Control in Robotics
Part 1: Preliminaries

Mohammad Motaharifar

Faculty of Engineering
University of Isfahan

Spring 2022

Table of contents

History

- 1940s: Manipulators for handling radio-active materials
- 1960s: First generation of industrial robots by Joe Engelberger and George Devol (Unimation)

- 1970s: PUMA

What is a Robot?

- A robot is a mechanical or virtual artificial agent, usually an electro-mechanical system, which, by its appearance or movements, conveys a sense that it has intent or agency of its own.
- A typical robot will have several, though not necessarily all of the following properties:
  - Is not natural, and it has been artificially created.
  - Can sense its environment.
  - Can manipulate things in its environment.
  - Has some degree of intelligence.
  - Is programmable.
  - Can move with one or more axes of rotation or translation.
  - Appears to have intent or agency.

What is a Robot?

- There is no one definition of robot which satisfies everyone, and many people have written their own.
- (According to ISO8373) "An automatically controlled, reprogrammable, multipurpose, manipu- lator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications."
- (According to Joe Engelberger, a pioneer in industrial robotics) "I cant define a robot, but I know one when I see one."
- (According to Cambridge Advanced Learners Dictionary)"A machine used to perform jobs automatically, which is controlled by a computer."

Robot Components

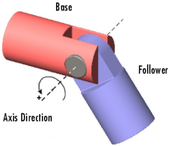
- A mechanism or a robotic manipulator is usually build from a number of links, connected to each other, to the ground or a movable base, by different types of joints.
- The individual rigid bodies that make up a robot is called the links.
- In industrial robots the rigidity of the links contribute significantly to the precision and performance of the robots, and usually in the design of the links, the rigidity is a vital requirement.
- However, in applications such as space robotics, or cable driven manipulators, due to the limitations and type of application, the links are constructed from flexible elements. Such robots are usually called flexible link manipulators.

Robot Components

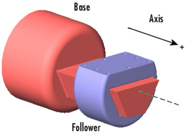
- In this course links are treated as rigid bodies for most of the manipulators.
- The assumption of the rigid bodies makes the analysis of robot manipulators much easier to understand.
- From the kinematic point of view, a single link can be defined as an assembly of members connected to each other, such that no relative motion can occur among them.
- In robots, the links are connected in pairs, and the connective element between two links is called a joint.
- A joint provides some physical constraints on the relative motion between the two connecting members.

Types of Joints

- **Revolute Joints (R)**: permit rotation about one axis between two paired elements.

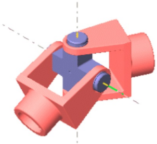


- **Prismatic Joints (P)**: permit linear motion along one axis between two paired elements.

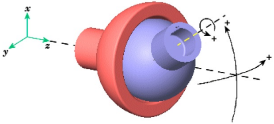


Types of Joints

- **Universal Joints (U)**: permit rotation about two independent axis.

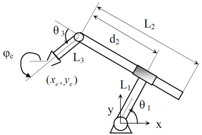


- **Spherical Joints (S)**: permits free rotation of one element with respect to other element about the center of a sphere in all possible three directions.

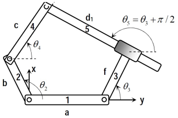


Kinematic Chain

- A kinematic chain is an assembly of links that are connected by joints.
- **Open-loop chain**: every link is connected to its pair by only one path



- **Closed-loop chain**: every link in a kinematic chain is connected to every other link by at least two distinct paths.



Robot Degrees-of-Freedom (DOF)

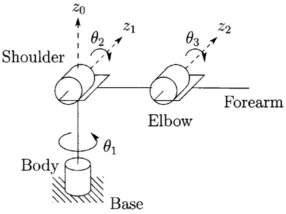
- The degrees of freedom (DOF) of a robot the number of independent position variables which would have to be specified in order to locate all parts of the robot.
- In the case of typical industrial Robots, because a serial Robot manipulator is usually an open kinematic chain, and because each joint position is usually defined with a single variable, the number of joints = number of DOF of the Robot
- The above formula is not necessary true for a closed closed-loop kinematic chain (Parallel Robot)
- 6 DOF Robot: Robot Hand can move freely in operational space (Hand Coordinates System), along 3 translational axes and around 3 rotational axes
- An 8-DOF Robot: This Robot has 8 DOF in actuator joint space and can have 6 motions in Robot operational space. This is a redundant Robot.

Robot Classification

- Classification of robots may be done according to various measures.
- An impoetant means to classify robots is based on their kinematic structure.
- A serial robot is constructed from an open-loop kinematic chain, while a parallel robot is made up from a number of closed-loop chains.
- Moreover, a hybrid robot consists of both open- and closed-loop chains.

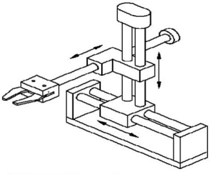
Serial Robots

- Serial robots are the most common robots used in industrial applications.
- Often they have a serial chain of rigid links, connected by joints.
- An Articulated robot (RRR) consists of three revolute joints

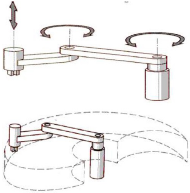


Serial Robots

- A cartesian (PPP) robot consists of three prismatic joints

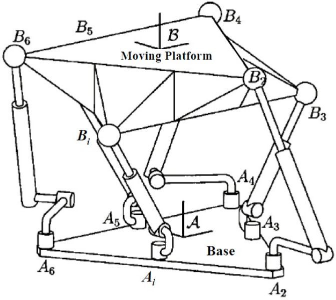



- A SCARA (RRP) robot consists of two revolute and one prismatic joints.

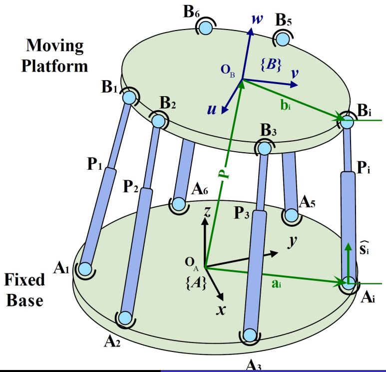
Parallel Robots

- A parallel robot is a mechanism whose moving platform is linked to the base by several independent kinematic chains.
- A parallel robot, consists of a fxed base platform, connected to a moving platform by means of a number of limbs.
- Parallel robots have high structural stiffness, since the moving platform is supported by several limbs.



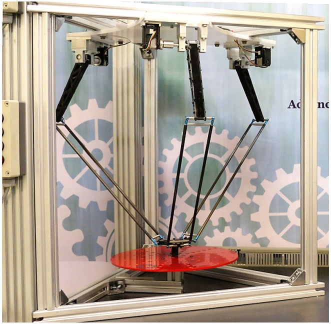
Parallel Robots

- Stewart-Gough Platform, as the most celebrated parallel robot, has six independently actuated limbs, where the lengths of the legs are changed to move the platform to a desire position and orientation.



Parallel Robots

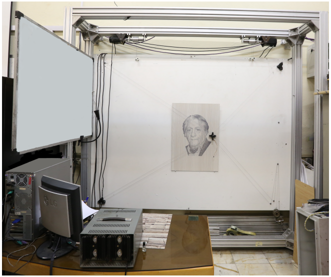
- A Delta robot is a type of parallel robot which consists of three arms connected to universal joints at the base. The key design feature is the use of parallelogram in tho arms, whichmaintains tho orientation of the end effector.



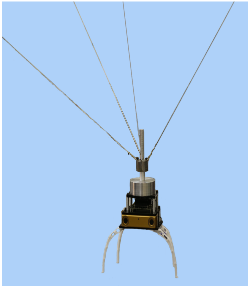
https://aras.kntu.ac.ir/kntu-delta-robot/

Parallel Robots

- In a cable-driven redundant parallel manipulator (CDRPM), the linear actuators of parallel manipulators are replaced with electrical powered cable drivers, which leads immediately to a larger workspace.



(a)

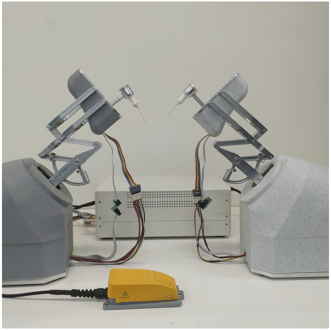


(b)

http://krobot.ir/

Hybrid Robots

- Hybrid Robots utilize the structure of both series and parallel robots. For example, a paralellogram mechanism provides remote center of motion (RCM) in minimally invasive surgeris (MIS).



https://aras.kntu.ac.ir/arash-asist/

Associated Problems

- Position and Orientation: Location of objects in three-dimensional space (3D) space.
- Kinematics: The study of motion without regard to the forces that produce that motion.
- Forward Kinematics: Computing the position and orientation of the end-effector.
- Inverse Kinematics: Given a position and orientation of the end-effector, calculate all possible sets of joint angles that attain such position/orientation.
- Statics: The study of forces and moments apart from motion.
- Dynamics: The study of motion including the forces that produce that motion.

Associated Problems

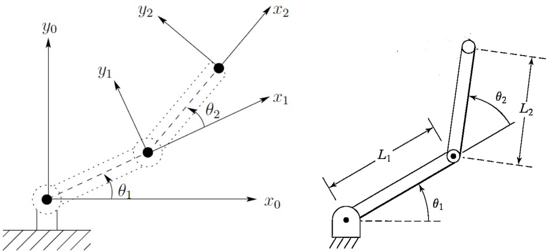
- Path planning: Providing a geometric description of collision free path for robot motion, but not dynamic aspects of motion
- Trajectory planing: specifing joint motions as smooth functions of time, so as to cause the manipulator to move in a coordinated and smooth manner.
- Position Control: Finding an appropriate algorithm to compute forces or torques which will cause a desired motion
- Force Control: The ability of a manipulator to control forces of contacts.

Kinematics

- In this part, kinematics analysis of robot manipulators is briefly explained.
- Kinematics analysis refers to the study of the geometry of motion in a robot, without considering the forces and torques that causes the motion.
- In this analysis the relation between the geometrical parameters of the manipulator with the final motion of the end effector is derived and analyzed.

Forward Kinematics of Simple Serial robots

- A 2DOF RR robot is shown here.
- It is customary to establish a fixed coordinate system, called the world or base frame to which all objects including the manipulator are referenced.



(c) (d)

Forward Kinematics of Simple Serial robots

- The coordinates (x, y) of the end-effector are expressed with regard to the base frame as

$$
\begin{aligned}
x &= x_2 = l_1 c_1 + l_2 c_{12} \\
y &= y_2 = l_1 s_1 + l_2 s_{12}
\end{aligned}
$$

where

$$
\begin{aligned}
c_1 &= \cos\theta_1 \\
s_1 &= \sin\theta_1 \\
c_{12} &= \cos(\theta_1 + \theta_2) \\
s_{12} &= \sin(\theta_1 + \theta_2)
\end{aligned}
$$

Definition of Jacobian Matrix

- The Jacobian is a multidimensional derivative. For example:

$$\begin{aligned} y_1 &= f_1(x_1, x_2, ..., x_6) \\ y_2 &= f_2(x_1, x_2, ..., x_6) \\ &\vdots \\ y_6 &= f_6(x_1, x_2, ..., x_6) \end{aligned}$$

- Differentials of y_i:

$$\begin{aligned} \delta y_1 &= \frac{\partial f_1}{\partial x_1}\delta x_1 + \frac{\partial f_1}{\partial x_2}\delta x_2 + ... + \frac{\partial f_1}{\partial x_6}\delta x_6 \\ \delta y_2 &= \frac{\partial f_2}{\partial x_1}\delta x_1 + \frac{\partial f_2}{\partial x_2}\delta x_2 + ... + \frac{\partial f_2}{\partial x_6}\delta x_6 \\ &\vdots \\ \delta y_6 &= \frac{\partial f_6}{\partial x_1}\delta x_1 + \frac{\partial f_6}{\partial x_2}\delta x_2 + ... + \frac{\partial f_6}{\partial x_6}\delta x_6 \end{aligned}$$

Definition of Jacobian Matrix

- Using vector notations

$$Y = F(X)$$
$$\Rightarrow \delta Y = \frac{\partial F}{\partial X}\delta X$$

- The 6×6 matrix of partial derivative matrix is called the Jacobian.

$$\delta Y = J(X)\delta X$$

- By dividing both sides by the differential time element, we can think of the Jacobian as mapping velocities in X to those in Y:

$$\dot{Y} = J(X)\dot{X}$$

Jacobian of The Robotic Manipulators

- In the field of robotics, Jacobian is a mapping between joint velocities and Cartesian velocities of the tip of the arm.

$$^{0}v = {}^{0}J(\Theta)\dot{\Theta}$$

- Θ is the vector of joint angles of the manipulator and v is a vector of Cartesian velocities.
- A leading superscript is added to Jacobian to indicate in which frame the resulting Cartesian velocity is expressed.
- This superscript is omitted when the frame is obvious or unimportant.
- For the general case of a six-jointed robot, the Jacobian is 6×6, $\dot{\Theta}$ is 6 x 1, and ^{0}v is 6×1.

Singularities

- For a matrix $J \in \mathbb{R}^{6\times n}$, it is always the case that rank$J \leq \min(6, n)$.
- For example, for the two-link planar arm, we always have rank$J \leq 2$.
- The rank of a matrix is not necessarily constant.
- Indeed, the rank of the manipulator Jacobian matrix will depend on the configuration q.
- Configurations for which the rank $J(q)$ is less than its maximum value are called **singularities** or **singular configurations**.

Singularities

Why identifying manipulator singularities is important?

1. Singularities represent configurations from which certain directions of motion may be unattainable.
2. At singularities, bounded end-effector velocities may correspond to unbounded joint velocities.
3. Singularities usually (but not always) correspond to points on the boundary of the manipulator workspace.
4. Near singularities there will not exist a unique solution to the inverse kinematics problem.

Singularities

- **Example :** Where are the singularities of the simple two-link arm? What is the physical explanation of the singularities? Are they workspace-boundary singularities or workspace-interior singularities?
- At the singular points, the determinant of J is equal to zero:

$$\det(J) = \begin{bmatrix} l_2 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix} = l_1 l_2 s_2 = 0$$

- Clearly, a singularity of the mechanism exists when θ_2 is 0 or 180 degrees.
- Physically, when $\theta_2 = 0$, the arm is stretched straight out. Likewise, when $\theta_2 = 180$, the arm is folded completely back on itself.
- They are workspace-boundary singularities, because they exist at the edge of the manipulator's workspace.

Jacobian in Force Domain

- In the multidimensional case, work is the dot product of a vector force or torque and a vector displacement. Thus, we have

$$F.\delta X = \tau.\delta\Theta$$

- F is a 6×1 Cartesian force-moment vector acting at the end-effector, δX is a 6×1 infinitesimal Cartesian displacement of the end-effector, τ is a 6×1 vector of torques at the joints, and $\delta\Theta$ is a 6×1 vector of infinitesimal joint displacements.
- The above expression can also be written as

$$F^T \delta X = \tau^T \delta\Theta$$

- The definition of the Jacobian is

$$\delta X = J\delta\Theta$$

Jacobian in Force Domain

- So we may write

$$F^T J \delta\Theta = \tau^T \delta\Theta$$

which must hold for all $\delta\Theta$; hence, we have

$$F^T J = \tau^T$$

- Transposing both sides yields this result:

$$\tau = J^T F$$

- When the Jacobian loses full rank, there are certain directions in which the end effector cannot exert static forces even if desired. Thus, with small joint torques, large forces could be generated at the end-effector.

Dynamics of Robots

- Matrix Form of robot dynamics is expressed as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \tag{1}$$

where $q \in \mathbb{R}^{n\times 1}$ is the vector of joint positions, $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n\times n}$ is the centrifugal and Coriolis matrix, $g(q) \in \mathbb{R}^{n\times 1}$ is the gravity vector, and $u \in \mathbb{R}^{n\times 1}$ is the control torque vector.

Dynamics of Robots

- **Example:** It is possible to find the components of dynamic model by using the Lagrange method. The inertia matrix is

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} \tag{2}$$

where

$$\begin{aligned} M_{11} =& m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} cos(q_2)) + I_1 + I_2 \\ M_{12} =& M_{21} = m_2(l_{c2}^2 + l_1 l_{c2} cos(q_2)) + I_2 \\ M_{22} =& m_2 l_{c2}^2 + I_2 \end{aligned}$$

Dynamics of Robots

- Also, matrix C is stated as

$$C(q,\dot{q}) = \begin{bmatrix} h\dot{q}_2 & h\dot{q}_2 + h\dot{q}_1 \\ -h\dot{q}_1 & 0 \end{bmatrix}$$

where

$$h = -m_2 l_1 l_{c2} sin(q_2).$$

- In addition, the gravity vector is expressed as

$$G = \begin{bmatrix} (m_1 l_{c1} + m_2 l_1) gcos(q_1) + m_2 l_{c2} gcos(q_1 + q_2) \\ m_2 l_{c2} gcos(q_1 + q_2) \end{bmatrix}$$

Properties of Robot Dynamic Equations

Proposition (*The Skew Symmetry Property*)

Let $M(q)$ be the inertia matrix for an n-link robot and define $C(q, \dot{q})$ in terms of the elements of $M(q)$. Then the matrix $N(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q})$ is skew symmetric, that is, the components n_{jk} of N satisfy $n_{jk} = -n_{kj}$.

- It is concluded from the skew symmetry property that

$$v^T(\dot{M}(q) - 2C(q, \dot{q}))v = 0 \quad \forall v \in \mathbb{R}^n$$

- Note that, the above equation is correct for any v, provided that $C(q, \dot{q})$ is in Christoffel form. Otherwise, it is valid only for $v = \dot{q}$.

Properties of Robot Dynamic Equations

Proposition (*The Passivity Property*)

There exists a constant, $\beta \geq 0$, such that

$$\int_0^T \dot{q}^T(\zeta)\tau(\zeta)d\zeta \geq -\beta, \quad \forall T > 0$$

- The term $\dot{q}\tau$ has units of power and therefore the expression in the left side of the above inequality is the energy produced by the system over the time interval $[0, T]$.
- Passivity therefore means that the amount of energy dissipated by the system has a lower bound.
- The word passivity comes from circuit theory where a passive system is one that can be built from passive components (resistors, capacitors, inductors).

Properties of Robot Dynamic Equations

Proposition (*Bounds on the Inertia Matrix*)

The inertia matrix for an n-link rigid robot is symmetric and positive definite. For a fixed value of the generalized coordinate q, let $0 < \lambda_1(q) \leq ... \leq \lambda_n(q)$ denote the n eigenvalues of M(q). These eigenvalues are positive as a consequence of the positive definiteness of $M(q)$. As a result, it can easily be shown that

$$\lambda_1(q) I_{n\times n} \leq M(q) \leq \lambda_n(q) I_{n\times n}$$

- If all of the joints are revolute then the inertia matrix contains only bounded functions of the joint variables (sin & cos) Hence, constants λ_m and λ_M provide uniform (independent of q) bounds in the inertia matrix as

$$\lambda_m I_{n\times n} \leq M(q) \leq \lambda_M I_{n\times n} < \infty$$

Properties of Robot Dynamic Equations

Proposition (*The Linearity in the Parameters Property*)

There exists an $n \times l$ function, $Y(q, \dot{q}, \ddot{q})$, which we assume is completely known, and an l-dimensional vector Θ such that the Euler-Lagrange equations can be written

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = Y(q, \dot{q}, \ddot{q})\Theta$$

- The function, $Y(q, \dot{q}, \ddot{q})$, is called the Regressor and Θ is the Parameter vector.
- The dimension of the parameter space, $\mathcal{R}^l$, i.e., the number of needed parameters is not unique.

Properties of Robot Dynamic Equations

- In general, a given rigid body is described by ten parameters, namely, the total mass, the six independent entries of the inertia tensor, and the three coordinates of the center of mass. An n-link robot then has a maximum of $10n$ dynamics parameters.
- **Example:** For the two link planar robot, find the regressor and the parameter vector.
- If we group the inertia terms appearing in Equation (2) as

$$\begin{aligned}
\theta_1 &= m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2) + I_1 + I_2 \\
\theta_2 &= m_2 l_1 l_{c2} \\
\theta_3 &= m_2 l_{c2}^2 + I_2
\end{aligned}$$

Properties of Robot Dynamic Equations

- Then we can write the inertia matrix elements as

$$m_{11} = \theta_1 + 2\theta_2 \cos(q_2)$$
$$m_{12} = m_{21} = \theta_3 + \theta_2 \cos(q_2)$$
$$m_{22} = \theta_3$$

- No additional parameters are required in the Christoffel symbols as these are functions of the elements of the inertia matrix.
- The gravitational torques require additional parameters as

$$\theta_4 = m_1 l_{c1} + m_2 l_1$$
$$\theta_5 = m_2 l_2$$

Properties of Robot Dynamic Equations

- Therefore,

$$Y(q,\dot{q},\ddot{q}) = \begin{bmatrix} \ddot{q}_1 & Y_{12} & \ddot{q}_2 & gcos(q_1) & gcos(q_1+q_2) \\ 0 & Y_{22} & \ddot{q}_1+\ddot{q}_2 & 0 & gcos(q_1+q_2) \end{bmatrix}$$

in which,

$$\begin{aligned} Y_{12} =& cos(q_2)(2\ddot{q}_1+\ddot{q}_2) - sin(q_2)(\dot{q}_1^2 + 2\dot{q}_1\dot{q}_2) \\ Y_{22} =& cos(q_2)\ddot{q}_1 + sin(q_2)\dot{q}_1^2 \end{aligned}$$

- and the parameter vector is given by

$$\Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} = \begin{bmatrix} m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2) + I_1 + I_2 \\ m_2 l_1 l_{c2} \\ m_2 l_{c2}^2 + I_2 \\ m_1 l_{c1} + m_2 l_1 \\ m_2 l_2 \end{bmatrix}$$

Path Planning

- The solutions of the forward and inverse kinematics problems depend only on the intrinsic geometry of the robot, and they do not reflect any constraints imposed by the workspace in which the robot operates.
- In particular, they do not take into account the possiblity of collision between the robot and objects in the workspace.
- The problem of planning collision free paths for the robot is addressed in the path planning problem.
- It is assumed that the initial and final configurations of the robot are specified, and that the problem is to find a collision free path for the robot that connects them.
- The description of this problem is deceptively simple, yet the path planning problem is among the most difficult problems in computer science.

Introduction

- A path planning algorithm is said to be complete if it finds a solution whenever one exists.
- The computational complexity of the best known complete path planning algorithm grows exponentially with the number of internal degrees of freedom of the robot.
- Hence, for robot systems with more than a few degrees of freedom, complete algorithms are not used in practice.
- **Path planning** provides a geometric description of robot motion, but it does not specify any dynamic aspects of the motion (e. g. joint velocities and accelerations).
- **Trajectory planning** computes a function $q_d(t)$ that completely specifies the motion of the robot as it traverses the path.

Trajectory Planning

- A Path is a geometric description of motion.
- A path planning algorithm will typically give only a sequence of points (called via points) along the path.
- A trajectory is a function of time $q(t)$ such that $q(t_0) = q_{\text{init}}$ and $q(t_f) = q_{\text{final}}$.
- In this case, $t_f - t_0$ represents the amount of time taken to execute the trajectory.
- Since the trajectory is parameterized by time, we can compute velocities and accelerations along the trajectories by differentiation.

Trajectory Planning

- The problem is to find a trajectory between initial and final configuration under velocity and/or acceleration constraints.
- Without loss of generality, we will consider planning the trajectory for a single joint, since the trajectories for the remaining joints will be created the same way.
- We suppose that the constraints are presented as

$$
\begin{aligned}
q(t_0) &= q_0 \\
\dot{q}(t_0) &= v_0 \\
q(t_f) &= q_f \\
\dot{q}(t_f) &= v_f
\end{aligned}
$$

- In addition, we may have acceleration constraints as

$$
\begin{aligned}
\ddot{q}(t_0) &= a_0 \\
\ddot{q}(t_f) &= a_f
\end{aligned}
$$

Trajectory Planning

Cubic Polynomial Trajectories

- One way to generate a smooth curve is by a polynomial function of t.
- If we have four constraints to satisfy, we require a polynomial with four independent coefficients that can be chosen to satisfy these constraints.
- Thus we consider a cubic trajectory of the form

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

- Then the desired velocity is given as

$$\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

Trajectory Planning

- Combining the above equations with the four constraints:

$$\begin{aligned} q_0 &= a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 \\ v_0 &= a_1 + 2a_2 t_0 + 3a_3 t_0^2 \\ q_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \\ v_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 \end{aligned}$$

- The matrix form:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ v_0 \\ q_f \\ v_f \end{bmatrix}$$

- The determinant of the coefficient matrix is $(t_f - t_0)^4$; hence, the above Equation always has a unique solution if $t_f \neq t_0$

Trajectory Planning

Quintic Polynomial Trajectories

- A cubic trajectory gives continuous positions and velocities at the start and finish points times but discontinuities in the acceleration.
- The derivative of acceleration is called the jerk. A discontinuity in acceleration leads to an impulsive jerk, which may excite vibrational modes in the manipulator and reduce tracking accuracy.
- For this reason, one may wish to specify constraints on the acceleration as well as on the position and velocity.
- In this case, we have six constraints (one each for initial and final configurations, initial and final velocities, and initial and final accelerations).

Trajectory Planning

- Therefore we require a fifth order polynomial

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

- By taking the appropriate number of derivatives we obtain the following equations

$$\begin{aligned}
q_0 &= a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 + a_5 t_0^5 \\
v_0 &= a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 + 5a_5 t_0^4 \\
a_0 &= 2a_2 + 6a_3 t_0 + 12a_4 t_0^2 + 20a_5 t_0^3 \\
q_f &= a_f + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\
v_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\
a_f &= 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3
\end{aligned}$$

Trajectory Planning

- The matrix form:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ v_0 \\ a_0 \\ q_f \\ v_f \\ a_f \end{bmatrix}$$