_5 + q_5)s(dq_1 + q_1))\sigma_3 - c(\sigma_1)s(dq_1 + q_1)\sigma_4}{(c(dq_5 + q_5)s(dq_1 + q_1) + s(\sigma_1)c(dq_1 + q_1)s(dq_5 + q_5))\sigma_2 - (s(dq_1 + q_1)s(dq_5 + q_5) - s(\sigma_1)c(dq_1 + q_1)c(dq_5 + q_5))\sigma_3 + c(\sigma_1)c(dq_1 + q_1)\sigma_4} \\ s(\sigma_1)\sigma_4 - c(\sigma_1)c(dq_5 + q_5)\sigma_3 - c(\sigma_1)s(dq_5 + q_5)\sigma_2 \end{array} \right), \quad (47)$$

where:

σ_1 - has the expression (10)

$$\sigma_2 = c(dq_5 + q_5)(\ddot{q}_2 + \ddot{q}_3 + \ddot{q}_4) - \dot{q}_5(c(dq_5 + q_5)\sigma_6 + s(dq_5 + q_5)(\dot{q}_2 + \dot{q}_3 + \dot{q}_4)) + s(dq_5 + q_5)\sigma_5 \quad (48)$$

$$\sigma_3 = \dot{q}_5(s(dq_5 + q_5)\sigma_6 - c(dq_5 + q_5)(\dot{q}_2 + \dot{q}_3 + \dot{q}_4)) + c(dq_5 + q_5)\sigma_5 - s(dq_5 + q_5)(\ddot{q}_2 + \ddot{q}_3 + \ddot{q}_4) \quad (49)$$

$$\sigma_4 = \ddot{q}_5 + c(dq_4 + q_4)\sigma_8 - s(dq_4 + q_4)\sigma_7 + \dot{q}_4\sigma_6 \quad (50)$$

$$\sigma_5 = c(dq_4 + q_4)\sigma_7 + s(dq_4 + q_4)\sigma_8 + \dot{q}_4(c(dq_4 + q_4)\sigma_9 + s(dq_4 + q_4)\sigma_{10}) \quad (51)$$

$$\sigma_6 = c(dq_4 + q_4)\sigma_{10} - s(dq_4 + q_4)\sigma_9 \quad (52)$$

$$\sigma_7 = \dot{q}_3\sigma_9 - c(dq_3 + q_3)\sigma_{11} + s(dq_3 + q_3)\sigma_{12} \quad (53)$$

$$\sigma_8 = \dot{q}_3\sigma_{10} + c(dq_3 + q_3)\sigma_{12} + s(dq_3 + q_3)\sigma_{11} \quad (54)$$

$$\sigma_9 = \dot{q}_1 c(dq_2 + q_2) s(dq_3 + q_3) + \dot{q}_1 c(dq_3 + q_3) s(dq_2 + q_2) \quad (55)$$

$$\sigma_{10} = \dot{q}_1 c(dq_2 + q_2) c(dq_3 + q_3) - \dot{q}_1 s(dq_2 + q_2) s(dq_3 + q_3) \quad (56)$$

$$\sigma_{11} = \ddot{q}_1 c(dq_2 + q_2) - \dot{q}_1 \dot{q}_2 s(dq_2 + q_2) \quad (57)$$

$$\sigma_{12} = \ddot{q}_1 s(dq_2 + q_2) + \dot{q}_1 \dot{q}_2 c(dq_2 + q_2). \quad (58)$$

The trigonometric functions abbreviations from the literature were used here, in order to get more compact results. The form $s(x)$ was used for $\sin(x)$, and the form $c(x)$ was used for $\cos(x)$.

The kinematic parameters presented in this paragraph are the equations of the direct kinematic model of FANUC LR Mate 100iB, considering the geometric errors, and they express the linear and angular operational velocities and accelerations, corresponding to the frame {6}, attached to the end-effector, with respect to the fixed frame {0}.

8. NUMERIC DATA

In order to make numeric substitutions into the equations of the DKM, the parameters $l0n...l4n$ are initialized with the nominal values of the constructive dimensions of FANUC LR Mate 100iB, taken from [3], [5], expressed in millimeters.

Some values of generalized coordinate from a robot program [6] are then stored into the variables $q1n...q5n$, expressed in degrees and converted to radians, using the function $deg2rad()$, as presented in [5].

The numerical values of the errors of constructive dimensions $dl0n...dl4n$, as well as the errors of the generalized coordinates,

$dq1n...dq5n$ were used, identical to those in [5]. In order to get a magnitude order of the kinematic errors to be obtained, the maximum values of the generalized velocities were used, as presented in [3].

```
% Joint velocities [rad/s]
% the max velocities were used
q_dot_1n = deg2rad(240);
q_dot_2n = deg2rad(270);
q_dot_3n = deg2rad(270);
q_dot_4n = deg2rad(330);
q_dot_5n = deg2rad(480);
% Joint accelerations [rad/s^2]
q_ddot_1n = deg2rad(5.25);
q_ddot_2n = deg2rad(6.3);
q_ddot_3n = deg2rad(8.44);
q_ddot_4n = deg2rad(10.53);
q_ddot_5n = deg2rad(12.35);
```

Because there is no data about the generalized accelerations in the robot documentation, some random data was considered, as presented in the above code section.

All numeric data will be substituted in the symbolic form of the operational kinematic parameters, yielding two forms of the corresponding vectors: one for the nominal mechanical structure (having the suffix nn) and

one for the real mechanical structure, including the errors (having the suffix *nr*).

The difference of the two pair of vectors (corresponding to the nominal and to the real geometry, respectively) will mark the errors of the operational kinematic parameters.

- Nominal linear velocity vector [mm/s]:

$$\bar{v}_{60nr} = 1.0e+03 \begin{pmatrix} 0.9961 \\ -3.9073 \\ 4.6844 \end{pmatrix} \quad (59)$$

- Real linear velocity vector [mm/s]:

$$\bar{v}_{60r} = 1.0e+03 \begin{pmatrix} 1.0503 \\ -3.9196 \\ 4.6643 \end{pmatrix} \quad (60)$$

- Delta linear velocity vector [mm/s]:

$$\Delta \bar{v}_{60n} = \begin{pmatrix} -54.1372 \\ 12.3631 \\ 20.0580 \end{pmatrix} \quad (61)$$

- Nominal angular velocity vector [deg/s]:

$$\bar{\omega}_{60nr} = \begin{pmatrix} 295.2398 \\ 935.1260 \\ 400.2273 \end{pmatrix} \quad (62)$$

- Real angular velocity vector [deg/s]:

$$\bar{\omega}_{60r} = \begin{pmatrix} 290.3560 \\ 936.0882 \\ 403.5003 \end{pmatrix} \quad (63)$$

- Delta angular velocity vector [deg/s]:

$$\Delta \bar{\omega}_{60n} = \begin{pmatrix} 4.8838 \\ -0.9621 \\ -3.2730 \end{pmatrix} \quad (64)$$

- Nominal linear acceleration vector [mm/s²]:

$$\dot{\bar{v}}_{60nr} = 1.0e+04 \begin{pmatrix} 7.1820 \\ -3.0805 \\ -3.3681 \end{pmatrix} \quad (65)$$

- Real linear acceleration vector [mm/s²]:

$$\dot{\bar{v}}_{60nr} = 1.0e+04 \begin{pmatrix} 7.2023 \\ -2.9975 \\ -3.4144 \end{pmatrix} \quad (66)$$

- Delta linear acceleration vector [mm/s²]:

$$\Delta \dot{\bar{v}}_{60n} = \begin{pmatrix} -203.0503 \\ -830.5429 \\ 462.4110 \end{pmatrix} \quad (67)$$

- Nominal angular acceleration vector [deg/s²]:

$$\dot{\bar{\omega}}_{60nr} = 1.0e+03 \begin{pmatrix} -2.1871 \\ -0.4576 \\ 6.8798 \end{pmatrix} \quad (68)$$

- Real angular acceleration vector [deg/s²]:

$$\dot{\bar{\omega}}_{60r} = 1.0e+03 \begin{pmatrix} -2.1454 \\ -0.5024 \\ 6.8621 \end{pmatrix} \quad (69)$$

- Delta angular acceleration vector [deg/s²]:

$$\Delta \dot{\bar{\omega}}_{60n} = \begin{pmatrix} -41.6652 \\ 44.8061 \\ 17.7180 \end{pmatrix}. \quad (70)$$

The last line of Live script file saves the workspace into the data file *FANUC_kin_err.mat*, making possible its usage for its further analysis.

9. CONCLUSION

The open chain architecture of serial robots leads to geometric and kinematic error propagation from link to link and from joint to joint. To keep the robot in its prescribed operating parameters, some important error compensating measures have to be taken. As a further development of this work, the analysis of dynamic [9], [10] errors must be performed, with the goal of detecting and minimizing the source of operational errors [11], [12]. Problems like path planning [6], [15] and vibration analysis [13, 14] can also be considered.

10. REFERENCES

- [1] Deteșan, O.A., *Cercetări privind modelarea, simularea și construcția miniroboților*, Ph.D. Thesis, Technical University of Cluj-Napoca, 2007.
- [2] Negrean, I., Duca, A., Negrean, C., Kacso, K., *Mecanică avansată în robotică*, U.T. Press, ISBN 978-973-662-420-9, Cluj-Napoca, 2008.
- [3] FANUC Robot Series, LR Mate 100iB, Maintenance Manual, B-81595EN/01, 2001.
- [4] Simion, M., Botean, A.I., Bejan, M., *Analytic Study of the Efforts from the Structure of a Industrial Serial Robot 5R*, Acta Technica Napocensis, Series: Applied Mathematics and Mechanics, Vol 54, No 1, ISSN 1221-5872, Cluj-Napoca, 2011.
- [5] Deteșan, O.A., Crișan, A.V., *Geometric Model of Fanuc LR Mate 100iB Considering the Dimensional and Generalized Coordinates Errors*, Acta Technica Napocensis, Series: Applied Mathematics, Mechanics and Engineering, Vol 62, Issue 1, pp. 57-62, ISSN 1221-5872, Cluj-Napoca, 2019.
- [6] Deteșan, O.A., *The Path Planning of Industrial Robots Using Polynomial Interpolation, with Applications to Fanuc LR-Mate 100iB*, Acta Technica Napocensis, Series: Applied Mathematics, Mechanics and Engineering, Vol 56, No 4, ISSN 1221-5872, Cluj-Napoca, 2013.
- [7] Deteșan, O.A., *The Definition of the Robot Mechanical Structure Using the Symbolic Computation in MATLAB*, Acta Technica Napocensis, Series: Applied Mathematics and Mechanics, Vol 51, No 4, ISSN 1221-5872, Cluj-Napoca, 2008.
- [8] Deteșan, O.A., Bugnar, F., *Using the Symbolic Computation in MATLAB for Determining the Geometric Model of Serial Robots*, Acta Technica Napocensis, Series: Applied Mathematics, Mechanics and Engineering, Vol 55, No 3, ISSN 1221-5872, Cluj-Napoca, 2012.
- [9] Negrean, I., *New Approaches on Notions from Advanced Mechanics*, Acta Technica Napocensis, Series: Applied Mathematics, Mechanics and Engineering, Vol 61, No 2, ISSN 1221-5872, Cluj-Napoca, 2018.
- [10] Negrean, I., *Advanced Notions and Differential Principles of Motion in Analytical Dynamics*, Journal of Engineering Sciences and Innovation, Technical Sciences Academy of Romania, Vol 1, No 1, 49-72, ISSN 2537-320X, Bucharest, 2016.
- [11] Antal, T.A., *Considerations on the Serial PC - Arduino Uno R3 Interaction, in Java, Using Jdeveloper, for a 3R Serial Robot, Based on the Ardulink Library*, Acta Technica Napocensis, Series: Applied Mathematics, Mechanics and Engineering, Vol 61, No 1, pp. 7-10, ISSN 1221-5872, Cluj-Napoca, 2018.
- [12] Antal, T.A., *3R Serial Robot Control Based on Arduino/Genuino Uno, in Java, Using Jdeveloper and Ardulink*, Acta Technica Napocensis, Series: Applied Mathematics, Mechanics and Engineering, Vol 61, No 1, pp. 11-16, ISSN 1221-5872, Cluj-Napoca, 2018.
- [13] Cristea, A.F., *Mechanical stress and strain properties, regarding the elbow joint*, Acta Technica Napocensis, series Applied Mathematics, Mechanics and Engineering, Vol. 57, No 2, June 2014, ISSN 1221-5872.
- [14] Cristea A.F., Bălcau, M., *Linear hand-arm system stability*, Acta Technica Napocensis Cluj-Napoca, series: Applied Mathematics, Mechanics and Engineering, Vol. 55, No 4, 2012, pp. 745-748, ISSN 1221-5872.
- [15] Crisan, A.V., Serdean, F.M., Morariu-Gligor, R., *The Analysis of Geometrical Errors Based on Polynomial Interpolation Functions for a 5 D.O.F. Serial Robot*, Acta Technica Napocensis, series: Applied Mathematics, Mechanics and Engineering, Vol. 60, No 4, 2017, pp. 573-578, Cluj-Napoca, ISSN 1221-5872.

Modelul cinematic al robotului FANUC LR Mate 100iB, ținând seama de erorile geometrice

Rezumat: Metodele de modelare cinematică a roboților industriali, utilizate în mod regulat în literatură, sunt strict legate de geometria și cinematica nominală a robotului, robotul fiind considerat un sistem de corpuri rigide, perfect din punct de vedere dimensional și pozițional. Această lucrare prezintă influența erorilor dimensionale și de coordonate generalizate asupra vitezelor și accelerațiilor operaționale, unghiulare și liniare, corespunzătoare robotului industrial FANUC LR Mate 100iB. Erorile dimensiunilor constructive și erorile coordonatelor generalizate sunt incluse în ecuațiile modelului geometric direct, determinate prin metoda matricelor de situare. Datele obținute vor fi utilizate ca date de intrare în algoritmul iterativ de modelare cinematică. Prin înlocuirea valorilor numerice în ecuațiile cinematice, se evidențiază influența acestor erori asupra parametrilor cinematici operaționali.

Ovidiu-Aurelian DETEȘAN, Ph.D., Assoc. Prof., Eng., Technical University of Cluj-Napoca, Faculty of Industrial Engineering, Robotics and Production Management, Dept. of Mechanical System Engineering, Ovidiu.Detesan@mep.utcluj.ro, phone: +40-64-401667, B-dul Muncii no.103-105, Cluj-Napoca, Romania.