Experimental Object Manipulation Assistive Robotic Arm for Pick and Place Task

Mahdi Yousefi Moghaddam [a], Farzad Cheraghpour Samavati [b,*]

[a] *Faculty of Electrical, Biomedical and Mechatronics Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

[b] *Department of Mechanical Engineering, Pardis Branch, Islamic Azad University, Tehran, Iran*

Received 11 September 2019;Accepted 16 August 2021

Abstract

For people that need total or partial assistance to perform daily tasks, assistive robots are one of the solutions. Force control of these robots when interact with human or manipulate objects, is one the challenging problems in this area. In this paper a ROS-based force control system is implemented on a JACO assistive robot for grasping tasks. To do this, we need to know exactly how the robot's internal performance works and how it is structured. It is usually achieved by designing sophisticated control techniques that meet these criteria. Advanced control architectures such as torque computation control allow tracking of desired paths with high accuracy, however, the need to integrate robotic models remains. The work presented in this study provides a basis for applying these techniques to the JACO robotic arm. The calculation is based on the Euler-Lagrange method of calculating the internal energy. The results are then analyzed to ensure the models estimated with control schemes. Therefore, more advanced analysis and control techniques can be implemented on this robotic arm. Finally, this study can be controlled by PID with respect to the torque entered to the end effector by the object so that the robotic arm can move from the initial position to the secondary position with optimum capture and torque control of all robot joints. The experimental results showed the effectiveness of proposed method to perform grasping and manipulation scenario successfully.

Keywords: Force Control, Assistive Robot, Grasp, JACO, Dynamic Modelling

1. Introduction

For people that need total or partial assistance to perform daily tasks, assistive robots are one of the solutions. Assistive robotics (AR) is referred to robots that assist people with physical disabilities through physical interaction or assist through non-contact interaction, such as those that interact with convalescent patients in a hospital or senior citizens in a nursing home. Feil-Seifer & Mataric were distinguished between robots that provide assistance through interaction and without physical contact, namely socially assistive robotics, [1].

An adequate definition of an assistive robot is one that gives aid or support to a human user. Research into assistive robotics includes rehabilitation robots, [2, 3], wheelchair robots and other mobility aides, [4], companion robots, [5], manipulator arms for the physically disabled, fig. 1, [6], upper-limb exoskeleton robots, [7, 8], lower-limb exoskeleton robots, [9], and educational robots, [10]. These robots are intended for use in a range of environments including schools, hospitals, and homes.

[*] Corresponding Author. Email:samavati@pardisiau.ac.ir

Force control of these robots when interact with human or manipulate objects, is one the challenging problems in this area. Especially for object manipulation tasks, strong grasping is one of the most important capabilities that assistive manipulators are expected to have Using. This ability becomes critical when the robot, in particular, starts operating in an unstructured environment. Due to the difference between the model and the real world, or even errors from precise robot control, its planned grasping is rarely performed without any errors, [11].

1.1. Grasping and its Applications

Grasping is the first step in the process of moving objects this, which examines the mechanical relationship between gripping the end-effector robot and the object. These mechanic communications introduce commit move situation conditions and, embedment arm robot circumstances in its grasp while implement duty. By looking at the process of object manipulation for human, we can understand that the most important point in performing this process is the grasp of the object. Grasping object by robot is one of the topics that in various applications, such as, move delicate object, Take irregular object and, bungle pieces.

1.2. Definition And Important Issue

Position control in grasp, whenever, grasp follow course in space, it's easy. But when the End-Effector (EE) of robotic arm touched with environment, controlling of its position will not enough alone. In this paper, we follow one of the common ways that use sensor feedback in EE, [12].

Limitations such as taking objects of different shapes and surfaces, the ability to modify small, high-precision positions, as well as taking into account the characteristics of the object taken, such as multi-finger robotic hands. A specific deployment strategy for the property of unknown objects is the shape recognition by the mechanical arm. The approach is considered by controlling the position, optimizing the position of the contact points, [13], its force and its grasp using the control force. Its grasp and sophisticated deployment are two of the important capabilities expected of a robot.

1.3. Robotic Arm

The KINOVA JACO^2 robotic manipulator is six degrees of freedom commercial robotic manipulator, shown in fig. 1. This robot mobilized with a three-finger controllable hand that acts as an end-effector.

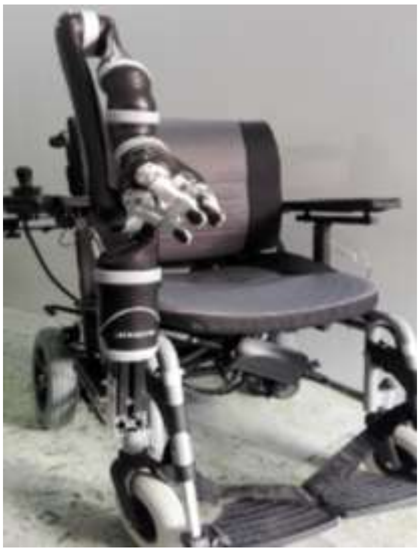

Fig. 1. The six degrees of freedom of JACO mounted on the wheelchair as a assistive robotic arm, [14].

The Robotic arm consists of six pieces of carbon fibre interconnected by six brushless direct current (BLDC) aluminium actuators. They have no mechanical limitation and allow unlimited rotations around their axis. Due to the carbon fibre in this device, the structure is very lightweight. Properties of the robot explained in [15].

In this paper, for experimental tests, the mobile robot of @home team of Mechatronics Research Laboratory (MRL) is used, as shown in fig. 2, [16]. In this robot, the JACO arm mounted on a wheeled mobile platform for executing at home tasks.

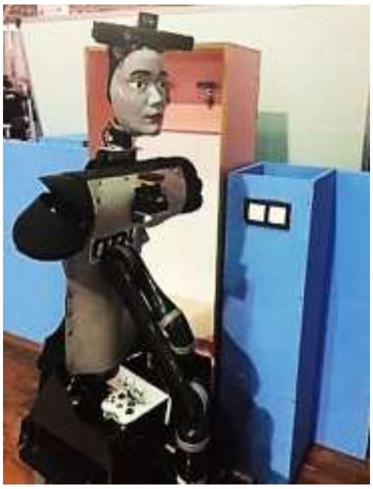

Fig. 2. The JACO mounted on the mobile autonomous robot as an assistive robotic arm, in at home MRL laboratory.

1.4. The Method

The method is implementing control algorithms for the arm in object manipulation task. The defined task consists of exerting force by EE to the grasped object and moving it on a specified path. The framework of control algorithms is implementing in Robot Operating System (ROS) environment. Also the proposed algorithm is implemented with the help of KINOVA's C++ software interface.

The system consists of an object by a commit of robotic arm and a 3-finger hand. The grasping of the physical object shows in **Error! Reference source not found.**. The end point of the robot moves to reaches the contact point for exerting the force according to **Error! Reference source not found.**. The forces/torques exerted to the arm by means of object. For controlling the force, the grasp control is modelled by human imitation. Capturing object is changing rapidly. Catch, is the manipulation of a body by the grasp so that it can be moved in a desired direction. For this purpose, when the robot is moving and moving the body, it must be appropriately and with minimal force applied to avoid fracture and/or deforming the body, preventing it from slipping and moving it steadily.

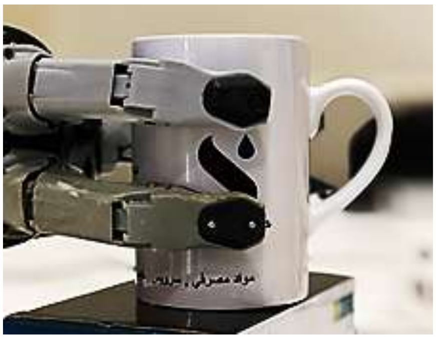

Fig. 3. Grasping of the object by JACO and its 3-finger hand

2. Mathematical Model of Robotic Arm

This section will focus on the derivation of the kinematic, differential kinematic and dynamic models of the JACO^2 Robot. These models need to be estimated, in order to perform tasks in the workspace with the best possible fashion. The kinematic model, or also called geometric model, establishes the relation between a given configuration of the robot in joint variables and the correspondent position and orientation coordinates of the end-effector. The differential kinematic, in short, entails the relationship between joint and EE velocities. On the other hand, the dynamic model is based on the relation between forces/torques and the resulting motion, and allows a deeper analysis considering additional criteria such as mass, inertia and even friction terms that might influence the behaviour of the robot during task operations.

2.1. Kinematic Model

The JACO^2 robot is comprised of six carbon-fibber links. Each link of the robot can be assumed as having a relative position and orientation with regard to the base of the arm. So the base of the robot assumed as reference frame, which is associated to the reference frames of the other links by applying homogeneous transformations.

Suppose an arbitrary point P in space needs to be represented with respect to the base frame O_0 as shown in Fig. . If its position is only known from the reference frame O_1 then, one way to achieve this would be by considering the position from that reference frame, adjust the orientation of the point to match that of the base frame, and add the distance between the origins of both reference frames. Assuming that the matrix R_1^0 is the rotation matrix between frames O_1 and O_0 then the position of P with respect to the base frame is,

$$p^0 = O_1^0 + R_1^0 p^1 \tag{1}$$

where $O_{1\,3\times1}^0$ is the distance between the reference frames, and $R_{3\times3}$ is the rotation matrix between two frames.

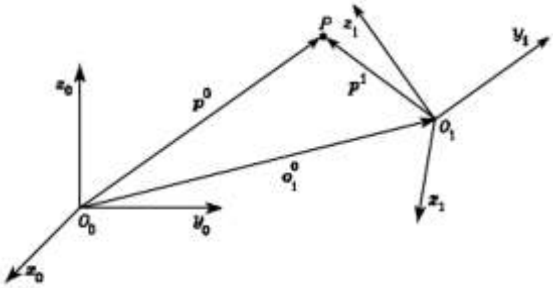

Fig. 4. Representation of a point between reference frames

Homogeneous transformation is the matrix representation between two frames.

$$T_1^0 = \begin{bmatrix} R_1^0 & o_1^0 \\ 0_{1\times3}^T & 1 \end{bmatrix} \tag{2}$$

In which T_1^0 is the homogeneous transformation matrix. These 4×4 dimensional matrices are essential in order to derive the kinematic model. Analogously, a 6-DOF robotic manipulator like the JACO^2 can be represented by a set of reference frames between each of its links, in a similar fashion as displayed by Fig. . The transformation between the EE and the base frame is given by

$$T_{EE}^0 = T_1^0\, T_2^1\, T_3^2\, T_4^3\, T_5^4\, T_{EE}^5 \tag{3}$$

where each homogenous transformation matrix (T_i^{i-1}), is a function of joint variable (q_i). Equation (3), presents the direct kinematics model.

2.2. Denavit-Hartenberg Parameters

To derive the transformation matrices of each link, it is necessary to assign frames to each joint according to Denavit-Hartenberg (DH) manner, [17]. The defined frames are shown in fig. 5 with length values shown in Table. 1.

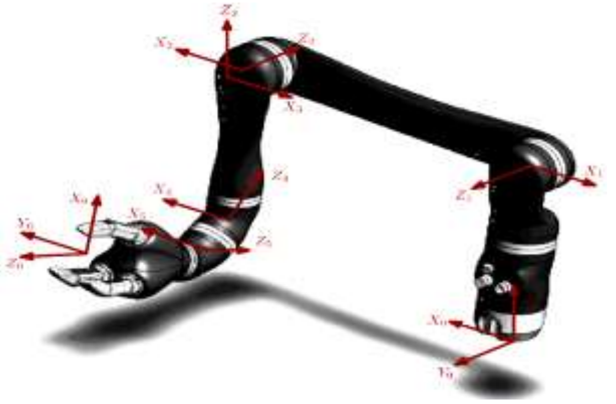

Fig. 5. The DH frames defined for joints of JACO arm

For Definition of DH parameters, first some auxiliary defined in Table 2, and DH parameters of Robot is shown in Table 3.

Table 1
Robot length values, [15]

Robot Lengths [mm]		
$\boldsymbol{D_1}$	275.5 mm	Base to Elbow
$\boldsymbol{D_2}$	410 mm	Arm lengths
$\boldsymbol{e_2}$	9.8 mm	Joint 3-4 lateral offset
$\boldsymbol{D_3}$	207.3 mm	Front Arm lengths
$\boldsymbol{D_4}$	74.1 mm	1st Wrist lengths
$\boldsymbol{D_5}$	74.1 mm	2nd Wrist lengths
$\boldsymbol{D_6}$	160 mm	Wrist to center of the Hand

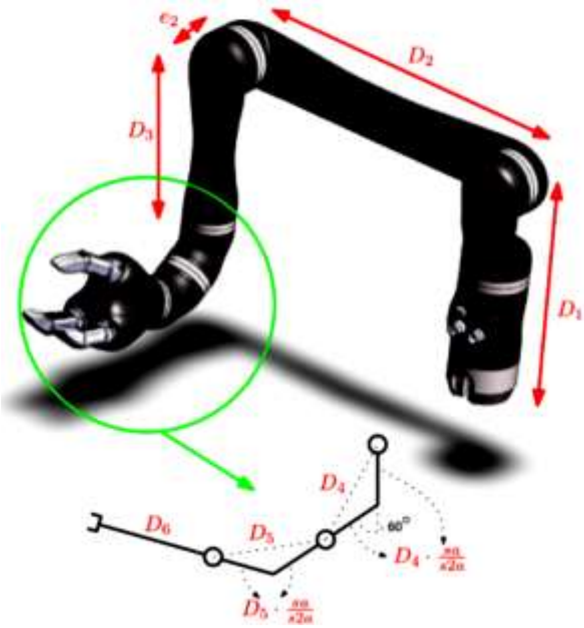

Fig. 2. JACO link lengths definitions

Table 2
Auxiliary Variables of Robot

Var.	Definition	Var.	Definition
$c\beta$	$\cos(\beta)$	$s\beta$	$\sin(\beta)$
$c2\beta$	$\cos(2\beta)$	$s2\beta$	$\sin(2\beta)$
β	$\frac{\pi}{6}$	$d4b$	$D_3 + D_5\left(\frac{s\beta}{s2\beta}\right)$
$d5b$	$D_4\left(\frac{s\beta}{s2\beta}\right) + D_5\left(\frac{s\beta}{s2\beta}\right)$	$d6b$	$D_5\left(\frac{s\beta}{s2\beta}\right) + D_6$

Table 3
DH parameters of Robot

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	$\pi/2$	0	D_1	q_1
2	π	D_2	0	q_2
3	$\pi/2$	0	$-e_2$	q_3
4	2β	0	$-d4b$	q_4
5	2β	0	$-d5b$	q_5
6	π	0	$-d6b$	q_6

$$T_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -c_{\alpha_i}s_{\theta_i} & s_{\alpha_i}s_{\theta_i} & \alpha_i c_{\theta_i} \\ s_{\theta_i} & c_{\alpha_i}c_{\theta_i} & -s_{\alpha_i}c_{\theta_i} & \alpha_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Additionally, the JACO^2 physical angles have to be converted into the angles of the DH algorithm, which is given by relation shown in Table 4. Since the DH parameters are known, by having the transformation matrices relating subsequent links by applying equation (4), then the forward kinematics (FK) model can be calculated through equation (3).

Table 4
Physical Angle Transformation and DH Parameters

DH Algorithm Angle	Physical Angle
q_1	$-q_{1Robot}$
q_2	$q_{2Robot} - 90°$
q_3	$q_{3Robot} + 90°$
q_4	q_{4Robot}
q_5	$q_{5Robot} - 180°$
q_6	$q_{6Robot} + 90°$

2.3. Jacobian Matrix

The relationship between the angular and linear velocities of the EE and the joint velocities is defined by *Jacobian matrix*. Jacobian can also be used for computing the torques being in effect at the joints when a specific force is being applied at the EE.

First, let the EE linear velocity be defined as $\dot{p}_{EE}$ and the angular velocity be defined as ω_{EE}, while the joint velocities vector be $\dot{\boldsymbol{q}}$, then the relation between joint velocities and EE velocity is,

$$\begin{cases} \dot{p}_{EE} = J_P(q)\,\dot{\boldsymbol{q}} \\ \omega_{EE} = J_O(q)\,\dot{\boldsymbol{q}} \end{cases} \rightarrow \begin{bmatrix} \dot{p}_{EE} \\ \omega_{EE} \end{bmatrix} = \begin{bmatrix} J_P(q) \\ J_O(q) \end{bmatrix} \dot{\boldsymbol{q}} \qquad (5)$$

Where $J_P(q)$ is a (3×6) matrix relating the linear velocity of the EE ($\dot{p}_{EE}$) with the joint velocities vector ($\dot{\boldsymbol{q}}$), and $J_O(q)$ is similarly a (3×6) matrix detailing the relation between the angular velocity of the EE (ω_{EE}) and the joint velocities vector.

Combined together, these sub matrices make the Jacobian matrix, $J(q)$.

$$J(q) = \begin{bmatrix} J_P(q) \\ J_O(q) \end{bmatrix} = \begin{bmatrix} J_{P1} & \cdots & J_{P6} \\ J_{O1} & \cdots & J_{O6} \end{bmatrix}$$
$$\rightarrow \boldsymbol{V_{EE}} = J(q)\,\dot{\boldsymbol{q}} \qquad (6)$$

Extending equation (6), the linear and angular velocity components can be computed as,

$$V_{EE} = \begin{cases} \dot{p}_{EE} = \displaystyle\sum_{i=1}^{6} \frac{\partial p_{EE}}{\partial q_i}\dot{q}_i = \sum_{i=1}^{6} J_{P_i}(q)\,\dot{q}_i \\ \omega_{EE} = \displaystyle\sum_{i=1}^{6} \omega_{i-1,i} = \sum_{i=1}^{6} J_{O_i}(q)\,\dot{q}_i \end{cases} \qquad (7)$$

The joint velocity $\dot{q}_i$ is expressed differently if the joints are prismatic or revolute. In this case, the JACO^2 has six joints with all of them being revolute; therefore the angular and linear velocity for a revolute joint for subsequent links is denoted as,

$$\begin{cases} \dot{\boldsymbol{p}}_{\boldsymbol{i}} = \dot{\boldsymbol{p}}_{\boldsymbol{i-1}} + \boldsymbol{\omega_i} \times \boldsymbol{r}_{\boldsymbol{i-1},\boldsymbol{i}} \\ \boldsymbol{\omega_i} = \boldsymbol{\omega_{i-1}} + \dot{\theta}_i \boldsymbol{z_{i-1}} \end{cases} \qquad (8)$$

where $\boldsymbol{z_{i-1}}$ is the unit vector of the joint i axis and $\boldsymbol{r}_{\boldsymbol{i-1},\boldsymbol{i}}$ corresponds to the distance from the origin of the coordinate frame i with respect to the origin of the coordinate frame $i-1$. This means that the velocity in link i is the same as in link $i-1$ the equalities presented in equation (7) are rearranged as,

$$\begin{cases} J_{P_i}\,\dot{q}_i = \boldsymbol{\omega_{i-1,i}} \times \boldsymbol{r_{i-1,EE}} = \dot{\theta}_i \boldsymbol{z_{i-1}} \times (\boldsymbol{p_{EE}} - \boldsymbol{p_{i-1}}) \\ J_{O_i}\,\dot{q}_i = \dot{\theta}_i \boldsymbol{z_{i-1}} \end{cases} \qquad (9)$$

Therefore,

$$\begin{bmatrix} J_{P_i} \\ J_{O_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{z_{i-1}} \times (\boldsymbol{p_{EE}} - \boldsymbol{p_{i-1}}) \\ \boldsymbol{z_{i-1}} \end{bmatrix} \qquad (10)$$

where $\boldsymbol{p_{EE}}$ is the distance from the origin of the EE coordinate frame to the base frame and $\boldsymbol{p_{i-1}}$ the analogous distance from link $i-1$.

With the FK model, it is possible to define the velocities of the executors for each of the arm frameworks. For this work, the Jacobian related to the linear velocities is referred to the base frame while the angular velocities are referred to the EE frame, which means it is necessary to apply the following remapping,

$$J_{\omega}^{EE} = R_0^{EE} J_{\omega}^{0} \qquad (11)$$

Since the JACO is non-redundant, the inverse kinematics (IK) model is calculated by simply inverting the Jacobian matrix,

$$\dot{\boldsymbol{q}} = J^{-1}\,\boldsymbol{V_{EE}} \qquad (12)$$

2.4. Dynamic Parameters

To obtain the dynamic parameters of the JACO^2, the computer-aided design (CAD) model of the robot is generated. The CAD model provides an overall visualization of the robot with its different aggregated components. The complete set of coordinate frames for the various links and actuators is illustrated along with their relative distances with respect to the base frame, fig. 7.

For a practical experiment such as this project, it is important to consider all the main components of the arm (even the inertial parameters of the actuators) to obtain the best results from the dynamic parameters.

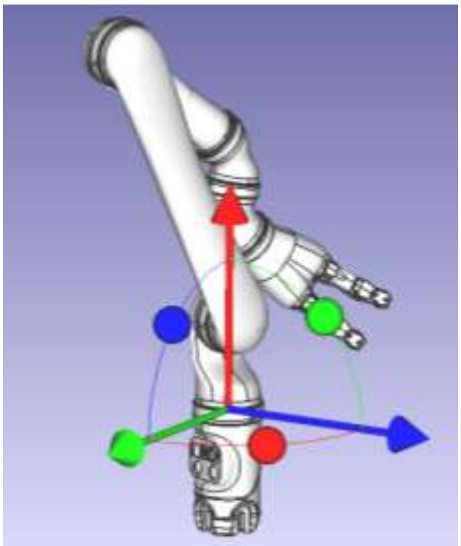

Fig. 7. CAD model of the JACO^2 robot

The total mass of each section consists of the mass of link and the plastic strip located on the joint,

$$m_i^{joint} = m_i^{actuator} + m_i^{ring} \tag{13}$$

where m_i^{joint} is the combined mass of actuator and plastic ring for the i joint. The dynamic parameters can be attained pursuing different methodologies. One can assume that each dynamic contribution of the links and actuators is calculated separately. The other approach is that the dynamic parameters of each degree of freedom of the robot consist of joint and link, are computed cumulatively.

2.5. Dynamic Model

The dynamic model of the robot is based on the relation between the position, velocity and acceleration of the joints and the resulting torques delivered to the joints. It based on the estimation of the motions of the robot, given the torques/forces applied to the joints. So the acceleration of each joint of robot, as the cause of joint torque array ($\boldsymbol{\tau}$), End-Effector Force array ($\boldsymbol{F_{EE}}$), and joint position array ($\boldsymbol{q}$) and velocity array ($\dot{\boldsymbol{q}}$),

$$\ddot{\boldsymbol{q}} = function(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}, \boldsymbol{F_{EE}}) \tag{14}$$

Also the torque exerted to joint by actuator, which we could control on it with control commands, is a function of joint position, velocity and acceleration.

$$\boldsymbol{\tau} = function(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{F_{EE}}) \tag{15}$$

Equations (14) and (15) are the basis of the torque/force control architecture design. Many parameters, such as the mass of each link and their inertia and friction, have a negative effect on the performance of arm movement. In most of the studies done by both Euler-Lagrange (EL) and Newton-Euler (NE) methods or combining these methods, the dynamic model of the robot arm is identified.

The EL technique has its foundation on the kinetic and potential energies existent during robot motion. This can be summarized by the Lagrangian (L), represented by,

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = T(\boldsymbol{q}, \dot{\boldsymbol{q}}) - U(\boldsymbol{q}) \tag{16}$$

where $T(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $P(\boldsymbol{q})$ are scalar values representing the kinetic and potential energy of the manipulator. The kinetic energy is given by,

$$T(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} \tag{17}$$

where $M(q)$ is the mass matrix of the robot. The potential energy is described as,

$$U_i(\boldsymbol{q}) = -m_i \boldsymbol{g}^T \boldsymbol{r_{Gi}} \tag{18}$$

where $\boldsymbol{g_{3\times1}}$ is the gravity acceleration vector and $\boldsymbol{r_{G3\times1}}$ is the position vector of the center of gravity (COG) for link $\boldsymbol{i}$ of the robotic arm. m_i is the mass of link $\boldsymbol{i}$ of the robot.

In EL method, the joint generalized non-conservative torque is obtained by,

$$\tau = \frac{d}{dt}\frac{\partial L(q,\dot{q})}{\partial q} - \frac{\partial U(q)}{\partial q} \tag{19}$$

which after some algebraic manipulation the equation of motion of the robotic manipulator could written with the following close form,

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\tau_g}(\boldsymbol{q}) \tag{20}$$

where $\boldsymbol{M}_{6\times6}$ is defined as a symmetric positive definite inertia matrix. $\boldsymbol{C}_{6\times1}$ is the Coriolis and centripetal forces array. $\boldsymbol{\tau_g}$ is the term relating to the gravity forces. With the derivation of equation (19), the calculation of these matrices in real-time, allow us to apply model based control architectures, for torque/force control.

With dynamic parameters, the kinetic energy, and therefore the total torque to each joint of manipulator is given in the general form as,

$$\tau = \sum_{i=1}^{n} \left(\tau_i^{link} + \tau_i^{joint}\right) \tag{21}$$

where being the summation of the contribution of each link and actuator. From equation (19),

$$M(q) = \sum_{i=1}^{6} \left(J_{Pi}^{\boldsymbol{T}} m_i^{joint} J_{Pi} + J_{Oi}^{\boldsymbol{T}} R_i \bar{I}_i R_i^{\boldsymbol{T}} J_{Oi}\right) \tag{22}$$

where for computing Jacobian suppose each joint and link are integrated as a augmented link. So $\bar{I}_i$ is the moment of inertia matrix of augmented links of the robot with respect to *Center of Mass* (COM) of each augmented link. The rotation matrix R_i , transform the rotational term of kinetic energy with respect to base frame (R_i^0).

Computing the Jacobian is adapted from equation (10). The column j of the Jacobian matrix could computed as,

$$\begin{aligned} J_{Pj} &= \hat{z}_{j-1} \times \left(p_{aug\,i} - p_{j-1}\right) \\ J_{Oj} &= \hat{z}_{j-1} \end{aligned} \tag{23}$$

where p_{j-1} is the position of the origin of the DH frame $j-1$ in base frame coordinates, $p_{aug\,i}$ is the position vector of COM of the augmented link i with respect to the base frame, and $\hat{z}_{j-1}$ is the unit vector of axis z of DH frame $j-1$.

3.Control Strategy

In this section the control strategy of the robot in executing object manipulation task, is proposed. The selection of the controller depends on the type of task to be performed. In motion control problems, the robot moves to a defined position to grasp and pick up a specific object, manipulates and transports that object to another defined position, and finally deposits or place it. Such a task is an integral part of any complicated manipulation tasks.

3.1. Conceptual Design of the Control Structure

Control commands are usually executed in the joint space to achieve the desired goals. In this paper, we used the joint space approach for controlling pick and place task. The main goal of this approach is to design a feedback controller such that the joint coordinates track the desired motion as closely as possible. To this end, consider the equation of motion (20) of the robot expressed in the joint space. In this case, the control of robot is normally achieved in the joint space, since the control inputs are the joint torques, fig. 8.

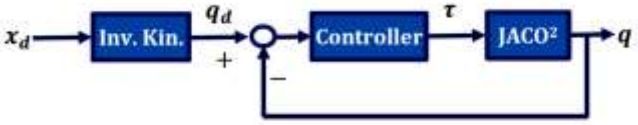

Fig. 8. Joint Space Control approach of the JACO^2

With a classic point of view, control design in robot manipulators can be understood as a PD or PID compensator at the level of each actuator driving the robot joints. Fundamentally, a PD controller is a position and velocity feedback that has suitable closed-loop properties when applied to physical robot as a second order system. Actually, the strong points of a PD or PID control lies in its simplicity and clear physical meaning. So the overall structure of control policy of robot is developed as shown in fig. 9.

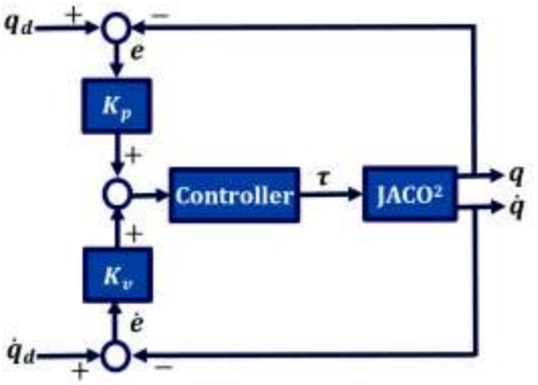

Fig. 9. The PD controller for outer loop Control of the JACO^2

The control approach was built applying the kinematics model, position and velocity feedback, without any attention to the dynamics of the manipulator. This can be enough for general task operations whose precision does not need to be very high. However, including the dynamic model into the control design generally improves the stability, performance and accuracy during task operations as long as its estimations are approximate to the real model. Additionally, there is the significant advantage of making the system linearized, reducing the coupling effects described in the last section.

In paying attention to the dynamic terms, most of robot control schemes can be considered as special cases of the class of computed-torque control or Computed-Torque Method (CTM), fig. 9, which is the technique of applying feedback linearization to nonlinear systems in general.

Feedback Linearization (FL) approach generalizes the concept of invers dynamics of rigid manipulators. The basic idea of FL is to construct a transformation as a so-called inner-loop control, which exactly linearizes the nonlinear system after a suitable state space change of coordinates. One can then design a second stage or outer-loop control in the new coordinates to satisfy the control design specifications such as tracking, disturbance rejection, etc.

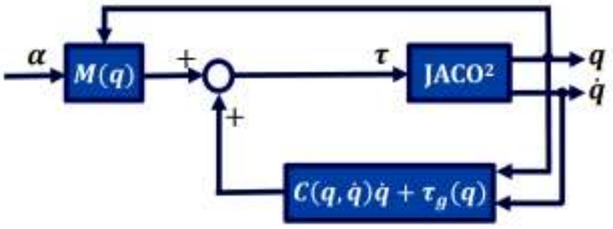

Fig. 10. The Computed-Torque Method (CTM) controller for inner- loop Control of the JACO^2

Finally, as the result of conceptual design of control strategy, the total structure with a CTM controller for inner-loop and a PID controller for outer-loop is as the block diagram shown in fig. 11.

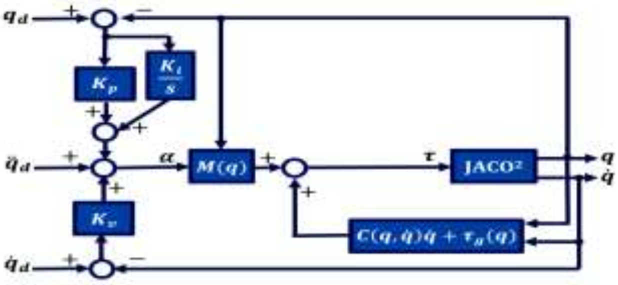

Fig. 11. Total control structure of the JACO^2 : The CTM controller for inner-loop and a PID controller for outer-loop

Computed torque control improves the control design accuracy until its approximation is approximately equal to the actual model itself. Additionally, there is the significant advantage of making the system linearized, reducing the coupling effects described in the last section.

3.2. Formulation of the Control law

As a starting point, according to concept of inverse dynamics control in joint space, our goal is based on cancelling nonlinear terms of dynamic model of the robot and decoupling the dynamics of each link. So consider the control input as below known as computed-torque control,

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{\alpha} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_{\boldsymbol{g}}(\boldsymbol{q}) \tag{24}$$

It consist of an inner nonlinear compensation loop and an outer loop with an auxiliary control signal α. If this control law applied to dynamical model of the JACO^2 manipulator, equation (20), it follows that,

$$\ddot{\boldsymbol{q}}^{*} = \boldsymbol{\alpha} \tag{25}$$

It is important to note that this control input converts a complicated nonlinear controller design problem into a simple design problem for a linear system consisting of six decoupled subsystems.

One approach to the outer-loop control α is PID feedback, as follow,

$$\boldsymbol{\alpha} = \ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}}_{\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}_{\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{i}}\int \boldsymbol{e}dt \tag{26}$$

where $\ddot{\boldsymbol{q}}_{\boldsymbol{d}}$, $\dot{\boldsymbol{q}}_{\boldsymbol{d}}$, and $\boldsymbol{q}_{\boldsymbol{d}}$ represent desired joint positions, velocities and accelerations Therefore the overall control input becomes,

$$\begin{aligned}\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}}_{\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}_{\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{i}}\int \boldsymbol{e}dt)\\ + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{\tau}_{\boldsymbol{g}}(\boldsymbol{q})\end{aligned} \tag{27}$$

It is note that for experiments with the relatively slow robots, the term $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ could be neglected. Also if the task consists of noticeable friction torque ($\tau_{friction}$) and/or external torque ($\tau_{external}$), they should take into account in forming control torque in the left hand side of the equation (27). Finally, the resulting linear error dynamics is,

$$\ddot{\boldsymbol{e}}_{\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}}_{\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}_{\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{i}}\int \boldsymbol{e}dt = \boldsymbol{0} \tag{28}$$

According to linear system theory, where $\boldsymbol{K}_{\boldsymbol{v}}$ and $\boldsymbol{K}_{\boldsymbol{p}}$ and $\boldsymbol{K}_{\boldsymbol{i}}$ are diagonal positive-definite gain matrices, convergence of the tracking error to zero, and therefore the stability of error system is guaranteed, [18]. We obtain the following results to obtain the coefficients for Equation (28) with the natural frequency ω_j , and damping ration ξ ,necessary for each joint,

$$K_{pj} = {\omega_j}^2 \tag{29}$$

And,

$$K_{vj} = 2\xi_j\omega_j \tag{30}$$

where K_{pj} and K_{vj} are the j the diagonal elements of gain matrices, and in relation to K_{ij} , the gains are manually chosen to each joint.

4. Experimental Results

4.1. Position control

The outer-loop control of the total control strategy proposed is a position control algorithm. Examine the control architecture as well as the anonymous geometric, differential, and dynamic models of the JACO^2 robot with the 6-DOF described in the preceding section. In fig. 12, it can be imagining that no object forces removed by the robot arm interfere with the main task forces.

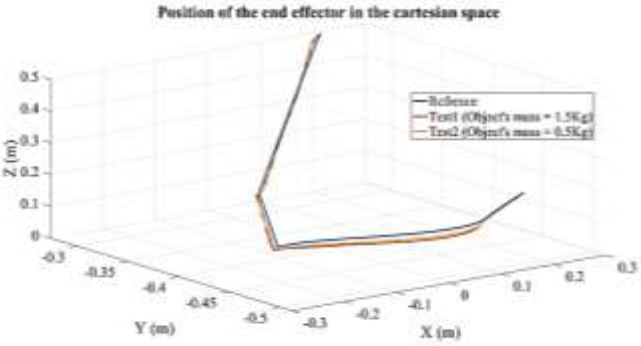

Fig. 12. Comparison of the end-effector of the robot in three main modes, 1.5 kg body weight and 0.5 kg body weight

It can be seen that although there is still an acceptable limit, despite the constant state of error, the path of movement as expected was not a big difference with the main path, but as the main path to the main path (Reference blue diagram and red and yellow graphs when the bottle bubbles with physical opposition conditions)

4.2. Torque Control

In fig. 10, a simple PID controller is described that is purposefully used as a quick way to compute the dynamic anonymous model. This has been done, mainly because the computation surrounding the dynamic model recognisability of a public robot is very troublesome and easily error-prone. Therefore, it is important to use as simple control architecture as possible.

Due to the facilities available in the lab and the existing sensors installed by the factory on its robot arm and grasp, the amount of force applied by the robot's fingers is obtained using the flow sensor.

On the other hand, due to the object that lifts the arm and moves from the initial position to the final position, it needs to identify the dimensions and weight of the object. There are two very important points in this situation.

1. The amount of force exerted by the robot on the object is commensurate to its physical condition, which does not cause damage to the object or grasping to be released during the intended path of the object by the robot.
2. Due to the designed route that the robotic arm is to be locomotive from the initial position to the second position, the arm should not depart from the path when carrying the object.

So, it is necessary to torque control and position. On the other hand, the object can be of different materials and dimensions that vary greatly in weight. Here, our main task is torque control applied by the object and each of the operators along the path for its grasping and transmission, and the second task is to control its position and orientation. In this case, the robotic fingers first grasp the object in the initial position with minimal force and with the coefficients specified in the preceding chapter relations increase the force applied to the object, in fact as soon as it touches, the force grows slightly.

The robotic arm then lifts the object slightly less than one millimetre from the initial position shown in fig. 13. In this case, all the torque inputs to the EE and the joint of the robot are measured and are estimated to be commensurate to the torque generated in the joint. Then, if the object mass is exceeded, the forces applied to the object by the fingers will be increased so that they are not lost or separated during transport along the path.

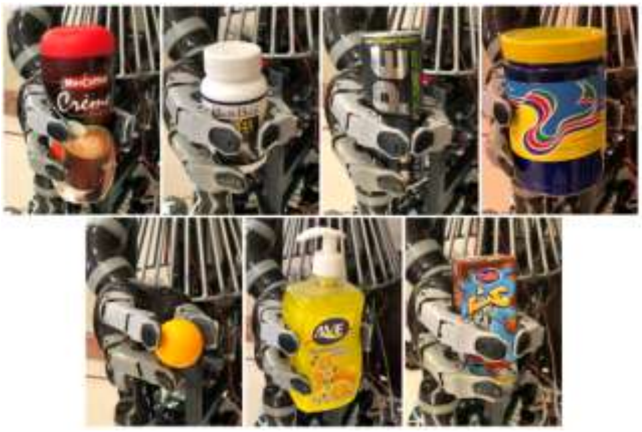

Fig. 13. Grasping object with differences in geometry and mass by the robotic arm.

In contrast, the position of each joint and the EE is measured simultaneously. In this case, the desired direction of travel from the primary to the secondary point will be reversed to the kinematics to obtain the detailed angles needed at any given moment along the route.

Figures 14, 16, 18, 20, 22 and 24, compare the position and orientation diagrams of the EE in the ideal state with the robotic arm moving the object along the path, along the X, Y, Z directions, and around them. The axis is compared. Blue line is ideal mode and red line is real robot.

In figures 15, 17, 19, 21, 23 and 25 the diagrams show the position and orientation errors of the EE in the original state that ideally designed with the robotic arm moving the object along the path, along the X, Y, Z directions, and around that axis.

As can be seen in the diagrams to verify and ensure the non-slip and shifting position of the EE of the arm, after reaching the end of the path, it remains fixed at the end of the path when there is no discernible difference in the graphs of the object as in the ideal case. This is a testament to the validity of the experiment.

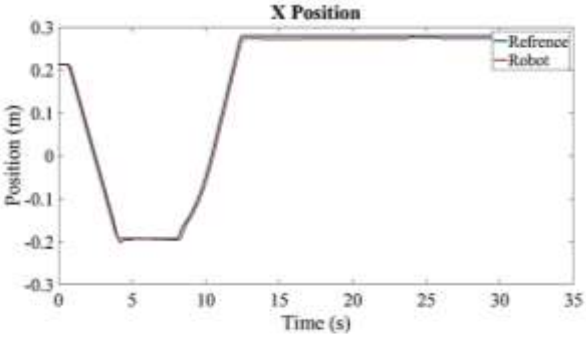

Fig. 14. Comparison of the end-effector diagram of the ideal position and

the actual position of the object in X position

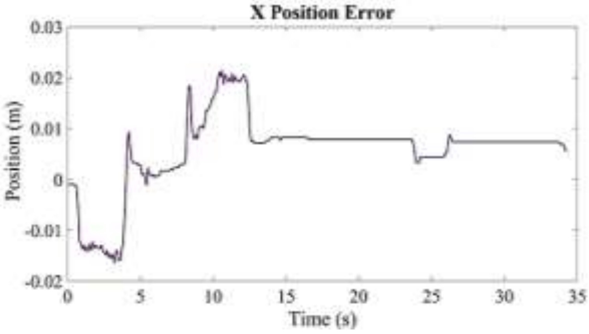

Fig. 15. Error diagram of the ideal position of the end-effector with the actual of the object X position

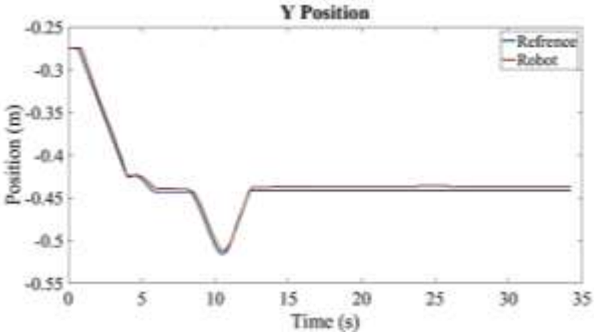

Fig. 16. Comparison of the end-effector diagram of the ideal position and the actual position of the object in Y position

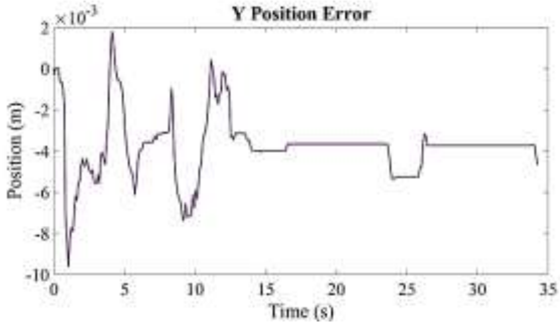

Fig. 17. Error diagram of the ideal position of the end-effector with the actual of the object Y position

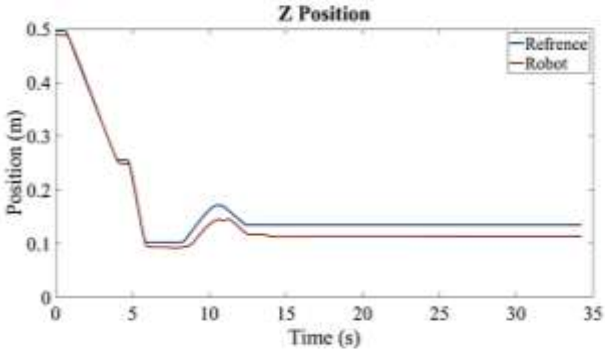

Fig. 18. Comparison of the end-effector diagram of the ideal position and the actual position of the object in Z position

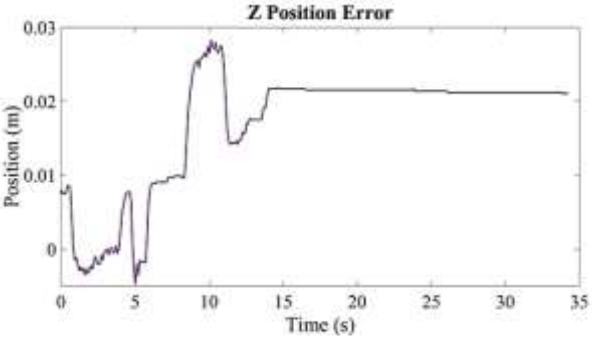

Fig. 19. Error diagram of the ideal position of the end-effector with the actual of the object Z position

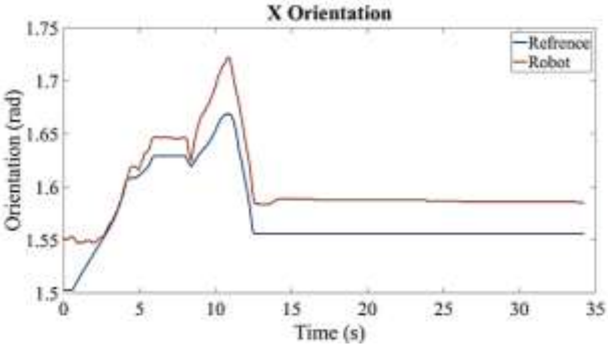

Fig. 20. Comparison of the end-effector diagram of the ideal position and the actual position of the object in X orientation

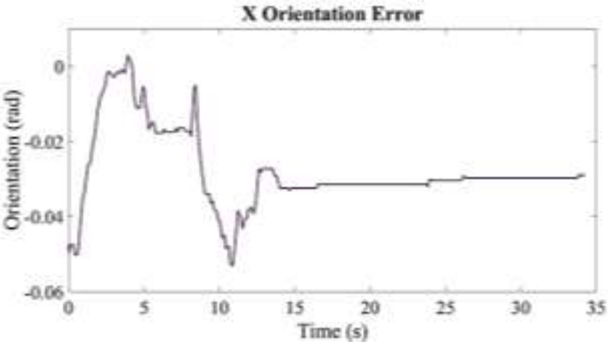

Fig. 21. Error diagram of the ideal position of the end-effector with the actual of the object in X orientation

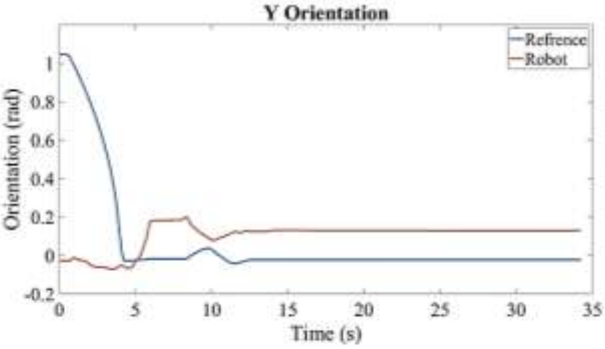

Fig. 22. Comparison of the end-effector diagram of the ideal position and the actual position of the object in Y orientation

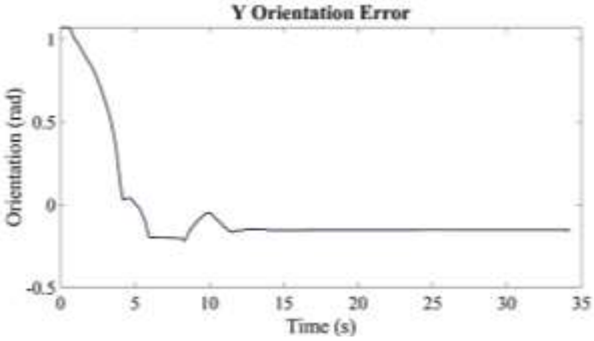

Fig. 23. Error diagram of the ideal position of the end-effector with the actual of the object in Y orientation

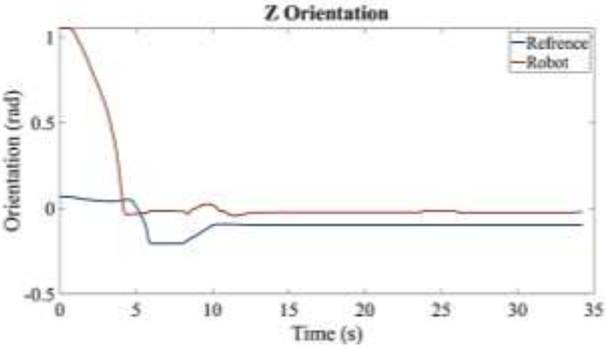

Fig. 24. Comparison of the end-effector diagram of the ideal position and the actual position of the object in Z orientation

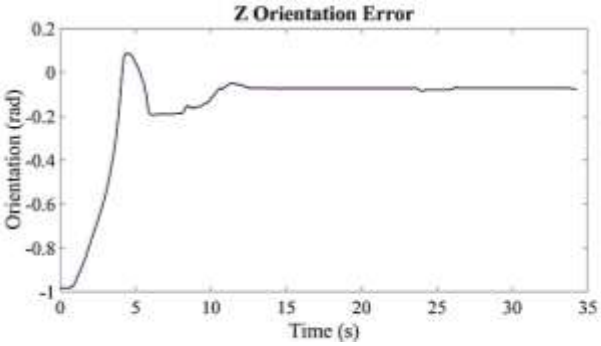

Fig. 25. Error diagram of the ideal position of the end-effector with the actual of the object in Z orientation

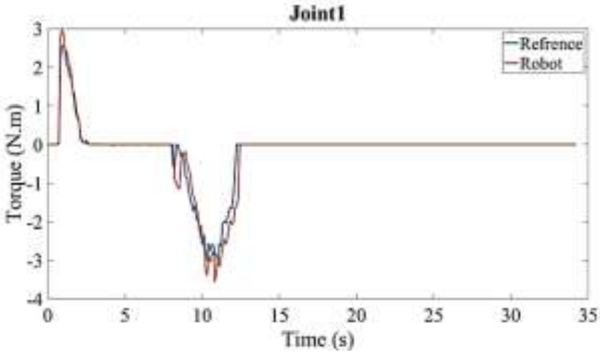

Fig. 26. Comparison of the torque diagram of arm number one in ideal and actual body displacement mode

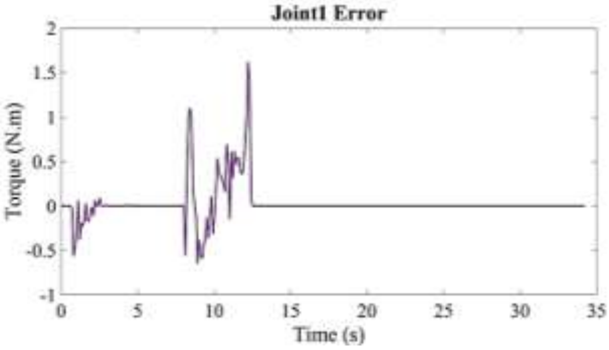

Fig. 27. Error diagram for joint torque number one

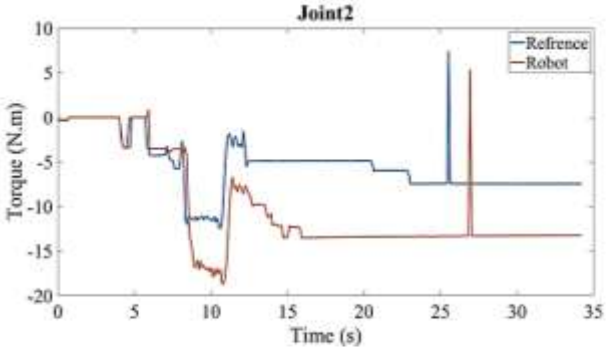

Fig. 28. Comparison of the torque diagram of arm number two in ideal and actual body displacement mode

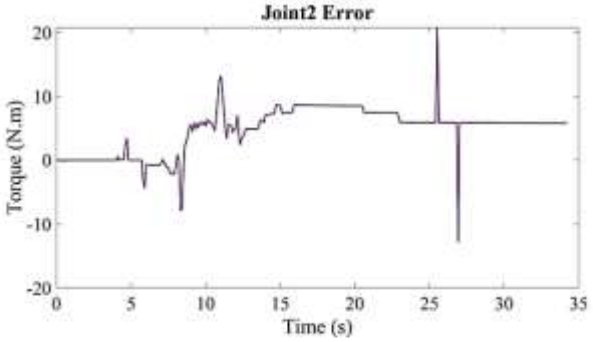

Fig. 29. Error diagram for joint torque number two

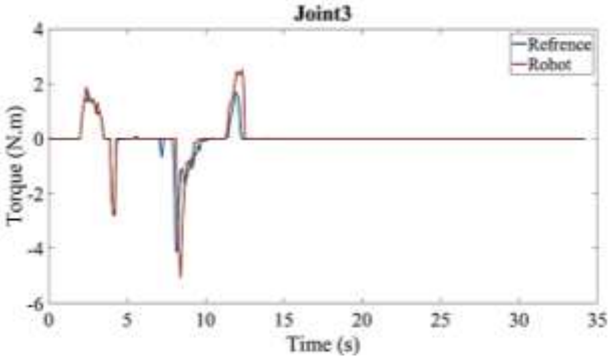

Fig. 30. Comparison of the torque diagram of arm number one in ideal and actual body displacement mode

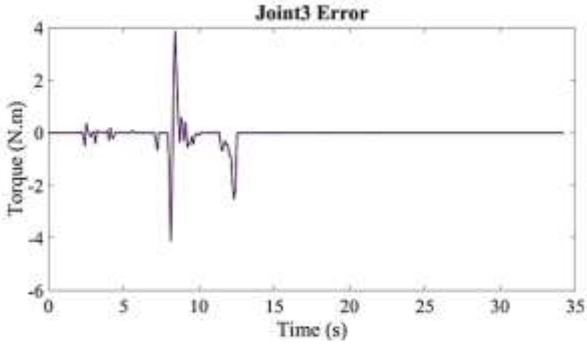

Fig. 31. Error diagram for joint torque number three

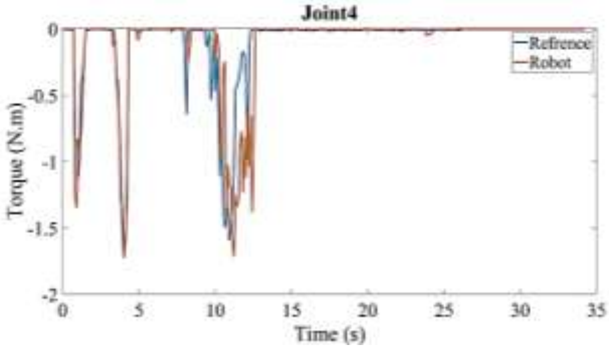

Fig. 32. Comparison of the torque diagram of arm number four in ideal and actual body displacement mode

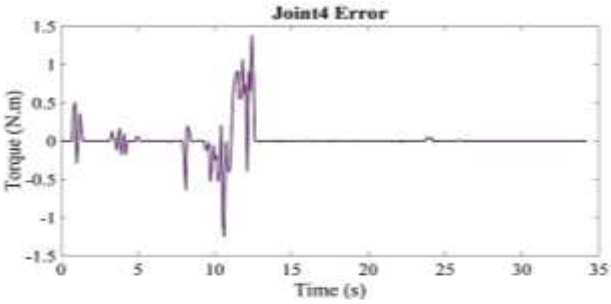

Fig. 33. Error diagram for joint torque number four

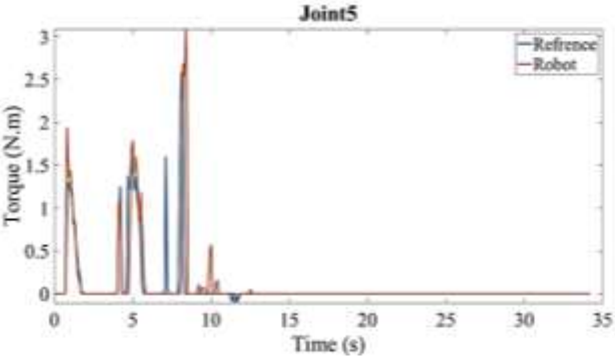

Fig. 34. Comparison of the torque diagram of arm number five in ideal and actual body displacement mode

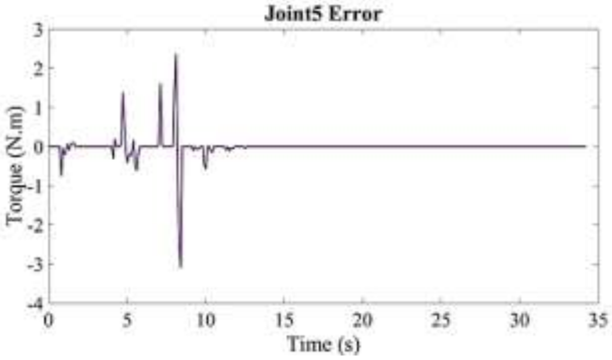

Fig. 35. Error diagram for joint torque number five

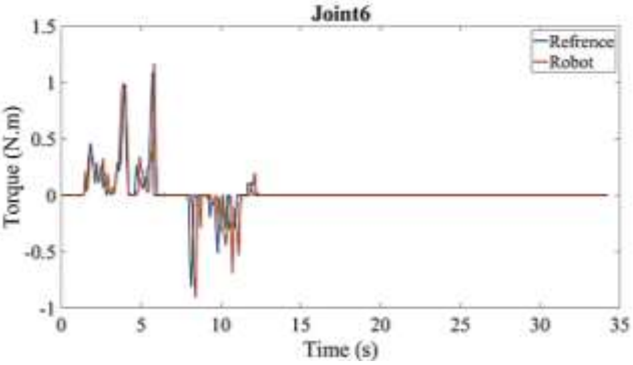

Fig. 36. Comparison of the torque diagram of arm number six in ideal and actual body displacement mode

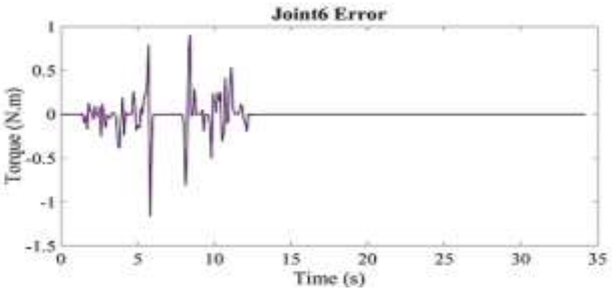

Fig. 37. Error diagram for joint torque number six

Finally, to confirm the work done in this paper, which is the control of the force exerted on the grasp and all the robot joints, we examine figures 26, 28, 30, 32, 34 and 36, which examine the torque of each joint separately. In figures 27, 29, 31, 33, 35 and 37, the error diagrams show the torque of each joint ideally with the actual state of the robotic arm.

In most joints, as can be seen in the figures, the results are very close and follow a similar behaviour in terms of comparing torque to the ideal robot position. In figure 28, joint2 diagram, the ideal torque with the actual mode of movement of the object there is some difference in torque and the delay in the applied torque due to the pressure applied on the 2nd joint, during removal and holding. It is the body. Since the torque applied to this joint does not have much effect on other joints and does not disturb the original work, so this amount of difference can be justified.

5.Conclusion

The main goal of this paper is moving the robot with grasped objects, for pick and place task, and tried to improve its performance by computed torque control for its grasp and robot arm. While performing the task, according to the experiments performed it can be stated that the main objectives have been achieved. The conclusions can be summarized as follows:

As an industrial worker in the field of rehabilitation, although the robot can be used publicly for routine and medical work that requires high precision, this article is the basis of projects that will be used in the future.

We understood the robot's intrinsic performance and how it was structured and implemented a control that calculated torque to allow tracking of the desired paths with the highest accuracy available, although robotic model integration was also required.

The calculations for the latter model are based on the Euler-Lagrange method of calculating the internal energy. Afterwards, the results were analysed to ensure the models estimated with the control program.

PID control was used in this study to control position control for its grasp and robotic arm. The amount of force exerted by the fingers of the robot is obtained using a flow sensor.

In this study, to adaptation grasping during the test, it required that in addition to controlling the torque of the robot, torque control was applied to all finger joints.

Finally, this study can be controlled by considering the torque applied to the EE and all the joints of the robot arm by the object so that the object with the optimal grip and torque control in all the robot joints can achieve the desired path without initially, along the way or at the end of the object, release the robotic armour with excessive pressure (Incompatible with physical conditions of objects) grasping cause damage to the object.

References

[1] Feil-Seifer, David, and Maja J. Mataric. "Defining socially assistive robotics." 9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005. IEEE, (2005).

[2] Van der Loos, HF Machiel, David J. Reinkensmeyer, and Eugenio Guglielmelli. "Rehabilitation and health care robotics." Springer handbook of robotics. Springer, Cham, pp. 1685-1728, (2016).

[3] Kermanshahani, Amir Hossein, and Farzad Cheraghpour Samavati. "Design, Analysis and Control of the 4 Fingers Rehabilitation Robot." Journal of Rehabilitation Sciences & Research, Vol.6, No.4, pp. 160-168, (2019).

[4] Udupa, Sumukha, Vineet R. Kamat, and Carol C. Menassa. "Shared autonomy in assistive mobile robots: a review", Disability and Rehabilitation: Assistive Technology, pp. 1-22, (2021).

[5] Santhanaraj, Karthik Kumar, M. M. Ramya, and D. Dinakaran, "A survey of assistive robots and systems for elderly care", Journal of Enabling Technologies, Vol. 5, No.1, pp. 66-72, (2021).

[6] Pană, Cristina Floriana, et al. "Fuzzy Control of the Robotic Arm for a Smart Electric Wheelchair to Assist People with Movement Disabilities", 2021 22nd International Carpathian Control Conference (ICCC), IEEE, (2021).

[7] Shen, Yang, Peter Walker Ferguson, and Jacob Rosen, "Upper limb exoskeleton systems—overview", Wearable Robotics, Academic Press, pp. 1-22, (2020).

[8] Cheraghpour, Farzad, et al. "FARAT1: an Upper Body Exoskeleton Robot", 2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM), IEEE, (2017).

[9] Shi, Di, et al. "A review on lower limb rehabilitation exoskeleton robots", Chinese Journal of Mechanical Engineering, Vol.32, No.1, pp. 1-11, (2019).

[10] Wu, Wen-Chi Vivian, Rong-Jyue Wang, and Yan-An Enya Jou, "Application of Educational Robots in the Elderly English Vocabulary Learning", 2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT), IEEE, (2020).

[11] Seo, Jungwon, Mark Yim, and Vijay Kumar, "A theory on grasping objects using effectors with curved contact surfaces and its application to whole-arm grasping", *The International Journal of Robotics Research,* Vol.35, No. 9, pp. 1080-1102, (2016).

[12] Kamali, M., S. Ali A. Moosavian, and F. Cheraghpour, "Improving grasp capabilities of KNTU hand using position & force sensors", 2014 22nd Iranian Conference on Electrical Engineering (ICEE), IEEE, (2014).

[13] Mesgari, Hamed, Farzad Cheraghpour Samavati, and S. Ali A. Moosavian, "Finding the Best Grasping Point in Object Manipulation Tasks-A Comparison between GA and PSO Methods", The 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Netherlands, pp. 199-204, (2011).

[14] Campeau-Lecours, Alexandre, et al. "Kinova modular robot arms for service robotics applications", Rapid Automation: Concepts, Methodologies, Tools, and Applications, IGI global, pp. 693-719, (2019).

[15] https://www.kinovarobotics.com, KINOVA™ Ultra lightweight robotic arm user guide, “Jaco^2 6 DOF Advanced specification guide”, Kinova Inc., (2017).

[16] Fathzadeh, R., et al. "MRL@ Home 2014 Team Description Paper", https://fei.edu.br/rcs/2014/TeamDescriptionPapers/RoboCup@Home/MRL_AtHome_TDP_2014.pdf, (2014).

[17] Craig, John J, Introduction to robotics: mechanics and control, 3/E. Pearson Education India, (2009).

[18] Lewis, F. L. "C, T. Abdallah, DM Dawson, Control of Robot Manipulators", pp.136-175, (1993).