

DYNAMIC MODELING OF MANIPULATORS WITH SYMBOLIC COMPUTATIONAL METHOD

Viviana FILIP

VALAHIA University of Targoviste, e-mail: v_filip@yahoo.com

This paper presents a symbolic computational method for manipulators dynamic equations. For a certain trajectory imposed to characteristic point, the necessary motor torque is calculated in the driving joint. The dynamic model Lagrange - Euler is considered for a manipulator with n degree of freedom, for which a symbolic computational algorithm is elaborated depending on input and output data. For example, this symbolic calculation is applied to a trimobil planar manipulator, as an open kinematical chain.

Key words: manipulator; dynamic model; symbolic calculation; generalized force; Coriolis force.

1. INTRODUCTION

The dynamic modeling of robots [5,6] represents the determination of the dynamic equations, which is the first information necessary for robots control [1,4]. These equations are useful for computational simulation of the robots motion and for the evaluation of kinematical structure of robots [1]. In the dynamic formulation of manipulators the following methods are used: Lagrange-Euler, Newton-Euler, D'Alembert. In [5,6] reference books there are discussed only plan manipulators with 2 degree of freedom. For the manipulators with more than 2 degrees of freedom (DOF), a very laborious calculation is necessary. The Lagrange-Euler method is relatively simple and systematical. As a rule, the dynamic for a device of electronic control and the frictions of gearing are not considering. Thus there is obtained a 2nd degree equation system. For the robots with 6 DOF, the dynamic equations are nonlinear and very laborious. Generally, each term of the inertial force and gravitational force depend of the instantaneous position of the kinematical links; the terms moment and force depend on the velocity and the position of kinematical links. The dynamic equations are obtained by the Lagrange-Euler method for the non-conservatives systems. If the non-dimensional method is used, the dynamic formulation is more efficient. The new method of kinematics and dynamics modeling use the homogenous matrix and the lagrangean formulation.

The objective of the paper is the determination of the robots dynamic equations, when the mechanic characteristics are known. In this paper there is proposed a program created in Mathematica programming language. This program is developed for the manipulator-robots with open kinematical chains, using only revolute and prismatic pairs. Through this program, we obtained: the Denavit-Hartenberg (D-H) parameters for robot, the geometric and kinematical model, the effective inertia of each joint, the coupling inertia, the Coriolis forces, the gravitational loads and the dynamic equations.

2. DYNAMIC MODEL LAGRANGE-EULER

Is considered the differential equation of Lagrange:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \quad (i = 1, 2, \dots, n) \quad (1)$$

where: $L = E_c - E_p$; n – number of DOF; q_j – coordinate of joint j ; \dot{q}_i – velocity of joint i ; Q_i – generalized force of joint i .

The kinematics energy of kinematical link i is:

$$E_c = \frac{1}{2} \sum_{j=1}^i \sum_{k=1}^i D \left[\frac{\partial \left({}^0_i [T] \right)}{\partial q_j} [J_i] \frac{\partial \left({}^0_i [T] \right)^T}{\partial q_k} \right] \dot{q}_j \dot{q}_k \quad (2)$$

where:

r_i is the position vector of mass

${}^0_i [T]$ is the transfer matrix from the i link attached system to the fixed reference system;

$J_i = \int_i \vec{r}_i \vec{r}_i^T dm$ is the 4x4 inertial matrix of kinematical link i , with the planar, centrifugal and static

moments.

The potential energy of kinematical link i in gravitational field is:

$$E_p = - \sum_{i=1}^n m_i [g]^T {}^0_i [T] \cdot {}^i [r_i] \quad (3)$$

The Lagrange function of one manipulator-robot with n kinematical links is:

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i D \left[\frac{\partial \left({}^0_i [T] \right)}{\partial q_j} [J_i] \frac{\partial \left({}^0_i [T] \right)^T}{\partial q_k} \right] \dot{q}_j \dot{q}_k + \frac{1}{2} \sum_{i=1}^n J_{mi} \dot{q}_i^2 + \sum_{i=1}^n m_i [g]^T {}^0_i [T] {}^i [r_i] \quad (4)$$

Replacing (2) in (1), we obtain:

$$Q_i = \sum_{j=1}^n D_{ij} \ddot{q}_j + J_{mi} \ddot{q}_i + \sum_{j=1}^n \sum_{k=1}^n D_{ijk} \dot{q}_j \dot{q}_k + D_i \quad (5)$$

where:

$$D_{ij} = \sum_{p=\max(i,j)}^n D \left[\frac{\partial \left({}^0_p [T] \right)}{\partial q_j} [J_p] \frac{\partial \left({}^0_p [T] \right)^T}{\partial q_i} \right] \quad (6)$$

$$D_{ijk} = \sum_{p=\max(i,j,k)}^n D \left[\frac{\partial \left({}^0_p [T] \right)}{\partial q_j \partial q_k} [J_p] \frac{\partial \left({}^0_p [T] \right)^T}{\partial q_i} \right] \quad (7)$$

$$D_i = - \sum_{p=i}^n m_p [g]^T \frac{\partial \left({}^0_p [T] \right)}{\partial q_i} {}^i [r_i] \quad (8)$$

The coefficients (6), (7) and (8) are: D_{ii} – effective inertia of joint i ; $D_{ij} = D_{ji}$ – coupling inertia between joints i and j ; $D_{ijk} = D_{ikj}$ – Coriolis inertia of joint i due to speeds of joint j and k ; D_i – gravitational loads of joint i ; J_{mi} – motor inertia of joint i ; J_i – pseudo inertia matrix.

3. GEOMETRICAL TRANSFORMATION D-H

The operator which describes the position and orientation of reference system i relative to system $i-1$ is defined through D-H matrix:

$${}^{i-1}_i[T] = \begin{bmatrix} c_i & -s_i c \alpha_i & s_i s \alpha_i & d_i c_i \\ s_i & c_i c \alpha_i & -c_i s \alpha_i & d_i s_i \\ 0 & s \alpha_i & c \alpha_i & a_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

where: $c = \cos$ and $s = \sin$.

In an open kinematical chain with revolute joints, the angular parameters φ_i, α_i and the linear parameters a_i, d_i have the following signification (Fig. 1): $\alpha_i = \angle(z_{i-1}, z_i)$, $a_i = \overline{O_{i-1}A}$, $d_i = \overline{AO_i}$.

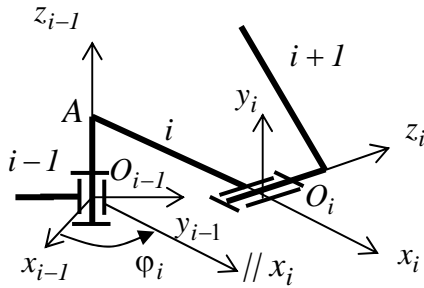


Fig.1 Kinematical chain $(i-1, i, (i+1))$

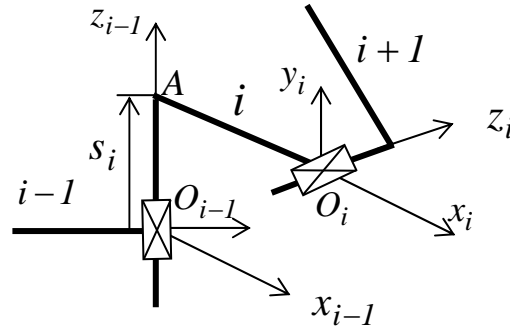


Fig.2. Kinematical chain with prismatic joints

In the case of the revolute joint $(i-1, i)$ in the point O_{i-1} , the angular displacement φ_i is variable (positive or negative). The others three parameters a_i, d_i, α_i have only constant values. The distance d_i is the length of the common normal of the axes z_{i-1} and z_i [2]. In a particular case of the kinematical chain, the axes z_{i-1} and z_i are parallels (planar kinematical chain) or perpendiculars (spatial kinematical chain): $\alpha_i = 0^\circ, 90^\circ$.

In the case of the prismatic joint between the kinematical link $(i-1)$ and i , $x_{i-1} // x_i$, that means the angular displacement is null ($\varphi_i = 0$), while the linear displacement $s_i = a_i$ is variable (Fig.2).

For $\varphi_i = 0$, the general matrix D-H (9) has this form:

$${}^{i-1}_i[T] = \begin{bmatrix} 1 & 0 & 0 & d_i \\ 0 & 1 & 0 & 0 \\ 0 & s \alpha_i & c \alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

In the particular case $\alpha_i = 0$, the matrix (10) is reduced at a translation matrix in the plan $[x_{i-1}, z_{i-1}]$:

$${}^{i-1}_i[T] = \begin{bmatrix} 1 & 0 & 0 & d_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

For a planar kinematical chain, the general form of the homogeneous matrix results on the basis of the matrix (9), where $\alpha_i = 0$ ($z_{i-1} // z_i$) and $a_i = 0$ (Fig.3).

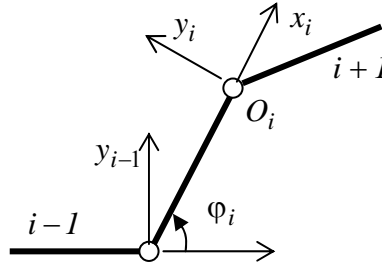


Fig.3. Planar kinematical chain with revolute joints

The matrix D-H of the planar kinematical chain (Fig.3) is:

$${}^{i-1}_i[T] = \begin{bmatrix} c\varphi_i & -s\varphi_i & 0 & d_i c\varphi_i \\ s\varphi_i & c\varphi_i & 0 & d_i s\varphi_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (12)$$

where $d_i = \overline{O_{i-1}O_i}$ and $\varphi_i = \angle(x_{i-1}, x_i)$.

4. ALGORITHM AND PROGRAM OF DYNAMIC CALCULATION. APPLICATION

The D-H method was used by many researchers for the study of kinematics and dynamics of manipulator robots. It forms the basis for the following programs GESIMA (GEometric SIMulation of the MANipulator), SPHEMA (SPHEREs Manipulator), IKIREM (Inverse KInematics REDundant Manipulator) which solves the inverse kinematics for a redundant manipulator by utilizing the optimum simulation of the system manipulator – obstacle [3].

Also based on the D-H method the SUMT program was developed [4] which computes the penalty function method for inverse kinematics. The upper mentioned programs were developed using FORTRAN programming language because this language also allowed the graphical representation of the robots position based on calculated solutions and through the communication with other commercial programs which contain subroutines for graphical communication. The program proposed in this paper is similar to the upper mentioned ones, except it's development in the Mathematica programming language, thus because the paper aims to establish the dynamic equations and the Mathematica already contains specific functions for this purpose.

The proposed algorithm and program for the dynamics calculation of the manipulator-robots has input and output data.

The input data are: D-H parameters; the coordinates of mass center for each kinematical link; the inertial couples axial and centrifugal of each kinematical link; the inertial motor (actuator) couple for each joint; the vector of gravitational acceleration.

The output data are: the geometric - kinematical model; the effective inertial force or couple of each joint; the inertia of coupling between the joints i and j ; the centrifugal force of link i due to speed of joint

j ; the Coriolis force of the joint i due to the speeds of joint j and k ; the gravitational loads of each joint; the manipulator – robots dynamic equations.;

The notations used in computational program are [5]:

$m[i]$ - the mass of i kinematical link;

g - the gravitational acceleration vector, with respect to the fixed-reference system;

$q[i][t]$ - the generalized coordinate of kinematical link as function of time;

$Q[i][t]$ - the generalized force of joint i corresponding the independent parameter $q[i][t]$.

It is consider, as example, a planar manipulator 3R (Fig. 4) actuating in the vertical plan.

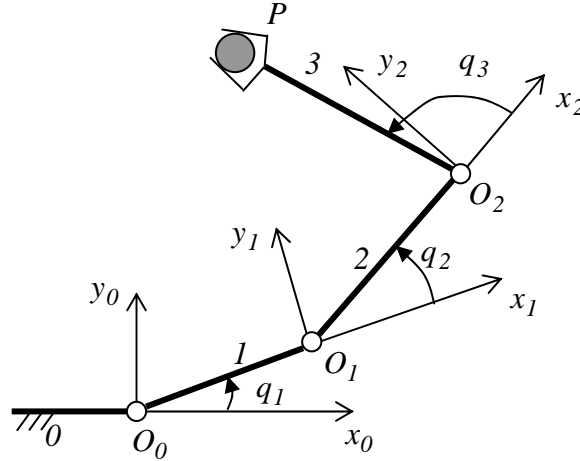


Fig.4. Trimobil planar manipulator 3R

In the particular case, the three links have equal lengths:

$$\overline{O_0O_1} = \overline{O_1O_2} = \overline{O_2P} = l.$$

The numerical value is: $l = 0,2m = 2dm$.

Also, all kinematical links are homogeneous bars, having the same density $\rho = 8000 \text{ kg} / \text{m}^3$. For each kinematical link, the transversal section is constant, having the same diameter (3cm).

The actuating systems are represented by the motoreductors placed in the 3 kinematical joint O_0, O_1 and O_2 (Fig.4). The actuators have same inertial mechanical couple: $J_{m1} = J_{m2} = J_{m3} = 0,11 \text{ kg} \cdot \text{dm}^2$

The generalized coordinates are represented through the angular displacements: $q_1 = \varphi_{10}$ (absolute position) and $q_2 = \varphi_{21}, q_3 = \varphi_{32}$ (relative position).

In the computational program, the axial inertial couples J_{xx}, J_{yy}, J_{zz} , and the centrifugal inertial couples are functions of mass $m[i]$ for each kinematical link i :

$$\begin{aligned} \int x^2 dm &= \frac{1}{2}(-J_{xx} + J_{yy} + J_{zz}); \\ \int y^2 dm &= \frac{1}{2}(J_{xx} - J_{yy} + J_{zz}); \\ \int z^2 dm &= \frac{1}{2}(J_{xx} + J_{yy} - J_{zz}). \end{aligned} \quad (13)$$

The centrifugal couples are:

$$J_{xy} = \int xy dm; J_{xz} = \int xz dm; J_{yz} = \int yz dm. \quad (14)$$

The static couples are:

$$S_x = \int x dm; S_y = \int y dm; S_z = \int z dm. \quad (15)$$

After program execution the following output data (dynamic equations) are obtained:

$$\begin{aligned} Q[1][t] \rightarrow & g \cos[q[1][t]] m[1] + (2g \cos[q[1][t]] + g \cos[q[1][t]+q[2][t]]) m[2] + (2g \cos[q[1][t]] + \\ & + q[2][t] + g \cos[q[1][t]+q[2][t]+q[3][t]]) m[3] + 2(-2 m[2] \sin[q[2][t]] + m[3](-4 \sin[q[2][t]] - 2 \sin[q[2][t]+q[3][t]])) \\ & (q[1])'[t](q[2])'[t] + (-2m[2]\sin[q[2][t]+m[3](-4\sin[q[2][t]]-2\sin[q[2][t]+q[3][t]]))(q[2])'[t]^2 + 2m[3](-2\sin[q[3][t]]-2 \\ & \sin[q[2][t]+q[3][t]]) (q[1])'[t] (q[3])'[t] + 2 m[3] (-2 \sin[q[3][t]] - 2 \sin[q[2][t] + q[3][t]]) (q[2])'[t] (q[3])'[t] + m[3] (-2 \\ & \sin[q[3][t]] - 2 \sin[q[2][t] + q[3][t]]) (q[3])'[t]^2 + J_{m1} (q[1])''[t] + (m[1] + (5+4\cos[q[2][t]])m[2] + (9+8 \\ & \cos[q[2][t]+4\cos[q[3][t]]+4\cos[q[2][t]+q[3][t]]) m[3]) (q[1])''[t] + ((1+2 \cos[q[2][t]]) m[2] + (5 + 4 \cos[q[2][t]] \\ & + 4\cos[q[3][t]]+2 \cos[q[2][t]+q[3][t]]) m[3]) (q[2])''[t] + (1+2\cos[q[3][t]]+2\cos[q[2][t]+q[3][t]]) m[3] (q[3])''[t] \end{aligned}$$

$$\begin{aligned} Q[2][t] \rightarrow & g \cos[q[1][t] + q[2][t]]m[2] + (2g \cos[q[1][t] + q[2][t]+g \cos[q[1][t]+q[2][t]+q[3][t]]) m[3] + (2m[2] \\ & \sin[q[2][t] + m[3](4\sin[q[2][t]] + 2\sin[q[2][t]+q[3][t]])) (q[1])''[t]^2 - 4m[3] \sin[q[3][t]](q[1])''[t] (q[3])'[t] - 4m[3] \\ & \sin[q[3][t]] (q[2])'[t] (q[3])'[t] - 2m[3]\sin[q[3][t]] (q[3])'[t]^2 + ((1+2\cos[q[2][t]]) m[2] + (5+4\cos[q[2][t]] + 4\cos \\ & [q[3][t]] + 2\cos[q[2][t]+q[3][t]]) + m[3]) (q[1])''[t] + J_{m2} (q[2])''[t] + (m[2] + (5+4\cos[q[3][t]]) m[3]) (q[2])''[t] + \\ & (1+2\cos[q[3][t]]) m[3] (q[3])''[t] \end{aligned}$$

$$\begin{aligned} Q[3][t] \rightarrow & g \cos[q[1][t]+q[2][t]+q[3][t]] m[3] + m[3] (2\sin[q[3][t]] + 2\sin[q[2][t]+q[3][t]]) (q[1])''[t]^2 + \\ & 4m[3]\sin[q[3][t]](q[1])'[t] (q[2])'[t] + 2m[3] \sin[q[3][t]] (q[2])''[t]^2 + (1+2\cos[q[3][t]] + 2\cos[q[2][t]+q[3][t]]) + \\ & m[3] (q[1])''[t] + (1+2\cos[q[3][t]]) m[3] (q[2])''[t] + J_{m3} (q[3])''[t] + m[3] (q[3] (q[3])''[t] \end{aligned}$$

5. CONCLUSIONS

The program was developed in Mathematica, which is an advanced mathematic computations programming language. By using Mathematica the coding of needed subroutines is not necessary for equations and systems of equations solving. These issues represent a drawback in older/classical programming languages like FORTRAN, Pascal, C+ because of prolonged computational periods needed. The efficiency of the program proposed through this paper can be observed especially in the modelling of inverse geometry of redundant structures, which needs non-linear equations systems solving through iterative methods. Another efficiency feature of this program can be observed in the simulation of manipulator robots through the integration of differential non-homogenous equations with given initial parameters.

REFERENCES

1. FILIP V., NEAGU A. - *A Symbolic Computational Method for Dynamic Simulation of Multibody Systems*, Proceeding of International Conference on Innovative Computing, Information and Control, Volume I, China, pp.130-133, 2006.
2. MANFRED L. HUSTY, MARTIN PFURNER, HANS-PETER SCHROCKER - *A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator*, Mechanism and Machine Theory 42, Elsevier, pp. 66-81, 2007
3. MITSU, S., BOUZAKIS K. D., - *Simulation of redundant manipulators for collision avoidance in manufacturing and assembly environments*, Mechanism and Machine Theory Vol. 28, No. 1, Elsevier, pp. 13 -21, 1993.
4. MITSU, S., BOUZAKIS K. D., MANSOUR G., - *Optimization of robot links motion in inverse kinematics solution considering collision avoidance and joint limits*, Mechanism and Machine Theory Vol. 30, No. 5, Elsevier, pp. 653 - 663, 1995.
5. PAUL R., - *Robot Manipulators - Mathematics, Programming and Control*, MIT Press Cambridge, 1981.
6. ZAHARIA S., FILIP V. - *A Symbolic Computational Method for a Dynamic Model of Robot Manipulators*, Proceeding of 2nd International Conference on Engineering Computational Technology Leuven, Belgium, pp. 51-56, 2000.

Received July 09, 2008