Dynamics of a General Multi-axis Robot with Analytical Optimal Torque Analysis

Atef A. Ata, Mohamed A. Ghazy, and Mohamed A. Gadou
Department of Engineering Mathematics and Physics, Faculty of Engineering, Alexandria University, Alexandria 21544, Egypt
Email: atefa@alexu.edu.eg, {mohghazy, eng.mohamed.aref}@gmail.com

***Abstract*—Robot dynamics is considered one of the most important issues in robot design and control. Many techniques were developed to find equations of motion. One of these techniques is Lagrange-Euler method which is suitable for numerical simulation. In this paper an implementation of Lagrange-Euler to find equations of motion for any general multi-axis robot giving only robot configurations is introduced. The program is verified for a 3 Degree-of-Freedom robot. The robot equations of motion are obtained from the implemented program and verified against those obtained using only Lagrange equation. The output of program for the 3 DOF robot was used to find the optimal torque using analytical optimization analysis for a given set of parameters. This procedure analysis can be used as a benchmark analysis for any optimization technique.**

***Index Terms*—dynamics, lagrange-euler, genetic algorithm, trajectory planning, optimal.**

I. Introduction

Optimal trajectory planning is regarded as a very important area for research where some constraints and objectives are required to be optimized. Some examples of objective functions are minimum path for a manipulator travel to achieve its target, minimum time in travel, and minimum applied torques on manipulator joints. Also there may be constraints on the maximum torque that can be applied on any joint. Robot dynamic model is important as it provides relationship between the applied forces/torques and the motion of robot manipulator.

Many techniques have been developed to find the equations of motion of a multi-degree-of-freedom robot such as Newton-Euler and Lagrange-Euler. A short review in the field of this research including fundamental work and present techniques can be found in Featherstone and Orin [1]. Hollerbach [2] proved that Newton-Euler approach can be formulated as a recursive structure which can be faster than treating the manipulator as a whole; he also showed that Lagrange-Euler can be used in a recursive manner. Lagrange-Euler technique is so suitable for numerical solution. Khalil [3] provided more details about these techniques and presented some techniques on conversion between Cartesian and generalized coordinates. In the comparison between Newton-Euler and Lagrange-Euler Silver [4] showed that the computational complexities of the two techniques are the same.

In this paper, an implementation based of Lagrange-Euler technique to determine the equations of motion for an n-axis robot is presented. An example for a 3 DOF robot is illustrated to verify the proposed algorithm. An analytical optimization approach is investigated as a benchmark for minimum energy using any optimization technique. The paper is organized as follows: section II contains the equations of motion in compact form and details of the algorithm to get each term. Section III presents the case study for a 3 DOF robot while section IV is devoted to analytical optimization analysis followed by discussion and conclusions in section V and references.

II. Dynamical Analysis

The objective of Lagrange-Euler method is to get equations of motion for a robot provided that robot configurations and the kinematic equation in terms of Denavit-Hartenberg are given. Lagrange-Euler technique depends on finding kinetic energy of a body which is changed with its spatial and angular velocity in general motion and finding its potential energy. In the case of rigid body dynamics, the only source for potential energy is gravity. It is suitable for rigid robot but there were some research activities on using this technique for flexible robotic manipulators and this is mentioned by Lin and Yuan [5].

The general form of robot equation of motion can be given using Lagrange-Euler technique in the following form:

$$T_i = \sum_{j=1}^{n} D_{ij}\ddot{q}_j + I_{i(act)}\ddot{q}_i + \sum_{j=1}^{n}\sum_{k=1}^{n} D_{ijk}\dot{q}_j\dot{q}_k + D_i \qquad (1)$$

where T_i is the actuator torque of joint i, $I_{i(act)}$ is the actuator inertia of joint i, q_j is the generalized coordinate represented by p if the joint is prismatic or represented by θ if the joint is revolute, n is the total number of links and D terms can begotten as follows:

$$D_{ij} = \sum_{p=\max(i,j)}^{n} Trace\left(U_{pj} J_p U_{pi}^{\ T}\right) \qquad (2)$$

Manuscript received September 11, 2012; revised December 22, 2012.

doi: 10.12720/joace.1.2.144-148

$$D_{ijk} = \sum_{p=\max(i,j,k)}^{n} Trace\left(U_{pjk} J_p U_{pi}{}^T\right) \tag{3}$$

$$D_i = \sum_{p=1}^{n} -m_p g^T U_{pi} \bar{r}_p \tag{4}$$

where J_i is the pseudo inertia matrix for link i and is defined by:

$$J_i = \begin{bmatrix} \frac{-I_{xx}+I_{yy}+I_{zz}}{2} & I_{xy} & I_{xz} & m_i x_i \\ I_{yx} & \frac{I_{xx}-I_{yy}+I_{zz}}{2} & I_{yz} & m_i y_i \\ I_{zx} & I_{zy} & \frac{I_{xx}+I_{yy}-I_{zz}}{2} & m_i z_i \\ m_i x_i & m_i y_i & m_i z_i & m_i \end{bmatrix} \tag{5}$$

where m_i is the mass of link i and x_i, y_i, z_i are coordinates of center of link i relative to the link coordinate frame. On the other hand U terms can be defined:

$$U_{ij} = A_1 * A_2 * \ldots * \frac{\partial A_j}{\partial q_j} * A_i \tag{6}$$

where A_j is the transformation matrix of link j, $m_p$is the mass of link p, g^T is the transpose of gravity matrix and $\bar{r}_p$ is the position vector of center of gravity of link p relative to its coordinate frame.In the above equations of motions there are three types of terms. The first type involves the second derivative of the generalized coordinates, it is called Angular Acceleration Inertia term. The second type of terms is quadratic terms in the first derivative of q; these terms are divided into two subtypes. The terms involving product of the type $\dot{q}_i{}^2$ are called Centrifugal terms, while those involving product of type $\dot{q}_i \dot{q}_j$ where $i \neq j$ are called Coriolis terms.For more details the reader is referred to [6] and [7].

It is obvious from the general equation of motion that it is based on finding value of D terms; so the MATLAB program described in this paper is developed to find those D terms. The program consists of main module which calculates angular acceleration inertia terms, Coriolis and Centrifugal forces, and gravity terms for each link torque by looping and invoking other procedures to calculate terms D_{ij}, D_{ijk}, and D_i. Once the different D terms are calculated, they will be inserted in equation (1) to find the equations of motion for each link. These equations of motions are introduced in a simple form and are verified using the Lagrange equation alone.

TABLE I. MAIN FEATURES OF THE PROPOSED CODE MODULE.

Input	Description
T	This variable is a matrix which results from concatenation of all A matrices in one matrix (i.e. $T = [A_0 A_1 \ldots . A_n]$
n	This variable is number of robot degrees of freedom
G	This variable is concatenation of gravity matrices
configuration	This is a vector expressing each link type; each element in vector can values 0 and 1, where 0 indicates a revolute joint and 1 a prismatic joint
symbols	This is a matrix containing robot links masses and lengths

The developed program is able to find equations of motion for any general multi-axis robot giving robot configuration and its A matrices. Table I describes program inputs in details.

Program main units are:

A. Model

The model is the main procedure of the program. It is used to do summations and to calculate D terms which are needed to be computed.

B. GetD2

This module is used to calculate terms D_{ij}.

C. GetD3

This module is used to calculate terms D_{ijk}.

D. GetD1

This module is used to calculate terms D_i.

The following is a flowchart for the main component of the implemented program

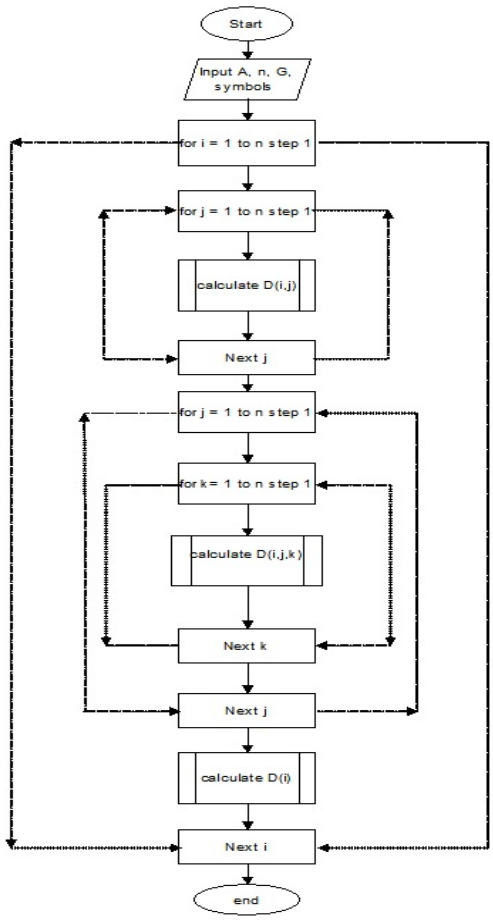

Figure 1. Flowchart of main module

III. ALGORITHM IMPLEMENTATION

The program was tested using a three link manipulator with revolute joint and it was verified against analytical solution using Lagrange equation only.

Consider a 3 DOF planar robot arm as shown in Fig. 2. The robot moves in the vertical plane so the gravity effect will be included in the analysis.

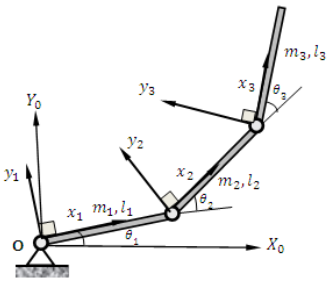

Figure 2. 3 DOF planner robot

The following table introduces the parameters based on Denavit-Hartenberg notations

TABLE II. PARAMETER TABLE

Link #	θ	d	a	$\propto$
Link 1	θ_1	0	l_1	0
Link 2	θ_2	0	l_2	0
Link 3	θ_3	0	l_3	0

And the following table identifies the model parameters as:

TABLE III. MODEL PARAMETERS

Joint variables	$\theta_1, \theta_2, \theta_3$
Mass of links	m_1, m_2, m_3
Link parameters	l_1, l_2, l_3

Thetransformation matrices can be obtained from the parameters tables as:

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & l_1c_1 \\ s_1 & c_1 & 0 & l_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & l_2c_2 \\ s_2 & c_2 & 0 & l_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & l_3c_3 \\ s_3 & c_3 & 0 & l_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The proposed program will use these matrices along with the robot parameters as inputs and the simulation will be carried out as shown in the flowchart. The required equations of motion for the three links are given in the form:

$$\begin{aligned}
T_1 = & \begin{bmatrix} \frac{1}{3}m_2\left(3l_1^2 + 3l_1l_2c_2 + l_2^2\right) \\ + \frac{1}{3}m_3\left(3l_1^2 + 6l_1l_2c_2 + 3l_1l_3c_{23} + 3l_2^2 + 3l_2l_3c_3 + l_3^2\right) + \frac{1}{3}m_1l_1^2 \end{bmatrix}\ddot{q}_1 \\
& + \begin{bmatrix} \frac{1}{6}m_3\left(6l_2^2 + 6l_2l_3c_3 + 6l_1l_2c_2 + 2l_3^2 + 3l_1l_3c_{23}\right) \\ + \frac{1}{6}m_2l_2\left(2l_2 + 3l_1c_2\right) \end{bmatrix}\ddot{q}_2 \\
& + \left[\frac{1}{6}m_3l_3\left(2l_3 + 3l_1c_{23} + 3l_2c_3\right)\right]\ddot{q}_3 + I_{1(act)}\ddot{q}_1 \\
& + 2\left[-\frac{1}{2}m_3l_1\left(l_3s_{23} + 2l_2s_2\right) - \frac{1}{2}m_2l_1l_2s_2\right]\dot{q}_1\dot{q}_2 \\
& + \left[-\frac{1}{2}m_3l_1\left(l_3s_{23} + 2l_2s_2\right) - \frac{1}{2}m_2l_1l_2s_2\right]\dot{q}_2^2 \\
& + \left[-\frac{1}{2}m_3l_3\left(l_1s_{23} + l_2s_3\right)\right]\dot{q}_3^2 + 2\left[-\frac{1}{2}m_3l_3\left(l_1s_{23} + l_2s_3\right)\right]\dot{q}_1\dot{q}_3
\end{aligned} \tag{7}$$

$$\begin{aligned}
& + 2\left[-\frac{1}{2}m_3l_3\left(l_1s_{23} + l_2s_3\right)\right]\dot{q}_2\dot{q}_3 + [gm_3\left(l_1c_1 + l_3c_{123} + l_2c_{12}\right) \\
& + gm_2\left(l_1c_1 + l_2c_{12}\right) - \frac{1}{2}gm_2l_2c_{12} + \frac{1}{2}gm_1l_1c_1 - \frac{1}{2}gm_3l_3c_{123}] \\
T_2 = & \begin{bmatrix} \frac{1}{6}m_3\left(6l_2^2 + 6l_2l_3c_3 + 6l_1l_2c_2 + 2l_3^2 + 3l_1l_3c_{23}\right) \\ + \frac{1}{6}m_2l_2\left(2l_2 + 3l_1c_2\right) \end{bmatrix}\ddot{q}_1 \\
& + \left[\frac{1}{3}m_3\left(3l_2^2 + 3l_2l_3c_3 + l_3^2\right) + \frac{1}{3}m_2l_2^2\right]\ddot{q}_2 \\
& + \left[\frac{1}{6}m_3l_3\left(2l_3 + 3l_2c_3\right)\right]\ddot{q}_3 + I_{2(act)}\ddot{q}_2 \\
& + \left[\frac{1}{2}m_3l_1\left(l_3s_{23} + 2l_2s_2\right) + \frac{1}{2}m_2l_1l_2s_2\right]\dot{q}_1^2 \\
& + \left[-\frac{1}{2}m_3l_2l_3s_3\right]\dot{q}_3^2 + 2\left[-\frac{1}{2}m_3l_2l_3s_3\right]\dot{q}_1\dot{q}_3 \\
& + 2\left[-\frac{1}{2}m_3l_2l_3s_3\right]\dot{q}_2\dot{q}_3 \\
& + \left[gm_3\left(\frac{1}{2}l_3c_{123} + l_2c_{12}\right) + \frac{1}{2}gm_2l_2c_{12}\right]
\end{aligned} \tag{8}$$

$$\begin{aligned}
T_3 = & \left[\frac{1}{6}m_3l_3\left(2l_3 + 3l_1c_{23} + 3l_2c_3\right)\right]\ddot{q}_1 + \left[\frac{1}{6}m_3l_3\left(2l_3 + 3l_2c_3\right)\right]\ddot{q}_2 \\
& + \left[\frac{1}{3}m_3l_3^2\right]\ddot{q}_3 + I_{3(act)}\ddot{q}_3 + \left[\frac{1}{2}m_3l_3\left(l_1s_{23} + l_2s_3\right)\right]\dot{q}_1^2 \\
& + 2\left[\frac{1}{2}m_3l_2l_3s_3\right]\dot{q}_1\dot{q}_2 + \left[\frac{1}{2}m_3l_2l_3s_3\right]\dot{q}_2^2 + \left[\frac{1}{2}gm_3l_3c_{123}\right]
\end{aligned} \tag{9}$$

IV. ALGORITHM IMPLEMENTATION

The final model equations of motions can be used to find torque on each link at any time knowing the prescribed trajectory for each joint. Fourth-order polynomial trajectory with rest-to-rest motion is assumed in the form:

$$\theta(t) = c_0 + c_1t + c_2t^2 + c_3t^3 + c_4t^4 \tag{10}$$

where the coefficients c_0, c_1, c_2, c_3, c_4 are constants that to be determined from the initial and final conditions. Initial and final positions as well as rest-to-rest motion reduces the unknown coefficients to only one. For the non optimized case, the fourth coefficient vanishes (third-order polynomial) and the initial and final conditions are sufficient to determine the coefficients $c_0, c_1, c_2, and\ c_3$. For the optimized case we need to find the fourth coefficient. It is the objectiveof this analysis to find how this coefficient can optimize the energy consumption for the robot arm. The cost function under consideration can be assumed as:

$$T = \sqrt{T_1^2 + T_2^2 + T_3^2} \tag{11}$$

This function was used by Garg and Kumar [8]. To find the optimized value of this coefficient there are many techniques to be used and the most of them are heuristics techniques like Genetic Algorithm, Neural Networks, and Particle Swarm Optimization. The heuristic technique is preferable because of the hardness or even disability to

use gradient techniques which requires a lot of time to find the solution. But there is a need first to ensure that the optimization would get a better solution than the non optimized case so analytical optimization analysis procedure is used here. This analytical optimization technique is a novel efficient way and it is used as a benchmark to ensure the benefit when using any optimization technique.

A small program is implemented in C programming language that takes equations of motion as input and also initial and final desired positions. In this program a loop that tries a range of possible values for this coefficient and evaluates the objective function and this range is changed manually to see the best possible value. Analytical analysis was done for initial position $(\theta_i) = 0$ radian and final position $(\theta_f) = 1$ radian for each link and travelling time of 5 seconds. Each link has mass of 0.5 kg and 1 meter in length. The optimized value for control variable that is obtained from program is -0.17 and the optimized case is compared to non optimized case and the numerical results are shown in figures 3, 4, and 5 respectively. Fig. 3 shows the comparison between the optimized and the optimized case:

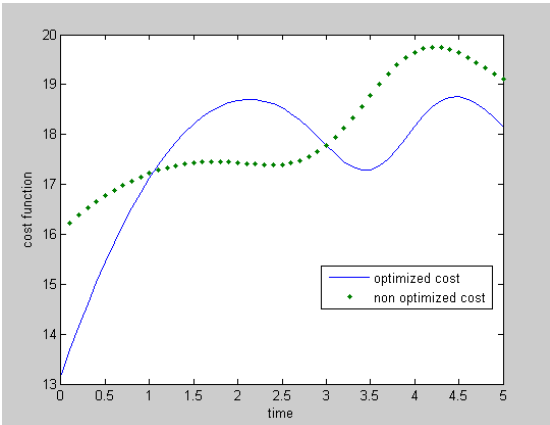

Figure 3. Optimized objective function versus non optimized

While in Fig. 4 the optimized torque for each link during time interval is shown

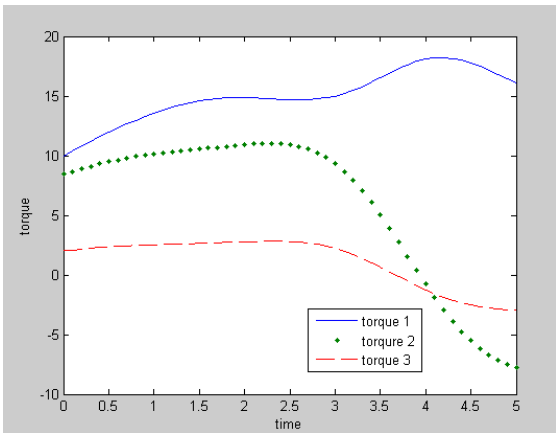

Figure 4. Torques on each link for optimized case

And Fig. 5 shows changing of position, velocity, and acceleration with time in the case of optimized value.

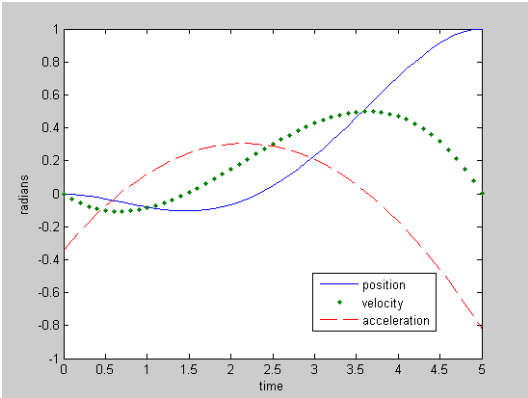

Figure 5. Optimized trajectory for each joint

V. Conclusions

An algorithm to find the equations of motion for multi-link robotic arm is presented in this paper. The implemented program can be used to find equations of motions for any robot configuration and this would save a lot of time that will be spent in solving many equations especially when number of DOF is high. Robot equations of motion can be used in optimal trajectory planning and as a benchmark of optimization analytical analysis is carried out for a certain set of parameters to ensure the benefit of optimization, so any optimization technique can be used to find the optimal trajectory planning for any set of given parameter.

References

[1] R. Featherstone and D. Orin, "Robot Dynamics: Equations and Algorithms," in *Proc. IEEE International Conference on Robotics and Automation*, 2000.

[2] J. M. Hollerbach, "A Recursive Lagrangian Formulation of Maniputator Dynamics and a Comparative Study of Dynamics Formulation Complexity Systems, Man and Cybernetics," *IEEE Transactions*, vol. 10, no. 11, pp. 730-736, 2007.

[3] W. Khalil, "Modeling of Rigid Robots," Robotics-INRIA-UNESCO Summer School, Nice-France, June 1992.

[4] W. M. Silver, "On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators," *The International Journal of Robotics Research*, vol. 1, no. 2, pp. 60-70, 1982.

[5] L. C. Lin and K. Yuan, "A Lagrange - Euler - assumed modes approach to modeling flexible robotic manipulators," *Journal of the Chinese Institute of Engineers*, 1988.

[6] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Dynamics and Control*, Second Edition, January, 2004.

[7] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*, Springer Verlag, 2000.

[8] D. P. Garg and M. Kumar, "Optimization techniques applied to multiple manipulators for path planning and torque minimization," *Journal for Engineering Applications of Artificial Intelligence*, vol. 15, pp. 241-252, 2002.

Dr. Atef A. Ata was born in Alexandria, Egypt in 1962 and received his B. Sc. Degree with Honor in Mechanical Engineering from Alexandria University in Egypt in 1985. After his graduation he joined the same university as a Lecturer where he obtained his M. Sc. Degree in Engineering Mathematics (Hydrodynamics) in 1990. In 1996 he obtained his Ph. D in Engineering Mathematics (Robotics) as a Joint-

Venture between University of Miami, Florida, USA and Alexandria University in Egypt. He joined Alexandria University again as Assistant Professor till 2001. Then he joined the Mechatronics Engineering Department, International Islamic University Malaysia as an Assistant Professor , Associate Professor (2004) and as a Head of Department (November 2005-June 2007). Currently he is a Professor of engineering mechanics at the Faculty of Engineering, Alexandria University, Egypt. Dr. Atef is a senior member of IACSIT as well as Egyptian Engineering Syndicate. He was also one of the Editorial Consultant Board for 2006-2008 for the International Journal of Advanced Robotic Systems, Austria**.** Currently he is one of the Editorial board of Mechanical Engineering Research, Canada and an Associate Editor of Alexandria Engineering Journal, Egypt hosted by Elsevier. His research interest includes Dynamic and Control of Flexible Manipulators, Trajectory Planning, Genetic Algorithms and Modelling and Simulation of Robotic Systems.

Dr. Mohammed A. Ghazy received the Bachelor of Science in Textile Engineering (Spinning) from University of Alexandria, Egypt with degree of honor in 1998. In 2000 he enrolled in the same institution to pursue the Master of Science degree in Engineering Mathematics which he received in 2004. He was formerly employed with the Department of Engineering Mathematics and Physics, University of Alexandria, as a Teaching Assistant from 2000 to 2004. Mohammed Ghazy was enrolled in 2006 in the Department of Aerospace Engineering, Old Dominion University to pursue the Doctor of Philosophy degree and was awarded a Doctoral Fellowship in August 2009. In May 2010 he received the Batten College of Engineering Excellence Award in Aerospace Engineering, and he graduated on August 2010. From June 2011 till now he is working as an assistant professor at the Department of Engineering Mathematics and Physics, University of Alexandria, Egypt.

Mohamed A. Gadou received his B.Sc. Degree with Honor in Computer Systems and Engineering from Alexandria University in Egypt in 2008. In 2009 he was formerly employed as a Teaching Assistant in the Department of Engineering Mathematics and Physics, University of Alexandria. In 2010 he enrolled in the same University to pursue his M.Sc. in Engineering Mathematics. Currently he is preparing for his master thesis.