Stability Simulation of Six DOF manipulators using ADAMS and MATLAB

V Vennishmuthu [1], S.Riyaz Ahammed [2], R.Hushein[3] , Karthik.P[4]

[1,2,3,]Assistant Professor, Department of Mechanical Engineering, Vel Tech Dr.RR & Dr.SR Technical University, Avadi, Chennai, Tamil Nadu, India

[4]Assistant professor, Department of Mechatronics Engineering, Trichy Engineering College , Konalai, Trichy Tamil Nadu, India

Abstract: - In cooperative manipulation, more than one robots deal with a single payload. The load distribution of one robot depends on the configuration and motion of the other robot. For successful cooperation, the two manipulators need to operate under a single controller. A number of software such as MSC.ADAMS is popular for the dynamic simulation of mechanical systems. The software also allows co-simulation with control system software such as MATLAB. In the present work, it is proposed to simulate the motion and dynamics of cooperative manipulators using MSC.ADAMS and MATLAB. The primary aim of the work will be to consider an industrial robot manipulator from KUKA or FANUC as manipulators involved in a cooperative task. While MSC.ADAMS will be used for estimation of forces and motion simulation, MATLAB will be used for generation of feasible trajectories for successful cooperative manipulation.

I.   INTRODUCTION

In recent decades, robot manipulators are widely used in industrial applications. Of the most important ones, welding and painting robots in car plants, electronic board assembly, repairing nuclear installations, and etc can be mentioned. Besides, academic and research Labs try to develop new methods and algorithms so as to release the results for industrial uses after validation. The researchers are applied to different areas such as motion planning, manipulation planning[1,2,3] and control algorithms like position control, force control, Impedance control and etc. Real robots are inaccessible due to high prices therefore creating precise simulated models are extensively demanded by research Labs. With the advent of new models of robot manipulators by manufacturers, still there is a need for exact simulated models. This paper introduces simulated models of 6-DOF robot. All procedures presented in this paper are in way that reader can model specific robot in software environments. The main advantage of this paper is to use multi-software environments interactively. After a 3D modeling of robot manipulator in PRO ENGINEER software, the model is exported to ADAMS dynamic modeling software. This software is capable of simulating kinematics and kinetics behavior of any mechanical system. As result dynamic behavior of systems with comparison to the real models can be simulated with an acceptable error. For obtaining an exact modeling, inertial and geometric parameters are accurately measured and recorded in the software database. Also for simulating control algorithms, co simulation between ADAMS and MATLAB is performed.

II.   MODELING

In modeling using PRO ENGINEER software, robot model is divided into seven parts. Each individual part is designed in "Part Design" separately. Afterwards these parts are reproduced in "Assembly" environment and assembled with constraint commands. The process of defining constraints starts with fixing the base part using "Fix" command and ends up with reproducing the other parts and applying "Mate" and "Align" to each couple of them. To validate the motion of 6-DOF robot, "manipulation" command is used. The assembled model of robot and the real one are shown in Figure 1.
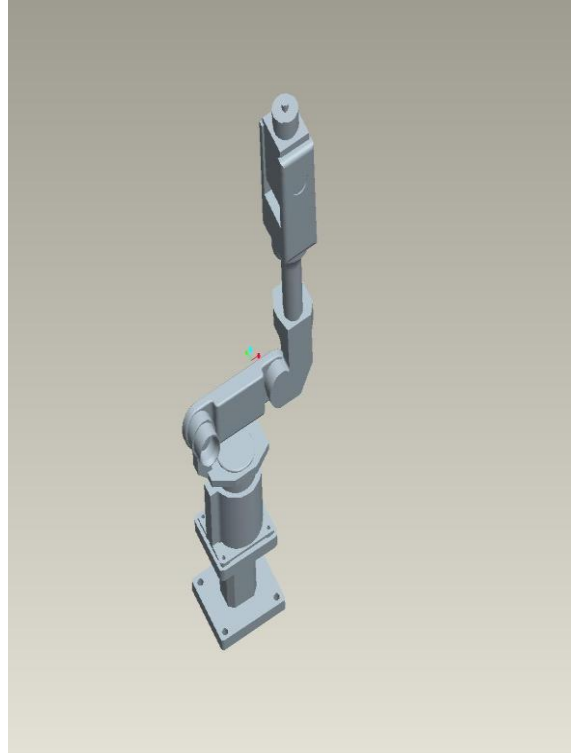
6- DOF ROBOT

III. ADAMS DYNAMIC MODELING SOFTWARE

To import the designed model to ADAMS, the initial step is to transfer each of the seven parts of robot from "Assembly" to "SolidWorks" software. Then each part is saved in "parasolid" format, since this format is better recognized in ADAMS. The next step is to import the model into ADAMS. It should be noted that since the parts are assembled in PRO ENGINEER and converted into a definite format, the parts are put together with pre-determined positions in ADAMS. After transferring model, inertial parameters of the parts such as density, are recorded in software database. The imported model to ADAMS is without mass and inertia in default. Determination of exact density of the robot parts is done by measuring mass and density of the second link lid (Fig. 2). Thus ADAMS is able to calculate the mass and moments of inertia of each part automatically. The values are illustrated in Table 1. The computed parameters are verified by making a comparison between the total weights of simulated robot with the data given in the corresponding catalogue.

Inverse Kinematics:

The inverse kinematics problem consists of the determination of joint variables corresponding to a given end effector position and orientation. The solution to this problem is of fundamental importance in order to transform the motion specifications, assigned to the end effector in operational space, into the corresponding joint space motions that allow execution of desired motion. Inverse kinematics of 6-DOF Stiiubli manipulator consists of establishing closed analytical equations of the joint space using algebraic and trigonometric methods and the final results are specified.

$$\theta_1 = \text{Atan2}(p_y, p_x) - \text{Atan2}\left(d_3, \pm\sqrt{p_x^2 + p_y^2 - d_3^2}\right)$$

$$\theta_2 = \text{Atan2}[(-a_3 - a_2 c_3)p_z + (c_1 p_x + s_1 p_y)(a_2 s_3 - d_4)$$
$$(a_2 c_3 - d_4)p_z + (a_3 + a_2 c_3)(c_1 p_x + s_1 p_y)] - \theta_3$$

$$\theta_3 = \text{Atan2}(a_3, d_4) -$$
$$\text{Atan2}\left(a_3 c_3 - d_4 s_3, \pm\sqrt{a_3^2 + d_4^2 + (a_3 c_3 - d_4 s_3)^2}\right)$$

$$\theta_4 = \text{Atan2}(-\tau_{13}s_1 + \tau_{23}c_1, -\tau_{13}c_1 c_{23} - \tau_{23}s_1 c_{23} + \tau_{33}s_{23})$$

$$\theta_5 = \text{Atan2}(s_5, c_5)$$

$$\theta_6 = \text{Atan2}(s_6, c_6)$$

IV. MATLAB AND ADAMS CO-SIMULATION:

MATLAB is a powerful tool for applying control commands to the robot so for having an efficient simulator package, the co-simulation between ADAMS an MATLAB is noticeable. In order to execute a co-simulation between ADAMS and MATLAB, deformation of an acceptable format for the inputs and outputs of each program is required. The objective of co-simulation is to make a connection so that any change in one of the programs affects the other one [6]. To provide a simulation that enables ADMAS to recognize the exported output from MATLAB, there is a need for activating "control" plug-in in "Plug-in Manager" of ADAMS and defining the robot as a plant. After activation, a new window appears for the determination of plant and its inputs and outputs [7]

Afterwards, ADAMS saves dynamic model of robot motion as a complex of seven matrices that is capable of sending to MATLAB. The first four are A, B, C, D matrices of state-space representation. The fifth and sixth matrices are associated with predefined inputs and outputs. And the last one includes information about state variables of the plant. Trajectory planning of each robot joint, as ADAMS inputs, is stated as five-degree polynomials for solving forward kinematics of robot. This function is suitable for each joint in the view point of dynamic considerations. ADAMS outputs are components of end effectors position (x,y,z) with respect to the frame attached to the base. To call the generated plant in MATLAB, the plant name should be entered in the "command window" so that the input and output information will appear. By executing "Adams_sys" command, the blocks containing information about dynamic model of robot in SIMULINK environment.


model plant and state space blocks
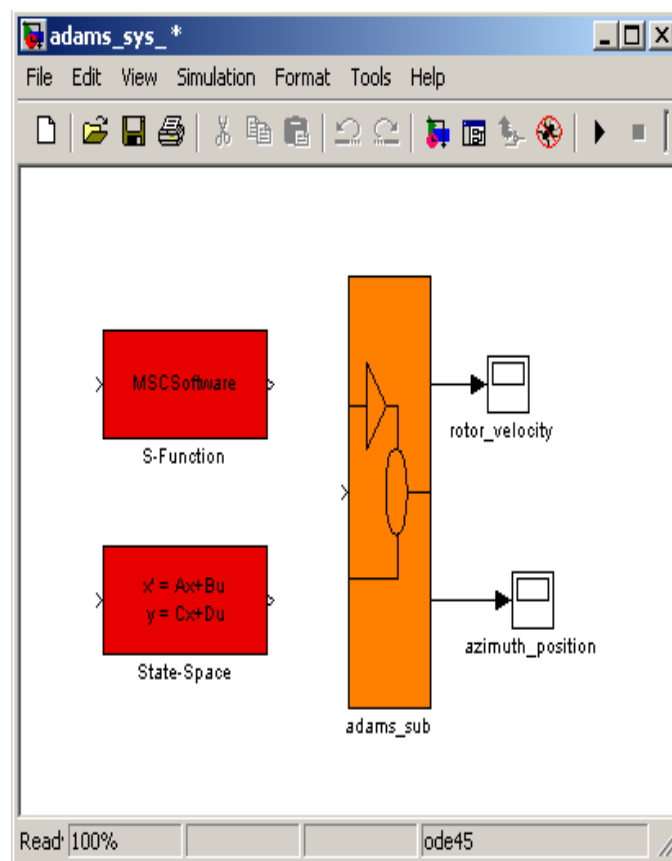
By having "Adams-sub" block in SIMULINK environment, the robot model is suitable for control and motion simulation as a defined system in MATLAB. Regarding the forward kinematics of robot, the final transformation matrix and trajectory planning block are generated in "Embedded MATLAB Function". Outputs of this block represent the plant inputs of robot in

ADAMS. By starting the simulation in SIMULINK, the signals of function coefficients will enter the plant and run ADAMS simultaneously which continues to a definite final state.

Generated trajectory functions for each motor yields the end effector to track the desired position and orientation.

V. CONCLUSIONS

In this paper modeling and simulation of 6-DOF robot have been performed using PRO ENGINEER, ADAMS, MATLAB programs. The Denavit-Hartenberg and inertia parameters and also kinematic analysis of robot were exploited. Then the procedure of establishing a co-simulation between ADAMS and MATLAB has been clearly expressed and the flowchart in Fig. 9 shows the whole process. Consequences of this simulation clarify the correct guidance of robot in the software modeling. With the help of created model, design and simulation of closed loop control systems and applying optimized design parameters are simply executable.
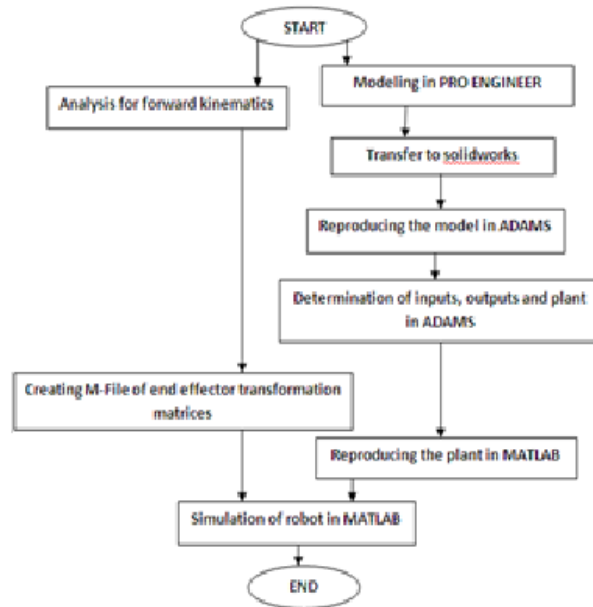
REFERENCES

[1] F. Cheraghpour., S.A.A Moosavian, and A. Nahvi, "Multiple Aspect Grasp performance index for cooperative object manipulation tasks", IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, pp 386- 391, 2009

[2] M. Eslamy, S.A.A Moosavian, "Dynamics and Cooperative object Manipulation Control of Suspended Mobile Manipulators", J Intell Robot Syst Springer Science+Business Media B.V, pp. 1-19, 2010

[3] K. Harada, K. Kaneko, F. Kanehiro, "Fast Grasp Planning for Hand/Arms Systems Based on Convex Model", IEEE International Conference onRobotics and Automation, pp. 1-7, 2008

[4] F. Cheraghpour., S.A.A Moosavian, and A. Nahvi, "Robotic grasp planning by multiple aspects grasp index for object manipulation tasks", 18th Iranian Conf. on Electrical Engineering (ICEE), pp 635-640, 2010

[5] M. Strandberg, B. Wahlberg, "A Method for Grasp Evaluation Based on Disturbance Force Rejection", IEEE transactions on robotics, pp. 1- 9, 2006

[6] Z. Yi, X. Min-min, Q. Jin-yi , Z. Hu , "Research on co-simulation using ADAMS and MATLAB for automobile active suspension system" , Computer Application and System Modeling (ICCASM), International Conference, p.p V14-366 - VI4-370 , 2010.

[7] T. Fraichard , INRIA Rhone Alpes, Grenoble ,"Dynamic trajectory planning with dynamic constraints: A 'state-time space' approach", Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEEIRSJ International Conference , p.p 1393 - 1400 vo1.2, 2002.