Research Journal of Applied Sciences, Engineering and Technology 6(20): 3778-3783, 2013
DOI:10.19026/rjaset.6.3591
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 14, 2013 Accepted: February 22, 2013 Published: November 10, 2013

Research Article

Co-Simulation Control of Robot Arm Dynamics in ADAMS and MATLAB

[1, 2]Luo Haitao, [2]Liu Yuwang, [1, 2]Chen Zhengcang and [1, 2]Leng Yuquan
[1]University of Chinese Academy of Sciences, Beijing 100049, China
[2]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

Abstract: The main objective of this study is how to quickly establish the virtual prototyping model of robot arm system and effectively solve trajectory tracking control for a given signal. Taking the 2-DOF robot arm as an example, a co-simulation control method is introduced to research multi-body dynamics. Using Newton-Euler and Lagrange method, respectively establish the dynamics model of robot arm and verify the correctness of equations. Firstly, the physical model of robot arm was built by PROE and ADAMS. Furthermore, a control model was created in MATLAB/SIMULINK. And then, the co-simulation model was established based on ADAMS/Control and MATLAB/SIMULINK. The simulation results indicate that the robot arm system has preferable response characteristics and nicer locus-tracking ability. The co-simulation method is intuitive and effective. It is no need to create dynamics equation of complicated physical system and has the important practical significance to study manipulation and control for robot arm.

Keywords: ADAMS, co-simulation, dynamics, MATLAB, robot arm

INTRODUCTION

Robot arm is a complex electromechanical system, in which exist complicated coupling relationship and nonlinear feature. It includes mechanical arm ontology performance and control system scheme to evaluate the arm robot. In traditional design method, repeat iteration was adopted. Use this method, it is not only difficult to improve mechanical property of robot arm, but can consume plenty of fund and manpower resource, which is worst that design of mechanical system break away from control system without finding coupling relationship between them (Zong and Li, 2005). If build virtual prototype first during early R&D process before building real model, we can conduct kinematics and dynamic simulation and analysis about arm robot and improve the whole machine mechanical system by simulating control system (Gu *et al*., 2005). So, this simulation method can shorten obviously development cycle and it is also in favor of robot arm dynamic problems and improvement of R&D efficiency.

Developed by MDI Company, ADAMS as software provides powerful model and simulation environment and has strong analysis function about kinematics and dynamics, which is popular in mechanical project fields in the world. While, MATLAB becomes indispensible utility software during scientific research owing to its powerful computed function, visualization of programming and calculating results and high efficiency of programming. For control system design, control toolbox in ADAMS is incomplete; however, SIMULINK section of MATLAB makes up this drawback (Xing, 2012). If combine two software together, in other words, combine mechanical system simulation with design of control system together, advantages of each software are utilized and electromechanical co-analysis is done (Li and Le, 2011). By that way, we can build complex control scheme, in the meantime observing motor process of target object, which is beneficial to research of complex electromechanical system.

In this study, taking the joint angle control of two DOF robot arm as an example, we mainly discuss the control method about co-simulation. After building virtual prototype of robot arm and simulating its kinematic model, the results reflect that the robot arm has well dynamic response and trajectory tracking ability. Also, feasibility of this method is confirmed.

ESTABLISHMENT OF ROBOT ARM DYNAMICS MODEL

For researching robot arm dynamics, firstly, build coordinate system of robot arm and set related dynamic parameters (Soon and Slotine, 2009) (Fig. 1).

Where, the base coordinate system coincides with pivot of link 1. Basic dynamic parameters of links below are defined. m_1 and m_2 are respectively mass of two links; l_1 and l_2 are respectively rotational inertia around Z axis; q_1 and q_2 are respectively angle variable; l_1 and l_2 are respectively length and l_{c1} and l_{c2} are

Corresponding Author: Luo Haitao, University of Chinese Academy of Sciences, Beijing 100049, China

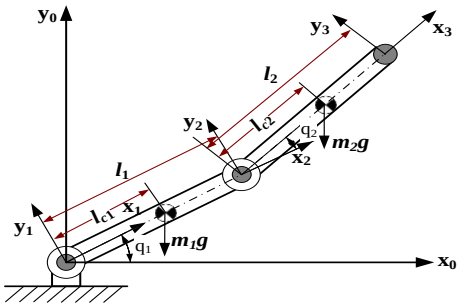


Fig. 1: Coordinate system and dynamic parameters of robot arm

respectively length between centre of mass and joint axis.

Newton-Euler method: First, using homogeneous transformation theory, 4×4 homogeneous transformation matrix ${}^{i}_{i+1}T$, which is from coordinate system {i} to {i + 1}, describes position and pose relationship. And homogeneous transformation matrix ${}^{i}_{i+1}T$ and rotation matrix between links are:

$$ {}^{i}_{i+1}T = \begin{bmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1} & 0 & 0 \\ \sin\theta_{i+1} & \cos\theta_{i+1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

$$ {}^{i}_{i+1}R = \begin{bmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1} & 0 \\ \sin\theta_{i+1} & \cos\theta_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2} $$

Newton-Euler method adopts recursive thought and for link1, initial recursive condition of its dynamic equation is:

$$ {}^{0}\omega_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, {}^{0}\dot{\omega}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, {}^{1}\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix}, {}^{1}\dot{\omega}_1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{q}_1 \end{bmatrix} \tag{3} $$

Second, outward recursive and calculate velocity and acceleration of each link in link coordinate system and geocentric coordinate system and then calculate inertia force and inertia moment of links by Newton-Euler method. For rotational joint, it just needs to calculate moment of each link ${}^{i}n_i$:

$$ {}^{1}n_1 = {}^{1}_{2}R \begin{bmatrix} 0 \\ 0 \\ m_2 l_{c2} g c_{12} + m_2 l_{c2} l_1 \dot{q}_1^2 s_2 + m_2 l_{c2} l_1 \ddot{q}_1 c_2 \\ + m_2 l_{c2}^2 (\ddot{q}_1 + \ddot{q}_2) + I_2 (\ddot{q}_1 + \ddot{q}_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I_1 \ddot{q}_1 \end{bmatrix} + \begin{bmatrix} l_{c1} \\ 0 \\ 0 \end{bmatrix} \times m_1 \tag{4} $$

$$ \begin{bmatrix} g s_1 - l_{c1} \dot{q}_1^2 \\ g c_1 + l_{c1} \ddot{q}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \times {}^{1}_{2}R m_2 \begin{bmatrix} g s_{12} - l_1 \dot{q}_1^2 c_2 + l_1 \ddot{q}_1 s_2 - l_{c2} (\dot{q}_1 + \dot{q}_2)^2 \\ g c_{12} + l_1 \dot{q}_1^2 s_2 + l_1 \ddot{q}_1 c_2 + l_{c2} (\ddot{q}_1 + \ddot{q}_2) \\ 0 \end{bmatrix} $$

$$ {}^{2}n_2 = \begin{bmatrix} 0 \\ 0 \\ I_2 (\ddot{q}_1 + \ddot{q}_2) \end{bmatrix} + \begin{bmatrix} l_{c2} \\ 0 \\ 0 \end{bmatrix} \times m_2 \begin{bmatrix} g s_{12} - l_1 \dot{q}_1^2 c_2 + l_1 \ddot{q}_1 s_2 - l_{c2} (\dot{q}_1 + \dot{q}_2)^2 \\ g c_{12} + l_1 \dot{q}_1^2 s_2 + l_1 \ddot{q}_1 c_2 + l_{c2} (\ddot{q}_1 + \ddot{q}_2) \\ 0 \end{bmatrix} \tag{5} $$

At last, list ${}^{1}n_1$ and ${}^{2}n^2$ Z components, we get the driving torque:

$$ \begin{cases} \tau_1 = m_2 l_{c2} g c_{12} + (m_1 l_{c1} + m_2 l_1) g c_1 - m_2 l_1 l_{c2} \dot{q}_2^2 s_2 \\ \quad - 2 m_2 l_1 l_{c2} \dot{q}_1 \dot{q}_2 s_2 + m_2 l_{c2} l_1 c_2 (2\ddot{q}_1 + \ddot{q}_2) + m_2 l_{c2}^2 (\ddot{q}_1 + \ddot{q}_2) \\ \quad + (m_1 l_{c1}^2 + m_2 l_1^2 + I_1 + I_2) \ddot{q}_1 + I_2 \ddot{q}_1 \\ \tau_2 = m_2 l_{c2} g c_{12} + m_2 l_{c2} l_1 \dot{q}_1^2 s_2 + m_2 l_{c2} l_1 \ddot{q}_1 c_2 + m_2 l_{c2}^2 (\ddot{q}_1 + \ddot{q}_2) \\ \quad + I_2 (\ddot{q}_1 + \ddot{q}_2) \end{cases} \tag{6} $$

Lagrange method: The total kinetic energy of robot arm is:

$$ \begin{aligned} E_k = & [\frac{1}{2} m_1 l_{c1}^2 + \frac{I_1 + I_2}{2} + \frac{1}{2} m_2 (l_1^2 + l_{c2}^2) + m_2 l_1 l_{c2} c_2] \dot{q}_1^2 \\ & + (\frac{1}{2} m_2 l_{c2}^2 + \frac{1}{2} I_2) \dot{q}_2^2 + (m_2 l_{c2}^2 + m_2 l_1 l_{c2} c_2 + I_2) \dot{q}_1 \dot{q}_2 \end{aligned} \tag{7} $$

Total potential energy is:

$$ E_p = m_1 g l_{c1} s_1 + m_2 g (l_{c2} s_{12} + l_1 s_1) \tag{8} $$

Build Lagrange function L = E_k - E_p and we get system dynamic equation:

$$ \tau = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \tag{9} $$

Calculate partial derivative:

$$ \begin{cases} \dfrac{\partial L}{\partial \dot{q}_1} = (m_1 l_{c1}^2 + I_1 + I_2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2 m_2 l_1 l_{c2} c_2) \dot{q}_1 \\ \quad + (m_2 l_{c2}^2 + m_2 l_1 l_{c2} c_2 + I_2) \dot{q}_2 \\ \dfrac{\partial L}{\partial \dot{q}_2} = (m_2 l_{c2}^2 + I_2) \dot{q}_2 + (m_2 l_{c2}^2 + m_2 l_1 l_{c2} c_2 + I_2) \dot{q}_1 \\ \dfrac{\partial L}{\partial q_1} = -(m_1 l_{c1} + m_2 l_1) g c_1 - m_2 g l_{c2} c_{12} \\ \dfrac{\partial L}{\partial q_2} = -m_2 l_1 l_{c2} s_2 \dot{q}_1^2 - m_2 l_1 l_{c2} s_2 \dot{q}_1 \dot{q}_2 - m_2 l_{c2} g c_{12} \end{cases} \tag{10} $$

Substitute Eq. (10) into (9) and we get dynamic equation of robot arm as follow:

$$ \begin{cases} \tau_1 = (m_1 l_{c1}^2 + I_1 + I_2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2 m_2 l_1 l_{c2} c_2) \ddot{q}_1 \\ \quad - 2 m_2 l_1 l_{c2} s_2 \dot{q}_1 \dot{q}_2 + (m_2 l_{c2}^2 + m_2 l_1 l_{c2} c_2 + I_2) \ddot{q}_2 \\ \quad - m_2 l_1 l_{c2} s_2 \dot{q}_2^2 + (m_1 l_{c1} + m_2 l_1) g c_1 + m_2 g l_{c2} c_{12} \\ \tau_2 = (m_2 l_{c2}^2 + I_2) \ddot{q}_2 + (m_2 l_{c2}^2 + m_2 l_1 l_{c2} c_2 + I_2) \ddot{q}_1 \\ \quad + m_2 l_1 l_{c2} s_2 \dot{q}_1^2 + m_2 l_{c2} g c_{12} \end{cases} \tag{11} $$

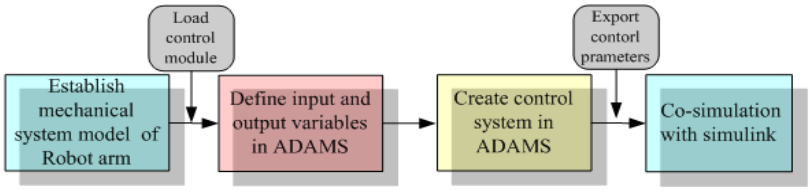


Fig. 2: Co-simulation flow chart of ADAMS and MATLAB

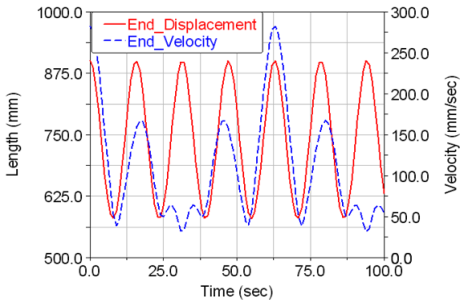


Fig. 3: Displacement curve and velocity curve on test point of robot arm end effectors

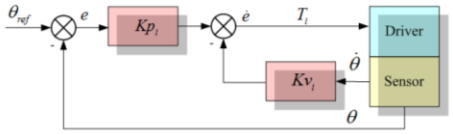


Fig. 4: Control system block diagram of robot arm

Through verification, results from two methods are consistent and we know the whole derivation process is correct.

ESTABLISHMENT OF ROBOT ARM CO-SIMULATION MODEL

Co-simulation with ADAMS and MATLAB /SIMULINK means that build multi-body system in ADAMS, output parameters related to system equation and then import information from ADAMS into MATLAB/SIMULINK and set up control scheme (Zhu *et al*., 2010). During calculation process, there exchanges data between virtual prototype and control program, where, ADAMS solves the mechanical system equation and MATLAB solves the control system equation. They both finish the whole control process. The flow chart of co-simulation displays in Fig. 2.

Establishment of robot arm physical system model: Firstly, build 3D model of two DOF robot arm in SOLIDWORKS; secondly, simplify and save it as *.x_t file; finally, import this file into ADAMS (Zhang *et al*., 2009). For giving the virtual prototype attributes like material, mass and inertia moment real machine has, we need edit each component to better simulate real system (Liu *et al*., 2009).

Add fixed joint on the base of robot arm, define rotational joint and set gravity (Li *et al*., 2007a). So far, there exists constrain relationship between links of robot arm and now physical system model is set up.

To verify correctness and effectiveness, here exerts sine moment function:

$$\begin{cases} Motion1 = 80d \cdot \sin(0.1 \cdot time) \\ Motion2 = 100d \cdot \sin(0.2 \cdot time) \end{cases} \quad (12)$$

With this function driving, set simulation time as 100 s and simulation step as 200 and then simulate. We measure displacement curve and velocity curve on centre of mass of robot arm end effector in postprocessor in ADAMS (Fig. 3).

Establishment of robot arm control system model: Proportional differential system is adopted, with dynamic model as the theoretical basis and error driven style. e represents error between ideal angle and real angle and $\dot{e}$ is error between ideal velocity and real velocity (Fig. 4).

Regardless of dynamic properties, we simplify Multiple Input Multiple Output system (MIMO) as a Single Input Single Output servo-control system (SISO). Without considering inductance, AC servo system is a stable second order system (Ma *et al*., 2010). To accelerate system response, we add proportional element; and to enlarge damp and reduce overshoot, we add differential element. And then we get PD position control. If ignore dynamic properties of joint driver, driven torque is:

$$T_i = Kp_i(\theta_{ref} - \theta) - Kv_i\dot{\theta} \quad (13)$$

where,

θ, $\dot{\theta}$ = Respectively feedback signal of location and speed of joint

Kp_i. Kv_i = Respectively related coefficient

When conduct position control for robot arm with PD control system, for reducing influence from external disturbance, greater gain should be used unless the system is unstable.

Build the interface between two software: The model built in ADAMS, as a sub-system, need to be imported into MATLAB/SIMULINK, on which SIMULINK constructs the co-simulation system. First, exchange data between ADAMS and MATLAB/SIMULINK through ADAMS/CONTROL interface. Second, define

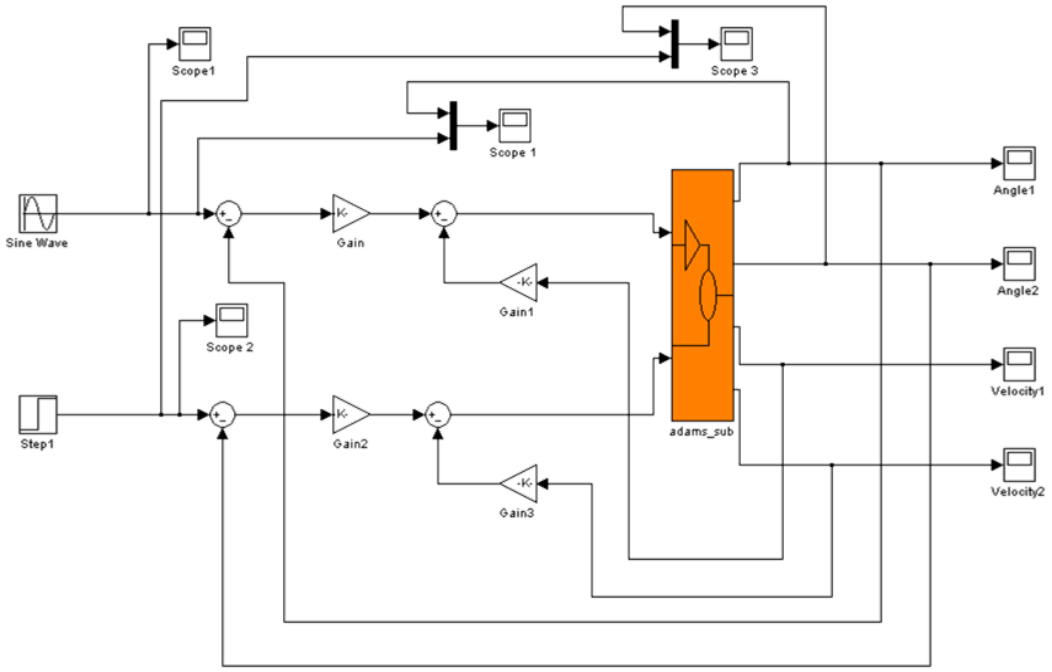


Fig. 5: Control structure chart of co-simulation

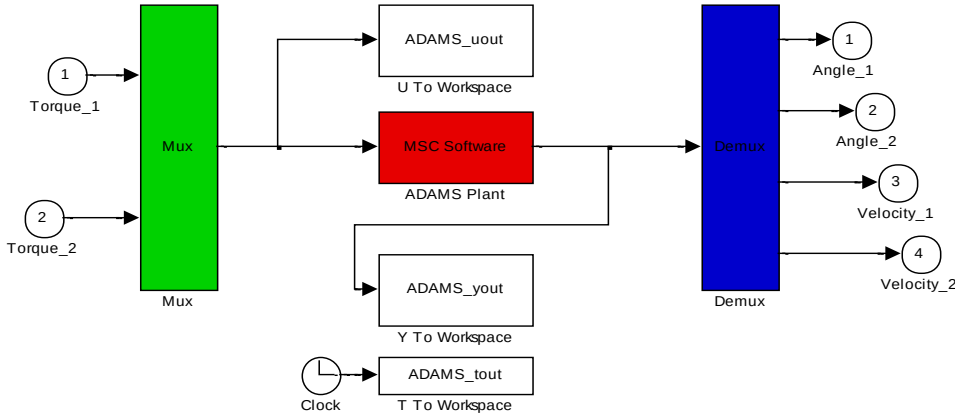
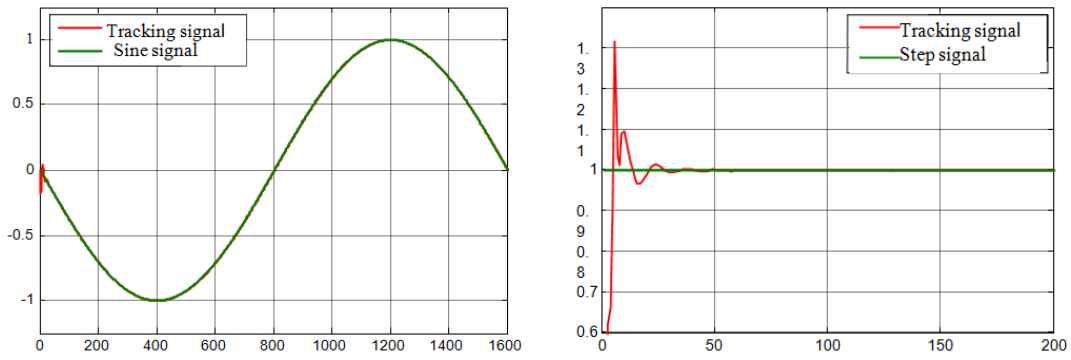


Fig. 6: Mechanical sub-system module



(a) (b)

Fig. 7: (a) Response of sine signal on joint1, (b) response of step signal on joint 2

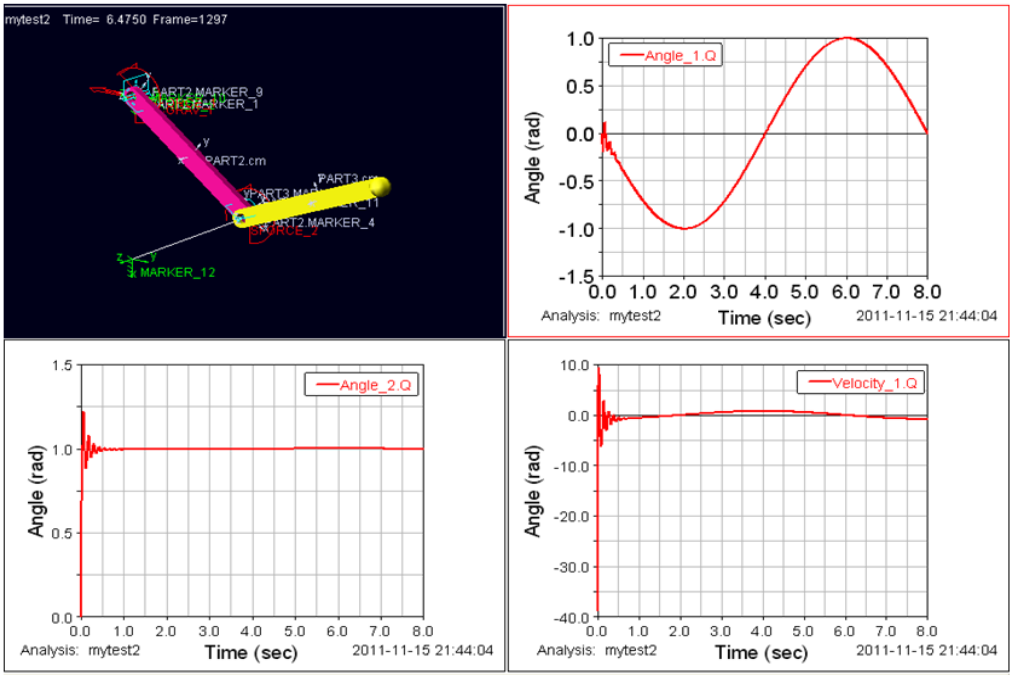


Fig. 8: Results in ADAMS post processor

6 system variables which is needed in co-simulation such as input variable Torque_1, Torque_2, output variable Angle_1, Angle_2, Velocity_1 and Velocity_2. In "Controls" menu, we can select and set those variables.

Special functions in ADAMS are real-timely called to output torque variables, which are regarded as instruction code to drive the joints of robot arm. Every joint variable returns to control system in time and completed closed loop comes up (Ye, 2010). The structure chart of co-simulation on robot arm is (Fig. 5).

After defining the variables, three files (.m, .cmd and .adm file) generate from ADMAS/CONTROL, which is useful in data-exchange between ADAMS and MATLAB (Yi *et al*., 2009). Then input command "adams_sys" into MATLAB, mechanical sub-system generates (Fig. 6).

CO-SIMULATION RESULTS OF ROBOT ARM

After that, it is time to conduct the co-simulation. This is a interactive closed process, in which simulation is real-time. During the simulation, real-time interactive process is shown on the ADAMS interface (Li *et al*., 2007b). At last, result curves generate from "Scope" module. Figure 7 describes the results.
Here are the co-simulation results:

- Joint 1 can follow the sine signal well with smaller time delay. When driven by sine signal, link 1 flaps up and down according to the sine law around the axis of the base.
- Under step signal, joint 2 follows the instruction shortly. And then link 2 reaches to horizontal position and keeps still.
- From simulation, we know that links track the given trajectory curve well with small error.

Loading .res file into ADAMS, we can measure inertia force, gravity, centrifugal force, coriolis force and instantaneous power of the driver. To know how they change according to time, we research the dynamics of robot arm better (Li *et al*., 2010). Response curve from ADMAS postprocessor module is shown in Fig. 8.

CONCLUSION

In this study, build co-simulation model of two DOF robot arm respectively in ADAMS and MATLAB, design the interface between mechanical system and control system and get the response performance of two links of robot arm. The system has good tracking ability and co-simulation result verifies that. Also, its dynamic performance meets the design requirement.

What's more, modeling in ADAMS, it is beneficial to avoid solving dynamic equations and to observe motion process visualized. Using MATLAB /SIMULINK toolbox, we operate the whole process simply and program efficiently and quickly. And we get lots of design parameters from co-simulation, which is useful for subsequent research.

Co-simulated method takes advantages of two software, with enhancing dynamic performance of robot arm, improving efficiency, reducing the cost and saving time. For the complex control system, it is a good solution selected.

REFERENCES

Gu, M.Y., R.R. Qin and D.Y. Yang, 2005. Co-simulated research method of robot arm dynamics in ADAMS and MATLAB. Mach. Des., 1: 227-228.

Li, S.Q. and H. Le, 2011. Co-simulation study of vehicle ESP system based on ADAMS and MATLAB. J. Softw., 6(5): 866-950.

Li, H., B.H. Fan and Q. Liu, 2007a. Research on co-simulation based on ADAMS and MATLAB. Sci. Technol. Inform., 8: 152.

Li, S.H., S.P. Yang and H.Y. Li, 2007b. Semi-active control co-simulation of automotive suspension based on ADAMS and MATLAB. J. Syst. Simul., 19(10): 2304-2307.

Li, B.M., Z.B. Qian and H.J. Cheng, 2010. Research on co-simulation of AUV engine based on ADAMS/MATLAB. J. Syst. Simul., 22(7): 1668-1673.

Liu, G.J., M. Wang and B. He, 2009. Co-simulation of autonomous underwater vehicle based on ADAMS and MATLAB/SIMULINK. Chinese J. Mech. Eng., 45(10): 22-29.

Ma, R.Q., S.H. Hao and W.F. Zheng, 2010. Co-simulation of robot arm based on ADAMS and MATLAB. Mach. Des. Manufact., 2(2): 93-95.

Soon, J.C. and J.J.E. Slotine, 2009. Cooperative robot control and concurrent synchronization of Lagrangian systems. IEEE T. Robot., 25(3): 686-700.

Xing, J.W., 2012. Getting Started Using ADAMS/Controls. MSC ADAMS Training Manual.

Ye, H.P., 2010. Application of co-simulation based on ADAMS/MATLAB in mechtronics. J. Zhangzhou Tech. Instit., 12(1): 4-6.

Yi, X.S., Z.G. Huang and M.T. Sun, 2009. Application of co-simulation technology based on ADAMS and MATLAB. J. Beijing Univ. Comm., 27(5): 14-17.

Zhang, X.Y., B.B. Sun and Q.H. Sun, 2009. Vehicel and terrain interaction based on ADAMS-MATLAB co-simulation. J. Southeastern Univ., 25(3): 335-339.

Zhu, D.L., J.Y. Qin and Y. Zhang, 2010. Research on co-simulation using ADAMS and MATLAB for active vibration isolation system. IEEE ICICTA, 2: 1126-1129.

Zong, X.Y. and Y.Y. Li, 2005. Control simulation of robot arm based on ADAMS and MATLAB. Microcomput. Informat., 35: 29-30.