

Real-Time Inverse Dynamic Deep Neural Network Tracking Control for Delta Robot Based on a COVID-19 Optimization

Mohamed A. Shamseldin ^{1*}

¹ Department of Mechanical Engineering, Future University in Egypt, Cairo, Egypt

Email: ¹ mohamed.abelbbar@fue.edu.eg

*Corresponding Author

Abstract—This paper presents a new technique to design an inverse dynamic model for a delta robot experimental setup to obtain an accurate trajectory. The input/output data were collected using an NI DAQ card where the input is the random angles profile for the three-axis and the output is the corresponding measured torques. The inverse dynamic model was developed based on the deep neural network (NN) and the new COVID-19 optimization to find the optimal initial weights and bias values of the NN model. Due to the system uncertainty and nonlinearity, the inverse dynamic model is not enough to track accurately the preselected profile. So, the PD compensator is used to absorb the error deviation of the end effector. The experimental results show that the proposed inverse dynamic deep NN with PD compensator achieves good performance and high tracking accuracy. The suggested control was examined using two different methods. The spiral path is the first, with a root mean square error of 0.00258 m, while the parabola path is the second, with a root mean square error of 0.00152 m.

Keywords—Parallel Robot; Delta Robot; Data Acquisition Tracking Control; System Identification.

I. INTRODUCTION

As technology advanced, parallel robots became more desirable in industrial applications where high speed, high accuracy, and high acceleration are required [1]-[4]. When compared to serial robots, parallel robots have clear advantages, such as their high speed and rigidity[5]-[9]. Delta robots are currently widely employed in the packaging business, the medical and pharmaceutical sectors, and surgery [10]-[12]. They are one of the most successful industrial parallel robots where it has high speed with satisfied accuracy.

The precision of the mathematical models and the sensors used in the control loop determines the accuracy of the controllers created using these models. However, after manufacture and assembly or because of joint wear and backlash from the prolonged operation, the mathematical modeling of a robot may differ from the real modeling [13]-[23].

So, the typical linear PID controller is unable to meet the need for precise trajectory in high-speed applications [14], [15]. The replacement to the PID control is the artificial intelligent control techniques such as neural networks and fuzzy PID control or the adaptive and robust control methods [24]-[34].

Recent research on deep learning and neural networks has shown that these techniques may be used for robotic control and that in some cases, neural networks can take the place of intricate mathematical modeling [35]-[37]. From data, neural networks can learn. They are qualified to estimate the mathematical modeling of a system since they can detect non-linearity between input and output data [38]-[45].

A 3-DOF Delta parallel robot's simulation, kinematics analysis, and control have been investigated in [46] where the ability of the controllers to reject white noise disturbance applied to the system's input signals was demonstrated using simulation results [46]-[60].

A real-time delta robot trajectory control system employing an inverse kinematic controller and neural networks has been presented in [2]. The results demonstrate that joint backlash's detrimental impact on trajectory tracking is diminished and the inaccuracy in trajectory tracking is constrained in the presence of external disturbance. The created method offers a novel method for controlling the inverse kinematics of a delta robot.

The approach for system identification for a delta robot is suggested in [3]. The dynamic behavior of the parallel-link robot was theoretically modeled using the virtual work concept. The suggested parameter estimation approach is a very helpful tool that can successfully identify a high-quality analytic dynamic model for a parallel-link robot, according to experimental data. The deep NN models used in recent research for several serial robots successfully.

In this work, an efficient inverse dynamic model for the experimental setup of a delta robot had been implemented based on the deep NN model using real-time measured data. The optimal values of the initial values of weights and bias were obtained using the COVID-19 optimization. Also, the PD compensator was added to absorb the error deviation of the inverse dynamic deep NN control. A comparative study between the two cases (with and without a compensator) had been investigated. Several trajectory profiles had been applied to measure the delta robot performance. The results demonstrate that the proposed Inverse dynamic deep NN with PD compensator can track accurately the preselected profile.

This article is organized as follows. The "Problem formulation" section introduces the experimental setup structure and the dynamics of the Delta robot. The "Inverse



Dynamic Control” section proposes the methodology for the development inverse dynamic model for an experimental delta robot. The “Delta robot performance” section presents the performance of the experimental setup of the delta robot using the proposed control techniques. The “Conclusions” section summarizes some significant conclusions.

II. PROBLEM FORMULATION

The experimental setup for the delta robot is illustrated in Fig. 1, which consists of two primary platforms. While the second platform is mobile, the first platform is fixed. Three independent, identical kinematic chains that are dispersed at a 120° angle connect the two platforms. Each drive is connected to the platform by two links that create a parallelogram. An output link may stay in a fixed orientation with respect to an input link thanks to a parallelogram. This type of architecture displays excellent high speed, low inertia, and precision performance.

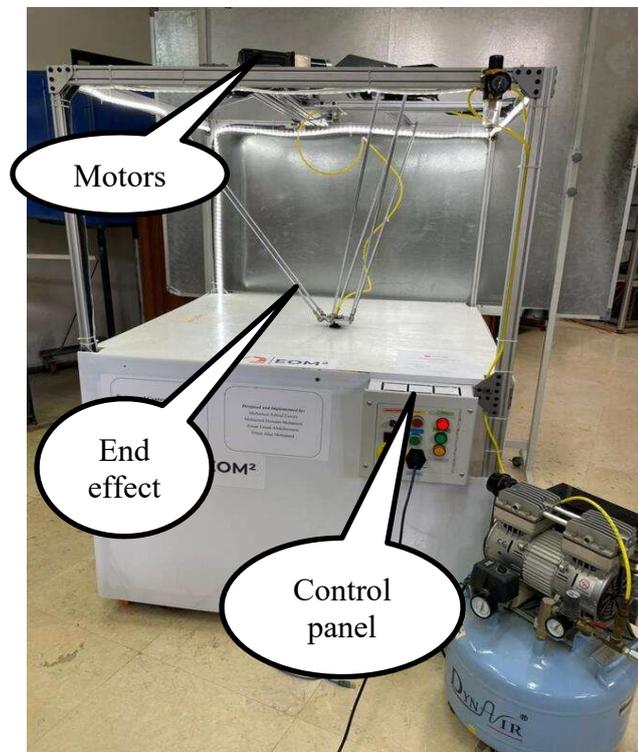


Fig. 1. Delta robot experimental setup

Lagrangian mechanics is a version of classical mechanics that was developed in physics and is based on the stationary-action principle. The dynamic equations of the Delta robot manipulator are found through the use of the Lagrangian formulation and can be written as (1).

$$\begin{aligned} \tau_1 &= (\gamma^2 I_m + I_1 + m_2 r_f^2) \ddot{\theta}_1 - (m_1 r_{fc} + m_2 r_f) g c \theta_1 \\ &\quad - 2r_f \lambda_1 [(x_0 c \phi_1 + y_0 s \phi_1 + b - a) s \theta_1 - z_0 c \theta_1] \\ \tau_2 &= (\gamma^2 I_m + I_1 + m_2 r_f^2) \ddot{\theta}_2 - (m_1 r_{fc} + m_2 r_f) g c \theta_2 \\ &\quad - 2r_f \lambda_2 [(x_0 c \phi_2 + y_0 s \phi_2 + b - a) s \theta_2 - z_0 c \theta_2] \\ \tau_3 &= (\gamma^2 I_m + I_1 + m_2 r_f^2) \ddot{\theta}_3 - (m_1 r_{fc} + m_2 r_f) g c \theta_3 \\ &\quad - 2r_f \lambda_3 [(x_0 c \phi_3 + y_0 s \phi_3 + b - a) s \theta_3 - z_0 c \theta_3] \end{aligned} \quad (1)$$

Also, the values of delta robot parameters are demonstrated in Table I.

TABLE I. PARAMETERS AND THEIR VALUES OF THE TORQUE EQUATIONS (1)

Parameter	Value	Parameter	Value
λ_1	-15.1458	r_f	366 mm
λ_2	-13.238	r_{fc}	183 mm
λ_3	-16.4696	γ	5:1
$\dot{\theta}_1$	15.02 rad/s ²	I_m	0.5x10 ⁻⁴ kgm ²
$\dot{\theta}_2$	12.02 rad/s ²	I_1	80x10 ⁻⁴ kgm ²
$\dot{\theta}_3$	17.36 rad/s ²	m_1	200 g
g	9.81 m/s ²	m_2	400 g
X_0	70.45 mm	m_p	200 g
Y_0	-116.6 mm	a	370 mm
Z_0	-819 mm	b	40 mm
θ_1	variable	ϕ_1	60°
θ_2	variable	ϕ_2	180°
θ_3	variable	ϕ_3	-60°

Two stages may be used in the design of the controllers for the fine motion of the delta robot: the primary controller, which is nothing more than the manipulator's inverse dynamics inserted in the feedforward path to account for the nonlinear effects and which attempts to cancel the nonlinear terms in the model. However, since the system is susceptible to disturbances and the mathematical model utilized is typically not accurate, unwanted mistakes can be remedied using a second controller, or secondary controller, as shown in Fig. 2.

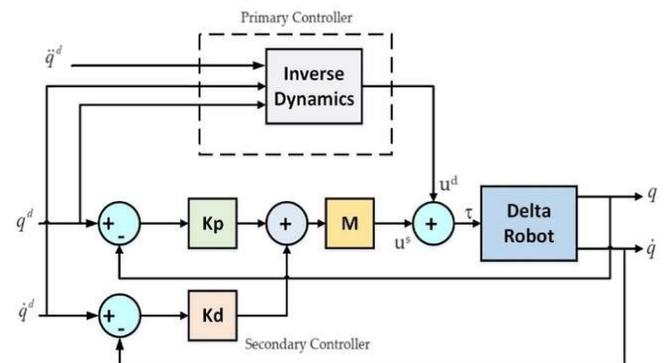


Fig. 2. Primary and secondary Delta robot control

III. INVERSE DYNAMICS CONTROL

The foundation of the inverse dynamic control technique is a control command that recognizes a priori system knowledge encoded in a dynamic model. The computed torque method is another name for this strategy. Joint angles, velocities, and accelerations are the desired trajectories because the control objective is trajectory tracking in a joint space. These are the tracking errors' definitions as equation (2) and (3).

$$e = q^d - q \quad (2)$$

$$e' = \dot{q}^d - \dot{q} \quad (3)$$

where the measurements of q and \dot{q} are required in the subsequent design. To conform to the inverse nonlinear NN control given in Fig. 3.

The main idea of the program has been designed to make the NI DAQ 6009 generates a random analog output signal to the three motor drives (-5V to 5V) which can change the three axis of the delta robot from -160° to 160°. Also, the corresponding analog output signal data from the torque

sensor has been collected at the same time. The developed torque of the motor will fluctuate when the generated signal change continuously. The positive signal will cause the motor position to fluctuate in the forward direction, while the motor will fluctuate in the reverse direction through the negative voltage ranges.

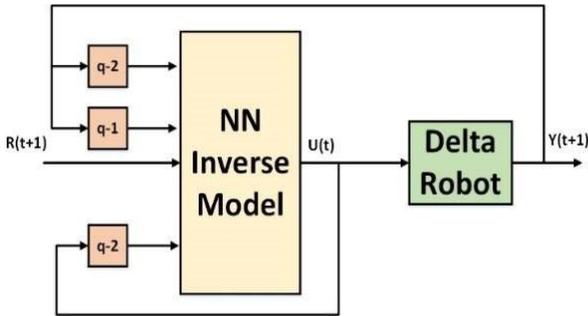


Fig. 3. Inverse control training structure

The camera, as shown in Fig. 4, is the most crucial sensor component. To offer sensory feedback in the form of vision, the Pixy 2 Camera was employed. Although this particular camera can determine an object's color, it cannot determine the object's topology. This camera can locate the end effector of a delta robot. So, using inverse kinematics calculations, one may determine the angles of linkages.



Fig. 4. Pixy2 Camera

Fig. 5 illustrates the generated position input signal to the motor driver which change randomly while Fig. 6, Fig. 7 and Fig.8 show the corresponding torque for each link. The input /output data will be collected and stored in excel sheet file and then this data will be used to develop inverse dynamic NN identified model for the experimental setup.

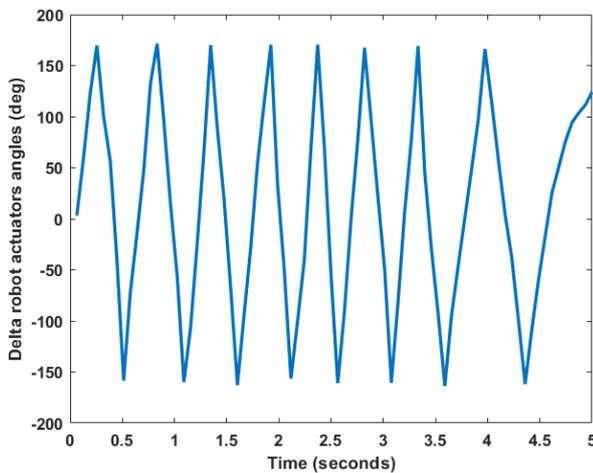


Fig. 5. The random motions for the three axes of the delta robot

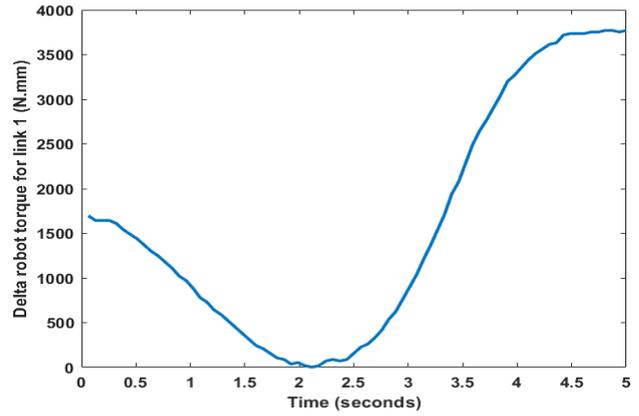


Fig. 6. The measured output torque for link 1

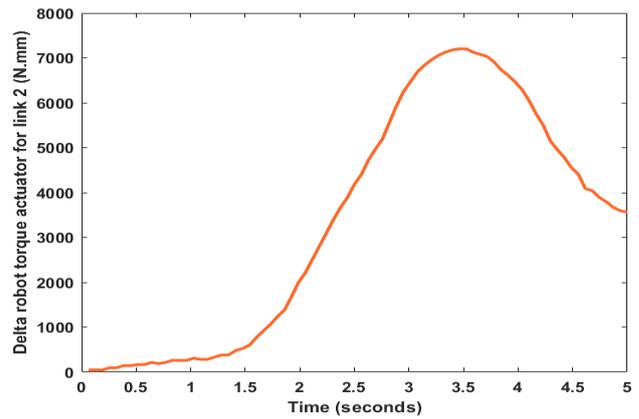


Fig. 7. The measured output torque for link 2

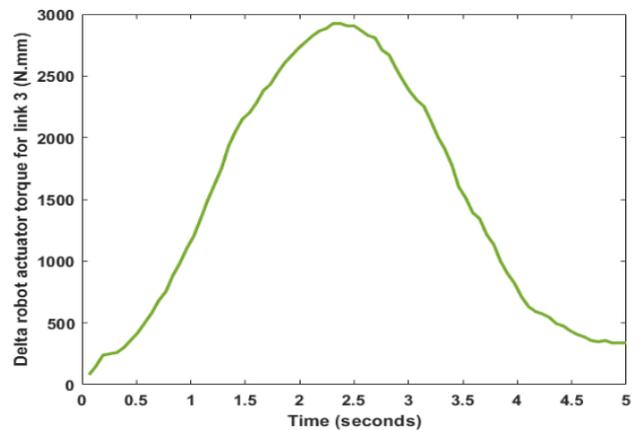


Fig. 8. The measured output torque for link 3

One of the first strategies described for using neural networks to control unknown nonlinear systems was training the network to serve as the system's inverse and using that as a controller.

Assuming that the system to be controlled can be described by (4).

$$y(t + 1) = g[y(t), \dots, y(t - n + 1), u(t), \dots, u(t - m)] \quad (4)$$

The desired network is then the one that isolates the most recent control input, $u(t)$ shown in (5).

$$\hat{u}(t) = \hat{g}^{-1}[y(t + 1), y(t), \dots, y(t - n + 1), u(t), \dots, u(t - m)] \quad (5)$$

Assuming such a network has somehow been obtained, it can be used for controlling the system by substituting the output at time $t+1$ with the desired output, the reference, $r(t+1)$. If the network represents the exact inverse, the control input produced by it will thus drive the system out at time $t+1$ to $r(t+1)$ as shown in Fig. 3.

The most straightforward way of training a network as the inverse of a system is to approach the problem as a system identification problem. A network architecture is selected, and the network is trained offline. The difference from the system identification lies in the choice of regressors and network output. They are now selected as shown in equation (6). The network is then trained to minimize the criterion.

$$j(\theta, Z^N) = \frac{1}{2N} \sum_{t=1}^N [u(t) - \hat{u}(t/\theta)]^2 \quad (6)$$

The fact that general training is fundamentally not a model-based design method is an appealing aspect of it. This should be taken to mean that the controller can be inferred directly from a piece of data without the need for a system model.

One should not be confused by the assumption that the reference is known one step in advance. This was just used as a pedagogical notation. The closed-loop transfer function from reference to the output of the system is in fact shown in (7).

$$H(q^{-1}) = q^{-1} \quad (7)$$

One can interpret this as if the controller is linearizing the system, resulting in a dead-beat controller: the system output will follow the reference signal exactly exact for a delay of one sampling period. If the system has a time delay exceeding one, using inverse models for control becomes slightly more complicated as in the case of the delta robot system. A dead-beat controller is still obtained but now the closed-loop transfer function becomes $H(q^{-1}) = q^{-d}$, with d being the time delay. The principle for handling time delays is outlined in the following. Assume that the system is governed by (8).

$$y(t+d) = g[y(t+d-1), \dots, y(t+d-n), u(t), \dots, u(t-m)] \quad (8)$$

Once again, a network is trained as the inverse model in (9).

$$\hat{u}(t) = \hat{g}^{-1}[y(t+d), y(t+d-1), \dots, y(t), \dots, y(t+d-n), u(t-1), \dots, u(t-m)] \quad (9)$$

Analogous to the case considered before, $y(t+d)$ is substituted for the desired output at time $t+d$. this leaves $d-1$ unknown quantities:

$$\{y(t+1), \dots, y(t+d-1)\}$$

A solution to this problem is to insert predictions of the unknown outputs which implies that one or more networks are trained to provide the necessary predictions. Alternatively, the predictor can be "incorporated" into the inverse model directly. Assume for example that the time delay $d=2$. In this case, there will be one unknown quantity,

namely $y(t+1)$. The prediction of this takes the form equation (10).

$$y(t+1) \cong \hat{y}(t+1) = \hat{g}_1[y(t), \dots, y(t+1-n), u(t-1), \dots, u(t-m-1)] \quad (10)$$

The inverse model is then trained by using as regressors the union of regressors from (4) and (10), then the equation obtained in (11).

$$\hat{u}(t) = \hat{g}^{-1}[y(t+2), y(t), \dots, y(t+2-n), \dots, y(t+1-n), u(t-1), \dots, u(t-m), u(t-m-1)] \quad (11)$$

When $d>2$, it is straightforward to proceed in a similar fashion.

- 1- Conduct an experiment to generate a data set.
- 2- Initialize the inverse model with general training. Use for example the Levenberg-Marquardt method.
- 3- Proceed with specialized training "offline" by using the model of the system instead of the actual system. Apply a recursive Gauss-Newton algorithm with forgetting for rapid convergence but be careful with "covariance blow-up."
- 4- Conclude the session with online specialized training. Terminate the training algorithm when an acceptable model-following behavior has been achieved.

In order to increase the accuracy of the resultant model, the starting weights and bias of the deep NN model will be discovered utilizing the COVID-19 optimization in this work. The first step in COVID-19 optimization is the creation of the starting population. The starting population (zero patients) has just one distinct vector. Similar to the COVID-19 epidemic scenario, it can locate the first affected individual.

The higher and lower values of the deep NN model's initial weights and bias, as given in equation (12), are contained in the initial population.

$$X = \begin{bmatrix} w_{00} & \cdot & w_{0j} & \theta_0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ w_{i0} & \cdot & w_{ij} & \theta_n \end{bmatrix} \quad (12)$$

In the second stage, several situations may be taken into account, but only one vector (zero patients) is responsible for the disease's spread. Some of the ill patients first perish away. There is a probability of dying, based on COVID-19's mortality rate. Such individuals are no longer able to infect new individuals.

In the second scenario, the COVID-19 survivors amplify the disease's transmission by infecting other people. Two ways of spreading diseases are consequently taken into account based on a given likelihood. Usual spreaders. According to the COVID-19 super spreading rate, infected persons will infect new people if the virus is widely diffused. According to the pace of infection propagation, sick persons

will infect new people. Super-spreaders and regular individuals can follow instructions and generate solutions in quite different ways. Since people are inclined to travel, the spread of the disease to environments that may be extremely different is made possible.

The final phase entails modernizing the populace. Three populations are updated and maintained for each generation. a dead population. Any deaths are permanently removed from this population and added to it. Those who have recouped. Sick persons are sent to the population that has recovered after the coronavirus has been spread as per the previous stage after each iteration. It is good knowledge that a reinfection is possible.

As a result, everyone in this group who fits the requirements for reinfection is susceptible to getting sick at any time. Another condition needs to be assessed since people might separate themselves while seeming to utilize social distance strategies. For the convenience of usage, it is assumed that an isolated individual is also sent to the recovered population when an isolation probability is fulfilled. Population recently impacted. The approach described in the prior steps is used to gather each ill person into this group. It is advisable to eliminate these people from the population before the next iteration starts since it is possible that fresh sick individuals form periodically after each iteration. The objective function that can treat the afflicted population is taken into consideration by the vaccines.

The fourth stage is the Stop situation. One of the most important features of the offered technique is its ability to operate completely without the need for any parameter management. This condition develops as a result of the populations that have recovered and died continuously rising over time and the newly infected population being unable to spread illness. Estimates show that the number of infected persons increases after a certain number of rounds. The populations that have been recovered and killed are too big, and because the size of the infected population shrinks with time, starting with a certain iteration, the newly infected population will be smaller than the population that currently exists.

The efficiency of each row will be evaluated using the target function in equation (6). The poor performance identifies the ill population, which runs the risk of dying. While the positive outcome indicates that Corona antivirus's population has been restored.

The optimization will be stopped if ($X_{new} = X_{old}$) where the newly infected populations cannot infect new individuals. If the number of iterations ended before this previous condition. The COVID-19 cannot give the optimal solution. Therefore, to obtain the optimal parameters of the NN model must be ($X_{new} = X_{old}$) to guarantee the global solution.

IV. DELTA ROBOT PERFORMANCE

To demonstrate and validate the effectiveness of the suggested neural network in inverse dynamic modeling and trajectory tracking, two distinct pathways are developed. A spiral is defined for the first reference track using the equation (13).

$$\begin{aligned} x &= 0.1 \cos(\gamma) \\ y &= 0.1 \sin(\gamma) \\ z &= (-0.1\gamma)/(4\pi - 0.4) \end{aligned} \quad (13)$$

where γ fluctuates between 0 and 4π . The initial position of the delta robot is random when it begins in a steady state.

Fig. 9 demonstrates the following of a spiral path using the inverse dynamic NN control with and without the PD compensator in real time. It can be noted that the inverse dynamic deep NN control can track the reference trajectory with a satisfied root mean square error of 0.0289 m due to the system noise and uncertainty. When the PD compensator had been added to the inverse dynamic deep NN control the performance will improve significantly where the root mean square will be 0.00258 m.

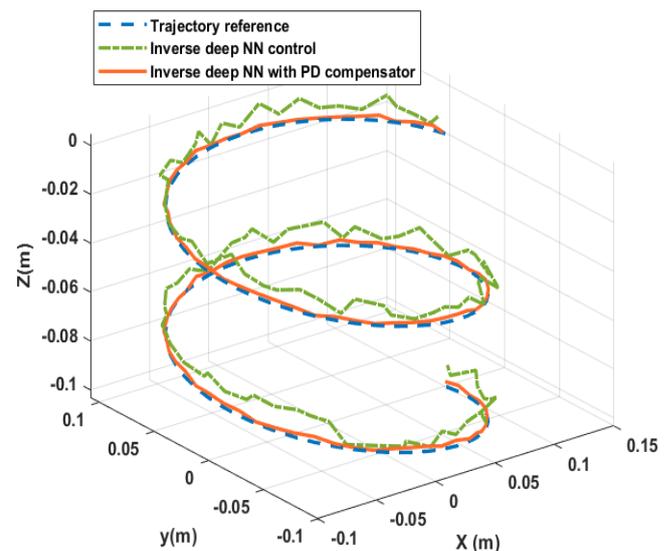


Fig. 9. Tracking of a spiral path using an inverse dynamic NN control with and without the PD compensator in real-time

For the second trajectory, we evaluate a non-smooth trajectory tracking using a parabola with a sudden change in the direction of the end effector motion. The equation (14) describes the required trajectory.

$$\begin{aligned} x &= 0.1\gamma \\ y &= 0.2\gamma \\ z &= -0.5 + 0.1 \cos(\gamma) \end{aligned} \quad (14)$$

where γ fluctuates between 0 and 4π .

Fig. 10 displays the tracing of a parabola path using an inverse dynamic NN control with and without the PD compensator in real time.

It is obvious that the end effector for both the proposed controllers can track smoothly the change of the path direction. But in the case of the inverse dynamic deep NN control the root mean square error (0.0198 m) is high compared to the inverse dynamic deep NN control with PD compensator (0.00152 m) which is used to absorb the measurement noise and uncertainty of the system.

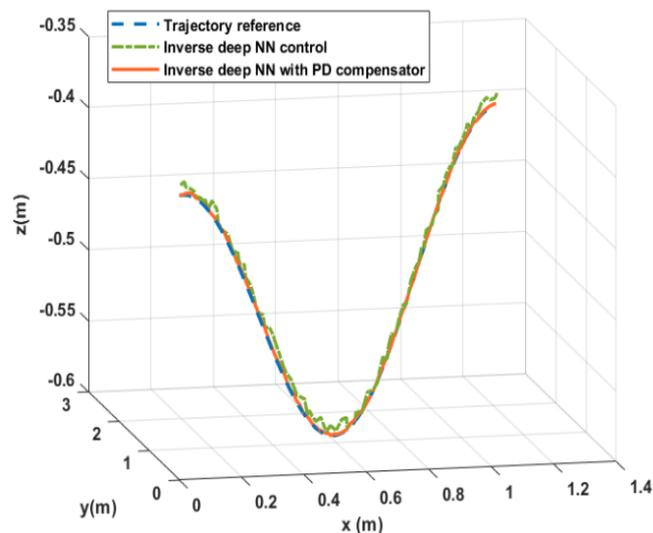


Fig. 10. Tracing of a parabola path using an inverse dynamic NN control with and without the PD compensator in real-time

REFERENCES

- [1] Y. L. Kuo and P. Y. Huang, "Experimental and simulation studies of motion control of a Delta robot using a model-based approach," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 6, pp. 1–14, 2017, doi: 10.1177/1729881417738738.
- [2] A. Gholami, T. Homayouni, R. Ehsani, and J. Q. Sun, "Inverse kinematic control of a delta robot using neural networks in real-time," *Robotics*, vol. 10, no. 4, 2021, doi: 10.3390/robotics10040115.
- [3] T. H. Kim, Y. Kim, T. Kwak, and M. Kanno, "Metaheuristic Identification for an Analytic Dynamic Model of a Delta Robot with Experimental Verification," *Actuators*, vol. 11, no. 12, pp. 1–20, 2022, doi: 10.3390/act11120352.
- [4] P. C. Pham and Y. L. Kuo, "Robust Adaptive Finite-Time Synergetic Tracking Control of Delta Robot Based on Radial Basis Function Neural Networks," *Appl. Sci.*, vol. 12, no. 21, 2022, doi: 10.3390/app122110861.
- [5] F. B. Setiawan, P. M. Siva, L. H. Pratomo, and S. Riyadi, "Design and implementation of smart forklift for automatic guided vehicle using raspberry pi 4," *J. Robot. Control*, vol. 2, no. 6, pp. 508–514, 2021, doi: 10.18196/jrc.26130.
- [6] J. M. Guirguis, S. Hammad, and S. A. Maged, "Path Tracking Control Based on an Adaptive MPC to Changing Vehicle Dynamics," *Int. J. Mech. Eng. Robot. Res.*, vol. 11, no. 7, pp. 535–541, 2022, doi: 10.18178/ijmerr.11.7.535-541.
- [7] M. R. Islami, M. R. T. Hossain, and S. C. Banik, "Synchronizing of stabilizing platform mounted on a two-wheeled robot," *J. Robot. Control*, vol. 2, no. 6, pp. 552–558, 2021, doi: 10.18196/jrc.26136.
- [8] S. R. Utama, A. Firdausi, and G. P. N. Hakim, "Control and Monitoring Automatic Floodgate Based on NodeMCU and IOT with Fuzzy Logic Testing," *J. Robot. Control*, vol. 3, no. 1, pp. 14–17, 2022, doi: 10.18196/jrc.v3i1.11199.
- [9] M. Rahmani, H. Komijani, and M. H. Rahman, "New Sliding Mode Control of 2-DOF Robot Manipulator Based on Extended Grey Wolf Optimizer," *Int. J. Control. Autom. Syst.*, vol. 18, no. 6, pp. 1572–1580, 2020, doi: 10.1007/s12555-019-0154-x.
- [10] V. Verma, A. Gupta, M. K. Gupta, and P. Chauhan, "Performance estimation of computed torque control for surgical robot application," *J. Mech. Eng. Sci.*, vol. 14, no. 3, pp. 7017–7028, 2020, doi: 10.15282/jmes.14.3.2020.04.0549.
- [11] S. I. Han, "Fractional-order Sliding Mode Constraint Control for Manipulator Systems Using Grey Wolf and Whale Optimization Algorithms," *Int. J. Control. Autom. Syst.*, vol. 19, no. 2, pp. 676–686, 2021, doi: 10.1007/s12555-020-0138-x.
- [12] N. P. Van Bach, Q. D. Hai, and T. B. Trung, "Optimization of trajectory tracking control of 3-DOF translational robot use PSO method based on inverse dynamics control for surgery application," *Journal of Vibroengineering*, vol. 23, no. 7, pp. 1591–1601, 2021.
- [13] M. A. Mossa, H. Khoudmi, and A. Ma'arif, "Robust Flux and Speed State Observer Design for Sensorless Control of a Double Star Induction Motor," *J. Robot. Control*, vol. 3, no. 4, pp. 464–475, 2022, doi: 10.18196/jrc.v3i4.15667.
- [14] Y. Zahraoui, M. Akherraz, C. Fahassa, and S. Elbadaoui, "Induction Motor DTC Performance Improvement by Reducing Flux and Torque Ripples in Low Speed," *J. Robot. Control*, vol. 3, no. 1, pp. 93–100, 2022, doi: 10.18196/jrc.v3i1.12550.
- [15] A. Latif, A. Z. Arfianto, H. A. Widodo, R. Rahim, and E. T. Helmy, "Motor DC PID system regulator for mini conveyor drive based-on matlab," *J. Robot. Control*, vol. 1, no. 6, pp. 185–190, 2020, doi: 10.18196/jrc.1636.
- [16] Y. Irawan, Muhardi, R. Ordila, and R. Diandra, "Automatic floor cleaning robot using arduino and ultrasonic sensor," *J. Robot. Control*, vol. 2, no. 4, pp. 240–243, 2021, doi: 10.18196/jrc.2485.
- [17] A. W. L. Yao and H. C. Chen, "An Intelligent Color Image Recognition and Mobile Control System for Robotic Arm," *Int. J. Robot. Control Syst.*, vol. 2, no. 1, pp. 97–104, 2022, doi: 10.31763/ijrc.v2i1.557.
- [18] M. Abdullah-Al-Noman, A. N. Eva, T. B. Yeahyea, and R. Khan, "Computer Vision-based Robotic Arm for Object Color, Shape, and Size Detection," *J. Robot. Control*, vol. 3, no. 2, pp. 180–186, 2022, doi: 10.18196/jrc.v3i2.13906.
- [19] M. Marsono, Y. Yoto, A. Suetno, and R. Nurmalsari, "Design and programming of 5 axis manipulator robot with grblgru open source software on preparing vocational students' robotic skills," *J. Robot. Control*, vol. 2, no. 6, pp. 539–545, 2021, doi: 10.18196/jrc.26134.
- [20] D. M. Harfina, Z. Zaini, and W. J. Wulung, "Disinfectant spraying system with quadcopter type unmanned aerial vehicle technology as an effort to break the chain of the covid-19 virus," *J. Robot. Control*, vol. 2, no. 6, pp. 502–507, 2021, doi: 10.18196/jrc.26129.
- [21] A. Ma'arif and N. R. Setiawan, "Control of dc motor using integral state feedback and comparison with pid: Simulation and arduino implementation," *J. Robot. Control*, vol. 2, no. 5, pp. 456–461, 2021, doi: 10.18196/jrc.25122.
- [22] I. Suwarno, Y. Finayani, R. Rahim, J. Alhamid, and A. R. Al-Obaidi, "Controllability and Observability Analysis of DC Motor System and a Design of FLC-Based Speed Control Algorithm," *J. Robot. Control*, vol. 3, no. 2, pp. 227–235, 2022, doi: 10.18196/jrc.v3i2.10741.
- [23] A. Ma'arif and A. Çakan, "Simulation and arduino hardware implementation of dc motor control using sliding mode controller," *J. Robot. Control*, vol. 2, no. 6, pp. 582–587, 2021, doi: 10.18196/jrc.26140.
- [24] H. Chhabra, V. Mohan, A. Rani, and V. Singh, "Robust nonlinear fractional order fuzzy PD plus fuzzy I controller applied to robotic manipulator," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2055–2079, 2020, doi: 10.1007/s00521-019-04074-3.
- [25] M. A. S. Mohamed A. Abdel Ghany, "Fuzzy type two self-tuning technique of single neuron PID controller for brushless DC motor based on a COVID-19 optimization Mohamed," *Int. J. Power Electron. Drive Syst.*, vol. 14, no. 1, 2023.
- [26] J. Poovarasam, R. Kayalvizhi, and R. K. Pongiannan, "Design of Fractional Order PID controller for a CSTR process," *Int. Ref. J. Eng. Sci.*, vol. 3, no. 1, pp. 8–14, 2014.
- [27] M. Omar, M. A. Ebrahim, A. M. Abdel Ghany, and F. Bendary, "Reduced size harmony search algorithm for optimization," *J. Electr. Eng.*, vol. 16, no. 1, pp. 395–402, 2016.
- [28] H. Abdelfattah, S. A. Kotb, M. Esmail, and M. I. Mosaad, "Adaptive Neuro-Fuzzy Self-Tuned-PID Controller for," *Int. J. Robot. Control Syst.*, vol. 3, no. 1, pp. 1–18, 2023.
- [29] H. Maghfiroh, M. Ahmad, A. Ramelan, and F. Adriyanto, "Fuzzy-PID in BLDC Motor Speed Control Using MATLAB/Simulink," *J. Robot. Control*, vol. 3, no. 1, pp. 8–13, 2022, doi: 10.18196/jrc.v3i1.10964.
- [30] A. Daraz, S. A. Malik, A. T. Azar, S. Aslam, T. Alkhalifah, and F. Alturise, "Optimized Fractional Order Integral-Tilt Derivative Controller for Frequency Regulation of Interconnected Diverse Renewable Energy Resources," *IEEE Access*, vol. 10, pp. 43514–43527, 2022, doi: 10.1109/ACCESS.2022.3167811.
- [31] G. I. Y. Mustafa, X. Li, and H. Wang, "A New Neural Network-Based Adaptive Time-Delay Control for Nonlinear Car Active Suspension System," *Stud. Informatics Control*, vol. 31, no. 4, pp. 13–24, 2022, doi: 10.24846/V31I4Y202202.

- [32] R. R. Ardeshiri, H. N. Kashani, and A. Reza-Ahrabi, "Design and simulation of self-Tuning fractional order fuzzy PID controller for robotic manipulator," *Int. J. Autom. Control*, vol. 13, no. 5, pp. 595–618, 2019, doi: 10.1504/IJAAC.2019.101912.
- [33] F. T. Li, L. Ma, L. T. Mi, Y. X. Zeng, N. B. Jin, and Y. L. Gao, "Friction identification and compensation design for precision positioning," *Adv. Manuf.*, vol. 5, no. 2, pp. 120–129, 2017, doi: 10.1007/s40436-017-0171-z.
- [34] E. Yuliza, H. Habil, R. A. Salam, M. M. Munir, M. Abdullah, and Khairurrijal, "Development of a Simple Single-Axis Motion Table System for Testing Tilt Sensors," *Procedia Eng.*, vol. 170, pp. 378–383, 2017, doi: 10.1016/j.proeng.2017.03.061.
- [35] A. G. M. A. G. Mohamed A. Shamseldin, Abdel Halim M. Bassiuny, "Design variable structure fuzzy control based on deep neural network model for servomechanism drive system," *Int. J. Power Electron. Drive Syst.*, vol. 13, no. 4, 2022.
- [36] P. Zhao and Y. Shi, "Robust control of the A-axis with friction variation and parameters uncertainty in five-axis CNC machine tools," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 228, no. 14, pp. 2545–2556, 2014, doi: 10.1177/0954406213519759.
- [37] R. Kristiyono and Wiyono, "Autotuning fuzzy PID controller for speed control of BLDC motor," *J. Robot. Control*, vol. 2, no. 5, pp. 400–407, 2021, doi: 10.18196/jrc.25114.
- [38] M. Chang and G. Guo, "Sinusoidal Servocompensator Implementations with Real-Time Requirements and Applications," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 2, pp. 645–652, 2017, doi: 10.1109/TCST.2016.2558476.
- [39] K. Vanchinathan and N. Selvaganesan, "Adaptive fractional order PID controller tuning for brushless DC motor using artificial bee colony algorithm," *Results in Control and Optimization*, vol. 4, p. 100032, 2021.
- [40] E. Cajueiro, R. Kalid, and L. Schnitman, "Using NARX model with wavelet network to inferring the polished rod position," *Int. J. Math. Comput. Simul.*, vol. 6, no. 1, pp. 66–73, 2012.
- [41] P. Zhao, J. Huang, and Y. Shi, "Nonlinear dynamics of the milling head drive mechanism in five-axis CNC machine tools," *Int. J. Adv. Manuf. Technol.*, vol. 91, no. 9–12, pp. 3195–3210, 2017, doi: 10.1007/s00170-017-9989-6.
- [42] F. Abdessemed, "SVM-based control system for a robot manipulator," *Int. J. Adv. Robot. Syst.*, vol. 9, no. 6, p. 247, 2012.
- [43] B. Feng, D. Zhang, J. Yang, and S. Guo, "A novel time-varying friction compensation method for servomechanism," *Math. Probl. Eng.*, vol. 2015, 2015, doi: 10.1155/2015/269391.
- [44] M. Saad, A. H. Amhedb, and M. Al Sharqawi, "Real time DC motor position control using PID controller in LabVIEW," *J. Robot. Control*, vol. 2, no. 5, pp. 342–348, 2021, doi: 10.18196/jrc.25104.
- [45] D. Handaya and R. Fauziah, "Proportional-integral-derivative and linear quadratic regulator control of direct current motor position using multi-turn based on LabView," *J. Robot. Control*, vol. 2, no. 4, pp. 332–336, 2021, doi: 10.18196/jrc.24102.
- [46] F. A. Azad, S. Rahimi, M. R. Hairi Yazdi, and M. T. Masouleh, "Design and Evaluation of Adaptive and Sliding Mode Control for a 3-DOF Delta Parallel Robot," *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, pp. 1–7, 2020, doi: 10.1109/ICEE50131.2020.9261040.
- [47] H. L. Tran and T. V. Dang, "An Ultra Fast Semantic Segmentation Model for AMR's Path Planning," *Journal of Robotics and Control (JRC)*, vol. 4, no. 3, pp. 424–430, doi: 10.18196/jrc.v4i3.18758.
- [48] A. A. Abed, A. Al-Ibadi, and I. A. Abed, "Vision-Based Soft Mobile Robot Inspired by Silkworm Body and Movement Behavior," *J. Robot. Control*, vol. 4, no. 3, pp. 299–307, 2023, doi: 10.18196/jrc.v4i3.16622.
- [49] H. A. Mintsá, G. E. Eny, N. Senouveau, J. P. Kenné, and R. M. A. Nzué, "An Alternative Nonlinear Lyapunov Redesign Velocity Controller for an Electrohydraulic Drive," *J. Robot. Control*, vol. 4, no. 2, pp. 192–201, 2023, doi: 10.18196/jrc.v4i2.17340.
- [50] F. Z. Baghli, Y. Lakhal, and Y. A. El Kadi, "The Efficiency of an Optimized PID Controller Based on Ant Colony Algorithm (ACO-PID) for the Position Control of a Multi-articulated System," *J. Robot. Control*, vol. 4, no. 3, pp. 289–298, 2023, doi: 10.18196/jrc.v4i3.17709.
- [51] P. Chotikunnan and Y. Pititheeraphab, "Adaptive P Control and Adaptive Fuzzy Logic Controller with Expert System Implementation for Robotic Manipulator Application," *J. Robot. Control*, vol. 4, no. 2, pp. 217–226, 2023, doi: 10.18196/jrc.v4i2.17757.
- [52] F. Umam, M. Fuad, I. Suwarno, A. Ma'arif, and W. Caesarendra, "Obstacle Avoidance Based on Stereo Vision Navigation System for Omni-directional Robot," *J. Robot. Control*, vol. 4, no. 2, pp. 227–242, 2023, doi: 10.18196/jrc.v4i2.17977.
- [53] J. Díaz-Téllez, R. S. García-Ramírez, J. Pérez-Pérez, J. Estevez-Carreón, and M. A. Carreón-Rosales, "ROS-based Controller for a Two-Wheeled Self-Balancing Robot," *Journal of Robotics and Control (JRC)*, vol. 4, no. 4, pp. 491–499, 2023.
- [54] M. H. Widianto and B. Juarto, "Smart Farming Using Robots in IoT to Increase Agriculture Yields: A Systematic Literature Review," *J. Robot. Control*, vol. 4, no. 3, pp. 330–341, 2023, doi: 10.18196/jrc.v4i3.18368.
- [55] I. Zaway, R. Jallouli-khlif, B. Maalej, and N. Derbel, "A Robust Fuzzy Fractional Order PID Design Based On Multi-Objective Optimization For Rehabilitation Device Control," *J. Robot. Control*, vol. 4, no. 3, pp. 388–402, 2023, doi: 10.18196/jrc.v4i3.18411.
- [56] T. A. Assegie, T. Suresh, R. Purushothaman, S. Ganesan, and N. K. Kumar, "Early Prediction of Gestational Diabetes with Parameter-Tuned K-Nearest Neighbor Classifier," *J. Robot. Control*, vol. 4, no. 4, pp. 452–457, 2023, doi: 10.18196/jrc.v4i4.18412.
- [57] T. Q. Ngo, T. T. H. Le, B. M. Lam, and T. K. Pham, "Adaptive Single-Input Recurrent WCMAC-Based Supervisory Control for De-icing Robot Manipulator," *J. Robot. Control*, vol. 4, no. 4, pp. 438–451, 2023, doi: 10.18196/jrc.v4i4.18464.
- [58] A. Ubaidillah, H. Sukri, U. T. Madura, J. Timur, and C. Author, "Application of Odometry and Dijkstra Algorithm as Navigation and Shortest Path Determination System of Warehouse Mobile Robot," *J. Robot. Control*, vol. 4, no. 3, 2023, doi: 10.18196/jrc.v4i3.18489.
- [59] M. A. Abdelghany, A. O. Elnady, and S. O. Ibrahim, "Optimum PID Controller with Fuzzy Self-Tuning for DC Servo Motor," *J. Robot. Control*, vol. 4, no. 4, 2023, doi: 10.18196/jrc.v4i4.18676.
- [60] P. Raksincharoensak, M. Nagai, and M. Shino, "Lane keeping control strategy with direct yaw moment control input by considering dynamics of electric vehicle," *Vehicle System Dynamics*, vol. 44, pp. 192–201, 2006, doi: 10.1080/00423110600870006.