Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Modelling and Simulation of Robot Arm Interaction Forces Using Impedance Control

José de Gea * **Frank Kirchner** *,**

* *Robotics Group, University of Bremen, Germany*
(e-mail: jdegea@informatik.uni-bremen.de)
** *DFKI (German Research Center for Artificial Intelligence)*
Robotics Lab Bremen, Germany
(e-mail: frank.kirchner@dfki.de)

Abstract: In this paper we present the implementation of a Cartesian impedance control method to regulate the interaction forces between a robotic arm and the environment. A complete description of the procedure to model and control both a two-link planar robot arm and its interaction with the environment is detailed and simulated using MATLAB/Simulink; from the generation of a mechanical model in SimMechanics (MATLAB), the description and tuning of a dynamic model-based controller to cancel-out the non-linearities present on the dynamic model of the robot, the modelling of an environment, and finally the control of the interaction forces making use of a Cartesian impedance control method. This type of control adjusts the dynamic behaviour of the robot manipulator when contacting the environment, basically controlling stiffness and damping of the interaction rather than the precise contact forces. Its implementation in the Cartesian Space permits future use of the results in an industrial robot, whose internal joint and torque controllers are commonly not accessible.

Keywords: Impedance control; Robotic manipulator; Simulation; Modelling; Interaction.

1. INTRODUCTION

For a robot manipulator to interact safely and human-friendly in unknown environments, it is necessary to include an interaction control method that reliably adapts the forces exerted on the environment in order to avoid damages both in the environment and in the manipulator itself. A force control method, or strictly speaking, a direct force control method, can be used on those applications where the maximum or the desired force to exert is known beforehand. In some industrial applications the objects to handle or work with are completely known as well as the precise moment on which these contacts are going to happen. In a more general scenario, such as one outside a well-defined robotic workcell, sometimes neither the objects nor the time when a contact is ocurring are known.

In such case, indirect force control methods find their niche. These methods do not seek to control maximum or desired force, but they try to make the manipulator compliant with the object being contacted. The major role in the control loop is given to the positioning but the interaction is also being controlled so as to ensure a safe and clear contact. In case contact's interaction forces have exceeded the desired levels, the positioning accuracy will be diminished to account and take care of the (at this moment) most important task: the control of the forces.

Impedance control [1] is one of these indirect force control methods. Its aim is to control the dynamic behaviour of the robot manipulator when contacting the environment, not by controlling the exact contact forces but the properties of the contact, namely, controlling the stiffness and the damping of the interaction. Moreover, the steady-state force can be easily set to a desired maximum value. The main idea is that the impedance control system creates a virtual new impedance for the manipulator, which is being able to interact with the environment as if new mechanical elements had been included in the real manipulator.

First industrial approaches were focused on controlling the force exerted on the environment by a direct force feedback loop. A state-of-the-art review of the 80s is provided in [2] and the progress during the 90s is described in [3]. In many industrial applications, where objects are located in a known position in space and where the nature of the object is also familiar, the approach is well-suited since it prevents the robot from damaging the goods. If a detailed model of the environment is not available, the strategy is to follow a motion/force control method obtained by closing a force control loop around a motion control loop [4]. If controlling the contact force to a desired value is not a requirement, but rather the interest is to achieve a desired dynamic behaviour of the interaction, indirect force control methods find their application. This would be the case when the environment is unknown and the objects to manipulate have non-uniform and/or deformable features. In this strategy, the position error is related to the contact force through mechanical stiffness or with adjustable parameters. This category includes compliance (or stiffness) control [5], [6] and impedance control [1], [11], [9] and [10].

This paper presents a case study where impedance control is used to control the interaction forces of a simulated two-link planar arm. To achieve this goal, first a mechanical model of the robot will be created using SimMechanics. A model-based controller will be used that requires of a mathematical description of the mechanical system in order to linearise and decouple it. If the mathematical description matches the mechanical system described in SimMechanics, the non-linearities of the system will be cancelled-out. Once the system is linear, a simple PD controller can be used to implement a joint control. As final step the environment is modelled and a Cartesian impedance controller is used to control the forces exerted for the robot. To test the control system, the robot is given a reference input trajectory that causes the robot to hit a wall so as to observe the performance of the robot-environment interaction.

2. DESCRIPTION

2.1 Mechanical Model

The first step is to create a model of a simple two-link planar arm to test our approach. Figure 1 shows the mechanical model of the two-link planar arm described with SimMechanics. The model is composed by two revolute joints and two bodies. The module receives torques as input and outputs joint angles, torques and end-effector position in Cartesian coordinates. The masses are considered to be concentrated at the end of each link, to simplify the modelling tasks. In SimMechanics this is realised setting the inertias as: 'zeros(3,3)', a 3-by-3 matrix that defines point masses and defining the Center of Gravity (CG) on the edge of each link, since Matlab computes the inertias at the CG.

The model of the system is non-linear and highly coupled. For the control system, there are two alternatives: either to use non-linear control techniques or to linearise the system and apply well-known linear control techniques. The second is usually the option adopted and will also be used here. An elegant way is described in literature to linearise and decouple the system [8]. The method makes use of the dynamic model of the system, and that means that the more accurate the dynamic model is, the better will be the linearisation/decoupling process and its posterior control success. The next section will describe the dynamical model of the mechanical manipulator.

2.2 Dynamical Model

The dynamic model of a robot manipulator relates the forces acting on the mechanical structure with the resulting displacements, velocities and accelerations. These forces can have different sources: the torques delivered by the motors, the inertia of the mechanical links, the gravity, Coriolis and centripetal forces, the friction forces and the possible forces exerted for the environment on the robot. Given an initial state of the mechanical structure and the time history of torques $\tau(t)$ acting at joints, the direct dynamic model allows to predict the resulting motion $\theta(t)$ (and its derivatives) in joint space. With this information and the direct kinematic model, a prediction of the trajectory $x(t)$ in Cartesian coordinates can be performed.

The dynamic model of a n-joint robot manipulator can be written in the Lagrangian form as

$$M(\theta)\ddot{\theta} + B(\theta, \dot{\theta}) + G(\theta) = u, \tag{1}$$

where θ is the joint variable n-vector and u is the vector of generalized forces acting on the robot manipulator. $M(\theta)$ is the inertia matrix, $B(\theta, \dot{\theta})$ are the Coriolis/centripetal forces, and $G(\theta)$ is the gravity vector. In Equation (1) we are not taking into account the friction torques that are always to be found in a real robot manipulator. The dynamic model of the two-link planar arm [7] following the Lagrangian formulation is:

$$M(\theta)\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + B(\theta, \dot{\theta}) + G(\theta) = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2}$$

where

$$B(\dot{\theta}, \ddot{\theta}) = \begin{bmatrix} -m_2 a_1 a_2 (2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^{\,2}) \sin\theta_2 \\ m_2 a_1 a_2 \dot{\theta}_1^{\,2} \sin\theta_2 \end{bmatrix} \tag{3}$$

$$G(\theta) = \begin{bmatrix} (m_1 + m_2) g a_1 \cos\theta_1 + m_2 g a_2 \cos(\theta_1 + \theta_2) \\ m_2 g a_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \tag{4}$$

$$M(\theta) = \begin{bmatrix} h & i \\ j & k \end{bmatrix} \tag{5}$$

with

$$h = (m_1 + m_2)a_1^2 + m_2 a_2^2 + 2 m_2 a_1 a_2 \cos\theta_2 \tag{6}$$
$$i = m_2 a_2^2 + m_2 a_1 a_2 \cos\theta_2 \tag{7}$$
$$j = m_2 a_2^2 + m_2 a_1 a_2 \cos\theta_2 \tag{8}$$
$$k = m_2 a_2^2 \tag{9}$$

The terms a_1 and a_2 are the lengths of links 1 and 2, respectively and m_1 and m_2 their masses. In our example, $a_1 = a_2 = 0.2m$ and $m_1 = m_2 = 10kg$.

2.3 Linearisation and Decoupling

Let's describe now the technique to linearise and decouple the system. Let's define a controller such as

$$\alpha u' + \beta \tag{10}$$

being u' the new control input, and define

$$\alpha = M(\theta) \tag{11}$$
$$\beta = B(\theta, \dot{\theta}) + G(\theta) \tag{12}$$

Combining the controller with the dynamic model $M(\theta)\ddot{\theta} + B(\theta, \dot{\theta}) + G(\theta) = \alpha u' + \beta$ and simplifying that leads to the system

$$\ddot{\theta} = u' \tag{13}$$

Our control input will have to deal with a linear and very simple model. This solution will work as long as

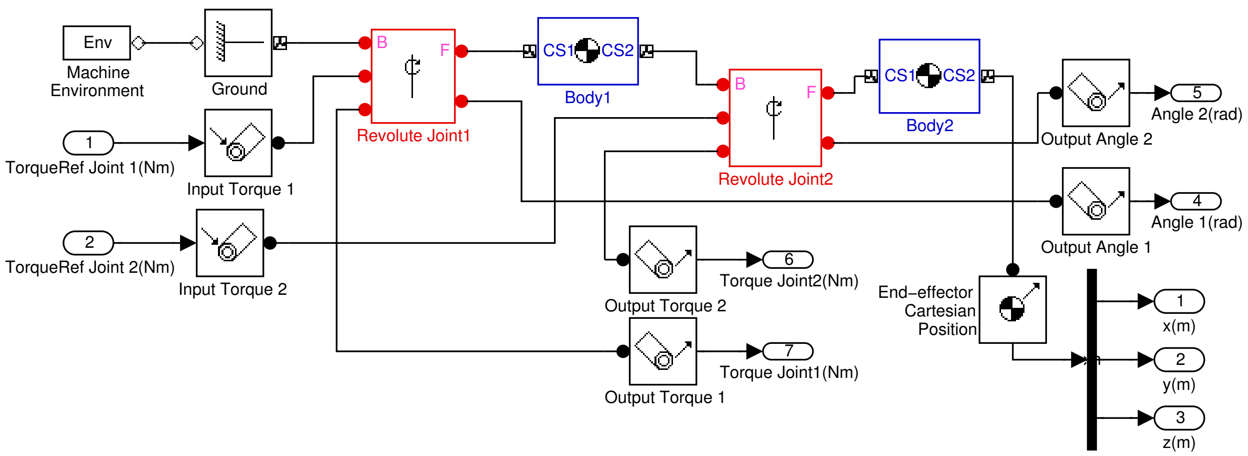


Fig. 1. Mechanical model in SimMechanics

it is possible to accurately represent and implement α and β. As we will see later with the simulations in MATLAB, the analytical solution for α and β matches precisely the dynamics described by the mechanical model in SimMechanics. The new control input u' might be then easily implemented as a typical PD controller:

$$u' = -K_P(\theta_d - \theta) - K_V(\dot{\theta}_d - \dot{\theta}) \qquad (14)$$

In other words, Equations (11) and (12) describe a dynamic model-based controller for the manipulator that will be used to cancel the non-linearities of the manipulator in order to achieve a model to control as simple as a double integrator represented in (13). The general structure of the dynamic model-based controller can be seen in Figure 2.

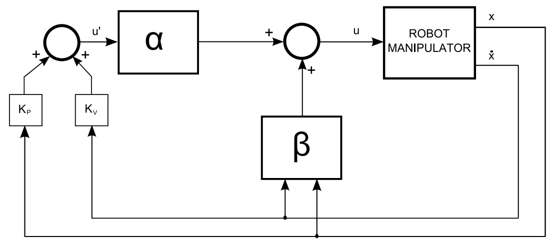


Fig. 2. General structure of the dynamic model-based controller

Its implemenation in Simulink can be seen in Figure 3.

2.4 Inverse Kinematics

In most of the industrial manipulators, the possibility to send torque commands directly to the robot is not available. To simplify the tasks of the control engineer, these industrial robots include an internal joint controller together with the necessary inverse kinematics algorithms. In other words, the designer can choose either to send direct joint angle commands to the robot or an end-effector position/orientation command in Cartesian coordinates.

In this case and in order to be able to easily port some of the results of this work to an industrial manipulator, an inverse kinematics module has been also integrated

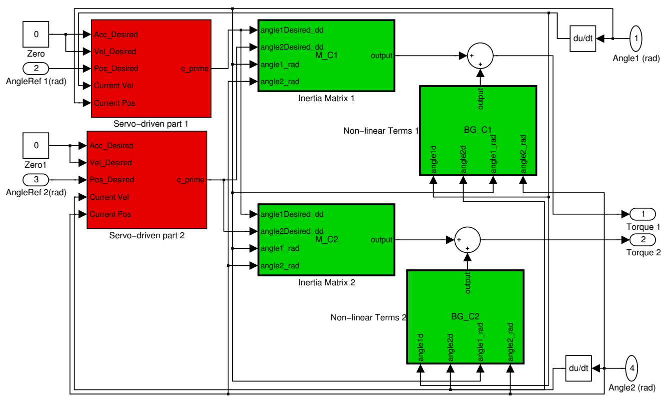
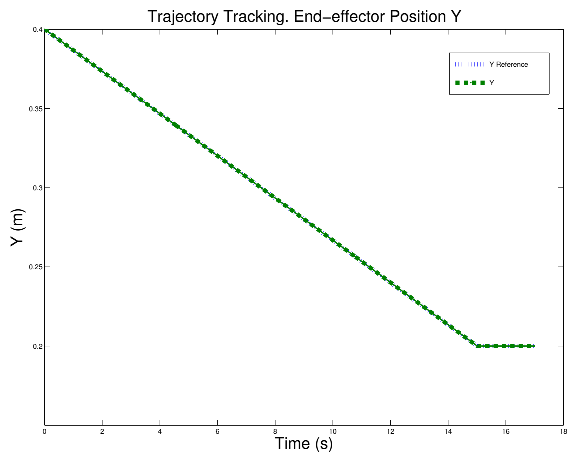


Fig. 3. Dynamic model-based controller in Simulink

in Simulink. In this way, the mechanical model of the manipulator, the dynamic model-based controller and the inverse kinematics module can be considered the "black box" that industrial robots are. From this point on, our strategies can build up outer control loops that will be easily tested on a real manipulator, namely, concentrating on Cartesian impedance controllers rather than on internal joint controllers.

Figure 4 shows the trajectory tracking performance of the system under control for a given ramp trajectory in Cartesian space. The parameters K_P and K_V were found empirically ($K_P = 10000N/m, K_V = 100Nms/rad$) but it is easy to demonstrate how to choose them depending on the desired dynamical response. If the error is defined as $e = \theta_d - \theta$, combining (13) and (14), the system's error equation is that of a second order system with the format $s^2 + 2\xi\omega_n + \omega_n^2 = 0$ where the resonant frequency ω_n and the damping coefficient ξ are computed as $\omega_n = \sqrt{K_P}$ and $\xi = \frac{K_V}{2\sqrt{K_P}}$. As the results show, after cancelling the non-linearities of the mechanical model with the use of the model-based part of the controller, it is fairly easy to tune the PD parameters so as to achieve the desired performance of the control system.



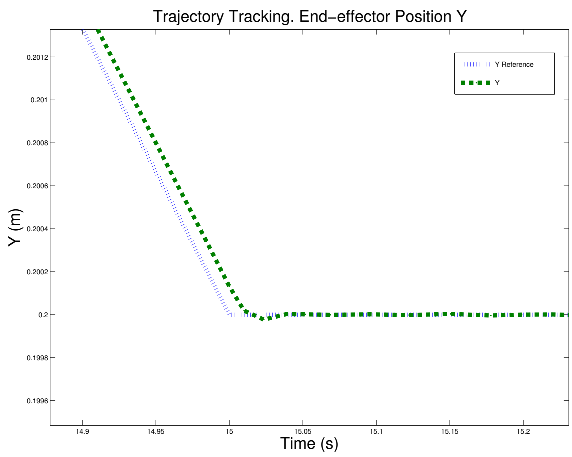


Fig. 4. Trajectory tracking performance *left:*End-effector Position Y *right:* Zoom

2.5 Modelling the Environment

So far, our design includes a manipulator model and its control system, a system that is able to track a desired trajectory in Cartesian space. Our dynamic model does not take into account possible external forces acting on the robot that would definitively change the dynamic behaviour of our manipulator. To account for this situation, a model of the environment will be included in our system whose interaction force will act on our robot. Often a simple linear spring model is used as model for the environment:

$$f = K_e(x - x_e) \tag{15}$$

where f is the contact force, K_e is the stiffness of the environment, x is the end-effector position at the contact point and x_e is the static position of the environment. We assume that the environmental stiffness can be modelled as a linear spring with a spring constant K_e. Figure 5 depicts such a concept, where a manipulator of mass m contacts the environment at position x_e trying to reach the desired end-effector position x_d. In our case, we will include a damping coefficient B_e such as the environment is modelled as

$$f = K_e(x - x_e) + B_e(\dot{x} - \dot{x_e}) \tag{16}$$

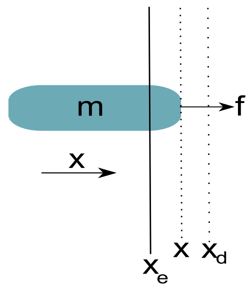


Fig. 5. Manipulator in contact with the environment

2.6 Modelling Contact Forces

The dynamic equation describing the behaviour of our system was defined in (1). In that case, u was considered to include the effects of all the forces acting on the robot, i.e. also the external contact forces. To make it clearer, we will modify (1) to show the effect of those forces and will compensate for them in our model. The dynamic equation governing the robot's behaviour might be defined as

$$M(\theta)\ddot{\theta} + B(\theta, \dot{\theta}) + G(\theta) = u - J^T(\theta)f \tag{17}$$

where now u defines the driving torques and the term $J^T(\theta)f$ translates the task-space forces f acting on the end-effector to the joint space making use of the traspose of the Jacobian. Our next Simulink model will include Equation (17) in order to take account for contact forces on the dynamic response. In the case of a two-link planar arm, and knowing that the relation between torques and forces is defined as $\tau_c = J^T f$, the following relation can be written:

$$\tau_{1c} = J_{11}f_x + J_{21}f_y \tag{18}$$

$$\tau_{2c} = J_{12}f_x + J_{22}f_y \tag{19}$$

where τ_c are the contact torques, J_{ij} are the elements of the traspose of the 2x2 Jacobian matrix and f_x and f_y are the forces along the X and Y directions, respectively. Since we assume that no forces are acting along the Y axis, because its path in this direction is free and we don't consider friction forces, we can assume that $f_y = 0$ so that

$$\tau_{1c} = J_{11}f_x \tag{20}$$

$$\tau_{2c} = J_{12}f_x \tag{21}$$

These torques are to be included in equation (17).

2.7 Impedance Controller

So far we have modelled and controlled the dynamics of a two-link planar arm and created a simple model for the environment. The next step is to design a controller that regulates the interaction when the robot contacts the environment. In the current state, if the robot follows a trajectory that finds on its way an object, the robot will collide with it, trying to reach the final end position of the given trajectory, and likely exerting such a huge forces into the environment that would likely cause damages to a real robot (and/or the object of collision). To avoid this situation, an impedance controller will be designed so that its input is the reference trajectory at each time step. Additionally, the measured contact forces will be included on the controller so as to get an immediate feedback of the contact state. The output will be a modified trajectory that takes into account the contact forces. That means, if no forces are sensed, the trajectory will be followed strictly. Otherwise, when forces are measured, the trajectory will be modified in order to limit the maximum steady-state forces and to dynamically behave as the mass-spring-damper system described in the control law given with (22). Thus the impedance controller is modelled as

$$M_T \ddot{e}_t + D_T \dot{e}_t + K_T e_t = f \tag{22}$$

where M_T, D_T and K_T are the inertia, damping and the stiffness coefficients, respectively, e_t is the trajectory error, defined as $e_t = p_d - p_r$ where p_d is the desired trajectory input and p_r will be the reference trajectory for the next module (the inverse kinematics), corrected depending on the value of the contact force f. M_T, D_T and K_T will define the dynamic behaviour of the robot that could be compared to the effect of including physical springs and dampers on the robot. Figure 6 shows the structure of the impedance controller. The controller parameters were $M_T = 5kg$, $D_T = 1000Ns/m$ and $K_T = 10N/m$.

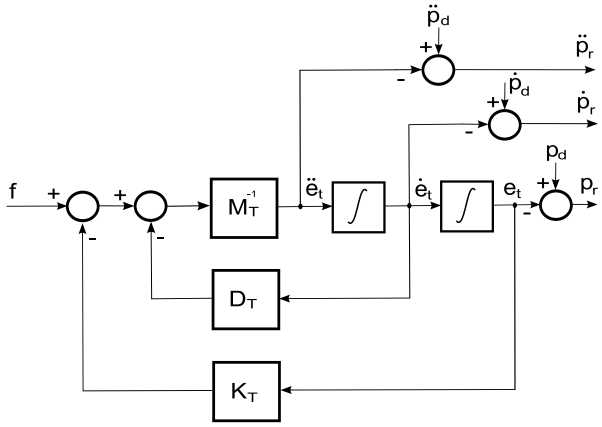


Fig. 6. Impedance controller diagram

3. RESULTS

Figure 7 shows the complete MATLAB model of the control system. It includes the mechanical model, the dynamic model-based controller, the model of the environment, the impedance controller and the desired input trajectories.

In order to test the performance of the control system, the robot is given a desired trajectory to follow. The input trajectory for the X axis is a ramp starting at $X = 0$ with *slope* $= 0.3/15$. The reference for the Y-axis is a ramp starting a $Y = 0.4$ with *slope* $= -0.2/15$. At $X = 0.3$ we installed a wall modelled as defined in (16) with $K_e = 500000N/m$ and $B_e = 0.1Ns/m$. Figure 8 shows the response of the system while approaching and contacting the environment. The upper two plots on Figure 8 show how at the contact point ($X = 0.3$) the contact force f increases dramatically and how the impedance controller reacts to redefine a new position trajectory that limits the steady-state forces to a value of around 20N by limiting, in this case, the X reference position to a value of 0.3. In Figure 9 we can see the results of the same experiment without using the impedance controller. The manipulator tries to follow the given input trajectory and after contacting the wall surface, the contact forces increase exponentially as the robot travels "inside" the wall.

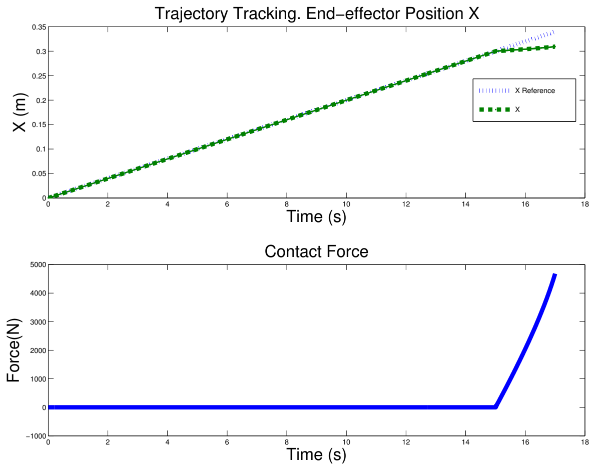


Fig. 9. Response without Impedance Controller *top:* reference and current X Position *bottom*: contact force

4. CONCLUSIONS

This paper describes the process of modelling a simple robotic manipulator and its interaction with the environment. The use of SimMechanics as a tool to model the mechanics of the robot allows the possibility to verify model-based control algorithms. The precise mechanical model, as in a real application, is unknown for the designer so that the ability of the algorithm to match the real model can be easily proven. That would allow the testing of techniques like the proposed (dynamic model-based control) but also learning techniques that extract the model of the plant under consideration. On the other hand, a first insight into the use of Cartesian impedance control shows its adequacy to control the robot-environment interaction, especially in those applications where the environment is completely or partially unknown and where a major concern is the robot compliance rather than controlling the exact forces exerted into the environment.

5. OUTLOOK

Once the theoretical aspects have been verified in a simulated environment and the results have shown the validity of the approach, some experiments are to be performed on a real platform. An experimental setup is being prepared where a Mitsubishi PA-10, a seven degrees of freedom industrial robot arm, is used to test the impedance control in a real scenario. The robot will make use of force sensors on its wrist to sense the contact forces and adapt its grasp to the object being manipulated.

REFERENCES

[1] N. Hogan, Impedance control-An approach to manipulation, *Journal of Dynamics Systems, Measurement,*

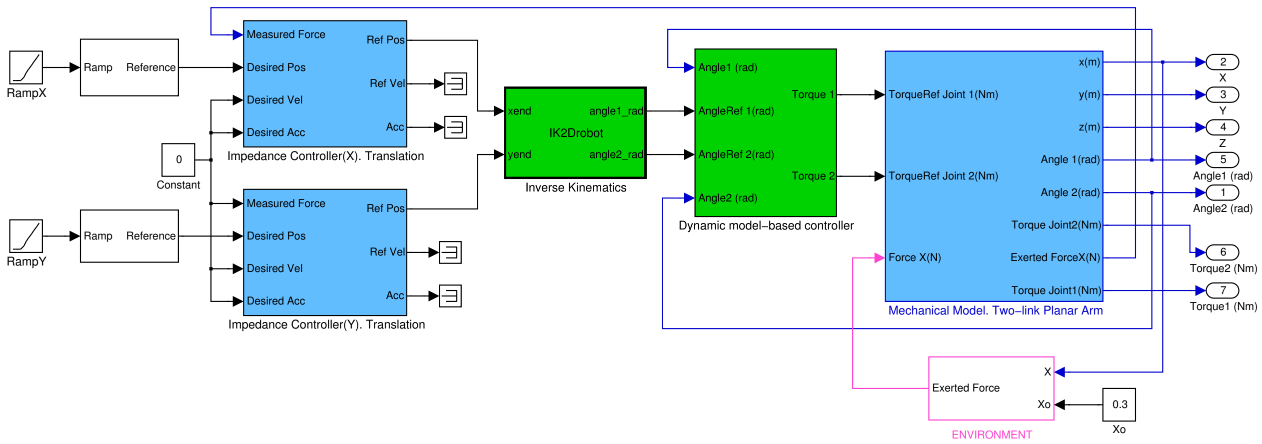


Fig. 7. Complete control system modelled with MATLAB/Simulink

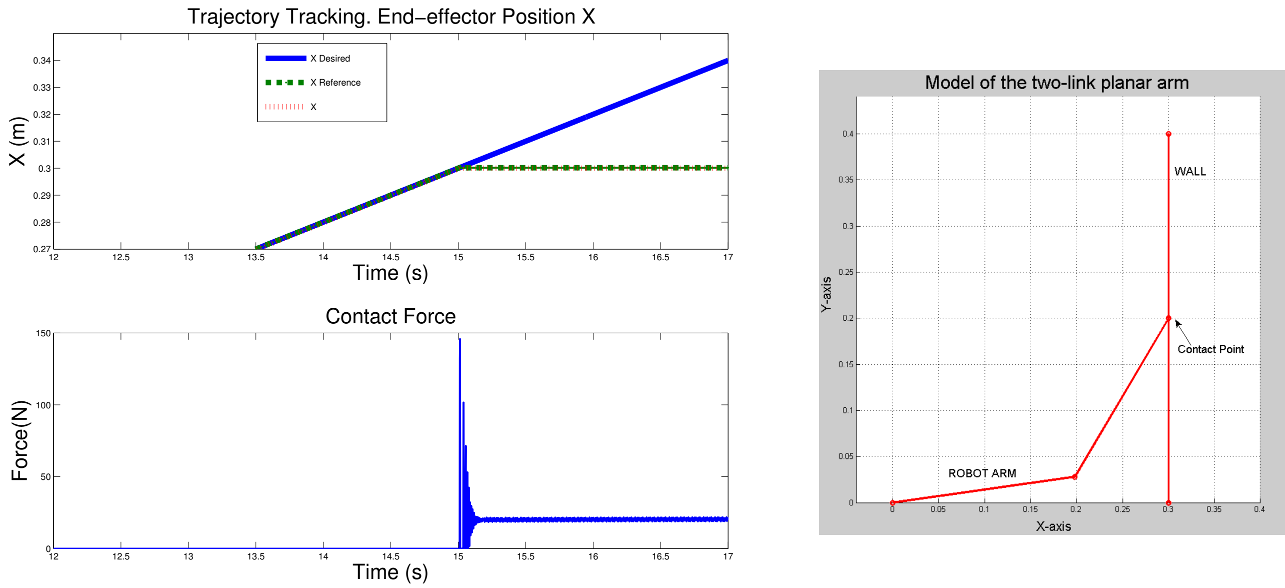


Fig. 8. Response using Impedance Controller *top left:* desired, reference and current X Position *bottom left*: contact force *right*: simulated mechanical model

and Control-Transactions of the ASME, Vol. 107, 1985, pp. 8-16

[2] D.E. Whitney, Historical perspective and the state-of-the art in robot force control, *International Journal of Robotics Research.* Vol. 6 No. 1, 1987, pp. 3-14

[3] J. De Schutter, H. Bruyninckx, W. H. Zhu, M. W. Spong, Force control: A bird's eye view, *In Siciliano and Valavanis (Eds.), Control problems in robotics and automation.* London, 1998, Springer

[4] J. De Schutter, H. Van Brussel, Compliant Robot Motion. A Control Approach Based on External Control Loops, *Int. Journal of Robotics Research*, vol. 7, No. 4, 1998, pp. 18-33

[5] R.P. Paul, B. Shimano, Compliance and Control, *Proceedings of the 1976 Joint Automatic Control Conference*, 1976, pp. 694-699.

[6] J.K. Salisbury, Active Stiffness Control of a Manipulator in Cartesian Coordinates, *IEEE Conference on Decision and Control*, 1980

[7] F.L. Lewis, D.M. Dawson, and C.T. Abdallah, Robot manipulator control: theory and practice, New York: Marcel Dekker, 2004

[8] J.J. Craig, *Introduction to Robotics Mechanics and Control*, Addison-Wesley, 1989

[9] S. Chiaverini, B. Siciliano, L. Villani, A survey of robot interaction control schemes with experimental comparison, *in IEEE/ASME Transactions on Mechatronics*, vol.4, pp.273-285, 1999

[10] A.M. Lopes, F. Almeida, Acceleration based force-impedance control, *in Proc. of IASTED Int.Conf. Modelling, Identification and Control*, pp.73-81, 2006

[11] F. Caccavale, C. Natale, B. Siciliano, L. Villani, Integration for the next generation - Embedding force control into industrial robots, *in IEEE Robotics and Automation Magazine*, pp.53-64, September 2005