Journal of Robotics and Control (JRC)
Volume 4, Issue 1, January 2023
ISSN: 2715-5072, DOI: 10.18196/jrc.v4i1.17121

Modelling and Simulation of a Redundant Agricultural Manipulator with Virtual Prototyping

A. Sridhar Reddy [1*], V. V. M. J. Satish Chembuly [2], V. V. S. Kesava Rao [3]
[1,3] Department of Mechanical Engineering, AU College of Engineering, Andhra University, Visakhapatnam, India
[2] Department of Mechanical Engineering, Aditya College of Engineering & Technology, Surampalem, India
Email: [1] asridhareddy@andhrauniveersity.edu.in, [2] satishchv@gmail.com, [3] prof.vvskrao@andhrauniveersity.edu.in
*Corresponding Author

***Abstract*—The development of autonomous robots for agricultural applications includes motion planning, fruit picking, and collision avoidance with surrounding environments, and these become challenging tasks. For harvesting applications, robust control of the manipulator is needed for the effective motion of the robot. Several combinations of Proportional(P)- Integrative(I)-Derivative(D) controllers are modelled and a simulation study was performed for trajectory tracking of a redundant manipulator in virtual agricultural environments. The article presents a comprehensive study on kinematic modelling and dynamic control of redundant manipulator for fruit-picking applications in virtual environments. The collisions with surrounding environment were eliminated using 'bounding box technique'. The joint variables are obtained by constructing Inverse Kinematics (IK) problem and are determined using a classical optimization technique. Different controllers are modelled in the 'Simulink' environment and are tuned to generate error-free trajectory tracking during harvesting. The task space locations (TSLs) are considered as via-points, and joint variables at each TSLs are obtained by Sequential Quadratic Programming (SQP) technique. Joint-level trajectories are generated using Quintic and B-spline polynomials. For effective trajectory tracking, torque variations are controlled using the PID and Feedforward (FF) controller. The dynamic simulations of the robot manipulator are performed in Simscape Multibody software. Results show that the during the trajectory tracking of the manipulator, the Feed-forward controller performs best with Quintic polynomial trajectory.**

***Keywords*—Redundant manipulator; PID & Feed-Forward controller; Matlab/Simulink; Simscape Multibody.**

I. Introduction

Extensive research has been carrying on the application of robots in the agricultural field [1] [2]. Working of a robot in an agricultural environment is a complex task due to the variety of plants and operations, and the reasons behind the low adaption of commercial agricultural robots in the agricultural market are discussed in [3]. Despite recent advancements in agricultural automation, high-value crops such as fruit and vegetable crops are still dependent on labour. To overcome the shortage of labour in the agricultural field, the robot should be able to perform operations like crop seeding [4], crop weeding, selective spraying, thinning and pruning, fruit picking etc. For precision spraying and fertilization in greenhouse environments, Belforte *et al.* [5] designed and developed a robot. This robot is capable of performing agricultural operations for indefinite time periods. Song Han *et al.* [6] designed a manipulator for vegetable picking in greenhouse environments. Baeten *et al.* [7] utilized a 6 DOF industrial robot manipulator for fruit-picking applications with the addition of a newly designed suction gripper. Similarly, a 7 DOF robot manipulator was designed for grasping apples by Abhisesh Silwal *et al.* [8]. In agricultural robots, Object recognition capability, operational stability, trajectory errors, low operational speeds, high operational costs, and a wide variety of plant features are some of the factors preventing the commercialization of harvesting robots as pointed out by Peilin li *et al.* [9].

The configuration of robot manipulator for agricultural applications is chosen based on the plant features and available workspace. Kondo *et al.* [10] designed a 7 DOF robot manipulator based on operational space and manipulability. Seiichi Arima *et al.* [11] designed a 7-DOF robot manipulator for harvesting cucumbers by considering the characteristics of cucumbers. Baur *et al.* [12] developed a 9 DOF redundant robot manipulator for accessing the plant in all directions to pick the fruits.

To perform agricultural operations, It is required to identify the position of the harvesting target in three-dimensional space and the end-effector of the robot is required to reach these harvesting targets i.e., Task space locations (TSLs) [13], [14], [15]. The development of the kinematic model analysis of a robot is essential for motion simulation. A possible set of joint variables are determined through Inverse Kinematics (IK) for the manipulator, to reach the TSL with desired orientation [16]. These variables can be determined either analytically or numerically [17]. In an analytical method, the solution can be obtained as a closed form, and these solutions are inherently dependent on the DOF of the robot manipulator. In the numerical approach, these are obtained by iterative methods and this method is computationally effective even with an increase in DOF. Aristidou and Lasenby [18] reviewed Jacobian

Inverse methods, Sequential Monte Carlo methods, Heuristic IK algorithms, and Newton Methods for solving IK problems numerically. The IK problem is also solved by optimization techniques by many researchers. A constrained optimization technique-based algorithm is utilized for redundant manipulators in [19]. Kumar *et al.* [20] solved IK as a nonlinear optimization problem for trajectory tracking of end-effector. Bagheri *et al.* [21] utilized neural networks to determine the IK solutions for 6 DOF robot. Satish and Hari [22] used an optimization-based approach to determine IK solutions for a redundant manipulator in a cluttered environment with constraints of obstacle and singularities avoidance. This algorithm also eliminates the use of the inverse of Jacobian to increase the computation effectiveness [22]. Rokbani and Alimi solved IK using particle swarm optimization [23], Starke *et al.* [24] used evolutionary optimization approach with Genetic Algorithm to determine the IK.

In the actual agricultural field, tree branches are the obstacles to robot motion planning which will increase the complexity of robot motion. Van Henten *et al.* [25] enveloped the robot with 'bounding sphere' to detect the collisions with surrounding objects. Puiu1 and Moldoveanu [26] detected collisions using 'oriented bounding boxes'. By considering the distance between the convex hull of dynamic obstacle and the manipulator, collisions are avoided in [27].

The collision-free path for robot manipulator is generated using motion planning algorithms such as Rapidly exploring Random Tree (RRT), Bi-RRT [28], and Potential field method [29] [30] [31]. By extending the target directional nodes, exploring speed is improved by Wei and Ren [32]. Trapping at local minima during the motion planning can be overwhelmed with regression mechanism as proposed by Zhang *et al.* [33]. Zang *et al.* [34] pointed out the need of integration of path planning algorithms with motion controllers along with collision avoidance to access the fruits located in the canopies. He Wang *et al.* [35] generated collision free cubic polynomial trajectories for robot manipulator. Motion planning of a redundant manipulator in obstacle hidden environment was presented in [36].

To derive the equations of motion for manipulators, several methods are available, the 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 [37]. 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 a force balance approach in dynamic analysis. The Outward recursion propagates kinematics 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 E-L method [34]. To track workspace trajectory for a redundant serial manipulator, Ping *et al.* [38] developed a control law based on the pseudo-inverse of Jacobian in the general form of a manipulator dynamic equation.

The external disturbances and sensor errors will result in uncertainties in the robot model [39]. Among robots, manipulators are highly nonlinear-coupled systems during operations [40] [41]. To accomplish the required tasks, the development of a mathematical model and control laws are required. By using proper programming tools, simulations are carried out for different tests i.e., position tracking, velocity control etc. [42]. To obtain high productivity and efficiency in harvesting applications, the design and control of robot manipulators is one of the main research areas. In general, manipulators are complex in structure and highly non-linear due to link couplings and friction. Therefore, designing and/or selecting proper controllers for effective trajectory tracking of robot manipulators is a challenging task. Various combinations of Proportional (P), Integrate (I) and Derivative (D) terms are utilized for controlling the robot manipulators to compensate for external disturbances [43]. Generally, PID controllers are linear in nature and are less robust to uncertainties presented in the model. Rakesh *et al.* [44] presented a review of PID controllers and their applications in robot manipulators, mechanical systems and biomedical fields. A. Dhyani *et al.* [45] linearized the non-linear behaviour of manipulator dynamics for trajectory tracking of redundant manipulators. Ajwad *et al.* [46] reiviewed current approaches for control of multi-DOF robot manipulator. A model-based Feed-Forward (FF) is discussed to improve the robot path accuracy in [47]. PID and PID with (FF) controllers are implemented in [48] for the parallel manipulator. PID plus FF controller is improved motor response in this manipulator. The estimated torques for the preferred joint motions of the manipulators are calculated using the Inverse dynamic model of the system. Kaixiang *et al.* [34] controlled the non-linear motion of a 3 DOF harvesting manipulator using a PI controller. Enver Tatlicioglu *et al.* [49] proposed an adaptive controller using Lyapunov-based stability analysis for a redundant manipulator to adjust the parametric uncertainties in the dynamic model. Abhilash Silwal [8] developed a 7-DOF robot for harvesting fruits in a purely open-loop manner (without sensors) with feedforward control. Halit and Oguz [50] Modelled a 5-legged robot with a serial manipulator in Simscape multibody, and the position of the manipulator is controlled with a PID controller.

To accelerate the design process and reduce the cost of the

testing process, modelling and simulation of robots in a virtual environment are very much needed. The process of designing and inspecting the robots in the real world is simplified by executing the simulations using different softwares as listed in [51], [52], [53]. From these simulations, the performance of the robot can be estimated and it is an essential role in the evaluation of the automation process.

This article presents a comprehensive study of the agricultural robot working in a virtual environment. A redundant manipulator of 9 DOF is modelled in Solidworks, Inverse Kinematics are determined to reach task space locations (TSLs) using a classical optimization technique. Collisions are avoided using 'bounding box' technique. Dynamic simulations are performed and are controlled using PID and Feed-Forward for smooth trajectory tracking and the methodology followed in the present article are presented in Fig. 1.

The research contribution is the kinematic modelling and obstacle avoidance of the robot manipulator in agricultural environments for harvesting applications. In addition to that the second contribution is achieving the smooth trajectory tracking of manipulator by considering the uncertainties due to friction. In the following sections, the kinematic, dynamic modelling and trajectory generation for the harvesting robot is discussed in Section 2. In Section 3, detail explanation about PID and Feed-forward controllers presented. Section 4 contains the model development of robot and the controllers in 'Simulink' environment, in Section 5 contains results, and followed by conclusions in Section 6.

II. Agricultural Robot

A. Kinematic Analysis

The manipulator required for the harvesting purpose is designed based space available in the agricultural field. A virtual workspace is modelled in Solidworks software. a redundant manipulator of 1P8R configuration (9DOF) was selected to access the task space location (TSLs). The Denavit-Hartenberg (D-H) frames assignment for the redundant manipulator is shown in Fig. 2 . The D-H parameters for the robot are given in Table I. For kinematic analysis, transformations between the two adjacent frames, are determined using a homogeneous transformation matrix as given in Equation (1). The location of the end-effector with respect to the base frame is determined using Equation (2). The position and orientation end-effector at TSLs are given for determining the joint variables. The IK are solved using a classical optimization technique i.e., Sequential Quadratic Programming (SQP) [54], which is an advanced nonlinear programming technique. The reachability of the manipulator is the objective function of the IK problem, which minimizing the total cumulative error in Euclidean distances and Euler angles, i.e., the error in the position and orientation of the manipulator's end-effector. A schematic diagram of a serial redundant manipulator is shown in Fig. 3.

TABLE I. D-H parameters of redundant manipulator

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

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

$$ {}^{i-1}_{i}T = {}^{0}_{1}T\ {}^{1}_{2}T\ {}^{2}_{3}T\ {}^{3}_{4}T\ {}^{4}_{5}T\ {}^{5}_{6}T\ {}^{6}_{7}T\ {}^{7}_{8}T\ {}^{8}_{9}T \tag{2} $$

B. Collision Detection

During the manipulation of robot in the agricultural environment, tree trunks and branches cause an obstacle to motion. To avoid collision with these, the outer boundaries of each obstacle are represented as a set of points. Extreme coordinates of these point sets are used to construct a bounding box for the obstacle as show in Fig. 4. To detect collisions, the robot links are modelled as cylindrical objects, and a number of uniformly distributed points are considered on these surfaces. Whenever any of these points are lies within the volume of the bounding box during robot manipulation, Collisions were detected as discussed in [55], [56]. To eliminate the configuration leading to the collision, a penalty approach is adapted during the inverse kinematics. IK problem is formulated as a classical optimization problem as in [22] to determine joint variables for performing agricultural operations at required TSLs.

C. Trajectory Planning

1) Quintic Polynomial Trajectory: The robot manipulator is intended to reach different Task space locations for performing the harvesting tasks. The TSLs are considered as the 'path points', in between these points, a smooth trajectory is needed for motion planning of the robot end-effector. The position, velocity and acceleration of each joint with respect to time period are obtained for the robot manipulator using a trajectory generator. To generate a trajectory for path segments, higher-order polynomial namely the Quintic polynomial used. The equation of quintic polynomial is considered as follows.

$$ \theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{3} $$

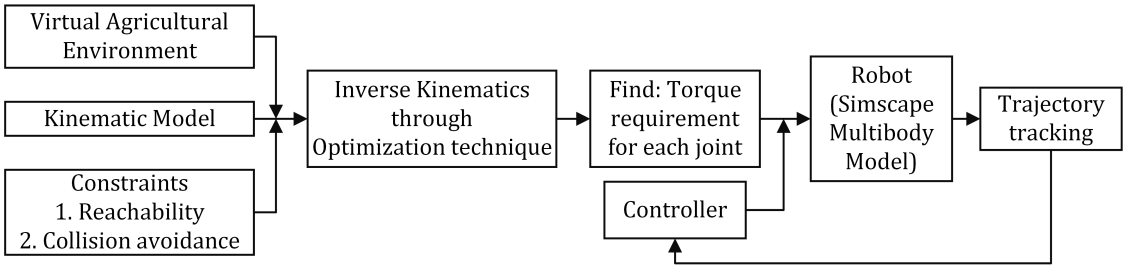


Fig. 1. Flowchart of the methodology adapted.

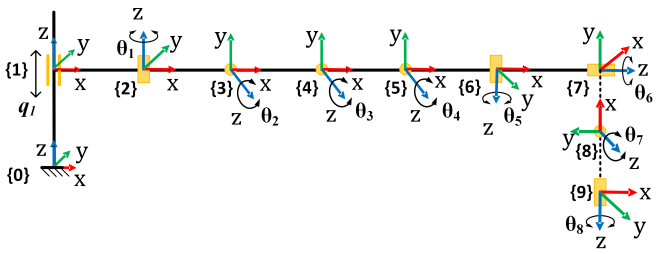


Fig. 2. Frame assignment for the 9 DOF robot manipulator.

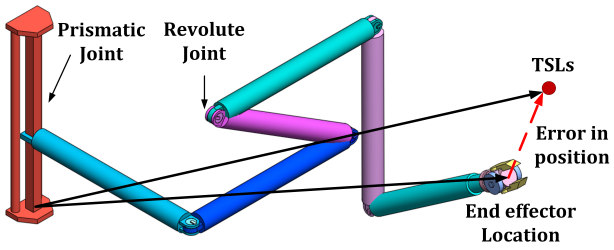


Fig. 3. Representation of positional error of end effector.

The time derivatives of Equation (3) gives the velocities and accelerations, as given in Equation (4) and Equation (5)

$$\dot{\theta} = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4 \tag{4}$$

$$\ddot{\theta} = 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3 \tag{5}$$

Considering the end conditions during the trajectory generation, the problem is formulated as $Ax = b$, where

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix}, x = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}, \& \; b = \begin{bmatrix} \theta_0 \\ \theta_f \\ \dot{\theta}_0 \\ \dot{\theta}_f \\ \ddot{\theta}_0 \\ \ddot{\theta}_f \end{bmatrix}$$

The polynomial coefficients in vector $'x'$ can be determined by using the end conditions given in vector $'b'$ at initial (t) and final (t_0) time.

2) B-Spline Polynomial: When the number of task space locations are increased, trajectory generation with the *quantic polynomial* will results in oscillations, and not suitable robot motion planning. To overcome the problem, a number of polynomials are considered in between the task space locations and it constitutes a spline i.e., to generate a *B-spline* curve, a piece-wise polynomials are defined between a set of control points. The degree of polynomial curve is independent of these control points. Changing the position of a single control point will cause only change in the particular polynomial piece of the *B-spline*. These curves are invariant during affine (translation, rotation and scaling) transformations. These properties are much useful for curve approximation during path generation. These are the basis for the vector of C^{th} order splines.

A B-spline of order C is having the $C-1$ order of piece-wise polynomials, and it is $C-2$ times differentiable. The B-spline polynomial was given in Equation (6), and the derivative of B-spline is given in Equation (7), from this, velocity, acceleration and jerk can be computed.

$$B_{i-j}(t) = \frac{t - t_i}{t_{i+j-1} - t_i} B_{i(j-1)}(t) + \frac{t_{i+j} - t}{t_{i+j} - t_{i+j}} B_{(i+1)(j-1)}(t) \tag{6}$$

$$B'_{i.j}(t) = (j-1)\left[\frac{B_{i(j-1)}(t)}{t_{i+j-1} - t_i} - \frac{B_{(i+1)(j-1)}(t)}{t_{i+j} - t_{i+1}}\right] \tag{7}$$

The trajectory is generated using Quintic polynomial and B-spline polynomials with required velocity and accelerations for the robot motion.

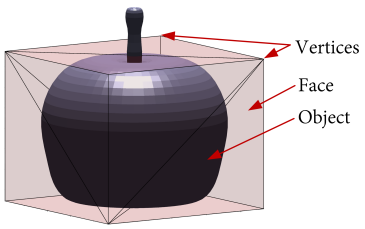


Fig. 4. Fruit encompassed by a bounding box.

D. Dynamic Analysis

The number of joint space coordinates 'm', for the redundant manipulator is exceeds the number of workspace coordinates 'n', resulting in redundancy in nature. The differential equation of manipulator Jacobian, as given in Equation (8).

$$\dot{x} = J(\theta)\dot{\theta} \tag{8}$$

where J is the Jacobian matrix of $n \times m$, $n > m$. is the workspace velocity and is the joint space velocity. The general form of manipulator dynamics is given in Equation (9).

$$M(q)\ddot{q} + V(q, \dot{q} + G(q) + F = \tau \tag{9}$$

where M is the mass matrix, V is a vector containing Centripetal and Coriolis terms. F is the friction, τ is the joint input torques.

III. Controllers

The classical linear controllers i.e., PID controller, linearize the non-linear behaviour of robot manipulators. This controller is easy to implement, and usage of these controllers is restricted to slow and small robotic manipulator systems [57]. If any deviation is presented in trajectory tracking, these controllers provide the required torques at each joint. To control the position, velocity, and acceleration of each joint, linear type controllers i.e., Proportional-Integrate-Derivative (PID), and Feed-Forward (FF) Controllers are used in the present study. The position, velocity and acceleration errors during trajectory tracking are computed using Equation (10).

$$e = q - q_m; \;\; \dot{e} = \dot{q} - \dot{q}_m \;\; and \;\; \ddot{e} = \ddot{q} - \ddot{q}_m \tag{10}$$

Where e, $\dot{e}$ and $\ddot{e}$ are errors in position, velocity and acceleration respectively. q, q_m are the desired and measured positions, $\dot{q}$, $\dot{q}_m$ are the desired and measured velocities, and $\ddot{q}$, $\ddot{q}_m$ are the desired and measured accelerations of the manipulator's joints.

A. PID Controller

In the PID controller, the controller term is given as Equation (11). The K_p, K_d, and K_i are the controller gains and positive definite diagonal matrices. The controller gains are tuned using the trial and error method.

$$\tau = K_p e(t) + K_i \int e(t)dt + K_d \dot{e}(t) - F \tag{11}$$

The schematic diagram of the PID controller is shown in Fig. 5, where represents the required position, velocity and accelerations of the joints respectively, are measured positions, velocities and accelerations of the manipulator joints.

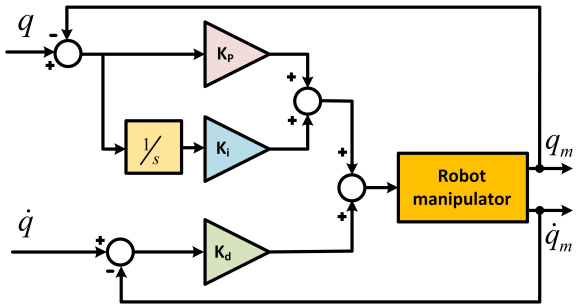


Fig. 5. Schematic diagram of PID controller.

B. Feedforward Controller

A feedback controller is unable to eliminate the persistence disturbances in the robot model. So, a Feed-Forward controller is combined with a feedback controller to reject the disturbances presented in the system. The feedback recovers the errors due to uncertainty in the un-modelled forces, external disturbances and inertial parameters. The schematic diagram of the FF controller is shown in Fig. 6. The control law is given in Equation (12).

$$\begin{aligned} \tau = \; & \{M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q})\} \\ & + \{K_d(\dot{q} - \dot{q}_m) + K_p(q - q_m)\} \\ & = D(q, \dot{q}, \ddot{q}) + \{K_d(\dot{(}q - \dot{q}_m) + K_p(q - q_m)\} \end{aligned} \tag{12}$$

Where, K_p and K_d are the controller gain matrices for position and velocity respectively. D is the inverse dynamics function.

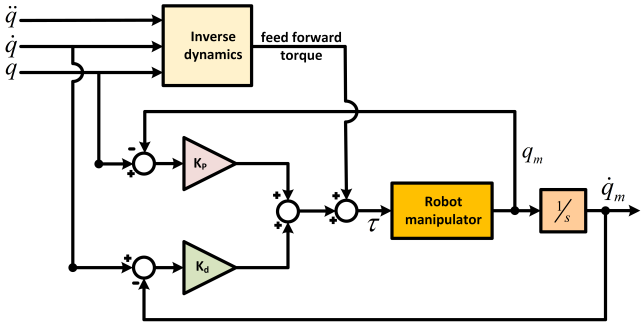


Fig. 6. Schematic diagram of Feed-Forward controller.

IV. Model Development

A. Robot and Environment Model

The 9 DOF robot model is designed in Solidworks software, and the robot model is imported to Matlab environment using Unified Robotic Description Format (URDF) file, as generated by the Solidworks software. The URDF file contains all the physical contacts and joint details. In the Simulink environment, manipulator links are represented with a 'Brick solid' block. Joints between the links are chosen from the 'prismatic joint'

and 'revolute joint' blocks as per the selected robot manipulator configuration. Frictional effects at each joint is given as damping co-efficient in the 'block parameters'. A virtual environment of an agricultural field is modelled in Solidworks as shown in Fig. 7 to replicate the agricultural environment The virtual agricultural environment is saved in *.STL* file format and it is imported to Simulink canvas using the 'file solid' block. The robot model in Simulink is also attached to the virtual agricultural field using the 'Rigid Transform' block as shown in Fig. 8.

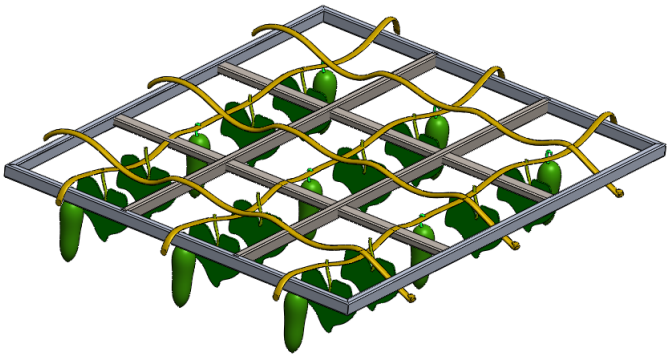


Fig. 7. Solidworks model of agricultural environment.

B. Controller Model

To perform the agricultural applications at different TSLs in Cartesian space coordinates, joint trajectories are generated using the *'polynomial trajectory generator'* block in the Simulink environment available from the *'Robotic System Toolbox'*. In the present work, two different trajectories i.e., Quintic polynomial and *B-spline* trajectories are chosen from the drop-down menu available in the trajectory generator block. The trajectory generator block provides joint positions, velocities and accelerations required for the robot to reach TSLs. Due to uncertainties in the physical model, the robot is unable to trace the joint trajectories. To minimize the joint positional, velocity and acceleration errors, PID and FF controllers are utilized for effective trajectory tracking of the end-effector. The torque required by the robot in the FF controller unit is obtained by the *'Inverse Dynamics block'*. The schematic view of this model is shown in Fig. 9. The controller model for PID and FF in the Simulink environment is shown in Fig. 10a and Fig. 10b respectively.

V. Results

A. Inverse Kinematics

A number of images are taken at the agricultural field. The virtual 3-dimensional environment is modelled using Solidworks software, which replicates the actual agricultural environment. Different task space locations are identified in the virtual agricultural field to perform the fruit picking/plucking operations. These TSLs are the target locations for the robot manipulator. The robot manipulator is required to reach each TSLs (6 in number) by avoiding collisions with surrounding elements of the targeted locations. The robot joint variables are determined by constructing an optimization problem along with collision avoidance technique as in Section 2. The optimized collision-free posture of the robot manipulator is obtained from IK solutions. The inverse kinematic solution at each TSLs are listed in Table II. To generate trajectory between the TSLs, the joint variables are fed to the 'trajectory generator' block in the Simulink environment for the time period of 18 seconds. In the present work, trajectories of manipulator joints are generated by considering Quintic polynomial and B-spline polynomials, for both PID and FF controllers. The joint positions, velocities and accelerations given by the trajectory generator are provided to the controller unit. The controller unit supplies the required torque for the robot model in the Simulink environment. Simulations are shown with the use of Simscape multibody software, as depicted in Fig. 11. Each joint trajectory is observed from the 'scope' box in the Simulink environment.

The torque required for the robot manipulator to reach the TSLs are computed using Equation (9). The variation of torques for reaching each TSLs are shown in Fig. 12, where F_1 is the force required to move the prismatic joint (J_1)and τ_1 to τ_8 are the torques required for the revolute joints J_2 to J_9 respectively.

B. Simulation Results

In the Simulink environment, the robot model, agricultural environment and controller unit are connected sequentially as shown in Fig. 9. The Simulink model is executed to simulate the robot in the virtual environment. The robot motion simulations are carried out in Simscape Multibody as shown in Fig. 11. Joint motions are generated by the proper supply of torque by the controller units. The difference between the desired and current position of the joints are shown using the scope box. During the motion analysis, the trajectory of each joint is compared with the desired trajectory which is generated by the trajectory generator. To reduce the trajectory tracking errors, PID and FF controllers are modelled. The control parameters of these controllers are chosen based on the tuning process. The results obtained from the PID controller and FF controller for trajectory tracking are discussed in this section. All simulations were performed using Matlab/Simulink 2022a on a PC with intel $i3$ processor @2.40 GHz with 8GB RAM.

1) PID Controller with Quintic Polynomial: The trajectory generator block is supplied with robot joint variables for the different TSLs. The position, velocity, and acceleration are generated at each time step. The data generated from the trajectory generator is supplied to the PID control unit. The terms K_i and K_d are control parameters of the PID. The control parameters are chosen randomly at the initial condition, and

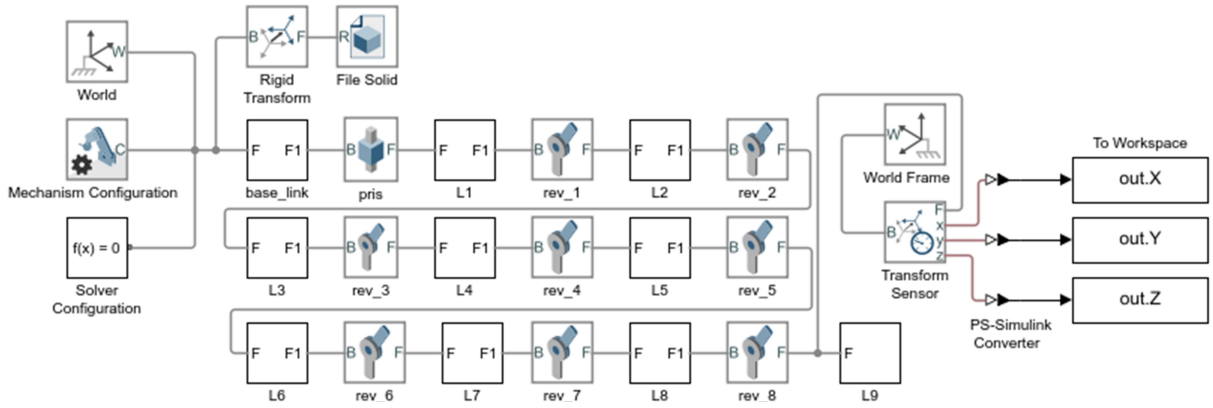


Fig. 8. Simulink model of manipulator.

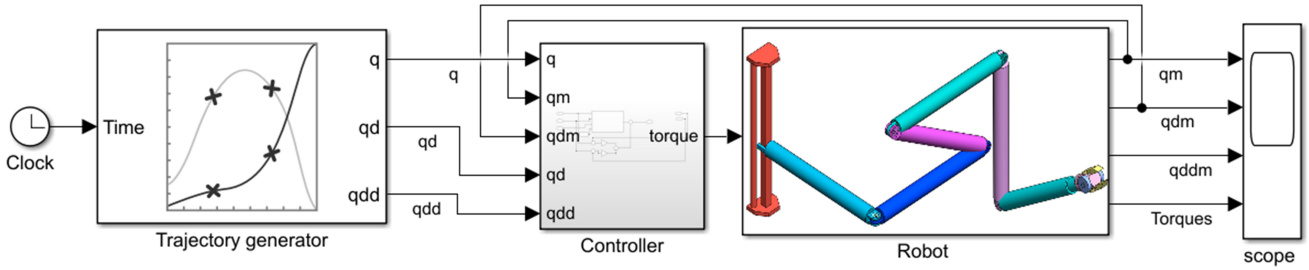


Fig. 9. Trajectory generator, controller, and robot arrangement in Simulink.

TABLE II. INVERSE KINEMATIC SOLUTIONS FOR TASK SPACE LOCATIONS

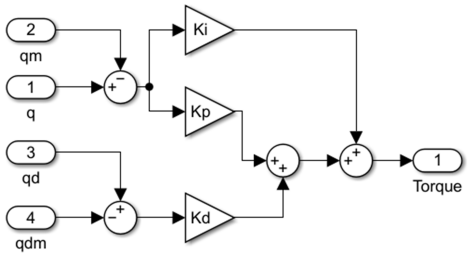
S.No	Task space locations		Joint variables									error
	Position (m)	Orientation (radians)	(m)	(radians)								
			q	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	
1	(0.55, 0.70, 0.55)	(0.349, 0.418, 0.262)	0.076	1.007	0.203	0.457	0.285	−0.401	0.453	0.558	0.209	$1.76E-7$
2	(0.43, 0.38, 0.85)	(0.523, 0.558, 0.628)	0.150	0.880	0.676	0.517	0.352	−0.331	0.593	0.279	0.401	$1.90E-7$
3	(0.35, 0.35, 0.80)	(0.785, 0.733, 0.227)	0.098	0.944	0.713	0.564	0.393	−0.431	0.314	0.366	0.261	$2.69E-7$
4	(0.54, 0.50, 0.60)	(0.872, 0.593, 0.959)	0.082	0.830	0.045	0.671	0.729	−0.446	0.401	0.314	0.074	$3.09E-7$
5	(0.65, 0.63, 0.55)	(1.047, 0.663, 0.750)	0.088	0.839	0.134	0.493	0.361	−0.447	0.209	0.401	0.261	$3.11E-7$
6	(0.70, 0.55, 0.49)	(0.959, 0.837, 0.628)	0.080	0.718	−0.116	0.572	0.681	−0.442	0.61	0.698	0.349	$3.74E-7$

then controllers are tuned for minimum trajectory errors. The control parameters attained after the tuning process are 150, 45 and 25 for K_p, K_i and K_d respectively. From the controller unit, the required torques are supplied to the robot model. The robot motion simulations are observed in the Mechanics Explorer as shown in Fig. 11. The position, velocity and acceleration tracking errors are shown in Fig. 13.

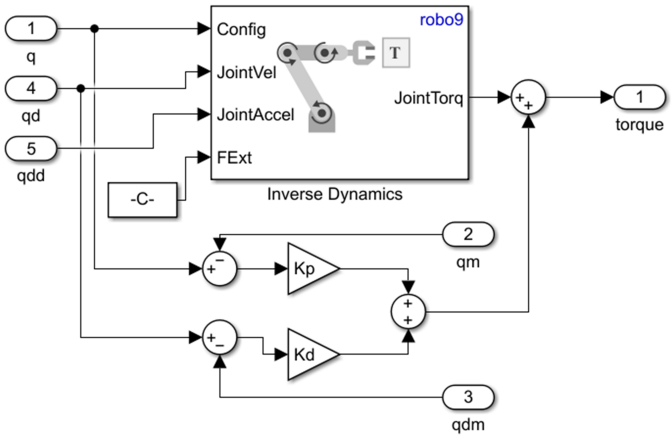
From Fig. 13, the variation of position velocity and acceleration using PID controller with Quintic polynomial are as follows. In the prismatic joint, the maximum error is observed as 0.073 m, 0.0478 m/s and 0.098 m/s^2 for position, velocity and accelerations respectively. In revolute joints, the errors observed as 0.0295 rad, 0.1540 rad/s, and 33 rad/s^2.

2) PID Controller with B-Spline: From Fig. 14, the variation of position velocity and acceleration in B-spline polynomial when controlling using PID controller are as follows. In the prismatic joint, the maximum error is observed for position, velocity and accelerations as 0.073 m, 0.0776 m/s and -48.68 m/s^2 respectively. In revolute joints, the errors are 0.0198 rad, 0.6713 rad/s, and -1.32×105 rad/s^2. When PID controller is used for trajectory tracking, variation of position and velocity in Quintic and B-spline are similar. Although the acceleration in the B-spline throughout the motion is small, during the starting of the motion, there is a sudden change in acceleration, so this type of motion is not desirable for practical applications. The change in accelerations in the Quintic polynomial is also present during the starting of the motion and these values are finite in number. Afterwards, the accelerations are within the limit, which can be adapted for practical applications.

3) FF Controller with Quintic Polynomial: The control parameters attained after the tuning process as K_p and K_d are 10 and 2 respectively for the FF controller. The controlled signal



(a) PID controller in Simulink



(b) FF controller in Simulink.

Fig. 10. Controllers model in Simulink

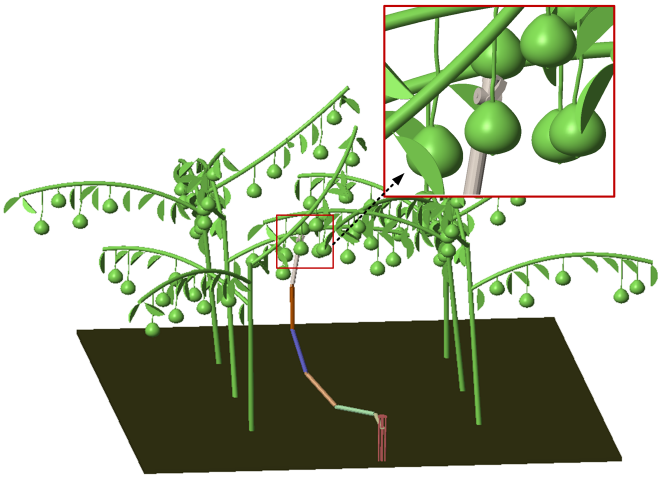


Fig. 11. Snapshot of robot simulation in virtual environment.

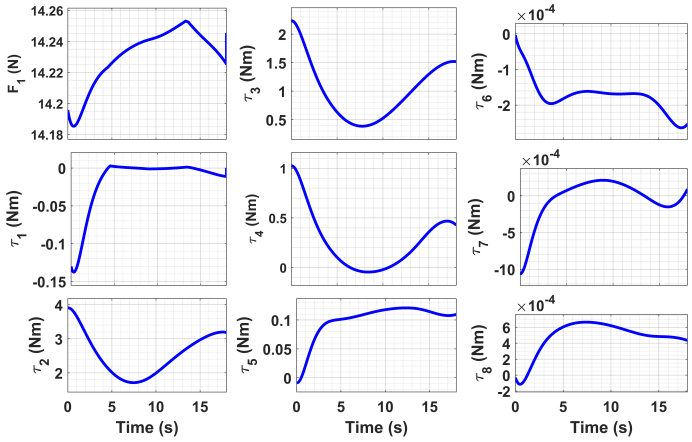


Fig. 12. Torque requirements during motion planning.

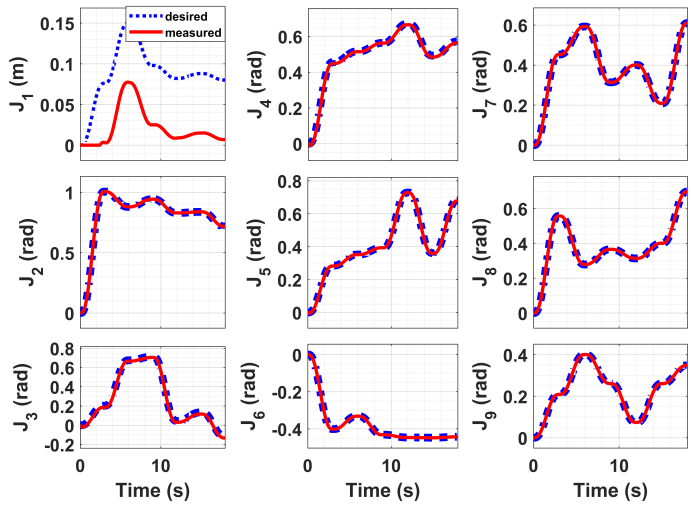
is fed back to the robot model for trajectory generation. The position velocity and acceleration tracking errors are shown in Fig. 15.

From Fig. 15, the variation of position velocity and acceleration using the FF controller with Quintic polynomial are as follows. In the prismatic joint, the maximum error is observed as $6.13 \times 10^{-4}m$, $-7.614 \times 10^{-4}m/s$ and $9.86 \times 10^{-3}m/s^2$ for position, velocity and accelerations respectively. In revolute joints, the errors were observed as $6.51 \times 10^{-4}rad$, $3.101 \times 10^{-3}rad/s$, and $4.9 \times 10^{2}rad/s^2$.

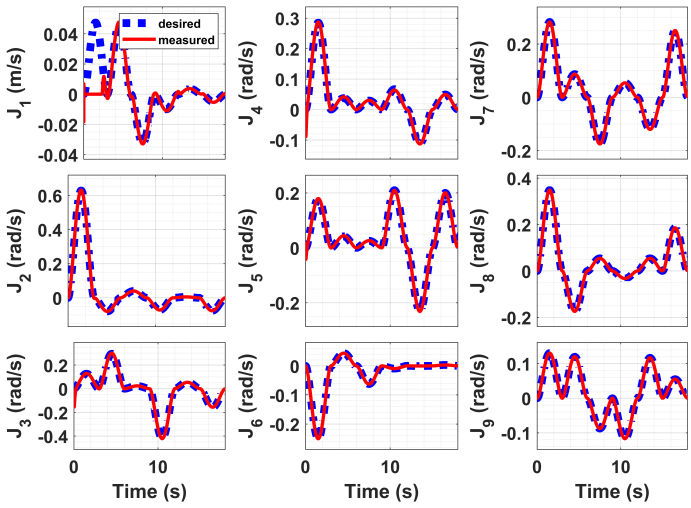
4) FF Controller with B-spline: From Fig. 16, the variation of position velocity and acceleration in the B-spline polynomial when controlling using the FF controller are as follows. In the prismatic joint, the maximum error is observed for position, velocity and accelerations as $0.0213cm$, $-0.0573m/s$ and $-5.122m/s^2$ respectively. In revolute joints, the errors are $0.0745rad$, $0.6713rad/s$, and $-2.511 \times 10^{-3}rad/s^2$. During the implementation of FF controller for the trajectory tracking applications, positional errors in both Quintic and B-spline are negligible and there is considerable errors are presented in FF controller at velocities. In B-spline trajectory, acceleration variations are is small during the motion, but at the starting point of the motion, sudden changes in accelerations are occurred, which is not recommended for practical applications. The accelerations in the Quintic polynomial during the starting of the motion are low and then accelerations are within the limit, which can be implemented for practical applications

The obtained results corresponding to prismatic and revolute joints during controlling with PID and FF controllers for Quintic polynomial and B-spline polynomials are listed in Table III. Bold letters indicate the least errors obtained for a particular joint.

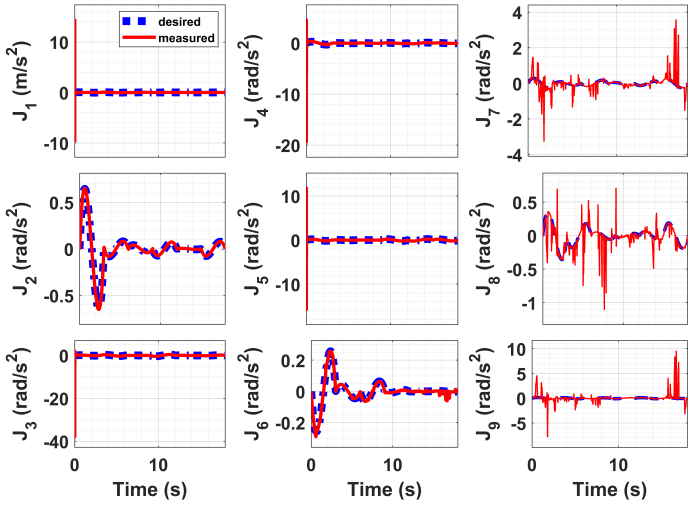
In this study, IK is solved for redundant manipulator working in a cluttered environment using classical optimization technique. the computational time to obtain the IK solutions



(a) Position tracking of Quintic polynomial using PID controller

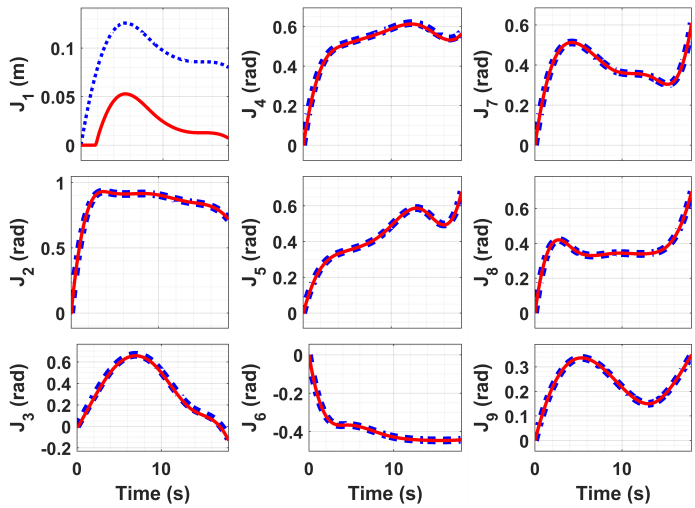


(b) Velocity tracking of Quintic polynomial using PID controller

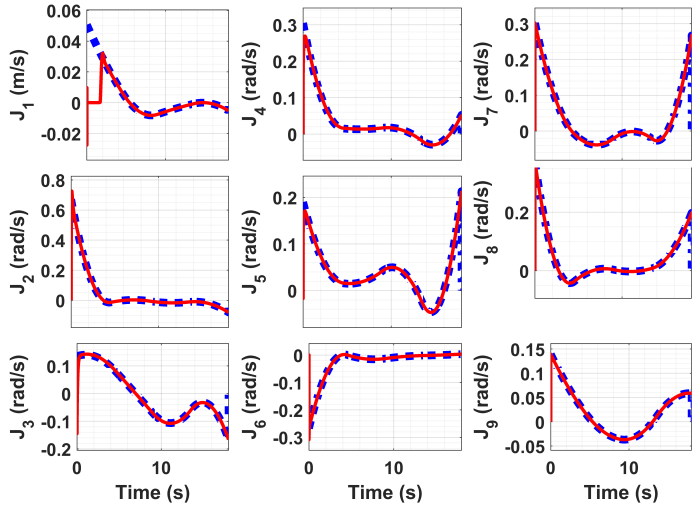


(c) Acceleration tracking of Quintic polynomial using PID controller

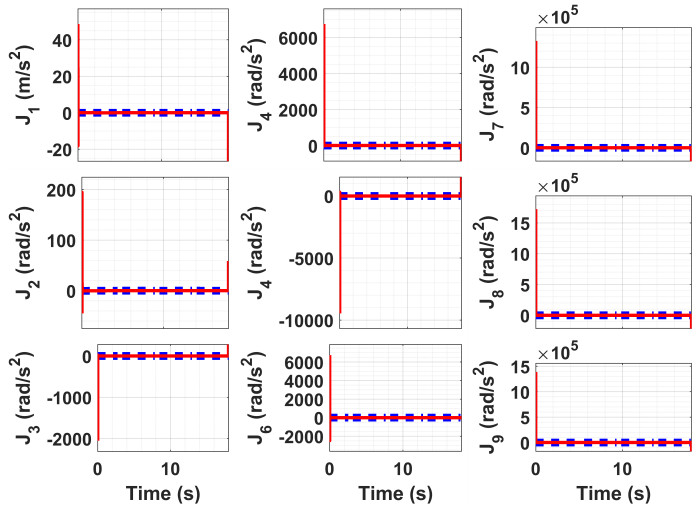
Fig. 13. PID controller for Quintic polynomial trajectory



(a) Position tracking of B-spline polynomial using PID controller

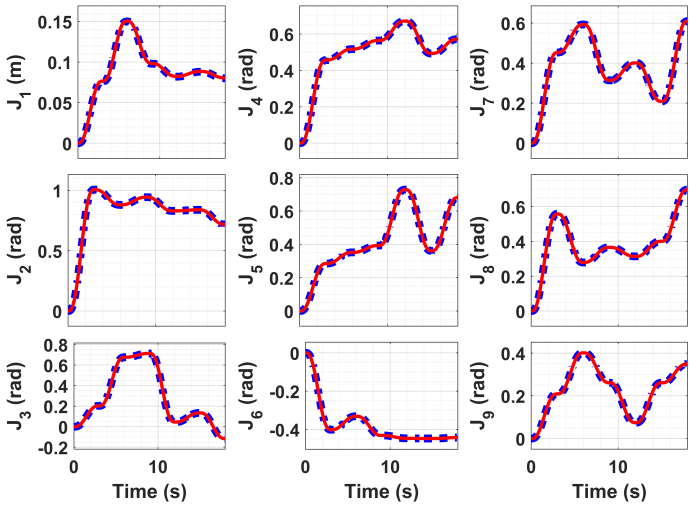


(b) Velocity tracking of B-spline polynomial using PID controller

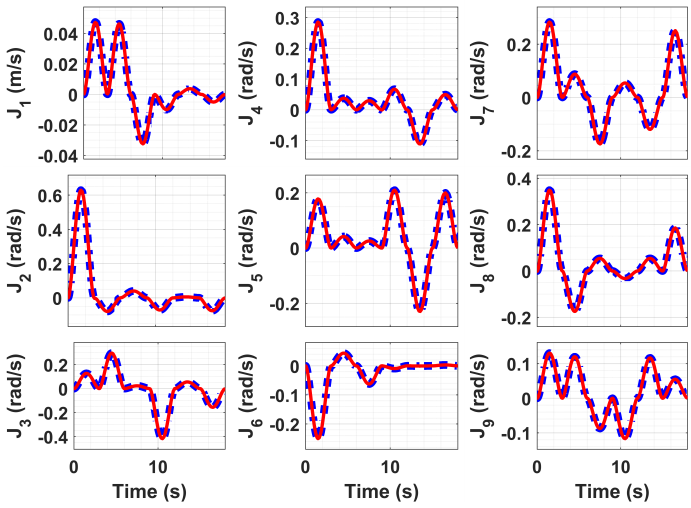


(c) Acceleration tracking of B-spline polynomial using PID controller

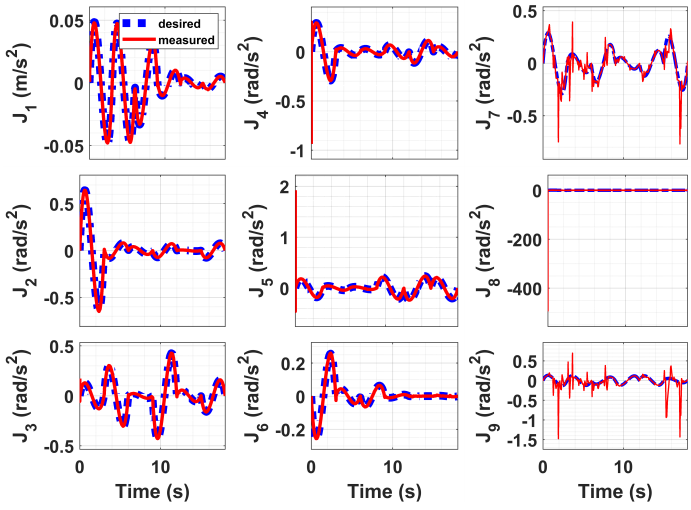
Fig. 14. PID controller for B-spline polynomial trajectory



(a) Position tracking of Quintic polynomial using FF controller

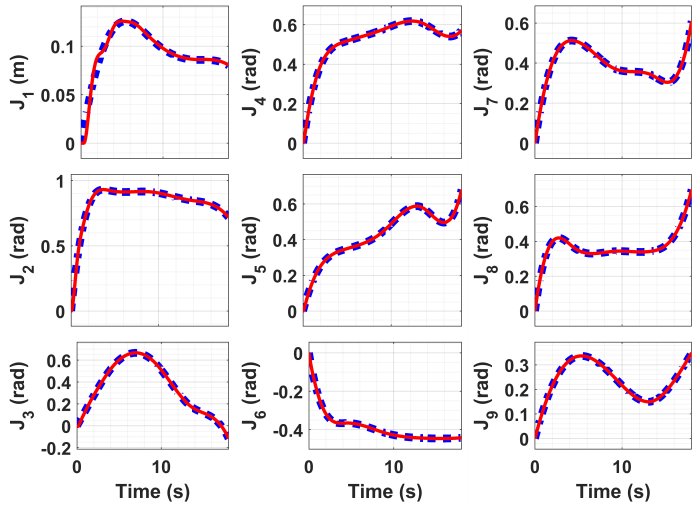


(b) Velocity tracking of Quintic polynomial using FF controller

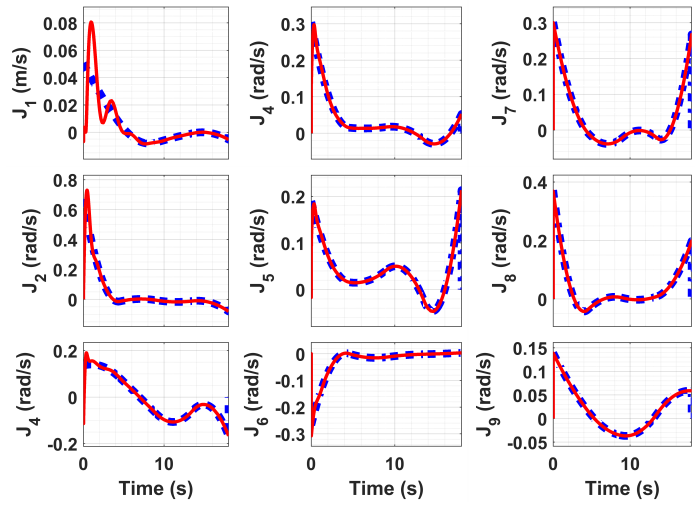


(c) Acceleration tracking of Quintic polynomial using FF controller

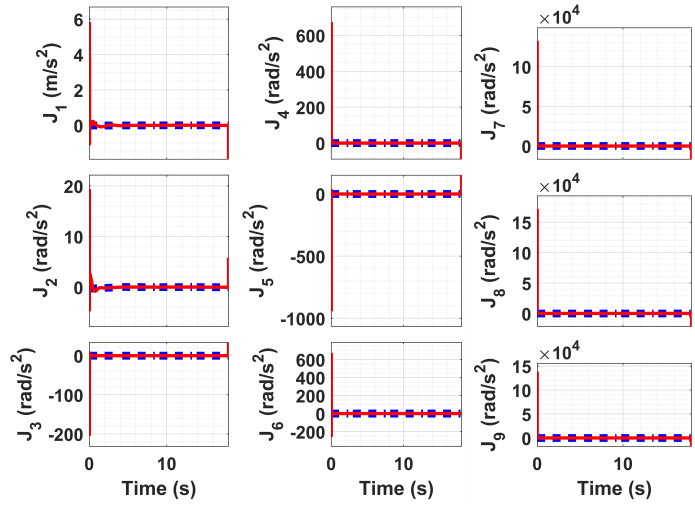
Fig. 15. FF controller for Quintic polynomial trajectory



(a) Position tracking of B-spline polynomial using FF controller



(b) Velocity tracking of B-spline polynomial using FF controller



(c) Acceleration tracking of B-spline polynomial using FF controller

Fig. 16. FF controller for B-spline polynomial trajectory

TABLE III. JOINT SPACE TRAJECTORY TRACKING ERRORS

Joint	Error	PID control		Feed-forward control	
		Quintic	B-Spline	Quintic	B-Spline
Prismatic	pos (m)	0.073	0.073	**-6.1E-4**	0.021
	vel (m/s)	0.048	0.078	**-7.6E-4**	−0.057
	acc (m/s^2)	0.098	−48.68	**9.9E-3**	−5.122
Revolute	pos (rad)	0.021	0.020	**6.5E-4**	0.075
	vel (rad/s)	0.154	0.671	**3.1E-3**	0.671
	acc (rad/s^2)	33	−1.32E5	4.9E2	**-0.0025**

is 54 seconds, which is less when compared to evolutionary algorithms used in [58]. The functional error value for a ten-link manipulator working in a cubical workspace is in order of 10^{-3} [59] which is higher than the nine-link manipulator error in the order of 10^{-7}, due to that the end-effector can reach TSLs precisely. So this method can be suitable for real-time harvesting applications. Joint trajectories are generated between the TSLs using Quintic and B-Spline polynomials and smooth motion of the manipulator was achieved by controlling the torques using PID and FF controllers. The error for the trajectory is in Quintic polynomial is 0.613 mm whereas the standard error for a serial manipulator is 2.18 mm [60], which is 71.8% more than the tuned controller in the present study. Dynamic aspects of the robot are considered in this study, so it can be easily extended to a physical robot model. But manual tuning of the controllers is one of the limitations of this study. From the dynamic simulation study, trajectory tracking of Quintic polynomial with FF controller gives the least errors for smooth motion of the manipulator. The design of the controller is also easy, so it is recommended to use it for agricultural applications. This work was further extended with a real-time robot model.

VI. CONCLUSIONS

Trajectory tracking of 9-DOF redundant serial robot manipulator working in a virtual agricultural environment has been carried out in this work. To perform the fruit-picking operations at different locations, a collision-free trajectory is generated at the joint level. Collisions are avoided using the bounding box technique. The joint variables for reaching the TSLs were obtained through IK solutions and determined using the classical optimization technique. Quintic and B-spline motion trajectories are generated by considering the dynamic aspects of the robot manipulator. For effective trajectory tracking, PID and Feed-forward controllers are tuned for minimal errors in position, velocity and acceleration. From the results, it is observed that trajectory tracking of Quintic polynomial with Feed forward controller has errors less than 1% for both prismatic and revolute joint during position and velocity tracking but it is having 490 rad/s^2 in acceleration. For trajectory tracking of redundant manipulator the Feed-forward controller is suitable for smooth position tracking and minimum disturbances at velocity levels. This work can be further investigated by adding more DOF at each joint and/or using modular robots will increase the flexibility of the robot. IK can also be solved using meta-heuristic algorithms. This study can further be extended the development of real-time robot models for agricultural applications.

REFERENCES

[1] L. F. Oliveira, A. P. Moreira, and M. F. Silva, "Advances in agriculture robotics: A state-of-the-art review and challenges ahead," *Robotics*, vol. 10, no. 2, p. 52, 2021.

[2] S. Fountas, N. Mylonas, I. Malounas, E. Rodias, C. Hellmann Santos, and E. Pekkeriet, "Agricultural robotics for field operations," *Sensors*, vol. 20, no. 9, p. 2672, 2020.

[3] G. Gil, D. Casagrande, L. P. Cortés, and R. Verschae, "Why the low adoption of robotics in the farms? challenges for the establishment of commercial agricultural robots," *Smart Agricultural Technology*, vol. 3, p. 100069, 2023.

[4] N. S. Naik, V. V. Shete, and S. R. Danve, "Precision agriculture robot for seeding function," in *2016 international conference on inventive computation technologies (ICICT)*, vol. 2. IEEE, 2016, pp. 1–3.

[5] G. Belforte, R. Deboli, P. Gay, P. Piccarolo, and D. R. Aimonino, "Robot design and testing for greenhouse applications," *Biosystems Engineering*, vol. 95, no. 3, pp. 309–321, nov 2006.

[6] S. Han, S. Xueyan, Z. Tiezhong, Z. Bin, and X. Liming, "Design optimisation and simulation of structure parameters of an eggplant picking robot," *New Zealand Journal of Agricultural Research*, vol. 50, no. 5, pp. 959–964, dec 2007.

[7] J. Baeten, K. Donné, S. Boedrij, W. Beckers, and E. Claesen, "Autonomous fruit picking machine: A robotic apple harvester," in *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, pp. 531–539.

[8] A. Silwal, J. R. Davidson, M. Karkee, C. Mo, Q. Zhang, and K. Lewis, "Design, integration, and field evaluation of a robotic apple harvester," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1140–1159, apr 2017.

[9] P. Li, S. heon Lee, and H.-Y. Hsu, "Review on fruit harvesting method for potential use of automatic fruit harvesting systems," *Procedia Engineering*, vol. 23, pp. 351–366, 2011.

[10] N. Kondo, M. Monta, Y. Shibano, and K. Mohri, "Basic mechanism of robot adapted to physical properties of tomato plant," in *Proceedings of the Korean Society for Agricultural Machinery Conference*. Korean Society for Agricultural Machinery, 1993, pp. 840–849.

[11] S. Arima, "Cucumber harvesting robot and plant training system," *Journal of Robotics and Mechatronics*, vol. 11, no. 3, pp. 208–212, 1999.

[12] J. Baur, J. Pfaff, H. Ulbrich, and T. Villgrattner, "Design and development of a redundant modular multipurpose agricultural manipulator," in *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, jul 2012.

[13] S. A. Magalhães, A. P. Moreira, F. N. d. Santos, and J. Dias, "Active perception fruit harvesting robots—a systematic review," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, pp. 1–22, 2022.

[14] J. Jun, J. Kim, J. Seol, J. Kim, and H. I. Son, "Towards an efficient tomato harvesting robot: 3d perception, manipulation, and end-effector," *IEEE access*, vol. 9, pp. 17 631–17 640, 2021.

[15] T. Yoshida, T. Kawahara, and T. Fukao, "Fruit recognition method for a harvesting robot with rgb-d cameras," *ROBOMECH Journal*, vol. 9, no. 1, pp. 1–10, 2022.

[16] M. Korayem, R. Ahmadi, N. Jafari, Y. Jamali, M. Kioumarsi, and A. Habibnezhad, "Design, modeling, implementation and experimental analysis of 6r robot," 2008.

[17] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson Educacion, 2005.

[18] A. Aristidou and J. Lasenby, "Inverse kinematics: a review of existing techniques and introduction of a new fast iterative solver," 2009.

[19] A. A. Goldenberg and D. L. Lawrence, "A generalized solution to the inverse kinematics of robotic manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 103–106, mar 1985.

[20] S. Kumar, N. Sukavanam, and R. Balasubramanian, "An optimization approach to solve the inverse kinematics of redundant manipulator," *International Journal of Information and System Sciences (Institute for Scientific Computing and Information)*, vol. 6, no. 4, pp. 414–423, 2010.

[21] A. Bagheri, N. Narimanzadeh, A. S. Siavash, and A. Khoobkar, "Gmdh type neural networks and their application to the identification of the inverse kinematics equations of robotic manipulators (research note)," 2005.

[22] V. V. M. J. S. Chembuly and H. K. Voruganti, "An optimization based inverse kinematics of redundant robots avoiding obstacles and singularities," in *Proceedings of the Advances in Robotics on - AIR '17*. ACM Press, 2017.

[23] N. Rokbani and A. M. Alimi, "Inverse kinematics using particle swarm optimization, a statistical analysis," *Procedia Engineering*, vol. 64, pp. 1602–1611, 2013.

[24] S. Starke, N. Hendrich, S. Magg, and J. Zhang, "An efficient hybridization of genetic algorithms and particle swarm optimization for inverse kinematics," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2016, pp. 1782–1789.

[25] E. Van Henten, E. Schenk, L. Van Willigenburg, J. Meuleman, and P. Barreiro, "Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot," *Biosystems engineering*, vol. 106, no. 2, pp. 112–124, 2010.

[26] D. Puiu and F. Moldoveanu, "Real-time collision avoidance for redundant manipulators," in *2011 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, 2011, pp. 403–408.

[27] D. Han, H. Nie, J. Chen, and M. Chen, "Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection," *Robotics and computer-integrated manufacturing*, vol. 49, pp. 98–104, 2018.

[28] L. Ye, J. Duan, Z. Yang, X. Zou, M. Chen, and S. Zhang, "Collision-free motion planning for the litchi-picking robot," *Computers and Electronics in Agriculture*, vol. 185, p. 106151, 2021.

[29] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan," *Algorithmic and Computational Robotics*, pp. 303–307, 2001.

[30] C. W. Bac, T. Roorda, R. Reshef, S. Berman, J. Hemming, and E. J. van Henten, "Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment," *Biosystems Engineering*, vol. 146, pp. 85–97, jun 2016.

[31] X. Cao, X. Zou, C. Jia, M. Chen, and Z. Zeng, "RRT-based path planning for an intelligent litchi-picking manipulator," *Computers and Electronics in Agriculture*, vol. 156, pp. 105–118, jan 2019.

[32] K. Wei and B. Ren, "A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved rrt algorithm," *Sensors*, vol. 18, no. 2, p. 571, 2018.

[33] H. Zhang, Y. Wang, J. Zheng, and J. Yu, "Path planning of industrial robot based on improved rrt algorithm in complex environments," *IEEE Access*, vol. 6, pp. 53 296–53 306, 2018.

[34] K. Zhang, K. Lammers, P. Chu, Z. Li, and R. Lu, "System design and control of an apple harvesting robot," *Mechatronics*, vol. 79, p. 102644, 2021.

[35] H. Wang, Q. Zhao, H. Li, and R. Zhao, "Polynomial-based smooth trajectory planning for fruit-picking robot manipulator," *Information Processing in Agriculture*, vol. 9, no. 1, pp. 112–122, 2022.

[36] A. Sepehri and A. M. Moghaddam, "A motion planning algorithm for redundant manipulators using rapidly exploring randomized trees and artificial potential fields," *IEEE Access*, vol. 9, pp. 26 059–26 070, 2021.

[37] S. K. Saha, *Introduction to robotics*. Tata McGraw-Hill Education, 2014.

[38] P. Hsu, J. Mauser, and S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, vol. 6, no. 2, pp. 133–148, apr 1989.

[39] G. Zhong, C. Wang, and W. Dou, "Fuzzy adaptive PID fast terminal sliding mode controller for a redundant manipulator," *Mechanical Systems and Signal Processing*, vol. 159, p. 107577, oct 2021.

[40] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[41] A. Mohammadi, M. Tavakoli, H. J. Marquez, and F. Hashemzadeh, "Nonlinear disturbance observer design for robotic manipulators," *Control Engineering Practice*, vol. 21, no. 3, pp. 253–267, 2013.

[42] C. Urrea and J. Kern, "Modeling, simulation and control of a redundant scara-type manipulator robot," *International Journal of Advanced Robotic Systems*, vol. 9, no. 2, p. 58, 2012.

[43] A. Kumar and V. Kumar, "Evolving an interval type-2 fuzzy PID controller for the redundant robotic manipulator," *Expert Systems with Applications*, vol. 73, pp. 161–177, may 2017.

[44] R. P. Borase, D. Maghade, S. Sondkar, and S. Pawar, "A review of pid control, tuning methods and applications," *International Journal of Dynamics and Control*, vol. 9, no. 2, pp. 818–827, 2021.

[45] A. Dhyani, M. K. Panda, and B. Jha, "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*, vol. 162, p. 113021, dec 2020.

[46] S. A. Ajwad, J. Iqbal, M. I. Ullah, and A. Mehmood, "A systematic review of current and emergent manipulator control approaches," *Frontiers of mechanical engineering*, vol. 10, pp. 198–210, 2015.

[47] M. Grotjahn and B. Heimann, "Model-based feedforward control in industrial robotics," *The International Journal of Robotics Research*, vol. 21, no. 1, pp. 45–60, jan 2002.

[48] I. F. Zidane, Y. Khattab, M. El-Habrouk, and S. Rezeka, "Trajectory control of a laparoscopic 3-puu parallel manipulator based on neural network in simscape simulink environment," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9335–9363, 2022.

[49] E. Tatlicioglu, D. Braganza, T. C. Burg, and D. M. Dawson, "Adaptive control of redundant robot manipulators with sub-task objectives," in *2008 American Control Conference*. IEEE, jun 2008.

[50] H. Hülako and O. Yakut, "Control of three-axis manipulator placed on heavy-duty pentapod robot," *Simulation Modelling Practice and Theory*, vol. 108, p. 102264, 2021.

[51] Y. Shaoqiang, L. Zhong, and L. Xingshan, "Modeling and simulation of robot based on matlab/simmechanics," in *2008 27th Chinese Control Conference*. IEEE, 2008, pp. 161–165.

[52] L. Žlajpah, "Simulation in robotics," *Mathematics and Computers in Simulation*, vol. 79, no. 4, pp. 879–897, 2008.

[53] M. Gouasmi, M. Ouali, B. Fernini, and M. Meghatria, "Kinematic modelling and simulation of a 2-r robot using solidworks and verification by matlab/simulink," *International Journal of Advanced Robotic Systems*, vol. 9, no. 6, p. 245, 2012.

[54] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4, pp. 1–51, jan 1995.

[55] V. V. M. J. S. Chembuly and H. K. Voruganti, "An efficient approach for inverse kinematics and redundancy resolution of spatial redundant robots for cluttered environment," *SN Applied Sciences*, vol. 2, no. 6, may 2020.

[56] A. S. Reddy, V. Chembuly, and V. Rao, "Collision-free inverse kinematics of redundant manipulator for agricultural applications through optimization techniques," *International Journal of Engineering, Transactions A: Basics*, vol. 35, no. 7, pp. 1343–1354, 2022.

[57] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer Science & Business Media, 2001.

[58] S. Tabandeh, W. W. Melek, and C. M. Clark, "An adaptive niching genetic algorithm approach for generating multiple solutions of serial manipulator inverse kinematics with applications to modular robots," *Robotica*, vol. 28, no. 4, pp. 493–507, 2010.

[59] E. Singla, S. Tripathi, V. Rakesh, and B. Dasgupta, "Dimensional synthesis of kinematically redundant serial manipulators for cluttered environments," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 585–595, 2010.

[60] Y. Zhang, D. Kim, Y. Zhao, and J. Lee, "Pd control of a manipulator with gravity and inertia compensation using an rbf neural network," *International Journal of Control, Automation and Systems*, vol. 18, no. 12, pp. 3083–3092, 2020.