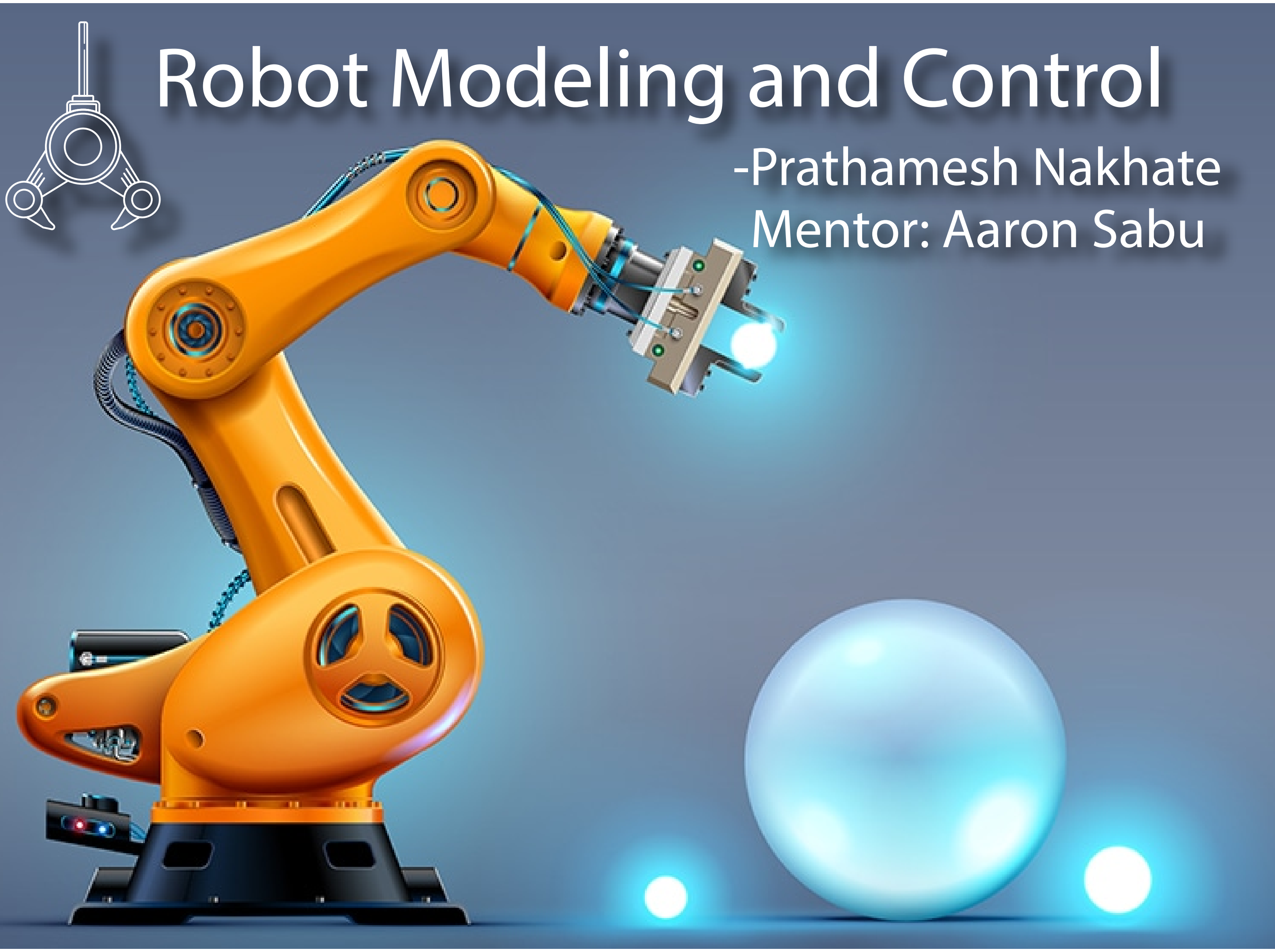

Robot Modeling and Control
-Prathamesh Nakhate
Mentor: Aaron Sabu

Robot Modeling and Control

Prathamesh Nakhate
190100095
Undergraduate, Department of Mechanical Engineering
Indian Institute of Technology, Bombay

June 10, 2020

Mentor's Name: Aaron John Sabu

Abstract

We start with learning basic linear algebra and rotation matrices then move onto homogeneous transformations. We then learn basics of path planning and trajectory, then we move on to learning kinematics and dynamics of a manipulator and end with learning about the formulation of equations of motion of manipulators.

Contents

Introduction

Robotics is a relatively young field of modern technology that crosses traditional engineering boundaries. Understanding the complexity of robots and their applications requires knowledge of electrical engineering, mechanical engineering, systems and industrial engineering, computer science, economics, and mathematics. New disciplines of engineering, such as manufacturing engineering, applications engineering, and knowledge engineering have emerged to deal with the complexity of the field of robotics and factory automation. This report is concerned with fundamentals of robotics, including kinematics, dynamics and motion planning

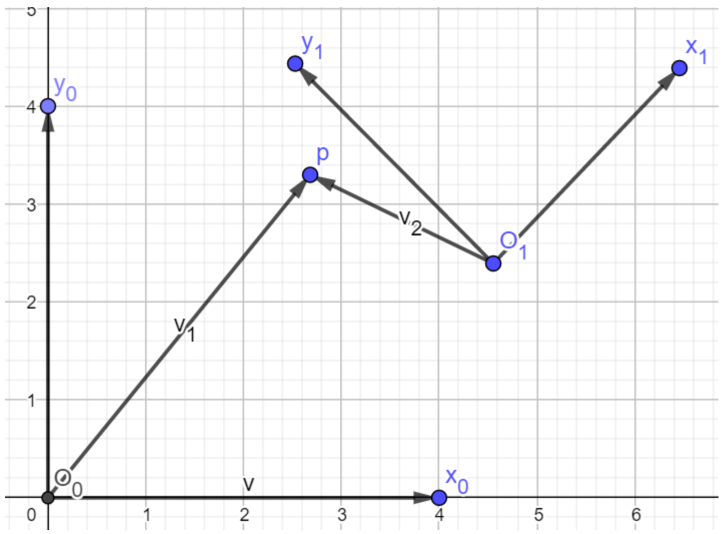
The report begins by introducing some mathematical tools which will be used throughout the report. To understand this report one should have some prior knowledge of matrices (MA-106 stuff).

Acknowledgments

This work was written as a part of the Summer Of Science, 2020 by the MnP Club, IIT Bombay. I have followed the book "Robot Modeling and Control"[1]. All content here is highly inspired by this book and at several points might simply be a paraphrasing of it's content. Nevertheless, I have tried to compress the contents of the book while still preserving sufficient depth. Also, I would like to acknowledge the help of my mentor, Aaron John Sabu, for his help and support.

Rigid motions and Homogeneous transformations

3.1 Representing Positions



We represent position as follows

$p^0 = \begin{bmatrix} 2.68 \\ 3.3 \end{bmatrix}$ $p^1 = \begin{bmatrix} -0.5 \\ 1.6 \end{bmatrix}$ $v_1^0 = \begin{bmatrix} 2.68 \\ 3.3 \end{bmatrix}$ $v_1^1 = \begin{bmatrix} 0.5 \\ 3.7 \end{bmatrix}$ $v_2^0 = \begin{bmatrix} -1.87 \\ 0.91 \end{bmatrix}$ $v_2^1 = \begin{bmatrix} -0.5 \\ 1.6 \end{bmatrix}$
$o_1^0 = \begin{bmatrix} 4.55 \\ 2.39 \end{bmatrix}$ $o_0^1 = \begin{bmatrix} -5.6 \\ 0.3 \end{bmatrix}$

The superscript represents the coordinate frame with respect to which we are viewing and the subscript describes the object which we are viewing.

3.2 Representing Rotations

We define rotation matrices as, $R_1^0 = [x_1^0|y_1^0|z_1^0]$ where x_1^0, y_1^0 and z_1^0 are the coordinates in frame $o_0x_0y_0z_0$ of unit vectors x_1, y_1 and z_1 respectively.
Therefore,

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix} \in SO(3) \tag{1}$$

Following are some of properties of matrix group SO(n):-

- $R \in SO(n)$
- $R^{-1} \in SO(n)$

- $R^{-1} = R^t$
- The columns (and therefore the rows) of R are mutually orthogonal.
- Each column (and therefore the row) of R are unit vectors.
- det R = 1

Using the definition of R_1^0 we can find the following three basic rotation matrices.

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \qquad R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \qquad R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

They represent rotation about one of the coordinate axes by an angle of θ.
These rotation matrices also follow some properties such as

$$R_{z,0} = I, R_{z,\theta}R_{z,\phi} = R_{z,\theta+\phi}$$

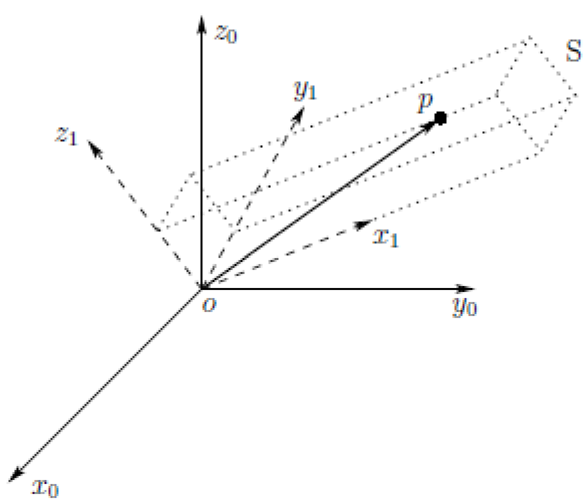
and as a consequence of these two

$$(R_{z,\theta})^{-1} = R_{z,-\theta}$$

3.3 Rotational Transformation

Consider a rigid object S to which a coordinate frame $o_1x_1y_1z_1$ is rigidly attached. Let $p^1 = (u, v, w)^t$, therefore we can write

$$p = ux_1 + vy_1 + wz_1$$



We can also write $p^0 = \begin{bmatrix} p.x_0 \\ p.y_0 \\ p.z_0 \end{bmatrix}$ Combining the above two equations,

$$\begin{aligned} p^0 &= \begin{bmatrix} (ux_1 + vy_1 + wz_1).x_0 \\ (ux_1 + vy_1 + wz_1).y_0 \\ (ux_1 + vy_1 + wz_1).z_0 \end{bmatrix} \\ &= \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \end{aligned}$$

$$\boxed{p^0 = R_1^0 p^1}$$

3.3.1 Similarity Transformation

The matrix representation of a general linear transformation is transformed from one frame to another using a so-called similarity transformation.

If A is the matrix representation of a given linear transformation in $o_0x_0y_0z_0$ and B is the representation of the same linear transformation in $o_1x_1y_1z_1$ then A and B are related as

$$B = (R_1^0)^{-1} A R_1^0$$

3.4 Composition of Rotations

3.4.1 Rotation w.r.t current frame

Consider three coordinate frames $o_0x_0y_0z_0$, $o_1x_1y_1z_1$ and $o_2x_2y_2z_2$ and take a point p.

$$p^0 = R_1^0 p^1 \tag{2}$$
$$p^1 = R_2^1 p^2 \tag{3}$$
$$p^0 = R_2^0 p^2 \tag{4}$$

Substituting equation 3 in equation 2 we get,

$$p^0 = R_1^0 R_2^1 p^2$$

Now comparing with equation 4 we get,

$$\boxed{R_2^0 = R_1^0 R_2^1}$$

3.4.2 Rotation w.r.t fixed frame

Many times it is desired to perform a sequence of rotations, each about a given fixed coordinate frame, rather than about successive current frames. For example we may wish to perform a rotation about x_0 followed by a rotation about y_0 (and not y_1). Here we will refer to $o_0x_0y_0z_0$ as the fixed frame.

Suppose we name the second rotation by R, then the representation for R in the current frame $o_1x_1y_1z_1$ is given by $(R_1^0)^{-1}RR_1^0$. Using the result we proved in section 3.4.1 we can write,

$$R_2^0 = R_1^0(R_1^0)^{-1}RR_1^0$$
$$\boxed{R_2^0 = RR_1^0}$$

Observe that this result is exactly opposite of what we had got in section 3.4.1.

3.5 Parametrization of Rotations

3.5.1 Euler Angles

We can specify the orientation of the frame $o_1x_1y_1z_1$ relative to the frame $o_0x_0y_0z_0$ by three angles (ϕ, θ, ψ), known as Euler Angles.Generally, the required rotation is obtained by rotating w.r.t current axes Z-Y-Z respectively. Therefore,

$$\begin{aligned} R_1^0 &= R_{ZYZ} = R_{z,\phi}R_{y,\theta}R_{z,\psi} \\ &= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix} \end{aligned} \tag{5}$$

Now, we want to find Euler angles for a given $R \in SO(3)$. We break this problem into cases and solve.
Suppose, $R = \left(\begin{smallmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{smallmatrix}\right)$

<u>Case-1</u> (Compare equation 5 with R)
Not both of r_{13} and r_{23} are zero.
$\therefore$we know $s_\theta \neq 0$
$\therefore c_\theta = r_{33}, s_\theta = \pm\sqrt{1 - r_{33}^2}$
$\therefore \theta_1 = atan2(r_{33}, \sqrt{1 - r_{33}^2})$
or
$\theta_2 = atan2(r_{33}, -\sqrt{1 - r_{33}^2})$

where $atan2$ is the double argument tan function.

If we take $\theta = \theta_1$ then $s_\theta > 0$
$\phi = atan2(r_{13}, r_{23})$
$\psi = atan2(-r_{31}, r_{32})$

Similarly do for $\theta = \theta_2$

<u>Case-2</u> (Compare equation 5 with R)

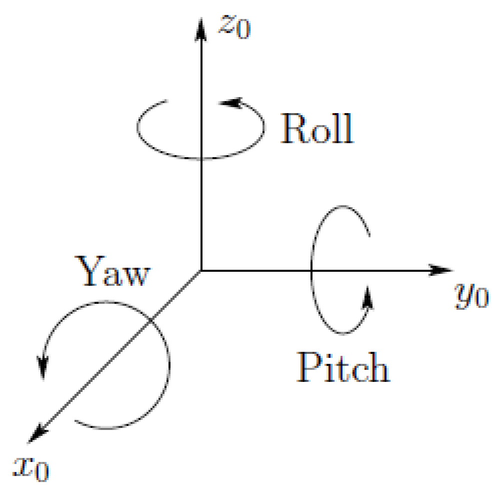
$r_{13} = r_{23} = 0 \implies r_{33} = \pm 1$
If $r_{33} = 1 \implies c_\theta = 1 \implies \theta = 0$

$$\therefore R = \begin{bmatrix} c_{\phi+\psi} & -s_{\theta+\psi} & 0 \\ s_{\phi+\psi} & c_{\phi+\psi} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\therefore \phi + \psi = atan2(r_{11}, r_{21})$

Similarly do for $r_{33} = -1$

3.5.2 Roll, Pitch, Yaw Angles



This similar to Euler angles, the only difference is that the rotations with (ψ, θ, ϕ) specified in x-y-z order (roll-pitch-yaw) are w.r.t fixed frame i.e world frame.

$$\therefore R_{XYZ} = R_{z,\phi} R_{y,\theta} R_{x,\psi}$$

3.5.3 Axis/Angle Representation

Let $\mathbf{k} = (k_x, k_y, k_z)^t$, expressed in the frame $o_0x_0y_0z_0$, be a unit vector defining an axis. Then we can claim that any rotation matrix $R \in SO(3)$ can be represented by a single rotation about a suitable axis in space by a suitable angle.

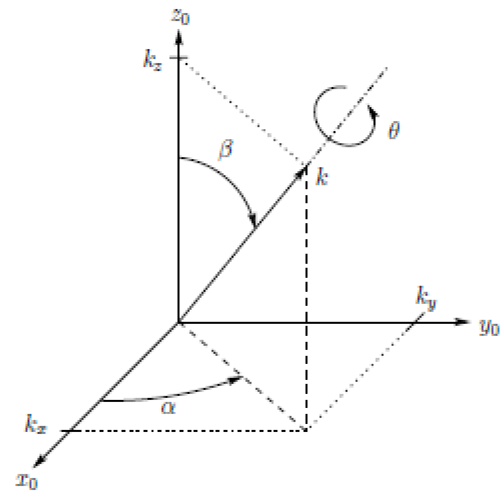
Let the axis defined by the vector k is along the z-axis of our new coordinate frame $o_1x_1y_1z_1$

$$\therefore R_1^0 = R_{z,\alpha} R_{y,\beta}$$

.

Therefore, a rotation about the axis k can be computed using a similarity transformation as

$$R_{k,\theta} = R_1^0 R_{z,\theta} {R_1^0}^{-1}$$



$$\boxed{R_{k,\theta} = R_{z,\alpha}R_{y,\beta}R_{z,\theta}R_{y,-\beta}R_{z,-\alpha}}$$

We can find α and β from the figure.

Given an arbitrary rotation matrix R with components r_{ij},we can find the equivalent angle θ and equivalent axis k by the following expressions.

$$\theta = \arccos\left(\frac{Tr(R) - 1}{2}\right)$$

where $Tr(R)$ denotes trace of R.

$$k = \frac{1}{2\sin\theta}(R - R^t)$$

Note that the axis/angle representation is not unique since a rotation of $-\theta$ about $-k$ is the same as a rotation of θ about k, that is,

$$R_{k,\theta} = R_{-k,-\theta}$$

3.6 Rigid Motions

A rigid motion is an ordered pair (d, R) where $d \in \mathbb{R}^3$ and $R \in SO(3)$. The group of all rigid motions is known as the Special Euclidean Group and is denoted by SE(3). We see then that $SE(3) = \mathbb{R}^3 \times SO(3)$.A rigid motion is a pure translation together with a pure rotation.

Suppose $o_1x_1y_1z_1$ is obtained from frame $o_0x_0y_0z_0$ by first applying a rotation specified by R_1^0 followed by a translation given (with respect to $o_0x_0y_0z_0$) by d_1^0, then the coordinates p^0 are given by,

$$p^0 = R_1^0p^1 + d_1^0 \tag{6}$$

3.7 Homogeneous Transformation

Rigid motions can be represented in matrix form are known as homogeneous transformations. These help us in easily handling multiple rigid motions.

The rigid motions can be represented by the set of matrices of the form

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \in SE(n)$$

where 0 denotes the row vector (0,0,0).

In order represent transformations as we did in equation 6 we augment the vectors p^0 and p^1 as

$$P^0 = \begin{bmatrix} p^0 \\ 1 \end{bmatrix}$$
$$P^0 = \begin{bmatrix} p^1 \\ 1 \end{bmatrix}$$

These new augmented vectors are called homogeneous representations of p^0 and p^1. Transformation given by equation 6 is equivalent to

$$P^0 = H_1^0 P^1$$

Composition of homogeneous transformation happens in the exact same way as that of rotation matrices.

Forward and Inverse Kinematics

4.1 Kinematic Chains

In general a manipulator has n joints, n+1 links and each joint connects two links. We number joints from 1 to n and links from 0 to n, starting from the base. Therefore we can see that joint i connects links i and i-1. We rigidly attach coordinate frame $o_i x_i y_i z_i$ to link i. Therefore, when joint i is actuated both link i and the coordinate frame $o_i x_i y_i z_i$ move.

A_i is the homogeneous transformation which represents position and orientation of $o_i x_i y_i z_i$ w.r.t $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$.

$$\therefore A_i = A_i(q_i)$$

where q_i is the joint variable of i^{th} joint.

We define a transformation matrix as follows,

$$T_j^i = \begin{matrix} A_{i+1}A_{i+2}\dots A_j & for \;\; i < j \\ I & for \;\; i = j \\ T_i^{j-1} & for \;\; i > j \end{matrix}$$

We can use this T_j^i to relate between different joints of the manipulator.

In principle, that is all there is to forward kinematics; determine the functions $A_i(q_i)$, and multiply them together as needed. However, it is possible to achieve a considerable amount simplification by introducing further conventions, such as the Denavit-Hartenberg representation of a joint.

4.2 Forward Kinematics

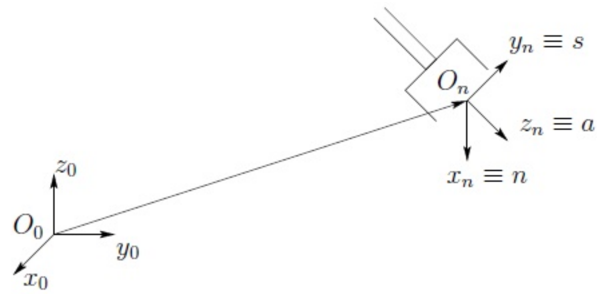
Each homogeneous transformation A_i is represented as a product of four basic transformations

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \tag{7}$$

This is valid only if the coordinate frames are made according to the DH convention.

Following are the rules of DH Convention:-

- Rule 1
  z_{i-1} is the axis of actuation of joint i.
- Rule 2
  Axis x_i is set so it is perpendicular to and intersects z_{i-1}.
- Rule 3
  Derive y_i from x_i and z_i.



- <u>Rule 4</u>
  Coordinate frame of tool is always made according to the fig given above.

If we make all the coordinate frames according to this convention then the parameters of equation 7 are

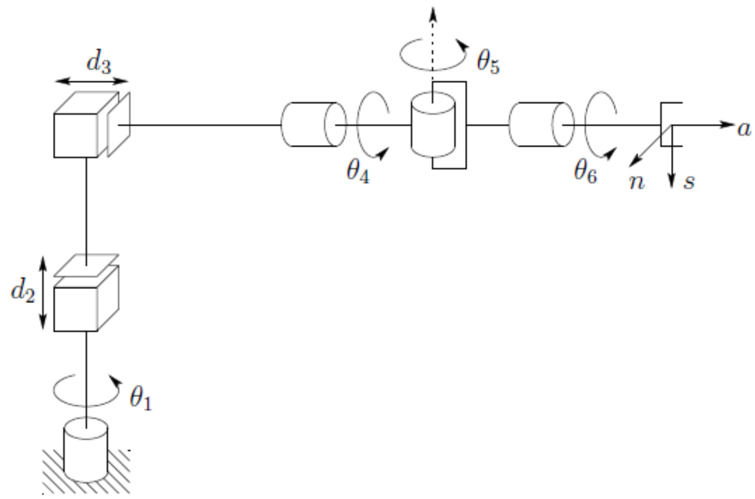
- θ_i is the angle between x_{i-1} and x_i measured about z_{i-1}.
- d_i is the distance between x_{i-1} and x_i measured along z_{i-1}.
- a_i is the distance between z_{i-1} and z_i measured along x_i.
- α_i is the angle between z_{i-1} and z_i measured about x_i.

For a given joint α_i and a_i remain constant unless the configuration of the robot is not changed.

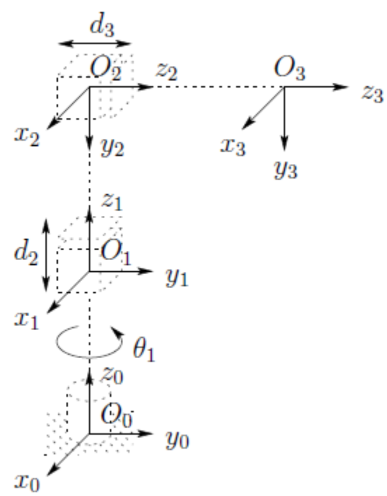
Further, there are three cases in Rule 2.

1. z_{i-1} and z_i are not co-planar.
   - Only one possible x_i, which is the shortest line from z_{i-1} to z_i.
   - o_i is the intersection of x_i and z_i.
2. z_{i-1} and z_i are parallel.
   - Infinite possibilities for x_i.
   - Choose the x_i which passes through o_{i-1} (since it gives $d_i = 0$).
   - o_i is the intersection of x_i and z_i.
   - $\alpha_i = 0$ for this case.
3. z_{i-1} intersects z_i
   - x_i is normal to the plane of z_{i-1} and z_i .
   - o_i naturally sits on the intersection of z_{i-1} and z_i but can be anywhere on z_i.
   - $a_i = 0$ for this case.

Now we will look at an example of cylindrical manipulator with spherical wrist.



Now, we can break this arm into two parts, the cylindrical part and the spherical part and draw the coordinate frames on them individually (at the end of this example I will tell why we can do this).

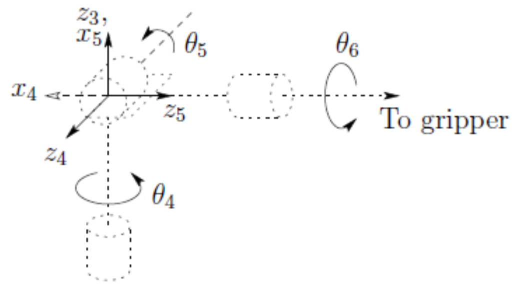


The figure given above shows the coordinate frames for the cylindrical part. The DH parameters for this are given below

Link	a_i	α_i	d_i	θ_i
1	0	0	d_1	θ_1^*
2	0	-90	d_2^*	0
3	0	0	d_3^*	0

* variable

Similarly we draw the coordinate frame and find the DH parameters for the spherical wrist.



Link	a_i	α_i	d_i	θ_i
4	0	-90	0	θ_4^*
5	0	90	0	θ_5^*
6	0	0	d_6	θ_6^*

* variable

Using all this information about the DH parameters we can find T_n^0 and thus we have solved the forward kinematics problem for this manipulator.

The reason why we were able to break this manipulator into two parts was because all the three coordinate axes of the spherical wrist intersect at a point and therefore we can collectively represent them by one coordinate frame (which was $o_3x_3y_3z_3$ in the diagram of cylindrical part). This point is also know as the wrist center (symbolized by o_c). This property of the spherical wrist becomes very useful while solving the inverse kinematic problem.

4.3 Inverse Kinematics

4.3.1 The general inverse kinematics problem

Inverse kinematic problem is basically finding joint angles using the position of the end-effector i.e finding the values of $q_1, q_2, \ldots q_n$ from equation 8

$$T_n^0(q_1, q_2, \ldots q_n) = H \tag{8}$$

where H is the homogeneous transformation between base frame and end-effector frame.

Equation 8 will give 12 non-linear equations in n unknown variables. Therefore we see that the inverse kinematic problem may not have a unique solution. Also solving 12 equations is a very tedious task so we simplify this problem further by the concept of kinematic decoupling.

4.3.2 Kinematic Decoupling

For manipulators having 6 joints with the last three joint axes intersecting at a point (for example the spherical wrist), it is possible to decouple the equation into inverse position kinematics and inverse orientation kinematics.

Assumption of spherical wrist means thatz_3, z_4 and z_5 intersect at o_c. $\therefore$ origins o_4 and o_5 will always be at o_c which means that motion of final three links about its axes will not change the position of o_c. Therefore position of o_c is a function of first three joint variables.

Now, we find the relation between o (position of tool frame i.e o_6^0) and o_c. Using this relation we find first three joint variables. Now using R_6^0 (given) and R_3^0 (which we just found) we find R_6^3 and then using this we find $\theta_4, \theta_5, \theta_6$ by the analysis we had done in 3.5.1. This is the basic idea behind kinematic decoupling.

4.3.3 Inverse Position: A geometric approach

Now we will see how we an solve an inverse position problem using a gemetric approach.We represent the coordinates of wrist centre by (x_c, y_c)//

Let us take an example of elbow manipulator with shoulder offset and solve our problem.//

The basic approach is always the same. To find a joint variable q_i we project the manipulator in $x_{i-1}y_{i-1}$ plane and then use basic trigonometry.

Now we find θ_1 by projecting the manipulator in x_0y_0 plane.

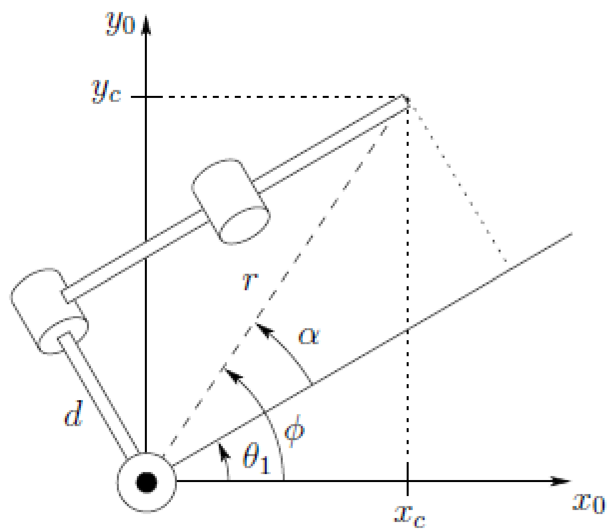


Figure 1: Left arm configuration

$$\phi = atan2(x_c, y_c) \tag{9}$$

$$\alpha = atan2(\sqrt{r^2 - d^2}, d) \tag{10}$$

$$= atan2(\sqrt{x_c^2 + y_c^2 - d^2}, d)$$

$$\theta_1 = \phi - \alpha \tag{11}$$

Now, we can find θ_1 using equations 9, 10 and 11.

Observe that the coordinates (x_c, y_c) can be achieved by one more configuration other than that shown in figure 1. This configuration is shown below.

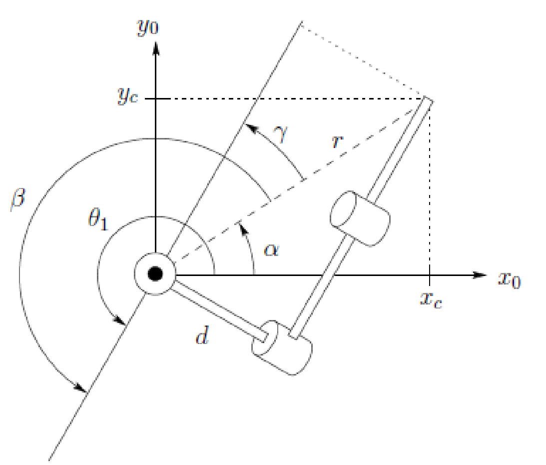
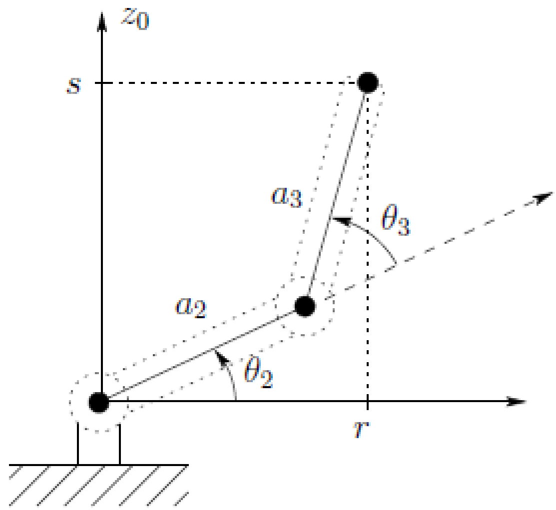


Figure 2: Right arm configuration

This gives us another solution to the same problem.

$$
\begin{aligned}
\alpha &= atan2(x_c, y_c) \\
\gamma &= atan2(\sqrt{r^2 - d^2}, d) \\
\beta &= \gamma + \pi \\
\therefore \beta &= atan2(-\sqrt{r^2 - d^2}, -d) \\
\theta_1 &= \beta + \alpha
\end{aligned}
$$

Now, to find the angles θ_2 and θ_3 for the elbow manipulator, given θ_1, we consider the plane formed by the second and third links.



Now we apply cosine law to obtain,

$$
\begin{aligned}
\cos\theta_3 &= \frac{r^2 + s^2 - a_2^2 - a_3^2}{2a_2a_3} \\
&= \frac{x_c^2 + y_c^2 - d^2 + (z_c - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3} = D \\
\therefore \theta_3 &= atan2(D, \pm\sqrt{1 - D^2})
\end{aligned}
$$

where d_1 is the length of link 1.
We get two solutions for θ_3 which correspond to elbow up and elbow down position.

Similarly using basic trigonometry we get,

$$
\begin{aligned}
\theta_2 &= atan2(r, s) - atan2(a_2 + a_3c_3, a_3s_3) \\
&= atan2(\sqrt{x_c^2 + y_c^2 - d^2}, z_c - d_1) - atan2(a_2 + a_3c_3, a_3s_3)
\end{aligned}
$$

In this way we have solved the inverse position problem.

4.3.4 Inverse Orientation

In the previous section we used a geometric approach to solve the inverse position problem. This gives the values of the first three joint variables corresponding to a given position of the wrist origin. The inverse orientation problem is now one of finding the values of the final three joint variables corresponding to a given orientation with respect to the frame $o_3x_3y_3z_3$. For a spherical wrist, this can be interpreted as the problem of finding a set of Euler angles corresponding to a given rotation matrix R. We have already made a method to do this in section 3.5.1. We solve for the Euler angles (ϕ, θ, ψ) then use the following mapping

$$\begin{aligned} \theta_4 &= \phi \\ \theta_5 &= \theta \\ \theta_6 &= \psi \end{aligned}$$

In this way we have solved both the forward and inverse kinematics for a manipulator.

Velocity Kinematics-The Manipulator Jacobian

This chapter basically deals with finding the solution for forward and inverse velocity kinematics and finding joint variables at singular configurations.

5.1 Skew Symmetric Matrices

An nn matrix S is said to be skew symmetric if and only if

$$S^t + S = 0$$

If $\mathbf{a} = (a_x, a_y, a_z)^t$ is a 3-vector, we define the skew symmetric matrix $S(a)$ as

$$S(a) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

5.1.1 Properties of Skew Symmetric Matrices

1. The operator S is linear i.e,

$$S(\alpha a + \beta b) = \alpha S(a) + \beta S(b)$$

2. For any vectors a and p in $\mathbb{R}^3$,

$$S(a)p = a \times p$$

3. If $R \in SO(3)$ and a, b are vectors in $\mathbb{R}^3$ then,

$$R(a \times b) = R(a) \times R(b)$$

4. For $R \in SO(3)$ and $a \in \mathbb{R}^3$,

$$RS(a)R^t = S(Ra)$$

5.1.2 Derivative of Rotation Matrix

Suppose, $R(\theta) = R \in SO(3)$ for all θ, then it follows that

$$R(\theta)R(\theta)^t = I$$

Differentiate both sides w.r.t θ,

$$\frac{dR}{d\theta}R^t + R\frac{dR^t}{d\theta} = 0$$

Let $S = \frac{dR}{d\theta}R^t \implies S^t = R\frac{dR^t}{d\theta}$. Therefore we can conclude that S i.e $\frac{dR}{d\theta}R^t$ is a skew symmetric matrix.

$$\boxed{\frac{dR}{d\theta} = SR(\theta)}$$

Therefore computing the derivative of the rotation matrix R is equivalent to a matrix multiplication by a skew symmetric matrix S. Further it can be shown that,

$$\frac{dR_{k,\theta}}{d\theta} = S(k)R_{k,\theta}$$

5.2 Angular Velocity

Suppose that a rotation matrix R is time varying, so that $R = R(t) \in SO(3)$ for every $t \in \mathbb{R}$.

$$\begin{aligned} \frac{dR}{dt} &= \frac{dR}{d\theta}\frac{d\theta}{dt} \\ &= SR(t)\dot{\theta} \end{aligned} \tag{12}$$

Suppose we consider axis/angle representation with axis k and angle θ, $\quad \therefore R = R_{k,\theta}$. Putting this is equation 12 we get

$$\begin{aligned} \frac{dR}{dt} &= S(k)R_{k,\theta}\dot{\theta} \\ &= S(\dot{\theta}k)R_{k,\theta}(t) \\ \dot{R(t)} &= S(\omega(t))R(t) \end{aligned}$$

where $\omega(t)$ is the angular velocity of the rotating frame w.r.t fixed frame at time t.

$\omega^0_{1,2}$ represents angular velocity that corresponds to derivative of R^1_2 expressed in the $o_0x_0y_0z_0$ frame and $\omega_2 \equiv \omega^0_{0,2}$

5.2.1 Addition of Angular velocities

We now find expressions for compositions of angular velocities of two frames $o_1x_1y_1z_1$ and $o_2x_2y_2z_2$ w.r.t fixed frame $o_0x_0y_0z_0$. For now we assume that they all have same origin.

$$\begin{aligned} R^0_2 &= R^0_1R^1_2 \\ \dot{R}^0_2 &= \dot{R}^0_1R^1_2 + R^0_1\dot{R}^1_2 \\ S(\omega_2)R^0_2 &= S(\omega_1)R^0_1R^1_2 + R^0_1S(\omega^1_{1,2})R^1_2 \\ &= S(\omega_1)R^0_1R^1_2 + R^0_1S(\omega^1_{1,2}){R^0_1}^tR^0_1R^1_2 \\ &= S(\omega_1)R^0_1R^1_2 + S(R^0_1\omega^1_{1,2})R^0_1R^1_2 \\ S(\omega_2)R^0_2 &= S(\omega_1 + R^0_1\omega^1_{1,2})R^0_1R^1_2 \end{aligned}$$

$$\boxed{\omega_2^0 = \omega_{0,1}^0 + R_1^0 \omega_{1,2}^1}$$

Therefore, angular velocities can be added once they are expressed relative to same coordinate frame.
Extending what we prove above we get,

$$\omega_{0,n}^0 = \omega_{0,1}^0 + R_1^0 \omega_{1,2}^1 + R_2^0 \omega_{2,3}^2 + \ldots + R_{n-1}^0 \omega_{n-1,n}^{n-1}$$

$$\omega_{0,n}^0 = \omega_{0,1}^0 + \omega_{1,2}^0 + \omega_{2,3}^0 + \ldots + \omega_{n-1,n}^0$$

5.3 Linear Velocity of a pint attached to a moving coordinate frame

Suppose a point p is rigidly attached to $o_1x_1y_1z_1$ and $o_1x_1y_1z_1$ is rotating relative to $o_0x_0y_0z_0$. Then,

$$\begin{aligned}
p^0 &= R_1^0(t) p^1 \\
\dot{p}^0 &= \dot{R}_1^0 p^1 \\
&= S(\omega(t)) R_1^0 p^1 \\
&= S(\omega^0) p^0 \\
\dot{p}^0 &= \omega^0 \times p^0
\end{aligned}$$

Now suppose the $o_1x_1y_1z_1$ frame is translating and rotating relative to $o_0x_0y_0z_0$. Now we use the time dependent homogeneous transformation relating the two frames to find the linear velocity of point p. Let

$$H_1^0(t) = \begin{bmatrix} R_1^0(t) & o_1^0(t) \\ 0 & 1 \end{bmatrix}$$

Therefore,

$$\begin{aligned}
p^0 &= R_1^0 p^1 + o_1^0 \\
\dot{p}^0 &= S(\omega^0) R_1^0 p^1 = \dot{o}_1^0 \\
\dot{p}^0 &= \omega \times r + v
\end{aligned}$$

where r is the vector from o_1 to p expressed in frame $o_0x_0y_0z_0$ and v is the rate at which o_1 is moving.

If point p is also moving relative to $o_1x_1y_1z_1$ then add to the term v the term $R_1^0\dot{p}^1$, which is the rate of change of coordinate of p^1 expressed in $o_0x_0y_0z_0$ frame.

5.4 Derivation of Jacobian

Consider a n-link manipulator with joint variables $q_1, q_2, \ldots, q_n$.

$$T_n^0 = \begin{bmatrix} R_n^0(q) & o_n^0(q) \\ 0 & 1 \end{bmatrix}$$

We need to relate angular and linear velocity of end-effector to $q(\dot{t})$

Let $S(\omega_n^0) = \dot{R}_n^0 {R_n^0}^t$ define the angular velocity ω_n^0 of the end-effector and $v_n^0 = \dot{o}_n^0$ denote linear velocity.
We want to find J_v and J_ω such that,

$$v_n^0 = J_v \dot{q}$$

$$\omega_n^0 = J_\omega \dot{q}$$

where J_v and J_ω are $3 \times n$ matrices.
Writing these two equations together,

$$\xi = J\dot{q}$$

where $\xi = \begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix}$ is known as body velocity and $J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$ is known as manipulator jacobian/geometrical jacobain.

5.4.1 Angular Velocity

We know that if i^{th} joint is revolute, then its axis of rotation is z_{i-1}. Therefore,

$$\omega_i^{i-1} = \dot{q}_i z_{i-1}^{i-1} = \dot{q}_i k$$

If i^{th} joint is prismatic

$$\omega_i^{i-1} = 0$$

$$\begin{aligned} \therefore \omega_n^0 &= \rho_1 \dot{q}_1 k + \rho_2 R_1^0 \dot{q}_2 k + \cdots + \rho_n R_{n-1}^0 \dot{q}_n k \\ &= \sum_{i=1}^{n} \rho_i \dot{q}_i z_{i-1}^0 \end{aligned}$$

$\rho_i = 1$ if revolute and $\rho_i = 0$ if prismatic.

$$\boxed{J_\omega = [\rho_1 z_0 \ \rho_2 z_1 \cdots \rho_n z_{n-1}]}$$

5.4.2 Linear Velocity

By chain rule we can say that,

$$v = \dot{o}_n^0 = \sum_{i=1}^{n} \frac{\delta o_n^0}{\delta q_i} q_i$$

$$\therefore J_{v_i} = \frac{\delta o_n^0}{\delta q_i}$$

Therefore i^{th} column of J_v can be generated by holding all the joints except i fixed and actuating i^{th} joint with unit velocity.

Case-1: Prismatic Joints

$$\begin{aligned} T_n^0 &= T_{i-1}^0 T_i^{i-1} T_n^i \\ &= \begin{bmatrix} R_{i-1}^0 R_i^{i-1} R_n^i & R_i^0 o_n^i + R_{i-1}^0 o_i^{i-1} + o_{i-1}^0 \\ 0 & 1 \end{bmatrix} \\ \therefore o_n^0 &= R_i^0 o_n^i + R_{i-1}^0 o_i^{i-1} + o_{i-1}^0 \end{aligned} \tag{13}$$

If only joint i moves then o_n^i and o_{i-1}^0 remains constant. Also if joint i is prismatic then R_i^0 is also constant.
We also know that $o_i^{i-1} = (a_i c_i, a_i s_i, d_i)^t$

$$\begin{aligned} \frac{\delta o_n^0}{\delta q_i} &= \frac{\delta}{\delta d_i} R_{i-1}^0 o_i^{i-1} \\ &= R_{i-1}^0 \frac{\delta}{\delta d_i} \begin{bmatrix} a_i c_i \\ a_i s_i \\ d_i \end{bmatrix} \\ \dot{o_n^0} &= \dot{d}_i R_{i-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ &= \dot{d}_i z_{i-1}^0 \end{aligned}$$

$$\boxed{J_{v_i} = z_{i-1}}$$

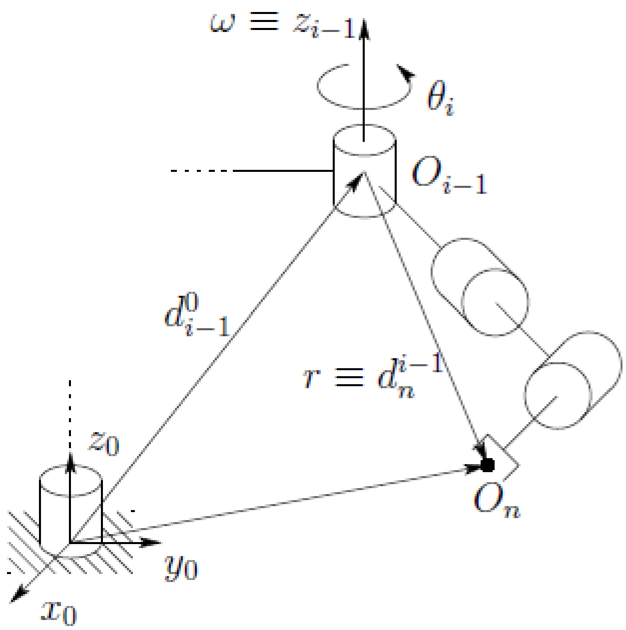
Case-2: Revolute Joint

Now in equation 13 R_i^0 is not constant w.r.t θ_i

$$
\begin{aligned}
\frac{\delta}{\delta\theta_i} o_n^0 &= \frac{\delta}{\delta\theta_i}(R_i^0 o_n^i + R_{i-1}^0 o_i^{i-1}) \\
&= \frac{\delta}{\delta\theta_i} R_i^0 o_n^i + R_{i-1}^0 \frac{\delta}{\delta\theta_i} o_i^{i-1} \\
\dot{o_n^0} &= \dot{\theta}_1 S(z_{i-1}^0) R_i^0 o_n^i + \dot{\theta}_i R_{i-1}^0 \frac{\delta}{\delta\theta} \begin{bmatrix} a_i c_i \\ a_i s_i \\ d_i \end{bmatrix} \\
&= \dot{\theta}_1 S(z_{i-1}^0) R_i^0 o_n^i + \dot{\theta}_i R_{i-1}^0 \begin{bmatrix} -a_i s_i \\ a_i c_i \\ 0 \end{bmatrix} \\
&= \dot{\theta}_1 S(z_{i-1}^0) R_i^0 o_n^i + R_{i-1}^0 S(k\dot{\theta}_i) o_i^{i-1} \\
&= \dot{\theta}_1 S(z_{i-1}^0) R_i^0 o_n^i + R_{i-1}^0 S(k\dot{\theta}_i) {R_{i-1}^0}^t R_{i-1}^0 o_i^{i-1} \\
&= \dot{\theta}_1 S(z_{i-1}^0) R_i^0 o_n^i + S(R_{i-1}^0 k\dot{\theta}_i) R_{i-1}^0 o_i^{i-1} \\
&= \dot{\theta}_1 S(z_{i-1}^0) R_i^0 o_n^i + \dot{\theta}_i S(z_{i-1}^0) R_{i-1}^0 o_i^{i-1} \\
&= \dot{\theta}_i (z_{i-1}^0)[R_i^0 o_n^i + R_{i-1}^0 o_i^{i-1}] \\
&= \dot{\theta}_i (z_{i-1}^0)(o_n^0 - o_{i-1}^0) \\
&= \dot{\theta}_i z_{i-1}^0 \times (o_n^0 - o_{i-1}^0)
\end{aligned}
$$

$$\boxed{\therefore J_{v_i} = z_{i-1}^0 \times (o_n^0 - o_{i-1}^0)} \tag{14}$$

Equation 14 can also be interpreted as



As can be seen in the figure, $o_n^0 - o_{i-1}^0 = r$ and $z_{i-1}^0 = \omega$ gives us the familiar expression

$v = \omega \times r$

5.5 Analytical Jacobian

Analytical jacobian is denoted by $J_a(q)$ and is based on a minimal representation for the orientation of the end-effector frame.
Let,

$$X = \begin{bmatrix} d(q) \\ \alpha(q) \end{bmatrix}$$

denote the end-effector pose, where $d(q)$ is the usual vector from the origin of the base frame to the origin of the end- effector frame and α denotes a minimal representation for the orientation of the end-effector frame relative to the base frame. For example, let $\alpha = [\phi, \theta, \psi]^t$ be a vector of Euler angles.

An expression of the form

$$\dot{X} = \begin{bmatrix} \dot{d} \\ \dot{\alpha} \end{bmatrix} = J_a(q)\dot{q}$$

defines the analytical jacobian

Note that $\dot{\alpha} \neq \omega$ and $\dot{d} = v$. Therefore the analytical jacobian will not give correct values for angular velocity of the end-effector.

Suppose, $R = R_{z,\psi} R_{y,\theta} R_{z,\phi}$ then,

$$\dot{R} = S(\omega)R$$

where

$$\begin{aligned} \omega &= \begin{bmatrix} c_\psi s_\theta \dot{\phi} - s_\psi \dot{\theta} \\ s_\psi s_\theta \dot{\psi} + c_\psi \theta \\ \dot{\psi} + c_\theta \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} c_\psi s_\theta & -s_\psi & 0 \\ c_\psi s_\theta & c_\psi & 0 \\ c_\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = B(\alpha)\dot{\alpha} \end{aligned}$$

Components of ω are known as nutation, spin and precession.

$$\begin{aligned} J(q)\dot{q} &= \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{d} \\ \omega \end{bmatrix} \\ &= \begin{bmatrix} \dot{d} \\ B(\alpha)\dot{\alpha} \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & B(\alpha) \end{bmatrix} J_a(q)\dot{q} \end{aligned}$$

$$\boxed{\therefore J_a(q) = \begin{bmatrix} I & 0 \\ 0 & B(\alpha)^{-1} \end{bmatrix} J(q)}$$

5.6 Singularities

$$\xi = J(q)\dot{q}$$
$$\Downarrow$$

all posible end-effector velocities are linear combination of the columns of J (the range space).

$$\xi = J_1\dot{q}_1 + J_2\dot{q}_2 + \cdots + J_n\dot{q}_n$$

Since $\xi \in \mathbb{R}^6$ it is necessary that J has linearly independent columns for the end-effector to achieve any arbitrary velocity.

Rank is number of linearly independent columns. Therefore when $rankJ = 6$ end-effector can execute any velocity.

For a matrix $J \in \mathbb{R}^{6\times n}$, it is always true that $rankJ \leq min(6, n)$
Therefore for an anthropomorphic arm having spherical wrist $rankJ \leq 6$.
Rank J always depends of the configuration of the arm.

Configurations for which rank J is less than the maximum value are known as singular configurations. Therefore to find the values of q for which singular configurations occur we solve the equation

$$detJ(q) = 0$$

To make this problem easier we decouple singularities, which can be done when there is a spherical wrist.

5.6.1 Decoupling of Singularities

The first is to determine so-called arm singularities, that is, singularities resulting from motion of the arm, which consists of the first three or more links, while the second is to determine the wrist singularities resulting from motion of the spherical wrist.

Suppose $n = 6$ i.e manipulator has 3 DOF arm and 3 DOF spherical wrist.
Partition J into 3×3 blocks,

$$J = [J_P | J_O] = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

Since final three joints are revolute,

$$J_O = \begin{bmatrix} z_3 \times (o_6 - o_3) & z_4 \times (o_6 - o_4) & z_5 \times (o_6 - o_5) \\ z_3 & z_4 & z_5 \end{bmatrix}$$

Since wrist axes intersect at common point o, we choose the coordinate frames so that,

$$o_3 = o_4 = o_5 = o_6 = o$$

Therefore J_O becomes,

$$J_O = \begin{bmatrix} 0 & 0 & 0 \\ z_3 & z_4 & z_5 \end{bmatrix}$$

Therefore,

$$J = \begin{bmatrix} J_{11} & 0 \\ J_{21} & J_{22} \end{bmatrix}$$

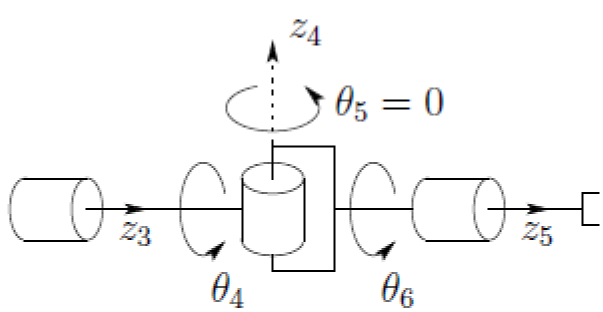
$$detJ = detJ_{11}detJ_{22}$$

Therefore the set of singular configurations of the manipulator is the union of the set of arm configurations satisfying $detJ_{11} = 0$ and the set of wrist configurations satisfying $detJ_{22} = 0$.

Note that this form of jacobian does not give a correct relation between end-effector velocity and joint velocities.

5.6.1.1 Wrist Singularities

Spherical wrist is in singular configuration when z_3, z_4 and z_5 are linearly independent.
This is only possible when



This is the only singularity of the spherical wrist, and is unavoidable without imposing mechanical limits on the wrist design to restrict its motion in such a way that z_3 and z_5 are prevented from lining up.

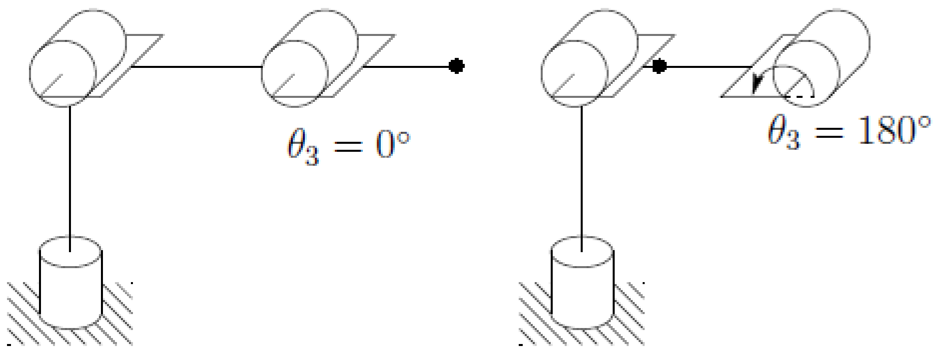
5.6.1.2 Arm Singularities

To investigate this we compute J_{11}.
Let us understand this topic by taking an example of the elbow manipulator.
It can be shown that for the elbow manipulator,

$$detJ_{11} = a_2a_3s_3(a_2c_2 + a_3c_{23})$$

$$detJ_{11} = 0$$

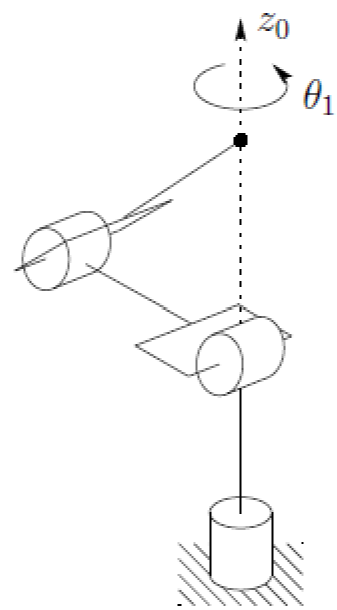
$$\Downarrow$$



$$s_3 = 0 \implies \theta_3 = 0, \pi$$

and

$$a_2c_2 + a_3c_{23} = 0$$



This occurs when wrist centre intersects the axis of base rotation, z_0.
This can be avoided by having an elbow or shoulder offset.

5.7 Inverse velocity and acceleration

Inverse velocity problem is to find the joint velocities from the given/desired end-effector velocity.

When J is a square and non-singular matrix

$$\dot{q} = J^{-1}\xi$$

If a manipulator has a non-square jacobian matrix then a solution of inverse velocity problem exists only when xi lies in the range space of of J i.e

$$rank J(q) = rank[J(q)|\xi]$$

If a solution exists then to find that solution we use psuedoinverse of J.

For $J \in \mathbb{R}^{m\times n}$, if $m < n$ and $rank J = m$, then $(JJ^t)^{-1}$ exists.
In this case $JJ^t \in \mathbb{R}^{m\times m}$, and has rank m.

$$\begin{aligned} \therefore (JJ^t)(JJ^t)^{-1} &= I \\ J[J^t(JJ^t)^{-1}] &= I \\ JJ^+ &= I \end{aligned}$$

Here $J^+ = J^t(JJ^t)^{-1}$ is called right psuedoinverse of J.

Note that, $J^+J \in \mathbb{R}^{n\times n}$, and in general, $J^+J \neq I$.
It can be shown that,

$$\dot{q} = J^+\xi + (I - J^+J)b$$

in which $b \in \mathbb{R}^n$ is an arbitrary vector.

All vector of the form $(I - J^+J)b$ lie in the null space of J, i.e if $\dot{q'}$ is a joint velocity vector such that $\dot{q'} = (I - J^+J)b$ then when joints move with velocity $\dot{q'}$, the end-effector remain fixed since $J\dot{q'} = 0$. If the goal is to minimize the resulting joint velocities, we choose $b = 0$
Therefore if q is a solution for $J\dot{q} = \xi$ then $\dot{q} + \dot{q'}$ is also a solution.

The right pseudoinverse of J can be easily computed using Singular Value Decomposition (SVD)

We can apply a similar approach as above when the analytical Jacobian is used in place of the manipulator Jacobian

$$J_a(q)\dot{q} = \dot{X}$$

Differentiating this we get,

$$\ddot{X} = J_a(q)\ddot{q} + \left(\frac{d}{dt}J_a(q)\right)\dot{q}$$

Thus, given a vector $\ddot{X}$ of end-effector accelerations, the instantaneous joint acceleration vector $\ddot{q}$ is given as a solution of the above equation.

Path and Trajectory Planning

6.1 Configuration Space

A complete specification of the location of every point on the robot is referred to as a configuration, and the set of all possible configurations is referred to as the configuration space and is denoted by $\mathcal{Q}$

For example for one link revolute arm $\mathcal{Q} = S'$ whre S' represents a unit circle.// For two link planar arm $\mathcal{Q} = S' \times S' = T^2$ where T^2 represents the torus.
For cartesian arm $\mathcal{Q} = \mathfrak{R}^3$
For a rigid object moving in a plane $q = (x, y, \theta)$ $\mathcal{Q} = \mathfrak{R}^2 \times SO(2)$

Now we describe collision of robot with obstacle in a workspce. We denote robot by $\mathcal{A}$, and the subset of workspace occupied by tje robot at configuration q by $\mathcal{A}(q)$. $\mathcal{O}_i$ are the obstacles in the workspace.

The set of configurations for which the robot collides with an obstacle is known as configuration space obstacle, defined as

$$\mathcal{QO} = \{q \in \mathcal{Q} | \mathcal{A}(q) \cap \mathcal{O} \neq \phi\}$$

here $\mathcal{O} = \cup \mathcal{O}_i$

$$\mathcal{Q}_{free} = \mathcal{Q} \backslash \mathcal{QO}$$

Path planning using this approach becomes very tedious, therefore we use potential fields to make our jobs easier.

6.2 Path Planning using Configuration Space Potential Fields

The robot is treated as a point particle in the configuration space, under the influence of an artificial potential field U. The field U is constructed so that the robot is attracted to the final configuration, q_{final}, while being repelled from the boundaries of $\mathcal{QO}$.

$$U(q) = U_{att}(q) + U_{rep}(q)$$

$$F(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q)$$

6.2.1 The Attractive Field

U_{att} should be monotonically increasing from q_{final}.
The simplest choice for such a field is a field that grows linearly with the distance from q_{final}, a so-called conic well potential. However, the gradient of such a field has unit magnitude everywhere but the origin, where it is zero. This can lead to stability problems. Therefore

to solve this problem we choose a field that increases quadratically from q_{final}.

Let $\rho_f(q)$ be the euclidean distance between q and q_{final} i.e $\rho_f(q) = \|q - q_{final}\|$. Then,

$$U_{att}(q) = \frac{1}{2}\zeta\rho_f^2(q)$$

where ζ is the parameter used to scale the effects of attractive potential.
For $q = (q^1, q^2, \ldots, q^n)^t$,

$$\begin{aligned}
\nabla U_{att}(q) &= \nabla\frac{1}{2}\zeta\rho_f^2(q) \\
&= \nabla\frac{1}{2}\zeta\|q - q_{final}\|^2 \\
&= \frac{1}{2}\zeta\nabla\sum(q^i - q^i_{final})^2 \\
&= \zeta(q - q_{final})
\end{aligned}$$

F converges linearly to 0 as $q \longrightarrow q_{final}$ which is what we want.
But now we have another problem. If q_{init} is very far away from q_{final} then this will result in very lrge attractive force.Therefore, for large distances we take conic field ensuring that ∇U is defined at the boundaries.

$$U_{att}(q) = \begin{cases} \frac{1}{2}\zeta\rho_f^2(q) & : \rho_f(q) \leq d \\ d\zeta\rho_f(q) - \frac{1}{2}\zeta d^2 & : \rho_f(q) > d \end{cases}$$

$$F_{att}(q) = -\nabla U_{att}(q) = \begin{cases} -\zeta(q - q_{final}) & : \rho_f(q) \leq d \\ \dfrac{-d\zeta(q - q_{final}}{\rho_f(q)} & : \rho_f(q) > d \end{cases}$$

6.2.2 The Repulsive Field

The repulsive field should repel the robot from obstacles, never allowing the robot to collide with an obstacle, and, when the robot is far away from an obstacle, that obstacle should exert little or no influence on the motion of the robot.

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\dfrac{1}{\rho(q)} - \dfrac{1}{\rho_0}\right)^2 & : \rho(q) \leq \rho_0 \\ 0 & : \rho(q) > \rho_0 \end{cases}$$

where ρ_0 is the distance of influence of the obstacle. $\rho_f(q)$ is the shortest distance from q to a configuration space obstacle boundary, and η is a scalar gain coefficient that determines the influence of the repulsive field.

$$F_{rep}(q) = \begin{cases} \eta\left(\dfrac{1}{\rho(q)} - \dfrac{1}{\rho_0}\right)\dfrac{1}{\rho^2(q)}\nabla\rho(q) & : \rho(q) \leq \rho_0 \\ 0 & : \rho(q) > \rho_0 \end{cases}$$

When $\mathcal{OQ}$ is convex, the gradient of distance to the nearest obstacle is given by

$$\nabla\rho(q) = \frac{q-b}{\|q-b\|}$$

where b is the point on the boundary of $\mathcal{OQ}$ nearest to q.

If $\mathcal{OQ}$ is not convex then ρ will not be necessarily differentiable everywhere which implies discontinuity in force vector.

6.2.3 Gradient Descent Planning

Starting at the initial configuration, take a small step in the direction of the negative gradient (i.e., in the direction that decreases the potential as quickly as possible). This gives a new configuration, and the process is repeated until the final configuration is reached.

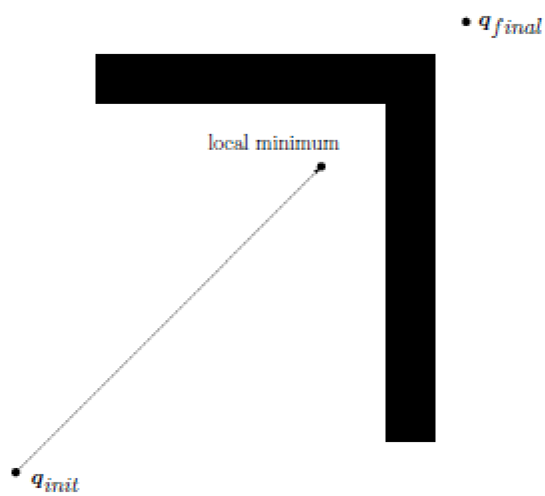
1. $q^0 \leftarrow q_{init}$, $i \leftarrow 0$

2. IF $q^i \neq q_{final}$
$q^{i+1} \leftarrow q^i + \alpha^i \frac{F(q^i)}{\|F(q^i)\|}$
ELSE return$< q^0, q^1, \cdots, q^i >$

3. GO TO 2.

??he final path consists of the sequence of iterates $< q^0, q^1, \cdots, qi >$. The value of the scalar α^i determines the step size at the i^{th} iteration.It is important that α^i be small enough that the robot is not allowed to "jump into" obstacles, while being large enough that the algorithm doesn't require excessive computation time.

Finally, it is unlikely that we will ever exactly satisfy the condition $q^i = q_{final}$. For this reason, this condition is often replaced with the more forgiving condition $\|q^i - q_{final}\| < \epsilon$, in which ϵ is chosen to be sufficiently small, based on the task requirements. This method



may not always work as local minima of potential function may come at some point other than q_{final} and that may cause problem.

6.3 Planning using workspace potential fields

6.3.1 Defining Workspace Potentials

In this section we will modify the potential field approach of Section 6.2 so that the potential function is defined on the workspace, $\mathcal{W}$, instead of the configuration space, $\mathcal{Q}$. Since $\mathcal{W}$ is a subset of a low dimensional space (either $\Re^2$ or $\Re^3$), it will be much easier to implement and evaluate potential functions over $\mathcal{W}$ than over $\mathcal{Q}$.

In $\mathcal{Q}$ we treated robot as an point particle but in $\mathcal{W}$ robot is an articulated arm having finite volume.
TO deal with this problem we select a subset of points on robot known as control points, and define potential for each of these points.

Let $A_{att} = \{a_1, a_2, \ldots, a_n\}$ be set of control points used to define the attractive potential fields. For n-link arm we can use COM of links or the origins of DH frames.

$a_i(q)$ is the position of i^{th} control point when the robot is at configuration q.
For each $a_i \in A_{att}$

$$U_{att_i}(q) = \begin{cases} \frac{1}{2}\zeta_i \|a_i(q) - a_i(q_{final})\|^2 & : \|a_i(q) - a_i(q_{final})\| \leq d \\ \\ d\zeta_i \|a_i(q) - a_i(q_{final})\| - \frac{1}{2}\zeta_i d^2 & : \|a_i(q) - a_i(q_{final})\| > d \end{cases}$$

When all points in A_{att} reach their global minima then configuration arm will be in q_{final}.

$$F_{att_i}(q) = -\nabla U_{att_i}(q) = \begin{cases} -\zeta_i(a_i(q) - a_i(q_{final})) & : \|a_i(q) - a_i(q_{final})\| \leq d \\ \\ \dfrac{-d\zeta_i(a_i(q) - a_i(q_{final}))}{\|a_i(q) - a_i(q_{final})\|} & : \|a_i(q) - a_i(q_{final})\| > d \end{cases}$$

Let $A_{rep} = a_1, \cdots, a_m$, and define the repulsive potential for a_j as

$$U_{rep_j}(q) = \begin{cases} \frac{1}{2}\eta_j \left(\dfrac{1}{\rho(a_j(q))} - \dfrac{1}{\rho_0} \right)^2 & : \rho(a_j(q)) \leq \rho_0 \\ \\ 0 & : \rho(a_j(q)) > \rho_0 \end{cases}$$

$\rho(a_j(q))$ is the shortest distance between control point a_j and any workspace obstacle. ρ_0 is the distance of influence of the obstacle.

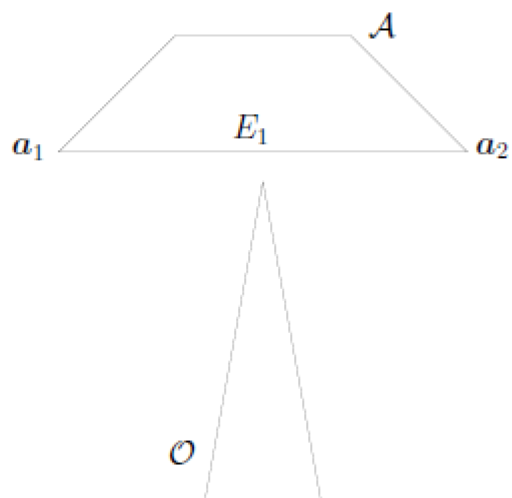
$$F_{rep_j}(q) = \begin{cases} \eta_j \left(\dfrac{1}{\rho(a_j(q))} - \dfrac{1}{\rho_0} \right) \dfrac{1}{\rho^2(a_j(q))} \nabla \rho(a_j(q)) & : \rho(a_j(q)) \leq \rho_0 \\ \\ 0 & : \rho(q) > \rho_0 \end{cases}$$

If b is a point on workspace obstacle boundary that is closest to repulsive control point a_j, then

$$\rho(a_j(q)) = \|a_j(q) - b\|$$

$$\nabla\rho(x)|_{x=a_j(q)} = \frac{a_j(q) - b}{\|a_j(q) - b\|}$$

It is important to note that this selection of repulsive control points does not guarantee that the robot cannot collide with an obstacle.



In this figure, the repulsive control points a_1 and a_2 are very far from the obstacle $\mathcal{O}$, and therefore the repulsive influence may not be great enough to prevent the robot edge E_1 from colliding with the obstacle. To cope with this problem, we can use a set of floating repulsive control points, $a_{float,i}$, typically one per link of the robot arm. The floating control points are defined as points on the boundary of a link that are closest to any workspace obstacle. Obviously the choice of the $a_{float,i}$ depends on the configuration q. For the example shown in Figure, $a_f loat$ would be located at the center of edge E_1, thus repelling the robot from the obstacle.

6.3.2 Mapping workspace forces to joint forces and torques

Suppose a force, $\mathcal{F}$ were applied to a point on the robot arm. Such a force would induce forces and torques on the robot's joints. If the joints did not resist these forces, a motion would occur. This is the key idea behind mapping artificial forces in the workspace to motions of the robot arm.

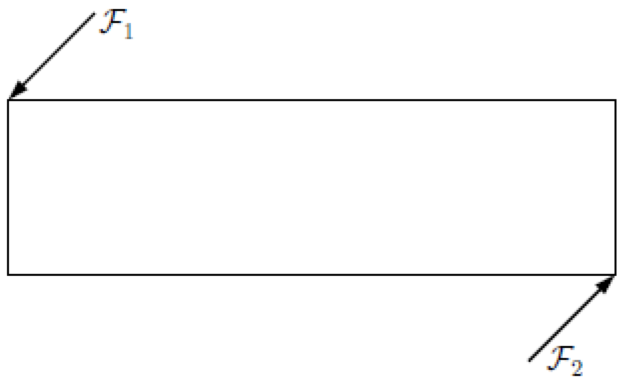
Let F denote the vector of joint torques (for revolute joints) and forces (for prismatic joints) induced by the workspace force.
Let $(\delta x, \delta y, \delta z)^t$ be a virtual displacement in the workspace and let δq be a virtual displacement

of the robot's joints.
Applying virtual work,

$$\mathcal{F}.(\delta x, \delta y, \delta z)^t = F.\delta q$$
$$\mathcal{F}^t(\delta x, \delta y, \delta z)^t = F^t \delta q$$
$$\begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} = J\delta q$$
$$\mathcal{F} J \delta q = F^t \delta q$$

$$\boxed{J^t \mathcal{F} = F}$$

This figure illustrates why forces must be mapped to the configuration space before they are added. The two forces illustrated in the figure are vectors of equal magnitude in opposite directions. Vector addition of these two forces produces zero net force, but there is a net moment induced by these forces.



6.3.3 Motion Planning Algorithm

Having defined a configuration space force, we can use the same gradient descent method for this case as in Section 6.2.3. As before, there are a number of design choices that must be made.

ξ_i controls the relative influence of the attractive potential for control point a_i. It is not necessary that all of the ξ_i be set to the same value. Typically, we weight one of the control points more heavily than the others, producing a "follow the leader" type of motion.

η_j controls the relative influence of the repulsive potential for control point a_j. As with the ξ_i it is not necessary that all of the η_j be set to the same value. In particular, we typically set the value of η_j to be much smaller for obstacles that are near the goal position of the robot.

As with the η_j , we can define a distinct ρ_0 for each obstacle. In particular, we do not want any obstacle's region of influence to include the goal position of any repulsive control point. We may also wish to assign distinct ρ_0's to the obstacles to avoid the possibility of overlapping regions of influence for distinct obstacles.

6.4 Probabilistic Roadmap Method

PRM's are one-dimensional roadmaps in $\mathcal{Q}_{free}$ that can be used to quickly generate paths. Once a PRM has been constructed, the path planning problem is reduced to finding paths to connect q_{init} and q_{final}.
There are four major steps for constructing a PRM which will be discussed below.

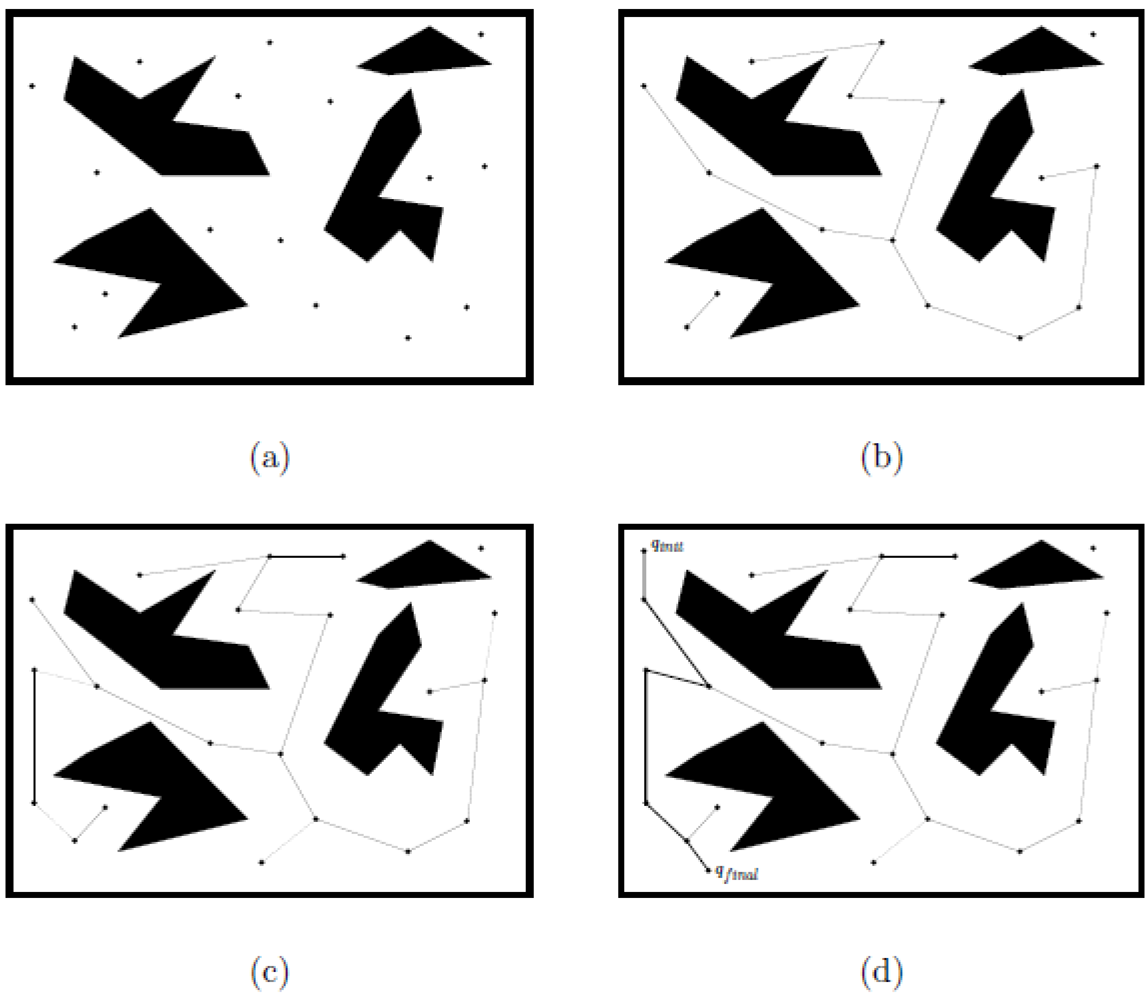


Figure 3: (a) A two-dimensional configuration space populated with several random samples (b) One possible PRM for the given configuration space and random samples (c) PRM after enhancement (d) path from q_{init} to q_{final} found by connecting q_{init} and q_{final} to the roadmap and then searching the roadmap for a path from q_{init} to q_{final}

6.4.1 Sampling the configuration space

The simplest way to generate sample configurations is to sample the configuration space uniformly at random (figure 3(a)). Sample configurations that lie in $\mathcal{QO}$ are discarded. A simple collision checking algorithm can determine when this is the case. The disadvantage of this approach is that the number of samples it places in any particular region of $\mathcal{Q}_free$ is proportional to the volume of the region. Therefore, uniform sampling is unlikely to place samples in narrow passages of $\mathcal{Q}_free$. In the PRM literature, this is refered to as the narrow passage problem. It can be dealt by using an enhancement phase during the construction of the PRM.

6.4.2 Connecting Pair of Configurations

Given a set of nodes that correspond to configurations, the next step in building the PRM is to determine which pairs of nodes should be connected by a simple path (figure 3(b)). The typical approach is to attempt to connect each node to it's k nearest neighbors, with k a parameter chosen by the user.Once pairs of neighboring nodes have been identified, a simple local planner is used to connect these nodes. Often, a straight line in configuration space is used as the candidate plan, and thus, planning the path between two nodes is reduced to collision checking along a straight line path in the configuration space. If a collision occurs on this path, it can be discarded.The simplest approach to collision detection along the straight line path is to sample the path at a sufficiently fine discretization, and to check each sample for collision. This method works, provided the discretisation is fine enough.

6.4.3 Enhancement

After the initial PRM has been constructed, it is likely that it will consist of multiple connected components. Often these individual components lie in large regions of $\mathcal{Q}_free$ that are connected by narrow passages in $\mathcal{Q}_free$. The goal of the enhancement process is to connect as many of these disjoint components as possible.

A common approach is to identify the largest connected component, and to attempt to connect the smaller components to it. The node in the smaller component that is closest to the larger component is typically chosen as the candidate for connection.

A second approach to enhancement is to add samples more random nodes to the PRM, in the hope of finding nodes that lie in or near the narrow passages. One approach is to identify nodes that have few neighbors, and to generate sample configurations in regions around these nodes.

6.4.4 Path Smoothing

After the PRM has been generated, path planning amounts to connecting q_{init} and q_{final} to the network using the local planner, and then performing path smoothing, since the resulting path will be composed of straight line segments in the configuration space. The simplest path smoothing algorithm is to select two random points on the path and try to connect them with the local planner. This process is repeated until until no significant progress is made.

6.5 Trajectory Planning

A trajectory is a function of time $q(t)$ such that $q(t_0) = q_{init}$ and $q(t_f) = q_{final}$.
$t_f - t_0$ is the time taken to complete the trajectory.

6.5.1 Trajectories for point to point motion

Without loss of generality, we will consider planning the trajectory for a single joint, since the trajectories for the remaining joints will be created independently and in exactly the same way.

Suppose,

$$q(t_0) = q_0$$
$$\dot{q}(t_0) = v_0$$
$$q(t_f) = q_f$$
$$\dot{q}(t_f) = v_f$$

6.5.1.1 Polynomial Trajectories

Consider a cubic trajectory satisfying the above conditions, 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,

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

The four constraints yields four equations in four unknowns,

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

These four equations can be combined into a single matrix equation,

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ v_0 \\ q_f \\ v_f \end{bmatrix} \tag{15}$$

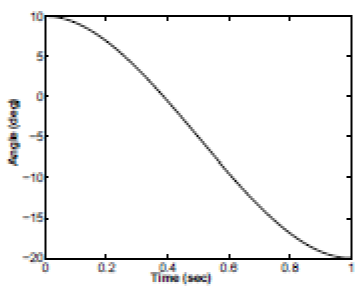
The determinant of the coefficient matrix in the above equation is equal to $(t_f - t_0)^4$ and, hence, the equation always has a unique solution provided a nonzero time interval is allowed for the execution of the trajectory.

As an example, we may consider the special case that the initial and final velocities are zeroand $t_0 = 0$ and $t_f = 1s$.
From equation 15 we get,

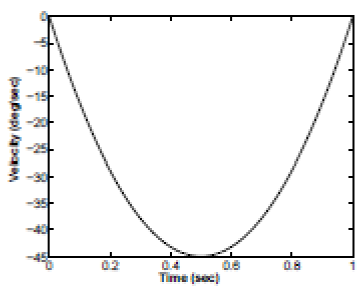
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ 0 \\ q_f \\ 0 \end{bmatrix}$$

Solving the above equation we get,

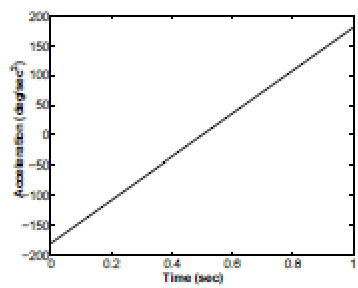
$$q_i(t) = q_0 + 3(q_f - q_0)t^2 - 2(q_f - q_0)t^3$$



(a)



(b)



(c)

Figure 4: (a) Cubic polynomial trajectory (b) Velocity profile for cubic polynomial trajectory (c) Acceleration profile for cubic polynomial trajectory

Figure 4(a) shows this trajectory with $q_0 = 10°, q_f = -20°$. The corresponding velocity and acceleration curves are given in Figures 4(b) and 4(c).

As can be see in in Figure 4, 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. Therefore consider,

$$q(t_0) = q_0$$
$$\dot{q}(t_0) = v_0$$
$$\ddot{q}(t_0) = \alpha_0$$
$$q(t_f) = q_f$$

$$\dot{q}(t_f) = v_f$$
$$\ddot{q}(t_f) = \alpha_f$$

In this case, we have six constraints (one each for initial and final configurations, initial and final velocities, and initial and final accelerations). Therefore we require a fifth order polynomial

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

We can solve for the coefficients by using the same procedure as we used to get equation 15.

6.5.1.2 Linear Segments with Parabolic Blends (LSPB)

Another way to generate suitable joint space trajectories is by so-called Linear Segments with Parabolic Blends or (LSPB) for short. This type of trajectory is appropriate when a constant velocity is desired along a portion of the path.To achieve this trajectory we specify the desired trajectory in three parts. The first part from time t_0 to time t_b is a quadratic polynomial. This results in a linear "ramp" velocity. At time t_b, called the blend time, the trajectory switches to a linear function. This corresponds to a constant velocity. Finally, at time $t_f - t_b$ the trajectory switches once again, this time to a quadratic polynomial so that the velocity is linear.We choose the blend time t_b so that the position curve is symmetric.

For convenience suppose that $t_0 = 0$ and $\dot{q}(t_f) = 0 = \dot{q}(0)$. Then between times 0 and t_b we have

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

so the velocity is

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

The constraints $q_0 = 0$ and $\dot{q}(0) = 0$ imply that

$$a_0 = q_0$$
$$a_1 = 0$$

At time t_b we want the velocity to equal a given constant, say V . Thus, we have

$$\dot{q}(t_b) = 2a_2 t_b = V$$
$$\Downarrow$$
$$a_2 = \frac{V}{2t_b}$$

Therefore the required trajectory between 0 and t_b is given as

$$\begin{aligned} q(t) &= q_0 + \frac{V}{2t_b} t^2 \\ &= q_0 + \frac{\alpha}{2} t^2 \\ \ddot{q} = \frac{V}{t_b} &= \alpha \end{aligned}$$

where α is the acceleration
Now, between time t_f and $t_f - t_b$, the trajectory is a linear segment (corresponding to a constant velocity V)

$$q(t) = a_0 + a_1 t = a_0 + Vt$$

Since, by symmetry,

$$q\left(\frac{t_f}{2}\right) = \frac{q_0 + q_f}{2}$$

we have

$$\frac{q_0 + q_f}{2} = a_0 + V\frac{t_f}{2}$$

$$\Downarrow$$

$$a_0 = \frac{q_0 + q_f - Vt_f}{2}$$

Since the two segments must "blend" at time t_b we require

$$q_0 + V\frac{t_b}{2} = \frac{q_0 + q_f - Vt_f}{2} + Vt_b$$

which gives upon solving for the blend time t_b

$$t_b = \frac{q_0 - q_f + Vt_f}{V}$$

Note that we have the constraint $0 < t_b \leq \frac{t_f}{2}$. This leads to the inequality

$$\frac{q_f - q_0}{t_f} < V \leq \frac{2(q_f - q_0)}{t_f}$$

Thus the specified velocity must be between these limits or the motion is not possible.

The portion of the trajectory between $t_f - t_b$ and t_f is now found by symmetry considerations. The complete LSPB trajectory is given by

$$q(t) = \begin{cases} q_0 + \dfrac{\alpha}{2}t^2 & : 0 \leq t \leq t_b \\[2ex] \dfrac{q_0 + q_f - Vt_f}{2} + Vt & : t_b < t \leq t_f - t_b \\[2ex] q_f - \dfrac{\alpha t_f^2}{2} + \alpha t_f t - \dfrac{a}{2}t^2 & : t_f - t_b < t \leq t_f \end{cases}$$

6.5.1.3 Minimum time trajectory

An important variation of this trajectory is obtained by leaving the final time tf unspecified and seeking the "fastest" trajectory between q_0 and q_f with a given constant acceleration α, that is, the trajectory with the final time t_f a minimum. This is sometimes called a Bang-Bang trajectory since the optimal solution is achieved with the acceleration at its maximum value $+\alpha$ until an appropriate switching time t_s at which time it abruptly switches to its minimum value $-\alpha$ (maximum deceleration) from t_s to t_f.

We consider the trajectory begins and ends at rest. We also know that,

$$t_s = \frac{t_f}{2}$$

Suppose V_s is velocity at time t_s

$$V_s = \alpha t_s$$

also,

$$t_s = \frac{q_0 - q_f + V_s t_f}{V_s}$$

Putting $t_s = \frac{t_f}{2}$ in this gives

$$\boxed{t_s = \sqrt{\frac{q_f - q_0}{\alpha}}}$$

6.5.2 Trajectories for path specified by via points

Now that we have examined the problem of planning a trajectory between two configuration, we generalize our approach to the case of planning a trajectory that passes through a sequence of configurations, called via points. Consider the simple of example of a path specified by three points, q_0, q_1, q_2, such that the via points are reached at times t_0, t_1 and t_2. We have the following constraints

$$q(t_0) = q_0$$
$$\dot{q}(t_0) = v_0$$
$$\ddot{q}(t_0) = \alpha_0$$
$$q(t_1) = q_1$$
$$q(t_2) = q_2$$
$$\dot{q}(t_2) = v_2$$
$$\ddot{q}(t_2) = \alpha_2$$

We have 7 constraints, therefore 6 degree polynomial can satisfy this

$$q(t) = a_6 t^6 + a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

Disadvantage of this method is that when the number of via points increase the method becomes very tedious.

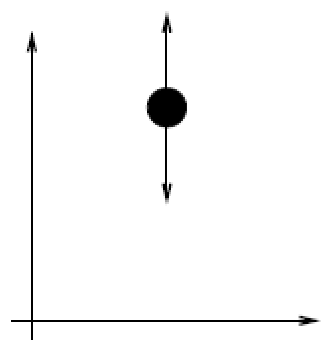
To solve this problem we use low order polynomials for trajectory segments between adjacent via points. These polynomials sometimes refered to as interpolating polynomials or blending polynomials. With this approach, we must take care that continuity constraints (e.g., in velocity and acceleration) are satisfied at the via points, where we switch from one polynomial to another.

Dynamics

7.1 The Euler-Lagrange equations

7.1.1 One-Dimensional System

Here we show how the Euler-Lagrange equations can be derived from Newton's Second Law for a single degree of freedom system consisting of a particle of constant mass m, constrained to move in the y-direction, and subject to a force f and the gravitational force mg, as shown in Figure.



From Newton's second law of motion we can say that,

$$m\ddot{y} = f - mg$$

LHS of this equation can be written as,

$$m\ddot{y} = \frac{d}{dt}(m\dot{y}) = \frac{d}{dt}\frac{\partial}{\partial \dot{y}}(\frac{1}{2}m\dot{y}^2) = \frac{d}{dt}\frac{\partial \mathcal{K}}{\partial \dot{y}}$$

RHS of the equation can be written as,

$$mg = \frac{\partial}{\partial y}(mgy) = \frac{\partial \mathcal{P}}{\partial y}$$

Now let us define,

$$\mathcal{L} = \mathcal{K} - \mathcal{P} = \frac{1}{2}m\dot{y}^2 - mgy$$

We also see that,

$$\frac{\partial \mathcal{L}}{\partial \dot{y}} = \frac{\partial \mathcal{K}}{\partial \dot{y}} \quad and \quad \frac{\partial \mathcal{L}}{\partial y} = -\frac{\partial \mathcal{P}}{\partial y}$$

Therefore we can write,

$$\boxed{\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{y}} - \frac{\partial \mathcal{L}}{\partial y} = f}$$

The function $\mathcal{L}$ is known as Lagrangian of the system and the equation is known as Euler-Lagrange Equation.

We can also derive these equations for n-dimensional system.
In general, any system, an application of the Euler-Lagrange equations leads to a system of n coupled, second order nonlinear ordinary differential equations of the form

$$\boxed{\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \qquad i = 1, \cdots, n}$$

The order, n, of the system is determined by the number of so-called generalized coordinates that are required to describe the evolution of the system. The n Denavit-Hartenberg joint variables serve as a set of generalized coordinates for an n-link rigid robot.

7.2 General Expressions for Kinetic and Potential Energy

The kinetic energy of a rigid object is the sum of two terms: the translational energy obtained by concentrating the entire mass of the object at the center of mass, and the rotational kinetic energy of the body about the center of mass.

$$\mathcal{K} = \frac{1}{2}mv^t v + \frac{1}{2}\omega^t \mathcal{I}\omega \tag{16}$$

where m is the total mass of the object, v and ω are the linear and angular velocity vectors, respectively, and $\mathcal{I}$ is a symmetric 3×3 matrix called the Inertia Tensor.

7.2.1 The Inertia Tensor

Since the ω in the above expression is with respect to inertial frame, the $\mathcal{I}$ should also be wrt inertial frame. The inertia tensor wrt inertial frame will depend on the configuration of the object. Let I be the inertia tensor in body attached frame (this won't change with the configuration of the object) then,

$$\mathcal{I} = RIR^t$$

Let the mass density of the object be represented as a function of position, $\rho(x, y, z)$. Then,

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

where,

$$
\begin{aligned}
I_{xx} &= \iiint (y^2 + z^2)\rho(x, y, z)dxdydz \\
I_{yy} &= \iiint (x^2 + z^2)\rho(x, y, z)dxdydz \\
I_{zz} &= \iiint (y^2 + x^2)\rho(x, y, z)dxdydz \\
I_{yx} = I_{xy} &= \iiint xy\rho(x, y, z)dxdydz \\
I_{zx} = I_{xz} &= \iiint xz\rho(x, y, z)dxdydz \\
I_{yz} = I_{zy} &= \iiint yz\rho(x, y, z)dxdydz
\end{aligned}
$$

7.2.2 KE of a n-link robot

We know that $v_i = J_{v_i}(q)\dot{q}$ and $\omega_i = J_{\omega_i}(q)\dot{q}$. Substitute these in equation 16

$$
\begin{aligned}
\mathcal{K} &= \frac{1}{2}\dot{q}^t(mJ_{v_i}(q)^t J_{v_i}(q) + J_{\omega_i}(q)^t R_i(q) I_i R_i(q)^t J_{\omega_i}(q))\dot{q} \\
\mathcal{K} &= \frac{1}{2}\dot{q}^t D(q)\dot{q}
\end{aligned}
$$

$D(q)$ is a symmetric positive definite matrix known as inertia matrix.

7.2.3 PE of a n-link robot

The potential energy of the i-th link can be computed by assuming that the mass of the entire object is concentrated at its center of mass and is given by

$$P_i = g^t r_{c_i} m_i$$

where g is the direction of gravity in inertial frame, r_{c_i} are coordinates of CM of link i.

$$\mathcal{P} = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} g^t r_{c_i} m_i$$

In the case that the robot contains elasticity, for example, flexible joints, then the potential energy will include terms containing the energy stored in the elastic elements.

7.3 Equations of Motion

In this section, we specialize Euler-Lagrange equations in two special conditions, KE is quadratic function of $\dot{q}$ and PE is independent of q. As we saw earlier all these conditions hold for a manipulator.

$$\begin{aligned}\mathcal{L} &= \mathcal{K} - \mathcal{P} \\ &= \frac{1}{2}\sum_{(} i,j)d_{ij}\dot{q}_i\dot{q}_j - P(q)\end{aligned}$$

From above equation we have that,

$$\frac{\partial L}{\partial \dot{q}_k} = \sum_j d_{kj}\dot{q}_j$$

and

$$\begin{aligned}\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} &= \sum_j d_{kj}\ddot{q}_j + \sum_j \frac{d}{dt}d_{kj}\dot{q}_j \\ &= \sum_j d_{kj}\ddot{q}_j + \sum_{i,j}\frac{\partial d_{kj}}{\partial q_i}\dot{q}_i\dot{q}_j\end{aligned}$$

Also,

$$\frac{\partial L}{\partial q_k} = \frac{1}{2}\sum_{i,j}\frac{\partial d_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j - \frac{\partial P}{\partial q_k}$$

Therefore Euler-Lagrange equation can be written as,

$$\sum_j d_{kj}\ddot{q}_j + \sum_{i,j}\left\{\frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i\dot{q}_j + \frac{\partial P}{\partial q_k} = \tau_k$$

By interchanging the order of summation and taking advantage of symmetry we get that,

$$\sum_{i,j}\left\{\frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i\dot{q}_j = \sum_{i,j}\frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i\dot{q}_j \tag{17}$$

The expression in the RHS of equation 17 is known as Christoffel symbols (of the first kind). Note that, for a fixed k, we have $c_{ijk} = c_{jik}$, which reduces the effort involved in computing these symbols by a factor of about one half. Now we also define,

$$\phi_k = \frac{\partial P}{\partial q_k}$$

Now we can write Euler-Lagrange equations as,

$$\boxed{\sum d_{kj}(q)\ddot{q}_j + \sum c_{ijk}(q)\dot{q}_i\dot{q}_j + \phi_k = \tau_k \quad k = 1, 2, \cdots, n}$$

In the above equation, there are three types of terms. The first involve the second derivative of the generalized coordinates. The second are quadratic terms in the first derivatives of q, where the coefficients may depend on q. These are further classified into two types. Terms involving a product of the type $\dot{q}_i^2$ are called centrifugal, while those involving a product of the type $\dot{q}_i\dot{q}_j$ where $i \neq j$ are called Coriolis terms. The third type of terms are those involving only q but not its derivatives, these are called potential energy terms.

We can also have a matrix form of this equation,

$$\boxed{D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau}$$

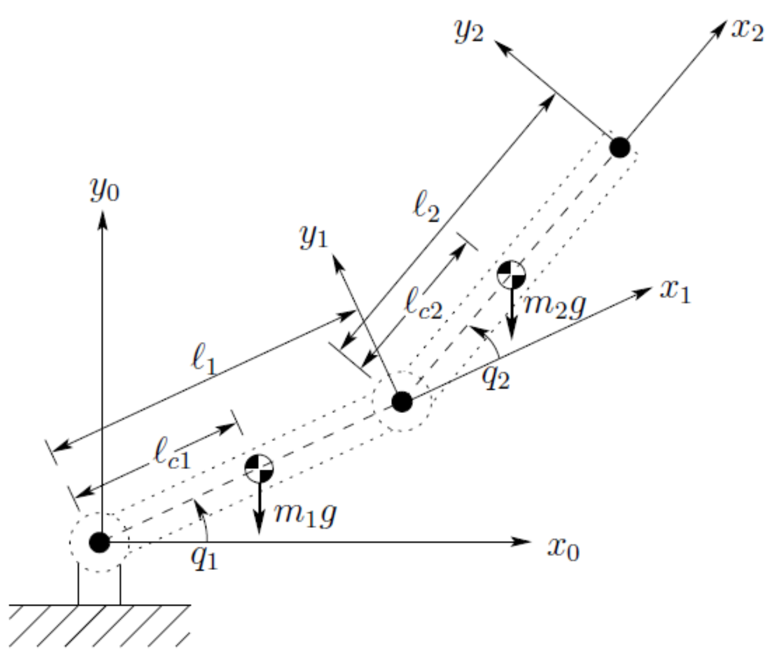
In special case when D is diagonal and independent of q, all c_{ijk} terms become zero and d_{kj} becomes constant and not equal to zero only if $k = j$. Therefore equations decouple into the form,

$$d_{kk}\ddot{q} - \phi_k(q) = \tau_k$$

Now let us apply this discussion to study specific robot configurations.

7.4 Some Common Configurations

7.4.1 Planar Elbow Manipulator



For $i = 1, 2, q_i$ denotes the joint angle, which also serves as a generalized coordinate; m_i denotes the mass of link i, l_i denotes the length of link i; l_{ci} denotes the distance from the previous joint to the center of mass of link i; and I_i denotes the moment of inertia of link i about an axis coming out of the page, passing through the center of mass of link i.

$$J_{v_{c1}} = \begin{bmatrix} -l_{c1}\sin q_1 & 0 \\ l_{c1}\cos q_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$J_{v_{c2}} = \begin{bmatrix} -l_{c2}\sin(q_1+q_2) - l_1\sin q_1 & -l_{c2}\sin(q_1+q_2) \\ l_{c2}\cos(q_1+q_2) + l_1\cos q_1 & l_{c2}\cos(q_1+q_2) \\ 0 & 0 \end{bmatrix}$$

Since this system is fairly easy we can directly calculate ω of each frame (without using jacobian) wrt inertial frame.

$$\omega_1 = \dot{q_1}k \quad \omega_2 = (\dot{q_1} + \dot{q_2})k$$

When expressed in base inertial frame, ω_i is aligned with k, therefore the product $\omega_i^t I \omega_i$ reduces to $(I_{33})_i \times \omega_i^2$. Let us represent $(I_{33})_i$ by I_i.

$$D(q) = m_1 J_{v_{c1}}^t J_{v_{c1}} + m_2 J_{v_{c2}}^t J_{v_{c2}} + I_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + I_2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Simplify this to get,

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

Now we find c_{ijk}'s

$$\begin{aligned} c_{111} &= \frac{1}{2}\frac{\partial d_{11}}{\partial q_1} = 0 \\ c_{121} = c_{211} &= \frac{1}{2}\frac{\partial d_{11}}{\partial q_2} = -m_2 l_1 l_{c2}\sin q_2 = h \\ c_{221} &= \frac{\partial d_{12}}{\partial q_2} - \frac{1}{2}\frac{\partial d_{22}}{\partial q_1} = h \\ c_{112} &= \frac{\partial d_{21}}{\partial q_1} - \frac{1}{2}\frac{\partial d_{11}}{\partial q_2} = -h \\ c_{122} = c_{212} &= 0 \\ c_{222} &= 0 \end{aligned}$$

Now we find potential energy terms,

$$P = (m_1 l_{c1} + m_2 l_1) g \sin q_1 + m_2 l_{c2} g \sin(q_1 + q_2)$$

Therefore

$$\phi_1 = \frac{\partial P}{\partial q_1} \quad \phi_2 = \frac{\partial P}{\partial q_2}$$

Therefore equations of motion are

$$\begin{aligned} d_{11}\ddot{q_1} + d_1 2\ddot{q_2} + c_{121}\dot{q_1}\dot{q_2} + c_{211}\dot{q_1}\dot{q_2} + c_{221}\dot{q_2}^2 + \phi_1 &= \tau_1 \\ d_{21}\ddot{q_1} + d_{22}\ddot{q_2} + c_{112}\dot{q_1}^2 + \phi_2 &= \tau_2 \end{aligned}$$

References

[1] *Robot Modeling and Control*, Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, John Wiley and Sons, Inc. New York, First Edition