

Dynamic Modeling and Simulation of Articulated Robotic Arm with MATLAB Robotics System Toolbox

¹Khin Soe Nwe, ²Wai Phyto Maung, ³Ei Ei Htwe

¹Student, ²Lecture, ³Professor

¹Department of Mechanical Engineering,

¹Mandalay Technology University, Mandalay, Myanmar

Abstract: In this paper, dynamic behavior of a six-axis robotics system for pick and place process through a defined trajectory using MATLAB Robotics System Toolbox is presented. Robotics System Toolbox™ provides algorithms and hardware connectivity for developing autonomous robotics applications for aerial and ground vehicles, manipulators, and humanoid robots. The Euler-Lagrange equations have several very important properties that can be exploited to design and analyze feedback control algorithms. Trajectory for pick place operation is generated by using this toolbox. Joint configurations for a desired end-effector pose is calculated using Rigid body tree kinematics Simulink block. Dynamic simulations are carried out on the model to examine the properties such as torque variation at the joints and trajectory of the manipulator. According to the simulation result, the manipulator has expected response similar to physical model. The maximum joint torque required about 3 Nm and the minimum joint torque required about 0.04 Nm at the articulated robotic arm.

Index Terms: MATLAB Robotics System Toolbox, Euler-Lagrange, trajectory, Simulink, dynamic simulation.

I. INTRODUCTION

Analyzing a robotic system requires software packages that can perform mathematics, control, data acquisition, communication with external devices, image processing, real time data manipulation, 3D visualization, etc., and also programming support under a single platform. Graphical programming language is always an intuitive way to use robotic simulation. There are packages like LabVIEW [1], MATLAB Simulink [2], Simscape Multibody [3], [4] and others [5] which support such an environment for programming and visualization. Robotics system toolbox provides an interface between MATLAB and Simulink and provides functions for manipulator algorithms discarding the tedious derivation of equation of motion. None of the literature survey so far reported robotics manipulator dynamic study using MATLAB Robotics System Toolbox. Hence, the dynamic simulation of a robotics manipulator was carried out here to observe its behavior and check its performances as it was planned to be procured for some real-life applications. And comparison analysis is performed by comparing with Euler-Lagrange dynamic calculation. This forms a major contribution of the paper. MATLAB Robotics System Toolbox has been used to perform kinematic and dynamic formulation and results are validated using physical simulation with Simscape Multibody. Virtual simulations being a demanding field in robotics, several attempts have been made by the researchers to create and simulate robots in virtual environments.

II. EULER-LAGRANGE METHOD OF ROBOTIC ARM

The dynamic equations of any mechanical system can be obtained from the known Newtonian classical mechanics, the drawback of this formalism is the use of the variables in vector form, complicating considerably the analysis when increasing the joints or there are rotations present in the system. In these cases, it is favorable to employ the Lagrange equations, which have formalism of scale, facilitating the analysis for any mechanical system. The dynamics of serial manipulator can be solved by applying the Euler-Lagrange equation. The formulation is defined as the difference between the kinetic and potential energies of the 6-DOF serial manipulator part. The Lagrange's approach is described here briefly for serial link manipulators with rigid links. The specification of articulated robotic arm data are shown in Table 1.

Table 1. Specification of Articulated Robotic Arm

Quality	Symbols	Values	Units
Density (Aluminum)	ρ	2700	kg/m ³
Radius of link 1	R	50	mm
Length of link 2	L ₂	100	mm
Length of link 3	L ₃	63.15	mm
Length of link 4	L ₄	50	mm
Length of link 5	L ₅	63.15	mm
Length of link 6	L ₆	120	mm

Kinetic Energy

The kinetic energy of a rigid object is the sum of two terms: the translational energy obtained by concentrating the entire mass of the object at the center of mass, and the rotational kinetic energy of the body about the center of mass. The expression for the kinetic energy of i^{th} link of the manipulator is given by KE_i as:

$$KE_i = \frac{1}{2} v_{cm_i}^T \cdot m_i \cdot v_{cm_i} + \frac{1}{2} \cdot \omega_{B_i}^T \cdot [I_{cm_i}] \cdot \omega_{B_i} \quad (1)$$

The total kinetic energy of the manipulator can be written as the sum of all the kinetic energies of the links of the manipulator. That is,

$$KE_{system} = \sum_{i=1}^N KE_i \quad (2)$$

Potential Energy

The potential energy of the i^{th} link can be computed by assuming that the mass of the entire object is concentrated at its center of mass and is given by

$$PE_i = m_i g h_i \quad (3)$$

The total potential stores in the manipulator is the sum of the potential energy in the individual links, that is,

$$PE_{system} = \sum_{i=1}^N KE_i \quad (4)$$

Euler-Lagrange Equations

In general, the Lagrangian of a manipulator system can be defined as a function of the generalized coordinates:

$$L = KE_{system} - PE_{system} \quad (5)$$

The Euler-Lagrange formula will be used to drive the equation of motion for robotic manipulator, that is,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = Q_k \quad (6)$$

The generalized forces acting on the system,

$$Q_k = \sum_{i=1}^{N_{fnc}} (\vec{F}_i \cdot \frac{\partial \vec{v}_i}{\partial \dot{q}_k}) + \sum_{j=1}^{N_{\tau nc}} (\vec{\tau}_j \cdot \frac{\partial \vec{\omega}_j}{\partial \dot{q}_k}) \quad (7)$$

The dynamic equation of motion of the robotic arm,

$$M(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + K(q, q) + G(q) = Q \quad (8)$$

III. DYNAMIC MODELING EULER-LAGRANGE

The implementation of the block diagram was performed to numerically solve the dynamic equations of the system, thus obtaining torque values along the defined time interval in Fig 1. Wherein, the block MATLAB Function Input, has the input applied to the system, which in this case is joint angles, obtained from inverse kinematic sub system block with input trajectory way points. The block called MATLAB Function System, contains the mathematical model in its matrix representation, with robot parameter inputs and torque outputs. The derivative blocks are used to obtain angular speed and acceleration respectively.

Numerical values necessary to solve the Euler-Lagrange's equations of motion are shown as Table 2.

Table 2. Numerical Variable System

Physical characteristics	Symbols	Values	Units
Acceleration of gravity	g	9.81	m/s ²
Mass of link 1	mass_1	0.202	kg
Mass of link 2	mass_2	0.042	kg
Mass of link 3	mass_3	0.195	kg
Mass of link 4	mass_4	0.021	kg
Mass of link 5	mass_5	0.195	kg
Mass of link 6	mass_6	0.334	kg

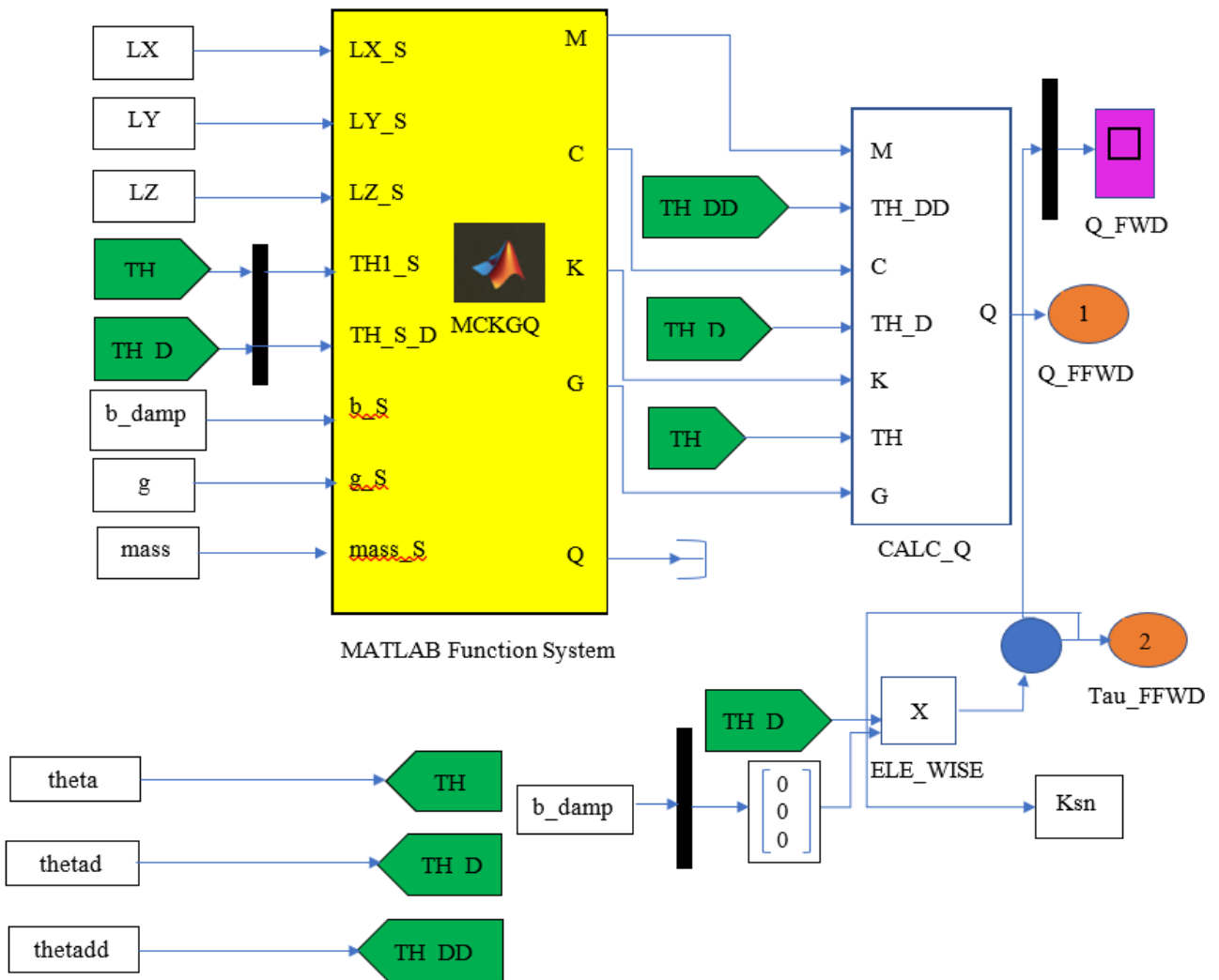


Figure-1. Block diagram of Euler-Lagrange's equations

IV. IMPORTING CAD ASSEMBLY AND SIMULATION

The initial step is to create CAD model of the six degrees of freedom robotic arm that is going to be developed. SolidWorks was used to develop CAD model of articulated robotic arm parts and its assembly shown in Fig 2. 3D model of the 6 degrees of freedom robotic arm was designed by using Xacro macro language and mesh files generated from SolidWorks by saving as STL file format. For the robot Xacro file is the best way to create readable and shorter XML code. The design of the robotic arm includes the joint coordinates, orientations (roll, pitch, yaw) link lengths and so on.

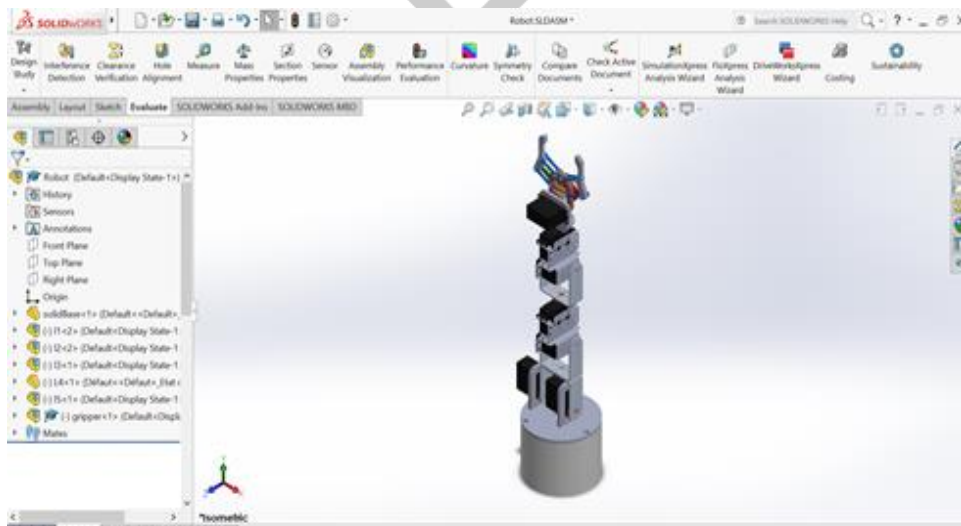


Figure-2. Assembly of six degree of freedom robotic arm

In the Xacro file, link lengths were fixed to variables so that link length can be reused later by giving the respective variables. For the link of the robotic arm shape, size and colour has to be given. Mesh link can also be inserted that can be called at any instant at a later point of time. The position of the link in the space along with the position of the same link with respect to its parent link has to be mentioned properly. Joints are specified with kinematics, dynamics and set of angle limits of the joint. The joint tag represents the joint of the robotic arm. This tag includes different types of joints such as revolute, prismatic, fixed, planar and so on. The joint is formed with a parent link and a child link. The link details such as orientation, joint measurements and angle limits has to be included in the child link and parent link properly in the file. 6 degree of freedom robotic arm was designed with all the respective measurements and the joint angle limits to get a precise result. The rigid body tree model for MATLAB Robotics System Toolbox from urdf file can be created by the following command:

```
robot = importrobot('filename.urdf')
```

A manipulator task is usually given in terms of desired end-effector trajectory. Firstly, setting the desired way points of end effector, the trajectory is generated by using cubic spline curve fitting function. shown in Fig 3.

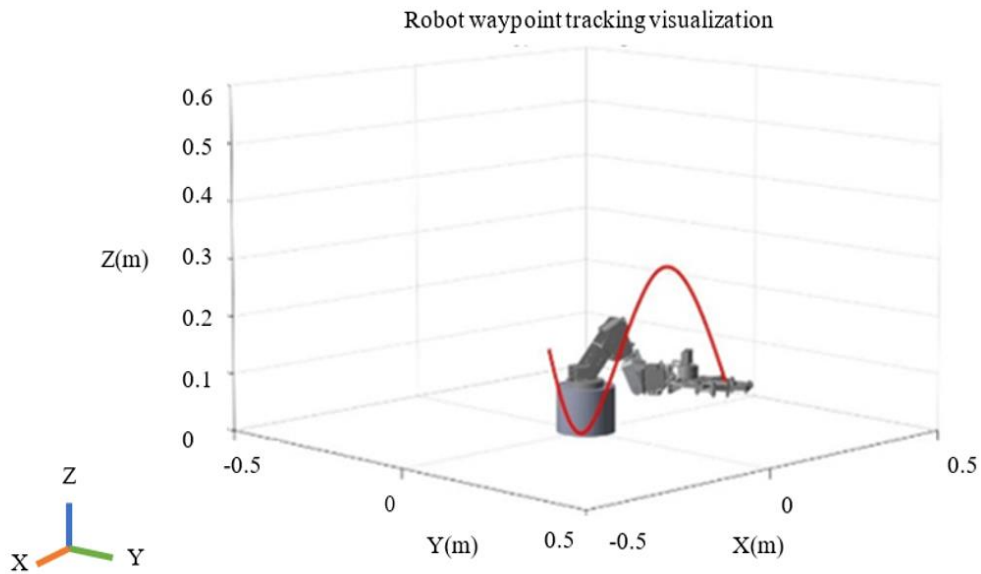


Figure-3. Trajectory planning of robotic arm

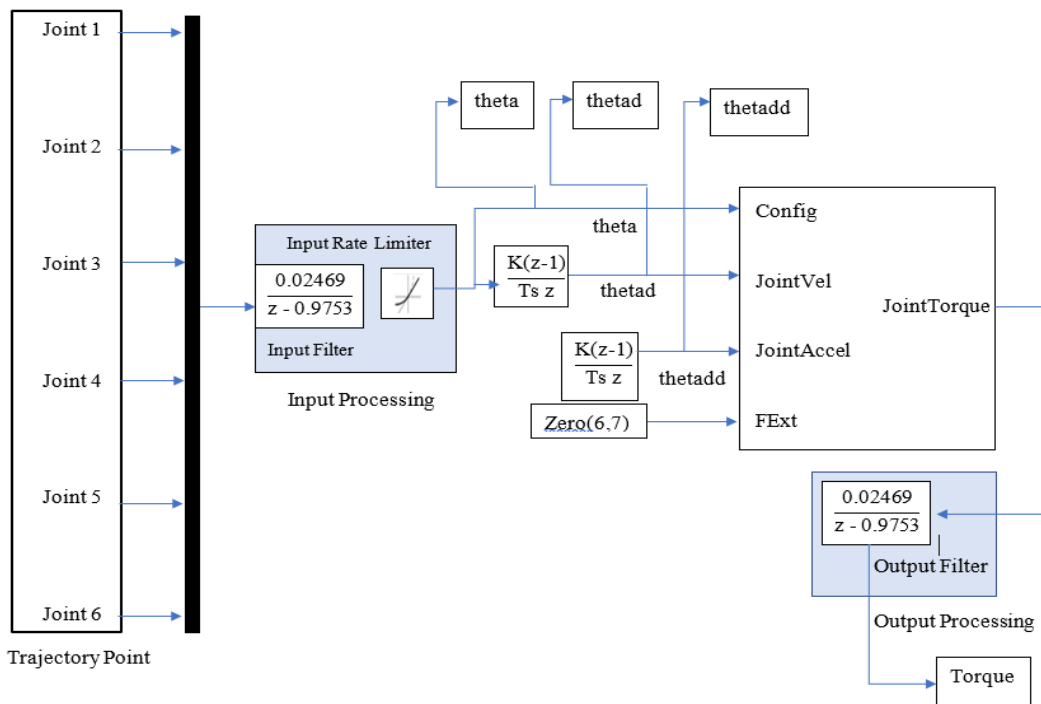


Figure-4. Inverse Dynamic System in Simulink

As an optional extension of the MATLAB software package, SIMULINK provides in order to achieve the dynamic systems models which are represented in the block diagram Fig 4. A large library, including a lot of different blocks, is available for the user. This allows fast and accurate systems modeling, without having to write even a line of simulation code. The 3D coordinate points of trajectory are fed to inverse kinematic sub system block in which the associated robotics rigid body tree model is assigned

and sets of joint angles for each robot configuration is acquired. The joint position information is passed through discrete derivative blocks and the required joint velocities and accelerations are obtained. The joint positions, velocity and acceleration are fed to the inverse dynamic block and the required torque to drive the robot along a given joint trajectory is computed.

V. RESULTS AND DISCUSSION

The computed joint torque is used to drive the robot model in Simcape Multibody and validate the movement along the trajectory. Same trajectory and joint configurations are also provided to Euler-Lagrange System. And then the torque results of Robotic System Toolbox and the results of theoretical Euler-Lagrange equation block are compared. It was observed that the torques obtained by both approaches are in close agreement to each other. The variation of joint torques with time during along the trajectory can be seen in Fig 5. Base joint, shoulder joint and elbow joint are the manipulator of the robot. Base joint (Joint 1) occurred maximum torque at 1.9 Nm and minimum torque at -1.5 Nm with respect time. Shoulder joint (Joint 2) is the rotary joint, this joint carried the weight of the another joint. So, the shoulder joint required the maximum torque. According to the result, the maximum torque about 3 Nm occurred at the shoulder joint. The required joint torque of the elbow joint (Joint 3) is less than base joint. The value of the elbow joint is 1.6 Nm. The wrist of the robot is defined joint 4, joint 5, joint 6. In the wrist three joint, the maximum torque at 1.2 Nm occurred in joint 4 and the minimum torque at -0.04 Nm take place in joint 6. It is importance to know the torques of each joint that will drive the robotic arm because it is necessary to decide selection of the motor for the system.

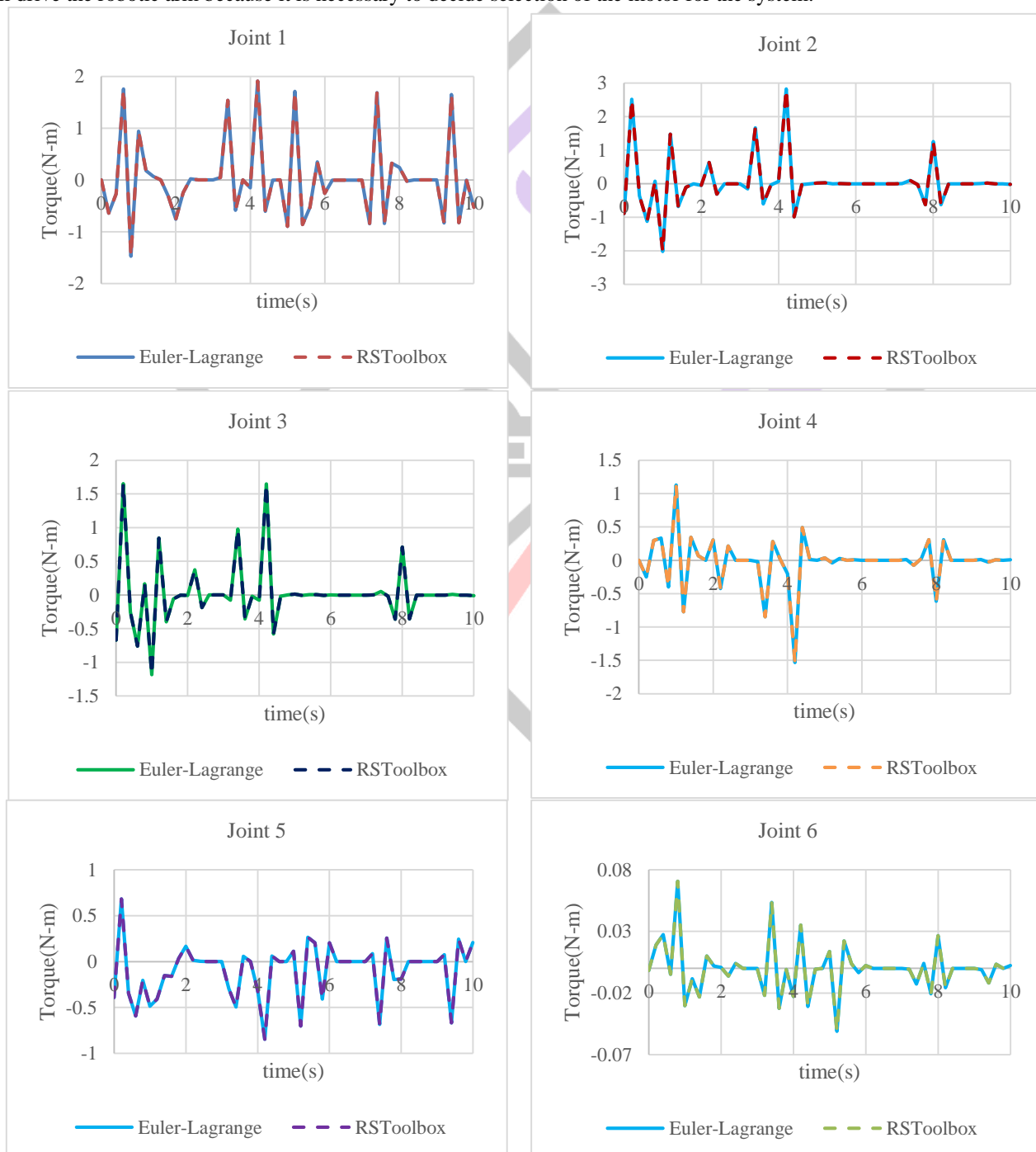


Figure-5. Required Torques of each joint

VI. CONCLUSIONS

The Unified Robot Description Format (URDF) file make the robotic modelling work convenient and smart. MATLAB Robotic System Toolbox support the required algorithms for robot modeling and control. The method used is based on Euler-Lagrange equations. The structural modeling is done by defining connection parameters and dynamics of each element, generating Simulink dynamic equations necessary to solve the problem. The kinematic analysis, dynamic analysis and control design can be validated by Simscape Multibody and co-simulation with other software package is not needed. This paper presents a different approach to inverse kinematics analysis of an articulated robotic arm with open chain mechanism. Inverse dynamic problem of an articulated robot is solved by using MATLAB Robotic System Toolbox and Euler-Lagrange equation. Maximum torque takes place at joint 2 (the shoulder joint) and the minimum torque takes place at joint 6. According the trajectory, maximum torque occurring at the shoulder joint (about 3 Nm) is logical. As a result of the paper, proposed analysis method verified by simulations and derived torques of each joint are given in the form of the graphics. The main contribution of the paper to the literature is that different type inverse dynamic analysis approach is implemented. Furthermore, paper can be control studies of the robots for the future works.

VII. ACKNOWLEDGMENT

The first of all, the author wish to express my deep gratitude to Dr. Sint Soe, Pro-Rector, Mandalay Technological University, for his kindness valuable permission to summit this paper. The author also grateful to Dr. Htay Htay Win, Professor and Head, Mechanical Engineering Department, Mandalay Technological University, for her helpful, invaluable suggestions and comments during the submit this paper. Moreover, special thanks to my supervisor Dr. Wai Phyo Maung for his kind supervision and encouragement, suggestions valuable guidance for throughout the entire process of this research work. This paper would not have been possible without his support. The author also would like to express her sincere and deepest gratitude to her co-supervisor Dr. Ei Ei Htwe, Mechanical Engineering Department, Mandalay Technological University, for her kindness, understanding, invaluable suggestions and feedback. Finally, the author would love to say thank to her family especially her parents for their noble support, encouragement, and guidance throughout her entire life.

REFERENCES

- [1] H. G. Sage, M. F. De Mathelin, and E. Ostertag, "Robust control of robot manipulators: A survey," *International Journal of Control*, vol. 72, no. 16, pp. 1498–1522, 2000.
- [2] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons Hoboken, NJ, 2006.
- [3] W. Lu and Q. Meng, "Regressor formulation of robot dynamics: computation and applications," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 3, pp. 323–333, 2011.
- [4] Alireza Khatamian, "Solving Kinematics Problems of a 6-DOF Robot anipulator" *Int'l Conf. Scientific Computing | CSC'15*
- [5] Bikash Kumar Sarkar, "Modeling and validation of a 2-DOF parallel manipulator for pose control application", *Robotics and Computer-Integrated Manufacturing* no. 50 pp. 234–241, 2018.
- [6] Zhijiang Du, Yixuan Sun, Yanyu Su, Wei Dong, "A ROS/Gazebo Based Method in Developing Virtual Training Scene for Upper Limb Rehabilitation", *International Conference on Progress in Informatics and Computing*, 14790784, 2014.
- [7] Qunshan Xu, Jianghai ZHAO, Chunxia Zhang and Feng He, "Design and Implementation of an ROS based Autonomous Navigation System", *Proceedings of International Conference on Mechatronics and Automation*, 2015.
- [8] G. D. Wood, and D. C. Kennedy, "Simulating Mechanical Systems in Simulink with SimMechanics," *Tech. Report*, The Mathworks Inc., 2010.
- [9] I. Hassanzadeh, A. Harifi, and F. Arvani, "Design and Implementation of an Adaptive Control for a Robot," *Am. J. of Appl. Sc.* 4(2), pp: 56-59, 2017.
- [10] S. K. Saha, *Introduction to Robotics*, New Delhi: Tata McGraw Hill, 2008. M. Young, *The Technical Writer's Handbook*.