A comparison study on the dynamic control of OpenMANIPULATOR-X by PD with gravity compensation tuned by oscillation damping based on the phase-trajectory-length concept

1[st] Amirhossein Dadbin, 2[nd] Ahmad Kalhor, 3[rd] Mehdi Tale Masouleh
Human and Robot Interaction Laboratory
University of Tehran
Tehran, Iran
amirhosseindadbin@ut.ac.ir, akalhor@ut.ac.ir, m.t.masouleh@ut.ac.ir

***Abstract*—In this paper, the dynamic control of a 4-DOF serial manipulator, the so-called OpenMANIPULATOR-x, is investigated by means of different performance indices, including, among others, oscillation damping. To do so, the kinematics and dynamics equations are obtained from a systematic approach where both models are verified by simulating the under study robot in Simscape. The stability of the applied controller, which is a PD with gravity compensation controller, is investigated, and it reveals that it is asymptotically stable. Thereafter, the coefficient of the latter PD controller is first optimized by means of different performance indices, namely, IAE, ISE, ITAE, ITSE then, a new criterion called oscillation damping, which is based on the optimization of a cost function defined on the phase trajectory length concept is used in order to evaluate the performance of the implemented controller. The obtained results revealed that the step response of the oscillation damping controller eliminates the overshoot, but it is slower than ones tuned by other performance indices.**

***Index Terms*—Dynamics, kinematics, OpenMANIPULATOR-X, oscillation damping, PD with gravity compensation**

I. INTRODUCTION

For decades, serial manipulators have been used in industries, and nowadays, with significant progress in computers' computational powers and reduced sizes, employing robots is proliferating. Thus, controlling them to behave as the user expected is needed, and mathematical modeling is essential in most control approaches. The under study robot, OpenManipulator-X, is an open source and low-cost 4-Degree-Of-Freedom (DOF) serial manipulator with revolute joints which can be controlled to the so-called Robot Operating System (ROS). DYNAMIXEL XM430-W350-T motors used in OpenMANIPULATOR-X are capable of performing current-mode control, which is equivalent to joints' torque control. Moreover, being an open-hardware-oriented platform makes OpenMANIPULATOR-X a good choice for researchers to enhance their mathematical modeling and control skills [1]. The first step in modeling serial manipulators is using Denavit-Hartenberg (DH) parameters as a standard convention to completely and uniquely describe kinematic chains and extract forward kinematics of kinematic chains [2]. The Euler-Lagrange formulation for extracting the dynamic model of a serial manipulator is commonly used in literature [3] [4] [5] [6]. Kinematic analysis of OpenMANIPULATOR-X is discussed in [7]. A simple way to control serial manipulators is using PD-with-gravity-compensation(PDGC). A good review of this approach can be found in [8]. In [9], PD-with-gravity-compensation is used in the presence of input saturation. Tuning PID gains by optimizing integral of square error using several optimization methods are compared in [10]. In [11], different performance indices are used as a cost function, and particle swarm optimization is used to tune PID gains. This paper is organized as follows. Section II presents the mathematical model of OpenMANIPULATOR-X, including forwarding and inverse kinematics and Lagrange dynamics. Section III is devoted to the control of the OpenMANIPULATOR-X, and formulation for oscillation damping based on the phase-trajectory-length concept is given. In Section IV results of simulation in simscape are shown. Finally, in Section V the paper concludes by providing some hints as ongoing works.

II. MATHEMATICAL MODELING

The under study robotic arm is shown in Fig. 1. Using Denavit-Hartenberg (DH) parameterization [4] for the forward kinematics, the robotic arm is modeled as a serial chain RRRR manipulator. Then inverse kinematics of the arm is extracted using the position of End-Effector(EE) equations. Then applying the Euler-Lagrange method for the dynamic model leads to derive the dynamic model of the arm with some simplification assumptions such as there is no friction in revolute joints.

A. Kinematics of OpenMANIPULATOR-X

Dimensions and DH frames of the OpenManipulatorX are shown in Fig. 1. DH parameters of robot are given in Table I. Parameters a, α, d and θ are in standard DH convention and $\theta_1, \theta_2, \theta_3$ and θ_4 are joints revolution angels. $\theta_{i,0}$ refers to the angle of the i^{th} joint at the initial configuration shown in Fig. 1.

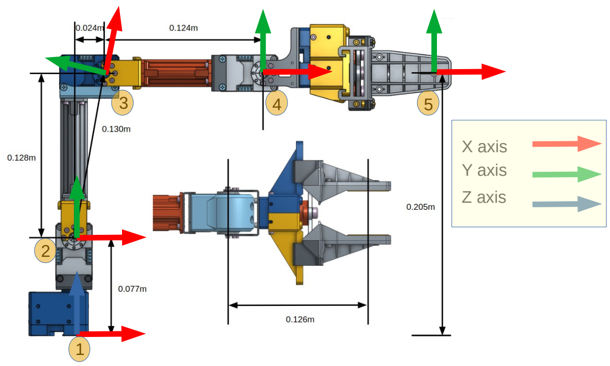


Fig. 1: Dimensions and DH frames of the OpenMANIPULATOR-X.

TABLE I: DH parameters of the OpenMANIPULATOR-X.

i	a_i	α_i	d_i	θ_i	$\theta_{i,0}$
1	0	$\frac{\pi}{2}$	0.077	θ_1	0
2	0.13	0	0	θ_2	$\arctan(128/24)$
3	0.124	0	0	θ_3	$-\arctan(128/24)$
4	0.126	0	0	θ_4	0

1) Forward kinematics: Consider the rotation matrix $\boldsymbol{Q}_i$ which rotates the $(i+1)^{\text{th}}$ joint's frame in order to be aligned with the i^{th} joint's frame:

$$[\boldsymbol{p}]_i = \boldsymbol{Q}_i[\boldsymbol{p}]_{i+1} \tag{1}$$

where $[p]_i$ is the representation of point p in the i^{th} joint's frame. Based on DH parameters $\boldsymbol{Q}_i$ is represented as follows [12].

$$\boldsymbol{Q}_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \tag{2}$$

Vector $\mathbf{a}_i$ which transfers the origin of the i^{th} joint's frame to the origin of the $(i+1)^{\text{th}}$ joint's frame is given by [12].

$$[\mathbf{a}_i]_i = \mathbf{a}_i = \begin{bmatrix} a_i\cos\theta_i & a_i\sin\theta_i & b_i \end{bmatrix}^T \tag{3}$$

Thus position P_{EE} and orientation Q_{EE} of the EE of the arm can be obtained easily:

$$\boldsymbol{Q}_{EE} = \boldsymbol{Q}_1\boldsymbol{Q}_2\boldsymbol{Q}_3\boldsymbol{Q}_4 \tag{4}$$

$$\begin{aligned}[\boldsymbol{P}_{EE}]_1 = \begin{bmatrix} X_{\text{EE}} & Y_{\text{EE}} & Z_{\text{EE}} \end{bmatrix}^T &= [\mathbf{a}_1]_1 + [\mathbf{a}_2]_1 + [\mathbf{a}_3]_1 + [\mathbf{a}_4]_1 \\ &= \mathbf{a}_1 + \boldsymbol{Q}_1\mathbf{a}_2 + \boldsymbol{Q}_1\boldsymbol{Q}_2\mathbf{a}_3 + \boldsymbol{Q}_1\boldsymbol{Q}_2\boldsymbol{Q}_3\mathbf{a}_4\end{aligned} \tag{5}$$

With some simplifications, one has:

$$[\boldsymbol{P}_{EE}]_1 = \begin{bmatrix} \mathbf{c}(\theta_1)[0.126\mathbf{c}(\theta_{234}) + 0.124\mathbf{c}(\theta_{23}) + 0.13\mathbf{c}(\theta_2)] \\ \mathbf{s}(\theta_1)[0.126\mathbf{c}(\theta_{234}) + 0.124\mathbf{c}(\theta_{23}) + 0.13\mathbf{c}(\theta_2)] \\ 0.077 + 0.126\mathbf{s}(\theta_{234}) + 0.124\mathbf{s}(\theta_{23}) + 0.13\mathbf{s}(\theta_2) \end{bmatrix} \tag{6}$$

where $\mathbf{c}$ stands for cos and $\mathbf{s}$ stands for sin. Moreover, it is assumed that $\theta_{234} = \theta_2 + \theta_3 + \theta_4$ and $\theta_{23} = \theta_2 + \theta_3$.

2) Inverse kinematics: Each joint's angle of the OpenMANIPULATOR-X can be calculated by assumpting that the position of the EE $(X_{\text{EE}}, Y_{\text{EE}}, Z_{\text{EE}})$ and its pitch angle (θ_{234}) is given. According to the forward kinematics expressed in Eq. (6), the following can be written for θ_1:

$$\theta_1 = \text{atan2}(Y_{\text{EE}}, X_{\text{EE}}) \tag{7}$$

After calculating θ_1, it follows that:

$$\begin{cases} A - 0.13\cos\theta_2 = 0.124\cos(\theta_2+\theta_3) \\ B - 0.13\sin\theta_2 = 0.124\sin(\theta_2+\theta_3) \end{cases} \tag{8}$$

where B can be calculating by considering Z_{EE}:

$$B = Z_{\text{EE}} - 0.077 - 0.126\sin(\theta_2+\theta_3+\theta_4) \tag{9}$$

In order to calculate A, two cases should be taken into account. If $\cos\theta_1 \neq 0$, by considering X_{EE}:

$$A = \frac{X_{\text{EE}}}{\cos\theta_1} - 0.126\cos(\theta_2+\theta_3+\theta_4) \tag{10}$$

and if $\sin\theta_1 \neq 0$ considering Y_{EE}, the above can be expressed as follows:

$$A = \frac{Y_{\text{EE}}}{\sin\theta_1} - 0.126\cos(\theta_2+\theta_3+\theta_4)\ . \tag{11}$$

By squaring both sides of Eq. (8) and summing them, it leads to:

$$a\cos\theta_2 + b\sin\theta_2 + c = 0 \tag{12}$$

where

$$a = 0.26A \quad , \quad b = 0.26B \tag{13}$$

$$c = 0.124^2 - A^2 - B^2 - 0.13^2\ . \tag{14}$$

Now by considering the so-called tangent half-angle formula substitution, $t = \tan\frac{\theta_2}{2}$ which required to have:

$$\sin\theta_2 = \frac{2t}{1+t^2} \quad , \quad \cos\theta_2 = \frac{1-t^2}{1+t^2}\ . \tag{15}$$

Eq. (12) can be rewritten as:

$$(c-a)t^2 + 2bt + a + c = 0\ . \tag{16}$$

If $a = c$ then

$$t_1 = t_2 = -\frac{a}{b} \tag{17}$$

and if $a \neq c$

$$t_{1,2} = \frac{-b \pm \sqrt{a^2+b^2-c^2}}{c-a} \tag{18}$$

are two possible answers for t. Then two possible answers for θ_2 can be calculated using the following equations:

$$\theta_2^+ = \text{atan2}(\frac{2t_1}{1+t_1^2}, \frac{1-t_1^2}{1+t_1^2}) \tag{19}$$

$$\theta_2^- = \text{atan2}(\frac{2t_2}{1+t_2^2}, \frac{1-t_2^2}{1+t_2^2}) \tag{20}$$

Now considering:

$$C = \cos(\theta_2 + \theta_3) = \frac{A - 0.13\cos\theta_2}{0.124} \quad (21)$$

$$D = \sin(\theta_2 + \theta_3) = \frac{B - 0.13\sin\theta_2}{0.124} \quad (22)$$

Reaching this step θ_3 and θ_4 can be calculated readily:

$$\theta_3 = \text{atan2}(D, C) - \theta_2 \quad (23)$$

$$\theta_4 = \text{pitch}_{\text{EE}} - \theta_2 - \theta_3 = \theta_{234} - \theta_2 - \theta_3 \quad (24)$$

B. Dynamics of OpenMANIPULATOR-X

In this part, the dynamics of OpenMANIPULATOR-X are obtained using the Euler-Lagrange method [3]. The kinetic energy T and the potential energy of the system V are the sums of the kinetic energies and the potential energies of all the manipulator's Links. ($T = \sum_{i=1}^{4} T_i \quad ; \quad V = \sum_{i=1}^{4} V_i$)

$$T_i = \frac{1}{2} m_i \|\dot{\boldsymbol{c}}_i\|^2 + \frac{1}{2}\boldsymbol{\omega}_i^T \boldsymbol{I}_i \boldsymbol{\omega}_i \quad , \quad \dot{\boldsymbol{c}}_i = \boldsymbol{N}_i \dot{\boldsymbol{\theta}} \quad (25)$$

$$[\boldsymbol{N}_i]_1 = \begin{bmatrix} \boldsymbol{e}_1 \times \boldsymbol{r}_{1i} & \ldots & \boldsymbol{e}_i \times \boldsymbol{r}_{ii} & 0_{3\times1} & \ldots & 0_{3\times1} \end{bmatrix}_{3\times n} \quad (26)$$

where m_i is the mass of link i, $\boldsymbol{c}_i$ is the center of gravity (CoG) of link i and $\boldsymbol{I}_i$ is the inertia matrix of link i presented in the i $^{\text{th}}$ link's DH frame, $\boldsymbol{r}_{ij}$ is the translation vector which moves the origin of the i$^{\text{th}}$ frame to the CoG of the j$^{\text{th}}$ link's frame and presented in the DH frame of the first link, $\boldsymbol{e}_i$ is the revolution axis of the i$^{\text{th}}$ revolute joint presented in the DH frame of the first link. The rotation axis of each joint presented in its local DH frame is $\boldsymbol{z} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ and one has:

$$\boldsymbol{e}_1 = \boldsymbol{z} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (27)$$

$$\boldsymbol{e}_i = (\prod_{k=1}^{i-1} \boldsymbol{Q}_k)\boldsymbol{z} = \begin{bmatrix} \sin\theta_1 & -\cos\theta_1 & 0 \end{bmatrix}^T \;\text{;for } i \neq 1 \quad (28)$$

While each joint of the manipulator revolutes around its z-axis, the rotation matrix is given by [3]:

$$\boldsymbol{Q}_z(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (29)$$

so one has:

$$\boldsymbol{r}_{11} = \boldsymbol{Q}_z(\theta_1 - \theta_1^{\text{init}})\boldsymbol{c}_1 \quad (30)$$

$$\boldsymbol{r}_{ij} = \sum_{k=i}^{j-1} [\boldsymbol{a}_k]_1 + [\boldsymbol{Q}_z(\theta_j - \theta_j^{\text{init}})\boldsymbol{c}_j]_1; \text{ for } j \neq 1 \text{ and } i \leq j \quad (31)$$

where $\theta_1^{\text{init}} = 0$, $\theta_2^{\text{init}} = \arctan(128/24)$, $\theta_3^{\text{init}} = -\arctan(128/24)$, $\theta_4^{\text{init}} = 0$ and

$$[\boldsymbol{Q}_z(\theta_j - \theta_j^{\text{init}})\vec{\boldsymbol{c}}_j]_1 = (\prod_{k=1}^{j-1} \boldsymbol{Q}_k)\boldsymbol{Q}_z(\theta_j - \theta_j^{\text{init}})\vec{\boldsymbol{c}}_j. \quad (32)$$

On the other hand, one has:

$$\boldsymbol{\omega}_i = \boldsymbol{W}_i \dot{\boldsymbol{\theta}} \quad (33)$$

where $\boldsymbol{W}_i = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \ldots & \boldsymbol{e}_i & 0_{3\times1} & \ldots & 0_{3\times1} \end{bmatrix}$. It is simple to represent $\boldsymbol{W}_i$ in i$^{\text{th}}$ joint DH frame.

$$[\boldsymbol{W}_1]_1 = \begin{bmatrix} \boldsymbol{z} & 0 & 0 & 0 \end{bmatrix} \quad (34)$$

$$[\boldsymbol{W}_2]_2 = \begin{bmatrix} \boldsymbol{Q}_1^T \boldsymbol{z} & \boldsymbol{z} & 0 & 0 \end{bmatrix} \quad (35)$$

$$[\boldsymbol{W}_3]_3 = \begin{bmatrix} \boldsymbol{Q}_2^T \boldsymbol{Q}_1^T \boldsymbol{z} & \boldsymbol{Q}_2^T \boldsymbol{z} & \boldsymbol{z} & 0 \end{bmatrix} \quad (36)$$

$$[\boldsymbol{W}_4]_4 = \begin{bmatrix} \boldsymbol{Q}_3^T \boldsymbol{Q}_2^T \boldsymbol{Q}_1^T \boldsymbol{z} & \boldsymbol{Q}_3^T \boldsymbol{Q}_2^T \boldsymbol{z} & \boldsymbol{Q}_3^T \boldsymbol{z} & \boldsymbol{z} \end{bmatrix} \quad (37)$$

By defining

$$\boldsymbol{\theta} \triangleq \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}^T \quad (38)$$

$$\boldsymbol{M}_i(\boldsymbol{\theta}) \triangleq m_i \boldsymbol{N}_i^T \boldsymbol{N}_i + [\boldsymbol{W}_I]_i^T \boldsymbol{Q}_z(\theta_i - \theta_i^{init}) \boldsymbol{I}_i \boldsymbol{Q}_z(\theta_i - \theta_i^{init})^T [\boldsymbol{W}_i]_i \quad (39)$$

one can formulate kinetic energy of each joint as:

$$T_i = \frac{1}{2} \dot{\boldsymbol{\theta}}^T \boldsymbol{M}_i \dot{\boldsymbol{\theta}}. \quad (40)$$

Calculating potential energy of each link is straight forward, which is:

$$V_i = m_i g \boldsymbol{h}_i \quad (41)$$

where g = 9.81 and $\boldsymbol{h}_i = \boldsymbol{r}_{1i}^T \boldsymbol{z}$.

Now, the Lagrangian term can be calculated and the dynamic model of manipulator can be expressed as follows:

$$\mathcal{L} = T - V \Rightarrow \boldsymbol{\tau} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{\theta}}}\right) - \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \quad (42)$$

$$\Rightarrow \boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{M}}\dot{\boldsymbol{\theta}} - \frac{\partial T}{\partial \boldsymbol{\theta}} + \frac{\partial V}{\partial \boldsymbol{\theta}} \quad (43)$$

where $\boldsymbol{M}(\boldsymbol{\theta}) = \sum_{i=1}^{4} \boldsymbol{M}_i(\boldsymbol{\theta})$.

It is worth mentioning that it is very common to represent the dynamic model of the manipulator using Christoffel Symbols:

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) \quad (44)$$

where

$$\boldsymbol{B}(\boldsymbol{\theta}) = [b_{ij}] = M(\boldsymbol{\theta}) \in \mathcal{R}^{4\times4} \quad (45)$$

$$\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = [c_{ij}] \in \mathcal{R}^{4\times4} \quad (46)$$

$$\boldsymbol{G}(\boldsymbol{\theta}) = \left(\frac{\partial V}{\partial \boldsymbol{\theta}}\right)^T \in \mathcal{R}^{4\times1} \; . \quad (47)$$

While the choice of matrix $\boldsymbol{C}$ is not unique, a particular choice can be obtainted to have $\boldsymbol{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \triangleq \dot{\boldsymbol{B}}(\boldsymbol{\theta}) - 2\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ a skew-symmertry matrix. The latter to:

$$c_{ij} = \sum_{k=1}^{n} c_{ijk} \dot{\theta}_i \quad (48)$$

where

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial \theta_k} + \frac{\partial b_{ik}}{\partial \theta_j} - \frac{\partial b_{jk}}{\partial \theta_i}\right) \; . \quad (49)$$

So the expression of the generic element of the matrix $\boldsymbol{N}$ is:

$$n_{ij} = \dot{b}_{ij} - 2c_{ij} = \sum_{k=1}^{n}\left(\frac{\partial b_{jk}}{\partial \theta_i} - \frac{\partial b_{ik}}{\partial \theta_j}\right)\dot{\theta}_k \quad (50)$$

and can be seen that $n_{ij} = -n_{ji}$.

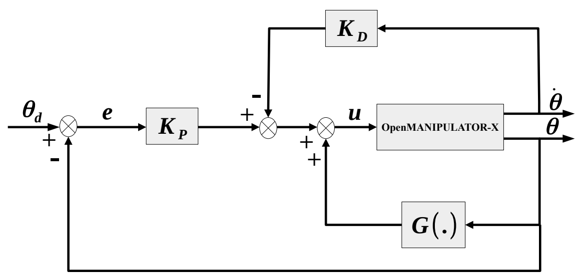


Fig. 2: Block scheme of joint space PD control with gravity compensation.

III. Control of OpenMANIPULATOR-X

In this section, centralized control of the manipulator's joint angles is used to stabilize the manipulator and eliminate the error. It will be shown that a PD controller with gravity compensation can result to achieve the goal [4]. Then several performance indices are used to tune the coefficients of the PD controller.

A. PD control with gravity compensation

Let $\boldsymbol{\theta}_d \triangleq \begin{bmatrix}\theta_{1,d} & \theta_{2,d} & \theta_{3,d} & \theta_{4,d}\end{bmatrix}^T$ be the vector of desired joint angles of the manipulator and $\boldsymbol{e} \triangleq \boldsymbol{\theta}_d - \boldsymbol{\theta}$ be the error between actual joint's angles and desired angles. Let choose the following positive definite function V as Lyapunov function candidate:

$$V(\dot{\boldsymbol{\theta}}, \boldsymbol{e}) = \frac{1}{2}\dot{\boldsymbol{\theta}}^T \boldsymbol{B}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \frac{1}{2}\boldsymbol{e}^T \boldsymbol{K}_p \boldsymbol{e} > 0 \qquad \forall \dot{\boldsymbol{\theta}}, \boldsymbol{e} \neq 0 \quad (51)$$

where $\boldsymbol{K}_p$ is an $(n \times n)$ symmetric positive definite matrix. Differentiating Eq. (51) with respect to time and recalling that $\boldsymbol{\theta_d}$ is constant (step signal) yeilds:

$$\dot{V} = \dot{\boldsymbol{\theta}}^T \boldsymbol{B}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \frac{1}{2}\dot{\boldsymbol{\theta}}^T \dot{\boldsymbol{B}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}^T \boldsymbol{K}_p \boldsymbol{e}. \qquad (52)$$

Using Eq. (44) and control structure shown in Fig. 2, one can solve $\boldsymbol{B}\ddot{\boldsymbol{q}}$ and substituting it in Eq. (52) which gives:

$$\dot{V} = \frac{1}{2}\dot{\boldsymbol{\theta}}^T \left(\dot{\boldsymbol{B}}(\boldsymbol{\theta}) - 2\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\right)\dot{\boldsymbol{\theta}} + \dot{\boldsymbol{\theta}}^T \left(\boldsymbol{u} - \boldsymbol{G}(\boldsymbol{\theta}) - \boldsymbol{K_P}\boldsymbol{e}\right). \qquad (53)$$

The first term on the right hand side is null since $\boldsymbol{N} = \dot{\boldsymbol{B}} - \boldsymbol{2C}$ is skew symmetric. Then by choosing

$$\boldsymbol{u} = \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{K}_P\boldsymbol{e} - \boldsymbol{K}_D\dot{\boldsymbol{\theta}} \qquad (54)$$

leads to a semi-definite $\dot{V}$;

$$\dot{V} = -\dot{\boldsymbol{\theta}}^{\boldsymbol{T}}(\boldsymbol{K_D})\dot{\boldsymbol{\theta}} \qquad (55)$$

where $\boldsymbol{K_D}$ and $\boldsymbol{K_P}$ are arbitrary positive definite matrices. According to this, V is decreasing function, and $\dot{V} \equiv 0$ only if $\dot{\boldsymbol{\theta}} \equiv 0$. The controlled system's dynamics using control law (54) is given by:

$$\boldsymbol{B}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{K}_P\boldsymbol{e} - \boldsymbol{K}_D\dot{\boldsymbol{\theta}} \qquad (56)$$

Thus at the equilibrium where $\dot{\boldsymbol{\theta}} \equiv 0$ and $\ddot{\boldsymbol{\theta}} \equiv 0$ one has:

$$\boldsymbol{K}_P\boldsymbol{e} = 0 \Rightarrow \boldsymbol{e} = \boldsymbol{\theta}_d - \boldsymbol{\theta} = 0 \qquad (57)$$

Since $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are arbitrary positive definite matrices they can be used as degrees of freedom to achieve other goals. In other words, one can tune these matrices using many available algorithms such as minimizing performance metrics like Integral Square Error (ISE), Integral Absolute Error (IAE), Integral time Squared Error (ITSE), Integral time Absolute Error (ITAE), etc. by applying optimization algorithms such as Genetic algorithm(GA) [13].

B. Minimum oscillation controller (MOC)

According to approach presented in [14] and [15] consider the following system:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{g}(\boldsymbol{u}, t) \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \qquad (58)$$

which is asymptotically stable. The oscilation number index for this system is defined by:

$$r_{on}(\boldsymbol{x}(t)) = \frac{L(\boldsymbol{x}(t))}{r(\boldsymbol{x}(t))} \qquad (59)$$

where

$$L(\boldsymbol{x}(t)) = \int_{\tau=t}^{\tau=\infty} \sqrt{\boldsymbol{f}_c(\boldsymbol{x}(\tau), \tau)^T \boldsymbol{f}_c(\boldsymbol{x}(\tau), \tau)}d\tau \qquad (60)$$

$$r(\boldsymbol{x}(t)) = \|\boldsymbol{x}(t)\| \qquad (61)$$

$$\boldsymbol{f}_c = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{g}(\boldsymbol{u}, t)\ . \qquad (62)$$

In order to tune the $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ of the (54) with oscillation damping porpose based on the Phase Trajectory Length(PTL) concept, the following cost function could be considered:

$$J = \rho\boldsymbol{x}^2(t_f) + \int_{\tau=t_0}^{\tau=t_f} \sqrt{\boldsymbol{f}_c(\boldsymbol{x}(\tau), \tau)^T \boldsymbol{f}_c(\boldsymbol{x}(\tau), \tau)}d\tau \quad (63)$$

C. MOC for OpenMANIPULATOR-X

Considering Eq. (56), one has:

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{B}^{-1}(\boldsymbol{K}_P\boldsymbol{e} - \boldsymbol{K}_D\dot{\boldsymbol{\theta}} - \boldsymbol{C}\dot{\boldsymbol{\theta}}) \qquad (64)$$

$$\Rightarrow \ddot{\mathbf{e}} = \boldsymbol{B}^{-1}(\boldsymbol{K}_P - \boldsymbol{K}_D - \boldsymbol{C})\boldsymbol{e} + \phi(t) \qquad (65)$$

where $\phi(t) = B^{-1}(\boldsymbol{K}_D + C)\dot{\boldsymbol{\theta}}_d - \ddot{\boldsymbol{\theta}}_d$ which leads to:

$$\begin{cases} x_1 \triangleq \boldsymbol{e} \\ x_2 \triangleq \dot{\boldsymbol{e}} \end{cases} \Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \boldsymbol{B}^{-1}(\boldsymbol{K}_P - \boldsymbol{K}_D - \boldsymbol{C})x_1 + \phi(t) \end{cases} \qquad (66)$$

By choosing $\boldsymbol{X} \triangleq \begin{bmatrix}x_1 & x_2\end{bmatrix}^T$ one has:

$$\dot{\boldsymbol{X}} = \boldsymbol{f}_c(\boldsymbol{X}, t) \qquad (67)$$

where

$$\boldsymbol{f}_c(\boldsymbol{X}, t) = \begin{bmatrix} 0_{4\times4} & I_{4\times4} \\ \boldsymbol{B}^{-1}(\boldsymbol{K}_P - \boldsymbol{K}_D - \boldsymbol{C}) & 0_{4\times4} \end{bmatrix}\boldsymbol{X} + \begin{bmatrix} 0 \\ \phi(t) \end{bmatrix} \qquad (68)$$

Reaching this step, the cost function introduced in Eq. (63) can be used to tune PD gains for oscillation damping based on PTL concept purpose. This PD controller is referred to as MOC in this paper.

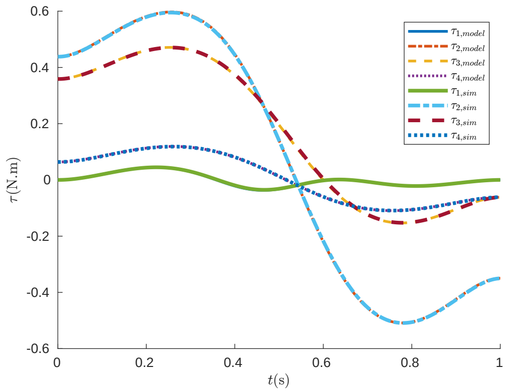


Fig. 3: Comparison of torques obtained using the mathematical model and Simscape simulation.

TABLE II: Performance indices considered for evaluating the proposed controller.

i	J	formula
1	IAE	$\int_{t=t_0}^{t=t_f} \sum_{i=1}^{4} \lvert e_i(t)\rvert dt$
2	ISE	$\int_{t=t_0}^{t=t_f} \sum_{i=1}^{4} e_i^2(t) dt$
3	ITAE	$\int_{t=t_0}^{t=t_f} \sum_{i=1}^{4} t\lvert e_i(t)\rvert dt$
4	ITSE	$\int_{t=t_0}^{t=t_f} \sum_{i=1}^{4} te_i^2(t) dt$
5	MOC	$\int_{t=t_0}^{t=t_f} \sqrt{\sum_{i=1}^{4} \dot{e}_i^2(t)} dt$

IV. RESULTS

A. Dynamic model verification

For the sake of verifying the obtained mathematical dynamic model of the understudy robot, a pick-and-place operation using a 4-5-6-7 interpolating polynomial is considered. More details about this interpolation can be found in [3]. Torques calculated for this trajectory with $\boldsymbol{\theta}_{\text{init}} = \begin{bmatrix}0 & \arctan(128/24) & -\arctan(128/24) & 0\end{bmatrix}^T$ and $\boldsymbol{\theta}_{\text{final}} = \boldsymbol{\theta}_{\text{init}} + \begin{bmatrix}\pi/4 & \pi/4 & -\pi/4 & \pi/3\end{bmatrix}^T$ using the mathematical model described in detail in this paper are compared with torques represented by the MatLab simscape model of the OpenMANIPULATOR-X. As shown in Fig. 3 the error is neglectable and depends on the MatLab solver settings.

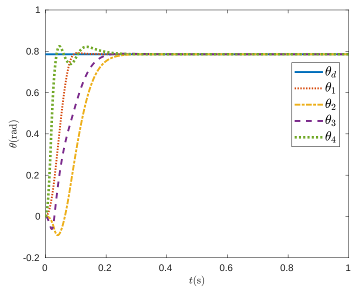
B. Tuning PD controllers

In previous sections, it has been shown that the PD controller with gravity compensation stabilizes the manipulator and error converges to zero if $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are arbitrary positive definite matrices. So these DOFs are used to optimize some performance indices. These performance indices are given in Table II. In order to decrease error at $t = t_f$ the term $\rho \sum_{i=1}^{n} e_i^2(t_f)$ is added to the cost function where ρ is a positive constant. Moreover, adding this term also decreases raise time of the system implicitly.

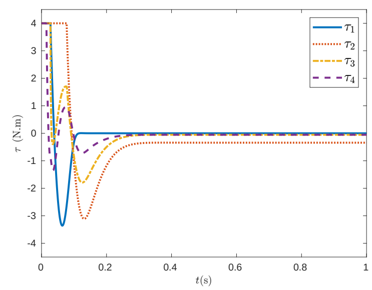
By choosing $\rho = 10^6$, $t_0 = 0$, and $t_f = 1$, five PD controllers are tuned by minimizing cost functions containing performance indices and square of error value at $t = t_f$ using Genetic Algorithm in Matlab and the desired values of joint angles are step signal with $\pi/4$ amplitude. The population size of GA was chosen to be 100, and the only constrain was that optimization variables must be positive. Other parameters of GA are based on MatLab default values. The GA ran for at least 90 generations. The gains provided by GA are reported in Table III. The aforementioned performance indices for these 5 PD controllers are compared in Table IV. It is worth mentioning that saturation is used to limit the control input in the range $-4 \leq u < 4$ (N.m) according to the OpenMANIPULATOR-X datasheet. The step response and the control input of the system tuned with IAE are shown in Fig. 4. Two joint angles have overshoot, and two others have undershoot, but the maximum overshoots and undershoots are less than those in the system, which its controller has been tuned with the ITAE criterion. Besides, in this case, and the oscillation damping one, the control input has no negative saturation. The step response and the control input of the system tuned with ISE are shown in Fig. 5 and those of the system tuned with ITSE are shown in Fig. 7 which are quite similar. Both overshoot and undershoot can be seen in the step response. The step response is faster than the oscillation damping one but slower than IAE and ITAE. The Step response and the control input of the system tuned with ITAE are shown in Fig. 6. It has the maximum overshoot and undershoot, among others, and it has maximum oscillation based on the criterion described before. The Step response and the control input of the system tuned with oscillation damping based on the PTL concept are shown in Fig. 8. The step response has no overshoots, and just one joint angle has a little undershoot. It has the least oscillation, but although it is the slowest transient among others.

V. CONCLUSION

In this paper, the dynamic control of the OpenMANIPULATOR-x was investigated by using a PD with gravity compensation which guarantees the system to be asymptotically stable. A systematic approach was used to obtain Dynamics of OpenMANIPULATOR-x using Euler-Lagrange formulation, which can be easily applied to any serial manipulator with revolute joints. Also, Christoffel symbols of the first type used for control purpose. To have comprehensive modeling of OpenMANIPULATOR-x, forward and inverse kinematics were also represented. The dynamic model of the robotic arm was verified by simulating the understudy robot in Simscape, and the results were exact. Optimization of different performance indices, including IAE, ISE, ITAE, ITSE, and a new criterion called oscillation damping based on the PTL concept was used to tune the PD gains by genetic algorithm. The formulation of closed-loop error dynamics was obtained for applying the oscillation damping approach. The obtained results revealed that the proposed control structure works as expected. The step response of the oscillation damping controller is without overshoot, but it is slower than of ones tuned by other performance indices. Controller tuned by ITAE has maximum

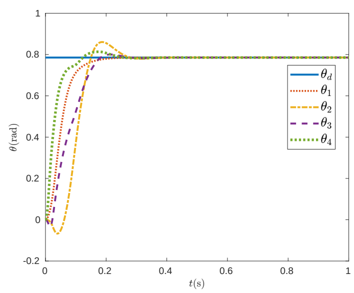


(a) Measured output(θ)

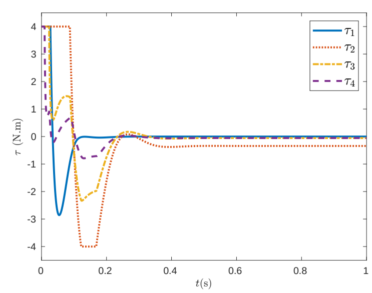


(b) Control input(u)

Fig. 4: PD tuned by minimizing IAE + J_0.

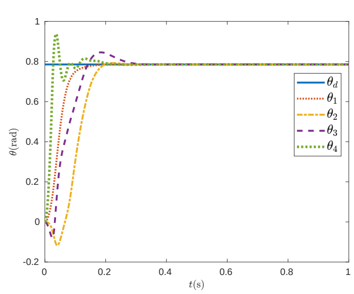


(a) Measured output(θ)

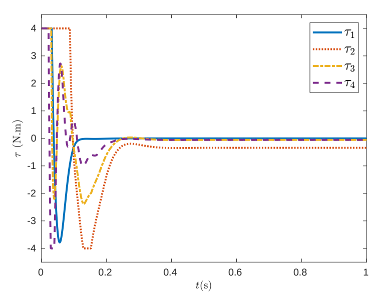


(b) Control input(u)

Fig. 5: PD tuned by minimizing ISE + J_0.

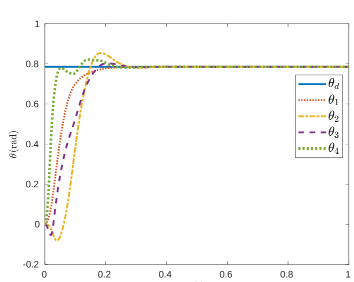


(a) Measured output(θ)

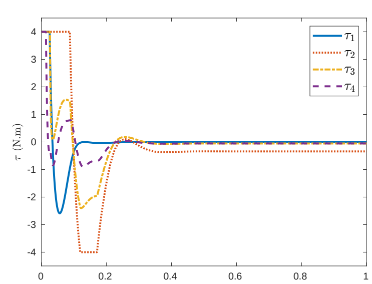


(b) Control input(u)

Fig. 6: PD tuned by minimizing ITAE + J_0.



(a) Measured output(θ)



(b) Control input(u)

Fig. 7: PD tuned by minimizing ITSE + J_0.

TABLE III: PD controller coefficents obtained by minimizing cost functions using GA.

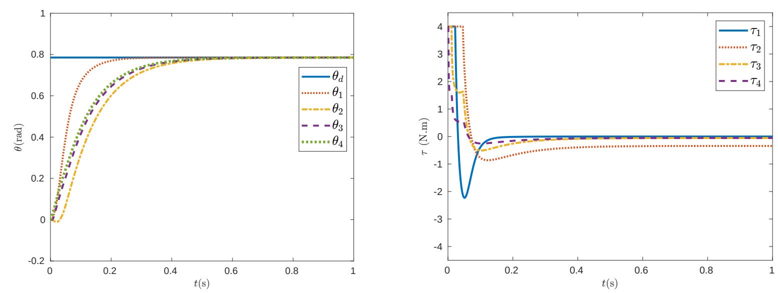
	considered J	PD controller coefficents							
		$K_{P,1}$	$K_{D,1}$	$K_{P,2}$	$K_{D,2}$	$K_{P,3}$	$K_{D,3}$	$K_{P,4}$	$K_{D,4}$
1	IAE + J_0^*	17.9184	0.6200	25.3079	1.5364	23.5159	1.3586	17.3861	0.1550
2	ISE + J_0	27.2611	1.1672	27.9522	1.0416	19.1607	1.2212	21.2494	0.4937
3	ITAE + J_0	34.1013	1.1599	35.8940	1.7048	15.7779	0.6879	31.5020	0.0926
4	ITSE + J_0	23.6809	1.0937	32.4686	1.2104	19.8735	1.1949	20.0180	0.2419
5	$L_{tf}^{**} + J_0$	20.6809	1.0937	20.4686	2.4688	15.2202	1.8000	8.2685	0.9280

$^* J_0 \triangleq 10^6 \times \sum_{i=1}^{4} e_i^2(tf)$

$^{**} L_{tf} \triangleq \int_{t=t_0}^{t=t_f} \sqrt{\sum_{i=1}^{4} \dot{e}_i^2(t)}dt \equiv \int_{t=t_0}^{t=tf} \sqrt{\boldsymbol{f}_c(\boldsymbol{x}(t),t)^T \boldsymbol{f}_c(\boldsymbol{x}(t),t)}dt$

TABLE IV: Performance indices for multiple PD controllers tuned by different performance indices.

	considered J	performance indices value						
		IAE	**ISE**	**ITAE**	**ITSE**	**L**$_{tf}$	**ONI**	J_0
1	IAE + J_0	0.2164	0.1315	0.0114	0.0051	828.4333	6300.1	1.97×10^{-4}
2	ISE + J_0	0.2232	0.1280	0.0120	0.0047	619.8830	4843.8	1.6179×10^{-4}
3	ITAE + J_0	0.2146	0.1316	0.0111	0.0050	1091.2	8291.6	9.79×10^{-5}
4	ITSE + J_0	0.2189	0.1285	0.0117	0.0047	699.5834	5445.1	1.196×10^{-4}
5	$L_{tf} + J_0$	0.3526	0.1641	0.0353	0.0089	397.0680	2419.4	0.0110



(a) Measured output(θ)

(b) Control input(u)

Fig. 8: PD tuned by minimizing L + J$_0$.

oscillation with respect to oscillation number index, and it also has maximum overshoot, among others, but it has minimum raise time instead. Future work will be devoted to integrating the OpenMANIPULATOR-X with a quadrotor for aerial manipulation purposes with minimum oscillation.

REFERENCES

[1] Robotis. OpenMANIPULATORX. [Online]. Available: https://emanual.robotis.com/docs/en/platform/openmanipulator_x/overview/

[2] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," 1955.

[3] J. Angeles, *Fundamentals of Robotic Mechanical Systems — Theory, Methods, and Algorithms*, 01 2007, vol. 124.

[4] P. Sanz, "Robotics: Modeling, planning, and control (siciliano, b. et al; 2009) [on the shelf]," *IEEE Robotics Automation Magazine*, vol. 16, no. 4, pp. 101–101, 2009.

[5] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2005. [Online]. Available: https://books.google.com/books?id=A0OXDwAAQBAJ

[6] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st ed. USA: Cambridge University Press, 2017.

[7] H. Z. Ting, M. Hairi, M. Zaman, M. Ibrahim, and A. Moubark, "Kinematic analysis for trajectory planning of open-source 4-dof robot arm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 769–777, 2021.

[8] R. Kelly, "Pd control with desired gravity compensation of robotic manipulators: A review," *The International Journal of Robotics Research*, vol. 16, no. 5, pp. 660–672, 1997. [Online]. Available: https://doi.org/10.1177/027836499701600505

[9] A. Zavala-Rio and V. Santibanez, "A natural saturating extension of the pd-with-desired-gravity-compensation control law for robot manipulators with bounded inputs," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 386–391, 2007.

[10] B. Nagaraj and N. Murugananth, "A comparative study of pid controller tuning using ga, ep, pso and aco," in *2010 INTERNATIONAL CONFERENCE ON COMMUNICATION CONTROL AND COMPUTING TECHNOLOGIES*, 2010, pp. 305–313.

[11] M. I. Solihin, L. F. Tack, and M. L. Kean, "Tuning of pid controller using particle swarm optimization (pso)," in *Proceeding of the International Conference on Advanced Science, Engineering and Information Technology*, vol. 1, 2011, pp. 458–461.

[12] C. Gosselin and H. Liu, "Polynomial inverse kinematic solution of the jaco robot," 08 2014, p. V05BT08A055.

[13] M. I. Solihin, L. F. Tack, and M. L. Kean, "Tuning of pid controller using particle swarm optimization (pso)," in *Proceeding of the International Conference on Advanced Science, Engineering and Information Technology*, vol. 1, 2011, pp. 458–461.

[14] M. Zarei and A. Kalhor, "Employing phase trajectory length concept as performance index in linear power oscillation damping controllers," *International Journal of Electrical Power & Energy Systems*, vol. 98, 06 2018.

[15] M. Zarei, A. Aflakian, A. Kalhor, and M. Masouleh, "Oscillation damping of nonlinear control systems based on the phase trajectory length concept: An experimental case study on a cable-driven parallel robot," *Mechanism and Machine Theory*, vol. 126, 08 2018.