Impedance control of 2dof serial robot manipulator

A. ALSHAWI[1] and S. KİZİR[1]

[1] Kocaeli University, Kocaeli/Turkey, ali-sh90@hotmail.com
[1]Kocaeli University, Kocaeli/Turkey, selcuk.kizir@kocaeli.edu.tr

***Abstract*-This paper shows the implementation of an impedance control method to regulate the interaction forces between a robotic arm and the environment, when there is connect between them. A complete description of the system to model and control both an RR robot arm and its interaction with the environment is simulated and detailed by Matlab/Simulink; from the generation of a mechanical model in SimMechanics (Matlab) after export its 3D drawing. The description and setting of a dynamic model based on computed torque control, to cancel out the nonlinearities existing on the dynamic model of the robot. It is based on feedback linearization and computes the required arm torques using the nonlinear feedback control law, and lastly control of the reaction forces is done using the impedance control method after modeling the environment. This type of control modifies the dynamic behavior of the robot when there is contacting with the environment and it is widely used in industrial robots.**

***Keywords*- Impedance control, computed torque control, Robotic manipulator, Interaction, Simulation, Modelling.**

I. INTRODUCTION

APPLIYNG robot manipulators to a wider class of tasks, that will be necessary to control not only the position of a manipulator but also the force exerted by the end-effector to the environment. While many of the tasks performed by the robot manipulator r, that the robot interacts with its environment such as surgeries, pushing, touching, carrying sensitive objects and cutting, etc. The implementation of all these tasks require that the robot, besides realizing the expected position, provide the necessary force either to overcome resistance from the environment, or comply with the environment. Therefore, the robot manipulator to interact and contact safely and friendly to humans or to objects in unknown environments, it is necessary to include an interaction control method that adapts the forces exerted on the environment in order to avoid damages the manipulator and the environment. A force control methods can be used on those applications where the desired force or maximum force to exert is known in advance. However, in other case indirect force control methods can be used. That these methods try to make the manipulator compliant with the object connecting to, without any problem to control maximum or desired force. In the control loop, the position control loop takes the main role but when there is, an interaction is also being controlled for getting safe and clear contact. Impedance control [1] is one of these indirect force control methods. The main philosophy of impedance control, according to Hogan [2], is that the manipulator control system should be designed not to track a motion trajectory alone, but rather to regulate the mechanical impedance of the manipulator. Therefore, when the robot manipulator contacting the environment, impedance control aims to control the dynamic behavior of robot by controlling the properties of contact, that means controlling the damping and stiffness of the interaction. Similar to our work in this study, we find that the impedance control method is used in different fields, such as robots are able to move and carry sensitive things without any damage to them, industrial robots [3] used in vehicle assembly operations or surgical robots like Da Vinci surgical robot [4].

This paper presents a case study where impedance control is used to control the interaction forces of a simulated 2dof serial robot manipulator. In order to achieve this goal, the mechanical model of the robot will be created using SimMechanics after export its 3D drawing. A computed torque [5], [6] controller will be used to cancel out the nonlinearities present on the robot's dynamic model, that the mechanical system described in SimMechanics. It is depends on feedback linearization and computes the required arm torques using the nonlinear feedback control law. Finally, after the environment is modelled, a Cartesian impedance controller [7], [8] is used to control the forces exerted for the robot. To test the performance of the control system, the robot is given a desired trajectory as input that causes the robot to hit a wall, in order to observe the effectiveness of the controller and performance of the robot environment interaction.

II. MECHANICAL AND DYNAMIC MODEL

The first step in this paper is to create a model of a 2-dof robot arm to test our approach. Figure 1 shows the mechanical model of the 2-dof robot arm drew with 3D program (SolidWorks) and Figure 2 shows the mechanical model of the 2-dof robot arm described with SimMechanics after export its 3D drawing. The model is consisted of fixed part, two bodies tow and revolute joints. The module receives torques as input and the outputs are end-effector position in Cartesian Coordinates, joint angles and torques.

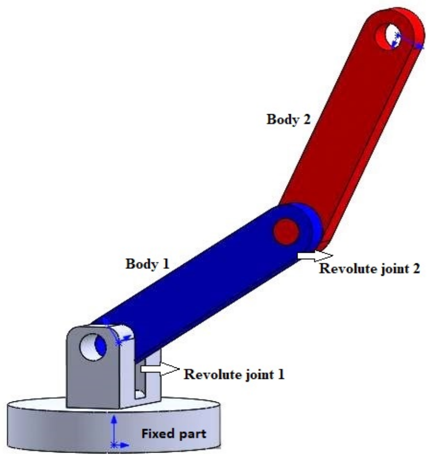


Figure 1: Model of 2-dof robot arm in SolidWorks.

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 linearize the system and apply well-known linear control techniques. The second option will be used here like many of the studies in literature. The next section will describe the dynamical model of the mechanical manipulator.

The dynamics of a robot arm is explicitly derived based on the Lagrange-Euler formulation to explain the problems involved in dynamic modelling. That the dynamic model relates the forces acting on the mechanical structure with the resulting displacements, velocities, and accelerations. Where these forces can take different sources like, the torques generated by the motors, the friction forces, the inertia of the mechanical links, and the possible forces exerted from the environment on the robot. The mechanism of the arm was treated as a combination of open Kinematic-chains [9]. Through the formulation by Denavit Hartenberg convention, the forward kinematics of the system is derived. Then by using the Lagrange equation, we get the dynamics of the model.

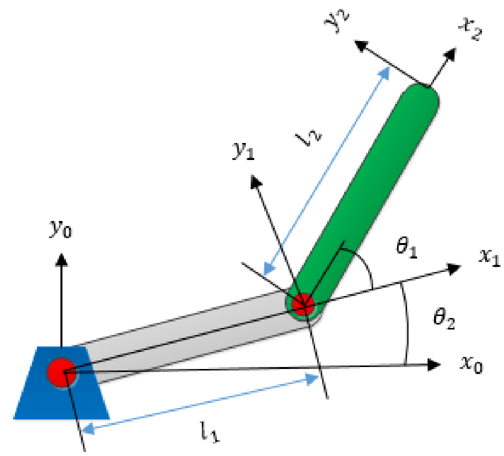


Figure 3: Coordinate systems of 2dof Robot manipulator.

The Lagrangian of a dynamic system is the difference between the kinetic and potential energy at an arbitrary instant [10]. The dynamic equations for the robot manipulator are usually represented by the coupled non-linear differential equations which was derived from lagrangian method

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

where θ is the joint variable and u is the vector of generalized forces acting on the robot manipulator. $M(\theta)$ n×n is the inertia matrix of manipulator. $C(\theta,\dot{\theta})$ n×1 is the vector of centripetal and Coriolis, and $G(\theta)$ n×1 is the gravity vector. Unlike the real robot manipulator the friction torques in our model will not be exist as we see in equation (1). Figure3 shows the Coordinate systems and details of 2-DOF robotic arm. The dynamic model following lagrangian formulation is:

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

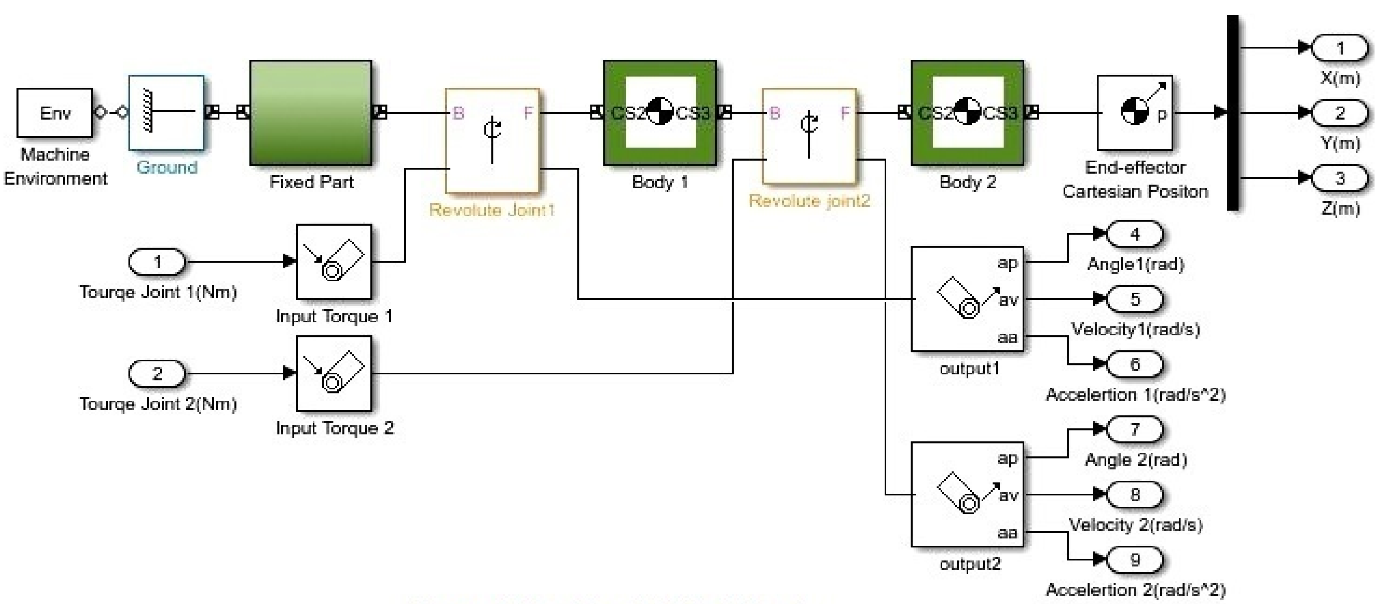


Figure2 : Mechanical model in SimMechanics

Where

$$C(\theta,\dot{\theta}) = \begin{bmatrix} -m_2 l_1 l_2 (2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_1^{\ 2})\sin\theta_2 \\ m_2 l_1 l_2 \dot{\theta}_1^{\ 2}\sin\theta_2 \end{bmatrix} \quad (3)$$

$$G(\theta) = \begin{bmatrix} (m_1+m_2) g l_1 \cos\theta_1 + m_2 g l_2 \cos(\theta_1+\theta_2) \\ m_2 g l_2 \cos(\theta_1+\theta_2) \end{bmatrix} \quad (4)$$

$$M(\theta) = \begin{bmatrix} (m_1+m_2) l_1^{\ 2} + m_2 l_2^{\ 2} + 2 m_2 l_1 l_2 \cos\theta_2 & m_2 l_2^{\ 2} + m_2 l_1 l_2 \cos\theta_2 \\ m_2 l_2^{\ 2} + m_2 l_1 l_2 \cos\theta_2 & m_2 l_2^{\ 2} \end{bmatrix} \quad (5)$$

The terms l_1 and l_2 are the lengths of link1and 2, respectively and m_1, m_2 their masses. In this work $l_1 = l_2 = 0.16m$ and $m_1 = m_2 = 1kg$.

III. Position Control

In this paper, a Computed torque controller will be used to control the position of the robot end-effector. The computed torque control is an effective motion control strategy for robotic manipulator systems, which the Computed torque controller is a significant nonlinear controller to certain systems, which it is depend on feedback linearization and computes the required arm torques using the nonlinear feedback control law. Let us describe now the technique to linearize and decouple the system. Figure 4 shows configuration of computed torque scheme.

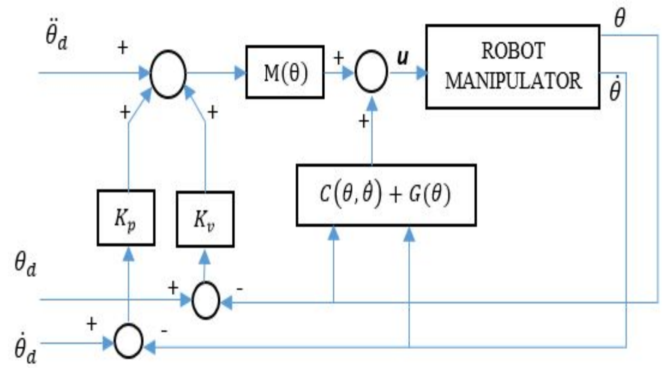


Figure 4: Conventional computed torque control scheme

The control law in (6) represents the standard structure of the compute torque control strategy.

$$u = M(\theta)\left[\ddot{\theta}_d + K_p\,\dot{e} + K_v\,e\right] + C(\theta,\dot{\theta}) + G(\theta) \quad (6)$$

where $\ddot{\theta}_d$, $\dot{\theta}_d$ and θ_d are re the vectors n×1 of desired acceleration, velocity, and position, respectively. The joint position error is denoted by the vector $e = \theta_d - \theta$while $\dot{e} = \dot{\theta}_d - \dot{\theta}$ the n×1 vector of velocity error. The computed torque control in (6) has two parameters, $K_p$and K_v which are the n×1 Proportional and Derivative gains, respectively.

In most of the industrial manipulators, the possibility to send torque commands directly to the robot is not available. To find solution and 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 or orientation command in Cartesian coordinates. Therefore, by using the Geometric solution approach [11] that is based on decomposing the spatial geometry of the manipulator into several plane geometry problems, the possible solutions for θ_1 and θ_2 angles of robotic arm can be written as in (7) and (8)

$$\theta_2 = \mathrm{Atan2}\left(\mp\sqrt{1-\left[\frac{P_x^2+P_y^2-l_1^2-l_2^2}{2l_1l_2}\right]^2},\ \frac{P_x^2+P_y^2-l_1^2-l_2^2}{2l_1l_2}\right) \quad (7)$$

$$\theta_2 = \mathrm{Atan2}(P_y, P_x) + \mathrm{Atan2}\left(\sqrt{P_y^2+P_x^2-(l_2\cos\theta_2+l_1)^2},\ l_2\cos\theta_2+l_1\right) \quad (8)$$

Where P_y , P_x are the end-effector position, and l_1 and l_2 are the lengths of links 1 and 2 respectively. In this case and in order to be able to obtain the results of this work in an easy and useful manner to robot manipulator, Inverse kinematics of our module (7), (8) has been modeled in Simulink. In this way, the mechanical model of the manipulator, the computed torque controller, and the inverse kinematics module can be integrated into one block, this block represents the industrial robots in general. Then, the next step is to build up outer control loops that will be easily tested on a robot that explained above. In other words, focusing on the impedance controller in terms of design and selection of variables values, and study of the results obtained after application of the impedance controller on the robot rather than on internal joint controllers.

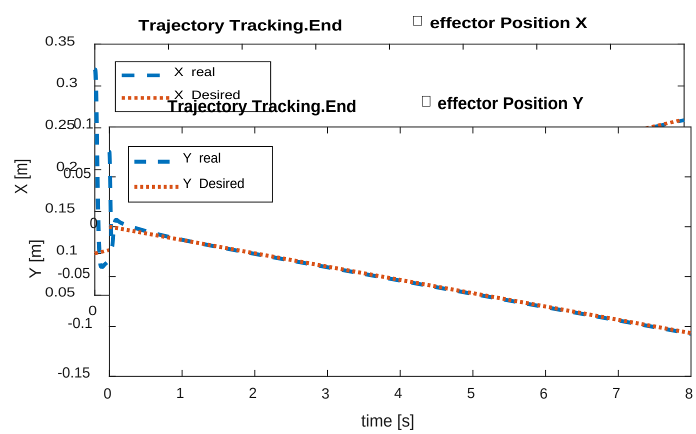


Figure 5: Trajectory tracking performance, top: End-effector Position X, right: End-effector Position Y.

Figure5 shows the trajectory tracking performance of the system under control for a given ramp trajectory as input signal in Cartesian space. The home position of our system when: $\theta_1 = \theta_2 = 0$ is $X = 0.32m$ and $Y = 0.075m$, as seen in figure5. The parameters K_p and K_v gains of computed torque controller were found empirically (K_p= 4000, K_v=2000) but it is not difficult to demonstrate how to choose them depending on the desired dynamical response. After canceling, the non-linarites of the mechanical model and tuned the gain parameters of the computed torque controller. It will be possible to obtain the desired performance for the control system.

IV. THE ENVIRONMENT AND CONTACT FORCES

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 behavior 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. Widely the environment's model is used as a linear spring with a spring constant K_e as in (9).

$$f = K_e(X - X_e) \tag{9}$$

Nevertheless, in our case, we will include a damping coefficient with the spring, as the environment is modelled in (10).

$$f = K_e(X - X_e) + B_e(\dot{X} - \dot{X}_e) \tag{10}$$

where K_e is the stiffness of the environment, B_e is the damping coefficient of the environment, f is the contact force, X_e is the static position of the environment, and X is the end-effector position at the contact point. Figure 6 appears the environment concept, where a manipulator of mass m contacts the environment at position X_e trying to reach the desired end-effector position X_d.

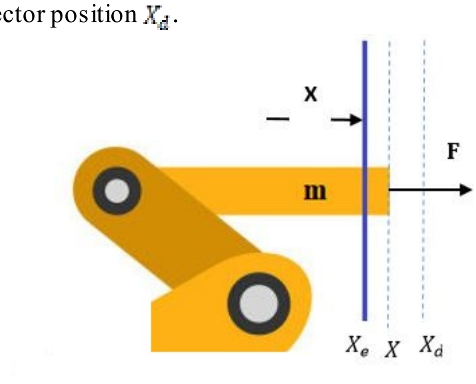


Figure 6: The contact between robot and environment

The dynamic equation of our system was defined in (1). That, u was the vector of generalized forces acting on the robot manipulator, also the external contact force. Therefore, to make our system closer to a real and clearer, we will modify (1) to show the effect of those forces and will represent for them in our model. The dynamic equation controlling the robot's behavior might be defined as in (11).

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta}) + G(\theta) = u - J^T(\theta)f \tag{11}$$

The term $J^T(\theta)f$ translates the task-space forces to the joint. Then, equation (11) will include in Simulink model in order to take account the forces of contact on the dynamic response. As knowing, the relation between forces and torques is defined as in (12).

$$\tau_c = J^T(\theta)f \tag{12}$$

In this work, the case is 2dof robot manipulator, so the contact force can be written as in (13), (14).

$$\tau_{1c} = J_{11} f_X + J_{21} f_Y \tag{13}$$

$$\tau_{2c} = J_{12} f_X + J_{22} f_Y \tag{14}$$

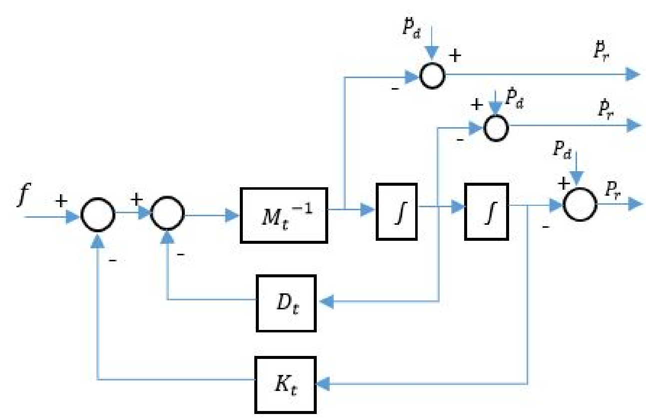
where τ_c are the contact torques, Jij are the elements of the transpose of the 2×2 Jacobian matrix, f_X and f_Y are the forces over the X and Y axes.

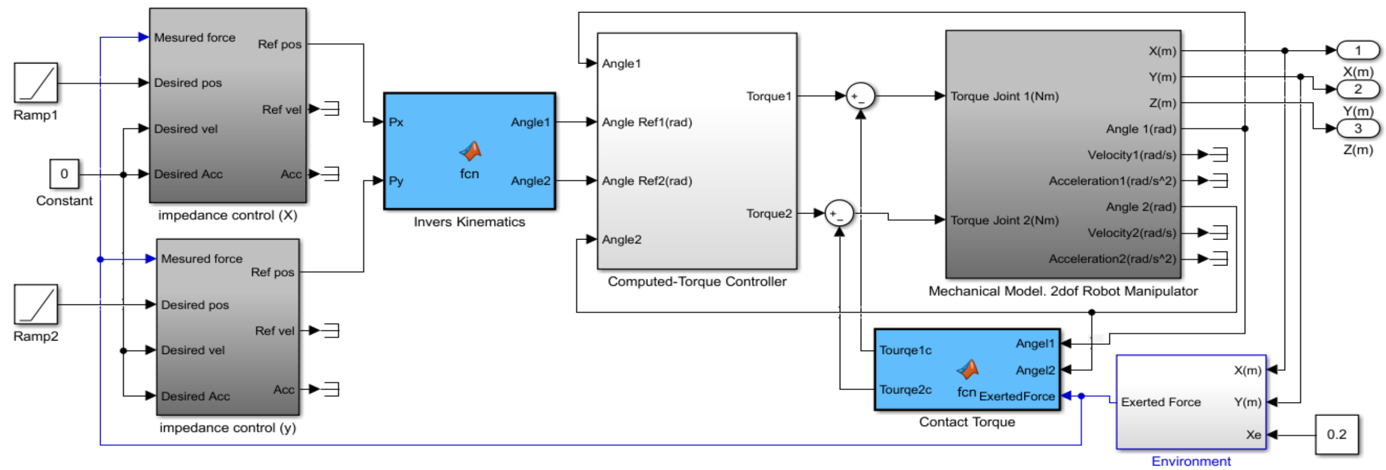
V. IMPEDANCE CONTROLLER

Finally, after we have modelled and controlled a 2dof robot manipulator and designed a simple model for the environment, the next step is to design a controller, which regulates the interaction when the robot contacts the environment. In the current situation, if suppose the robot follows a trajectory and suddenly an object or obstacle shows on its way, the robot will collide with it, trying to reach the final end position of the given trajectory, and exerting such a huge forces into the environment that would likely cause damages to a real robot or to both the object of collision and robot. To solve this situation and avoid it, an impedance controller will be designed. The impedance controller input is the desired trajectory at each time step, and the measured contact forces will be included on the controller for get a quick feedback of the contact state. The output of the controller will be a modified trajectory. That means if the forces are not sensed, the trajectory will be followed accurately. Otherwise, when forces are sensed, the trajectory will be modified in order to regulate the maximum forces. Equation (15) gives the control law of the impedance controller

Figure 7: Impedance controller structure.

Figure 8: Complete control system modelled in MATLAB/Simulink





$$M_t\ddot{e}_t + D_t\dot{e}_t + K_t e_t = f \qquad (15)$$

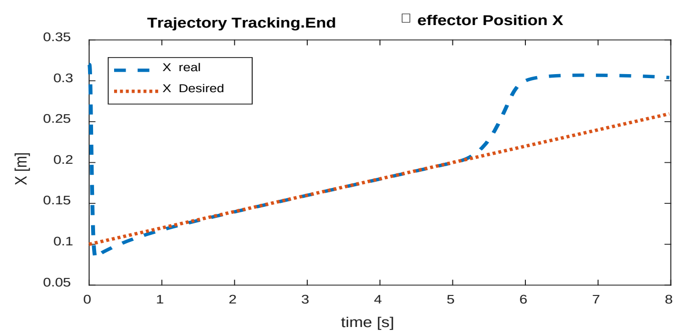
where M_t, D_t and K_t are the inertia, damping and the stiffness coefficients, respectively, e_t is the trajectory error.That, the error defined as $e_t = P_d - P_r$, where P_d is the desired input trajectory and P_r will be the modified trajectory, output of the impedance controller and the input for the inverse kinematics module (P_X,P_Y). Figure7 shows the structure of the impedance controller. The controller parameters have been optimized. $M_t = 2kg$, $D_t = 2\,Ns/m$, $K_t = 90\,N/m$.

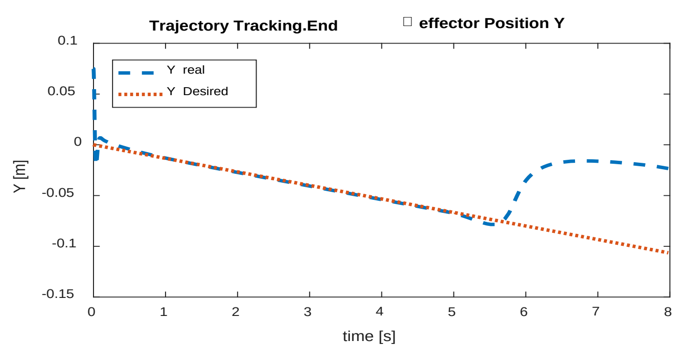
Figure8 shows the complete simulation model of the control system in this paper, after designing and modeling the all components of them as discussed above. It contains the mechanical model, the computed torque controller, the invers kinematics of robot manipulator, the model of the environment, the impedance controller and the desired input trajectories.

VI. SIMULATION RESULTS

In order to monitor and test the performance of the control sys-tem, the robot is given a desired input trajectory to follow. That, the X-axis input is a ramp with slop = 0.3/15 starting at *X=0.1*, and the Y-axis input is a ramp with slop = -0.2/15 starting at *Y=0*. In this experiment we put a wall modelled as defined in (11) like a barrier at X=0.2 with K_e=2000*N/m* and B_e =0.1*Ns/m*. Our goal from this experience, when the robot contacts to the surface of the wall, the robot remains at the point of contact as long as the wall exist, either in the case of removal, the robot follows the trajectory given to him to reach the target.

In Figure9, we can see the results of the experiment without using the Impedance controller. The manipulator tries to follow the given desired input trajectory and reaches to the target point, after contacting the wall surface, the contact forces increase exponentially as the robot travels "inside" the wall, and robot position changes because of a term $J^T(\theta)f$ as seen in (11) .In Figure10, we can see the results of the same experiment with using the impedance controller, that at the contact point (X=0.2), the contact force increases dramatically and the impedance controller reacts and works to redefine a new position trajectory.





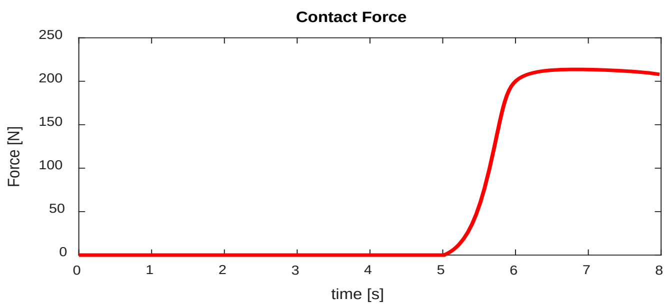
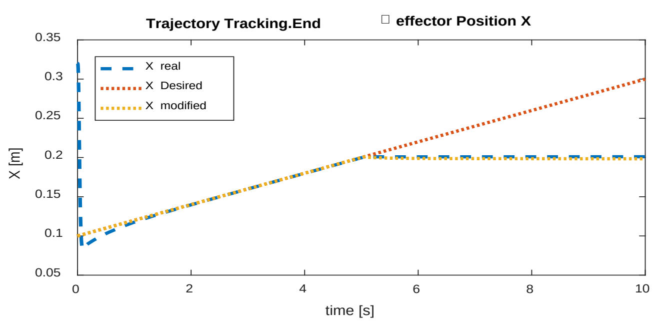


Figure 9: Response without Impedance Controller. top: End effector Position X and Y, bottom: Contact Force.

This new trajectory will be the reference position to value of contact point. The new trajectory regulates forces to a value of around $2N$. X and Y real signals are end-effector position, X and Y Desired signals are desired input trajectory, X and Y modified signals are impedance controller outputs.



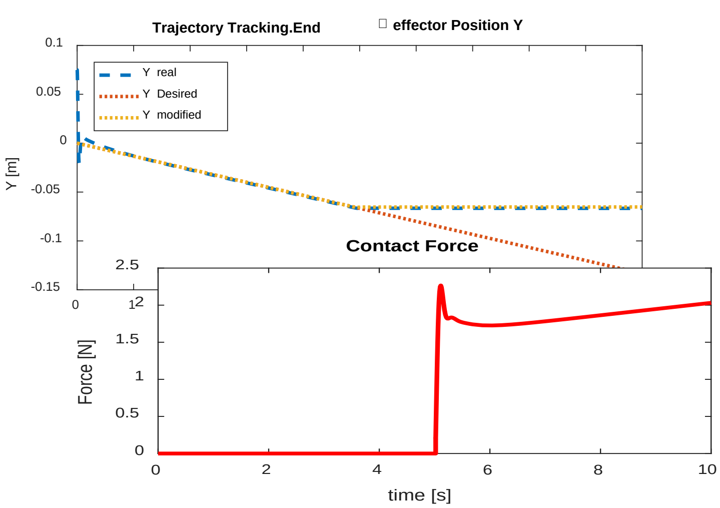


Figure 10: Response with Impedance Controller. top: End-effector Position X and Y, bottom: Contact Force.

VII. CONCLUSION

In this paper, the steps of modeling and simulation a 2dof robot manipulator, and the interaction between the robot and the environment was explained. The 2dof robot manipulator was modeled by using SimMechanics after export it's 3D drawing. That, the mechanics model of the robot gives a great potential ability to verify the control algorithms applied to the model. Then, a computed torque controller was applied and used to control the position of the robot end-effector. Therefore, an inverse kinematics was used to send direct joint angle commands to the robot or an end-effector position command in Cartesian coordinates. On the other hand, impedance controller was developed and applied to the system. When the robot contact with the environment that was modeled as (spring and damper), the contact forces increase exponentially and changes the robot position. By using the impedance controller, the impedance controller reacts and works to redefine a new position trajectory. This new trajectory regulates forces to a value of around2N. The parameters of the impedance controller were optimized. The use of impedance controller shows the ability to control the interaction between the environment and the robot, especially in those applications like Human-interaction robots and surgical robots.

REFERENCES

[1] Hogan N. (1984). *Impedance Control an Approach to Manipulation*, American Control Conference, pp. 304 – 314.

[2] Hogan N. (1985). *Impedance Control, an Approach to Manipulation*: Part I, II, Int. Journal of Robotics Res. 107, 1–24.

[3] Lange, Friedrich, Wieland Bertleff, and Michael Suppa. *"Force and trajectory control of industrial robots in stiff contact." Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013.*

[4] Haidegger, Tamás, et al. "Force sensing and force control for surgical robots." IFAC Proceedings Volumes 42.12 (2009): 401-406.

[5] Middle tone, R. H., and G. C. Goodwin. "Adaptive computed torque control for rigid link manipulators." *Decision and Control, 1986 25th IEEE Conference on.* Vol. 25. IEEE, 1986.

[6] Gilbert, Elmer G., and In Joong Ha. "An approach to nonlinear feedback control with applications to robotics." *IEEE transactions on systems, man, and cybernetics* 6 (1984): 879-884

[7] Chiaverini, Stefano, Bruno Siciliano, and Luigi Villani. "A survey of robot interaction control schemes with experimental comparison." IEEE/ASME Transactions on mechatronics 4.3 (1999): 273-285.

[8] Caccavale, Fabrizio, et al. "Integration for the next generation: embedding force control into industrial robots." IEEE Robotics & Automation Magazine 12.3 (2005): 53-64.

[9] Xie, Ming. *Fundamentals of robotics: linking perception to action*. Vol. 54. World Scientific Publishing Company, 2003.

[10] Jazar, Reza N. *Theory of applied robotics: kinematics, dynamics, and control*. Springer Science & Business Media, 2010.

[11] *Welman, Chris. Inverse kinematics and geometric constraints for articulated figure manipulation. Simon Fraser University, 1994.*