Modeling and Simulation of 6DOF Robot Manipulators with Tactile Position-Force Control

Mohammad Reza Najafi *
Department of Mechanical Engineering,
Imam Hossein comprehensive University, Tehran, Iran
E-mail: Drmrnajafi@ihu.ac.ir
*Corresponding author

Ahmad Reza Khoogar
Faculty of Material and Manufacturing Technologies,
Malek Ashtar University of Technology, Tehran, Iran

Hadi Darabi
Department of Mechanical Engineering,
Imam Hossein comprehensive University, Tehran, Iran

Received: 19 December 2020, Revised: 2 April 2021, Accepted: 9 April 2021

Abstract: In this paper, a joint position-force controller is used to control a 6R general-purpose robot manipulator. The manipulator comes into interaction with a spherical object in a numerically simulated environment. A controller has been implemented using the MATLAB Simulink software which uses the Simmechanics second-generation toolbox. A useful numerical contact model is used for modelling the interaction between the manipulator's end-effector and the environment which generates the interaction feedback forces. The control algorithm presented in this paper is developed in the Cartesian space and the original control algorithm was modified to satisfy the desired input position in the base coordinate frame. The control algorithm was verified using a virtual environment, before hardware implementation. The novelty of the controller is determining the input tactile forces for the robot without actually causing a collision between the end-effector and the object in the environment which can lead to fracture and damage to the environment or the manipulator. The modeling process of interaction with the spherical environment was investigated using Simmechanics to model precise mechanical characteristics of manipulator that are unknown to the designers and provide a great advantage in the simulation for them. The considered position and tactile force were tracked successfully with good accuracy. The results show that the proposed manipulator system controls the position and force with more than 95% accuracy and the accuracy of desired tracing trajectory is 99%.

Keywords: Force Control, Multibody Simulation, Simmechanics, Tactile Control

How to cite this paper: Mohammad Reza Najafi, Ahmad Reza Khoogar, and Hadi Darabi, "Modeling and Simulation of 6DOF Robot Manipulators with Tactile Position-Force Control", Int J of Advanced Design and Manufacturing Technology, Vol. 15/No. 1, 2022, pp. 51–63. DOI: 10.30495/ADMT.2021.1918232.1235.

Biographical notes: Mohammad Reza Najafi received his PhD in Mechanical Engineering from Imam Hossein comprehensive University, Tehran, Iran, in 2022. His current research interests include control, dynamic and vibrations. **Ahmad Reza Khoogar** is an associate professor of Malek Ashtar University of technology. His current interest includes control and robotics in faculty of material and manufacturing technologies. He received his PhD from the University of Alabama in United States, in 1989. He also published over 80 papers in the field of control and robotics. He has authored and translated four books in the field of control and robotics. **Hadi Darabi** is a PhD student in Mechanical Engineering at Imam Hossein comprehensive University, Tehran, Iran. He received his MSc in Mechanical Engineering from Malek-e-Ashtar University of Technology, Shahin Shahr, Iran, in 2016. His current research interests include control and robotics.

1 INTRODUCTION

In many applications, manipulators must interact with unknown environments safely and reliably. It is necessary to include an interaction control method that reliably adapts the forces exerted on the environment to avoid damage. The hybrid position/force control method can be used on those applications where the maximum or exerted force is known. It is important to access to precise mechanical characteristics and application of manipulator.

Hybrid position/force control was first proposed by Raibert and Craig [1] and then was distributed by others such as: Gueaieb et al. [2-3]; Kumar et al. [4]; Matsuno and Yamamoto [5]; Queiroz et al. [6]; Xiao et al. [7]; Yoshikawa [8]. In this method, force and position terms are separated and decoupled. Then, separate control laws are used for each term. After that, Zhang and Paul Zhang and Paul [9] proposed a Cartesian space scheme for hybrid control. Neha and et.al [10] analysis grasp of a four-finger robotic hand based on the MATLAB Simmechanics in which various bones, joints, tendons, and muscles functioning together in order to produce the desired motion. Pliego-Jiménez and Arteaga-Pérez [11] proposed an adaptive position/force control for robot manipulators in contact with rigid surface with uncertain parameters. Guibin Ding [12] worked on the position/force control problem for constrained reconfigurable manipulator via dynamic model decomposition. Based on a nonlinear transformation, the dynamic model of the reconfigurable manipulator system has been divided into the positioning subsystem and the force subsystem. This way, the control system is more convenient to design. In the positioning subsystem, an adaptive neural network is used to approximate the nonlinear term whose upper bound is unknown in different reconfigurable manipulators. In the force subsystem, based on the computed torque method, another adaptive neural network is employed to compensate for the model uncertainties [12]. Homayounzade and Keshmiri [13] investigated an adaptive position/force designed controller for the case of robot motions gained access with manipulators. Considering the position and velocity of the manipulator, parametric uncertainties were compensated by the mentioned controller although forces at contact positions were not required. Yoichi Hori [14] reported the efficacy of a robot control method without using non-linear compensation but using the inverse dynamics based on the Two-Degrees-Of-Freedom (TDOF) for the robust position. He showed the experimental results of various kinds of robot motion controls using the life-size 6-axis manipulator and DSP based controllers [14]. Haptic communication is one of the most important cases of study between different structures. The deterioration of haptic performance is dependent on an interference term. Also, a novel hybrid controller for decoupling of response and analyzing its performance, stability, and robustness have been proposed. Simulations and experiments toward cardiac surgery are shown, and the effectiveness of the proposed method is verified. W.G. Guo et al. studied the control method for hybrid position/force manipulators. In this method, by using the disturbance observer in joint space, the dynamics of manipulators can be linearized [15].

This paper presents a case study that describes hybrid position/force control of the interaction forces for simulated 6DOF Serial Manipulators and thus, Simmechanics (second generation) is utilized to model the robot. A model-based controller is used in which the mathematical model of the mechanical system is to decoupled and linearize the force and position terms. If the mathematical description matches the mechanical system described in Simmechanics, the non-linearity of the system will be nullified. As long as the system is linear, simple control laws can be used to implement the force and position control. To test the control system, the robot is given a reference input trajectory that causes the robot to track a trajectory on the surface of a sphere, to observe the environmental interaction performance. In this trajectory, the robot passes through a singular position and the stability of the system in singular points can be studied. Some of the programs for implementing the controller are obtained from the MATLAB Robotic toolbox, after little changes. The performance of the system was verified using a virtual environment and the results are shown and discussed. Determination of the input tactile forces of manipulator without any collision between the end-effector and the environment is desired for designers that can inhibit fracture and damage to the environment or the manipulator presumably. The main outstanding of the present work are the estimation of the feedback force and obtaining the manipulator controller design without the need to manufacture the desired robot and workspace. Also, the initial contact of the end-effector on the environment, which sometimes causes damage or fracture on the surface of the impact or tool, is also controlled.

2 SYSTEM DESCRIPTION

Manipulator structure chosen for study in this paper is demonstrated in "Fig. 1" . The manipulator has six revolute joints. Axes of three last adjacent joints intersect each other on a point. In robotics, the manipulator Denavit and Hartenberg (MDH) notation for describing serial-link mechanism geometry is fundamental. Given such a description of a manipulator, we can make use of established algorithmic techniques to find kinematics, Jacobians, dynamics and other motion planning of the manipulator and simulation [16].

Table 1 shows the MDH parameter of the manipulator. In this work, tactile contact forces between the end-effector and the environment are generated by numerical simulation of the contact environment dynamically and this is done without causing any actual contact or collision between the manipulator and the environment which can cause damage.

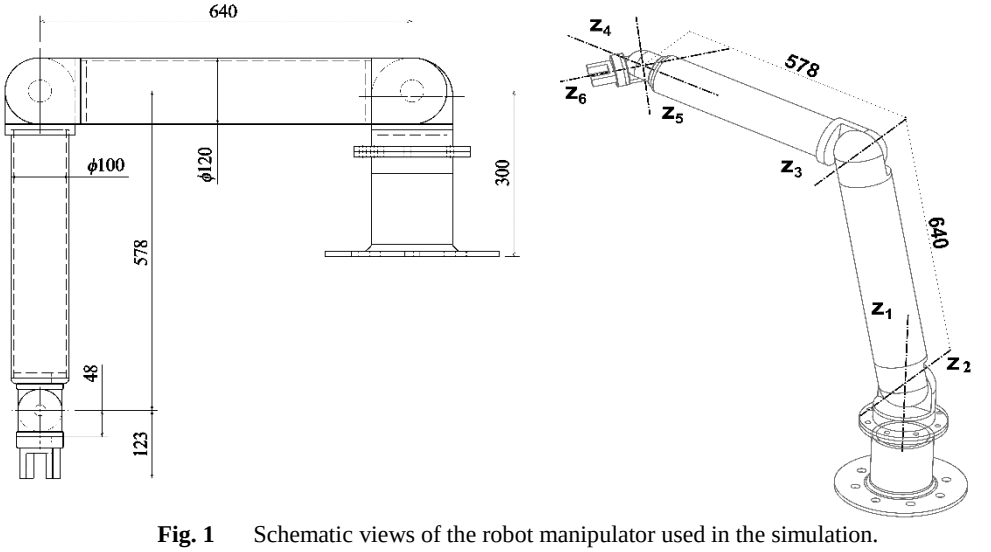

Fig. 1 Schematic views of the robot manipulator used in the simulation.

Table 1 Modified Denavit Hartenberg parameters of the manipulator

i	a_{i-1} [mm]	$\alpha_{i-1}[deg]$	d_i [mm]	θ_i [deg]
1	0	0	0	θ_1
2	0	90	0	θ_2
3	640	0	0	θ_3
4	0	90	578	θ_4
5	0	90	0	θ_5
6	0	90	0	θ_6

2.1. Direct kinematics

Direct kinematic of the manipulator can be written with six consecutive transformations such as the one shown in (1). Each of the six transformations are extracted from (2). In Equation (2), four-link parameters exist which are extracted from "Table 1" . In Equation (2) $c\theta_i$ stands for $\cos(\theta_i)$, and $s\theta_i$ stands for $\sin(\theta_i)$.

$$ {}^{0}_{6}\mathrm{T} = {}^{0}_{1}\mathrm{T}\ {}^{1}_{2}\mathrm{T}\ {}^{2}_{3}\mathrm{T}\ {}^{3}_{4}\mathrm{T}\ {}^{4}_{5}\mathrm{T}\ {}^{5}_{6}\mathrm{T} \tag{1} $$

By means of Equation (1) kinematic equation of the manipulator can be derived completely.

$$ {}^{i-1}_{i}\mathrm{T} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i\, c\alpha_{i-1} & c\theta_i\, c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i\, s\alpha_{i-1} & c\theta_i\, s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

2.2. Jacobin Matrix

Jacobin matrix defines relationship between the velocity of the end effector and the joint velocities, as stated in (3).

$$ \begin{bmatrix} v \\ \omega \end{bmatrix} = J\dot{q} \tag{3} $$

In which, v and ω are linear and angular velocity of end effector respectively and q is a 6×1 vector representing joint variables. This Jacobin matrix also defines the relation between the task-space forces acting on the end-effector and torques in the joint-space.

$$ \tau = J^{T}(\mathrm{q})\mathrm{F} \tag{4} $$

Equation (5) shows the non-zero elements of the Jacobin matrix for the given manipulator. Another simple method for driving the Jacobin matrix is by using MATLAB's robotic toolbox which can automatically generate the Jacobin matrix in the {0} or the {n} coordinate system. Figure 2 shows the developed library with RTB for the given manipulator. For example, The T0_1 block in this figure is used for computing coordinate transformation from the {1} frame to the {0} frame. The Jacobean matrix, inertia matrix, gravity and Coriolis terms, can be calculated as well.

Point masses are used in the dynamic formulation of the manipulator and this formulation is then used in a dual position-force controller for on-line computations. The desired position is represented as a line rout on the

surface of a physical solid environment. The results are used as a reference for performance evaluation of the simulator and the control system. In some situations, deteriorative signals may be generated due to the Jacobean matrix becoming zero or values close to zero. In these situations, the pseudo-inverse method is used to avoid division by zero and the control signal becoming too large.

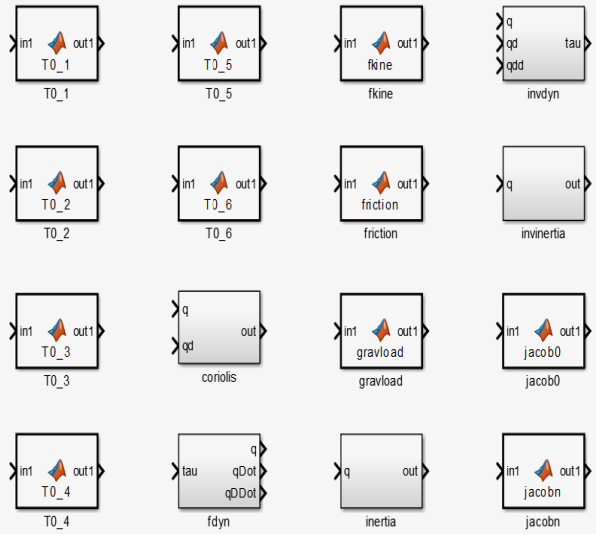

Fig. 2 Library generated with the MATLAB robotics toolbox.

$$
\begin{aligned}
&\mathbf{1.1}: -\mathbf{d_4c_2s_1s_3} + \mathbf{c_3s_1s_2} - \mathbf{a_2c_2s_1}\\
&\mathbf{1.2}: -\mathbf{d_4c_1s_2s_3} - \mathbf{c_1c_2c_3} - \mathbf{a_2c_1s_2}\\
&\mathbf{1.3}: -\mathbf{d_4c_1s_2s_3} - \mathbf{c_1c_2c_3}\\
&\mathbf{2.1}: \mathbf{d_4c_1c_2s_3} + \mathbf{c_1c_3s_2} + \mathbf{a_2c_1c_2}\\
&\mathbf{2.2}: -\mathbf{d_4s_1s_2s_3} - \mathbf{c_2c_3s_1} - \mathbf{a_2s_1s_2}\\
&\mathbf{2.3}: -\mathbf{d_4s_1s_2s_3} - \mathbf{c_2c_3s_1}\\
&\mathbf{3.2}: \mathbf{d_4c_2s_3} + \mathbf{c_3s_2} + \mathbf{a_2c_2}\\
&\mathbf{3.3}: \mathbf{d_4c_2s_3} + \mathbf{c_3s_2}\\
&\mathbf{4.2}: \mathbf{s_1}\\
&\mathbf{4.3}: \mathbf{s_1}\\
&\mathbf{4.4}: \mathbf{c_1c_2s_3} + \mathbf{c_1c_3s_2}\\
&\mathbf{4.5}: -\mathbf{c_4s_1} - \mathbf{s_4c_1s_2s_3} - \mathbf{c_1c_2c_3}\\
&\mathbf{4.6}: \mathbf{s_5s_1s_4} - \mathbf{c_4c_1s_2s_3} - \mathbf{c_1c_2c_3} -\\
&\mathbf{c_5c_1c_2s_3} + \mathbf{c_1c_3s_2}\\
&\mathbf{5.2}: \mathbf{0}\\
&\mathbf{5.3}: \mathbf{0}\\
&\mathbf{5.4}: \mathbf{c_2s_1s_3} + \mathbf{c_3s_1s_2}\\
&\mathbf{5.5}: \mathbf{c_1c_4} - \mathbf{s_4s_1s_2s_3} - \mathbf{c_2c_3s_1}\\
&\mathbf{5.6}: -\mathbf{s_5c_1s_4} + \mathbf{c_4s_1s_2s_3} - \mathbf{c_2c_3s_1} -\\
&\mathbf{c_5c_2s_1s_3} + \mathbf{c_3s_1s_2}\\
&\mathbf{6.1}: \mathbf{1}\\
&\mathbf{6.4}: \mathbf{s_2s_3} - \mathbf{c_2c_3}\\
&\mathbf{6.5}: \mathbf{s_4c_2s_3} + \mathbf{c_3s_2}\\
&\mathbf{6.6}: \mathbf{c_5c_2c_3} - \mathbf{s_2s_3} + \mathbf{c_4s_5c_2s_3} + \mathbf{c_3s_2}
\end{aligned}
\tag{4}
$$

2.3. Derivation of the Manipulator Dynamics

A common way of modelling the dynamics of robot manipulators is by using the recursive Newton-Euler method. Equation (6) shows the general dynamic equation of the manipulator shown in "Fig. 3" .

$$\tau = M(\mathrm{q})\ddot{q} + V(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}} + G(\mathrm{q}) \tag{5}$$

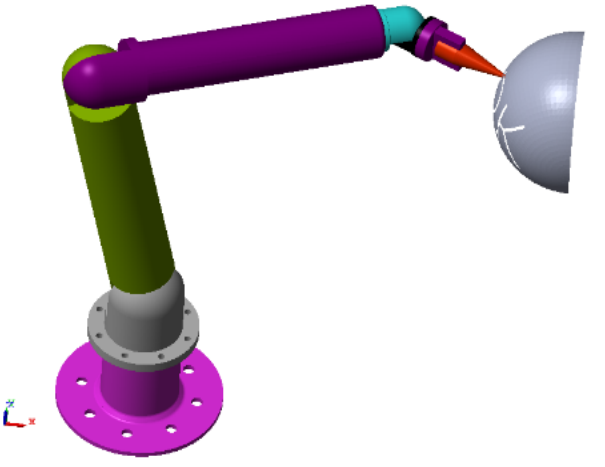

Fig. 3 The manipulator simulated in the Simmechanics.

In which, q is the vector of join parameters, M(q) is the inertia matrix, V(q, q □) is the coefficient matrix of the Coriolis and Centrifugal terms, and G(q) is the vector of gravitational terms and τ is vector of the input joint actuation torques. Note that M(q) is symmetric and positive definite and the actuator dynamics and friction have been neglected. So, the closed-form dynamics equations motion in join space can be extracted by using the MATLAB robotic toolbox.

For Cartesian control of the robot manipulator Equation (3) must be written in Cartesian form:

$$\mathrm{F} = M_X(\mathrm{q})\ddot{x} + V_X(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{x}} + G_X(\mathrm{q}) \tag{6}$$

Equation (4) gives the Cartesian form of the robot manipulator dynamics. The following relations are then used for the calculation of the required torques:

$$M_X(\mathrm{q}) = J^{-T}(q)M(\mathrm{q})J^{-1}(q) \tag{7}$$

$$
\begin{aligned}
V_X(\mathrm{q},\dot{\mathrm{q}}) &= J^{-T}(q)(V(\mathrm{q}, \dot{\mathrm{q}})J^{-1}(q)\\
&+J^{-T}(q)M(\mathrm{q})J^{-1}(q)\dot{J}(q)
\end{aligned}
\tag{8}
$$

$$G_X(\mathrm{q}) = J^{-T}(q)G(\mathrm{q}) \tag{9}$$

3 BACKGROUND OF HYBRID CONTORL

Hybrid position-force control is used for tracking position and force trajectories simultaneously. The goal of this hybrid position-force control is to separate and decouple the position and force control in two separate problems. Then, separate control laws can be used for each problem [17-18]. The general dynamic model used here is:

$$\tau = \mathrm{M}(\mathrm{q})\mathrm{J}^{-1}(\mathrm{q})(\ddot{\mathrm{x}} + \dot{\mathrm{J}}(\mathrm{q})\dot{\mathrm{q}}) + \mathrm{V}(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}} + \mathrm{G}(\mathrm{q}) + \mathrm{F}_{\mathrm{end}} \quad (10)$$

In which, $\mathrm{J}(\mathrm{q})$ is the Jacobin matrix in the task space coordinate frame. Equation (12) was obtained with feedback linearization law:

$$\tau = \mathrm{M}(\mathrm{q})\mathrm{J}^{-1}(\mathrm{q})(\mathrm{a} + \dot{\mathrm{J}}(\mathrm{q})\dot{\mathrm{q}}) + \mathrm{V}(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}} + \mathrm{G}(\mathrm{q}) + \mathrm{F}_{\mathrm{end}} \quad (11)$$

Where, a is a $\mathrm{n} \times 1$ vector that is used to represent position and force control strategies.

$$\ddot{\mathrm{x}} = \mathrm{a} \quad (12)$$

By definition of the tangent space components of x as x_{Ti}:

$$\ddot{\mathrm{x}}_{\mathrm{Ti}} = \mathrm{a}_{\mathrm{Ti}} \quad (13)$$

The tangent space tracking error can be written as:

$$\tilde{\mathrm{x}}_{\mathrm{Ti}} = \mathrm{x}_{\mathrm{Tdi}} - \mathrm{x}_{\mathrm{Ti}} \quad (14)$$

The linear position controller is given as:

$$a_{Ti} = \ddot{x}_{Tdi} + K_{Tvi}\,\dot{\tilde{x}}_{Ti} + K_{Tpi}\,\tilde{x}_{Ti} \quad (15)$$

Where, $K_{Tpi}$and K_{Tvi} are the positive control gains. Substituting (11) into (12) gives the position tracking error system:

$$\ddot{\tilde{x}}_{Tdi} + K_{Tvi}\,\dot{\tilde{x}}_{Ti} + K_{Tpi}\,\tilde{x}_{Ti} = 0 \quad (16)$$

Using the fact that K_{Tpi} and K_{Tvi} are positive in (14), it yields:

$$\lim_{t\to\infty} \tilde{x}_{Ti} = 0 \quad (17)$$

The normal space components of (10) are given as a:

$$\ddot{\mathrm{x}}_{Nj} = a_{Nj} \quad (18)$$

With modeling environment as linear spring:

$$f_{Nj} = K_{ej}\left(x_{Nj} - x_{ej}\right) \quad (19)$$

That K_{ej} is the jth component of environmental stiffness. Taking the second derivative of (17) to time gives the following expression:

$$\ddot{x}_{Nj} = \frac{1}{K_{ej}}\,\ddot{f}_{Nj} \quad (20)$$

Substituting (16) into (18) yields the force dynamics:

$$\frac{1}{K_{ej}}\,\ddot{f}_{Nj} = a_{Nj} \quad (21)$$

The force tracking error in space perpendicular to the interaction surface can be defined as:

$$\tilde{f}_{Nj} = f_{Ndj} - f_{Nj} \quad (22)$$

The linear force controller is then given by:

$$a_{Nj} = \frac{1}{K_{ej}}\left(\ddot{f}_{Nj} + K_{Nvj}\,\dot{\tilde{f}}_{Nj} + K_{Npj}\,\tilde{f}_{Nj}\right) \quad (23)$$

With K_{Nvj} and K_{Npj} being the jth positive control gains. Substituting (12) into (21) gives the tracking error as:

$$\ddot{\tilde{f}}_{Nj} + K_{Nvj}\,\dot{\tilde{f}}_{Nj} + K_{Npj}\,\tilde{f}_{Nj} = 0 \quad (24)$$

Using the fact that K_{Nvj} and K_{Npj} are positive in (22), it yields:

$$\lim_{t\to\infty} \tilde{f}_{Nj} = 0 \quad (25)$$

The hybrid position-force controller block diagram in the MATLAB Simulink environment is obtained in following sections.

4 ERROR CALCULATION

Variant methods have been used for translational and rotational errors. In this paper, we used transformation matrices to calculate the tracking error. The transformation matrices can be redefined using (27).

$$T=\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} n\ o\ a\ p \end{bmatrix} \quad (27)$$

The desired and the actual translation transformations can be compared using (28) as:

$$d = p_{des} - P_{act}. \quad (28)$$

And, rotation transformations can be computed by (29):

$$\Delta R = R_{des} R_{act}{}^{T}. \quad (29)$$

Using Equations (24) and (25), the rotational error can be obtained as:

$$\partial = \frac{1}{2}(n_{act} \times n_{des} + o_{act} \times o_{des} + p_{des} \times p_{des}) \quad (26)$$

Now, the skew-symmetric matrix of angular velocities is defined by (31):

$$\dot{R}R^{T}=\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}=\Omega \quad (27)$$

Angular velocity error can then be calculated as the following difference:

$$\dot{\partial} = \omega_{des} - \omega_{act}. \quad (28)$$

5 CONTACT MODEL

When manipulator contacts with the environment, one degree of freedom in motion lost, since the manipulator cannot move through the environment surface. At this time the manipulator exerts force to the environment's surface. Usually, these constraints are expressed with the task selection matrix and the coordinate frame that defined constraints are frame constraints and are shown with {C}. It is a diagonal matrix that the value of constraining direction is 1 and free direction is equal to zero [19-20]. Figure 4 shows a typical task and the geometry of the tool and environment.

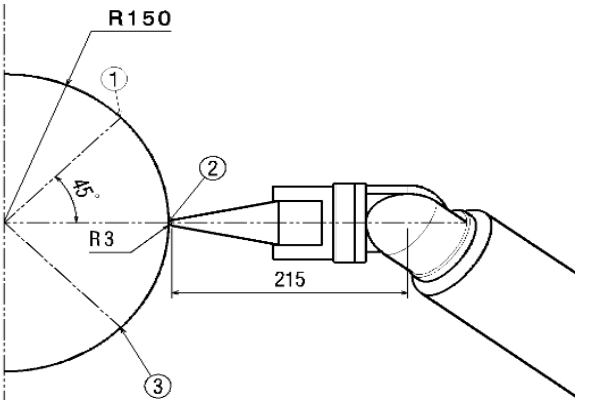

Fig. 4 Typical geometry of contact between end-effector tool and environment.

The task selection matrix is shown with S and defines such as Equation (33).

$$S=\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \quad (29)$$

The position feedback is tracked through the simulated model of the environment. The end-effector makes an interaction with a spherical object in the environment with a certain depth and direction. At the same time, the position controller is also active and pushing the position error to zero. Figure 4 shows the manipulator position trajectory from point 1 to point 3. The total simulation time in this run is three seconds and the step size is 0.01 second. Table 2 describes that the way input data is commanded to the control system over the time with equal steps using the direction cos matrix and the end-effector location in Cartesian coordinates [21-22]. The position controller is used to trace the position trajectory. Terms such as Kv=100 and Kp=2500 have considerable effect on the position controller and its running error. The damping term Kv and Kp are selected by computing the un-damped Natural frequency and the critical damping of the system as shown in the following Equations, [23-24]:

$$\omega_n = \sqrt{2500} = 50 \ \ rad/s \quad (34)$$

$$\tau = \frac{2\pi}{50} = 0.126 sec \quad (35)$$

$$\zeta = \frac{K_v}{2\sqrt{K_p}} = \frac{100}{2\sqrt{2500}} = 1 \quad (36)$$

Table 2 End-effector attitude versus time for a few consecutive steps, the first column is the running time, the next nine columns are the components of the direction cosine matrix of the end-effector coordinates, and the last three columns are the Cartesian coordinates of the end-effector tip point

t	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	x	y	z
1.97	-0.02762	0	0.999618	0.999618	0	0.027624	0	1	0	0.597058	-0.00423	0.6
1.98	-0.01973	0	0.999805	0.999805	0	0.019732	0	1	0	0.59703	-0.00302	0.6
1.99	-0.01184	0	0.99993	0.99993	0	0.01184	0	1	0	0.597011	-0.00181	0.6
2	-0.00395	0	0.999992	0.999992	0	0.003947	0	1	0	0.597001	-0.0006	0.6
2.01	0.003947	0	0.999992	0.999992	0	-0.00395	0	1	0	0.597001	0.000604	0.6
2.02	0.01184	0	0.99993	0.99993	0	-0.01184	0	1	0	0.597011	0.001812	0.6
2.03	0.019732	0	0.999805	0.999805	0	-0.01973	0	1	0	0.59703	0.003019	0.6
2.04	0.027624	0	0.999618	0.999618	0	-0.02762	0	1	0	0.597058	0.004226	0.6
2.05	0.035513	0	0.999369	0.999369	0	-0.03551	0	1	0	0.597097	0.005433	0.6
2.06	0.0434	0	0.999058	0.999058	0	-0.0434	0	1	0	0.597144	0.00664	0.6
2.07	0.051285	0	0.998684	0.998684	0	-0.05128	0	1	0	0.597201	0.007847	0.6

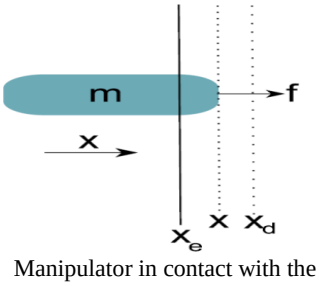

Fig. 5 Manipulator in contact with the virtual environment.

6 MODELING ENVIRONMENT

A model of the environment is included in the system. Linear spring model is used for environment, where f is the contact force, Ke is the stiffness of the environment, x is the end-effector position at the contact point and Xe is the static position of the environment. The force exerted to the environment was modeled as the force of a linear spring exerted normal to the curved surface, as shown in "Fig. 5" that is such a concept, where a manipulator of mass m contacts the environment at position Xe trying to reach the desired end-effector position X_d:

$$f = K_e\left(X - X_e\right) \tag{37}$$

7 IMPLENENTATION

Figure 6 shows the hybrid position-force controller block diagram in the MATLAB Simulink environment. The Robot1 block in "Fig. 6" represents that the manipulator and the contact model was modeled in Simmechanics. Figure 7 shows manipulator modeling and environment in the Simmechanics. Because of using MATLAB Robotic toolbox for creating a set of block libraries for the manipulator, we cannot use traditional translators for generating the Simmechanics model. Coordinate frames in the Simmechanics translator is not identical to the Denavit and Hartenberg notation [25-26]. Figure 7 shows a developed link block model in the Simmechanics. Visualization solids and their corresponding mass properties are defined in the blocks. Block mass can simply be calculated by the block geometrical volume and the corresponding density. In "Fig. 7" , the transform1 block defines the kinematic property of links. And in this work, successive links are connected with the revolute joint. Each joint consists of an input torque signal and two-position and velocity signal [27-28]. Figure 8 illustrates the modeling of this serial 6R robot chains in the Simmechanics simulator. Inputs of the Robot1 block are the joints moments and the outputs are the angular velocities of revolute joints and the contact force obtained from the proposed contact model [28].

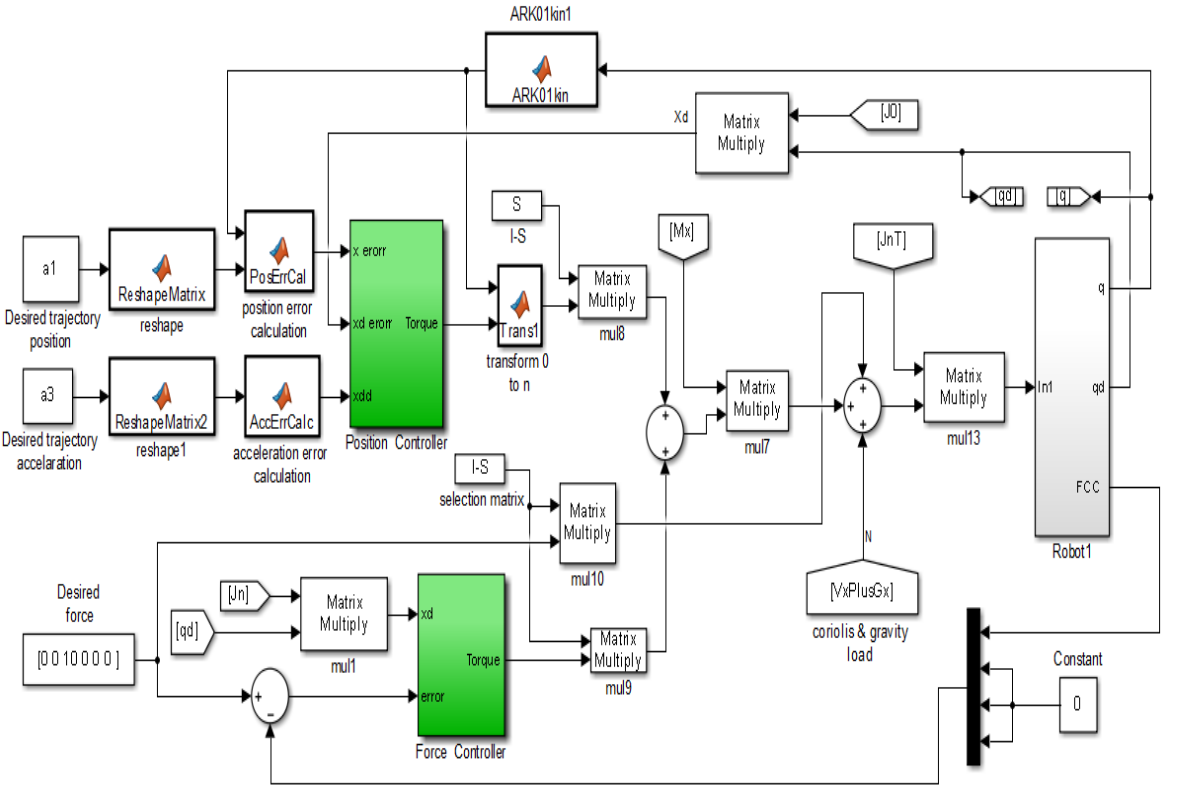

Fig. 6 Hybrid position/force controller block diagram in MATLAB Simulink environment.

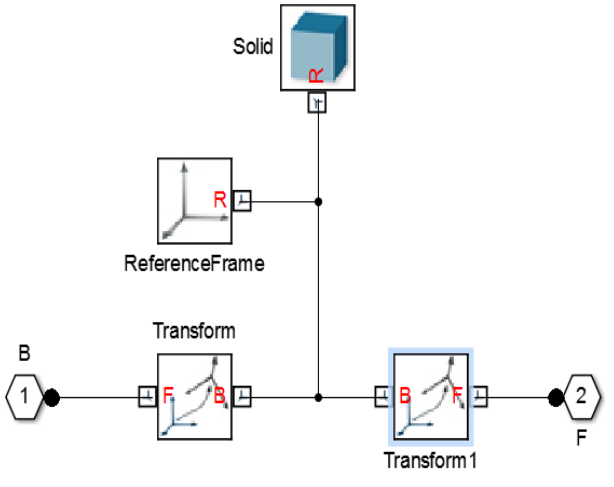

Fig. 7 A developed link model in the Simmechanics.

The position of the end effector in the base coordinate system can be obtained by multiplying ${}^{0}_{6}T$ by joint angles. The velocity of the end effector in the base coordinate system is calculated from Equation (3). Then, position value is compared with the desired position and the error is calculated [29]. Thus, the orientation error is obtained using (30). Control torque (per unit mass) can be calculated with position control law and then transferred to the task space by multiplying with the rotation matrix. After that we extract tangent space moment by task specification matrix (S), on the other side, contact force sensed from the robot1 simulator is transferred to task space and compared with the desired force trajectory. The force control law is then, applied. Figures 9 -10 show the force and the position control laws. Due to the presence of noise in the force sensors, velocity feedbacks were used instead of the velocity derivatives. The control torque arising from the force control law is multiplied with the S' and is then added to the position value obtained from the control signal. i.e., the summed value multiplied with M_x (q) obtained from (8). The Coriolis and gravity load blocks are calculated by Equations (9) and (10). The lower part of " Fig. 11" shows the modeling of the contact. If the distance between the tool and sphere is smaller than the threshold value. In "Fig. 12" the Jacobian matrix, the inertia terms, the gravity load, the Coriolis terms in task space calculated from joints angle, the angular velocity and the outputs are used for implementing the control law.

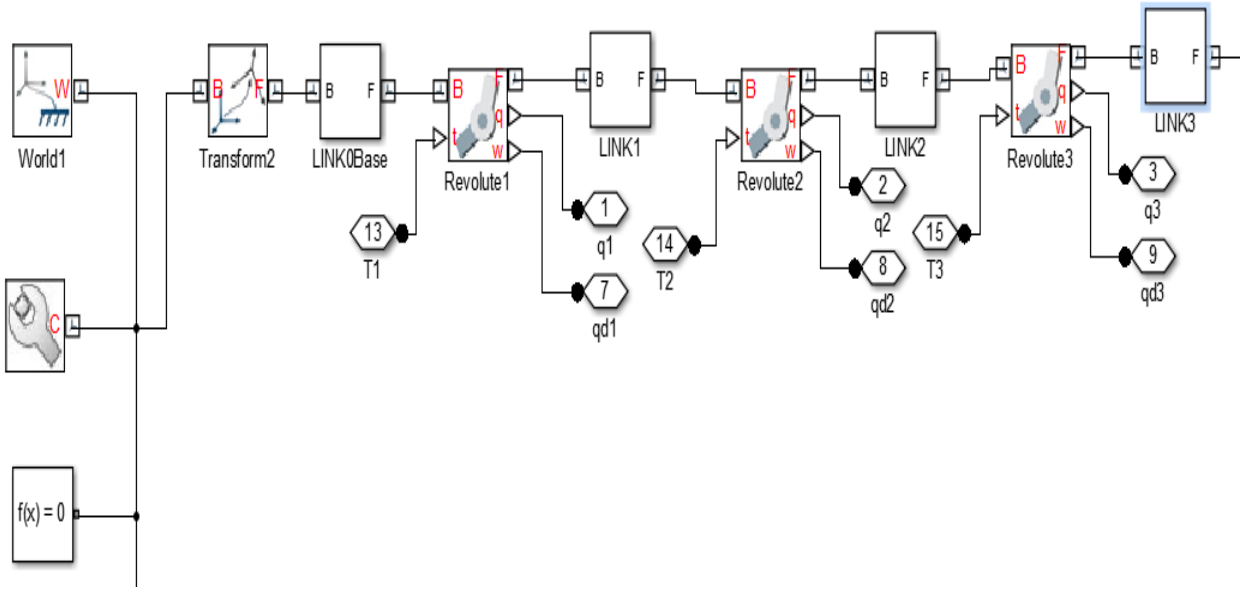

Fig. 8 Manipulator kinematic modeling chains defined using the Simmechanics.

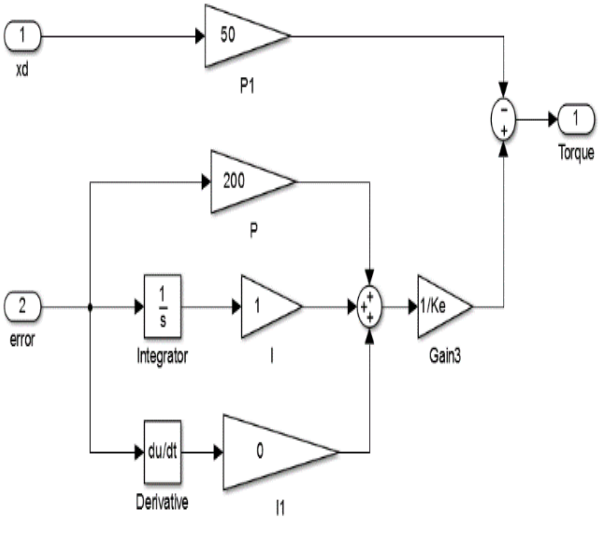

Fig. 9 Flow diagram of the force control system.

The introduced singularities from the inversion of the Jacobin matrix can make the system unstable. To eliminate this problem, the pseudoinverse function in the Matlab was used [30-31]. Figure 13 represents the system response for the applied force of the manipulators on the surface of the spherical environment.

The end-effector makes available the desired commanded force. Results as shown in "Fig. 13" indicates that in case of the error even when the system is corrupted with random noise, the control system was able to keep the error to within less than 5%. Fiures 13-14 show that the contact-force Error is less than 5% while, the end-effector position error is less than one percent which confirms the performance accuracy of the hybrid Position-Force controller [32-33].

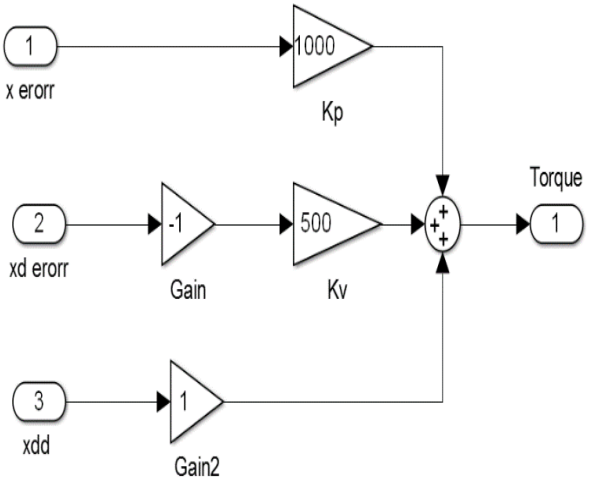

Fig. 10 Flow diagram of the position control system.

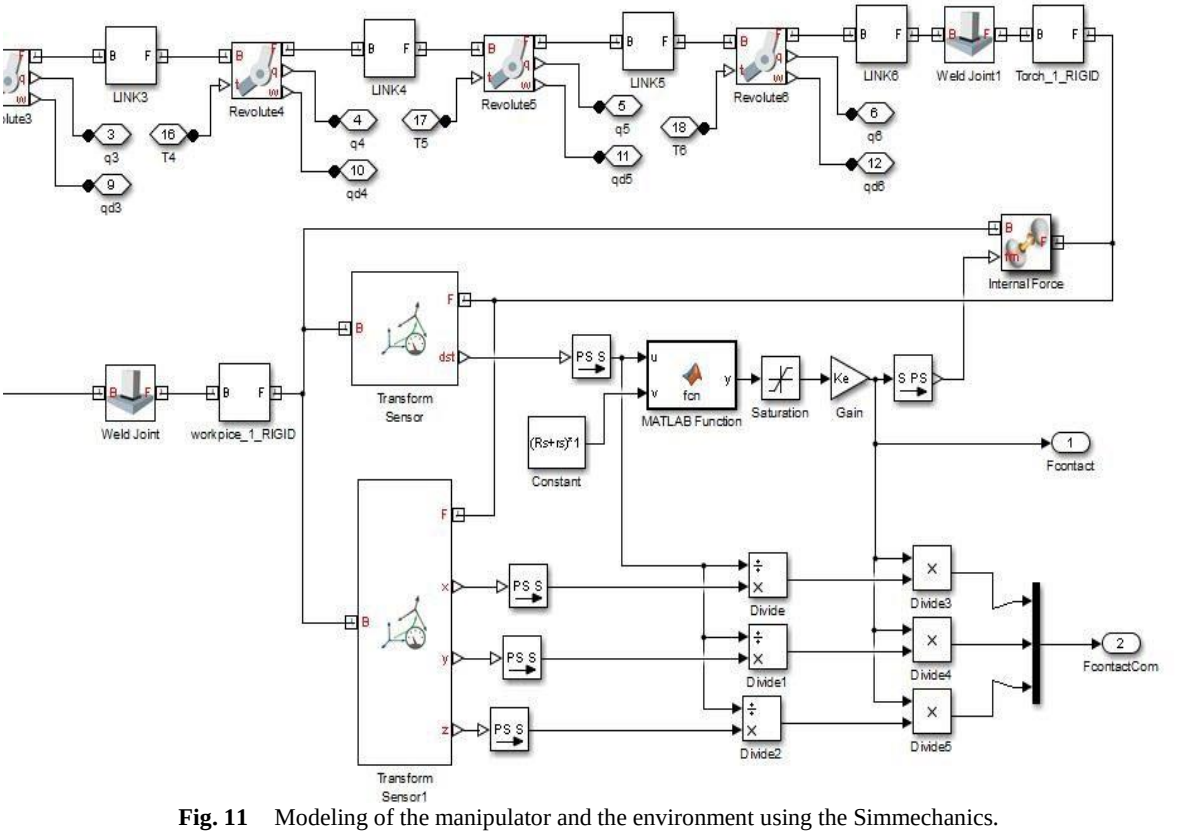

Fig. 11 Modeling of the manipulator and the environment using the Simmechanics.

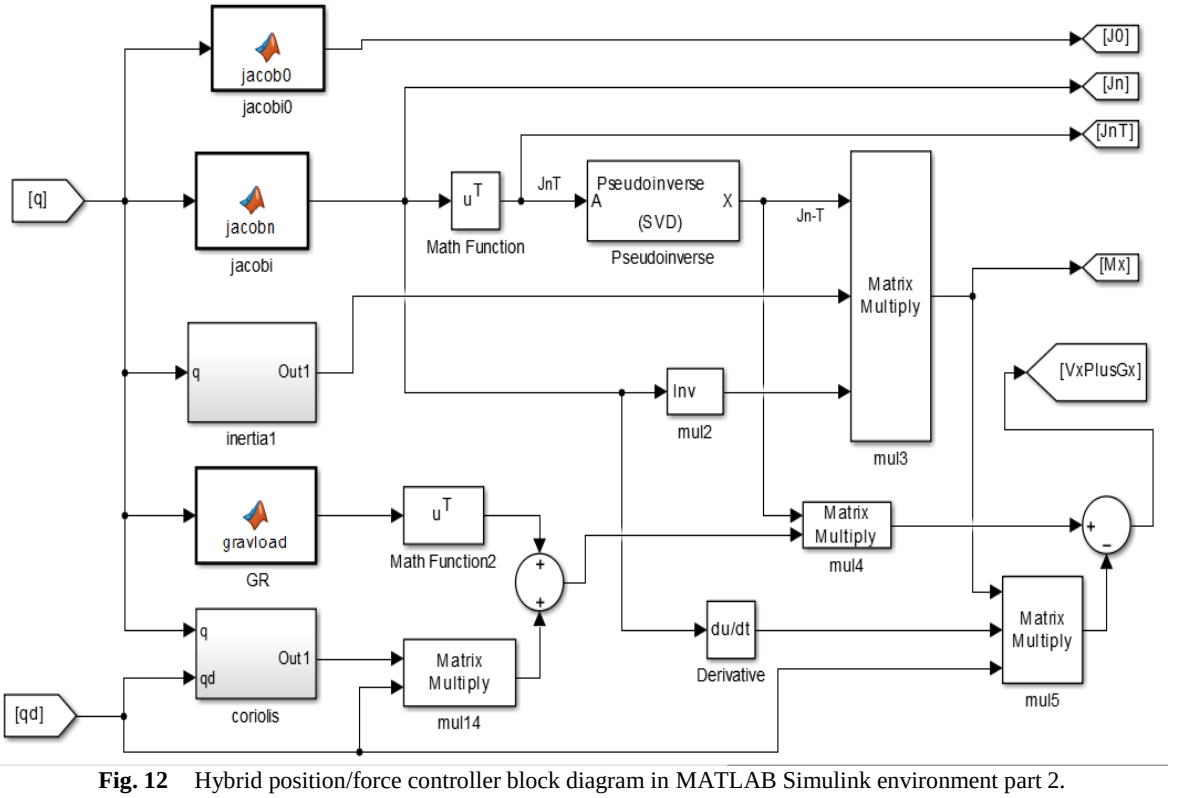

Fig. 12 Hybrid position/force controller block diagram in MATLAB Simulink environment part 2.

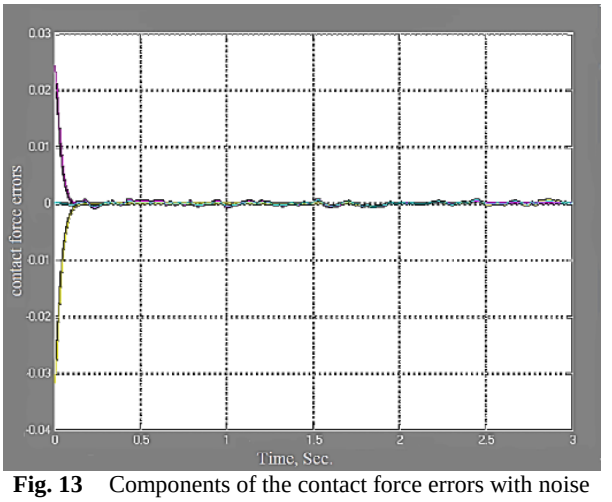

Fig. 13 Components of the contact force errors with noise corrupted position controller and gains of Kp=2500, Kv=100.

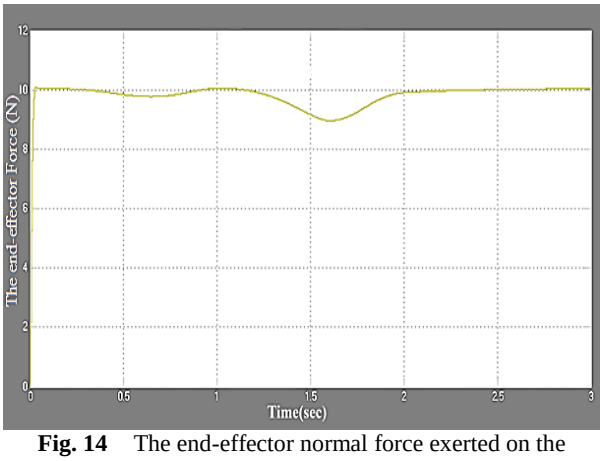

Fig. 14 The end-effector normal force exerted on the surface with controller gains of K_v=350, Kp=100, K_i=20.

The Tracking trajectories in the x, y, and z axes are shown in "Figs. 15 to17" .

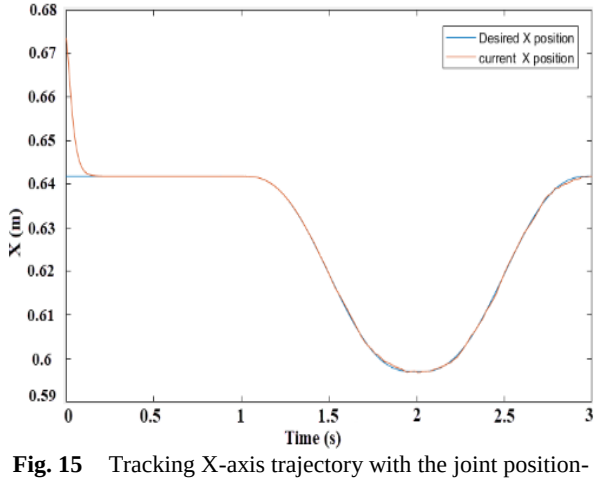

Fig. 15 Tracking X-axis trajectory with the joint position-force controller.

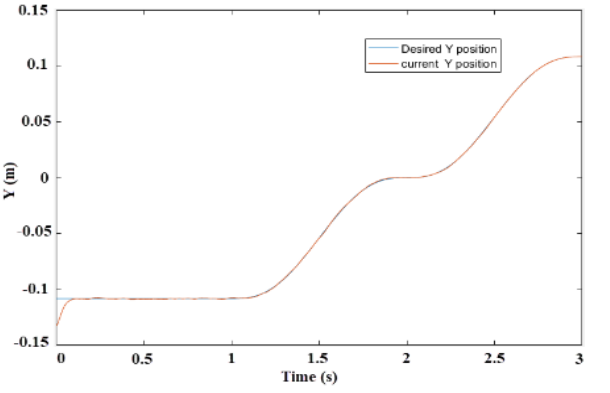

Fig. 16 Tracking the Y-axis trajectory with the joint position-force controller.

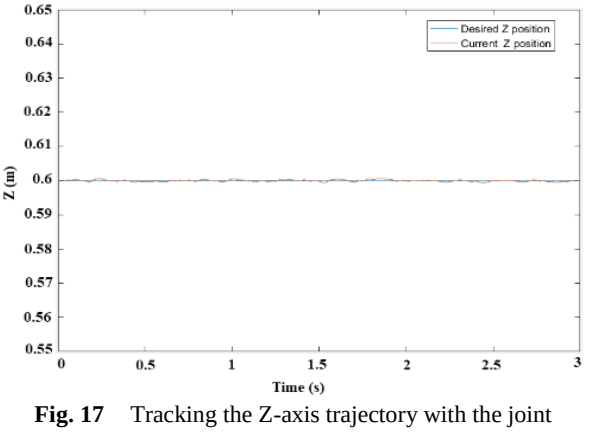

Fig. 17 Tracking the Z-axis trajectory with the joint position-force controller.

Figur 18" shows the simulator for the noise generator using a random number generator with an amplitude between 1 and -1 and discreet time steps of 0.001 second.

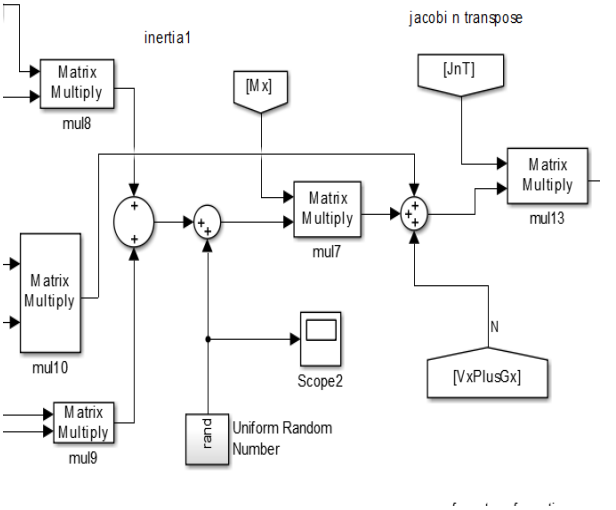

Fig. 18 Noise generator simulator using the random number generator.

8 RESULT

Figure 6 shows the complete MATLAB model for this control system. It includes the manipulator mechanical model, the model-based controller dynamics, the virtual environment model, the hybrid position-force controller, and the desired input trajectories.

To test the performance of the control system, the desired force and position trajectories are given as the system inputs. These inputs are indicated in "Fig. 10" by "Figs. 1, 2 and 3" . The points are connected with straight lines and the time intervals are one second. The Input force was kept constant, and equal to the value of 10 [N].

The environment stiffness was selected as 100000 [N/m] and the Position control gains were chosen by trial and error as KTv=500, KTp=1000. In the force control law, the gains were chosen as KNv=35, KNp=100, and KNp=20. The following graphs show the simulation results.

Figure 14 shows that the contact force profile follows the desired trajectory i.e., the hybrid position-force controller performs as desired and, the Steady-state force converges to the value of around 10 N, which indicates that the steady-state error is zero. Therefore, the manipulator tries to follow the given input position trajectory but, the resulting contact force doesnot permit the end-effector tool to enter the spherical surface. Thus, the manipulator is expected to have position errors in the normal space.

9 CONCLUSION

This paper describes the process of modeling a robotic manipulator and its end-effector force interaction with a virtual environment. The use of Simmechanics as a tool to model the mechanics of the robot allows verification of this model-based control algorithm. In real applications, the precise mechanical characteristics of the model are usually unknown to the designer. Therefore, the ability to model the robot manipulator using SimMechnics toolboxes provides a great advantage in the simulation. It allows the testing of the techniques like the proposed dynamic model-based control system. Also, numerical testing of these techniques can be used to extract the model parameters of the plant under consideration. On the other hand, a first insight into the use of the Cartesian joint position-force controller shows its adequacy to control the robot-environment interaction, especially in the completely or partially unknown environment where robot compliance comes before the controlling of the exact forces applied to the environment. In other words, the proposed technique decreases cost because it avoids actual collisions, the possible failure of end-effector or damage to the environment especially when the geometry of the environment is unknown for the controller.

The proposed joint position-force controller has successfully tracked the commanded position and the requested tactile force with relatively good accuracy. The tactile forces in a steady-state condition show less than 5 percent error. The position control performance even with added noise shows less than 1 percent error. Also, the proposed technique reduces cost by using the animated simulation of the robot manipulator and the objects in the environment without any physical damage to the objects in the environment or the robot mechanism before completion and verification of the design.

REFERENCES

[1] Raibert, M. H., Craig, J. J., Hybrid Position/Force Control of Manipulators, Journal of Dynamic Systems Measurement and Control, Vol. 103, No. 2, 1981, pp. 126–33.

[2] Gueaieb, W., Karray, F., and Al-Sharhan, S., A Robust Adaptive Fuzzy Position/Force Control Scheme for Cooperative Manipulators, IEEE Transactions On Control Systems Technology, Vol. 11, 2003, pp. 516–528.

[3] Gueaieb, W., Karray, F., and Al-Sharhan, S., A Robust Hybrid Intelligent Position/Force Control Scheme for Cooperative Manipulators, IEEE/ASME Transactions On Mechatronics, Vol. 12, 2007, pp. 109–125.

[4] Kumar, N., Panwar, V., Sukavanam, N., Sharma, S. P., and Borm, J., Neural Network-Based Hybrid Force/Position Control for Robot Manipulators, Int. Journal of Precision Engineering and Manufacturing, Vol. 12, No. 3, 2011, pp. 419–426.

[5] Matsuno, F., Yamamoto, K., Dynamic Hybrid Position/Force Control of a Two-Degree-of-Freedom Flexible Manipulator, Journal of Robotic Systems, Vol. 11, 1994, pp. 355–366.

[6] Queiroz, M. S. De, Hu, J., Dawson, D. M., Burg, T., and Donepudi, S. R., Adaptive Position/Force Control of Robot Manipulators Without Velocity Measurements: Theory and Experimentation, IEEE Transactions On Systems, Man, and Cybernetics, Part B, Vol. 27, No. 5, 1997, pp. 796–809.

[7] Xiao, D., Ghosh, B, K., Xi, N., and Tarn, T. J., Sensor-Based Hybrid Position/Force Control of a Robot Manipulator in an Uncalibrated Environment, IEEE Transactions On Control Systems Technology, Vol. 8, 2000, pp. 635–645.

[8] Yoshikawa, T., Dynamic Hybrid Position/Force Control of Robot Manipulators- Description of Hand Constraints and Calculation of Joint Driving Force, IEEE Journal On Robotics and Automation, Vol. 3, pp. 386–392, 1987.

[9] Zhang, H., Paul, R., Hybrid Control of Robot Manipulators, Proceedings of IEEE International

Conference on Robotics and Automation, Vol. 2, 1985, pp. 602–607.

[10] Neha, E., Suhaib, M., Asthana, S., and Mukherjee, S., Grasp Analysis of a Four-Finger Robotic Hand Based On Matlab Simmechanics, Journal of Computational and Applied Research in Mechanical Engineering, Vol. 9, 2019, pp. 169-182.

[11] Pliego-Jiménez, J., Arteaga-Pérez, M, A., Adaptive Position/Force Control for Robot Manipulators in Contact with A Rigid Surface with Uncertain Parameters, European Journal of Control, 22, 2015, pp. 1–12.

[12] Ding, G., Zhao, B. Dong, B., Liu, Y., and Li, Y., Adaptive Neural Network Position/Force Decomposed Control for Constrained Reconfigurable Manipulator, IEEE International Conference on Mechatronics and Automation, 2015, pp. 1561–1566.

[13] Homayounzade, M., and Keshmiri, M., Adaptive Position/Force Control of Robot Manipulators with Force Estimation, Second RSI/ISM International Conference on Robotics and Mechatronics (ICROM), IEEE, 2014, pp. 736–741.

[14] Hori, Y., Shimura, K., and omizuka, M., Position/Force Control of a Multi-Axis Robot Manipulator Based On the Tdof Robust Servo Controller for Each Joint, IEEE American Control Conference, 1992, pp. 753–759.

[15] Guo, W. G., Komada, S., Ishida, M., and Hori, T., Coordinated Motion of Two Redundant Manipulators by Simple Hybrid Position/Force Controller with Disturbance Observer Industrial Technology, (ICIT '96), Proceedings of The IEEE International reference, 1996.

[16] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., Robotics: Modelling, Planning and Control, Springer, 2009.

[17] Zhang, L., Wang, J., Chen, J., Chen, K., Lin, B., and Xu, F., Dynamic Modeling for a 6-dof Robot Manipulator Based On a Centrosymmetric Static Friction Model and Whale Genetic Optimization Algorithm, Elsevier Journal of Advances in Engineering Software, 2019.

[18] Ghasemi, M. H., Korayem, A. H., Nekoo, S. R., and Korayem, M. H., Improvement of Position Measurement for 6R Robot Using Magnetic Encoder AS5045, Journal of Computation and Applied Research in Mechanical Engineering, Vol. 6, No. 1, 2017, pp. 11-20.

[19] Iqbal, J., Ul Islam, R., and Khan, H., Modeling and Analysis of a 6 dof Robotic Arm Manipulator, Canadian Journal On Electrical and Electronics Engineering, Vol. 3, No. 6, 2012, pp. 300-306.

[20] Ghaemi Osgouie, K., Gard, B., Using the Matrix Method to Compute the Degrees of Freedom of Mechanisms, Journal of Applied and Computational Mechanism, Vol. 3, No. 3, 2017.

[21] Ghorbanian, S. M., Rezaei, A. R., Khoogar, M., Zareinejad, and K., Baghestan, A Novel Control Framework for Nonlinear Time-Delayed Dual-Master Single-Slave Teleportation, Elsevier ISA Transactions, 2012.

[22] Seyed Mousavi, S. M., Khoogar, A. R., and Tale Masouleh, M., Accuracy Comparison of Spherical Parallel Manipulators Based On Joint Clearance, International Conference on Robotics and Mechatronics, ICROM, 2017.

[23] Hancheol Cho, On Robust Adaptive PD Control of Robot Manipulators, Journal of Applied and Computational Mechanism, Vol. 6, 2020.

[24] Bazargan-Lari, Y., Khoogar, A. R., A More Realistic Replica of a Walking Biped for Implying Dynamics and Control On Ascending and Descending Stairs by Considering Self-Impact Joint Constraint, Iranian Journal of Science and Technology, Transaction of Mechanical Engineering, 2019.

[25] Mousavi, S. M., Khoogar, A. R., and Ghasemi, H., Time Domain Simulation of Ship Motion in Irregular Oblique Waves, Journal of applied fluid mechanics, 2020.

[26] Khoogar, A. R., Tehrani, A. R. K., and Tajdari, M., A Dual Neural Network for Kinematic Control of Redundant Manipulators Using Input Pattern Switching, Journal of Intelligent Robotic Systems, 2010.

[27] Tajdari, M., Hajabasi, A., and Khoogar, A. R., A New Approach to Finite Element Modeling and Simulation of Flexible Robot Manipulators, Journal of Mechanics and Aerospace Engineering, 2008.

[28] Bayati Chaleshtari, M. H., Norouzi, E., and Ahmadi, H., Optimizing Control Motion of a Human Arm with PSO-PID Controller, Journal of Computation and Applied Research in Mechanical Engineering, Vol. 7, No. 1, 2017, pp. 23-34.

[29] Zahari T., Abdelhakim D., and Azeddein K., Drilling Force Control for Robot Manipulator with Combined Rigid and Soft Surface, Journal of Applied Mechanics and Materials, 2013, pp. 1741-1747.

[30] Wells, T. S., Yang, S., MacLachlan, R. A., Lobes Jr, L. A., Martel, J. N., and Riviere, C. N., Hybrid Position/Force Control of an Active Handheld Micromanipulator for Membrane Peeling, International Journal of Medical Robot, 2016, pp. 85-95.

[31] Bazargan Lari, Y., Eghtesad, M., and Khoogar, A. R., Dynamics and Control of a Novel Constrained Biped, Lap Lambert Academic Publishing, 2017.

[32] Ferre, M., Ernst, M., and Wing, A., Springer Series on Touch and Haptic Systems, Published by Springer Nature and Eurohaptics Society, ISSN: 2192-2977, 2020.

[33] Salazar, S. V., Pacchierotti, C., Xavier de Tinguy, Maciel, A., Marchal, M., Altering the Stiffness, Friction, And Shape Perception of Tangible Objects in Virtual Reality Using Wearable Haptics, Ieee Transactions On Haptics, Vol. 13, Issue 1, 2020.