

Realistic Modeling and Dynamic Simulation of KUKA KR5 Robot using RecurDyn

Shivesh Kumar

Department of Mechanical Engineering
National Institute of Technology
Karnataka Surathkal
Surathkal, Karnataka-575025, India
Email: shivesh.mecheng@gmail.com

Rajeevlochana C.G.

Department of Mechanical Engineering
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi, India
Email: rajeevlochan.iitd@gmail.com

Subir Kumar Saha

Department of Mechanical Engineering
Indian Institute of Technology Delhi
Hauz Khas, New Delhi, India
Email: saha@mech.iitd.ac.in

Abstract

The design of an industrial robot often involves a multi-body mechanical system driven by a microprocessor based sophisticated control system. With the advent of various dynamic simulation packages (Ex- ADAMS, RecurDyn, Autodesk Inventor etc.), one can easily perform dynamic simulation of complex mechanical systems assuming an oversimplified control scheme. Similarly, with various control system simulation packages (Ex- MATLAB/Simulink, LabVIEW etc.), one can perform the task of control system simulation but without a realistic model of the plant. For performing dynamic co-simulation of any robot, RecurDyn and Simulink can be connected such that RecurDyn provides the actual plant model to Simulink for its control system simulation. This strategy saves a lot of time as one need not derive the complex kinematics and dynamics equations of the robot to prepare its mathematical model in MATLAB/Simulink. However, with the introduction of CoLink toolkit in RecurDyn, the whole task of dynamic and control system simulation can be performed independently by RecurDyn. In this paper, we present realistic modeling and dynamic simulation of a KUKA KR5 robotic manipulator with the help of RecurDyn software. CAD assembly of the robot is first imported in the RecurDyn environment. Then, different mechanical joints, solid contacts, plant inputs and plant outputs are defined and RecurDyn is connected to the CoLink interface. Realistic control system is modeled in CoLink to provide cycloidal trajectories with the help of available CoLink blocks. These are fed as input to each joint and dynamic simulation is performed. Using the plot tool in RecurDyn, time dependent driving torque variation at joint 1 is plotted against time (See Figure 2). From these curves, maximum torque required at each joint can be used for selection of actuators, gear box and other purposes. This will also help students in learning and understanding robot kinematics and dynamics concepts. In future, motor model in CoLink and exact dynamic parameters of the robot and joint friction in RecurDyn will be implemented, with control feedback loop so as to get a realistic co-simulation of control system and dynamics.



Fig. 1 KUKA KR5 Assembly

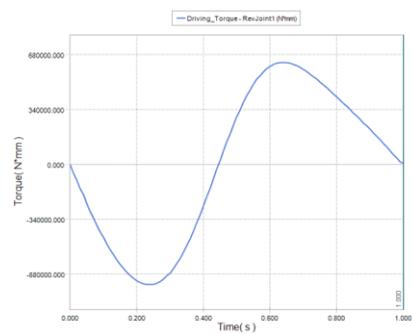


Fig. 2 Torque v/s time (Joint1)

Realistic Modeling and Dynamic Simulation of KUKA KR5 Robot using RecurDyn

Shivesh Kumar

Department of Mechanical Engineering
National Institute of Technology Karnataka
Surathkal, Karnataka-575025, India
Email: shivesh.mecheng@gmail.com

Rajeevlochana C.G.

Department of Mechanical Engineering
Indian Institute of Technology Delhi
Hauz Khas, New Delhi-110016, India
Email: rajeevlochan.iitd@gmail.com

Subir Kumar Saha

Department of Mechanical Engineering
Indian Institute of Technology Delhi
Hauz Khas, New Delhi-110016, India
Email: saha@mech.iitd.ac.in

ABSTRACT

The advent of highly efficient computer simulation tools has brought about a paradigm shift in any product design and development process. The growing use of computer simulation in the robot design process is not an exception. Since, the design of a robot generally involves a multi-body mechanical system driven by microprocessor based sophisticated control system, a wide variety of simulation software (e.g., ADAMS, MATLAB/Simulink, LabVIEW etc.) are required for the simulation of several aspects of the design of a robot. In that respect, RecurDyn is a suitable multi-body dynamics software which has the capabilities of simulating a control system along with the mechanical system using its CoLink toolkit. In this paper, we present a method for realistic modeling and dynamic simulation of KUKA KR5 industrial robot using RecurDyn. The control algorithm is modeled in CoLink and is integrated with RecurDyn to carry out the dynamic co-simulation of the robot. The plots showing end-effector coordinates and driving torques with time were obtained which can give useful information in optimizing the robot design. Also, it can help students in learning and understanding robot's kinematics and dynamics in an enhanced way.

1 INTRODUCTION

An industrial robot is a spatial multi-body system with many degrees of freedom and hence it is essentially a multi-variable, multi-parameter coupled non-linear system. Hence, their kinematic and dynamic analyses are complex and difficult to solve. Also, it is very difficult for beginners to learn and visualize how these manipulators will perform under certain dynamic conditions.

Computer Simulation has been recognized as an important research tool since the beginning of the 20th century as reported by Zlajpah (2010). It was an academic research tool in the beginning but now it has evolved as a

powerful tool supporting the design, planning, analysis, and decisions in different areas of research and development. It has been used by various researchers, developers and manufacturers in a variety of fields. The use of computer simulation in the study of robotics is not an exception. For complex systems as robots, the simulation tools can certainly enhance the design, development, and even the operation of the robotic systems. Augmenting the simulation with visualization tools and interfaces, one can simulate the operation of the robotic systems in a very realistic way.

With the advent of various dynamic simulation packages (e.g., ADAMS, RecurDyn, Autodesk Inventor, CATIA etc.), one can easily perform dynamic simulation of complex mechanical systems like an industrial robot assuming an oversimplified control scheme, as demonstrated by Doan (2008) and Alshamasin (2009). Similarly, with various control simulation packages (e.g., MATLAB/Simulink, LabVIEW etc.), one can simulate the task of a controller but without a realistic model of the plant, as shown by Rastogi (2011), Meng (1995) and Schlotter (2003). For example, while performing the control simulation of a robot using MATLAB, one generally needs to prepare the mathematical model of the robot first which requires the derivation of complex kinematics and dynamics equations. This makes the process very time consuming. MATLAB/SimMechanics can however aid this process.

RecurDyn, as a multi-body dynamics analysis software, can overcome some of the shortcomings in other traditional dynamics analysis software such as excessive simplification, low solving efficiency and bad solving stability. The solver of RecurDyn consists of equations of motion theory with recursive formulation, which makes RecurDyn to be a fast and high precision method. RecurDyn also supports almost all types of file created by different CAD software. Furthermore, it can be interfaced

with MATLAB/Simulink, which facilitates co-simulation of mechanical and control system [Wang (2011) and Hwang (2007)]. Hence using these two software, dynamic co-simulation of robots can easily be performed. For performing dynamic co-simulation of any robot, RecurDyn and Simulink can be connected such that RecurDyn provides the actual plant model to Simulink to perform its control system simulation. This strategy saves a lot of time as one need not derive the complex kinematics and dynamics equations of the robot to prepare its mathematical model in Matlab/Simulink. However, with the recent introduction of CoLink toolkit in RecurDyn, the whole task of dynamic simulation can be performed independently by RecurDyn. Thus, the use of this toolkit eliminates the use of Matlab/Simulink from this process.

In this paper, we present a potential application of this toolkit for realistic modeling and dynamic co-simulation of a robot using RecurDyn. Time dependent variations of several parameters such as joint torques, velocity, acceleration can be easily obtained from the software. Also, the actual robot motion can be visualized. We demonstrate this by taking the example of KUKA KR5 [KUKA Robotics Corp., Germany (2011)] robot which is an industrial robotic manipulator with 6 degrees-of-freedom (DOF).

2 ROBOT MODELING IN RECURDYN

KUKA KR5 is a 6-DOF industrial serial robotic manipulator. The precise CAD model of the KUKA KR5 robot was downloaded from the website of KUKA Robotics, (2011). The CAD model of the full robot was imported into RecurDyn using STEP file format. Since, RecurDyn does not support constraint based assembly environment, it is always advised to import the full assembly in STEP format instead of importing them part by part and then assembling them with Object Control feature in RecurDyn. As reported in ISO 10303 (2006), STEP stands for “International Standard for Product Data Exchange” and is a standard format for the exchange of solid models across different CAD software. The STEP file does not contain any information about the kinematic constraints present in the robot assembly. Hence, mechanical joints in the form of kinematic constraints were in RecurDyn. Since, the robot at hand was 6-DOF robot; six revolute joints were defined in the assembly. Figure 1 shows the completed model of KUKA KR5 robot with all the joints defined.

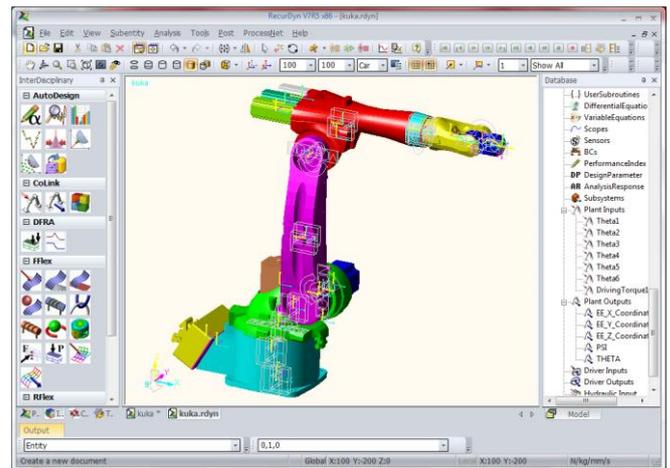


Figure 1. KUKA KR5 robot model with revolute joints

3 DYNAMIC CO-SIMULATION

Colink is a toolkit provided with RecurDyn software which can help in control system design and simulation. Moreover, it can be coupled with a mechanical system defined in RecurDyn (also called RecurDyn Plant) to perform dynamic co-simulations. A co-simulation means a simulation performed by associating or linking two different software. Wang (2011) and Hwang (2007) have shown the dynamic co-simulations of different robots by linking RecurDyn and Matlab/Simulink software. However, using Colink toolkit in RecurDyn, dynamic co-simulation of any mechanical system can be performed solely using RecurDyn. Such simulations will save time involved in learning two different software and costs incurred due to the purchase of two different software licenses. Since mechanical system modeled in RecurDyn environment needs to communicate with the Colink toolkit, various inputs and outputs of the mechanical system (RecurDyn Plant) were defined. Similarly, there can be different inputs

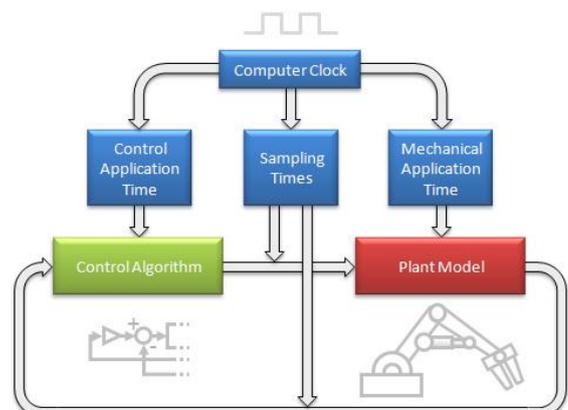


Figure 2. Schematic diagram showing simulation in RecurDyn (Motion Port (2009))

and outputs for any control algorithm which must be defined in the Colink environment to perform any co-simulation. Also, sampling time, control application time and mechanical application time can be set according to the user's requirement. Figure 2 shows a schematic block diagram of simulation approach implemented in RecurDyn, as reported in Motion Port (2009).

3.1 INVERSE DYNAMICS

A very simple but crude method of a robot control can be an open-loop independent joint control, as reported by Craig (2002). In this case, the robot control depends upon the independent angular positions of its joints. Hence, six angular position inputs for the six joints were defined. These inputs to the mechanical system are also termed as Plant Inputs. In open loop control system there is no feedback. As a result, no need of defining various plant outputs for this model. However, to have the plots of end-effector coordinate variations with time, plant outputs of this model are defined as (X, Y, Z) coordinates of the end-effector.

Now to perform the dynamic co-simulation of the robot, joint trajectories for each revolute joint in the robot are to be defined. Cycloidal trajectory was chosen as input to each joint as the velocity and acceleration are zero at the starting and end points [Saha (2008)]. The cycloidal motion trajectory i.e., joint angle, velocity and acceleration, can be mathematically defined by the following equations:

$$\theta(t) = \theta(0) + \frac{\theta(T) - \theta(0)}{T} \left[t - \frac{T}{2\pi} \sin\left(\frac{2\pi}{T} t\right) \right] \quad (1)$$

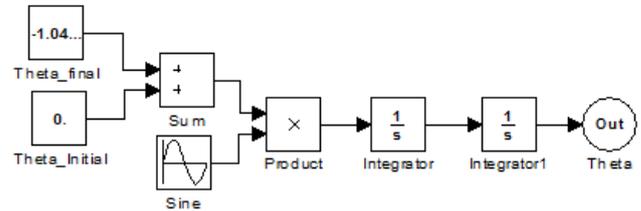
$$\dot{\theta}(t) = \frac{\theta(T) - \theta(0)}{T} \left[1 - \cos\left(\frac{2\pi}{T} t\right) \right] \quad (2)$$

$$\ddot{\theta}(t) = \frac{\theta(T) - \theta(0)}{T} \left[\frac{2\pi}{T} \sin\left(\frac{2\pi}{T} t\right) \right] \quad (3)$$

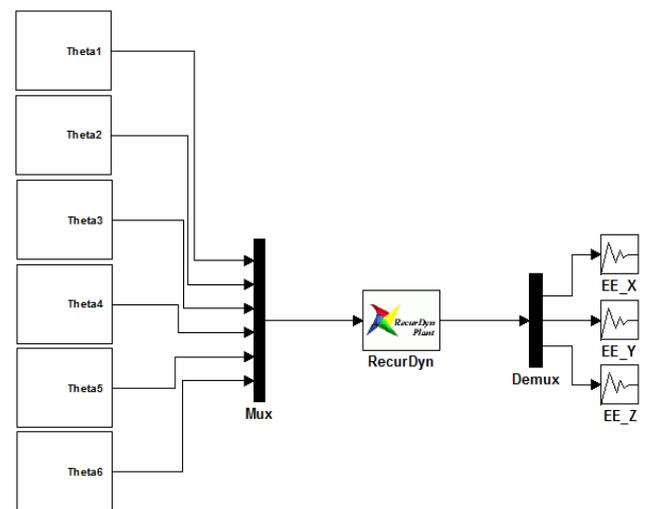
where t is the time elapsed and T is the total time of simulation.

Colink provides graphical programming environment in which one can use readily available blocks for writing the program. Comparing equations (1), (2), and (3), equation (3) is in the simplest mathematical form. Hence, this expression was used as a function for generating the trajectory input to each joint in the form of joint acceleration. For a simulation time of 1 second, a sinusoidal function generator block with frequency and amplitude of 2π was used. The acceleration was then integrated twice with initial and final conditions imposed as boundary values to get the angular position. Various Colink blocks (e.g., sum, product, sine, integration etc.) were used.

for this purpose. Figure 3 (a) shows the subsystem defined in Colink to generate cycloidal trajectory for desired initial and final values of the angular position.



(a) Subsystem for generating cycloidal trajectories



(b) Overall flow of control

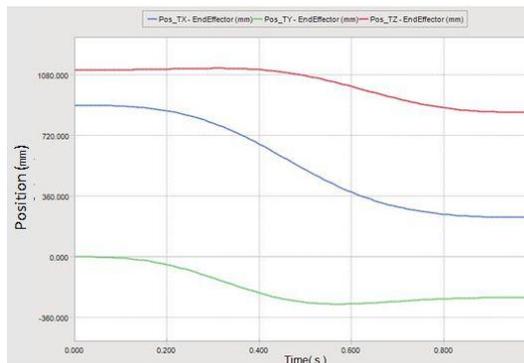
Figure 3. Graphical programming environment in Colink

The angular position signal generated using equations of cycloidal trajectory for each joint were then multiplexed and were provided as input to the RecurDyn plant block in the Colink toolkit. While defining plant outputs, a DeMux block was used with three scope blocks to see the variation of end-effector's coordinates with time. The overall flow of control of the system is shown in the Figure 3 (b).

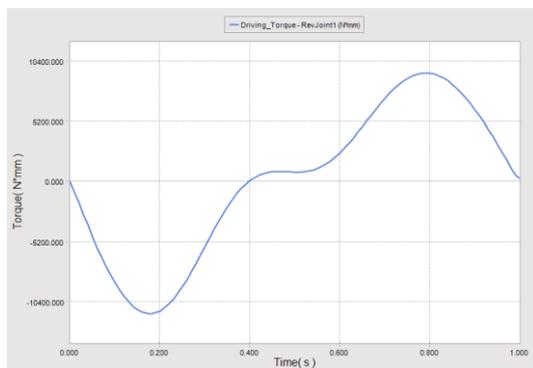
To make the simulation more realistic, gravity ($g=9.81\text{m/s}^2$) and friction ($\mu_{\text{static}}=0.5$ and $\mu_{\text{dynamic}}=0.3$) at each joint were also introduced in RecurDyn environment. For serial manipulators, RecurDyn uses Newton-Euler formulations to compute the driving torques for each input joint trajectory, as reported by Wang (2011). The formulations used in RecurDyn also take into account the joint friction and effect of gravity. The dynamic co-simulation of the robot was performed for a simulation time of 1 second divided into 100 equal time steps. The accuracy

of the solution can be easily varied by varying the number of time steps for a given simulation time.

Using the plot tool in RecurDyn, the time dependent coordinate variation of the end-effector (i.e., X, Y and Z v/s time) in spatial coordinate system was plotted. They are shown in Figure 4(a). Driving torque in the first joint is also plotted in Figure 4(b). Similar plots for other joints can be obtained. From such driving torque v/s time plots, maximum torque required at each joint can be easily predicted and this information can be used for selection of actuators, gear box and other purposes.



(a) End-effector position with time



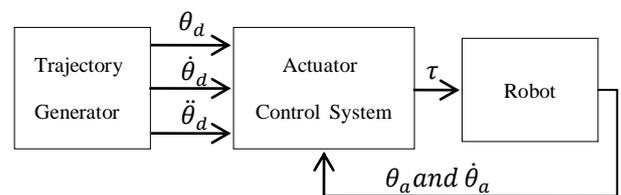
(b) Driving torque with time for Joint 1

Figure 4. Graph plots in RecurDyn

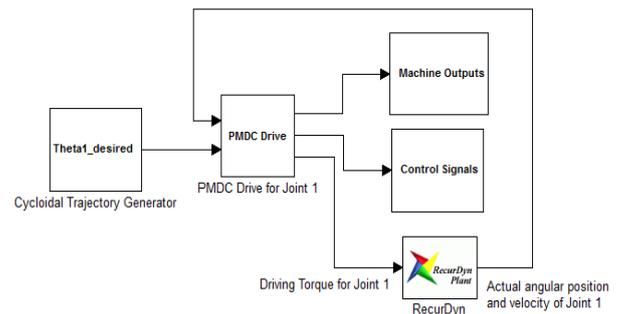
3.2 CLOSED-LOOP JOINT CONTROL

The discussion in section 3.1 can help anyone in finding out the right actuator for any joint as it gives information about the torque requirement at each joint for performing a specific motion. However, as a real actuator is introduced in the system, it becomes more complex since there are various non-linearities associated with the actuator itself. For example, back EMF, friction at the motor bearings, etc. To take care of each and every possible cause of these inevitable disturbances, a control system is highly essential for any robotic system. Generally, the only

way to make a highly efficient control system is to make use of feedback from the joint sensors in the form of joint angle (θ_a) and joint velocity ($\dot{\theta}_a$). Typically, this feedback is required to compute the servo error by finding out the differences between the desired position and actual position, and likewise between the desired joint velocity and actual joint velocity. The control system can then find out the torques required at the joints as some function of servo errors. Obviously, the basic idea is to compute the actuator torques which can reduce these servo errors. Such a control system can be employed at every joint independently. Hence, it is called a closed-loop independent joint control system [Craig (2002)]. Figure 5(a) shows a block diagram for closed-loop independent joint control system.



(a) Block diagram for closed-loop independent joint control system.



(b) Block diagram for Joint 1 implemented in Colink

Figure 5. Closed loop independent joint control

Note that PMDC (Permanent Magnet DC) electric drive was chosen as the actuator control system as PMDC motors are widely adopted as actuators in robotic systems. Also, PMDC electric drive block was readily available in Colink library. Now, there are two different systems, robot as RecurDyn Plant and actuator as PMDC drive, each having their own sets of inputs and outputs and interacting with each other. The main input to the PMDC drive is the output from the cycloidal trajectory generator and the second input is the actual values of the angular position and velocity of Joint 1 fed as feedback to the PMDC drive control system from the robot. On the other hand, the

RecurDyn plant (or robot) has torque as input and angular position and angular velocity as its outputs. For simplicity of explanation and to avoid redundancy, control system for the Joint 1 of the robot was demonstrated. The position of all other joints except Joint 1 was locked. Since, the control system is independent for each of the joints; it can be extended to any number of joints. Figure 5(b) shows the block diagram for the closed-loop control system for Joint 1 implemented in Colink environment.

The mathematical model of the PMDC motor used in RecurDyn is explained below. The voltage equation used to simulate a PMDC drive is given by

$$V_a = R_a I_a + L_a \frac{dI_a}{dt} + E \quad (4)$$

where V_a is the input voltage to the motor, R_a is the armature resistance, L_a is the armature inductance, I_a is the armature current, and E is the back electromotive force (BEMF).

The BEMF of the motor is directly proportional to the mechanical angular speed of the rotor as represented by

$$E = K_e \omega_m \quad (5)$$

where K_e is the BEMF constant and ω_m is the net mechanical angular speed.

The electrical torque (T_e), which is developed by the PMDC motor is proportional to the armature current I_a , i.e.,

$$T_e = K_t I_a = K_e I_a \quad (6)$$

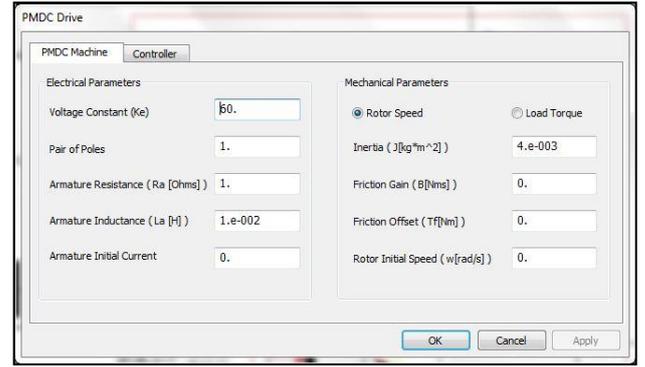
where K_t is the torque constant and is generally equal to K_e .

Since there is always a reduction in torque delivered to the joint due to friction in motor bearings, the net torque delivered to the joint can be determined as

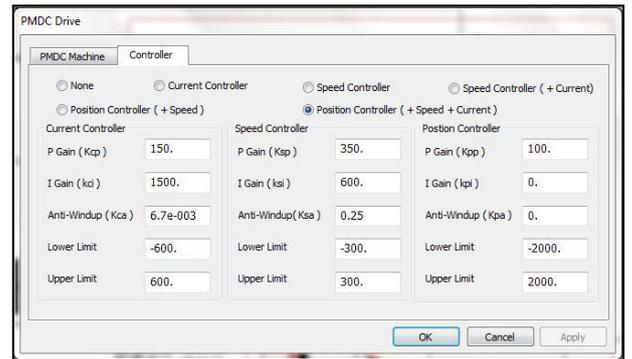
$$T_d = T_e - \text{sgn}(\omega_m)(B|\omega_m| + T_f) \quad (7)$$

where T_d is the driving torque or net torque delivered, B is the viscous friction gain and T_f is the viscous friction offset.

The various electrical and mechanical parameters for the PMDC actuator can be set in the Colink environment, as shown in Figure 6 (a). The controller parameters involving the values of different gains can also be set in the Colink environment, as shown in Figure 6 (b).



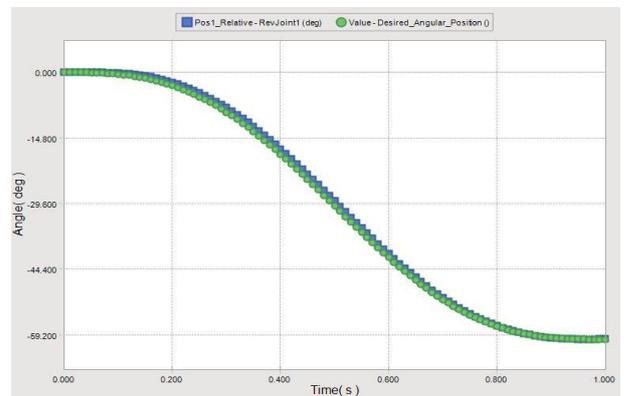
(a) Mechanical and electrical properties



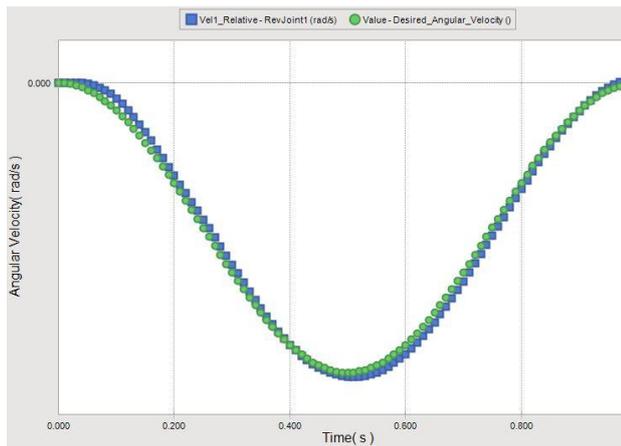
(b) Controller parameters

Figure 6. Interface to set parameters of the PMDC drive

The dynamic co-simulation of the robot with closed-loop control for Joint 1 with all other joints locked was carried out for the simulation time of 1 second divided into 100 equal time steps. Using the plot tool in RecurDyn, time dependent variation of angular position and angular velocity are plotted. Since appropriate values of gains were chosen for the simulation, it can be observed in Figure 7(a) that the desired and actual angular positions are almost matching. Moreover, desired and actual angular velocities also approach each other as evident from Figure 7(b).



(a) Desired and actual angular position



(b) Desired and actual velocity

Figure 7. Graph plots of Joint 1

Similarly, some other important plots like armature current variation with time, voltage variation with time, generated electrical torque with time can be easily plotted using the plot tools. Hence, this simulation approach can be used for tuning our controller gains. This will also help in actuator design as one can get very precise values of various motor properties like armature inductance, resistance, BEMF constant etc. from the simulation.

4 CONCLUSIONS

Two types of simulations were presented in this paper. In the first simulation, inverse dynamics of a robot under certain kinematic and dynamic constraints was performed without using any actual actuator model. Such simulations can help us finding the torque requirements at each joint of the robot. The second simulation was more realistic where a robot with a closed-loop control system with an actual actuator model were considered. Such simulations can help in actuator design and finding the optimal gain values for the controller. Closed-loop independent joint control for KUKA KR5 robot was demonstrated only for its first joint to avoid the redundancy and for the simplicity of explanation. However, it can be extended to any number of joints based on the system's requirement. PMDC motor was used as an actuator model in the simulation as PMDC motor block was readily available in the Colink library. In reality, 3 phase AC servo motors are used in KUKA KR5 robot instead of PMDC motors. In Colink, such motor models can also be prepared using Ch Script language for RecurDyn. This feature provides even greater flexibility to the users as one can make custom blocks and use them in RecurDyn for their simulation.

In future, simulation of KUKA KR5 robot will be carried out using 3 phase AC servo motors and the results will be compared to the actual robot specifications. Also,

more complex control schemes will be modeled in RecurDyn and will be tested.

ACKNOWLEDGMENTS

The authors are highly indebted to Mr. B Sridhar, Director, Function Dynamics India Pvt. Ltd. and FunctionBay Pvt. Ltd., Korea for providing the extended trial license of the software for this project and financial support to the first two authors to present this work in this conference.

REFERENCES

- [1] Alshamasin, M.S.; Ionescu, F.; Al-Kasasbeh, R.T.: Kinematic Modeling and Simulation of a SCARA Robot by Using Solid Dynamics and Verification by MATLAB/Simulink, European Journal of Scientific Research, Vol. 37 No. 3, pp. 388-405, 2009.
- [2] Craig John J., Introduction to Robotics: Mechanics and Control, Addison-Wesley Publishers, 2002.
- [3] KUKA Robotics website (<http://www.kuka-robotics.com/usa/en/downloads/>), accessed in October, 2011.
- [4] Leon Zlajpah: Robot Simulation for Control Design, Robot Manipulators Trends and Development, 2010.
- [5] Michael Schlotter : Multibody System Simulation with SimMechanics, Technical Report, Darmstadt University of Technology, 2003.
- [6] Motion Port, White paper on Simulating Computer Controlled Multi-body dynamic systems, 2009.
- [7] Oh, K.; Hwang, J. P.; Kim, E. and Lee H.: Path Planning of a Robot Manipulator using Retrieval RRT Strategy, World Academy of Science, Engineering and Technology 25, pp. 171-174, 2007.
- [8] Rastogi, A.; Guruprasad K.R.: Kinematic Analysis of 5DOF manipulator arm for mine detection, 7th International Conference on Trends in Industrial Measurements and Automation. 2011.
- [9] Saha S. K., Introduction to Robotics, New Delhi: Tata McGraw Hill, 2008.
- [10] STEP Application Handbook ISO 10303, Version 3 prepared by SCRA, 2006.
- [11] Sun, S. S. ; Meng, Q. H. M. : Dynamic Simulation of Robot Manipulators Using Graphical Programming Packages, Proceedings of IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing, pp: 521-524, 1995.
- [12] The Thao Doan ; Ill Soo Kim ; Hak Hyoung Kim ; Jae Won Jeong ; Bong Yong Kang: Developed

Simulation Model-Kinematics for Robotic Arc Welding. Asian International Journal of Science and Technology in Production and Manufacturing, Vol. 1, No. 2, pp. 69-76, 2008.

- [13] Yan-Shen Wang : Dynamics co-simulation of a type of spot welding robot by RecurDyn and Simulink, International Conference on Consumer Electronics, Communications and Networks, 2011.