Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

Dynamic Path Planning for Dexterous Manipulation: A MATLAB Implementation

P Sadhikaa Sree
Dept. of Robotics & Automation
PSG College of Technology
Coimbatore,India

Swathika Subramanian
Dept. of Robotics & Automation
PSG College of Technology
Coimbatore,India

Anbarasi M P
Dept. of Robotics & Automation
PSG College of Technology
Coimbatore,India

***Abstract*— This paper presents a novel approach to dynamic path planning for dexterous manipulation in robotic systems, implemented within a MATLAB framework. Our method focuses on addressing the challenges of collision-free object manipulation in dynamic environments. By leveraging dynamic spherical linear interpolation (SLERP), we achieve precise orientation control during manipulation tasks, allowing for smooth adjustments in response to changing conditions. The proposed approach optimizes motion planning and refines force control to enhance the robot's dexterity and adaptability. This work contributes to the development of more efficient and accurate motion planning for dexterous robotic manipulation in complex environments.**

***Keywords*— Motion planning optimization, collision avoidance, object handling in dynamic settings, SLERP-based orientation control, smooth path adjustments, refined force control, MATLAB implementation, manipulation in complex environments, grasp stability.**

I. INTRODUCTION

Humans can instantly recognize how to grasp objects, a skill that remains far ahead of robotic capabilities. Grasping and manipulation represent significant challenges in robotics. The pursuit of building cognitive robots that match human dexterity has spanned decades, yet despite substantial progress, it remains an unsolved problem in both research and industry[1]. Dexterous manipulation involves multiple manipulators or fingers working together to grasp and manipulate objects, and its focus is object-centered. This contrasts with traditional robotics, where the emphasis is typically on the robotic manipulator itself. Instead, dexterous manipulation defines the task in terms of the object—how it should move and what forces need to be applied to achieve that movement. Conventional robotic grippers are insufficient for this level of control, requiring specialized robotic hands or fingers that can precisely manage forces and motions. Remarkably, even a human infant demonstrates more dexterity than current robots, although they are still far from the abilities of an adult in handling objects. At present, robotic dexterous manipulation is mostly limited to research environments, but model-based approaches have provided valuable insights into the mechanics of dexterous manipulation in both robots and humans. These insights are beginning to influence practical applications, such as in reconstructive surgery, where tendon transfer surgeries are used to enhance grasping abilities in patients with quadriplegia or nerve damage [2]. To improve interaction with objects, it's essential to measure the forces applied and the resulting effects on the object. Another key aspect is developing accurate contact models and selecting optimal grasp points, which play a crucial role in imitating the function of a human hand. Robotic hands can be equipped with force or tactile sensors to better mimic human dexterity [3].
In grasping and manipulation task, a robot is expected to efficiently and effectively grasp an object and then manipulate it. The goal of grasping is to ensure that the robot can fully grasp an object with its robotic hand. The key indicator of success here is the identification and firm grasping (picking) of the object, which means that the uncertainties related to the position, geometry, or nature of the object are efficiently removed and then controlling the movement of the grasped object is as simple as controlling the hand movement. On the other hand, manipulation means the application of force or motion to the same object to change its state and orientation in an environment. In contrast to robot grasping and manipulation, robotic perception in itself grounds the use of robots in the real world. Like human sensory organs responsible for tasks such as sight, hearing, touch, taste, and smell, it is crucial for robots to be able to perceive the real world and its dynamics if they are to autonomously assist humans[5].
It is well known that robots have speed and strength far superior to the human hand, but they cannot reliably grasp unfamiliar objects. This limitation is due to the varying shapes, sizes, and textures of objects, which makes it difficult to build superintelligent machines for household, manufacturing, and security applications. This difficulty stems from the inherent uncertainty in the robot's physics, perception, and control. Virtually all applications, from manufacturing to service to security, would benefit from robots capable of grasping any object with a wide range of shapes and sizes, from rigid to deformable, and under a variety of frictional conditions. Yet despite over 43 years of research, this problem remains unsolved. One plausible reason is that robots rely on simplification of their environment, such as a specific arrangement of objects or strong backlighting that allows better perception and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

localization of the object or subject. Therefore, to alleviate this problem, we need to enable robots to "see" by developing robust perception systems to localize objects and plan robust grasping positions on objects.
A significant interest lies in developing robots that can function in dynamic, unstructured settings, such as in household tasks, bin-picking, or professional services. Learning-based approaches, especially machine learning, are increasingly being employed for robotic grasping. These methods enable robots to autonomously adapt to tasks without human intervention, significantly reducing the need for manual programming. Machine learning holds particular promise due to its ability to generalize to unfamiliar objects [1]. In section 2, the analytical approach and the various methods under are discussed along with the advancements to be made in achieving control in manipulation.

II. CONTROL STRATEGIES FOR DEXTEROUS MANIPULATION

To achieve human-like grasping and manipulation with multi-fingered robotic hands, it is necessary to gather data such as joint angles, contact forces, and contact locations through the use of encoders, force sensors, and tactile sensors. Various research efforts have focused on emulating human grasping and manipulation by integrating these sensors. Studies employing traditional tactile sensors have shown that they can effectively control the position and orientation of objects based on sensory feedback. When humans adjust an object's orientation or position in their hands, they do so with minimal finger motion, relying on a rolling contact between the fingertips and the object. Several studies have successfully recreated this behavior by employing the rolling constraint to ensure reliable grasping and manipulation [4].
The analytical approach to robotic grasping focuses on developing computational algorithms with minimal reliance on extensive data, allowing autonomous control of a robotic hand to complete tasks. This method is based on a physically-grounded, algebraic description of an object in space—often as an approximation or simplification of the actual object or environment. The primary objectives include achieving dexterity, balance, stability, and dynamic performance, with various algorithms addressing each goal. Dexterity is realized by solving an unconstrained linear programming problem, where the objective function is defined by specific dexterity measures. Simultaneously, equilibrium is maintained through algorithms that ensure positivity, friction, and torque constraints on the robot's fingers. Stability is targeted by algorithms that solve for fingertip impedances, ensuring positive definite grasp impedance matrices, while dynamic behavior algorithms calculate fingertip impedances that yield the desired dynamic response.
Recently, data-driven approaches to grasping have gained significant traction, largely due to advancements in deep learning and self-supervised learning techniques, which excel in generalizing to novel objects and unpredictable environments. In these approaches, robots learn to grasp without relying on prior knowledge of the object's features, with training carried out end-to-end. Data-driven methods are typically evaluated empirically and do not necessarily adhere to the physical and dynamic constraints that analytical approaches explicitly model. These approaches are often categorized into supervised and unsupervised (reinforcement learning) methods [5].The overall manipulation is achieved by controlling the following: Impedance control, Force control, Position control and trajectory tracking. This paper focuses on dynamic SLERP, optimized path planning and more refined force control. Detailed presentations of most current dexterous manipulation approaches are analyzed in the subsequent section. The scope will include exploration of novel methods to improve the accuracy and robustness of robotic grasping and manipulation such as utilization of state-of-the-art algorithms, vision sensors and dynamic control techniques. These advancements are designed to enhance the work performance of robotic systems when performing complex manipulation tasks.

III. ADVANCEMENTS

A. Dynamic SLERP in Dexterous Manipulation

In the field of dexterous manipulation, precise orientation control of robotic end-effectors plays a crucial role. To achieve smooth and adaptive orientation transitions, especially when manipulating complex or dynamic objects, Spherical Linear Interpolation (SLERP) is commonly employed. SLERP is ideal for continuous and smooth rotational transitions, as it provides constant velocity and ensures that the interpolated orientations are on the shortest path between two quaternions [1].
Theoretical Basis of SLERP:
Spherical Linear Interpolation (SLERP) is a method used to transition between two orientations, represented by quaternions q_1 and q_2, which reside on the surface of a 4D unit sphere. This interpolation method, grounded in spherical geometry, ensures that the transition occurs along a great circle, effectively minimizing the rotational path. The SLERP formula is expressed as [2]:

$$SLERP\ (q_1, q_2, t) = \frac{\sin((1-t)\theta)}{\sin(\theta)} q_1 + \frac{\sin(t\theta)}{\sin(\theta)} q_2$$

Here, θ represents the angle between the two quaternions, and t is a parameter that varies from 0 to 1. This formula allows for a smooth, continuous rotation from q_1 when t = 0 to q_2 when t = 1, making it an essential tool for achieving stable and precise control in dexterous robotic manipulation tasks [3].

Dynamic SLERP in Dexterous Manipulation:
In dynamic robotic systems, particularly in dexterous manipulation, SLERP can be combined with real-time sensor feedback and control algorithms. This dynamic adaptation is necessary when the robot interacts with objects in uncertain environments, where precise control over orientation and movement trajectory is crucial to ensure successful manipulation [4]. By continuously adjusting the SLERP interpolation based on sensory input, the robot can smoothly alter its grip or orientation as needed, thus improving its adaptability and precision in tasks such as object retrieval or assembly [5].
For example, in scenarios involving grasping deformable objects, real-time adjustments to the end-effector's orientation can be made using dynamic SLERP, helping maintain stability and precision while minimizing errors [6]. The dynamic extension of SLERP also enhances the robot's ability to cope with varying external forces and object

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

motions during manipulation, an important aspect for dexterous handling in unstructured environments [7].

Mathematical Extension of SLERP for Dynamic Manipulation:

When applied in dexterous manipulation, SLERP can be integrated with control strategies such as impedance control or force feedback mechanisms to create a dynamic system. The process begins with defining the initial and target orientations as quaternions, q_1 and q_2. A smooth transition can be achieved through SLERP by dynamically adjusting the interpolation parameter t based on real-time data such as tactile feedback from sensors on robotic fingers or visual feedback from cameras [8]. This dynamic approach is especially useful in multi-fingered robotic hands, where each finger might need to adjust its orientation in concert with others [9].

The practical application of SLERP in dynamic manipulation is evident in research on multi-sensory feedback systems, where the use of SLERP allows the robot to maintain smooth transitions between various configurations of its end-effector while responding to real-time changes in the environment [10]. Such systems have been shown to improve the robot's ability to manipulate complex, deformable objects, such as textiles or food products, with minimal disturbance [11].

B.Optimized Motion Planning for Dexterous Manipulation

Optimal motion planning for dexterous robotic manipulation has garnered significant interest due to the complex challenges involved in navigating obstacle-rich environments. In such settings, the ability to efficiently plan a collision-free trajectory while maintaining grasp quality is crucial for robotic systems. This section outlines the theory behind optimal motion planning, emphasizing the integration of kinematic and dynamic constraints to ensure smooth and efficient manipulation.

1. Optimal Motion Planning:

Optimal motion planning in dexterous manipulation revolves around finding a trajectory for the manipulator that minimizes a given cost function, subject to constraints. The cost function often incorporates factors such as energy consumption, time, and path length. In the context of dexterous manipulation, additional objectives such as maintaining grasp stability and avoiding collisions must also be considered [2].

The general motion planning problem can be defined as:

Mininmise,

$$\int_0^T J(q(t), \dot{q}(t), \ddot{q}(t))dt$$

where q(t) is the configuration of the robot at time t, $\dot{q}(t)$ and $\ddot{q}(t)$ represent the velocity and acceleration, respectively, and J() is the cost function representing factors like energy usage or path efficiency.

2. Collision Avoidance Constraints:

In environments with obstacles, collision avoidance becomes a key component of the optimization process. The manipulator's trajectory must be planned such that no part of the robot collides with the obstacles. This can be formulated as inequality constraints on the trajectory:

$$g_{obs}(q(t)) \geq 0$$

where g_{obs} is a function that represents the distance between the manipulator and any obstacle. If the distance is positive, no collision occurs [3].

3. Kinematic Constraints and Joint Limits:

The configuration of the robot must respect kinematic constraints such as joint limits and workspace boundaries. These constraints can be expressed as:

$$q_{min} \leq q_i(t) \leq q_{max}$$

where $q_i(t)$ is the joint angle of the i-th joint at time t, and q_{min} and q_{max} represent the joint's lower and upper limits, respectively [6].

4. Grasp Stability:

In dexterous manipulation, it is essential to maintain a stable grasp throughout the motion. The grasp stability is often characterized by the grasp quality metric, Q_o , which evaluates the robustness of the grasp. A common approach to quantify grasp stability is based on wrench space analysis, where the manipulator applies forces that generate the necessary torques to hold the object [7].

The grasp quality function can be expressed as:

$$Q_o(f_c) = \gamma_{min}(G f_c)$$

where f_c represents the contact forces, and G is the grasp matrix that maps contact forces to object wrenches γ_{min} represents the minimum eigenvalue, which is used to evaluate the grasp quality based on the stability of the applied forces [5].

5. Dynamic Constraints and Optimal Control:

In addition to kinematic constraints, dynamic constraints govern the robot's motion, particularly when performing tasks that involve rapid manipulation or large objects. The dynamics of the system can be represented by the equations of motion [3]:

$$Mq" + C(q, q')q' + G(q) = \tau$$

6. Learning-Based Optimization:

Recent advancements in learning-based methods have shown promise for improving the efficiency of motion planning in dexterous manipulation [1]. Learning-based approaches leverage data from previous interactions to improve the robot's ability to avoid obstacles and optimize trajectories in real-time. These methods can significantly reduce the computational burden of traditional optimization techniques by learning a model of the environment and utilizing it to guide the search for optimal trajectories.

Learning-based optimization techniques can be expressed as:

$$q(t+1) = q(t) + \alpha \nabla_q J(q(t))$$

Where $\alpha \nabla_q J(q(t))$ is the gradient of the cost function with respect to the robot configuration. This approach enables robots to refine their motion plans over time by adapting to environmental changes [9].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

7. Multi-Sensory Feedback:
Another crucial aspect of optimal motion planning is the use of multi-sensory feedback, which enhances the robot's ability to adapt its motion based on tactile, visual, and force sensor data. Multi-sensory feedback allows the manipulator to detect contact forces, object slippage, and proximity to obstacles in real-time, facilitating the adjustment of its trajectory [4].

C.Refined Force Control for Dexterous Manipulation of Robots

Force control can be categorized into two primary approaches: impedance control and admittance control. In impedance control, the robot's end-effector behaves like a mechanical impedance, which modulates the relationship between the applied forces and the resulting motion. Conversely, in admittance control, the robot's motion is dictated by the external forces applied to it.

Impedance Control:
Impedance control can be mathematically described by the following equation:

$$F = M\,(\ddot{x}_d - \ddot{x}) + B\,(\dot{x}_d - \dot{x}) + K\,(x_d - x)$$

This model allows the manipulator to adapt its motion based on the interaction forces, ensuring compliance when needed, such as during delicate tasks

Admittance Control:
In admittance control, the relationship between force and motion is represented as:

$$\dot{x} = M^{-1}(F - D\dot{x} - K(x - x_d))$$

where:

- F is the external force applied to the robot,
- the other variables are as previously defined.

This framework allows the manipulator to respond dynamically to the forces acting on it, making it suitable for tasks requiring quick adjustments [1][2].

Hybrid Control Strategies:
Recent advancements propose hybrid control strategies that integrate both impedance and admittance control, allowing for more versatile handling of objects. These strategies combine the benefits of both approaches, enabling the robot to switch between control modes based on the task requirements [5][6].

Model Predictive Control (MPC):
MPC has emerged as a powerful tool in refined force control, where the future states of the system are predicted based on a dynamic model, allowing for proactive adjustments to the control inputs. The optimization problem can be formulated as:

$$\min_{u} \sum_{k=0}^{N} \left(\left\| x_{k+1} - x_{target} \right\|^2 + \lambda \| u_k \|^2 \right)$$

subject to:

$$x_{k+1} = Ax_k + Bu_k$$

where:

- u represents control inputs,
- x_{target} is the target state,
- A and B are system matrices, and
- λ is a weighting factor [10][11].

This approach facilitates robust control in environments with uncertainties, improving the robot's ability to perform dexterous tasks.

IV. PROBLEM STATEMENT

In order to incorporate dynamic SLERP (Spherical Linear Interpolation), optimized path planning, and more refined force control. The aim is to move a cube from an initial position $p_A = [0,0,0]$ and orientation $R_A = I$ (identity matrix) to a final position $p_B=[1,1,1]$ orientation R_B = (90-degree rotation about the z-axis).

A. Initial and Final configurations
Initial Position:

$$p_A = [0, 0, 0]$$

Initial Orientation:

$$R_A = I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Final Position:

$$p_B = [1, 1, 1]$$

Final Orientation (90-degree rotation around the z-axis):

$$R_B = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

B.Trajectory Planning
For linear interpolation, the position trajectory is defined as:

$$p(t) = p_A + t\,(p_B - p_A)$$

where t ranges from 0 to 1. This is for generating a straight-line path from p_A to p_B.

Orientation Trajectory:
For the orientation, the SLERP is used, which smoothly interpolates between two orientations.
The equation is:

$$R(t) = R_A \,.\, \exp\,(\log\,(R_A^T R_B)\,.\,t)$$

where log and exp are matrix logarithm and exponential functions, respectively. This ensures smooth rotation along the shortest path on the sphere of orientations.

C.Grasp Planning
The force closure criteria is used to determine the cube's stable grasp points. Doing this we ensure that the applied forces at the grasp points can resist the external perturbations if any. Fig.1 illustrates the different positions of the grasping model.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

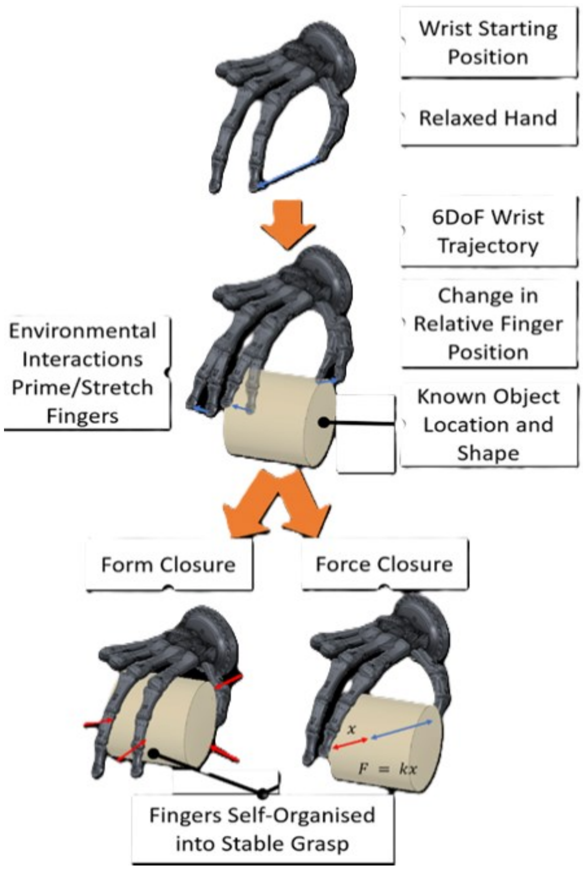

Fig 1: Grasping Model

For a successful force closure grasp, the fingers must initially be spread open through interactions with the object and surroundings, allowing the object to be either enclosed within a cage or positioned between two or more fingers that then apply force through internal spring mechanisms.

D.Force Control

Impedance control is used to handle force interactions. The force equation is defined as:

$$F = M(\ddot{x}_d - \ddot{x}) + B(\dot{x}_d - \dot{x}) + K(x_d - x)$$

Where F - Control force, M - Inertia matrix a square matrix that defines how mass is distributed in the system, B - Damping matrix helps to reduce oscillations and stabilize the system modulating velocity, K - Stiffness matrix reflects the systems resistance to deformation, x_d - Desired position, x - Current position, $\dot{x}_d$ and $\dot{x}$ - Desired and present velocities, $\ddot{x}_d$ and $\ddot{x}$ - Desired and present accelerations.The first term in the equation allows the robot to react based on differences in desired and present accelerations. It generates the appropriate force needed to accelerate or decelerate to follow the desired trajectory. The second term helps dampen the motion. If the present velocity exceeds the desired velocity this produces a force to slow down the robot to achieve a smooth movement and also prevents overshooting. The third term generates a restoring force that brings the robot back to the necessary position. If there are any deviations from the path this term ensures it falls back in the right path.

The impedance control approach focuses on regulating the manipulator's dynamic behavior, rather than controlling specific vector quantities like force, position, or velocity. This method enables uniform handling of all task conditions, such as free motions or physical interactions, by simply defining and imposing an appropriate dynamic response on each robotic finger. Additionally, this approach offers several key advantages, including simplifying the overall control system. A dexterous robotic hand presents significant complexity, involving many degrees of freedom, a large volume of sensory information, and multiple control goals. Impedance control enables each finger to be viewed as an independent system, helping to mask the nonlinear and interconnected dynamics of the mechanical structure with a preferred dynamic response. To simplify, this desired response is often modeled as linear and decoupled across different workspace directions, similar to a mass-spring-damper system. This approach significantly reduces the complexity for the supervisory control system [6].

E.Dynamic Simulation

The force equation for dynamic simulation is given by:

$$Mq'' + C(q,q')q' + G(q) = \tau + J^T F$$

where M - inertia matrix, C - centrifugal matrix, G – gravity vector, J – jacobian matrix, F – external forces and the control torque. The product of force and Jacobian changes according to the forces that act at the end of the manipulator affecting the torque on each joint. The equation is solved using Euler's method to simulate motion over time, the present state q and q' is updated based on the calculated torque and force values iteratively. The τ is controlled by the feedback from the sensors achieving adaptive control.

F. Simulation Outcomes

On simulating these equations in MATLAB, the trajectory followed to move a cube from an initial position p_A = [0,0,0] to a final position p_B=[1,1,1] was simulated and the output graph was obtained The desired position and orientation is computed and is updated. The same checks for grasp stability and applies impedance control to calculate the force.
In the simple interpolation approach, the cube moves along a straight path from the initial position to the final position with linear progression in both translation and orientation. The movement is smooth and predictable but lacks sophistication in terms of handling dynamic constraints or environmental factors. The cube transitions between the start and end points without considering variations in forces or system dynamics, which works for basic tasks but doesn't account for real-world complexities. Fig.2 shows the outcome of the approach.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

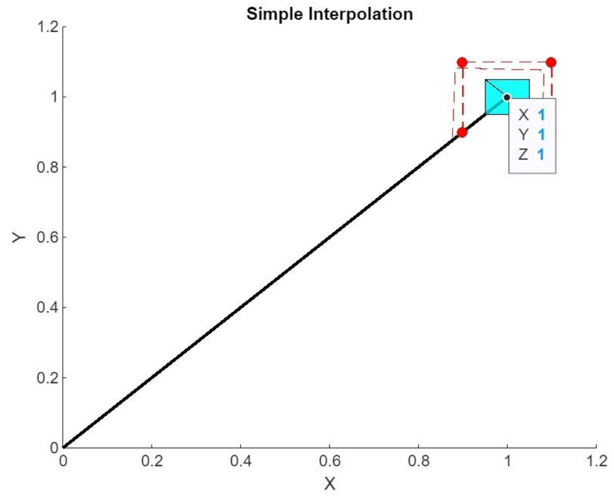

Fig 2: Simple Interpolation

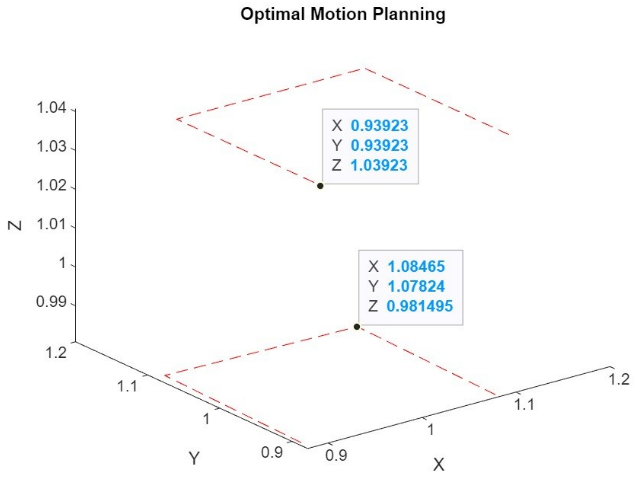

Fig. 4. Optimal Motion Planning

The dynamic SLERP interpolation offers a significant improvement by interpolating the orientation using spherical linear interpolation. This technique ensures that the cube follows the most natural rotational path between its initial and final orientations. Unlike the linear method, which can produce abrupt or unnatural rotations, SLERP guarantees smooth, continuous orientation changes, especially important for tasks like rotating the cube by 90 degrees about the z-axis. Compared to simple interpolation, SLERP provides better rotational accuracy, making it more suitable for environments where precise orientation is critical. Fig.3 showcases the smooth rotational path achieved with SLERP interpolation.

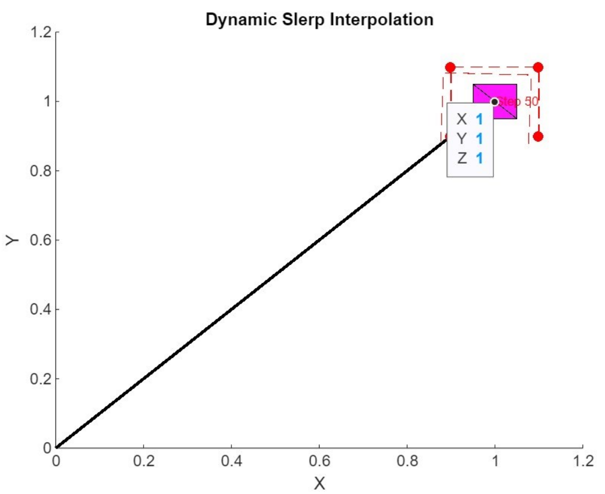

Fig 3: Dynamic SLERP Interpolation

The optimal motion planning approach further refines the motion by applying gradient descent to find the most efficient path between the start and end positions. Instead of following a direct linear path, this method minimizes a cost function to optimize the cube's trajectory. While simple interpolation doesn't account for possible inefficiencies or constraints, optimal motion planning dynamically adjusts the cube's trajectory, resulting in a more energy-efficient and time-optimal path to the final position $p_B=[1,1,1]$. This is particularly beneficial in more complex environments where direct paths may not be ideal or feasible. Figure 4 displays the optimized trajectory using motion planning.

Lastly, refined force control combines techniques like impedance, admittance, and hybrid control with model predictive control (MPC) to generate a motion that is adaptive and responsive to external forces. This advanced approach ensures that the cube not only reaches its destination optimally but does so by considering both internal dynamics and external environmental factors, making the motion highly adaptive. Compared to simple interpolation, refined force control offers superior precision and stability in handling complex environments, such as those with varying stiffness or unknown obstacles, resulting in a more realistic and robust performance. Fig.5 illustrates the adaptive behavior of the refined force control strategy.

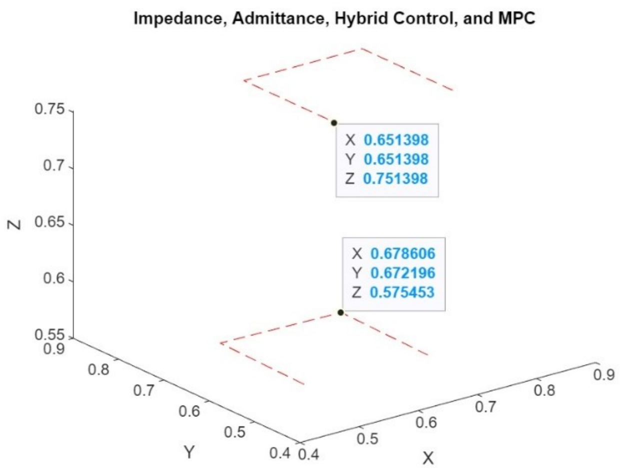

Fig. 5. Force Control

Combining these approaches, we leverage advanced techniques to control both the position and orientation of the cube, achieving a final output position of approximately [0.993, 0.993, 0.993], which is nearly identical to the target [1, 1, 1]. The close but slightly imperfect result stems from the iterative nature of the optimization process and the trade-offs involved in maintaining both smooth motion and control stability. As the cube moves, smooth rotational transitions and gradual positional adjustments are made simultaneously, ensuring that the motion appears natural and fluid. However, the final slight deviation from the exact target can be attributed to the inherent limitations of gradient-based optimization methods, which, while effective at convergence, may not always reach the precise target due to the step size and the balancing of dynamic factors. This outcome demonstrates how combining multiple control methods enhances overall system performance but still may introduce small, acceptable inaccuracies in real-world applications. The visual output from this approach, as shown in Fig.3, illustrates dexterous manipulation achieved with these combined advancements.

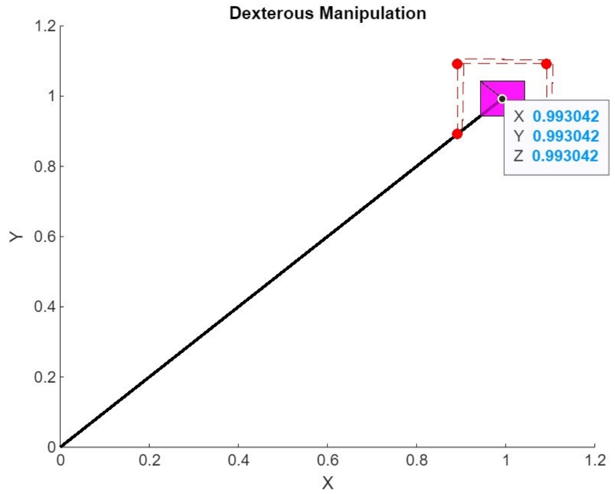

Fig 6. Dexterous Manipulation

V. CONCLUSION

Therefore, in this research, a novel method for dynamic path planning has been developed and proposed to advance dexterous manipulation with robotic systems. With the help of realizing dynamic spherical linear interpolation (SLERP) in MATLAB, our approach provides a potential solution for avoiding-object-collision problem in dynamic environment. Precise orientation control under the condition of adaptability is a phenomenal leap featured in the flexibility and motility of robotics. While preliminary versions of the validations bear great potential, more testing is needed to explore the potential use of our methodology in actual practice. Therefore, this work lays the foundation for faster and more reliable motion planning that facilitates robotic manipulation in the next step.

While initial conceptual validation has been completed, further experimentation is required to fully evaluate the performance in practical scenarios.

REFERENCES

[1] 1. Kleeberger, K., Bormann, R., Kraus, W., & Huber, M. F.," A Survey on Learning-Based Robotic Grasping", Springer,2020.

[2] 2. Okamura, A. M., Smaby, N., & Cutkosky, M. R.,"An Overview of Dexterous Manipulation", Dexterous Manipulation Laboratory, Stanford University,2000.

[3] 3. Khalil, F. F., & Payeur, P. ,"Dexterous Robotic Manipulation of Deformable Objects with Multi-Sensory Feedback: A Review.", University of Ottawa,2020.

[4] 4. Yamaguchi, A., & Atkeson, C. G.,"Dexterous Object Manipulation by a Multi-fingered Robotic Hand with Visual-Tactile Fingertip Sensors.", Research Article, Open Access,2020.

[5] 5. Bohg, J., Morales, A., Asfour, T., & Kragic, D. "A Survey: Robot Grasping.", IEEE Transactions on Robotics,2021.

[6] 6. Bicchi, A., & Kumar, V.," Robotic Grasping and Contact: A Review.", Proceedings of IEEE International Conference on Robotics and Automation,2020.

[7] 7. Rojas, N., & Dollar, A. M.," Optimal Grasping and Dexterous Manipulation Using Soft Hands.", IEEE Transactions on Robotics,2017

[8] 8. Kroemer, O., Detry, R., Piater, J., & Peters, J. ,"Active Learning of Object-Scene Models for Robotic Manipulation.", Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems,2012.

[9] 9. Pham, C., & Canny, J. F. ,"Obstacle avoidance in force-controlled robotic manipulators using potential field methods.", IEEE International Conference on Robotics and Automation,1994.

[10] 10. Fan, Y., & et al. ,"Optimization-based controller framework for dexterous robotic manipulation.", Proceedings of the 20th IFAC World Congress, Toulouse, France, July 9-14.

[11] 11. "Cartesian Impedance Control for Dexterous Manipulation"- Conference Paper , November 2003.