
AC 2011-714: DEVELOPMENT OF A ROBOTIC PLATFORM FOR TEACHING MODEL-BASED DESIGN TECHNIQUES IN DYNAMICS AND CONTROL PROGRAM

Bingen Yang, University of Southern California

Dr. Bingen Yang is Professor of Aerospace and Mechanical Engineering, who has taught courses (including AME 301, 420 and 451) and conducted research in the area of dynamics and control at USC for 21 years. He has expertise in modeling, analysis, and simulation of dynamic systems and structures. He has developed more than 300,000 lines of MATLAB codes for simulation of dynamic and control systems, which have been used in some courses taught by him. Dr. Yang is the author of the book "Stress, Strain, and Structural Dynamics: An Interactive Handbook of Formulas, Solutions, and MATLAB Toolboxes".

Cheng-Yuan Jerry Chen, USC

Dr. ChengYuan Jerry Chen is fulltime Lecturer of Aerospace and Mechanical Engineering, who has involved with AME laboratory teaching for more than 8 years in AME341, 441, and 443 classes. His expertise is not only in analytical and computational of dynamic and control systems, but also in experimental and laboratory hardware implementations. He has more than 20 years of advanced machining experience and has accomplished enormous projects in mechanical and electrical designs. He is currently the head leader of the instructional laboratory in the Aerospace and Mechanical Engineering Department.

Development of a Robotic Platform for Teaching Model-Based Design Techniques in Dynamics and Control Program

Abstract

This paper describes an on-going project of undergraduate curriculum innovation in our department, which is sponsored by Mathworks Inc. and our engineering school. The main purpose of the project is to develop a FANUC robotic platform, by which we shall significantly modify two existing undergraduate laboratory courses in dynamics and control: Senior Projects Laboratory (100-110 students per year), and Control Systems Laboratory (60-70 students per year). The platform will be integrated with Mathworks software, which will allow our undergraduate students to learn state-of-the art Model-Based Design (MBD) techniques.

1. Introduction

Design of complex dynamic systems requires the development of mathematical models with varying complexities, extensive simulation studies for validation of the proposed models, synthesis and analysis of control algorithms, verification of the performance the closed-loop systems via numerical simulation, and “hardware in the loop” simulations. As a modern industry standard, Model-Based Design (MBD) techniques allow for relatively inexpensive design iterations by manipulating parameters of the simulation model instead of costly and time consuming direct experimentation on hardware set-ups. Thus, MBD provides the designer with the ultimate playground for rapid testing of ideas and for investigating “what if” scenarios on their desktop or in the laboratory, so as thoroughly exploring the entire design space. It is therefore imperative to make MBD the central philosophy for teaching courses in the area of dynamics and control in mechanical engineering education.

This paper introduces an on-going project of undergraduate curriculum innovation in our department, which is sponsored by Mathworks Inc. and our engineering school. The objective of the project is to develop a FANUC robotic platform, by which we shall significantly modify two existing undergraduate laboratory courses in dynamics and control: AME 441 Senior Projects Laboratory (100-110 students per year), and AME 443 Control Systems Laboratory (60-70 students per year). The FANUC robotic platform in development consists of a FANUC robot, an interface between the robot and MATLAB and its toolboxes, and the necessary hardware and software for modeling and feedback control of the robot system (Fig. 1).

In this paper, we present the initial results obtained in this project, including those on the following issues: development of an accurate model of the robotic by importing 3-D data from Solidworks to SimMechanics; real-time simulation and control system implementation via conversion of the SimMechanics models into C code for Hardware-in-the-Loop tests using Real-Time Workshop; controller design and system-level analysis via MATLAB and SIMULINK; and experimental verification. Two sets of control experiments are to be performed to demonstrate control system development for robotic manipulators: open-loop inverse kinematic

control design, and basic joint space control design, which is a combination of open-loop torques computed using inverse kinematic and closed-loop PD and PID controllers.

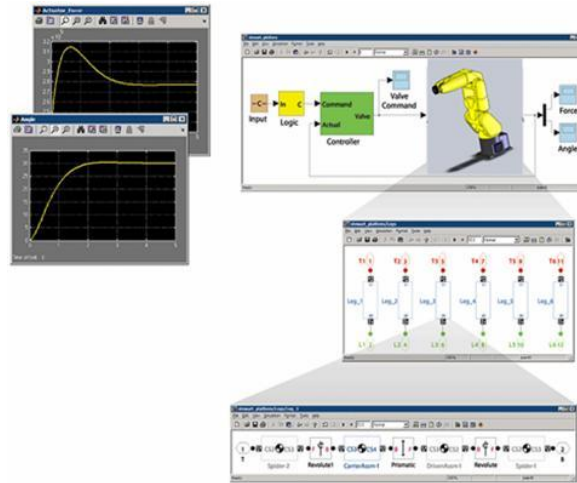


Figure 1. A FANUC robotic platform

These modified lab courses and related precursor courses, expose students to different aspects of modeling and simulation at an early stage of their studies. We plan to continue exposing students to the various aspects of Model-Based Design (MBD) techniques until the conclusion of their undergraduate education.

2. The Setting of Fanuc Robotic Platform

The Fanuc LR Mate 200iC robotic arm and R-30iA Mate controller box (see in Figs. 2 and 3) were successfully set up in the USC BHE301 lab (for undergraduate teaching in our department). The robotic arm has six degrees of freedom and six axis brakes. The six joints in total mimic the capabilities and reach of a human arm. Counting from the base the joints are named J1, J2, J3, J4, J5, and J6, respectively. The joints' maximum motion speed are J1=350, J2=350, J3=400, J4=450, J5=450, and J6=720 in units of degrees per second [1]. Maximum payload is 5 kg, and maximum reach is 0.704 meters with repeatability of ± 0.02 mm at full payload and full speed with the entire robot work envelope.

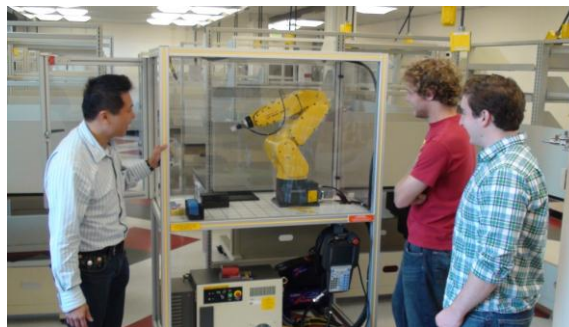


Figure 2. The LR Mate 200iC robotic arm

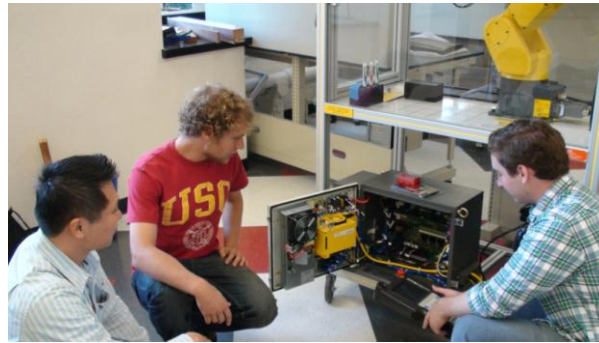


Figure 3. The R-30iA Mate controller box

For safety reasons, the arm is presently contained within a rectangular plastic casing measuring 110.5 ± 0.5 cm by 25.5 ± 0.5 cm by 86.3 ± 0.5 cm. Fanuc factory software is included in the current setup to limit the operating range of the arm such that limits are reached before the end effector comes into contact with the sides of the encasing. Dimensions and reach of the arm are detailed in Fig. 4.

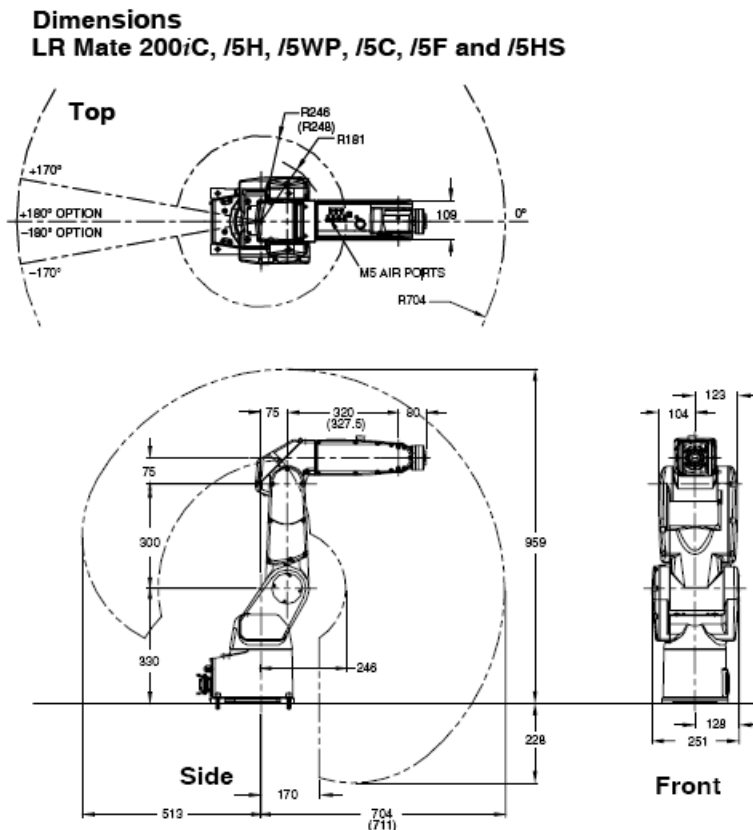


Figure 4. Dimensions and reach of the LR Mate 200iC robotic arm

The controller box, as it is currently set up, is designed to receive commands from the teaching pendant (a device provided by Fanuc for manually controlling the hardware) and to move the

motors accordingly. The controller box is equipped with an on/off switch, a circuit breaker, an emergency stop that activates dynamic braking for each joint, and a light indicating whether a fault has occurred. The controller box includes an 8-axis control board, a 6 channel servo amplifier, dual check safety, and a CPU featuring an Ethernet port and multi-processor based architecture. The teaching pendant is set up for manual control of the robot when the arm is not set to run a specific program. A product documentation CD is also included and comes with HandlingTool software for robot control, instructions on setup and operations, a mechanical unit operators manual, and directions for maintenance.

The goals of the project include the following. First, operator training will be completed, and HandlingTool software will be used for simple operation of the robot and design of simple algorithms. MATLAB software will then be written and incorporated into a “hardware in-the-loop” system, which will feature a MATLAB interface and will allow a user to send commands from the interface through the controller box to the robot. A modeling component will also be incorporated in order to demonstrate the result of algorithms as the hardware itself is responding to user-issued commands.

3. Mathematical Modeling for Model-Based Control Design

3.1 System model

One objective of the project is to introduce the students to all the aspects of model based design by using a system for which the equations are simple and can be derived analytically. To achieve this we lock all but two joints of the robot to establish a two degree of freedom system with rotational joints performing a motion in the plane. Shown in Fig. 5 is such a model of the robot.

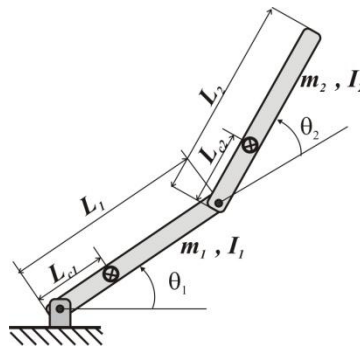


Figure 5. A 2-DOF model of the robotic arm

The dynamic model for the above system is described as follows [2, 3]:

a) Kinematics

$$\begin{aligned} x_e &= L_1 \cos \theta_1 + L_2 \cos (\theta_1 + \theta_2) \\ y_e &= L_1 \sin \theta_1 + L_2 \sin (\theta_1 + \theta_2) \end{aligned} \tag{1}$$

b) Inverse kinematics

$$\begin{aligned}\theta_1 &= \beta - \text{sign}(\theta_2)\psi \\ \theta_2 &= a \tan 2\left(D, \pm\sqrt{1-D^2}\right) \\ \text{where}\end{aligned}\tag{2}$$

$$D = \frac{x_e^2 + y_e^2 - L_1^2 - L_2^2}{2L_1L_2}, \quad \beta = a \tan 2(y_e, x_e), \quad \psi = \cos^{-1}\left[\frac{x_e^2 + y_e^2 + L_1^2 - L_2^2}{2L_1\sqrt{x_e^2 + y_e^2}}\right]$$

c) Dynamics

Denote $h = m_2L_1L_{c2} \sin \theta_2$ and $\varphi = \theta_1 + \theta_2$ then the equations of motion are given by

$$\begin{aligned}M_{i1}(\boldsymbol{\theta})\ddot{\theta}_1 + M_{i2}(\boldsymbol{\theta})\ddot{\theta}_2 + H_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) &= u_i \quad i=1,2 \\ \text{where} \\ M_{11}(\boldsymbol{\theta}) &= m_1L_{c1}^2 + I_1 + m_2L_{c2}^2 + I_2 + m_2L_1^2 + 2m_2L_1L_{c2} \cos \theta_2 \\ M_{12}(\boldsymbol{\theta}) &= M_{21}(\boldsymbol{\theta}) = m_2L_{c2}^2 + I_2 + m_2L_1L_{c2} \cos \theta_2, \quad M_{22}(\boldsymbol{\theta}) = m_2L_{c2}^2 + I_2 \\ H_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) &= -2h\dot{\theta}_1\dot{\theta}_2 - h\dot{\theta}_2^2 + (m_1L_{c1} + m_2L_1)g \cos \theta_1 + m_2gL_{c2} \cos \varphi \\ H_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) &= h\dot{\theta}_1^2 + m_2gL_{c2} \cos \varphi\end{aligned}\tag{3}$$

d) Differential kinematics

The relation between joint velocities and end-effector velocity components is

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = J(\boldsymbol{\theta}) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \quad J(\boldsymbol{\theta}) = \begin{bmatrix} -L_1 \sin \theta_1 - L_2 \sin \varphi & -L_2 \sin \varphi \\ L_1 \cos \theta_1 + L_2 \cos \varphi & L_2 \cos \varphi \end{bmatrix}\tag{4}$$

The relation between joint accelerations and end-effector acceleration components is

$$\begin{bmatrix} \ddot{x}_e \\ \ddot{y}_e \end{bmatrix} = J(\boldsymbol{\theta}) \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \dot{J}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \quad \dot{J}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \begin{bmatrix} -\dot{\theta}_1L_1 \cos \theta_1 - \dot{\varphi}L_2 \cos \varphi & -\dot{\varphi}L_2 \cos \varphi \\ -\dot{\theta}_1L_1 \sin \theta_1 - \dot{\varphi}L_2 \sin \varphi & -\dot{\varphi}L_2 \sin \varphi \end{bmatrix}\tag{5}$$

3.2. Description of the task performed by the robot

The first task chosen to be performed by the robot is to move the end-effector in the plane along a straight line from location (x_0, y_0) to location (x_f, y_f) in prescribed time t_f , see Figure 6. The commanded end-effector trajectory was computed as

$$\begin{aligned}x^*(t) &= x_0 + \alpha t^2 + \beta t^3, \quad \text{where } \alpha = \frac{3(x_f - x_0)}{t_f^2}, \quad \beta = -\frac{2(x_f - x_0)}{t_f^3} \\ y^*(t) &= mx^*(t) + n, \quad \text{where } m = \frac{y_f - y_0}{x_f - x_0}, \quad n = -\frac{2(x_f y_0 - x_0 y_f)}{t_f^3}\end{aligned}\tag{6}$$

The commanded velocities and accelerations in the task space are computed by successive differentiation of Eq. (6). The commanded trajectory in joint space is computed by using the inverse kinematic equations (2) and the commanded joint velocities and accelerations in joint space are computed by using the equations of inverse differential kinematics (4) and (5) for $\dot{\theta}_i$ and $\ddot{\theta}_i, i = 1, 2$

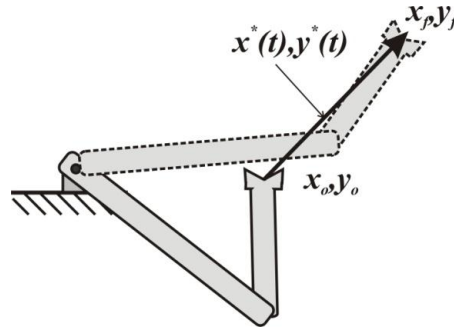


Figure 6: Commanded trajectory description

3.3. MATLAB simulation program

A simulation program has been developed in MATLAB to simulate control laws in joint space. The program is modular where each function implements a different part of the model and control algorithm. Because of its modular structure, we will be able to use it as a template for future studies when the robot will have more degrees of freedom that are active and more complex control laws, both in joint space and end-effector space are implemented.

The following MATLAB functions were developed:

- a. Function *rrmain* (main program) – integrates equations of motion, computes nominal trajectory, plots the results.
- b. Function *eetraj* – computes the commanded end-effector trajectory for a given task, see Eq. (6).
- c. Function *Kin* - computes end-effector trajectory, its velocity and acceleration given joint motion, see Eqs. (1), (4) and (5).
- d. Function *InvKn* - computes end- Computes trajectories in joint space, their velocity and acceleration given end-effector motion, see Eqs. (1), (4) and (5).
- e. Function *accrr* – computes robot accelerations in joint space for a given control torques, see Eq. (3).
- f. Function *cont* - implements control algorithm by computing torques given nominal and actual trajectory measurements.
- g. Function *par* – inputs robot and control algorithm parameters (inertial parameters, geometry, control gains, saturation limits, etc.)

3.4. Simulation of control

In the first stage of our project two relatively simple control laws are implemented on the robot as follows:

a. Independent joint control

The feedback controller is given by

$$\begin{aligned}
 u_i &= \bar{m}_i \left[\ddot{\theta}_i^* - k_{pi} (\theta_i - \theta_i^*) - k_{ri} (\dot{\theta}_i - \dot{\theta}_i^*) \right] \quad i = 1, 2 \\
 \bar{m}_1 &= m_1 L_{c1}^2 + I_1 + m_2 L_{c2}^2 + I_2 + m_2 L_1^2 + 2m_2 L_1 L_{c2} \\
 \bar{m}_2 &= m_2 L_{c2}^2 + I_2
 \end{aligned} \tag{7}$$

b. Model-based control

$$u_i = M_{i1}(\boldsymbol{\theta}) \ddot{\theta}_1^* + M_{i2}(\boldsymbol{\theta}) \ddot{\theta}_2^* - k_{pi} (\theta_i - \theta_i^*) - k_{ri} (\dot{\theta}_i - \dot{\theta}_i^*) + H_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \quad i = 1, 2 \tag{8}$$

In Eqs. (7) and (8), u_i are control torques shown in Eq. (3), θ_i^* are the reference angles (desired output) of the robotic system, and k_{pi} , k_{ri} are control gains. Simulation results for the independent joint control design are shown in Figs. 7 and 8. An error of 10% in the mass properties was assumed and a saturation of the actuators was also implemented.

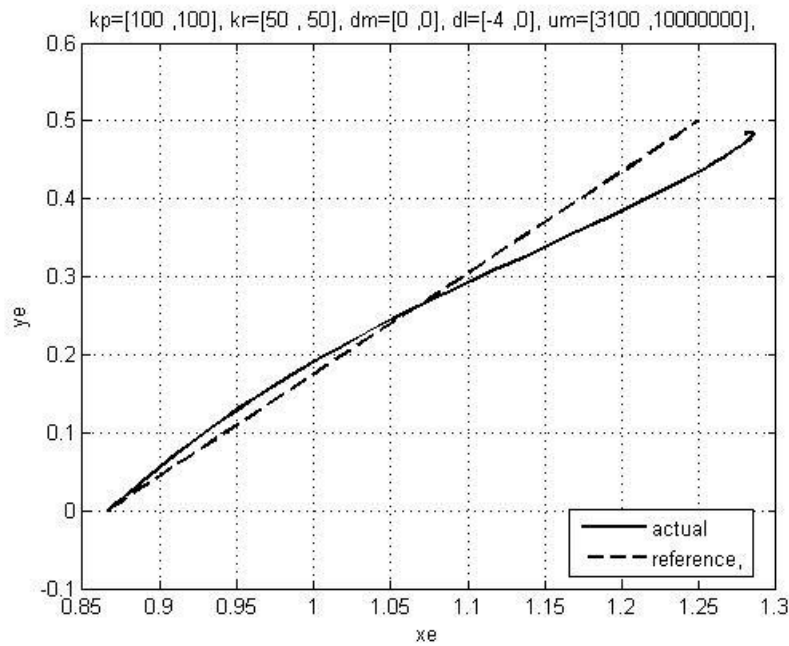


Figure 7. End-effector trajectory under independent joint control design

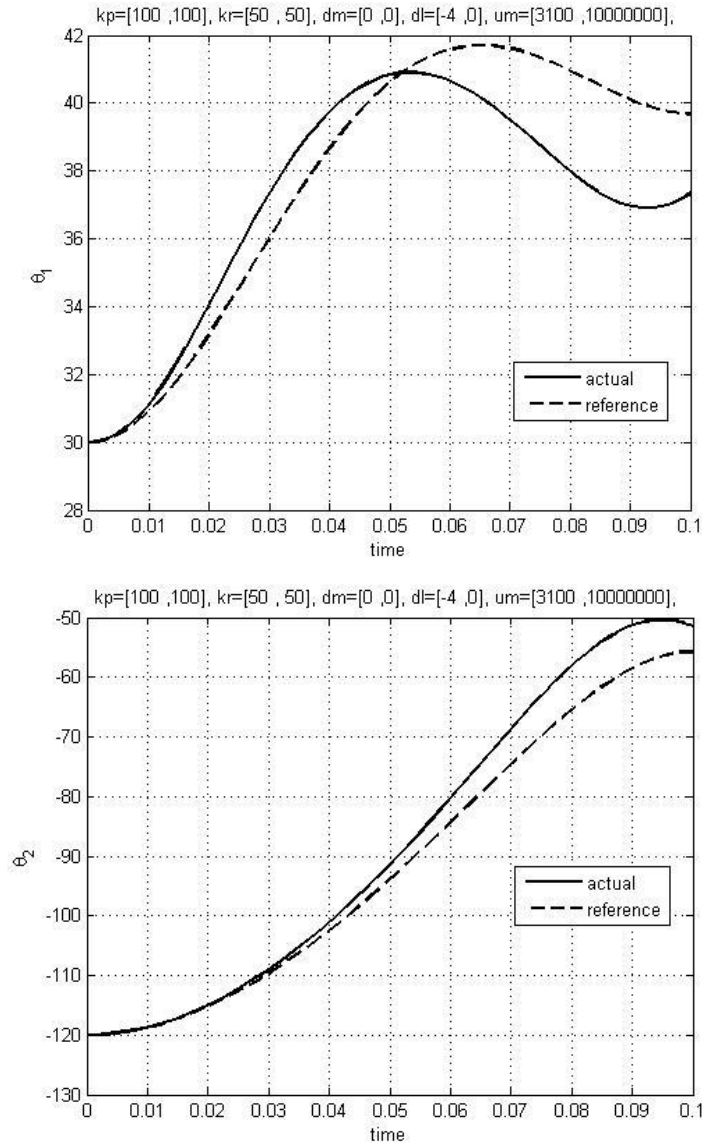


Figure 8. Trajectory in joint space for independent joint control design

Simulation results for model-based control design are shown in Figs. 9 and 10.

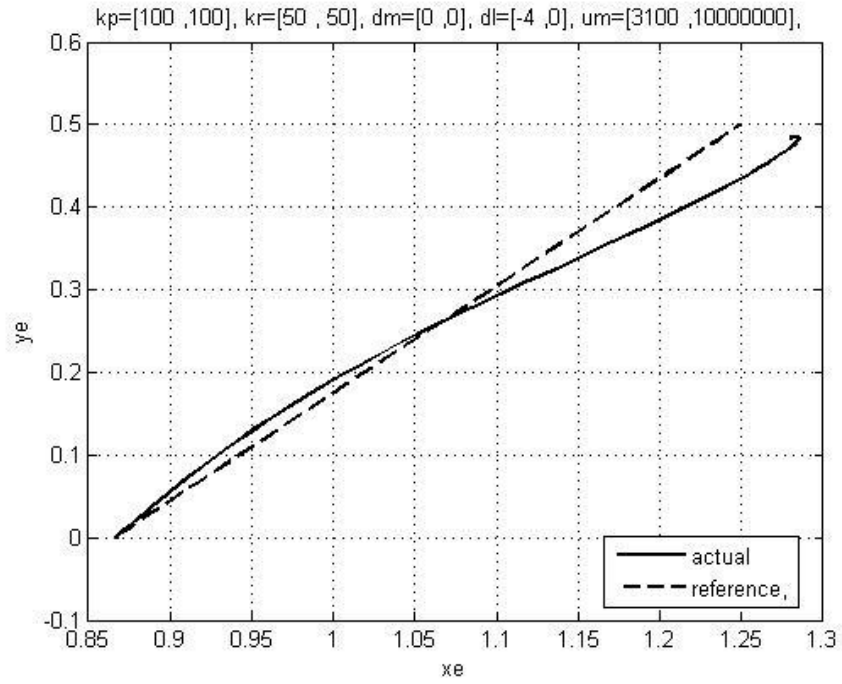
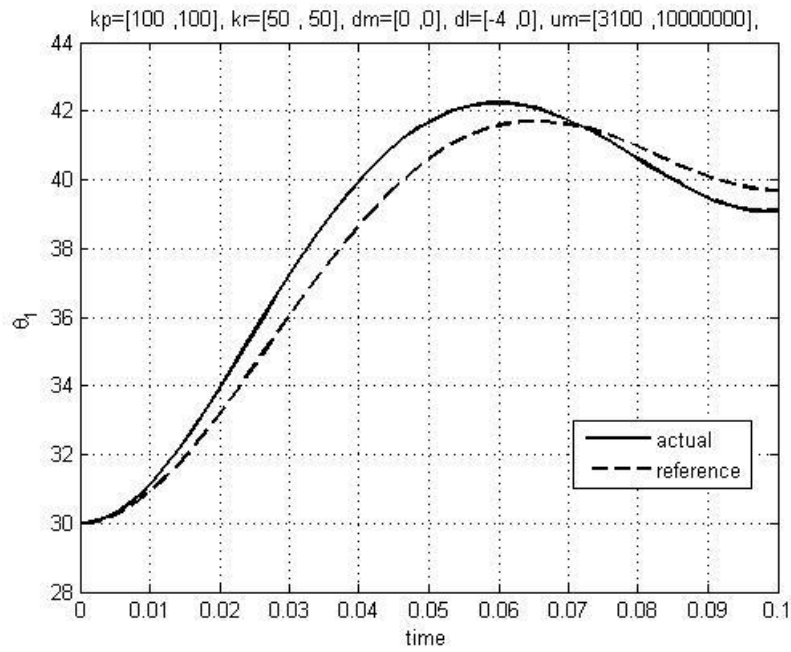


Figure 9. End-effector trajectory under model-based control design



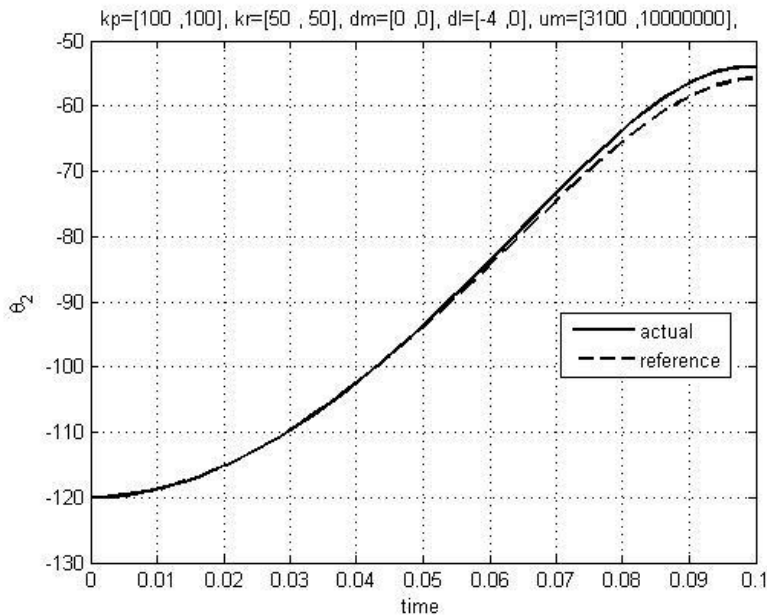


Figure 10. Trajectory in joint space for model-based control design

4. Implementation Plan

This effort is aimed at a major modification of the following two undergraduate laboratory courses in dynamics and control:

- AME 441 Senior Projects Laboratory (100-110 students per year)
- AME 443 Control Systems Laboratory (60-70 students per year).

We acquired the Fanuc robotic system at the end of 2010. The setting and mathematical modeling results, as described in Sections 2 and 3, will be demonstrated in the above two courses in Spring 2011, and later on integrated in the new course materials. During the summer of 2011, several selected undergraduate students will involve in developing new course materials related to the robotic platform. In Fall 2011, we expect full implementation of the obtained new results in our curriculum that we will use the robotic platform for teaching and projects in these two courses.

The communication between host station (MATLAB/SIMULINK) and Fanuc controller (R-30iA) is performed by TCP/IP based protocol (called socket message) through Ethernet cable. User-defined program (using Fanuc's KAREL) will reside in Fanuc controller to allow high-level controller implemented in MATLAB/SIMULINK to send commands and retrieve information such as encoder value, register value, and I/O status. User-defined program performs analyzing commands, storing data into registers and sending data back to the high level controller. According to the commands that high-level controller sends, commanded motion will be carried out by Fanuc controller, as shown in Fig. 11. The completion of the integration of the Fanuc robotic system with MATLAB/SIMULINK software, will enable the students of AME 443 and AME 441 to learn MBD techniques. In particular, the students in these classes will perform new experiments and conduct new projects as outlined below.

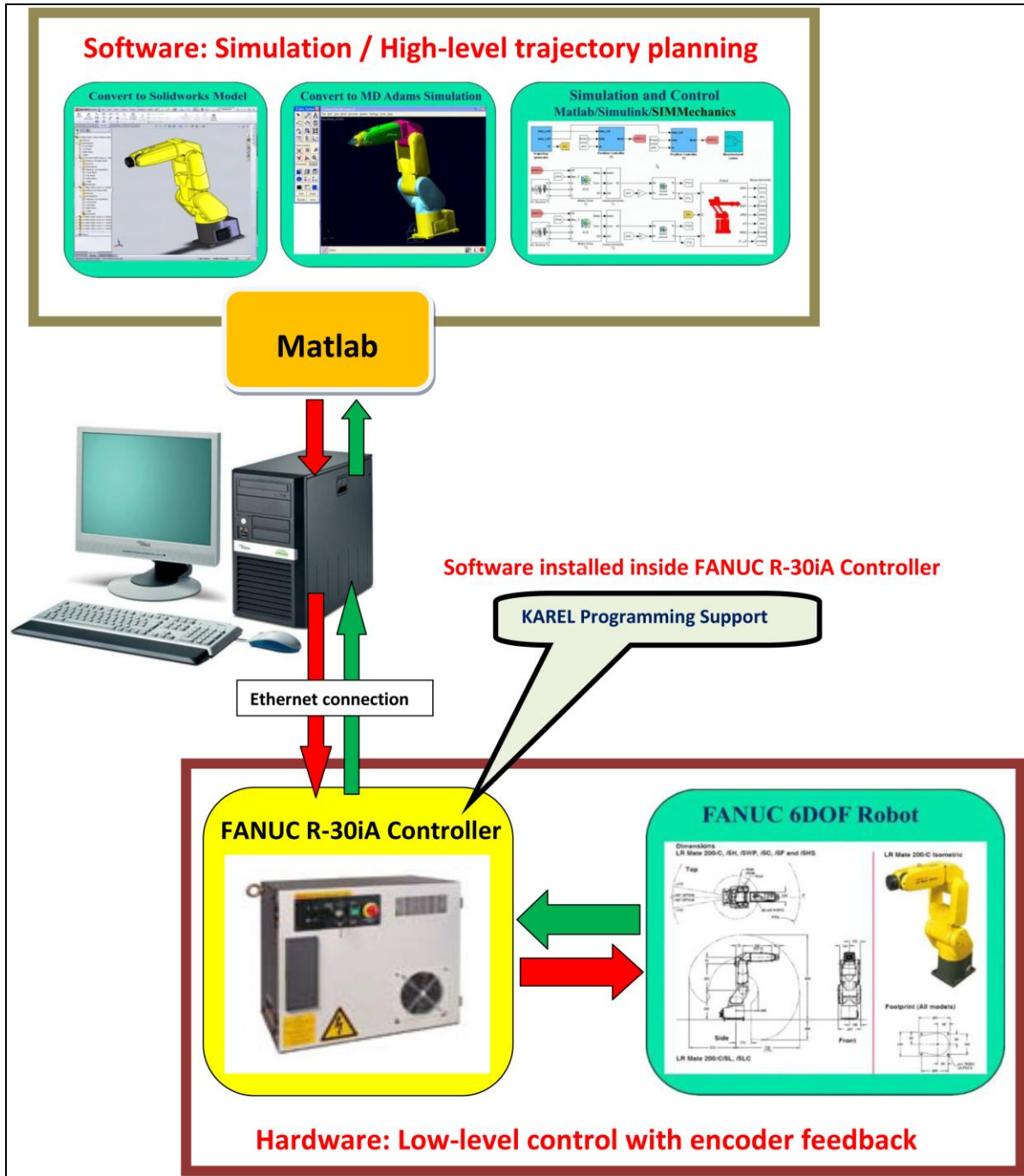


Figure 11. A FANUC robot platform with integrated MathWorks software

AME 443 Control Systems Laboratory: The following experiments will be added:

- System Modeling via MATLAB and MSC ADAMS
- System Identifications and System Verifications
- Kinematics Simulations vs Robot Experiments
- Gravity Compensation Simulations vs Experiments
- Controller Design and Implementation

- Open Loop control by Inverse Kinematics
- Feedback Controller Design to Meet Performance Specifications
- Single DOF vs Multiple DOF Motion
- Trajectory Tracking
- Vision Based Feedback Control

AME441 Senior Projects Laboratory: The following projects will be added:

- Design, simulation, and implementation for a trajectory control of a robotic manipulator. The project will require the use of MATLAB, SIMULINK, SIMMechanics and MSC Adams
- Design, simulation, and implementation of energy based control of a robotic manipulator. The project will require the use of MATLAB, SIMULINK, SIMMechanics and MSC Adams
- Design, simulation, and implementation for robust min-max control of a robotic manipulator. The project will require the use of MATLAB, SIMULINK, SIMMechanics and MSC Adams

This project will eventually achieve the following specific goals:

- Introduce students to state of the art techniques for development of dynamic systems needed in Mechtronics and robotics
- Demonstrate to the students the clear relation between analysis, design, modeling, and hardware implementation
- Teach the students all aspects of dynamic system development preparing them for work in this area
- Give students hands-on experience in designing and implementing dynamic systems thus enhancing their academic experience.

Conclusions

A Fanuc robotic platform is being developed at the USC Aerospace and Mechanical Engineering Department. The platform will be integrated with Mathworks software, which will allow our undergraduate students to learn state-of-the art Model-Based Design (MBD) techniques. The successful completion of this project will significantly modify two existing undergraduate laboratory courses in dynamics and control.

References

- [1] FANUC Robotics America online support [<http://www.fanucrobotics.com/file-repository/DataSheets/Robots/LR-Mate-200iC-Series-&-R-30iA-Mate-Controller.pdf>]
- [2] H. Asada and J. J. E. Slotine: *Robot Analysis and Control*, Wiley-Interscience (1986)
- [3] J. Y. S. Luh, and C. S. Lin, "Automatic Generation of Dynamic Equations for Mechanical Manipulators", *1981 Joint Automatic Control Conference*, Charlottesville, VA (1981)