ISSN No:-2456-2165

Point-to-Point Trajectory Generation under Joint Constraints for Industrial Robots

M.H.P.L Jayarathne and J.A.K.S. Jayasinghe
October 2023

Abstract:- Robotic systems are now used across various domains. Generating a precise, efficient, and time-optimal motion profile is a challenge. A point-to-point trajectory generation method under joint constraints, specifically addressing factors such as joint limits, velocity limits, acceleration limits, and jerk limits, while ensuring the safety and feasibility of the robot's movement is presented. Such a trajectory can be taken as the basis for creating more efficient trajectories by combining the motion profiles of intermediate waypoints where the robot can continue its trajectory without any stops. We comprehensively explore the minimum-time point-to-point trajectory generation problem, by calculating the maximum permissible velocity, acceleration, and jerk for each join.

I. INTRODUCTION

Robots are used in diverse disciplines, seamlessly weaving together the principles of engineering, computer science, and mathematics. In robotics, achieving precise and efficient motion from one point to another is a fundamental challenge. Point-to-point trajectory generation under joint constraints is a vital activity that focuses on controlling the movement of robotic arms or manipulators while adhering to various constraints such as joint limits, velocity limits, acceleration limits, and ensuring that the robot's motion remains both feasible and safe without any collision on the surrounding objects.

Generation of seamless trajectories is a very important aspect for optimum use of industrial robots. In such seamless trajectories, both joint and end-effect or motions follow a set of continuous curves, ensuring displacement, velocity, acceleration, and jerk without any discontinuities throughout the robot's motion. The challenge in seamless trajectory generation for industrial robots revolves around the delicate balance between optimizing the time and preserving the geometric path. Striking this balance is critical, as overly demanding trajectories can lead to torque saturation and path errors, potentially causing severe damage. Conversely, opting for a slower trajectory is a safer approach but may not fully harness the machine's maximum performance potential.

The minimum-time trajectory generation problem has been extensively studied in the literature. There are two main approaches to solve this problem, offline implementations and online implementations. In offline implementations of the minimum-time trajectory generation problem, the trajectory is computed in its entirety before the robot's movement begins. This means that all the calculations, including path planning and trajectory optimization, are performed in advance often based on a predefined map of the environment. In the field of offline trajectory generation methods, Bobrow et al. [1] introduced an iterative procedure that lays the foundation for subsequent developments in this field. Notably, a similar approach was independently presented by Shin and McKay [2]. The efficacy of these offline methods is underlined by their ability to produce trajectories of optimal quality, a fact corroborated by the work of Chen and Desrochers [3]. Their research demonstrated that the trajectories generated by these algorithms indeed represent optimal solutions.

Online implementations of the minimum-time trajectory generation problem involve computing or modifying the trajectory during the robot's motion based on current sensor readings and environmental feedback. Dahl and Nielsen [4] introduced a closed-loop approach for trajectory generation by augmenting the nominal trajectory during motion to address uncertainties regarding the dynamic model and torque limits. Dahl [5] subsequently presented experimental results stemming from this approach.

In our research, we focused on an experimental environment employing a 6-DOF articulated robot manipulator. Within this context, we have developed an offline trajectory generation algorithm aimed at determining the optimal acceleration and velocity profiles to minimize the overall execution time of the robot's path. Our approach leverages principles derived from Robot Dynamics to effectively govern the motion of the robot. We calculate the maximum permissible acceleration for each joint, by taking into account the rated torque values and the inertia along each joint's motion axis. Further, we determine the maximum velocity of each joint by taking the gearbox or servo motor speed limitations. Furthermore, the jerk is obtained by finding the right equilibrium between achieving speed and efficiency while ensuring the robot operates safely and remains in good condition. In any robot trajectory, there will be a set of longest motion joints which will significantly influence the minimumtime trajectory. We plan the robot trajectory such that the longest motion joints will run at the maximum permissible velocity, acceleration, and jerk limits while ensuring that such limits of other joints are not violated. In Section 2, we provide an overview of the background information relevant to our study. This background discussion serves as a foundation for the subsequent portions of our work. In Section 3, robot dynamic modeling is discussed. Section 4 and Section 5 present the methods used for determination of maximum permissible joint parameters for trajectory generation. The proposed trajectory generation methodology is presented in

Section 6. Finally, results and conclusions are given in Sections 7 and 8 respectively.

II. BACKGROUND

This section provides the required theoretical background for the study, and it is structured into two key sub-sections, Mechanical Dynamics and Torque Analysis for Robot Control and Forward and Inverse Kinematics for Robot Control.

A. *Mechanical Dynamics and Torque Analysis for Robot Control*

Considering a robot having *n* joints connected by *n+1* links numbered from 0 to *n* as depicted in Figure 1, its joint vector ***q***can be represented by

$$\boldsymbol{q}= [q_1,...,q_n]^T \qquad (1)$$

where ***q*** is an element of $\mathbf{R}^n$.

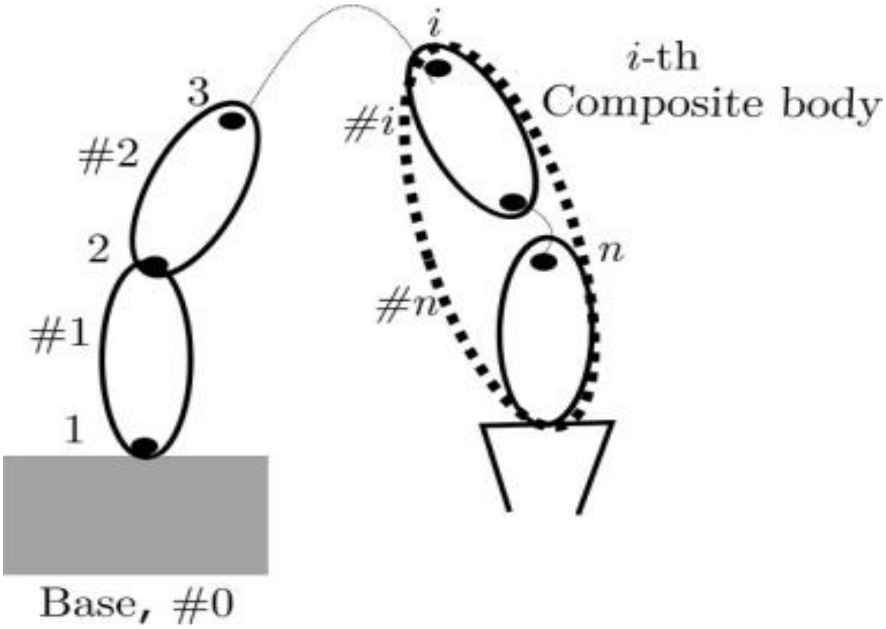


Fig. 1: *n* body serial manipulator

Link 0 represents the fixed”base” and each link is connected to its adjacent link by a one-degree-of-freedom kinematic joint, which can be either a prismatic or revolute joint. Considering that i^{th}body is connected to the *(i - 1)st*body in sequence, the rigid body dynamic model of the manipulator in the joint space can be written as [6],

$$\boldsymbol{B(q)\ddot{q}+ C(q,\dot{q})\dot{q} + Fv\dot{q} + Fsign(\dot{q}) + g(q) = \tau} \qquad (2)$$

where, ***B*(*q*)** is the *n* × *n* inertia matrix, ***C*(*q,q˙*)*q˙*** is the vector of Coriolis and centrifugal forces, ***Fvq˙***, is the vector of viscous friction, ***Fsign(q˙)***is the vector of static friction, ***g(q)***is the vector of gravitational forces and ***τ***is the vector of joint torques. For a 6-DOF articulated robot, *n* = 6, and all joints are revolute joints.

In our proposed methodology, the assessment of the location of the center of gravity of the mass and the moments of inertia relevant to the configuration of the robot play a pivotal role. The estimation of the location of the center of gravity and movement of inertia can be done by geometric analysis or by using solid modeling software used for CAD design. Figure 2 depicts three different configurations of the 6-DOF articulated robot used for the experimental setup. Table 1 depicts the location of the center of gravity with respect to each joint center for configuration A. Further, Table 2 shows the moment of inertia around each joint’s rotational axis for the three different configurations shown in Figure 2. To calculate critical forces such as the Coriolis and centrifugal forces, as well as the viscous and static friction forces, the elements of ***C(q,q˙)*** and ***Fv***must be known for given joint velocities which present a significant challenge. To address this inherent complexity, as a practical approach, a percentage of the rated torque value can be set off for the Coriolis and centrifugal forces, as well as for the viscous and static friction forces.

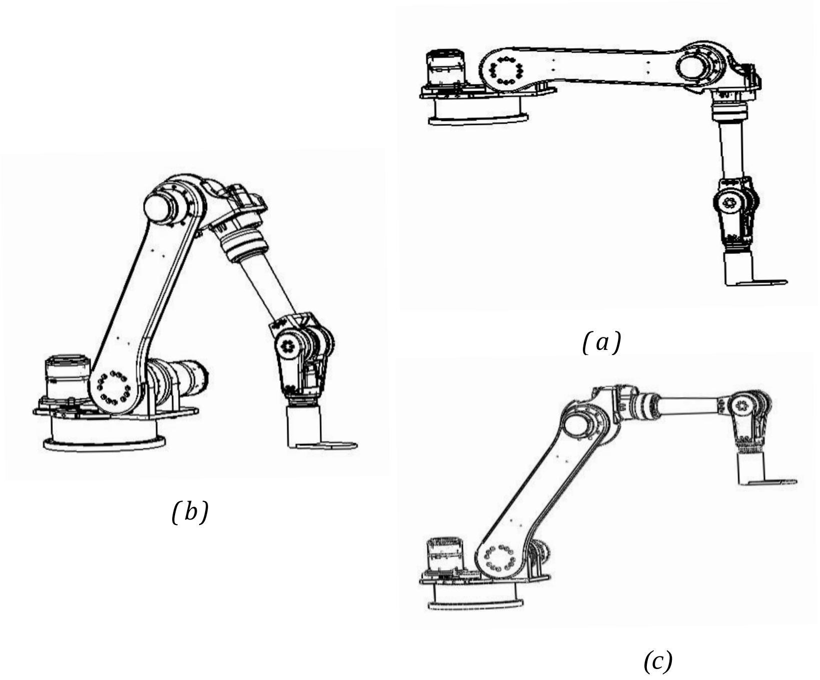


Fig. 2: Three different configurations of the 6-DOF articulated robot

Table 1: Center of gravities for configuration A

Joint	Location of center of gravity (*mm*)		
	C_x	C_y	C_z
J_1	349.64	−23.05	−7.85
J_2	538.91	−177.56	−41.21
J_3	340.60	−56.87	−29.97
J_4	−0.88	−7.00	322.54
J_5	140.70	2.22	46.36
J_6	−26.37	10.93	79.96

Table 2: Moments of inertia around each joint

Config.	Moment of Inertia ($Kgmm^2$) × 10^6					
	J_1	J_2	J_3	J_4	J_5	J_6
A	11.75	9.53	2.11	0.01	0.07	0.01
B	5.16	6.10	2.05	0.02	0.07	0.01
C	9.02	10.82	1.71	0.07	0.07	0.01

B. *Forward and Inverse Kinematics for Robot Control*

Forward and inverse kinematics are used in robotics to determine the positional and angular relationships of the robot. Further, these techniques play a pivotal role in robot control, motion planning, and task execution.

➢ *Forward Kinematics*

In forward kinematics, end-effector's position and orientation in relation to its base are calculated based on the current joint variable values. Based on the D-H parameters of the Robot, a set of coordinate transform matrices are formed for each joint first. By multiplying each coordinate transform matrix sequentially, one can easily find the end-effector position and orientation for a given set of joint variables [7].

➢ *Inverse Kinematics*

In inverse kinematics, values for joint variables are obtained for a given end-effector location and orientation. The specific method for solving the inverse kinematics problem can vary based on the robot's kinematic structure. It can be solved by analytical methods, numerical methods or using optimization techniques [8].

III. DYNAMIC MODELING

As the next step of the proposed methodology, we determine the torque required to overcome the gravitational forces using the computational technique proposed by S. K. Saha [9] involving forward and backward recursion.

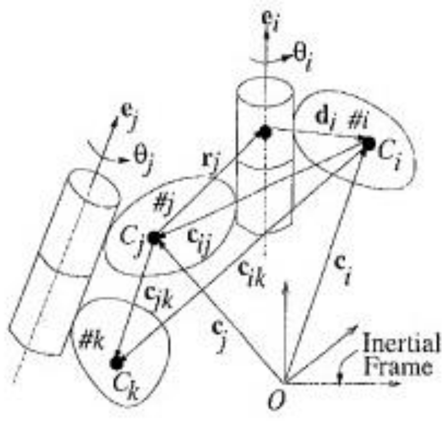


Fig. 3: Dynamic modeling system

With reference to the serial manipulator shown in Figure 1, two adjacent joints, namely *i*and *j*, are shown in Figure 3. The center of gravity of each link connected to the above two adjacent joints is denoted by C_i, C_j, and C_k. The inertial reference frame is represented by *O*. The vectors from the inertial frame *O* to the center of gravity of each link are denoted by $\boldsymbol{c}_i$, $\boldsymbol{c}_j$, and $\boldsymbol{c}_k$. The $\boldsymbol{r}_j$represents the vector from the center of gravity of $C_j$to the joint *i*. Conversely, the vector from joint *i*to the center of gravity of C_i is expressed as $\boldsymbol{d}_i$. $\boldsymbol{c}_{ij}$is the vector between $C_i$and C_j.

C. Inverse Dynamics Problem

Inverse dynamics is the problem of finding the forces/torques required to produce a given acceleration in a rigid-body system. Inverse dynamics calculations are used in motion control systems, trajectory design, and optimization. The inverse dynamics problem can be effectively solved using two recursive steps as shown below.

- *Forward Recursion*

Forward recursion proceeds from the base of the robot's kinematic chain and moves toward the end-effector or the tip of the chain. As described in [9], the forward recursion is formulated as,

$$\begin{aligned} \boldsymbol{v}1 &= \boldsymbol{p}1\dot{\boldsymbol{q}}1 \\ \boldsymbol{v}2 &= \boldsymbol{p}2\dot{\boldsymbol{q}}2 \ +\boldsymbol{v}1 \\ \boldsymbol{v}_3 &= \boldsymbol{p}_3\dot{\boldsymbol{q}}_3 + \boldsymbol{v}_2 \\ \boldsymbol{v}n &= \boldsymbol{p}n\dot{\boldsymbol{q}}n + \boldsymbol{v}n-1 \end{aligned} \tag{3}$$

$$\boldsymbol{\xi}_1 = \boldsymbol{p}_1\ddot{\boldsymbol{q}}_1 + \boldsymbol{\Omega}_1\boldsymbol{p}_1\dot{\boldsymbol{q}}_1 + \boldsymbol{\rho}$$

$$\boldsymbol{\xi}2 = \boldsymbol{p}2\ddot{\boldsymbol{q}}2 + \boldsymbol{\Omega}2\boldsymbol{p}2\dot{\boldsymbol{q}}2 + \boldsymbol{B}21\boldsymbol{\xi}1 + \dot{\boldsymbol{B}}21\boldsymbol{v}1 \tag{4}$$

$$\boldsymbol{\xi}n = \boldsymbol{p}n\ddot{\boldsymbol{q}}n + \boldsymbol{\Omega}n\boldsymbol{p}n\dot{\boldsymbol{q}}n + \boldsymbol{B}_n(n-1)\boldsymbol{\xi}\boldsymbol{n}-1 + \dot{\boldsymbol{B}}_n(n-1)\boldsymbol{v}n-1$$

where$\boldsymbol{v}_i$, $\boldsymbol{\xi}_i$and $\boldsymbol{p}_i$are 6-dimensional vectors and $\boldsymbol{B}_{ij}$ and $\boldsymbol{\Omega}_i$are 6 × 6 matrices.

In the above Equation (3) and Equation (4), $\boldsymbol{p}_i$ represents the joint rate propagation vector defined as follows:

for revolute joints:

$$\boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{ei} \\ \boldsymbol{ei} \ \times \boldsymbol{di} \end{bmatrix} \tag{5}$$

for prismatic joints:

$$\boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{ei} \ \times \boldsymbol{di} \end{bmatrix} \tag{6}$$

Further, $\boldsymbol{e}_i$is a unit vector along the axis of motion and axis of rotation for the prismatic and revolute joints respectively.

$\boldsymbol{B}_{ij}$is the twist propagation matrix, where

$$\boldsymbol{B}_{ij} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{c}_{ij} \times \boldsymbol{1} & \boldsymbol{1} \end{bmatrix} \tag{7}$$

The vector $\boldsymbol{c}_{ij} = -\boldsymbol{d}_i - \boldsymbol{r}_i$and $\boldsymbol{c}_{ij}\times$ **1** being the cross product tensor associated with vector $\boldsymbol{c}_{ij}$.

$\boldsymbol{\Omega}_i$is the matrix of angular velocity, where

$$\boldsymbol{\Omega}_i = \begin{bmatrix} \boldsymbol{\omega}_i \times \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\omega}_i \times \boldsymbol{1} \end{bmatrix} \tag{8}$$

In the above Equation, $\boldsymbol{\omega}_i$represents the 3-dimensional vectors of angular velocity. In Equation (4), $\boldsymbol{\rho}$represents the effect of gravity and $\boldsymbol{\rho}$is defined as [9],

$$\boldsymbol{\rho} = [\boldsymbol{0}^T,(-\boldsymbol{g})^T]^T \tag{9}$$

When the robot commences its trajectory $\boldsymbol{v}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$are zero. Hence, Equation (4) gets a simplified form as follows:

$$\begin{aligned} \xi_1 &= \boldsymbol{\rho} \\ \xi_2 &= \boldsymbol{B}_{21}\xi_1 \\ \xi n &= \boldsymbol{B}n(n-1)\xi n-1 \end{aligned} \tag{10}$$

- *Backward Recursion*

Backward recursion proceeds from the end-effector or the tip of the chain towards the base of the robot. As described in [9], the backward recursion is formulated as,

$$\gamma n = \boldsymbol{M}n\boldsymbol{\xi}n + \dot{\boldsymbol{M}}n\boldsymbol{v}n$$

$$\gamma n\text{-}1 = \boldsymbol{M}n\text{-}1\boldsymbol{\xi}n\text{-}1 + \dot{\boldsymbol{M}}n\text{-}1\boldsymbol{v}n\text{-}1 + \boldsymbol{B}^T{}_{n(n-1)}\gamma n \tag{11}$$

$$\gamma 1 = \boldsymbol{M}1\boldsymbol{\xi}1 + \dot{\boldsymbol{M}}1\boldsymbol{v}1 + \boldsymbol{B}^T{}_{21}\gamma 2$$

$$\tau n = \boldsymbol{p}^T{}_n\gamma n$$

$$\tau_n-1 = \boldsymbol{p}^T{}_n-{}_1\gamma_n-1 \tag{12}$$

$$\tau 1 = \boldsymbol{p}^T{}_1\gamma 1$$

where γ_i is a 6-dimensional vector. The 6×6 matrix, $\boldsymbol{M}_i$ is constructed as follows using the mass of each link m_i and inertial tensor $\boldsymbol{I}_i$:

$$\boldsymbol{M}_i = \begin{bmatrix} \boldsymbol{I}_i & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{m}_i\boldsymbol{1} \end{bmatrix}` \tag{13}$$

The matrix **1** represents a 3 × 3 identity matrix, and **0** represents a 3 × 3 matrix of zeros.

When the robot commences its trajectory, Equation (12) and Equation (13) get a simplified form as follows:

$$\gamma n = \boldsymbol{M} n \xi n$$

$$\gamma n\text{-}1 = \boldsymbol{M} n\text{-}1 \xi n\text{-}1 + \boldsymbol{B}^{T}{}_{n(n-1)} \gamma n \tag{14}$$

$$\gamma 1 = \boldsymbol{M} 1 \xi 1 + \boldsymbol{B}^{T}{}_{21} \gamma 2$$

$$\tau n = \boldsymbol{p}^{T}{}_{n} \gamma n$$

$$\tau_{n} - 1 = \boldsymbol{p}^{T}{}_{n} -{}_{1} \gamma_{n} - 1 \tag{15}$$

$$\tau 1 = \boldsymbol{p}^{T}{}_{1} \gamma 1$$

By using Equations (10), (14), and (15) we can determine the torque requirement to overcome the effect of the gravity for each individual joint.

IV. MAXIMUM PERMISSIBLE MOTION PARAMETERS

In this section, we calculate or estimate the maximum possible values for the joint velocity, acceleration, and jerk.

A. *Maximum Permissible Velocity*

In order to improve the reliability of the robot, it must operate below the speed constraints imposed by the gearbox or servo motor. By following steps, the upper limit for joint velocities can be estimated:

- Identify the rated speed (in RPM) of each servo motor and the reduction ratio of each gearbox.
- Determine the speed of the output shaft of the gearbox when the servo motor is operated at its rated speed.
- Identify the maximum permissible speed (in RPM) of the gearbox output shaft.
- Select the minimum from items 2 and 3 above as the maximum permissible RPM.
- Convert the maximum permissible RPM into maximum permissible joint velocity expressed as rad/sec.

Table 3 shows the maximum permissible velocity obtained for the experimental robot using the specifications of the motors and gearboxes.

Table 3: Maximum velocity ($\dot{q}_{i,max}$) for each joint

Joint	**Max Velocity (*rad/sec*)**
J_1	3.14
J_2	3.14
J_3	3.14
J_4	3.14
J_5	3.14
J_6	3.14

B. *Maximum Permissible Acceleration*

The gravitational torque acting on the robot when it commences its motion has been computed in Section 3.1. As the elements of C(q,$\dot{q}$), Fvand Fare unknown, we can leave a margin (say 20%) from the available motor/gearbox torque to overcome the forces such as the Coriolis and centrifugal forces, as well as the viscous and static friction forces. This designated torque percentage is henceforth referred to as the "Margin Torque" h.

Under these conditions, when the robot commences its motion, Equation (2) can be presented as:

$$B(q)\ddot{q} + h + g(q) = \tau \tag{16}$$

Hence, the maximum permissible acceleration is given by

$$\ddot{q}max = B-1(q)(\tau - h - g(q)) \tag{17}$$

Based on this concept, one can determine the maximum permissible acceleration for different robot configurations. Table 4 depicts the above limits for three different configurations of the experimental setup depicted in Figure 2.

Table 4: Maximum acceleration ($\ddot{q}_{i,max}$) for each joint

Joint	$\tau(Nm)$	**Max Acceleration(*rad/sec*^{2})**		
		Config. A	**Config. B**	**Config. C**
J_1	433	21.52	49.01	28.03
J_2	333	4.21	18.80	5.63
J_3	157	66.39	95.33	127.04
J_4	113	5760.00	3275.72	998.75
J_5	54	655.73	655.73	655.73
J_6	54	4590.00	4590.00	4590.00

C. *Maximum Permissible Jerk*

The maximum jerk (rate of change of acceleration) that a robot can experience while following a smooth trajectory depends on different things, including the type of robot, the specific application, payload, end-effector and tooling, motion profile, control system, regulations and standards, system dynamics and safety considerations. These limits can vary from robot to robot and from one job to another, but it’s always about balancing how fast the robot can move against how safe it needs to be.

It’s crucial to take into account all these factors and, if needed, perform tests and analyses to establish the allowable jerk limit for a specific robot and its intended use. The aim is to find the right equilibrium between achieving speed and efficiency while ensuring the robot operates safely and remains in good condition. In practice, there is no theoretical maximum jerk value, and it should be determined on a case-by-case basis taking into account robot reliability, operational safety, and time to achieve the constant acceleration phase, etc.

Table 5 depicts the jerk values needed to achieve the constant acceleration phase within 0.1 Sec.

Table 5: Maximum jerk ($J_{i,max}$) for each joint

Joint	**Max Jerk(*rad/sec*^{3})**		
	Config. A	**Config. B**	**Config. C**
J_1	215.20	490.10	280.30
J_2	42.10	188.00	56.90
J_3	663.90	953.30	1270.40
J_4	57600.00	32757.50	9987.50
J_5	6557.30	6557.30	6557.30
J_6	45900.00	45900.00	45900.00

V. VELOCITY, ACCELERATION, AND JERK SCALE FACTORS

Trajectory generation involves the design and calculation of a path that the end-effector should follow to achieve a specific task while considering various constraints and objectives. Once the maximum permissible velocity, acceleration, jerk, and joint variables have been established, the next crucial step is to identify the joint with the maximum displacement.

After identifying the maximum displacement joint, we can plan the motion of that particular joint using the maximum permissible velocity, acceleration, and jerk obtained above (Tables 3, 4 and 5) without violating the relevant limits for other joints. In case of violations, the velocity, acceleration, and jerk used for planning the motion of the maximum displacement joint can be scaled down as follows. Let’s define the Displacement Scale Factor for each joint DSFias.

$$DSF_i = \frac{q_i}{q_j} \tag{18}$$

where*j* is the maximum displacement joint. Further, we define the Velocity Scale Factor for each joint VSF_i as

$$VSF_i = \begin{cases} \frac{\dot{q}_{i,max}}{\dot{q}_{j,max} \times DSF_i} & if\ \ \dot{q}_{j,max} \times DSF_i > \dot{q}_{i,max} \\ 1 & Otherwise \end{cases} \tag{19}$$

and define the operational Velocity Scale Factor VSF_oas the minimum VSF_i. Similarly, we define the Acceleration Scale Factor for each joint ASF_ias

$$ASF_i = \begin{cases} \frac{q^{\cdot\cdot}{}_{i,max}}{q^{\cdot\cdot}{}_{j,max} \times DSF_i} & if\ \ q^{\cdot\cdot}{}_{j,max} \times DSF_i > q^{\cdot\cdot}{}_{i,max} \\ 1 & Otherwise \end{cases} \tag{20}$$

and define the operational Acceleration Scale Factor ASF_oas the minimum ASF_i. We define the Jerk ScaleFactor for each joint JSF_ias

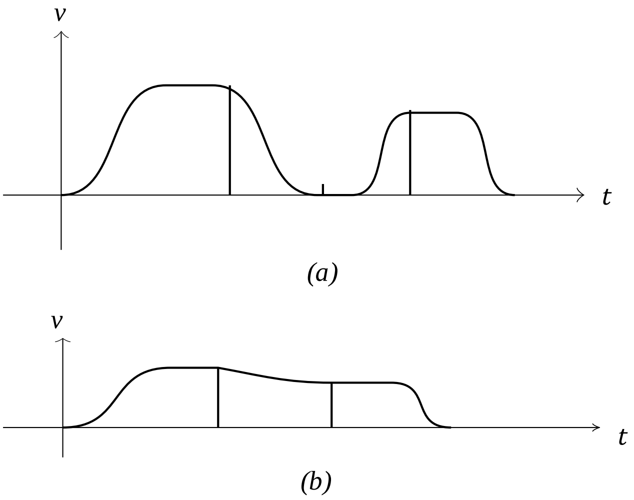
$$JSF_i = \begin{cases} \frac{q^{\cdot\cdot\cdot}{}_{i,max}}{q^{\cdot\cdot\cdot}{}_{j,max} \times DSF_i} & if\ \ q^{\cdot\cdot\cdot}{}_{j,max} \times DSF_i > q^{\cdot\cdot\cdot}{}_{i,max} \\ 1 & Otherwise \end{cases} \tag{21}$$

and define the operational Jerk Scale Factor JSF_oasthe minimum JSF_i.

VI. PROPOSED METHODOLOGY

The flowchart shown in Figure 5 outlines the trajectory generation methodology proposed in this paper. It encompasses various key stages and decision points in the process, ensuring that the robot follows a predefined path while adhering to safety and performance constraints.

The process begins with the determination of the maximum permissible velocity for each joint. Moving on from one waypoint to the next, trajectory segments are created and joint values at the ends of each trajectory segment are obtained. For each trajectory segment, the maximum permissible acceleration and jerk are then determined. Next step is to identify the joint with the maximum displacement for each trajectory segment. After that, the velocity, acceleration, and jerk limits are established using the maximum displacement joint, and a motion profile is created for the trajectory segment. The motion profile is described by indicating the time duration allowed for different segments of the velocity vs. time plot of the trajectory such that the velocity at each waypoint is zero. After that, the system checks for any violations of velocity, acceleration, and jerk constraints of the other joints. If violations are detected, the velocity, acceleration, and jerk are scaled down by Operational Velocity Scale Factor VSFo, Operational Acceleration Scale Factor ASFoand Operational Jerk Scale Factor JSForespectively to regenerate the motion profile. Then, the next trajectory segment is generated by repeating above steps until the whole trajectory is completed. The point-to-point trajectory formed by such a technique can be further improved if the trajectory can be continued without stopping at intermediate waypoints. Figure 4, demonstrates this concept and the formal procedure for such improvements will be published in a forthcoming publication.



(a)

(b)

Fig. 4: Formation of a continuous trajectory from a point-to-point trajectory (a) Point-to-point trajectory, (b) Continuous trajectory

VII. RESULTS

We conducted an experiment using the configuration C in Figure 2 to determine the time required to move along 8 waypoints given in Table 6 using the proposed trajectory generation method.

As a reference trajectory, we have executed the trajectory using minimum values of Velocity, Acceleration, and Jerk (The respective values extracted from Table 3, Table 4, and Table 5 are 3.14, 5.69, and 56.90). The reference trajectory execution time is 25.16 seconds while the trajectory generated by the proposed method needs 7.81 seconds to execute. Therefore, the improvement with respect to the reference method is 3.22 times. A similar trend was observed for the other trajectories as well.

VIII. CONCLUSION

In robotics, achieving precise and efficient motion from one point to another is a challenge. This process involves planning trajectories while adhering to various constraints, including joint limits, velocity limits, and acceleration limits, The main objective is to create seamless trajectories that maintain continuity in displacement, velocity, acceleration, and jerk throughout the robot's motion. To address this challenge, we presented a comprehensive trajectory generation methodology, which includes:

- Determining the maximum permissible parameters for velocity, acceleration, and jerk.
- Identifying the maximum displacement joint within the robot's kinematic structure, which significantly influences trajectory generation.
- Defining scale factors to ensure other joints' velocity, acceleration, and jerk limits are not violated.
- Implementing an iterative approach to complete the trajectory by joining segments between waypoints.

We emphasize the importance of striking a balance between optimizing execution time while preserving the geometric path without torque saturation and path errors. We note that overly cautious trajectories may not fully harness the robot's performance potential.

REFERENCES

[1.] Bobrow J.E., Dubowsky S., and Gibson J.S., "Time-Optimal Control of Robotic Manipulators Along Specified Paths", The International Journal of Robotics Research, 4(3), pp. 3–17, 1985.

[2.] Shin K.G. and McKay N.D., "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints", IEEE Transactions on Automatic Control, 30(6), pp. 531-541, 1985.

[3.] Chen Y. and Desrochers A.A., "Structure of Minimum-time Control Law for Robotic Manipulator with Constrained Path", 1989 International Conference on Robotics and Automation, Vol. 2, pp. 971-976, 1989.

[4.] Dahl O. and Nielsen L., "Torque Limited Path Following by On-line Trajectory Time Scaling" IEEE Transactions on Robotics and Automation, 6(5), pp. 554-561, 1990.

[5.] Dahl O., "Path Constrained Robot Control with Limited Torque — Experimental Evaluation", IEEE Transactions on Robotics and Automation, 10(5), pp. 658-669, 1994.

[6.] GianlucaAntonelli, Stefano Chiaverini, Marco Palladino, Gian Paolo Gerio, and Gerardo Renga, "Joint Space Point-to-point Motion Planning For Robots. An Industrial Implementation", IFAC Proceedings, 38(1), pp. 187-192, 2005.

[7.] Daniel Constantin, Marin Lupoae, C˘at˘alinBaciu, and Dan-IlieBuliga, "Forward Kinematic Analysis of an Industrial Robot", International Conference on Mechanical Engineering, pp. 90-95, 2015.

[8.] Abdullah Aamir Hayat, Ratan Sadanand O. M., and Subir. K. Saha, "Robot Manipulation through Inverse Kinematics", Proceedings of the 2015 Conference on Advances in Robotics, pp. 48:1-48:6, 2015.

[9.] S. K. Saha,"Dynamics of Serial Multibody Systems Using the Decoupled Natural Orthogonal Complement Matrices", 66(4), Journal of Applied Mechanics, pp. 986-996, 1999.

Table 6: Waypoints of a sample trajectory

Joint	**Joint Angels*(rad)***								
	Initial WP	**WP1**	**WP2**	**WP3**	**WP4**	**WP5**	**WP6**	**WP7**	**WP8**
J_1	0.00	0.00	0.34	0.00	0.00	−0.69	0.00	0.00	0.34
J_2	1.57	−0.52	0.00	−0.03	0.03	0.00	−0.01	0.01	0.00
J_3	0.00	1.04	0.00	0.20	−0.20	0.00	0.13	−0.13	0.00
J_4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
J_5	−1.57	1.57	0.00	−0.23	0.23	0.00	−0.14	0.14	0.00
J_6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

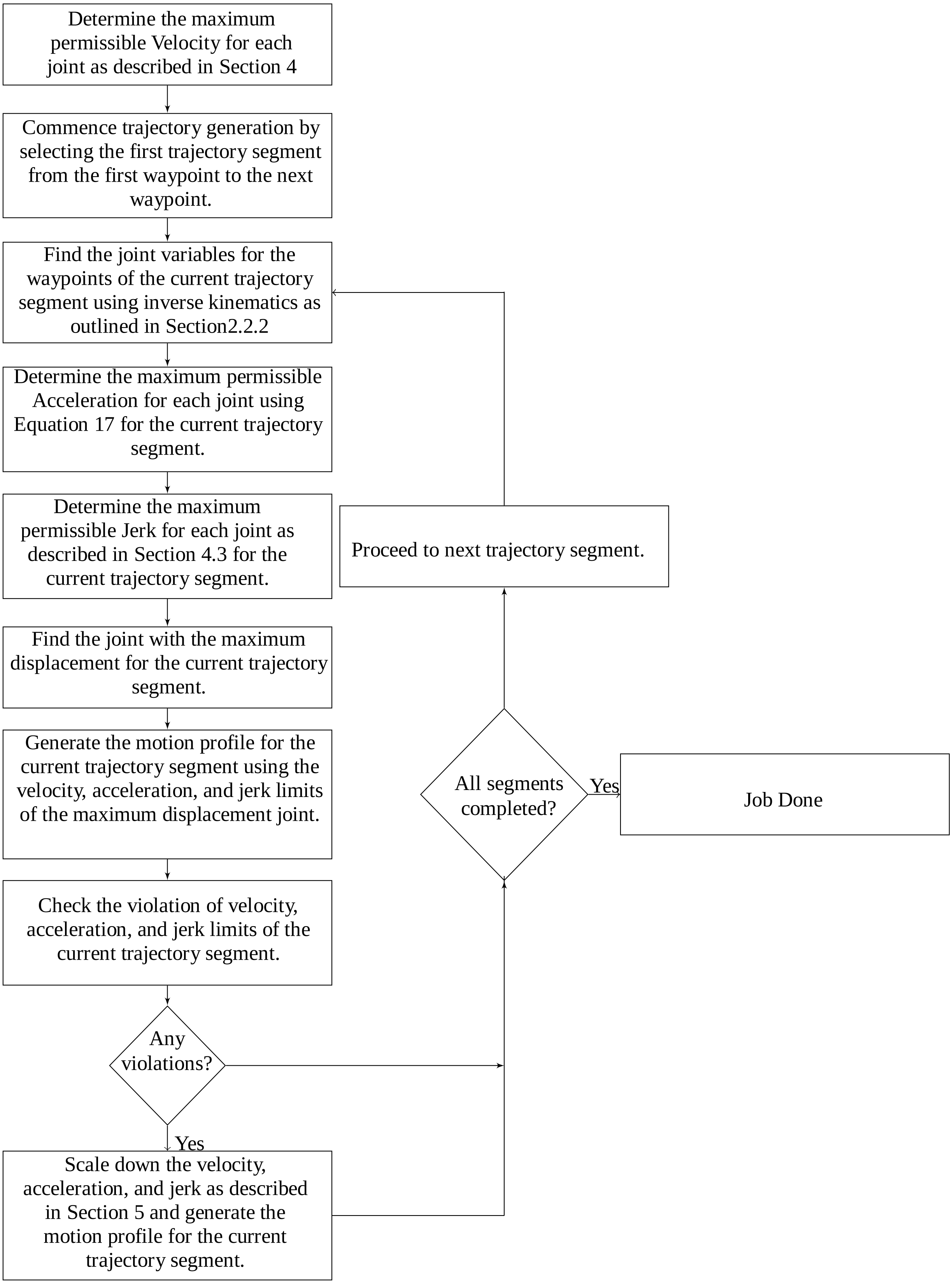

Determine the maximum permissible Velocity for each joint as described in Section 4
Commence trajectory generation by selecting the first trajectory segment from the first waypoint to the next waypoint.
Find the joint variables for the waypoints of the current trajectory segment using inverse kinematics as outlined in Section2.2.2
Determine the maximum permissible Acceleration for each joint using Equation 17 for the current trajectory segment.
Determine the maximum permissible Jerk for each joint as described in Section 4.3 for the current trajectory segment.
Find the joint with the maximum displacement for the current trajectory segment.
Generate the motion profile for the current trajectory segment using the velocity, acceleration, and jerk limits of the maximum displacement joint.
Check the violation of velocity, acceleration, and jerk limits of the current trajectory segment.
Any violations?
Yes
Scale down the velocity, acceleration, and jerk as described in Section 5 and generate the motion profile for the current trajectory segment.
All segments completed?
Yes
Job Done
Proceed to next trajectory segment.

Fig. 5: Flow chart of the proposed trajectory generation method