International Journal of Mathematical, Engineering and Management Sciences
Vol. 8, No. 4, 710-737, 2023
https://doi.org/10.33889/IJMEMS.2023.8.4.041

Dynamic Analysis and Control of Redundant Manipulator for Agricultural Applications in a Virtual Environment

A. Sridhar Reddy
Department of Mechanical Engineering,
AU College of Engineering, Andhra University, Visakhapatnam, 530003, Andhra Pradesh, India.
Corresponding author: asridhareddy@andhrauniversity.edu.in

V. V. M. J. Satish Chembuly
Department of Mechanical Engineering,
Aditya College of Engineering and Technology, Surampalem, 533437, Andhra Pradesh, India.
E-mail: vvmjsatish.chembuly@acet.ac.in

V. V. S. Kesava Rao
Department of Mechanical Engineering,
AU College of Engineering, Andhra University, Visakhapatnam, 530003, Andhra Pradesh, India.
E-mail: prof.vvskrao@andhrauniversity.edu.in

(Received on September 21, 2022; Accepted on April 27, 2023)

Abstract
In the development of automated manipulators for fruit and vegetable picking technologies, the challenge of ensuring an efficient, stable, and loss-free picking process has been a complex problem. In such an environment, manipulators require the most efficient and robust control for effective operations. In this paper, a serial 9-DOF redundant manipulator (1P8R) is proposed with various controllers for trajectory tracking problems in agricultural applications. The dynamic analysis of redundant manipulator has been carried out using the Recursive Newton-Euler method. The joint configurations of the robot are determined using optimization techniques for specific Task Space Locations (TSLs) by avoiding obstacles. The process of generating joint trajectories has been implemented by considering the cubic polynomial function. For the task of controlling the robot trajectory tracking in the virtual agricultural environment, different combinations of Proportional (P), Integrative (I), Derivative (D), and Feed-Forward (FF) controllers are employed, and a comparative analysis has been performed among these controllers. To verify the performance of the manipulator, simulations are carried out in a virtual environment using Simulink software. Results show that the robot is able to reach specific TSL accurately with better control and it is found that the implementation of Feed-Forward and PID-CTC controllers has better performance in a complex environment.

Keywords- Agricultural robot, Redundant manipulator, Dynamic analysis, PD/PID controllers, Matlab, Simulink.

1. Introduction

Different agricultural environments combined with intensive production require the development of robust systems in a short period of time and at a low cost. Despite recent developments in agricultural automation, labour is still necessary for high-value commodities like fruit and vegetable crops. For the production of a fresh market fruits like apples and guava, a large seasonal workforce is needed. To overcome the shortage of labour in the agricultural field, the robot should be able to perform operations like crop seeding, crop weeding, selective spraying, thinning and pruning, fruit picking, etc.

The environmental conditions of the crop field vary from one another due to these factors: (i) variation of crops- including technical challenges in fruit picking; (ii) variation of objects- including identification of position, size, shape, colour, and occlusion by tree branches; and (iii) environmental variation-including lighting conditions, are causing difficulties in the development of robots for commercial purposes (Bac et

al., 2014). A prototype robot is designed and developed by Belforte et al. (2006) for precision spraying and fertilization in the greenhouse environment. The designed robot has relatively low productivity per hour when compared with a human operator, but it is capable of performing given tasks continuously for indefinite time periods. Bak and Jakobsen (2004) addressed path generation for stability conditions and the path tracking mechanism of a four-wheel autonomous robot maneuver in the agricultural field for weed detection. Pons et al. (1996) presented a manipulator design based on parallelogram configuration, which gives better dynamic performance under various conditions when compared to joint constructors. The designed configuration achieves better control of manipulator motion by minimizing the centripetal and coupling torques. A 6-DOF industrial robot is taken as the manipulator for fruit plucking applications, which include a new design of suction gripper (Baeten et al., 2008). The chosen robot can execute the plucking cycle period in an average of 5 seconds, as verified experimentally. Similarly, a redundant manipulator (7-DOF) is proposed by Silwal et al. (2017) for grasping apples. Han et al. (2007) designed an optimized agricultural manipulator for vegetable picking in the greenhouse. In the article, Peilin Li et al. (2011) point out the merits and demerits of agricultural harvesting robots, which have a low success rate, slow operational speeds, trajectory error, object reorganization capability, operational stability, accuracy, high operational cost, maintenance and a wide variety of plant features. These factors are preventing commercialization and encourage further investigation on autonomous harvesters. To automate the process, robots need to be intelligent to conduct a wide range of agricultural operations like spraying and cutting at the targeted location with point-to-point motion (Edan et al., 2009).

The basic configuration of the agricultural robot manipulator is designed/ chosen for agricultural applications by considering the agricultural plant features and available 3D workspace. A brief review on manipulators by Tillett (1993) for the handling of biological material in horticultural industry is presented. The requirements of horticultural manipulators over the industry requirements like geometrical configuration, joint actuator choices and control strategy for the manipulator are highlighted in this paper. Kondo et al. (1993) designed a 7-DOF manipulator for agricultural applications using key parameters such as manipulability, operational space, and posture diversity. A multi-purpose agricultural robot with the configuration of Revolute-Revolute-Prismatic-Revolute- Revolute for berry thinning, spraying and bagging operations by utilizing various different end-effectors in the vineyards by Monta et al. (1995). A 7- DOF robot manipulator was designed for harvesting cucumbers by considering the characteristics of cucumbers (Arima and Kondo, 1999). Also attached a 4-wheel traveller device to move the robot in between the plants. A 9-DOF multi-purpose redundant manipulator is developed by Baur et al. (2012) to access the plant in all directions for selective fruit-picking operations. Effective harvesting and trajectory tracking of the developed manipulator require inverse kinematics, path planning, and motion control algorithms.

The vision system gives information about position, i.e., TSL and orientation of the fruits and information about objects lie in the robot workspace. The manipulator of the robot needs to reach TSL with the required orientation by avoiding collisions with other obstacles. Computing the IKs for the manipulator is required for a collision-free harvesting configuration. There is a non-existence of IK solutions through an analytical method; only a few manipulators with particular geometric configurations have an analytic solution in closed form (Chapelle and Bidaud, 2004). Instead of this, the IK problem can be solved by a numerical method based on iterative procedures. Nearchou (1998) and Parker et al. (1989) discussed the procedures for solving IK based on a non-linear programming problem, and these are easily adapted for different manipulators. Van Henten et al. (2010) reformulated a non-linear programming problem and solved it using a Genetic Algorithm (GA). The authors suggest combining a modified potential field method with a numerically based Jacobian approach (Park et al., 2020) to solve IK in real-time. The obtained IK solutions, i.e., joint variables, joint limit avoidance (Iossifidis and Schoner, 2006), and singularity avoidance (Yahya et al., 2012), of the robot manipulator need to be considered for the motion planning algorithm.

To achieve high production rates, the speed of the collision-free motion planning algorithm must be high. In order to meet this requirement, it is needed to get IK solutions in a few iterations and also quickly generate picking paths. In order to establish a path, the rapidly exploring random tree (RRT) algorithm creates random samples in the configuration space. For robots with a high degree of freedom, this approach works well. The Bi-RRT technique was proposed by Lavelle and Kuffner (2001) to shorten computation time by simultaneously traversing the tree at the initial and target states. Only a few authors have documented collision-free path planning for agricultural robots in the literature, where the majority of studies are concentrated on industrial robots in unrestricted environments. Bac et al. (2016) proposed constrained path planning for fruit-picking robots in agricultural environments. Cao et al. (2019) optimized the RRT algorithm using the potential field method and genetic algorithm for path planning of fruit-picking robots. It is noted that due to the presence of more tree branches and obstacles in dense areas, the motion planning time is increasing and there is a reduction in production. Among these robots, manipulators are highly nonlinear-coupled systems during operations. Sensor errors and external interference will cause uncertainties in the model (Zhang et al., 2020). In order to perform the required tasks, the study of a mathematical model and control laws is required. By using proper programming tools, simulations are carried out for different tests, like position tracking, and velocity control, etc. (Urrea and Kern, 2012).

Modelling and simulation of robots using different software will simplify the process of designing and inspecting the robots in the real world. Simulations are important for estimating and predicting the performance of the robot. Furthermore, it plays an essential role in the evaluation of the automation process. Dynamics is the study of systems that experience state changes over time. The development of the dynamic model of a robot is essential for motion simulation, structure analysis, and control algorithm design. From this information (dynamic modelling), joints, actuators, and transmission systems were designed. The dynamic analysis of robot manipulators can be formulated as "Forward Dynamics" (FD) and "Inverse Dynamics" (ID). In the FD joint force/torque vector, joint variables (position, velocity, and acceleration) are estimated. This method of formulation is useful for simulating or analyzing the robot manipulator. Whereas in ID, for the given trajectory point/variable (position, velocity, and acceleration), it needs to find joint torques. This type of formulation of dynamics is useful for controlling the manipulator (Craig, 2006). To obtain the manipulator's motion equations, several methods are available: these are '*Euler-Lagrange (E-L)* formulation', '*Newton- Euler (N-E)* formulation', '*Kane's* equations of motion', the *D'Alembart* principle, and equations based on the orthogonal complements of the velocity-constraint matrices (Saha, 2014). The *Euler-Lagrange* equation of motion was based on the concept of generalized coordinates and a scalar function. When generalized coordinates are chosen independently, constraints forces are eliminated and made suitable for motion control and simulation. This approach is based on energy balance in dynamic analysis. The *Newton-Euler* equation of motion contains vector cross-product terms and allows obtaining the model in a recursive form of Outward (forward) and Inward (backward) recursive equations. This is the force balance approach in dynamic analysis. The Outward recursion propagates kinematic information, total forces, and moments exerted at the centre of mass of each link from the base reference frame to the end-effector frame. The Inward recursion propagates the forces and moments exerted on each link from the end-effector frame to the base reference frame. The computations are simpler in N-E, so it allows a shorter computing time than the E-L method (Lee, 1982). To track the workspace trajectory for a redundant serial manipulator, Hsu et al. (1989) developed a control law on the basis of the pseudo-inverse of the Jacobian in the general form of the manipulator dynamic equation.

Designing of robot manipulators for specific applications has been increasing day by day because of their high productivity and efficiency. Generally, robot manipulators are complex in structure with non-linearity's such as link couplings and friction. Hence, it becomes a challenging task to design or choose a proper controller for the robot manipulator to achieve the desired trajectory tracking. Complex dynamic

equations (mathematical modelling) and trajectory planning are necessary to control the robot manipulator. The torques obtained from the dynamic analysis are tuned with the controller and given to each joint of the manipulator, and these are time variants. Each joint of the manipulator will follow its joint-level trajectory supplied by the trajectory generator for error-free trajectory tracking at the end effector. In Kumar and Kumar (2017), various variants of controllers like PI, PD, and PID are used for controlling the redundant robot manipulators against external disturbances. A manipulation system for an agricultural robot for handling heavy materials is proposed by Sakai et al. (2008). The system is developed in the existence of parametric perturbation and uncertainty by using PI and PID controllers for different joints. In general, PID controllers are linear in nature and only function under precise knowledge of the parameters of the robotic manipulator and system linearization techniques; they are less robust to model uncertainties and parameter changes, and their performance suffers under external disturbances. A 7-DOF redundant manipulator is proposed for trajectory tracking (Dhyani et al., 2020) by linearization of non-linear manipulator dynamics with linear control techniques. In many industries, feedback controller structures for specific applications cannot be eventually modified. Mitigation of non-linear dynamics can only be performed via Feed-Forward controllers. In Grotjahn and Heimann (2002), a model-based Feed-Forward controller is discussed with various approaches to improve or observe the path accuracy. The expected torques for the desired joint motions of the manipulators are calculated using the Inverse dynamics model of the system. A 3-DOF harvesting robot manipulator with a vacuum-based end effector to harvest the apples has been developed by Zhang et al. (2020). The non-linear motion control of the manipulator is controlled using a Proportion-Integrative controller (PI). In 2009, Tatlicioglu et al. (2009) proposed a feedback-linearized adaptive controller using Lyapunov-based stability analysis for a redundant manipulator. This controller uses a least-squares estimation algorithm to adjust the parametric uncertainties in the dynamic model. Silwal et al. (2017) designed a 7-DOF robot for harvesting apples. The end-effector of this manipulator is able to execute the grasping of fruits in a purely open-loop manner (without sensors) with Feed-Forward control. Though modern control theory has come a long way, this article focused on the widely used controllers in industries. As these are well-developed, simple to design, and require less computational time.

In the available literature, very few authors reported on redundant manipulators for agricultural robots. The designed agriculture harvesting robots are limited to some applications with low success rates because of the variations caused by the variety of plant features. To improve productivity and efficiency in agriculture, the design and control of robots are considered important areas by the research community.

In this work, a 9-DOF robot redundant manipulator with a 1P8R configuration is selected to access the wide variety of plants in all possible directions and also to get more work volume for reaching the selected task space locations. The chosen redundant manipulator and variations in dynamics are observed using various controllers. The research contribution is the dynamic modelling of the robot manipulator for harvesting applications and also performing the controlled motion of the robot in an agricultural environment using different controllers, which are modelled in Simulink environment. Finally, dynamic simulations are performed in Simscape Multibody. The methodology adopted in this paper is presented in Figure 1. The nomenclature followed in this paper is tabulated in Table 1.

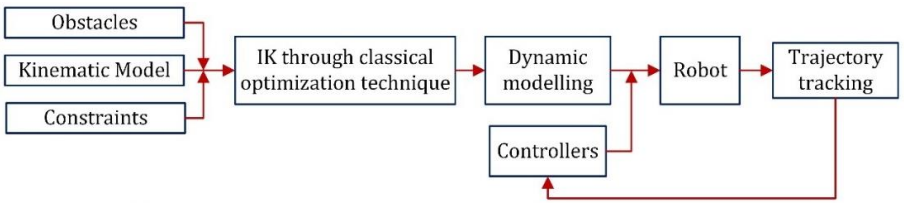


Figure 1. Flowchart of methodology.

Table 1. Nomenclature.

DOF	Degrees of Freedom
TSLs	Task Space Locations
1P8R	One prismatic eight revolute joints
PID	Proportional Integrative Derivative controller
FF	Feed-Forward controller
CTC	Computed Torque Control
IK	Inverse Kinematics
ID	Inverse Dynamics
DH	Denavit-Hartenberg
q, q_m, q_r	Joint variable, measure joint variable and required joint variable
$M(q)$	Inertia matrix
τ	Torque
d_1	Joint variable for a prismatic joint. Units: *m*
θ	Joint variable for a revolute joint. Units: *rad*
I	Moment of inertia
qd, *qdd* & *qdm*, *qddm*	First and second-time derivatives of joint variable (q). Similarly, the first and second derivatives of the measured variable (qm)
*J*1, *J*2, ..., *J*9	*J*1 is the prismatic joint, *J*2, *J*3, . . , *J*9 are the revolute joints
K_i, K_p, K_d	Controller parameters
NA	Not Applicable

This paper aims to analyze the dynamic aspect of agricultural serial robot manipulators and investigate the impact of various control schemes on the manipulator's trajectory tracking performance. Kinematic modelling and trajectory generation for the harvesting robot are discussed in Section 2. In Section 3, the dynamics and control simulations are presented. Section 4 contains the results, followed by the conclusions in Section 5.

2. Kinematic Analysis and Trajectory Planning

2.1 Kinematic Modelling

Agricultural robots are required to perform tasks like harvesting, pruning, or spraying on different plants and fruits. Usually, the trellis for Bitter gourd and Tindora is placed above the ground at a height of 1 m. In the Guava field, fruits are grown above the ground at a range of 450 mm to 1300 mm. Based on the work volume, a 1P8R configuration was chosen for the manipulator (Reddy et al., 2022) to perform the tasks. In the selected manipulator, the first 6-links are equal in length, and the last 3-links constitute a wrist to allow the roll-pitch-yaw motion to the gripper of the manipulator. Denavit-Hartenberg (D-H) frames are assigned to the designed manipulator as shown in Figure 2, and the D-H parameters are tabulated in Table 2.

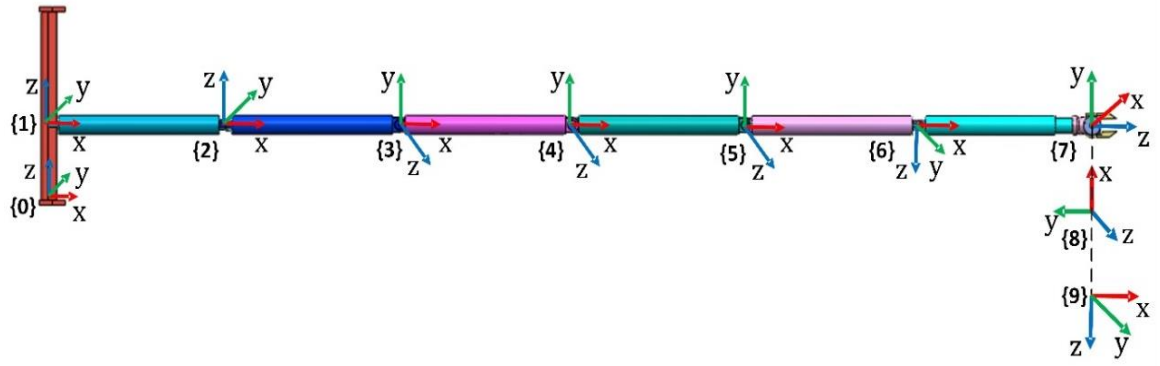


Figure 2. Solidworks model of the robot manipulator with D-H frames.

Table 2. DH parameters.

link	α	a	θ	d
1	0	l_1	0	d_1
2	90	l_2	0	0
3	0	l_3	0	0
4	0	l_4	0	0
5	90	l_5	0	0
6	–90	l_6	–90	0
7	–90	0	90	0
8	90	0	–90	0
9	0	0	0	0

The transformation between the frame and the base frame was performed using a generalized homogeneous transformation matrix to attain the position and orientation of the end-effector. The position and orientation required at the TSL become inputs for Inverse Kinematics. In the working environment tree trunks, branches, and Trellis are obstacles to the robot's motion. Using the vision system, a number of images are continuously taken in different directions. The outer boundaries of an object are considered a set of points. The extreme coordinates of these points are used to construct a bounding box for a particular obstacle. Collisions were detected whenever the point on the robot lay inside the bounding boxes, as discussed in (Chembuly and Voruganti, 2020). Joint variables were obtained by formulating an objective function to minimize the distance between the end effector and the TSLs, with constraints on joint limits. The problem is solved using the Sequential Quadratic Programming (SQP) technique (Boggs and Tolle, 1995), which is a classical optimization method. The required location (x_{id}, y_{id}, z_{id}) and orientation $(\alpha_{id}, \beta_{id}, \gamma_{id})$ in the agricultural environment are considered TSLs, for the end-effector. Similarly, (x_{ia}, y_{ia}, z_{ia}) and $(\alpha_{ia}, \beta_{ia}, \gamma_{ia})$ are taken as the actual position and orientation of the end effector. The joint variables are determined to minimize the error in the objective function formulated to reach TSLs as follows:

$$\text{Minimize: } f = [(x_{id} - x_{ia})^2 + (y_{id} - y_{ia})^2 + (z_{id} - z_{ia})^2 + (\alpha_{id} - \alpha_{ia})^2 + (\beta_{id} - \beta_{ia})^2 + (\gamma_{id} - \gamma_{ia})^2] + \sum_{i=1}^{m} C_i \quad (1)$$

2.2 Trajectory Generation

To perform agricultural operations sequentially, the end-effector of the robot moves from one TSL to the next TSL with the required orientation. These TSLs are considered via points for path generation. At each via point, joint variables are found by using Inverse Kinematics. To generate smooth motion for the manipulator with velocity and acceleration continuity, a third-order polynomial function was considered. Since cubic polynomials are continuous and differentiable, i.e., velocity and accelerations are continuous and smooth, and can be tuned to match a wide range of desired motion profiles. These polynomials can be expressed in closed form and are calculated using simple arithmetic operations, which makes them computationally efficient for real-time applications. For a 9-DOF robot manipulator, the joint position, velocity, and accelerations are computed using the following equations: For the cubic polynomial trajectory, three via points are considered; hence, the path can be approximated as four segments of the polynomial function as given in Equation (2).

$$\left.\begin{aligned}\theta_{i1}(t) &= a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 \\ \theta_{i2}(t) &= a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 \\ \theta_{i3}(t) &= a_{30} + a_{31}t + a_{32}t^2 + a_{33}t^3 \\ \theta_{i4}(t) &= a_{40} + a_{41}t + a_{42}t^2 + a_{43}t^3\end{aligned}\right\}. \quad (2)$$

where, $\theta_{i1}(t), \theta_{i2}(t), \theta_{i3}(t)$, and $\theta_{i4}(t)$ are the four numbers of cubic segments of i^{th} joint, $i = 1, 2, ..., 9$ and $a_{i0}, a_{i1}, a_{i2}, a_{i3}$ are the coefficients.

For the above cubic segments, the joint velocity is determined using the time derivative of joint variables, similarly, acceleration is determined by considering the time derivative of the velocities and represented in Equation (3).

$$\dot{\theta}_i(t) = a_{i1} + 2a_{i2}t + 3a_{i3}t^2 \text{ and } \ddot{\theta}_{i1}(t) = 2a_{i2} + 6a_{i3}t \quad (3)$$

The coefficients $a_{i0}, a_{i1}, a_{i2}, a_{i3}$ can be calculated at time t_0 and t_f by considering velocity and acceleration as boundary conditions.

3. Dynamics and Control Simulation

3.1 Iterative Newton-Euler Dynamic Formulation

As the trajectory of the manipulator was generated, torques needed to be determined for the motion of the manipulator. The dynamics are formulated using the N-E recursive algorithm. This algorithm computes link velocities and accelerations iteratively from the first link to the last link, and joint actuator torques are computed recursively from the last link to the first link. The forward and backward recursive equations for the frame $\{i\}$ to frame $\{i+1\}$ is given in Tables 3 and 4, respectively. The propagation of the velocity and acceleration of links is determined using the equations given in Table 1. The mass distribution of the links and the kinematics of the manipulator are known in Section 2. From the Newton-Euler equations, the inertial force and torque are also determined.

Table 3. Forward recursions (Newton-Euler).

The velocity propagation one link to the next link : ${}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}$
Angular acceleration transformation : ${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}$
Linear acceleration of each link-frame origin : ${}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i}\dot{v}_i)$
Linear acceleration of the centre of mass of each link : ${}^{i+1}\dot{v}_{C_{i+1}} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}} + {}^{i+1}\dot{\omega}_{i+1}({}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}}) + {}^{i+1}\dot{v}_{i+1})$
At centre of mass of each link, the inertial force and torque are computed using the Newton-Euler equation and it is given as: 1. Inertial force: ${}^{i+1}F_{i+1} = m_{i+1}\,{}^{i+1}\dot{v}_{C_{i+1}}$ 2. Torque : ${}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\omega_{i+1}$

Table 4. Backward recursions (Newton-Euler).

The force exerted on link *i* by link *i*-1 : ${}^{i}f_i = {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i$
The torque exerted on link *i* by link *i*-1 : ${}^{i}n_i = {}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1}$
Joint torque vector : $\tau = {}^{i}n_i^T\,{}^{i}\hat{Z}_i$

where,

ω, ν are angular and linear velocities.

$\dot{\omega}, \dot{\nu}$ are angular and linear accelerations.

R is the transformation matrix.

P_C is the centre of mass.

${}^{C}I$ is the inertia tensor of body written in a frame $\{C\}$, whose origin is located at the centre of mass.

f_i is the force exerted on link *i* by link *i*-1.

F is the force acting at centre of mass.

n_i is the torque exerted on link *i* by link *i*-1.

m is the total mass of the link.

After evaluation of Newton-Euler equations symbolically for the manipulator, the state-space equation for dynamic analysis is given in Equation (4) (Craig, 2006).

$$M(q)\ddot{q} + C(q,\ddot{q}) + G(q) + F(q,\dot{q}) = \tau \quad (4)$$

where, q is the joint variable representing displacement or angle vector, $\dot{q}$ is the velocity vector, $\ddot{q}$ is the acceleration vector. $M(q)$ represents the inertia matrix with $n \times n$ dimension, $C(q,\ddot{q})$ is the centrifugal and carioles force vector matrix with $n \times n$ dimension. $G(q)$ is the gravitational force vector with $n \times 1$ dimension. $F(\dot{q})$ is the joint viscous friction force vector with $n \times 1$ dimension. τ is the generalized actuator force/torque vector associated with the generalized coordinates.

3.2 Design of Controllers

The designing of closed-loop controllers for a system depends on the manipulated input and manipulated output variables. The linear-type controllers like PD and PID are widely used for controlling the position and velocity of the robot manipulators. These types of classic controllers are easy to operate and implement in order to achieve the desired set points (TSLs) and linearize the non-linear behaviour of the manipulator's joints in a simple manner. In this work, various controllers are implemented for effective trajectory tracking of the 9-DOF agricultural robot manipulator. The controllers and their combinations are enumerated as (i) PD, (ii) PID, (iii) Feed-Forward Control, (iv) PD-Computed Torque Control (PD-CTC), (v) PID-Computed Torque Control (PID-CTC), and (vi) Feed-Forward- 2DOF PID controller.

3.2.1 PD and PID Controllers

The schematic view of PD and PID controllers are presented in Figure 3 and Figure 4, respectively. The functionality of both controllers is similar, which computes and defines a weighted difference signal for each of the proportional, integral, and derivative operations by providing the set point weights. The output of these controllers is also equivalent to the summation of the proportional, integral, and derivative actions on the corresponding difference signals, where each action is weighted and adjusted in accordance with the gain settings.

In Figure 3 and Figure 4, q and $\dot{q}$ represent the desired position and velocity, respectively. q_m and $\dot{q}_m$ are measured positions and velocities of each joint of the manipulator. The K_p, K_d and K_i are the controller parameters, tuned based on the trial-and-error method.

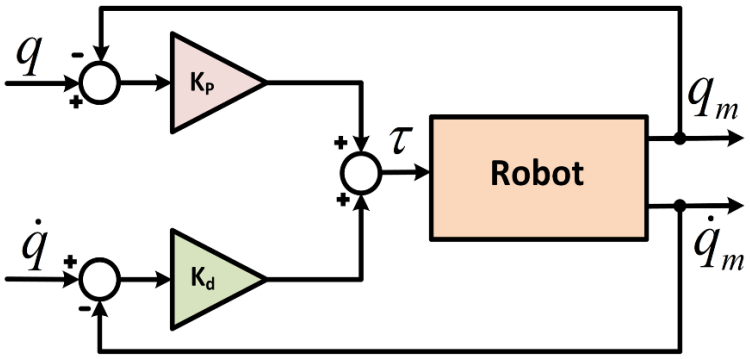


Figure 3. Schematic view of PD controller.

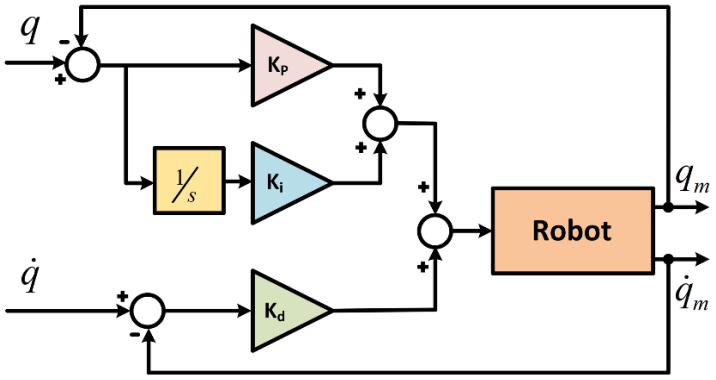


Figure 4. Schematic view of PID controller.

3.2.2 Feed-Forward Controller

The schematic diagram for the Feed-Forward controller is depicted in Figure 5. A Feed-Forward Controller (FFC) is combined with different feedback controllers to eliminate persistent disturbances that cannot be appropriately rejected with feedback controllers. In a 9-DOF manipulator, the required torque or force is obtained from the Feed-Forward controller and fed to the robot manipulator. The feedback recovers errors due to uncertainty in the inertial parameters, un-modelled forces, and/or external disturbances. To minimize the trajectory errors, the control parameters K_p and K_d in FFC are tuned using the trial-and-error method.

The torque generated by the Feed-Forward controller is given in Equation (5),

$$\left.\begin{aligned}\tau &= \{M(q)\ddot{q} + C(q,\dot{q}) + G(q) + F(\dot{q})\} + \{K_d(\dot{q} - \dot{q}_m) + K_p(q - q_m)\} \\ &= D(q,\dot{q},\ddot{q}) + \{K_d(\dot{q} - \dot{q}_m) + K_p(q - q_m)\}\end{aligned}\right\} \quad (5)$$

where, K_p and K_d are the position and velocity gain matrices, and D is the inverse dynamics function.

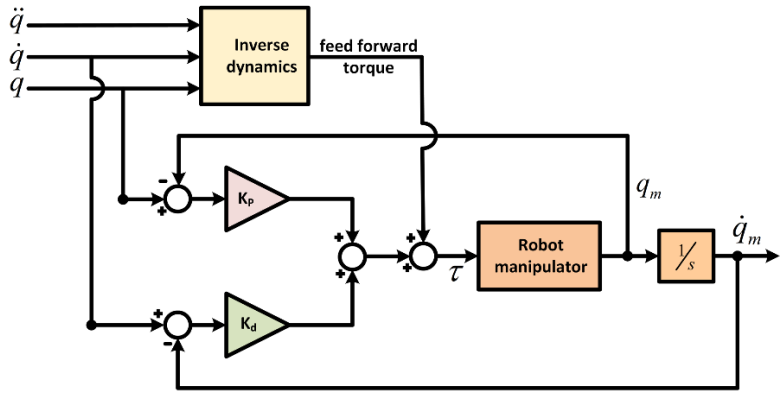


Figure 5. Schematic view of Feed-Forward controller.

3.2.3 PD-CTC and PID-CTC Controllers

The combination of PD and PID with Computed Torque Control is implemented on a 9-DOF manipulator for obtaining the desired output responses of the dynamic system. The inputs for both controller combinations are given using system inverse and are known as Computed Torque. CTC adopts the feedback linearization method so that the non-linearity's of the dynamic system is compensated with linear behaviour in a closed loop. The CTC is a more accurate and advanced controller when compared to PD and PID controllers. The complexity of constructing a CTC is dependent on the estimation of model parameters. The schematic diagrams of PD-CTC and PID-CTC are shown in Figures 6 and 7.

The control law for the PD-CTC controller is expressed in Equation (6) and Equation (7) by estimating the uncoupled error.

$$\tau = \hat{M}(q_m)\cdot u + \hat{C}(q_m, \dot{q}_m) \tag{6}$$

$$\left.\begin{aligned} u &= e\cdot K_p + \dot{e}\cdot K_d + \ddot{q}_m \\ e &= q_d - q_m \\ \dot{e} &= \dot{q}_d - \dot{q}_m \end{aligned}\right\} \tag{7}$$

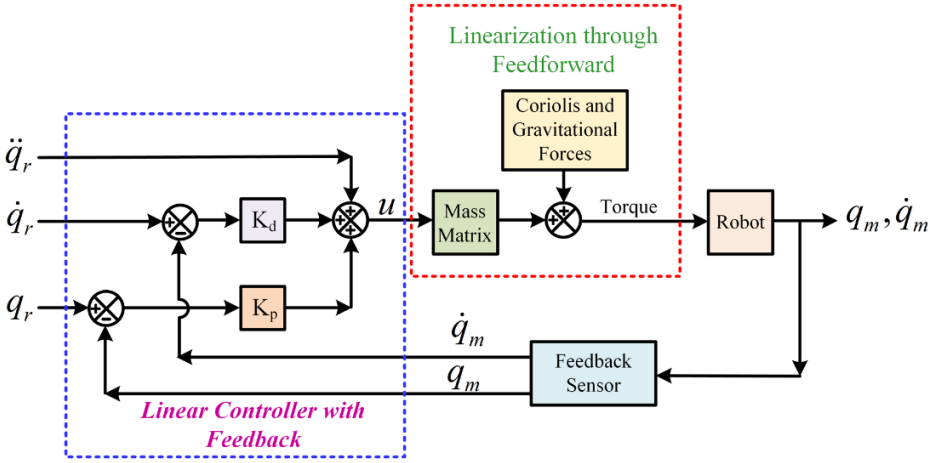


Figure 6. Schematic view of PD-CTC.

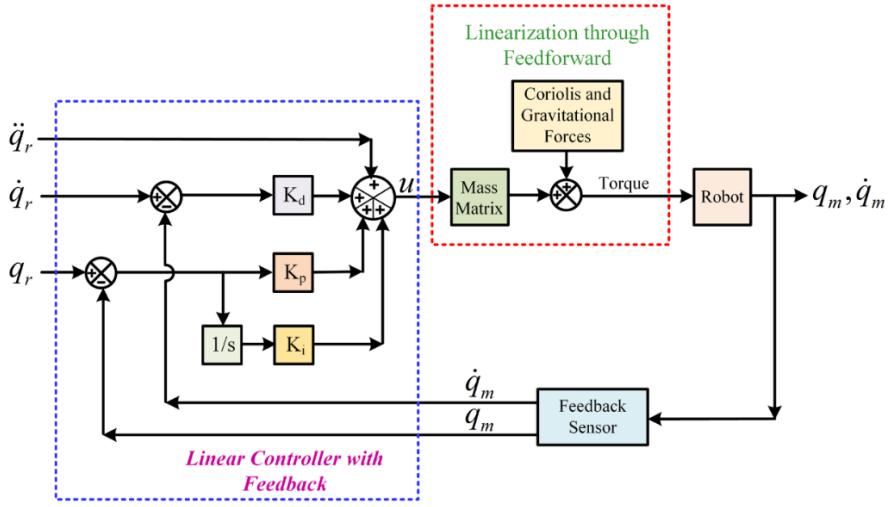


Figure 7. Schematic view of PID-CTC.

where,

$\hat{C}(q_m, \dot{q}_m)$ is an estimated model of $C(q_m, \dot{q}_m)$, that corresponds to the Coriolis, Centrifugal, frictional forces, and gravity effects.

$\hat{M}(q_m)$ is the estimated model of the inertial matrix $M(q_m)$.

The error can be expressed as given in Equation (8).

$$\ddot{e} + K_d\dot{e} + K_p e = \hat{M}^{-1}(\Delta M\ddot{q}_m + \Delta C + \Delta F) \tag{8}$$

where,

$\Delta M, \Delta C$ and ΔF represent the errors of the dynamic model.

Although it is well known that a PID controller has three configurable parameters, there are several issues to consider when modifying these values to provide a sufficient balance between system performance and robustness.

3.2.4 Feedforward- 2DOF PID Controller

Since there is only one feedback controller that the traditional PID controller has, it is more challenging to simultaneously guarantee a sufficient set point and disturbance responses. In a 2-DOF PID controller, a reference signal and system output are used as the input and output signals, respectively. For improving the robot's transient performance, a Feed-Forward gain is incorporated into the 2-DOF PID controller, as shown in Figure 8. From the theoretical robustness analysis, it is observed that both controllers, i.e., 2-DOF PID and Feed-Forward 2-DOF PID, attain the same results, but in real-time scenarios, the 2-DOF PID control scheme may have superior robustness if the Feed-Forward parameter is chosen appropriately in real-time.

The 2-DOF PID controller is decomposed into a single input, single output PID controller with a Feed-Forward controller and an error signal as input. The controller unit consists of a controller term and a filter term, computed as given in Equation (9).

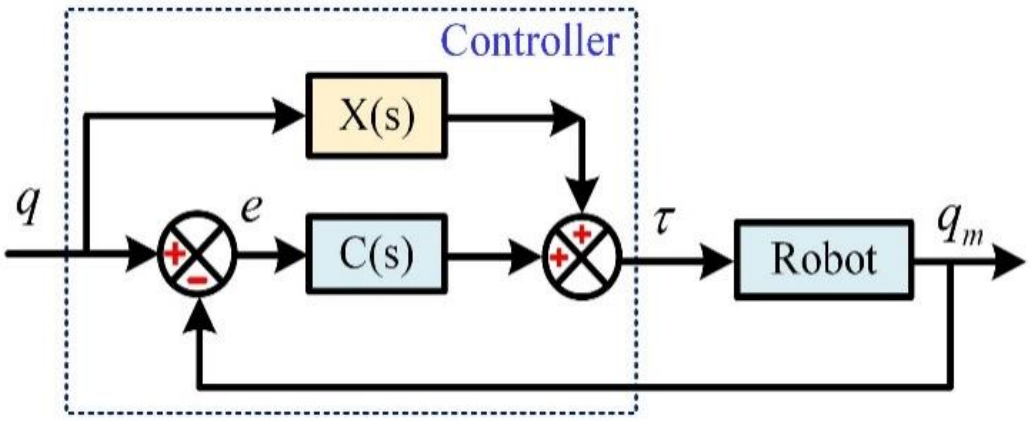


Figure 8. Schematic view of a Feed-Forward 2-DOF PID.

$$\left.\begin{aligned} C(s) &= K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1} \\ X(s) &= (b-1)K_p + \frac{(c-1)K_d s}{T_f s + 1} \end{aligned}\right\} \tag{9}$$

where, $C(s)$ is the controller, $X(s)$ is the filter, s is the frequency domine in Laplace Transformation, b - set point weight on a proportional term, c - set point weight on a derivative term, and T_f - derivative filter time.

In this work, 2-DOF PID controllers are designed for each joint in the manipulator individually and tuned for desired output responses, as shown in Figure 9.

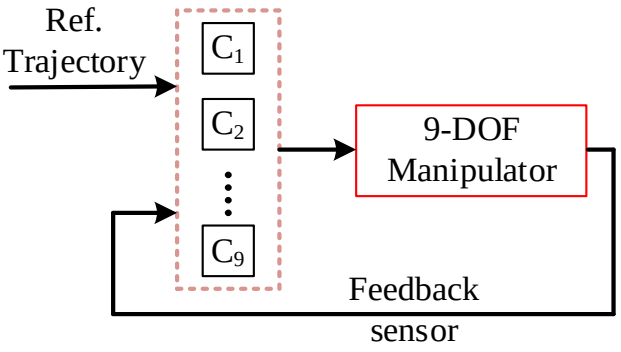


Figure 9. Closed loop Control for 9-DOF manipulator.

4. Results

4.1 Inverse Kinematics and Trajectory Generation

The task space locations, along with the orientation of fruit, are identified using a vision system in the agricultural environment. In this article, six TSLs were chosen in the guava field for fruit picking operations,

and Inverse Kinematics was solved for optimized collision-free posture of the robot manipulator, as mentioned in Section 2.1. The joint variables at each TSL are reported in Table 5. To move the end effector in the desired path for performing the agricultural task, joint variables are considered as 'via points' and fed into a '*polynomial trajectory generator*' for generating joint-level trajectories. The time step from one TSL to the succeeding TSL was given as 3 seconds, so the total time for the robot to complete its trajectory is 18 seconds. The joint-level trajectories of the manipulator are shown in Figure 10.

The trajectory tracking of a robot manipulator in different agricultural applications and virtual environments of bitter gourd, guava, and tindora fields has been modelled in Solidworks. The trajectory tracking of the end-effector in the guava field corresponding to the TSLs presented in Table 3 is plotted in Matlab and is shown in Figure 11, as is the working of the redundant manipulator in the bitter gourd environment.

Table 5. IK solutions of various TSLs.

S. No.	TSLs		Joint variables									error
	Position required (m)	Orientation required (radians)	(m)	(radians)								
			d_1	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	
1	(0.608, 0.743, 0.595)	(0.3491, 0.6981, 0.6109)	0.151	1.016	0.332	0.247	0.165	-0.217	0.349	0.698	0.610	3.20×10^{-7}
2	(0.478, 0.339, 0.902)	(0.2610, 0.4014 , 0.2094)	0.159	0.808	0.906	0.373	0.147	-0.167	0.261	0.401	0.209	3.17×10^{-7}
3	(0.398, -0.3, 0.896)	(0.0730, 0.3142, 0.4014)	0.149	-0.830	0.926	0.419	0.249	0.282	00.07	0.314	0.401	3.14×10^{-7}
4	(0.589, -0.548, 0.651)	(0.2618, 0.3665, 0.3142)	0.077	-0.828	0.329	0.615	0.191	0.428	0.261	0.366	0.314	1.05×10^{-7}
5	(0.40, -0.40, 0.70)	(0.4014, 0.2793, 0.5934)	0.082	-0.893	0.348	0.729	0.615	0.495	0.401	0.279	0.593	2.06×10^{-7}
6	(0.592, -0.68, 0.599)	(0.2094, 0.5585, 0.4538)	0.086	-0.989	0.384	0.308	0.214	0.226	0.209	0.558	0.453	5.20×10^{-7}

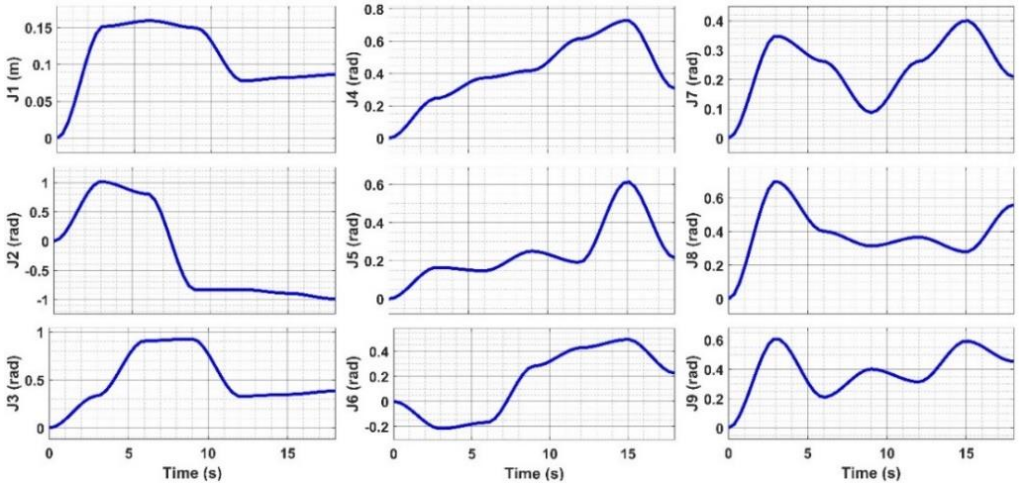
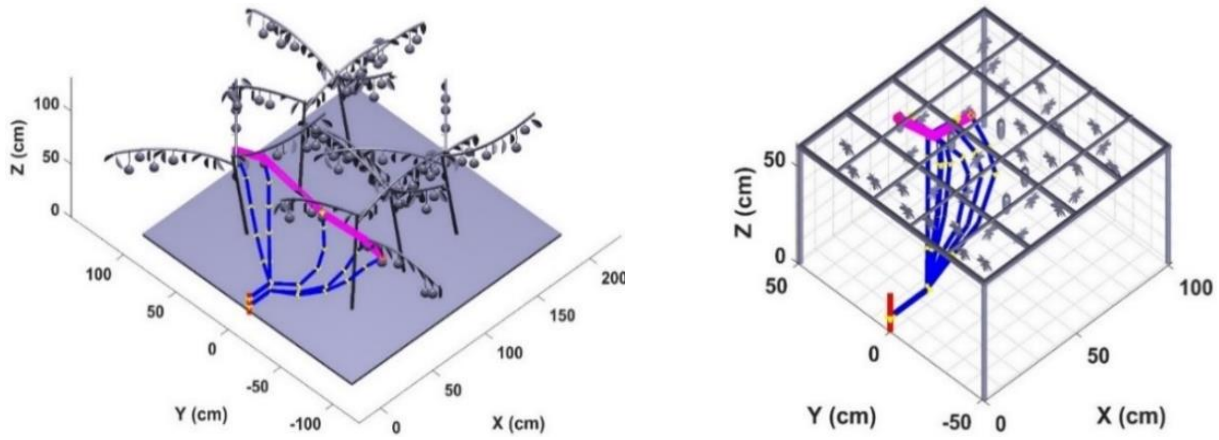


Figure 10. Joint trajectories for 9-DOF serial manipulator during time period.



a) Guava virtual environment b) Bitter gourd virtual environment

Figure 11. Trajectory tracking of a robot manipulator in the virtual environment.

4.2 Dynamic Analysis

A 9DOF (1P8R) serial manipulator is designed and simulated in *'Simulink/Simscape Multibody'* and multi-body simulations for observing the dynamic behaviour of the robot for trajectory tracking in the chosen virtual environments. The inertia properties of the robot manipulator are obtained by using 3D modelling software as given in Table 6.

For reaching the TSL, the required torque is necessary at each joint, determined by using the Recursive Newton-Euler method as discussed in Section 3.1. The variation of torques for complete joint-level trajectory generation is shown in Figure 12. The torque generated by the N-E method is fed to the robot placed in a virtual environment of guava; due to the inertia properties of the links, the robot is unable to trace the trajectory. The simulations of the robot during the fruit-picking application in a virtual guava environment were conducted in Simscape Multibody Dynamics, and a screenshot is presented in Figure 13.

Table 6. Redundant manipulator inertial properties.

Link Name	Mass (kg)	I_{xx} (m^4)	I_{xy} (m^4)	I_{xz} (m^4)	I_{yy} (m^4)	I_{yz} (m^4)	I_{zz} (m^4)
L1	0.23905	1.18×10^{-5}	-3.50×10^{-21}	-8.67×10^{-22}	7.5063×10^{4}	-1.17×10^{-22}	7.5077×10^{4}
L2	0.23811	1.17×10^{-5}	-4.34×10^{-22}	-1.05×10^{-21}	7.404×10^{4}	-2.80×10^{-22}	7.4044×10^{4}
L3	0.23811	1.17×10^{-5}	-4.16×10^{-21}	-1.79×10^{-21}	7.4039×10^{4}	-3.42×10^{-22}	7.4045×10^{4}
L4	0.23811	1.17×10^{-5}	-3.94×10^{-21}	-1.64×10^{-21}	7.4039×10^{4}	-7.33×10^{-22}	7.4045×10^{4}
L5	0.23811	1.17×10^{-5}	3.17×10^{-21}	9.80×10^{-21}	7.404×10^{4}	-4.37×10^{-22}	7.4044×10^{4}
L6	0.21292	5.4842×10^{4}	-1.30×10^{-21}	-5.45×10^{-21}	1.02×10^{-5}	2.46×10^{-9}	5.4847×10^{4}
L7	7.2186×10^{3}	1.14×10^{-6}	1.15×10^{-12}	-6.51×10^{-13}	1.17×10^{-6}	9.75×10^{-12}	5.51×10^{-7}
L8	0.029379	2.84×10^{-6}	2.63×10^{-13}	3.37×10^{-15}	2.87×10^{-6}	4.95×10^{-15}	1.32×10^{-6}
L9	0.011124	1.16×10^{-6}	-4.07×10^{-23}	-2.82×10^{-22}	2.02×10^{-6}	$6.08\text{E}\times10^{-23}$	1.05×10^{-6}

During the trajectory tracking of the robot manipulator, the position and velocity error graphs are depicted in Figure 14 and Figure 15. From the position tracking in Figure 14, it is shown and observed that there is a large variation at each joint level trajectory. Therefore, the velocity tracking of each joint is experiencing sudden changes compared to the required velocities. To obtain the smooth motion of the robot manipulator, different controllers are modelled and tuned in the following sections.

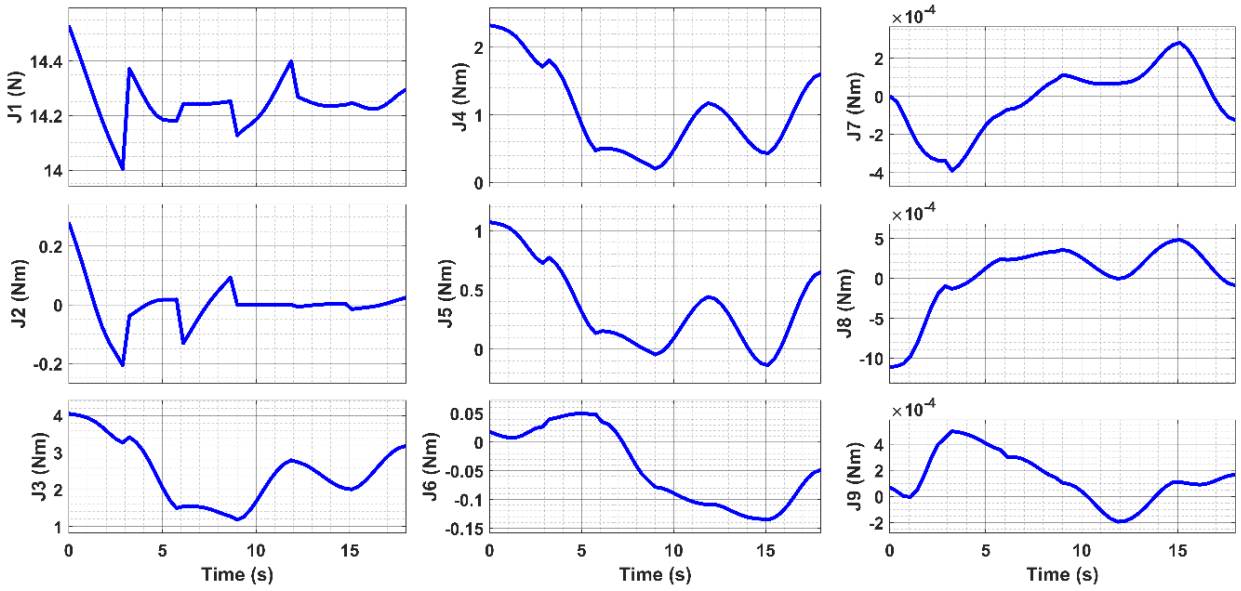


Figure 12. Torque at each joint obtained by N-E method.

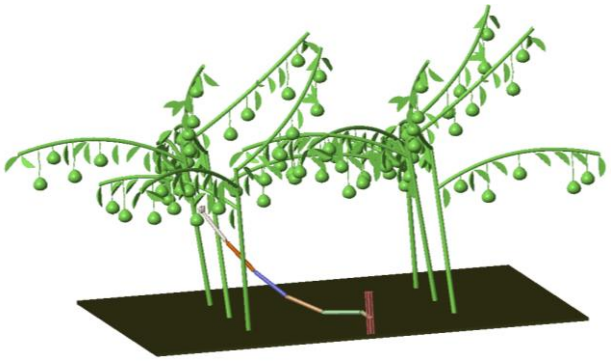


Figure 13. Robot manipulator in the virtual agricultural environment during simulations study in simscape multibody dynamics.

From the position and velocity error graphs, it is known that the trajectory tracking end-effector between the TSLs is difficult due to dynamic aspects. To minimize the trajectory tracking errors in position and velocity, several controllers were adopted in this study. The performance of the controllers is predicted by measuring the difference between the desired and measured values. The control parameters for adapted controllers are chosen after an effective tuning process. However, after the tuning, the controller parameters remain the same for different trajectory tracking contests. The results obtained from different combinations of PID controllers for trajectory tracking are discussed in this section. The computational time of the controllers is also reported for the selection of an effective controller. All simulations were carried out using Matlab/Simulink 2022a software.

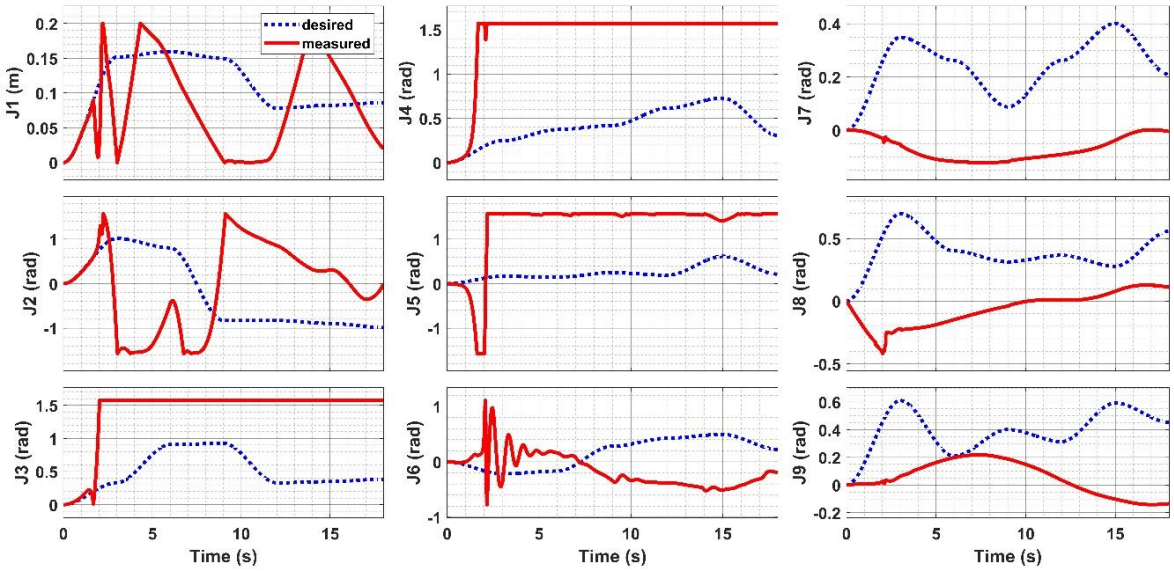


Figure 14. Position tracking without controller.

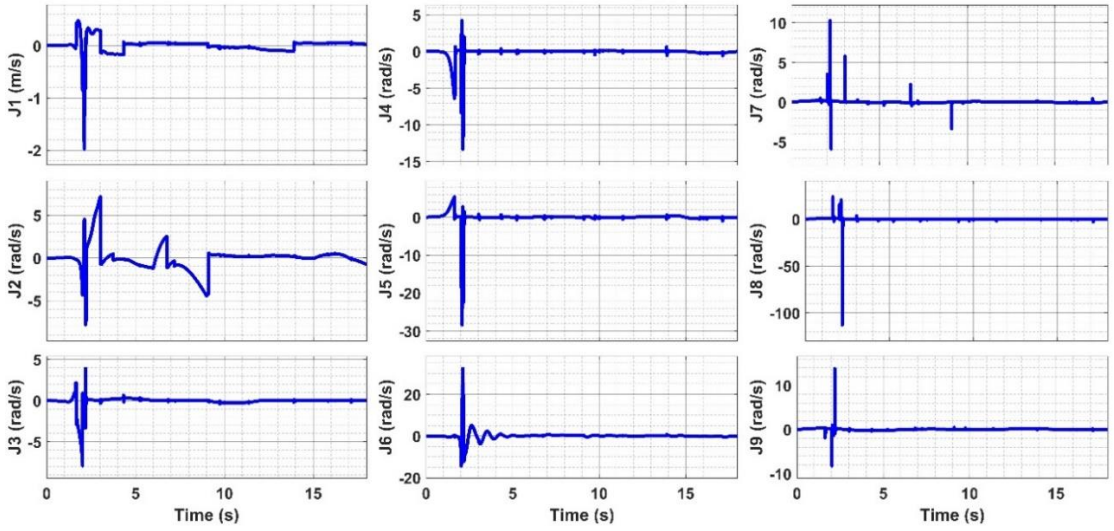


Figure 15. Velocity errors without controllers.

4.3 Controllers for Trajectory Tracking

4.3.1 PD and PID controllers

The PD and PID controllers are developed for the robot using Matlab/Simulink and are shown in Figure 16 and Figure 17. The robot joint variables are generated using the '*Polynomial trajectory generator*' and supplied to controller gains.

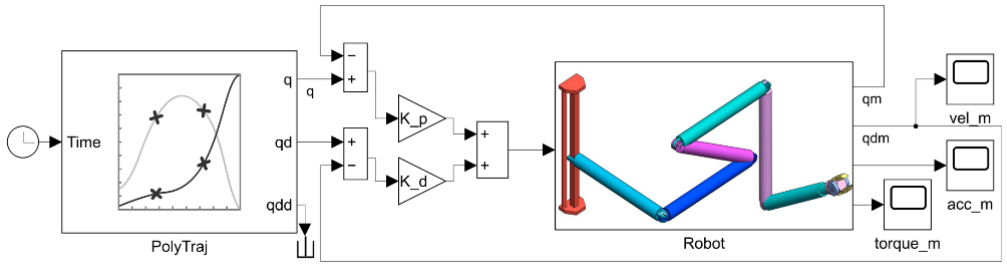


Figure 16. PD controller for robot manipulators.

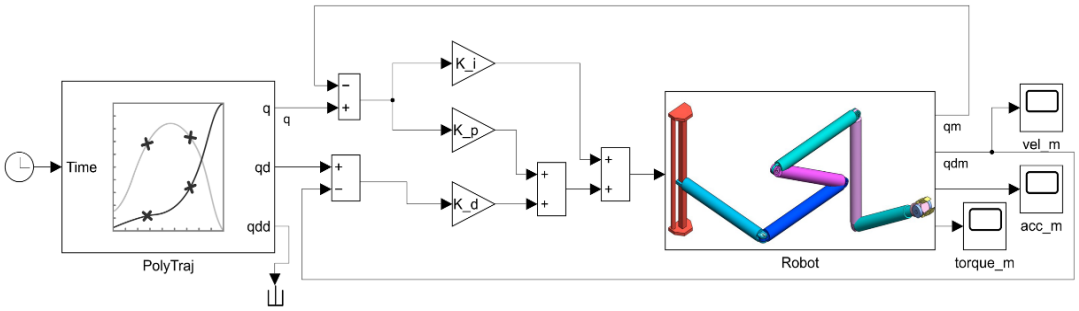


Figure 17. PID controller for robot manipulators.

The control parameters attained after tuning process K_p, K_d are 200, 40 were selected for PD controller. For the PID controller, the attained control parameters K_p, K_i and K_d are 300, 700, and 135 respectively. The controlled signal is fed back to robot model for trajectory generation. The position and velocity tracking errors after tuning with the PD controller are shown in Figure 18 and Figure 19 respectively.

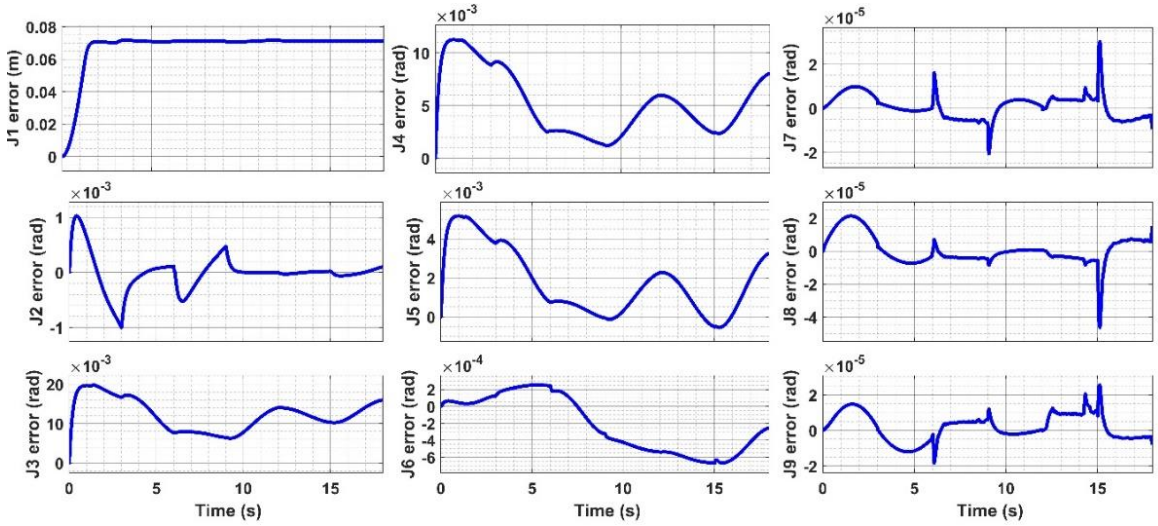


Figure 18. PD position control – error.

In Figure 18, the noted difference at joint 1 is nearly 7 cm, and for the remaining joints 2 to 9 the errors are in the order of 10^{-3} radians. The tuned PD controller during position tracking, results in maximum variation (error) for prismatic joint (*J*1) than errors in revolute joints (*J*2 to *J*9) from the observations. Similarly, for velocity tracking with the PD controller, the maximum error at the prismatic joint is 0.074 m/s, and for joints 2 to 9, the errors are in the order of 10^{-3} rad/sec.

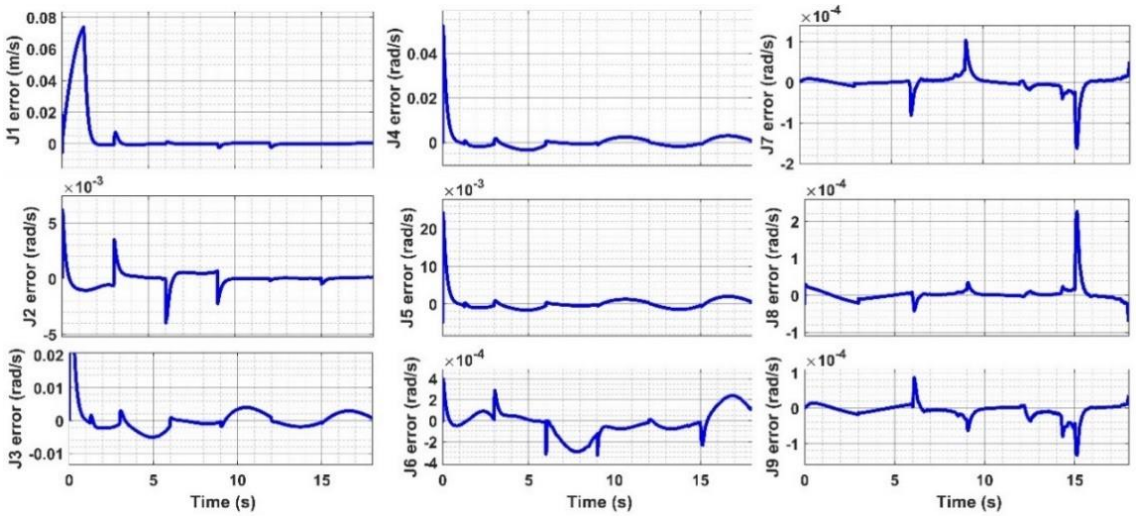


Figure 19. PD velocity control – error.

In the same way, the position and velocity tracking errors for the PID controller are shown in Figure 20 and Figure 21, respectively. The position tracking error for joint 1 is 1.4 cm, and for joints 2 to 9, the errors are a factor of 10^{-3}. Velocity tracking errors with the PID controller are 3.8 cm/s for joint 1, and 0.037 rad/sec is the maximum variation that occurs at joint 3. The position and velocity trajectories errors for the PID controller are less when compared to PD controller. The time consumed for tracking the trajectory with a PD controller is 16.7 sec, whereas with a PID controller it is 24.2 sec. The time taken by the PID controller was 45.2% higher than that of the PD controller.

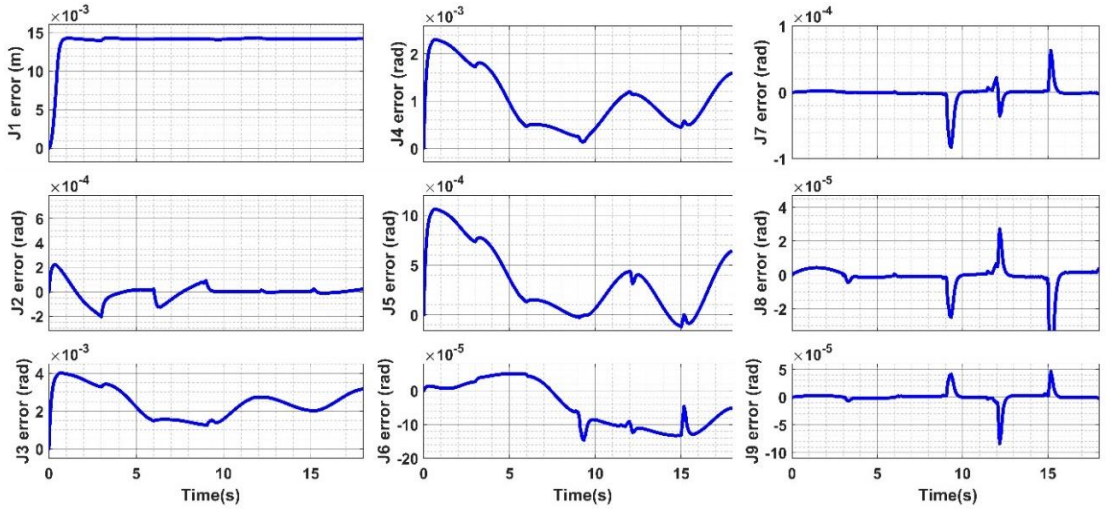


Figure 20. PID position control – error.

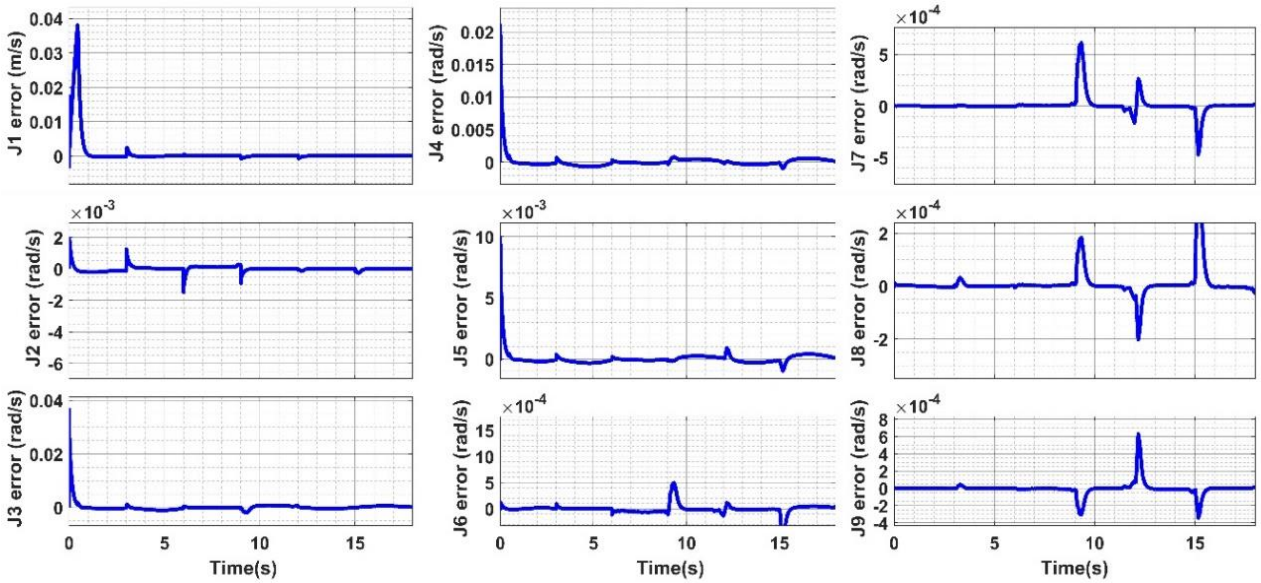


Figure 21. PID velocity control – error.

4.3.2 Feed-Forward Controller

The Feed-Forward (FF) controller is quite similar to the PD controller with the Inverse Dynamics (ID) block and is presented in Figure 22. In the FF controller, the control parameters K_p, K_d are chosen as 10 and 2, respectively. Figure 23 shows the error in joint 1 is 0.03 cm, and the maximum error is 0.0005 radians at the remaining joints. During velocity tracking from Figure 24, at joint 1, the error is noted as 0.3 cm/s, and 0.0028 rad/s is the maximum variation in revolute joints.

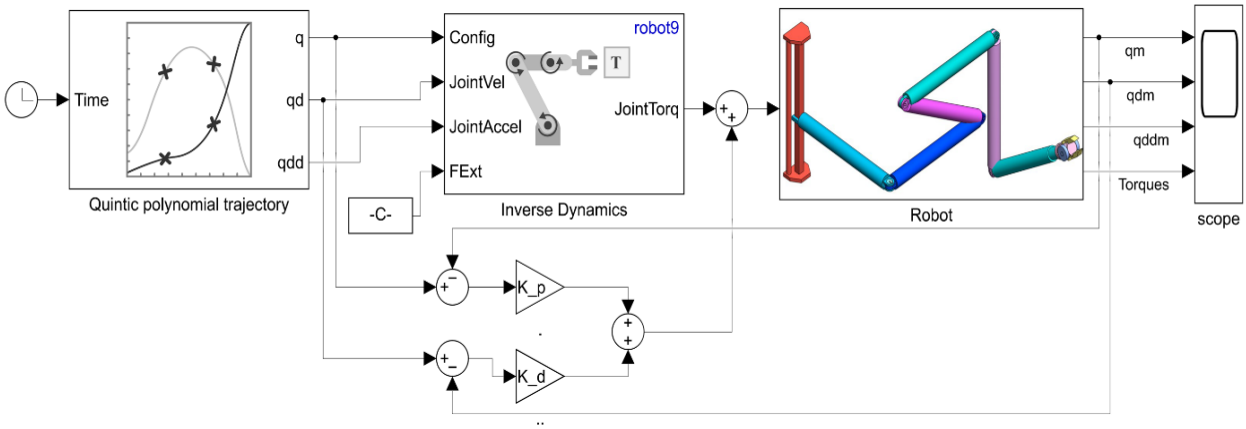


Figure 22. Feed-Forward controller for robot manipulator.

The FF controller is working effectively for tracking both position and velocity as compared to PD and PID controllers. The controller uses 15.15% more time than the PID controller.

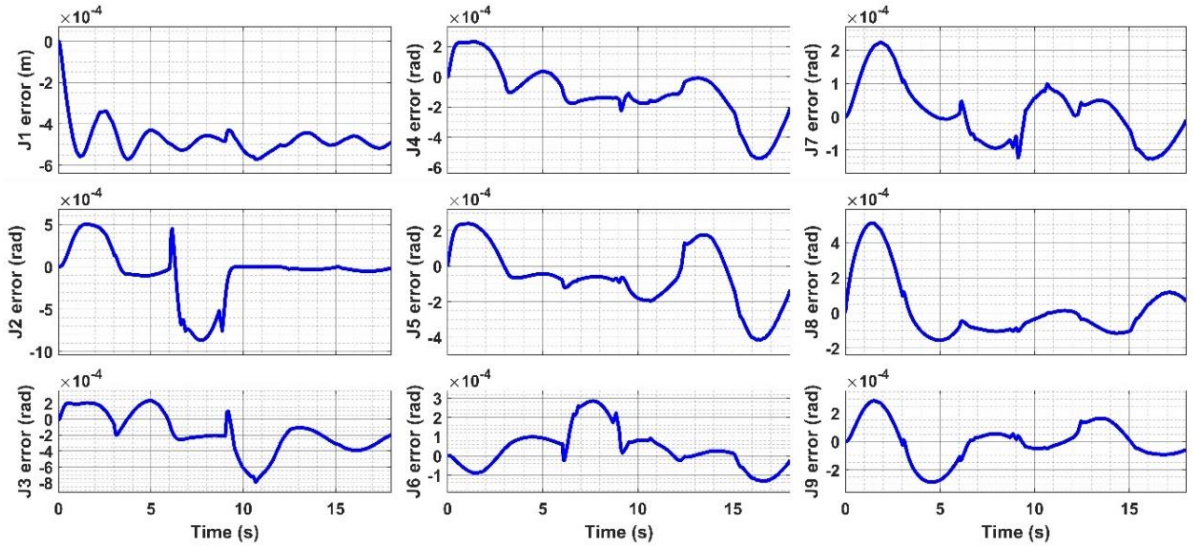


Figure 23. Feed-forward controller position error.

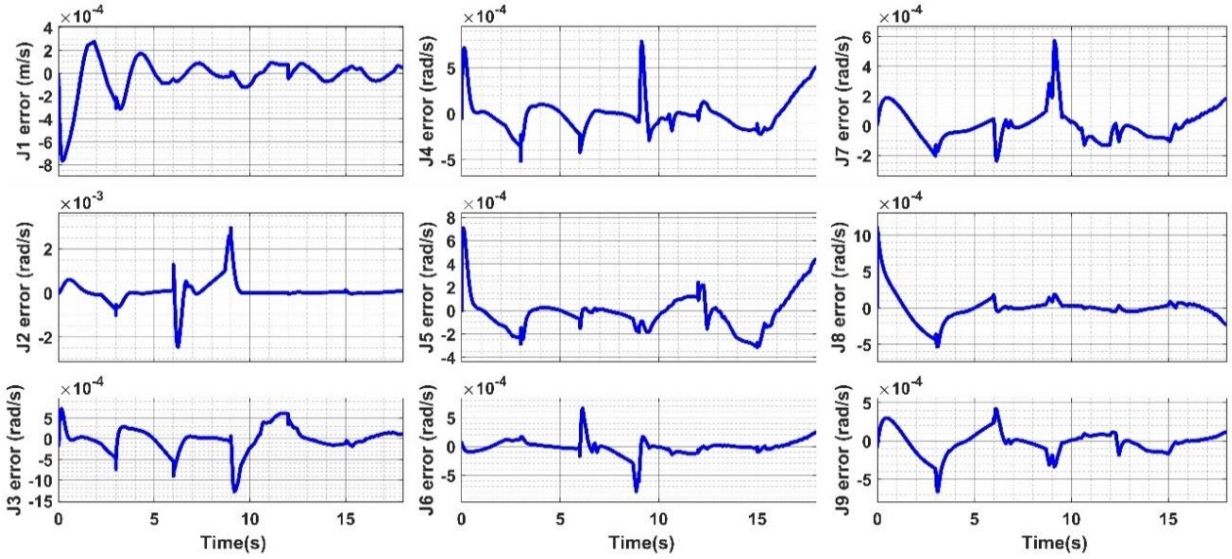


Figure 24. Feed-forward controller velocity error.

4.3.3 PD-CTC and PID-CTC Controllers

CTC, in combination with PD and PID, is chosen in this work for trajectory tracking**.** The controllers PD-CTC and PID-CTC with control parameters K_p, K_d are 150, 25 and K_p, K_i, K_d are 150, 375, and 25 respectively, and the Simulink diagram of the PD-CTC controller is shown in Figure 25. The detailed view of the CTC block in the Simulink environment is shown in Figure 26.

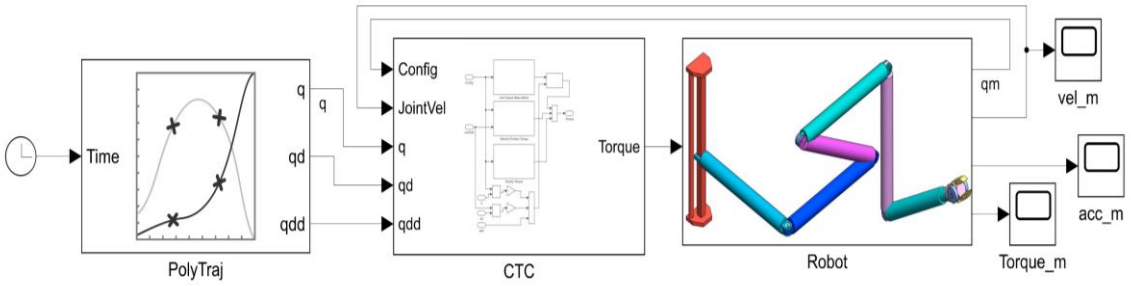


Figure 25. PD-computed torque control.

Figure 27 shows the errors in the PD-CTC controller. At joint 1, the variation of position is 0.036 cm, and 0.57 radians maximum variation occurs in the remaining joints. From Figure 28, during velocity tracking, there was a 0.03 cm/s error in joint 1 and a 0.012 rad/s maximum variation in other joints.

PID-CTC controller errors are shown in Figure 29 and Figure 30. At joint 1, the position variation is 0.016 cm, and the maximum variation of 2.02 radians occurs at the remaining joints. During velocity tracking, there was a 0.03 cm/s error at joint 1 and a 0.59 rad/s maximum variation at wrist joints. The chosen controllers are effectively tracking for the first six joints but are unable to execute for the last three joints, i.e., at the wrist location. The target position is attained using these controllers, and it is difficult to achieve orientation. PID-CTC is taking 51.62 seconds to track the trajectory; similarly, PD-CTC is taking 50.81 seconds. PD-CTC has taken 1.6% less time compared to PID-CTC to track the trajectory. Although PD-CTC has less time, effective tracking results are obtained from the PID-CTC controller.

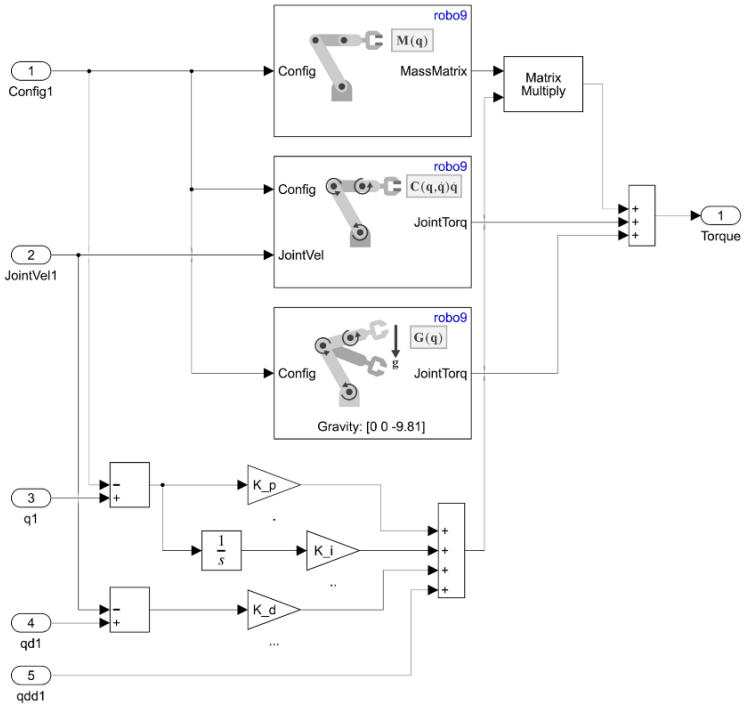


Figure 26. Block diagram of computed torque control.

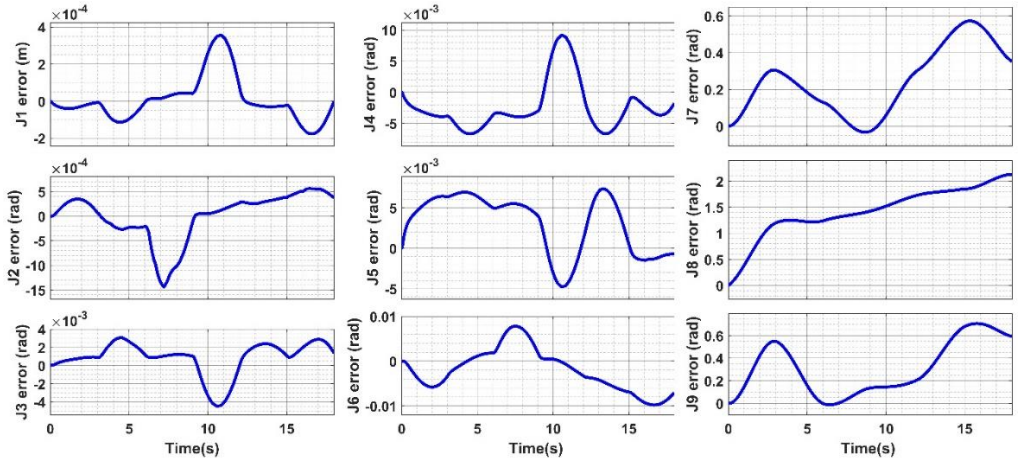


Figure 27. PD-CTC position error.

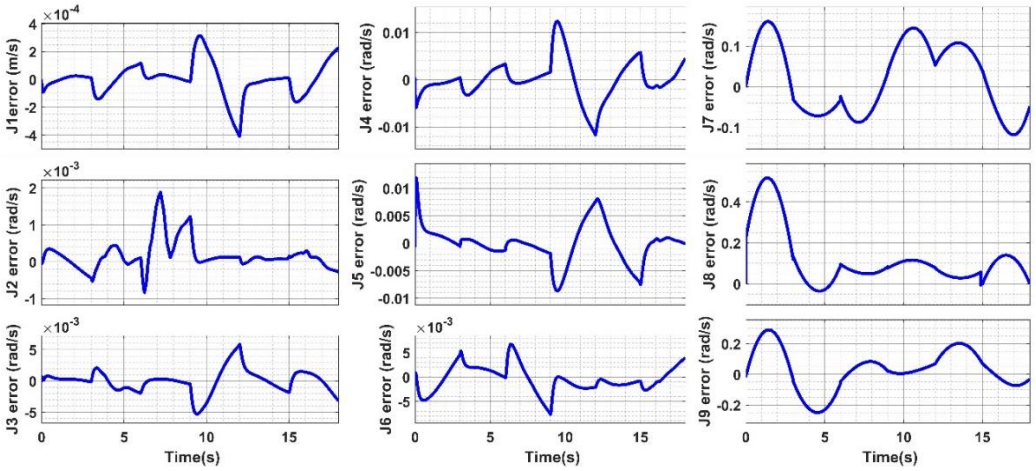


Figure 28. PD-CTC velocity error.

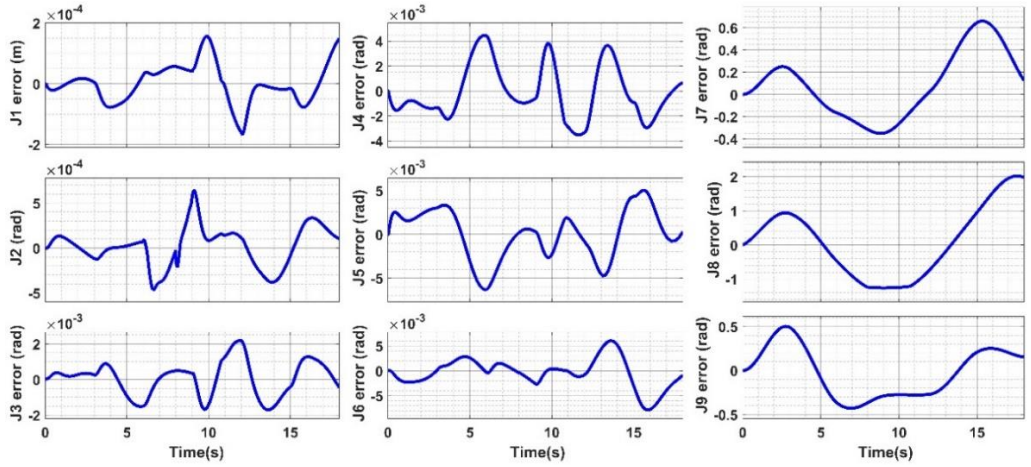


Figure 29. PID-CTC position error.

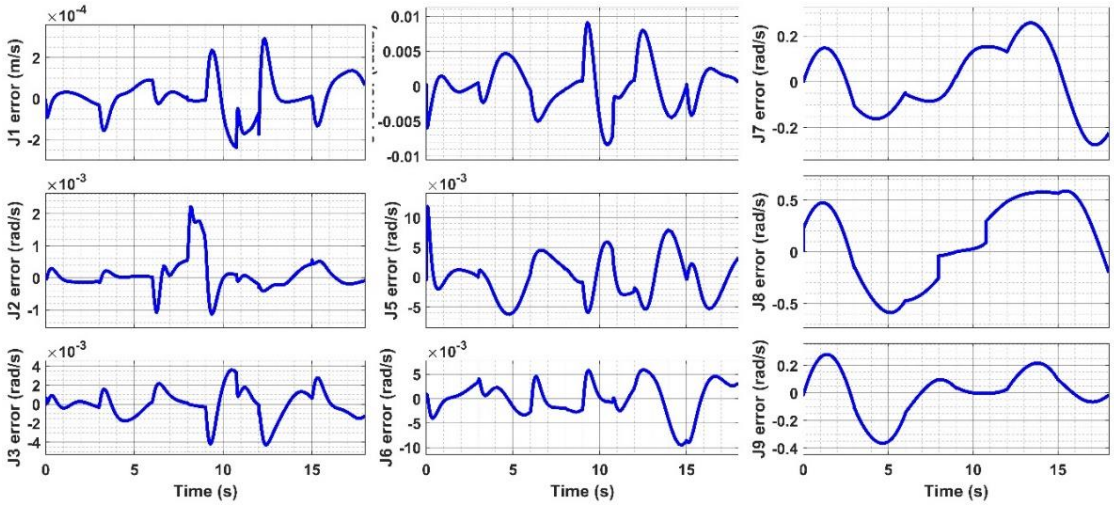


Figure 30. PID-CTC velocity error.

4.3.4 Feed-Forward-2DOF PID Controller

The FF-2DOF PID controller shown in Figure 31 is also implemented for the robot to track the trajectory. From Figure 32, it is observed that the maximum variation is 15 cm at the prismatic joint and 0.043 radians at the revolute joint. Similarly, during velocity tracking, there was a 1.5 m/s difference at joint 1 and a 0.03 rad/sec variation for other revolute joints, as observed in Figure 33. From the obtained results, the chosen controller is unable to control the joints and takes more time to track the trajectory compared to other controllers.

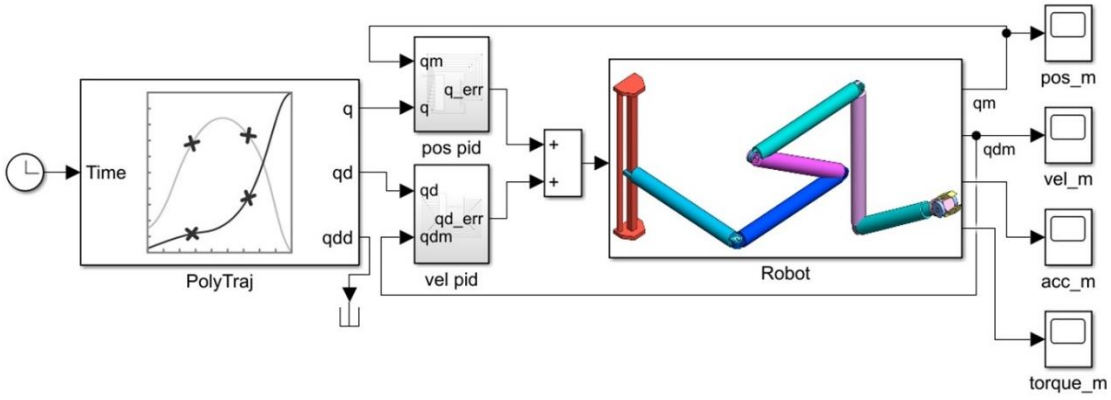


Figure 31. Feed-forward 2-DOF PID controller.

The gains K_p, K_i, and K_d of the controllers PD, PID, Feed-Forward, PD-CTC, PID-CTC, and FF-2DOF PID are tabulated in Table 7. The position and velocity absolute errors that occurred at each joint for different controllers are tabulated in Table 8, and the time taken by each controller to generate control signals is also observed and listed in this table. The minimum errors obtained from the controllers used in this paper are represented in bold letters in Table 8. The maximum absolute positional error by the PID-CTC controller is 0.2 mm, which is low when compared to the positional error of the serial manipulator used in Zhang et al. (2020). The consolidated ranking of the controllers is given in Table 9, along with the results obtained from Table 8.

From Table 9, the time taken by the PD controller is the minimum for controlling the robot manipulator. During the trajectory tracking, the positional and velocity errors are minimum with the PID-CTC controller for the prismatic joint, whereas the FF controller has minimum errors for controlling the revolute joint.

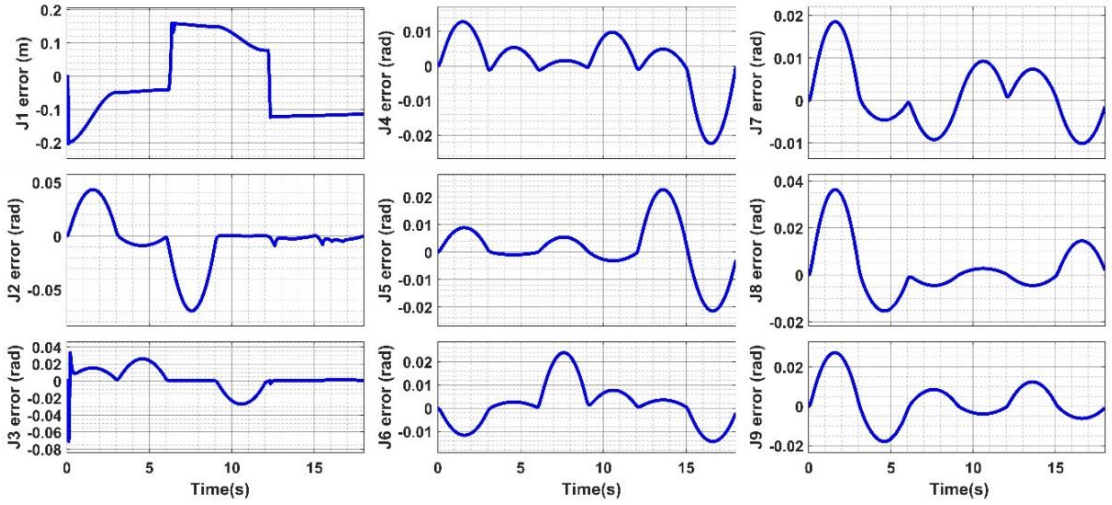


Figure 32. Feed-forward 2-DOF PID controller position error.

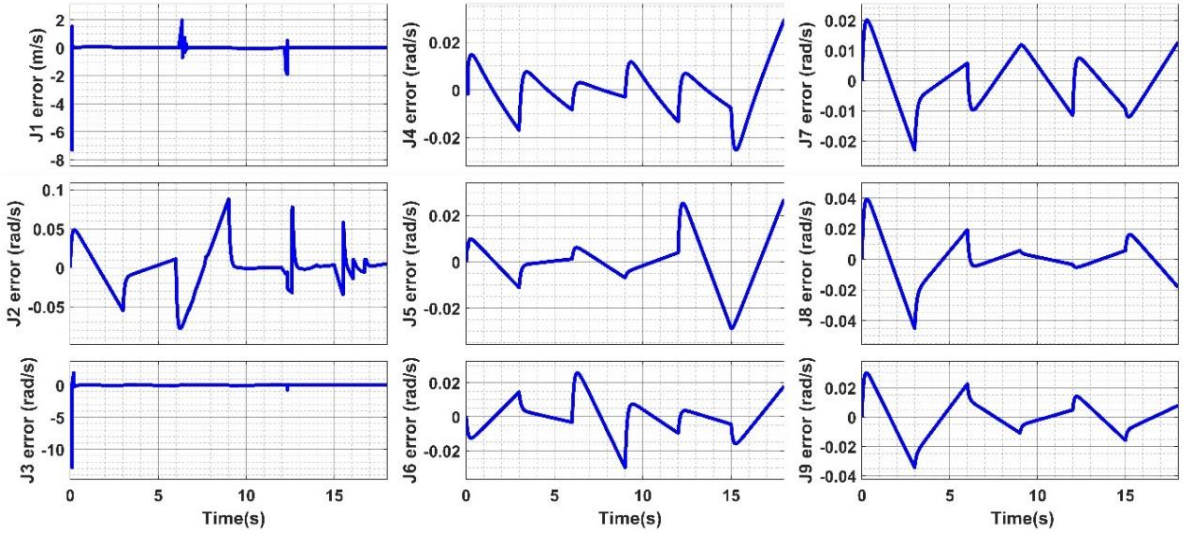


Figure 33. Feed-Forward 2-DOF PID controller velocity error.

Table 7. Controller parameters.

S. No.	Controller	K_p	K_i	K_d
1	PD	200	NA	40
2	PID	300	700	135
3	Feed-Forward control	10	NA	2
4	PD-CTC	150	NA	25
5	PID-CTC	150	375	25

Table 8. A Joint space trajectory tracking errors.

S. No.	Controller	Running time (sec)	Joint	POSITION		VELOCITY	
				Min error	Max error	Min error	Max error
1	PD	16.6513	Prismatic	0.07007	0.07193	0.005621	0.07375
			Revolute	0.0005421	0.01975	0.005099	0.09051
2	PID	24.183	Prismatic	0.01402	0.01438	0.005991	0.03805
			Revolute	0.0007415	0.004013	0.001704	0.003662
3	Feed-Forward control	27.847	Prismatic	0.0003384	0.0005716	0.0002753	0.0007644
			Revolute	0.0005048	**0.0007931**	0.002431	**0.002955**
4	PD-CTC	50.8087	Prismatic	0.0001766	0.0003566	0.0003143	0.0004094
			Revolute	0.03316	0.5735	0.01162	0.01234
5	PID-CTC	51.6213	Prismatic	0.0001555	**0.0001657**	0.0002377	**0.000292**
			Revolute	0.3508	2.022	0.5863	0.5876
6	FF-2DOF PID	797.754	Prismatic	0.1581	0.1985	1.53	7.185
			Revolute	0.04327	0.06969	0.02918	0.1282

Table 9. Ranking of controllers.

Rank	Time	PRISMATIC		REVOLUTE	
		POS	VEL	POS	VEL
1	PD	PID-CTC	PID-CTC	FF	FF
2	PID	PD-CTC	PD-CTC	PID	PID
3	FF	FF	FF	PD	PID-CTC
4	PD-CTC	PID	PID	FF-2-DOF PID	PD
5	PID-CTC	PD	PD	PD-CTC	FF-2-DOF PID
6	FF-2-DOF PID	FF-2-DOF PID	FF-2-DOF PID	PID-CTC	PID-CTC

5. Conclusions

In this work, dynamic analysis of a 9-DOF serial redundant robot manipulator working in a virtual agricultural environment has been carried out. The joint variables required to reach TSLs are determined using the classical optimization technique. During the motion planning of the robot, collisions with the surrounding environment are avoided by using the bounding box technique. Cubic polynomial trajectories are generated for each joint of the manipulator, and required torques are computed using the Recursive Newton-Euler method. For effective trajectory tracking of the manipulator, variations in torque are controlled by different combinations of PD and PID controllers, which are designed in the Simulink environment. A simulation study is presented for trajectory tracking of the robot manipulator, and the errors in position and velocities are presented and compared among the controllers. From the results, it is observed that for the prismatic joint, the PID-CTC controller is giving a better response with the least tracking errors at positional and velocity levels. In a similar way, for smooth motion in the revolute joints, the Feed-Forward controller has minimal errors compared to the remaining controllers used in this article. Although the computational time of the PD controller is less, it is performing poorly when compared to other controllers. From the results, it is concluded that in an agricultural environment that is less dense, the robot can choose a controller that has less processing time to obtain high productivity by compensating the

trajectory errors, and if the robot is working in highly dense environments, it will choose a controller that has the least trajectory errors. This work can be further continued by investigating the validity of a combination of meta-heuristic algorithms with PID controllers for agricultural environments.

Conflict of Interest

The author declares that they have no conflict of interest.

Acknowledgements

The authors would like to express sincere thanks to the chief editor, guest editors, and anonymous reviewers for their valuable suggestions and comments to improve the quality of this work.

References

Arima, S., & Kondo, N. (1999). Cucumber harvesting robot and plant training system. *Journal of Robotics and Mechatronics*, *11*(3), 208-212.

Bac, C.W., Van Henten, E.J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*(6), 888-911. https://doi.org/10.1002/rob.21525.

Bac, C.W., Roorda, T., Reshef, R., Berman, S., Hemming, J., & van Henten, E.J. (2016). Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment. *Biosystems Engineering*, *146*, 85-97.

Baeten, J., Donné, K., Boedrij, S., Beckers, W., & Claesen, E. (2008). Autonomous fruit picking machine: A robotic apple harvester. In: Laugier, C., Siegwart, R. (eds) *Field and Service Robotics. Springer Tracts in Advanced Robotics* (Vol. 42, pp. 531-539). Springer, Berlin, Heidelberg. https://hal.inria.fr/inria-00194739.

Bak, T., & Jakobsen, H. (2004). Agricultural robotic platform with four wheel steering for weed detection. *Biosystems Engineering*, *87*(2), 125-136. https://doi.org/10.1016/j.biosystemseng.2003.10.009.

Baur, J., Pfaff, J., Ulbrich, H., & Villgrattner, T. (2012). Design and development of a redundant modular multipurpose agricultural manipulator. In *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (pp. 823-830). IEEE. Kaohsiung, Taiwan.

Belforte, G., Deboli, R., Gay, P., Piccarolo, P., & Aimonino, D.R. (2006). Robot design and testing for greenhouse applications. *Biosystems Engineering*, *95*(3), 309-321.

Boggs, P.T., & Tolle, J.W. (1995). Sequential quadratic programming. *Acta Numerica*, *4*, 1-51. https://doi.org/10.1017/S0962492900002518.

Cao, X., Zou, X., Jia, C., Chen, M., & Zeng, Z. (2019). RRT-based path planning for an intelligent litchi-picking manipulator. *Computers and Electronics in Agriculture*, *156*, 105-118.

Chapelle, F., & Bidaud, P. (2004). Closed form solutions for inverse kinematics approximation of general 6R manipulators. *Mechanism and Machine Theory*, *39*(3), 323-338.

Chembuly, V.S., & Voruganti, H.K. (2020). An efficient approach for inverse kinematics and redundancy resolution of spatial redundant robots for cluttered environment. *SN Applied Sciences*, *2*, 1-20.

Craig, J.J. (2006). *Introduction to robotics*. Pearson Education.

Dhyani, A., Panda, M.K., & Jha, B. (2020). Design of an evolving fuzzy-PID controller for optimal trajectory control of a 7-DOF redundant manipulator with prioritized sub-tasks. *Expert Systems with Applications*, *162*, 113021.

Edan, Y., Han, S., Kondo, N. (2009). Automation in Agriculture. In: Nof, S. (ed) *Springer Handbook of Automation* (pp. 1095-1128). Springer Handbooks. Springer, Berlin, Heidelberg.

Grotjahn, M., & Heimann, B. (2002). Model-based feedforward control in industrial robotics. *The International Journal of Robotics Research*, *21*(1), 45-60. https://doi.org/10.1177/027836402320556476.

Han, S., Xueyan, S., Tiezhong, Z., Bin, Z., & Liming, X. (2007). Design optimisation and simulation of structure parameters of an eggplant picking robot. *New Zealand Journal of Agricultural Research*, *50*(5), 959-964.

Hsu, P., Mauser, J., & Sastry, S. (1989). Dynamic control of redundant manipulators. *Journal of Robotic Systems*, *6*(2), 133-148. https://doi.org/10.1002/rob.4620060203.

Iossifidis, I., & Schoner, G. (2006). Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for an redundant robot arm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 580-585). IEEE. Beijing, China.

Kondo, N., Monta, M., Shibano, Y., & Mohri, K. (1993). Basic mechanism of robot adapted to physical properties of tomato plant. In *Proceedings of the Korean Society for Agricultural Machinery Conference* (pp. 840-849). Korean Society for Agricultural Machinery.

Kumar, A., & Kumar, V. (2017). Evolving an interval type-2 fuzzy PID controller for the redundant robotic manipulator. *Expert Systems with Applications*, *73*, 161-177. https://doi.org/10.1016/j.eswa.2016.12.029.

LaValle, S.M., & Kuffner, J.J. (2001). Rapidly-exploring random trees: Progress and prospects. In: Donald, B., Lynch, K., Rus, D. (eds) *Algorithmic and Computational Robotics* (pp.303-307). New York.

Lee, C.G. (1982). Robot arm kinematics, dynamics, and control. *Computer*, *15*(12), 62-80.

Li, P., Lee, S.H., & Hsu, H.Y. (2011). Review on fruit harvesting method for potential use of automatic fruit harvesting systems. *Procedia Engineering*, *23*, 351-366. https://doi.org/10.1016/j.proeng.2011.11.2514.

Monta, M., Kondo, N., & Shibano, Y. (1995). Agricultural robot in grape production system. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation* (Vol. 3, pp. 2504-2509). IEEE. Nagoya, Japan.

Nearchou, A.C. (1998). Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm. *Mechanism and Machine Theory*, *33*(3), 273-292.

Pons, J.L., Ceres, R., & Jimenez, A.R. (1996). Mechanical design of a fruit picking manipulator: Improvement of dynamic behavior. In *Proceedings of IEEE International Conference on Robotics and Automation* (Vol. 1, pp. 969-974). IEEE. Minneapolis, MN, USA.

Park, S.O., Lee, M.C., & Kim, J. (2020). Trajectory planning with collision avoidance for redundant robots using jacobian and artificial potential field-based real-time inverse kinematics. *International Journal of Control, Automation and Systems*, *18*(8), 2095-2107.

Parker, J.K., Khoogar, A.R., & Goldberg, D.E. (1989). Inverse kinematics of redundant robots using genetic algorithms. In *1989 IEEE International Conference on Robotics and Automation* (pp. 271-272). IEEE Computer Society. Scottsdale, AZ, USA. DOI: 10.1109/ROBOT.1989.100000.

Saha, S.K. (2014). *Introduction to robotics*. Tata McGraw-Hill Education.

Sakai, S., Iida, M., Osuka, K., & Umeda, M. (2008). Design and control of a heavy material handling manipulator for agricultural robots. *Autonomous Robots*, *25*, 189-204. https://doi.org/10.1007/s10514-008-9090-y.

Silwal, A., Davidson, J.R., Karkee, M., Mo, C., Zhang, Q., & Lewis, K. (2017). Design, integration, and field evaluation of a robotic apple harvester. *Journal of Field Robotics*, *34*(6), 1140-1159.

Reddy, A.S., Chembuly, V.V.M.J.S., & Rao, V.V.S.K. (2022). Collision-free inverse kinematics of redundant manipulator for agricultural applications through optimization techniques. *International Journal of Engineering*, *35*(7), 1343-1354.

Tatlicioglu, E., Braganza, D., Burg, T.C., & Dawson, D.M. (2009). Adaptive control of redundant robot manipulators with sub-task objectives. *Robotica*, *27*(6), 873-881. https://doi.org/10.1017/S0263574708005274.

Tillett, N.D. (1993). Robotic manipulators in horticulture: A review. *Journal of Agricultural Engineering Research*, *55*(2), 89-105. https://doi.org/10.1006/jaer.1993.1035.

Urrea, C., & Kern, J. (2012). Modeling, simulation and control of a redundant SCARA-type manipulator robot. *International Journal of Advanced Robotic Systems*, *9*(2), 58.

Van Henten, E.J., Schenk, E.J., Van Willigenburg, L.G., Meuleman, J., & Barreiro, P. (2010). Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot. *Biosystems Engineering*, *106*(2), 112-124. https://doi.org/10.1016/j.biosystemseng.2010.01.007.

Yahya, S., Moghavvemi, M., & Mohamed, H.A. (2012). Singularity avoidance of a six degree of freedom three dimensional redundant planar manipulator. *Computers & Mathematics with Applications*, *64*(5), 856-868.

Zhang, Y., Kim, D., Zhao, Y., & Lee, J. (2020). PD control of a manipulator with gravity and inertia compensation using an RBF neural network. *International Journal of Control, Automation and Systems*, *18*(12), 3083-3092.