Trajectory Planning for robotic manipulators : Insights in to modern day practices with Algorithms and existing software libraries

*

Deepak Jowel
School of Computer Science
University of Lincoln
Lincoln, U.K

***Abstract*—This paper discusses the different methodologies for trajectory planning by highlighting the state-of-art algorithm practiced to achieve the task and the software libraries used to do so . In the domain of robotics, trajectory planning is pivotal as it enable a robot to interact with it environment by introducing paradigms that will make the robot capable to avoid obstacles and reach a desired state to successfully complete an operation or task. Moreover, for the real-time execution of the planned motion, it is necessary to define proper position/velocity control algorithms, in order to optimize the performances of the system and to compensate for disturbances and errors during the movements, such as saturations of the actuation system. This makes a robot efficient and optimize the workspace for the environment in which they have been deployed. The introduction part of this paper defines the basic priciple adapted to structure a solution for the task. Also, highlighting the do's and don'ts while developing a system. The current state of art practises have been briefly put together in the literature review section of this paper, where the solution for different application environment with their shortcomming have been put forward.**

***Index Terms*—Trajectory Planning, planning algorithms, Software Libraries**

I. Introduction

As described by Matteo (Massaro et al., 2023) "Trajectory planning is a classic problem in robotics and is usually defined as finding the timing of the motion law along a given geometric path while satisfying certain requirements and minimising certain objectives."

As per Castro,S. (Castro, 2019), trajectory planning can be considered as a subset of the the wider domain of problem which is navigation or motion planning. In the domain of manipulator it is necessary to consider algorithms, specific to a robots task. As stated in (Siciliano et al.,2009,161-189) the objective of trajectory planning is to support the motion control system by generating reference inputs, which enables the manipulator to execute a planned trajectories. Siciliano, further elaborates that the problem lies in making a decision to specify the motion at the joint or directly at end-effector with reference to any task assigned to the manipulator.

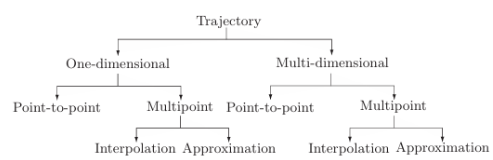

Fig. 1. Main trjectory categories as described in (Biagiotti and Melchiorri, 2008)

The foremost step in solving the trajectory planning problem is to choose a approach to solve the task. The author of (MathWorks Editor, 2019) states that these approaches are of two types which are , joint-space "means the way points and interpolation are directly on the joint positions (angles or displacements, depending on the type of joint)" and task-space "means the way points and interpolation are directly on the joint positions (angles or displacements, depending on the type of joint)".

The topic has been further elaborated on by the authors of (Biagiotti and Melchiorri, 2008) by stating that the concerns of the domain can be addressed as a problem for finding the relationship between two elements : time and space. They define trajectory as parametric function of time providing the desire position at each instance, further emphasizing on other aspects such as time discretization , saturation of the actuation system, vibrations induced on the load, and so on.

Moreover the author states that, a complex motion can be achieved by sequentially traversing from point-to point for a given trajectories which are individually optimized by boundary conditions(velocity, acceleration, etc.), and the constraints on these values. Thus arising a need to distinguish the motion profile as type of fitting that can be achieved:

- **Interpolation**: The motion practiced, exactly intersects the given points at a given time.
- **Approximation**: Where by defining a tolerance for error, the agents(in this case robotic manipulator) gets some leverage and may not exactly pass through the given points.

In (Gasparetto et al., 2015) the author argues by stating that ”main drawback of planning a trajectory in the joint space is given by the fact that the execution of a motion planned in the joint space is not so straightforward to predict in the operative space, due to the non-linearities introduced by the direct kinematics”. Moreover, Gasparetto emphasising on the fact that the resulting motion from the planning algorithms must not generate forces and torques at joints not compatible for any given constraint in time, thus reducing mechanical resonance modes. The author highlights that the importance of continuity in the acceleration of the joints in order to produce trajectories with minimal jerk. This is a crucial factor in reducing the vibration induced in the robot. As these vibration can damage the actuators and degrade the tracking performance of the trajectory. Furthermore, low jerk trajectories can be executed faster with higher accuracy.

According to Castro,S (Castro, 2019), in the process of solving the task the first step is to choose the profile of the trajectory space. In general practice there are two type of trajectory spaces that are practiced which are mentioned below:

- **Joint Space Scheme**: A method of path generation in which the path shapes are described in terms of function of joint angles(angle or displacement, depending on the type of joint).
- **Task Space Scheme**: A method of path generation in which the path shapes are described in terms of functions that compute Cartesian position and orientation as function of time. Thus at times, this approach is also referred to as Cartesian-space scheme.

To summarize this in short, it can be said in the case of joint shape the path taken by the end-effector is not smooth but a complicate shape that depends on the kinematics of the manipulator as the constraints are focused on the joints. Whereas in the case of task space the position of the end-effector is significant and a specific spatial shape of the path between waypoints can also be achieved. As Castro,S.(Castro,2019) emphasises, that the biggest drawback of implementing task-space trajectory is that more computation is involved to solve the kinematics.

Castro, S. write in (Castro, 2019) about various trajectories that can be practiced to geometrical interpolate poses over time some of which have been discussed here:

- **Trapezoidal Velocity**: The author refer to these as piecewise trajectories having constant acceleration, creating what is also referred to as S-curve position profile also know as Linier segment with parabolic blend(LSPB). He further states that these are relatively easy to tune and validate for any given position speed or acceleration.
- **Polynomial**: Castro,S. writes about the use of polynomials of various order to traverse between two waypoints. He mentioned that the most common order used in are (a). Cubic - which requires 4 boundary conditions:positional and velocity at both end (b). Quintic - which requires 6 boundary conditions: position, velocity, and acceleration at both ends. Further stating that higher order derivatives can be used for higher-order trajectories. These are considered to be more difficult to validate.
- **Spline**: In (Castro, 2019) the author writes about spline as being a combination of piecewise and polynomials.These are referred to as smooth function that pass through a set of waypoints.

In another writing of Castro, S.(Castro, 2018), the author state that the principle of kinematics where the motion is consider without the force factor, is used to compute the trajectories for any given space scheme. The two main type of kinematic principle to use for computation are forward and inverse kinematics which are further elaborate below:

- **Forwards Kinematics**: As discussed in (Siciliano,2008) this method is used to find the position and orientation of the end-effector relative to the base given the positions of all of the joints and the values of all of the geometric link parameters.
- **Inverse Kinematics**: Also Siciliano, B.(Siciliano,2008) describes this method to find the values of the joint positions given the position and orientation of the end-effector relative to the base and the values of all of the geometric link parameters.

In their work the authors of (Gasparetto et al., 2015) writes that only after a optimality criterion has been set, shall the trajectory be planned. Thus stating some of the most significant optimality criteria as mentioned below:

- minimum execution time criterion
- minimum energy criterion
- minimum jerk criterion
- hybrid criterion (e.g. time-energy, time-jerk, etc.)

II. Literature Review

This section will discuss the shortcoming encountered while implementing these baseline practices to real world problems and how they have been resolved with an objective of creating a better robotics system.

A. Trajectory Planning for Inspection Robot

The authors of (Liu et al., 2012) proposes a time and jerk optimal trajectory planning method, a hybrid optimality algorithm, to find an efficient and smooth trajectory. A newly devised cubic spline is presented tested on a 10-DOF EAST Tokamak inspection robot. Here the authors implemented motion planner to generate a series of waypoint in the task-space suggesting to convert them on into joint space using inverse kinematics. They use a optimization method called SQP(Sequential Quadratic Programming) to solve the mathematical formulation of the optimal problem, which they have discussed in their work.The algorithm, based on the technique of cubic spline interpolation, which can ensure the third order differentiable property of the trajectory and transform the complicated constraints into a simple form. An algebraic method was applied to modulate initial and ending cubic polynomial function so that the velocity and acceleration can be configurable at the initial and ending state. The propose model has been implemented and test in a simulation, thus not considering the issues that may occur for a transition from simulation to real world

B. Trajectory Planning for a fruit picking manipulator

In (Wang et al., 2021) the authors have proposed a planning an optimization strategy to tackle the problem of fruit picking where the agent happens to be robot manipulator mounted on a autonomous chassis base. A brief of their implementation has been shared here. The first step proposed in the implemented was to generate time optimal phase synchronized trajectory. Trajectory with minimum time is built by repeating strategy for eacj link and adjacent waypoints on the path. As the trajectory between start and end on each joint is supposed to be independent, the desired execution time of the shortcut is determined by the slowestjoint. As stated by the authors A trajectory with imposed time between two states is defined by 7 constraints: the imposed time, 3 initial conditions and 3 final conditions. However, The solution proposed used three cubic segments. In this case, the system was defined by 15 parameters (three cubic polynomials) and 13 constraints (3 initial conditions, 3 final conditions, 6 switching conditions and imposed Time).For collision detection an alternative local trajectory checking approach is used in this article. It is on the basis of concept of free configuration space bubbles and the divide-and-conquer algorithm.This algorithm recursively divides the trajectory into two halves and then computes bubbles of free space around the configuration. The collision-free state of the trajectory was devised, by overlapping those bubbles alongside each section, thus the collision-free state of the trajectory can be ensured. .Their approach is capable of real-time implementation due to direct computation without optimization. A continuous collision detection algorithm along trajectories was also presented.

C. Trajectory Planning for wielding Robot with multiple manipulators

In (Liu et al., 2020) the authors have proposed a hybrid criterion based optimal trajectory planning methodology to minimize energy consumption and run time while ensuring the smoothness of collaborative welding robot with multiple manipulators. In their work the trajectory is interpolated in each joint space by means of cubic B-spline and motion time nodes are optimized based on particle swarm optimization(PSO) algorithm. The collaborative welding robot with multiple manipulators two KR16 robots and a modified 3 - DOF KR180 robot .The authors achieved good results as the cubic B-spline curve and PSO combination which is also used in swarm intelligence reduced the energy consumption and run time by great margins.

D. RL for Robotic Manipulators

The authors in (Xia et al., 2023) proposed a reinforcement learning approach for generating trajectories. They utilized a soft-actor critic and Hindsight experience replay algorithm to train a URe5 manipulator. Their approach was tested in a simulated environment. The author practised the trajectory planning algorithm as a MDP. The authors formulated a reward function where the considered the reward function design was also explored. Our design considers quaternion representation for rotation to avoids "gimbal lock". The authors work did not consider moving objects in 3-D spaces which is still domain that requires extensive research.

III. Discussion

The motion and trajectory planning algorithms are a major research topic in the domain of robotics these are part of the wider field of navigation. The major approaches to the field have utilized dynamic programming and genetic algorithm to introduce solution to existing research space. These algorithms are generally devices for a specific robot and under a specific task space. Thus, the solution are constrained to that specific agent and application. With the current advancements in the fields it has become necessary to device a generic methodology for family of agents employed to perform multiple task in a gibe environment.

The current practices have being mostly refrained to simulation environment and these has created a sense of uncertainty as the 'sim2real' transition may hamper future advancement and also the agent are prone to error which can be consider as efficiency in a strict time constraint work environment where there as expected to work with human co-worker or assist them. Working in complex environment require to deliver more complex algorithms¿ The complexity of working with moving object in 3-D domain is a problem where lot of research being practiced

Current practices are focused on solving mathematical extensive computations kinematics and optimality trajectory generation, thus delayed feedback from the agent can be an issue in the real world application.

Current age Reinforcement learning algorithms combined with multi modal-transformer models have proven to increase the efficiency of the training curve and provide with solutions that conventional approach are not able to deliver. Moreover methods of training an agent from teleportation, kinesthetics and demonstration produce more sophisticated and efficient output for any given task.

References

[1] Massaro, M., Lovato, S., Bottin, M., Rosati, G., 2023. An Optimal Control Approach to the Minimum-Time Trajectory Planning of Robotic Manipulators. Robotics 12, 64. https://doi.org/10.3390/robotics12030064.

[2] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo,(2009), Robotics Modelling, Planning and Control, London ,Springer-Verlag London Limited

[3] Sebastian Castro, (2019),Trajectory Planning for Robot Manipulators[blog]. 6 November. Available from https://blogs.mathworks.com/student-lounge/2019/11/06/robot-manipulator-trajectory/[accessed on 11 August 2024]

[4] Luigi Biagiotti, Claudio Melchiorri,(2008),Trajectory Planning for Automatic Machines and Robots, Berlin, Springer-Verlag Berlin Heidelberg

[5] Gasparetto, A., Boscariol, P., Lanzutti, A., Vidoni, R., 2015. Path Planning and Trajectory Planning Algorithms: A General Overview, in: Carbone, G., Gomez-Bravo, F. (Eds.), Motion and Operation Planning of Robotic Systems, Mechanisms and Machine Science. Springer International Publishing, Cham, pp. 3–27. https://doi.org/10.1007/978-3-319-14705-5

[6] Sebastian Castro, (2018), Robot Manipulation, Part 1: Kinematics[blog], 11 April, Available from https://blogs.mathworks.com/student-lounge/2018/04/11/robot-manipulation-part-1-kinematics/[accessed on 12 August 2024]

[7] Liu, H., Lai, X. and Wu, W. (2012) Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. Robotics and Computer-Integrated Manufacturing, 29(2) 309.

[8] Wang, H., Zhao, Q., Li, H. and Zhao, R. (2021) Polynomial-based smooth trajectory planning for fruit-picking robot manipulator. Information Processing in Agriculture, 9(1) 112.

[9] Liu, X., Qiu, C., Zeng, Q., Li, A., Xie, N., 2020. Time-energy Optimal Trajectory Planning for Collaborative Welding Robot with Multiple Manipulators. Procedia Manufacturing 43, 527–534. https://doi.org/10.1016/j.promfg.2020.02.174

[10] Xia, W., Lu, Y., Xu, X., 2023. Application and Evaluation of Soft-Actor Critic Reinforcement Learning in Constrained Trajectory Planning for 6DOF Robotic Manipulators, in: 2023 29th International Conference on Mechatronics and Machine Vision in Practice (M2VIP). Presented at the 2023 29th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), IEEE, Queenstown, New Zealand, pp. 1–6. https://doi.org/10.1109/M2VIP58386.2023.10413400

[11] Liu, S., Liu, P., 2022. Benchmarking and optimization of robot motion planning with motion planning pipeline. Int J Adv Manuf Technol 118, 949–961. https://doi.org/10.1007/s00170-021-07985-5