

Trajectory Planning of Five DOF Manipulator: Dynamic Feed Forward Controller over Computed Torque Controller

Maged M. Abou Elyazed,
Mohamed H. Mabrouk, Mootaz E. Abo Elnor, Hussein M. Mahgoub
Mechanical Equipment Department
M.T.C.
Cairo, Egypt

Abstract— Industrial robot control has been a challenge for decades especially for those manipulators which are assigned to work in hazardous environment. The DFF (Dynamic Feed-Forward) controller and the CTM (Computed-Torque Method) based controller, are generally used for dynamic control of industrial robots. This paper presents a comparison of the two controllers applied on a five DOF manipulator case study. To verify the design of the controllers, the dynamic models of the manipulator used in this work are developed in two different ways. Firstly, the model is designed through Newton-Euler method for real time computation. Secondly the model is designed through simulations using Matlab Sim-mechanics for evaluating the developed controller. The experimental setup depends on using both controllers with a five DOF educational manipulator. For data analysis, a comparison between the computed torque controller and the dynamic feed forward controller is held on the same system under same conditions. The results show the efficiency of using the dynamic feed forward controller over the computed torque controller as a basis for manipulators' trajectory planning.

Keywords— *Graphical User Interface, Industrial Robots, Robot Modeling And Control.*

I. INTRODUCTION

It is well established that robot manipulators are dynamically coupled, time-varying, and highly nonlinear systems that are extensively used in industrial applications. The traditional manipulator control schemes are based on independent joint control or classical linear control as a Single-Input/Single Output (SISO) system and any coupling effects due to the motion of the other links can be treated as a disturbance.

Practically, Euler-Lagrange dynamic equations of robot form a nonlinear and Multi-Input/Multi Output (MIMO) multivariable system [1]. While the end effectors of the robot manipulators are to follow some desired trajectories as accurate as possible, the trajectory tracking problem becomes more complicated.

A conceptually simple nonlinear controller, called computed-torque controller, can fully compensate the nonlinear Coriolis and centripetal forces (continuous nonlinearities) in the robot motion and lead to high accuracy control for a variety of robot speeds and different shape workspace. With the use of the robots in critical applications

like dealing with hazardous materials and other sensitive applications, the precise control of the robot arms has become an essential requirement. Motivated by such control requirements, for practical and complex control problem of robot manipulators, many research contributions have been reported in the field of robotic control schemes such as proportional-integration-derivative (PID) control, PD control, PI control [2], feed-forward compensation control (DFF) [3], adaptive control [4,5], variable structure control [6], computed torque control (CTC) [1,7,8].

This paper presents the separate use of CTC controller and DFF controller with a 5-DOF robotic manipulator case study for trajectory tracking problem. The performance of CTC controller is then compared to that of DFF controller through calculating of the error in robot end-effector position with respect to the desired position for each controller. The dynamics of the robot manipulator is modeled using Matlab® Sim-mechanics toolbox. We modify the toolbox to include online evaluation of a nonlinear gravity compensation term to instantaneously compute current robot configuration. The use of such toolbox allows the possibility to verify model-based control algorithm [9, 10].

II. MATHEMATICAL MODEL

A. Newton-Euler Model

Now the problem of computing the joints' torques corresponding to a given trajectory of a manipulator is considered. Assuming that the position, velocity, and acceleration of the joints are determined, the joint torques corresponding to the required trajectory can be calculated using the kinematics and the mass distribution information of the robot. The algorithm for computing joint torques includes two main processes. First, link velocities and accelerations are iteratively computed from link 1 to link n applying Newton-Euler equations for each link. Second, forces and torques of interaction and joint actuator torques are computed recursively from link n back to link 1. For the outward iterations from link 1 to link n (i from 0 to $n-1$), the algorithm equations for the case of all rotational joints manipulator of n joints are as follows [11],

$$\omega_{i+1}^{i+1} = R_i^{i+1} \omega_i^i + \dot{\theta}_{i+1} Z_{i+1}^{i+1} \quad (1)$$

$$\dot{\omega}_{i+1}^{i+1} = R_i^{i+1} \dot{\omega}_i^i + R_i^{i+1} \omega_i^i \times \dot{\theta}_{i+1}^{i+1} Z_{i+1}^{i+1} + \ddot{\theta}_{i+1}^{i+1} Z_{i+1}^{i+1} \quad (2)$$

$$\dot{V}_{i+1}^{i+1} = R_i^{i+1} (\dot{\omega}_i^i \times P_{i+1}^i + \omega_i^i \times (\omega_i^i \times P_{i+1}^i)) + \dot{V}_i^i \quad (3)$$

$$\dot{V}_{c_{i+1}}^{i+1} = \dot{\omega}_{i+1}^{i+1} \times P_{c_{i+1}}^{i+1} + \omega_{i+1}^{i+1} \times (\omega_{i+1}^{i+1} \times P_{c_{i+1}}^{i+1}) + \dot{V}_{i+1}^{i+1} \quad (4)$$

$$F_{i+1}^{i+1} = m_{i+1} \dot{V}_{c_{i+1}}^{i+1} \quad (5)$$

$$N_{i+1}^{i+1} = I_{c_{i+1}}^{i+1} \dot{\omega}_{i+1}^{i+1} + \omega_{i+1}^{i+1} \times I_{c_{i+1}}^{i+1} \omega_{i+1}^{i+1} \quad (6)$$

Inward iterations from link n to link 1 (*i* from n to 1)

$$f_i^i = R_{i+1}^i f_{i+1}^{i+1} + F_i^i \quad (7)$$

$$n_i^i = N_{i+1}^{i+1} + R_{i+1}^i n_{i+1}^{i+1} + P_{c_i}^i \times F_i^i + P_{i+1}^i \times R_{i+1}^i f_{i+1}^{i+1} \quad (8)$$

$$\tau_i = n_i^{Ti} Z_i^i \quad (9)$$

Where ω is the rotational velocity, $\dot{\omega}$ is the angular acceleration, \dot{V} is the linear acceleration from link to link, \dot{V}_c is linear acceleration of the mass center, F is the inertial force; and N is inertial Torque at mass center of each link. f and n are force and torque so that they appear as iterative relationships from higher-numbered neighbor to lower-numbered neighbor. Finally, R is rotation matrix, P is position vector from joint to joint, and P_c is position vector from joint to center of mass.

B. Matlab® Sim-mechanics Model

Sim-mechanics is a Matlab® toolbox that enable importing CAD model from a CAD environment to the Simulink environment as blocks of links connected by joints. Moreover, Simulink provides users with a tool to specify bodies and their mass properties, their possible motions, kinematic constraints, coordinate systems and the means of initiating and measuring motions. This makes it easier to do the analytical modeling without going through a complex modeling process. Firstly, a CAD model for our case study, the five DOF (RV-2AJ) robot manipulator, needs to be built. Autodesk Inventor® is used as a suitable platform for building such a model, as shown in Fig. 1. Then the manipulator CAD model is imported to Matlab® Simulink as an XML file by the aid of Sim-mechanics toolbox. The RV-2AJ robot manipulator model is shown in Fig. 2.

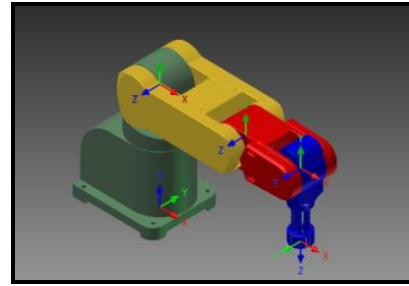


Fig. 1. RV-2AJ robot arm CAD model

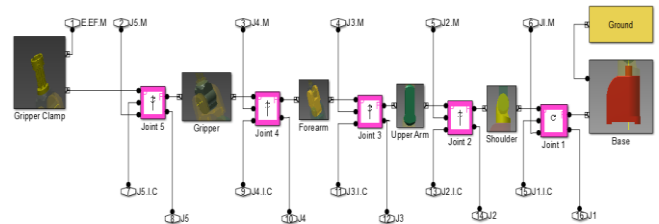


Fig. 2. RV-2AJ robot manipulator model

III. EXPERIMENTAL SETUP

In this paper, we use a five DOF Mitsubishi® RV-2AJ educational manipulator, as shown in Fig. 3. To feed the robot with the updated target point position calculated by the developed target positioning system, serial communication between the computer and robot controller need to be installed. MELFA basic language is used through MATLAB® environment to control the manipulator via the computer serial port [12]. Both CTC based controller and DFF controller are used one by one for controlling the robot manipulator for trajectory tracking problem. The design of both controllers is explained in the next section.

A graphical user interface (GUI) environment, shown in Fig. 4, is developed to facilitate the communication between the computer and the robot. Path defining which is simply determining the dominant points that define a path is performed easily through the interface. Also the interface gives charts which represent on-line evolution of robot end effector current position related to the defined path and robot joints operating parameters with time.



Fig. 3. Experimental setup (Mitsubishi® RV-2AJ) educational manipulator

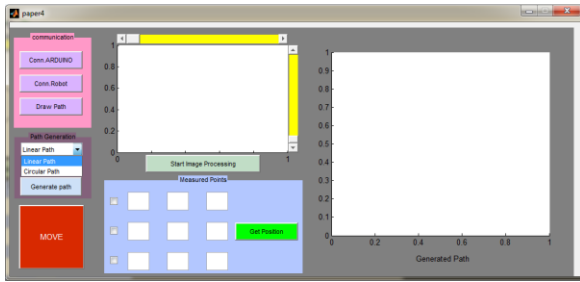


Fig. 4. Graphical user interface (GUI)

IV. DECOUPLING CONTROL

Decoupling control can be accomplished by using the feedback linearization of nonlinear systems, Lyapunov design, or inverse dynamics control. The work presented in this paper is influenced by feedback linearization control method to solve the problem of manipulator control. In some sense, the linearizing control law implements an inverse model of the system being controlled. The nonlinearities in the system cancel those in the inverse model; these, together with a servo law, results in a linear closed-loop system. There are many control algorithms for robot manipulators that depend on decoupling control. This paper used two of the most commonly used dynamic controllers for the industrial robot manipulators; the CTC (Computed Torque Controller) and DFF (Dynamic Feed-Forward) controller.

A. Computed Torque Controller (CTC)

In the previous section a model and the corresponding equations of motion of the robot is developed, and the dynamics are simply described by the following equation

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) \tag{10}$$

The problem of controlling a complicated nonlinear robot manipulator system can be handled by the partitioned controller scheme presented in [5, 11, 13]. This control scheme has two parts, the linearizing model-based portion expressed by equations from Eq. (11) to Eq. (13) and the servo-law portion expressed by equations from Eq. (14) to Eq. (15) as follows,

$$\tau = \alpha\tau' + \beta \tag{11}$$

$$\alpha = M(\theta) \tag{12}$$

$$\beta = V(\theta, \dot{\theta}) + G(\theta) \tag{13}$$

$$\tau' = \ddot{\theta}_d + K_v\dot{E} + K_pE \tag{14}$$

$$E = \theta_d - \theta \tag{15}$$

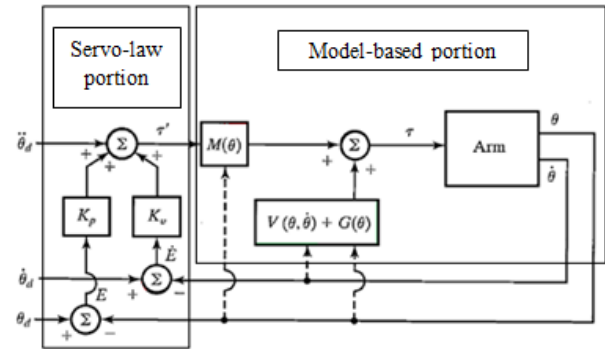


Fig. 5. A model-based manipulator-control system

The model-based manipulator-control system is shown in Fig. 5. Using equations from Eq. (10) to Eq. (15); it is easy to show that the closed-loop system is characterized by the error equation as follows,

$$\ddot{E}_d + K_v\dot{E} + K_pE = 0 \tag{16}$$

Equation (16) can be written on a joint-by-joint basis to be

$$\ddot{e}_d + K_v\dot{e} + K_p e = 0 \tag{17}$$

The computed torque controller computes the on-line dynamics of the manipulator, using the sampled joint position and velocity data. where; V is centrifugal and coriolis terms, $G(\theta)$ is gravity terms, θ is the joint positions, τ torque needed to achieve the given trajectory, θ_d joint target positions, E is the error between target and actual position. K_p controller proportional constant of the PD controller, K_v controller derivative constant of the PD controller.

It can be seen from Fig. 6 that the computed torque is calculated on-line by two function blocks ($\alpha = M(\theta)$, and $\beta = V(\theta, \dot{\theta}) + G(\theta)$). Both of these blocks consist of Matlab® m-file based on Newton Euler method to calculate required torque. The second block contains coriolis, centripetal, and gravity forces (continuous nonlinearities in the robot motion). These nonlinearities are calculated online using the measured values for positions (th1, th2, th3, th4, and th5) and the measured velocities (thd1, thd2, thd3, thd4, and thd5) for the five actuators respectively. These nonlinearities are delivered to the robot model as (C.T1, C.T2, C.T3, C.T4, and C.T5) to compensate its nonlinearities, therefore the robot system is linearized. On the other hand the error between the desired and the measured positions and velocities are calculated and then delivered to PD controller that work as a servo loop outside the linearized inner loop (linearized robot system) to get $(T'1, T'2, T'3, T'4, T'5)$, where $\tau' = \ddot{\theta}_d + K_v\dot{E} + K_pE$. The PD controller function block delivers the robot model with $\alpha\tau'$ as (B.T1, B.T2, B.T3, B.T4, and B.T5)

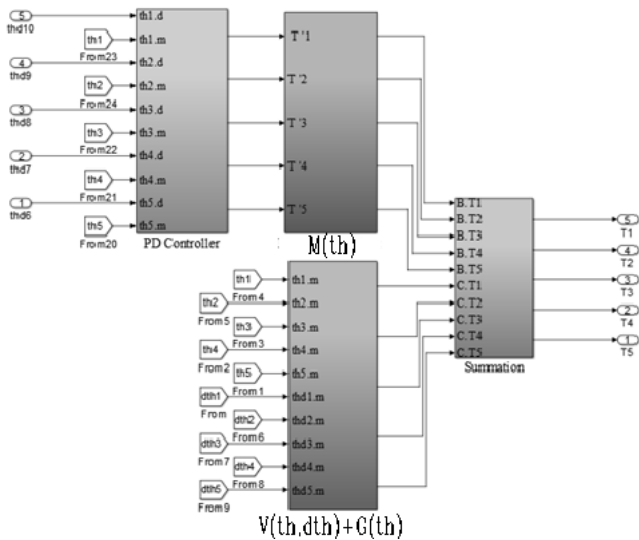


Fig. 6. Matlab® function block diagram for the CTC controller

V. RESULTS AND DISCUSSION

The results are discussed here in two main perspectives; the controllers performance during the path tracking process and the robustness of the controllers in spite of parameters uncertainties.

A. Controllers Performance during Path Tracking

For evaluating the performance of each controller, we use the CTC controller and the DFF controller to control the motion of the 5-DOF RV-2AJ robot manipulator to perform a linear quantic trajectory in three dimensional spaces forcing the path to have specific position, velocity, and acceleration at boundaries, so a fifth order polynomial is needed to express

$$X(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \tag{21}$$

B. Dynamic Feed Forward (DFF) Controller

DFF controller is usually used as an alternative to the CTC controller for the on-line computation requirements. The dynamic model is computed as a function of the desired path only, and so when the desired path is known in advance, values could be computed “off-line” before motion begins. At run time, the pre-computed torque histories would then be read out of memory. The control scheme for this kind of controllers is shown in Fig. 7.

It is clearly seen in Fig. 7 that the model based portion is outside the servo-law portion. It is worth mentioning that servo portion is the portion that work to eliminate the error using the feedback measured positions and velocities. It can be described in equation form from Eq. (18) to Eq. (20) as follows

$$\tau = T1 + T2 \tag{18}$$

$$T1 = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) \tag{19}$$

$$T2 = EK_p + \dot{E}K_v \tag{20}$$

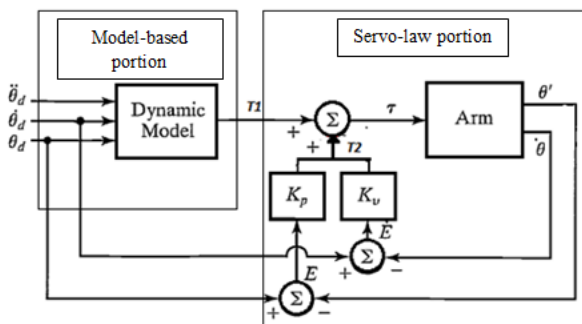


Fig. 7. Control scheme with the model-based portion "outside" the servo

The other coordinates kinematic behavior is determined as functions of (X) by a geometric space curve. In this case all velocities and accelerations at the start and the end are assigned to be zero. Fig. 8 and Fig. 9 respectively show the time evolution of position, velocity, and acceleration along the presented path for the robot end-effector and the robot joints.

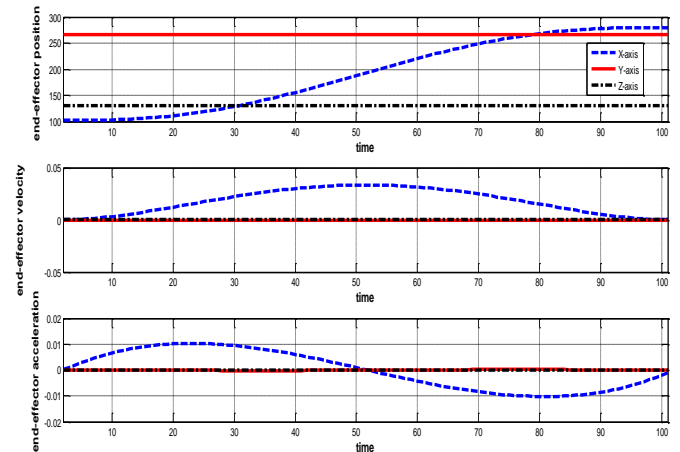


Fig. 8. Robot end-effector position, velocity, and acceleration along the along the quantic path versus time.

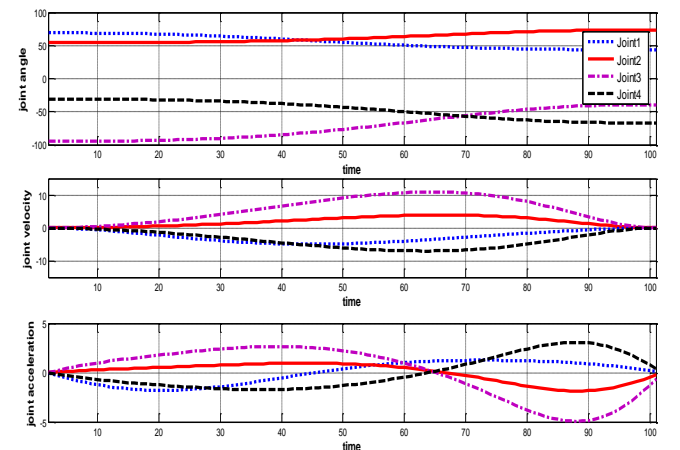
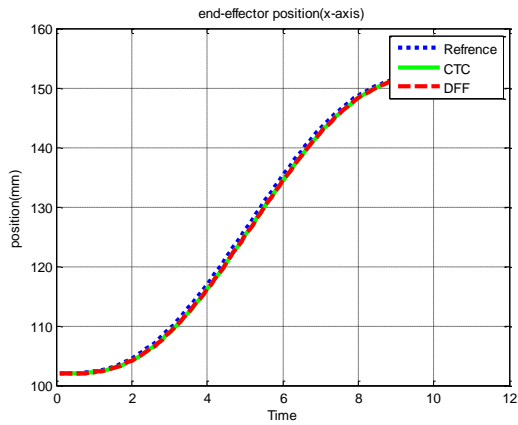
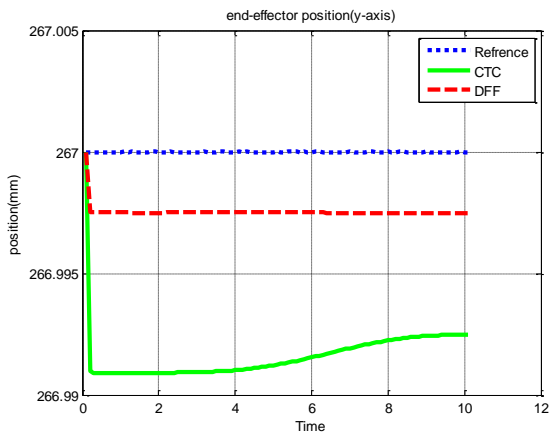


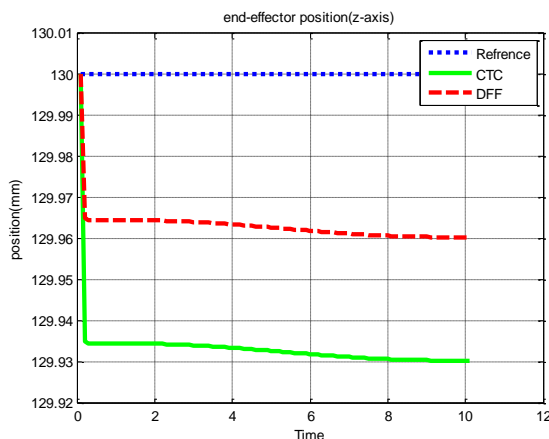
Fig. 9. Robot joints position, velocity, and acceleration along the quantic path versus time



(a)



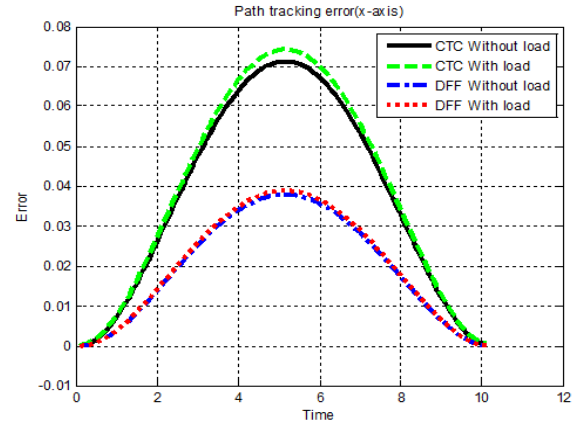
(b)



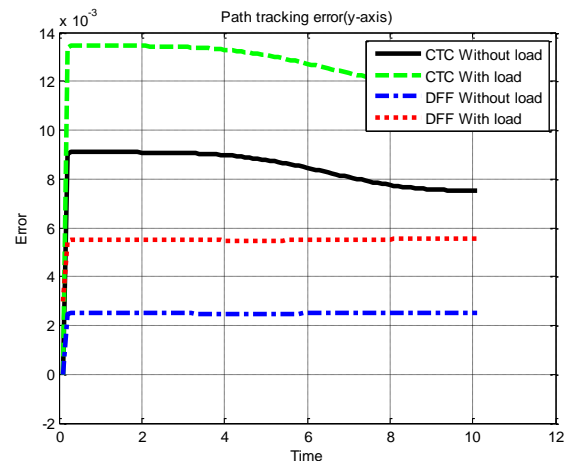
(c)

Fig. 10. Path tracking performance for end-effector (a) x-axis, (b) y-axis, and (c) z-axis

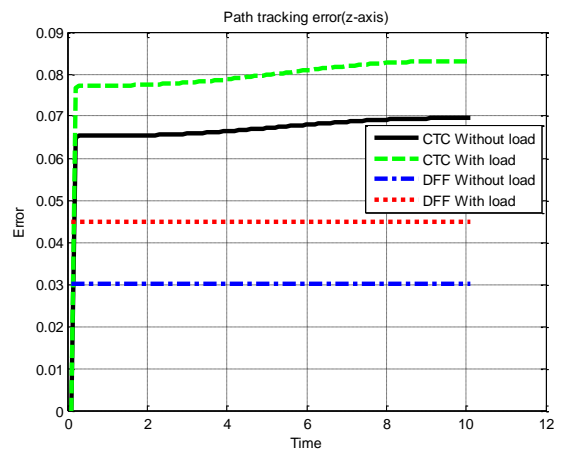
Fig. 10 (a, b, c) shows the end-effector position in x, y, and z coordinates respectively versus time using the CTC controller presented in solid line and the DFF controller presented in dashed line. The figure shows that both controller matching the desired Cartesian path, however results show that path tracking performance for DFF controller is closer to the reference path than that of CTC controller especially for y and z axes.



(a)



(b)



(c)

Fig. 11. End-effector position error in (a) x-axis, (b) y-axis, and (c) z-axis

Fig. 11 (a, b, c) respectively show the end-effector position error along x, y, and z axes versus time along the path for both CTC and DFF control methods with and without load. It is obviously seen that DFF have a better accuracy than the CTC. In the other hand for both controllers, accuracy has a noticeable change with load adding, but DFF still more accurate than CTC. These results verify the superiority of DFF controller over the CTC controller. The maximum error along the path with CTC method is about 78×10^{-6} m, in the other hand, the maximum error along the path with DFF method is about 45×10^{-6} m.

B. Robustness of the Controllers to Uncertainties

In robot control algorithm applications the problem of handling the uncertain parameter is very important. In this work, the controller robustness is evaluated by using uncertain payload mass where the same controllers are used to control the same robot in two cases. The first case is when the robot grips the maximum gripping load and the other case when it has an empty gripper.

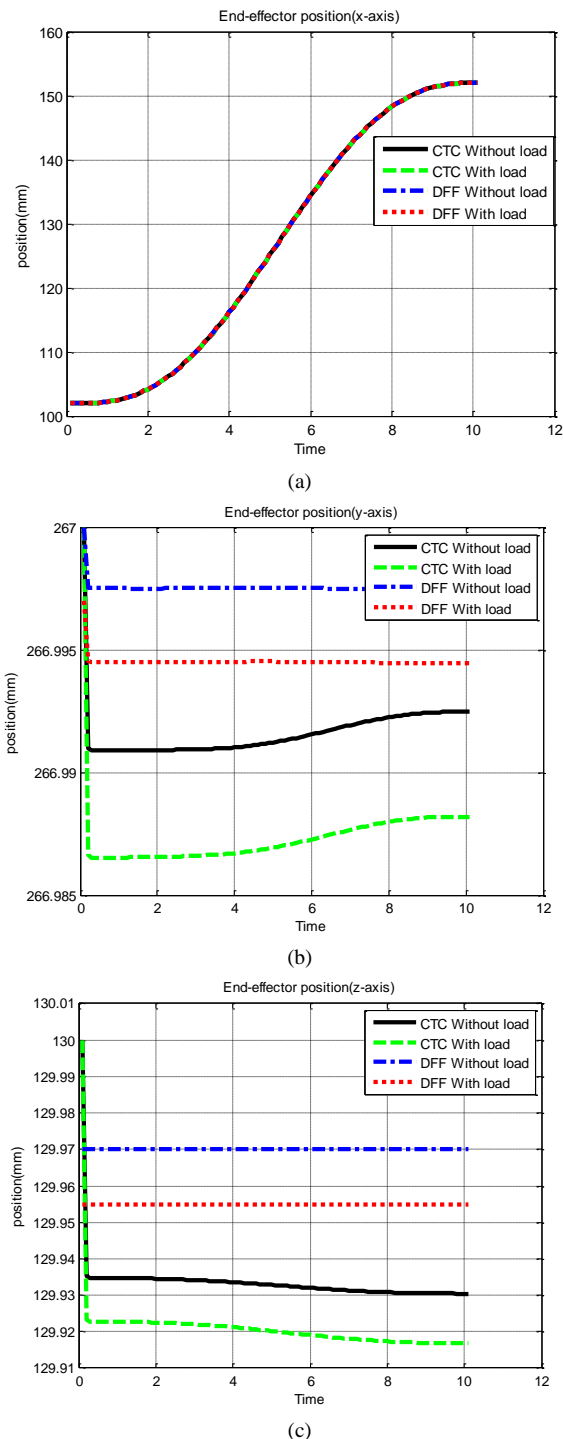


Fig. 12. Path tracking performance for the end-effector with & without load for (a) x-axis, (b) y-axis, (c) z-axis

Fig. 12 (a, b, c) shows the robot end-effector position in x, y and z coordinates respectively versus time using the CTC controller and the DFF controller both with and without load. The figure shows that the path error becomes larger when load is added, especially in case of CTC. Error increases because of the sensitivity of these methods to the model parameters. So the accurate parameters are necessary for dynamic control. It can be noticed that path error become larger when uncertainties are added, especially in case of CTC. The error increases because of the sensitivity of these methods to the model parameters. So the accurate parameters are necessary for dynamic control.

CONCLUSIONS

Robot path planning is one of the most important tasks especially for manipulators that need to follow a special path in hazard environments. In this paper, the performance of two controllers, dynamic feed forward (DFF) controller and the computed torque method (CTM) based controller, have been experimentally investigated using both controllers with a 5 DOF educational robot manipulator (Mitsubishi® RV-2AJ). The results show that the designed controllers are effective and useful. Also the results prove that DFF controller is better than CTC based controller, and more Robust. For some extent of system parameter uncertainties, the effectiveness of dynamics based model is still superior to the conventional one. But for the best performance, exact information of system parameters is required.

REFERENCES

- [1] F.L. Lewis, C.T. Abdallah, and D.M. Dawson, Control of Robot Manipulators, New York: Macmillan, 1993.
- [2] J.Y.S. Luh, "Conventional controller design for industrial robots-a tutorial", IEEE Trans. Systems Man Cybernet. 13 (1983) 298–316.
- [3] P.K. Khosla, T. Kanade, "Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulator", Int. J. Robot. Res. 7 (1988) 18–28.
- [4] J.J.E. Slotine, W. Li, "Adaptive manipulator control: a case study", IEEE Trans. Automat. Control 33(1988) 995–1003.
- [5] FarzinPiltan, N. Sulaiman, A. Jalali& F. Danesh Narouei, "Design of model free adaptive fuzzy computed torque controller: applied to nonlinear second order system." International Journal of Robotics and Automation, 2 (4):232-244,2011.
- [6] Mark W.Spong and M.Vidyasagar, Robot Dynamics and Control, John Wiley & Sons,2004.
- [7] Zuoshi Song, Jianqiang Yi, Dongbin Zhao, Xinchun Li, "A computed torque controller for uncertain robotic manipulator systems: fuzzy approach" Fuzzy Sets and Systems 154 (2005) 208–226, China.
- [8] C. S. Kim, K. W. Lee, "Robust control of robot manipulators using dynamic compensator", Control and Automation, 2009.ICCA 2009. IEEE International Conference, 9-11 Dec., pp 10071012,2009
- [9] FarzinPiltan, Mohammad Hossein Yarmahmoudi, "PUMA-560 robot manipulator position computed torque control methods using matlab/simulink and their integration into graduate nonlinear control and matlab courses" IJRA, Volume (3): Issue (3): 2012
- [10] DanutReceanu "Modeling and simulation of the nonlinear computed torque control in simulink/matlab for an industrial" Tech Science Press SL, vol.10, no.2, pp.95-106, 2013
- [11] John J. Craig, Introduction to Robotics Mechanics and Control, third edition, Pearson Prentice Hall, 2005.
- [12] Mitsubishi Melfa RV-2AJ manipulator user manual, 2010.
- [13] Reza N. Jazar, Theory of Applied Robotics, second edition, Springer Science+Business Media, LLC 2010