Constrained Kinematics
Typically, mechanical systems are made by connecting bodies (links) through sequence of joints. Each joint allows specific degrees of freedom and restrict the other degrees of freedom.

For example, a **hinge (revolute) joint** allows rotation and restricts the remaining degrees of freedom (5 in the 3-D space and 2 in a plane).

In the following we will explore the constraints associated with some common joints.
Examples of Planar Constraints

Revolute (Hinge) Joint
A revolute joint ensures relative rotation between the two joined bodies, i.e.
\[ r^i_p = r^j_p \]
This formulation will yield two equations that correspond to the x and y components.
However,
\[ \theta^i \neq \theta^j \]

Prismatic Joint
On the other hand, prismatic joint allows relative translation between two connected bodies. Relative rotation is however restricted.
These relations can be represented as follows:
1. The two bodies always rotate together
   \[ \theta^i - \theta^j = c \]
2. The dot product of these two vectors is always zero:
   - Line of action (axis) of the prismatic joint
   - A vector is one of the two bodies that is normal to this line of action:
     \[ h^i \cdot r^{ij}_p = 0 \]
     In matrix notation, the above equation becomes:
     \[ h^i^T r^{ij}_p = 0 \]
While easier formulation exist especially in the case when one of the two bodies is completely fixed, the above formulation is general and works when both bodies are moving.

We will expand on these formulations later.

Class Exercise: Derive the constraint equations for a wheel rolling on a flat ground
**Computational Kinematic Approach**

- Classical kinematics can be used to accurately formulate equations describing the kinematics of variables of a machine in terms of its input motion.
- Formulating these equations required a prior knowledge of the geometry and sequence of joints within the machine.
- However, in any computer-aided software, the user creates a machine on the spot. In this case, classical approach cannot be used. Instead, a computational approach can be general enough to solve the kinematics of any machine.
- Kinematic analysis can be viewed as solving a set of algebraic equations that describes the joint connectivity of a certain machine.
- Another approach is to combine the equations and solve them.
- The resulting system of the coupled equations can be solved using numerical techniques.
- This is the technique used by most computer simulation software packages.
The position of a body can be described using $R^i$ and $\theta$ (Absolute Cartesian Coordinates)

$$r^i_p = R^i + A^i \bar{u}^i_p$$

- $R^i$: global position vector of the reference point
- $A^i$: transformation matrix from the body to global coordinate systems

$$R^i = \begin{bmatrix} R^i_x \\ R^i_y \end{bmatrix}$$

$$\bar{u}^i_p = \begin{bmatrix} \bar{x}^i_p \\ \bar{y}^i_p \end{bmatrix}$$

$$A^i = \begin{bmatrix} \cos \theta^i & -\sin \theta^i \\ \sin \theta^i & \cos \theta^i \end{bmatrix}$$
Extension to Multibody System

A multi-body system consisting of $n_b$ unconstrained rigid bodies has $3n_b$ independent generalized coordinates that are described using the vector $q$

$$ q = \begin{bmatrix} R_1^x & R_1^y & \theta_1 & R_2^x & R_2^y & \theta_2 & \cdots & \cdots & R_{nb}^x & R_{nb}^y & \theta_{nb} \end{bmatrix}^T $$

$$ q = \begin{bmatrix} R_1^1 & \theta_1 & R_1^2 & \theta_2 & \cdots & \cdots & \cdots & \cdots & R_{nb}^1 & \theta_{nb} \end{bmatrix}^T $$

$$ q = [q^1 \ q^2 \ \cdots \ \cdots \ \cdots \ \cdots \ q^{nb}]^T $$

where,

$$ q^i = \begin{bmatrix} R^i \\ \theta^i \end{bmatrix} $$
Kinematic Constraints
The relation between two bodies $i$ and $j$ can be related by the following equation

$$r^i_p - r^j_p = f(t)$$

Or,

$$(R^i + A^i \bar{u}^i_p) - (R^j + A^j \bar{u}^j_p) = f(t)$$

$f(t)$ is a known function of time
Ground Constraints

A body that has zero degrees of freedom (ground or fixed link)

\[ q^i = c \]

It is convenient to have the coordinates of frame \( i \) matching those of the global frame or,

\[
\begin{pmatrix}
R^i_x \\
R^i_y \\
\theta^i
\end{pmatrix} = 0
\]
Revolute Joint Constraint

Revolute joint ensures relative rotation between the two joined bodies, i.e.

\[ r^i_P = r^j_P \]

\[ \theta^i \neq \theta^j \]

Therefore,

\[ (R^i + A^i \bar{u}^i_P) = (R^j + A^j \bar{u}^j_P) \]

or,

\[ (R^i + A^i \bar{u}^i_P) - (R^j + A^j \bar{u}^j_P) = 0 \]

We end with one matrix equation (two scalar equations).

\[
\begin{bmatrix}
R^i_x \\
R^i_y
\end{bmatrix} + \begin{bmatrix}
\cos \theta^i & -\sin \theta^i \\
\sin \theta^i & \cos \theta^i
\end{bmatrix} \begin{bmatrix}
\bar{x}^i_P \\
\bar{y}^i_P
\end{bmatrix} - \begin{bmatrix}
R^j_x \\
R^j_y
\end{bmatrix} - \begin{bmatrix}
\cos \theta^j & -\sin \theta^j \\
\sin \theta^j & \cos \theta^j
\end{bmatrix} \begin{bmatrix}
\bar{x}^j_P \\
\bar{y}^j_P
\end{bmatrix} = 0
\]
Prismatic Joint

Prismatic joint allows relative translation between the connected bodies. Relative rotation is however restricted as two bodies have to rotate together, which means,

\[ \theta^i - \theta^j = c \]

On the other hand, one of the two bodies is free to slide in and out of the other one along a specified line of action. Geometrically, this can be represented using two vectors:

- The line of action connecting two points on \( j \) and \( i \) respectively as in the figure above and
- A vector in Body \( i \) that is normal to this line of action

Regardless of the orientations of these two bodies, these two vectors will always remain normal to each other. This relation can be represented using dot product of these two vectors:

\[ h^T i r_{ij}^p = 0 \]

where,

\[ h_i = A^i \left( \bar{u}_i p - \bar{u}_i q \right) \]

\[ r_{ij}^p = r_i^p - r_j^p = (R^i + A^i \bar{u}_i p) - (R^j + A^i \bar{u}_j p) \]

These equations can be combined as,

\[
\begin{bmatrix}
\theta^i - \theta^j - c \\
h^T i r_{ij}^p
\end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]
Example
Derive the algebraic kinematic constraint equations of this two-link manipulator.

Assumptions:
• The two links are uniform rods.
• Each link has its frame at its geometric center, representing the center of mass. This is typical when modeling dynamic systems.
• For convenience, the frame of link 1 coincides with the global frame.

Solution:
• Since frame 1 is fixed and coincides with the global frame
  \[
  \begin{pmatrix}
  R_1^x \\
  R_1^y \\
  \theta_1
  \end{pmatrix}
  = 0
  \]

• Joint O connects links 1 and 2. It is easy to see that frame 2 traces a circle whose center is O. Therefore, this hinge can be described using this equation:
  \[
  (R^2 + A^2 \bar{u}_o) - R^1 = 0
  \]
  \[
  \begin{pmatrix}
  R^2_x \\
  R^2_y
  \end{pmatrix}
  + \begin{pmatrix}
  \cos \theta^2 & -\sin \theta^2 \\
  \sin \theta^2 & \cos \theta^2
  \end{pmatrix}
  \begin{pmatrix}
  -l^2 \\
  0
  \end{pmatrix}
  = 0
  \]

Remember that \( R^1 = 0 \), which means this equation becomes,
  \[
  \begin{pmatrix}
  R^2_x \\
  R^2_y
  \end{pmatrix}
  + \begin{pmatrix}
  \cos \theta^2 & -\sin \theta^2 \\
  \sin \theta^2 & \cos \theta^2
  \end{pmatrix}
  \begin{pmatrix}
  -l^2 \\
  0
  \end{pmatrix}
  = 0
  \]
The same process can be repeated for joint A by moving:

1. frame 2 a distance of \(\frac{l_2}{2}\) (forward) and

2. frame 3 a distance of \(\frac{-l_3}{2}\) (backward) to have both frames meet at joint A.

\[
(R^3 + A^3 \bar{u}_A^3) - (R^2 + A^2 \bar{u}_A^2) = 0
\]

These kinematic constraint equations can be written in a vector form as:

\[
C(q^1, q^2, q^3) = \begin{vmatrix}
R^1_x \\
R^1_y \\
\theta^1 \\
R^2_x - \frac{l^2 \cos \theta^2}{2} \\
R^2_y - \frac{l^2 \sin \theta^2}{2} \\
R^3_x - \frac{l^3 \cos \theta^3}{2} - R^2_x - \frac{l^2 \cos \theta^2}{2} \\
R^3_y - \frac{l^3 \sin \theta^3}{2} - R^2_y - \frac{l^2 \sin \theta^2}{2} \\
\end{vmatrix} = \begin{vmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{vmatrix}
\]

where \(C=[C_1, C_2, \ldots, C_7]^T\) is the vector of algebraic constraints that describe this system.

Since the system has nine absolute coordinates.

Therefore, the number of degrees of freedom is equal to two

**Homework**

- Derive the kinematic constraint equations for the machines in Figures P3.6 and P3.24.
Computational Method in Kinematics

- The presented approach is based on exploiting the idea of *kinematic constraints* numerically.
- This way we avoid deriving specific solution for every mechanical system.

Introduction

A multi-body system consisting of \( n_b \) unconstrained rigid bodies has \( 3n_b \) independent generalized coordinates that are described using the vector \( \mathbf{q} \)

\[
\mathbf{q} = \begin{bmatrix}
  R_{1x} & R_{1y} & \theta_1 & R_{2x} & R_{2y} & \theta_2 & \ldots & R_{nbx} & R_{nby} & \theta_{nb}
\end{bmatrix}^T
\]

where,

\[
q^i = \begin{bmatrix} R^i \theta^i \end{bmatrix}
\]

The vector of algebraic kinematic constraints that describe this system

\[
\mathbf{C}(\mathbf{q},t) = [C_1(\mathbf{q},t), C_2(\mathbf{q},t), \ldots, C_{n_c}(\mathbf{q},t)]^T
\]

\( n_c \) is the total number of constraints
Categories of Mechanical Systems

Mechanical systems can be divided into:

1. **Dynamically-Driven:** The number of the constraint equations is less than the number of the system coordinates \((n_c < n)\).
   
   Dynamics forces are needed to perform the analysis.

   *Example:* A ball falling toward ground.
   
   - No driving input
   - No kinematic constraint
   - The result is 0 equations that are less than the number of degrees of freedom of the system \((3)\)

2. **Kinematically-Driven:** The number of the linearly independent constraint equations is equal to the number of the system coordinates \((n_c = n)\).

   *Example:* The two-link manipulator

   - Adding the number of driving inputs to the constraints equations \((2 \text{ inputs to } 7 \text{ constraints})\).
   - The result is 9 equations that are equal the number of degrees of freedom of the system \((3 \times 3)\)

*This chapter focuses on the Kinematically-Driven systems only.*

*Dynamically-Driven system will be analyzed later.*
Example on Formulating a Kinematically Driven Mechanical Systems:

Two additional inputs are needed: $\theta^2$ and $\theta^3$ for links 2 and 3 respectively.
If both links are driven using constant speeds: $\omega^2$ and $\omega^3$.

\[
C(q^1, q^2, q^3) = \begin{bmatrix}
R^1_x \\
R^1_y \\
\theta^1 \\
R^2_x - \frac{l^2 \cos \theta^2}{2} \\
R^2_y - \frac{l^2 \sin \theta^2}{2} \\
R^2_x + \frac{l^2 \cos \theta^2}{2} - R^3_x + \frac{l^3 \cos \theta^3}{2} \\
R^2_y + \frac{l^2 \sin \theta^2}{2} - R^3_y + \frac{l^3 \sin \theta^3}{2} \\
\theta^2 - \theta^2_o - \omega^2 t \\
\theta^3 - \theta^3_o - \omega^3 t
\end{bmatrix} = \begin{bmatrix} 0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]
Homework

- Derive the kinematic constraint equations including driving input constraints for the machines in Figures P3.3 and P3.4.
Position Analysis

- The kinematic constraints equations are of trigonometric natures and are generally nonlinear.
- While a closed form solution can be obtained in many cases, it has to be performed at each case separately.
- A computer-driven (computational) approach requires solving these equations using a numerical method.
- A typical approach is to use *Newton-Raphson Algorithm*.

Sir Isaac Newton (1642-1727)  
Joseph Raphson (1648-1715)  

http://en.wikipedia.org/wiki/Raphson
Newton-Raphson Algorithm

Newton-Raphson’s algorithm is based on Taylor’s Theorem:

\[
f(x + \Delta x) = f(x) + \frac{df(x)}{dx} \Delta x + \frac{1}{2!} \frac{d^2 f(x)}{dx^2} (\Delta x)^2 + \cdots + \frac{1}{(n-1)!} \frac{d^{n-1} f(x)}{dx^{n-1}} (\Delta x)^{n-1}
\]

- Taylor’s Theorem can be expanded to multivariable functions after expressing it in matrix notation as follows:

-Brook Taylor (1685-1731)

http://en.wikipedia.org/wiki/Brook_Taylor

Hessian Matrix: \( m \times m \)
If we have a set of functions, we can apply Taylor’s Theorem to them by expanding the above equation,

\[ C(q_i + \Delta q_i, t) = C(q_i, t) + C_{q_i} \Delta q_i + \cdots \text{(higher order terms)} \]

\[ \Delta q = [\Delta q^1 \Delta q^2 \cdots \cdots \cdots \cdots \cdots \Delta q^n]^T \]

\( C_{qi} \) is the **Jacobian Matrix**, which can be defined as,

\[
C_{qi} = \begin{bmatrix}
\frac{\partial C_1}{\partial q_1} & \frac{\partial C_1}{\partial q_2} & \cdots & \frac{\partial C_1}{\partial q_n} \\
\frac{\partial C_2}{\partial q_1} & \frac{\partial C_2}{\partial q_2} & \cdots & \frac{\partial C_2}{\partial q_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial C_n}{\partial q_1} & \frac{\partial C_n}{\partial q_2} & \cdots & \frac{\partial C_n}{\partial q_n}
\end{bmatrix}
\]

**Note:** \( C_{qi} \) is a square matrix for a *kinematically-driven* system since \( n=n_c \)

---

Carl Gustav Jacob Jacobi (1804-1851)

If the constraint equations are *linearly independent*\(^1\), \(C\) is a nonsingular matrix.

Assuming \(q_i + \Delta q_i\) to be the exact solution, then

\[ C(q_i + \Delta q_i, t) = 0 \]

Therefore,

\[ C(q_i, t) + C_{q_i} \Delta q_i + \cdots = 0 \]

If the assumed solution is close enough to the correct solution, the second order and higher terms can be neglected, which means:

\[ C(q_i, t) + C_{q_i} \Delta q_i \approx 0 \]

\[ C_{q_i} \Delta q_i = -C(q_i, t) \]

\[ \Delta q_i = -C_{q_i}^{-1} C(q_i, t) \]

Remember that the constraint Jacobian matrix \(C_{q_i}\) is assumed to be *nonsingular*, i.e., an inverse of \(C_{q_i}\) exists.

The equation can be then solved for the vector of the *Newton Differences*, \(\Delta q_i\)

The vector is updated as,

\[ q_{i+1} = q_i + \Delta q_i \]

where, \(i\) is the iteration number. The updated vector can be used to reconstruct the equation above.

The process continues until:

\[ |\Delta q_i| < \epsilon_1 \text{ or,} \]

\[ |C(q_i, t)| < \epsilon_2 \]

The above termination criteria means that search stops because either:

1. The changes of the vector have become extremely small or,
2. The Kinematic Constraints matrix approaches zero

*Note*: \(\epsilon_1\) is a vector while \(\epsilon_2\) is a one value.

---

\(^1\) *Linearly independent constraints* is a set of unique constraints where no constraint can be expressed in terms of the other ones.
Example:

Two additional inputs are needed: $\theta^2$ and $\theta^3$ for links 2 and 3 respectively. If both links are driven using constant speeds: $\omega^2$ and $\omega^3$

$$C(q^1, q^2, q^3) = \begin{bmatrix}
R^1_x \\
R^1_y \\
\theta^1 \\
\frac{R^2_x - \frac{l^2 \cos \theta^2}{2}}{2} \\
\frac{R^2_y - \frac{l^2 \sin \theta^2}{2}}{2} \\
\frac{R^2_x + \frac{l^2 \cos \theta^2}{2} - R^3_x + \frac{l^3 \cos \theta^3}{2}}{2} \\
\frac{R^2_y + \frac{l^2 \sin \theta^2}{2} - R^3_y + \frac{l^3 \sin \theta^3}{2}}{2} \\
\theta^2 - \theta^2_o - \omega^2 t \\
\theta^3 - \theta^3_o - \omega^3 t
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}$$

$$q = [R^1_x \ R^1_y \ \theta^1 \ R^2_x \ R^2_y \ \theta^2 \ R^3_x \ R^3_y \ \theta^3]^T$$
The Jacobian matrix is,

\[
C_{qi} = \begin{bmatrix}
\frac{\partial C_1}{\partial q_i} & \frac{\partial C_1}{\partial q_2} & \cdots & \frac{\partial C_1}{\partial q_n} \\
\frac{\partial C_2}{\partial q_i} & \frac{\partial C_2}{\partial q_2} & \cdots & \frac{\partial C_2}{\partial q_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial C_n}{\partial q_i} & \frac{\partial C_n}{\partial q_2} & \cdots & \frac{\partial C_n}{\partial q_n}
\end{bmatrix}
\]

In this case \(C_{qi}\) can be obtained by partial differentiation of the constraint matrix as follows:

\[
C_q = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & l^2 \sin \theta^2/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -l^2 \cos \theta^2/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -l^2 \sin \theta^2/2 & -1 & 0 & -l^3 \sin \theta^3/2 \\
0 & 0 & 0 & 0 & 0 & 0 & l^2 \cos \theta^2/2 & 0 & -1 & l^3 \cos \theta^3/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}
\]
Velocity Analysis
Differentiating the vector constraint equation

\[
\frac{dC(q, t)}{dt} = 0
\]

\[
\frac{\partial C}{\partial q} \frac{dq}{dt} + \frac{\partial C}{\partial t} = 0
\]

Or,

\[
C_q \dot{q} + C_t = 0
\]

\(C_q\) is the constraint Jacobian Matrix
\(C_t\) is the partial differentiation of \(C\) with respect to time:

\[
C_t = \begin{bmatrix}
\frac{\partial C_1}{\partial t} & \frac{\partial C_2}{\partial t} & \cdots & \frac{\partial C_n}{\partial t}
\end{bmatrix}^T
\]

Rearranging,

\[
C_q \dot{q} = -C_t
\]

Or,

\[
\dot{q} = -C_q^{-1}C_t
\]

You may want to notice the similarity between:

- Position formulation (\(\Delta q_i = -C_q^{-1}C(q_i, t)\)) and
- Velocity formulation (\(\dot{q} = -C_q^{-1}C_t\))

*Why? What is the difference?*
**Acceleration Analysis**

Differentiating the velocity equation

\[
\frac{d}{dt} \left( C_q(q, t)\dot{q} + C_t(q, t) \right) = 0
\]

\[
\frac{d}{dt} \left( C_q(q, t)\dot{q} \right) + \frac{d}{dt} \left( C_t(q, t) \right) = 0
\]

\[
\left( \left( C_q\dot{q} \right)_q \dot{q} + C_{qt}\dot{q} \right) + \left( C_{qt}\dot{q} + C_{tt} \right) = 0
\]

Rearranging,

\[
C_q\dddot{q} + \left( (C_q\dot{q})_q + C_{qt} \right) \ddot{q} + (C_{qt}\dot{q} + C_{tt}) = 0
\]

\[
C_q\dddot{q} + (C_q\dot{q})_q \ddot{q} + 2C_{qt}\dot{q} + C_{tt} = 0
\]

\[
C_q\dddot{q} = -(C_q\dot{q})_q \ddot{q} - 2C_{qt}\dot{q} - C_{tt} = Q_d
\]

Or,

\[
\dddot{q} = C_q^{-1} Q_d
\]
Homework

- Solve Problem 3.19

Problem 3.19

19. The motion of a rigid body $i$ is such that the global coordinates of point $P$ on the rigid body is given by $r^i_P = [vt \ 0]^T$, where $v$ is a constant. The angular velocity of the rigid body is assumed to be $\dot{\theta}^i = a_0 + a_1 t$. Derive an expression for the kinematic constraint equations of this system in terms of the absolute coordinates $R^i_x, R^i_y,$ and $\theta^i$. Assume that $\bar{u}_P = [0.3 \ 1.2]^T$ m. Also determine the first and the second derivatives of the constraint equations. Use the resulting equations to determine the velocities $\dot{R}^i_x, \dot{R}^i_y,$ and $\dot{\theta}^i$ and the accelerations $\ddot{R}^i_x, \ddot{R}^i_y,$ and $\ddot{\theta}^i$ at $t = 0$, and 2 s. Use the data $v = 5$ m/s, $a_0 = 0$, $a_1 = 15$ rad/s$^2$.

\[
\begin{align*}
   r^i_p &= R^i + A^i \bar{u}_p \\
   \{vt\} &= R^i + A^i \begin{bmatrix} 03 \\ 1.2 \end{bmatrix} \\
   R^i &= \begin{bmatrix} vt \\ 0 \end{bmatrix} - A^i \begin{bmatrix} 03 \\ 1.2 \end{bmatrix}
\end{align*}
\]

where,

\[
A^i = \begin{bmatrix}
   \cos \theta^i & -\sin \theta^i \\
   \sin \theta^i & \cos \theta^i
\end{bmatrix}
\]

Hint:

- Start by identifying the generalized coordinates
- Develop the kinematic constraints equations
- Once you have both velocity and acceleration equations will be easy to derive
Example
Derive the velocity and acceleration equations of this two-link manipulator

In this case, the generalized coordinates are,

$$ q = [R^1_x \ R^1_y \ \theta^1 \ R^2_x \ R^2_y \ \theta^2 \ R^3_x \ R^3_y \ \theta^3]^T $$

The kinematic constraint equations are,

$$ C(q^1, q^2, q^3) = \begin{cases} 
R^1_x \\
R^1_y \\
\theta^1 \\
R^2_x - \frac{l^2 \cos \theta^2}{2} \\
R^2_y - \frac{l^2 \sin \theta^2}{2} \\
R^2_x + \frac{l^2 \cos \theta^2}{2} - R^3_x + \frac{l^3 \cos \theta^3}{2} \\
R^2_y + \frac{l^2 \sin \theta^2}{2} - R^3_y + \frac{l^3 \sin \theta^3}{2} \\
\theta^2 - \theta^2_o - \omega^2 t \\
\theta^3 - \theta^3_o - \omega^3 t 
\end{cases} = \begin{bmatrix} 0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \end{bmatrix} $$
Velocity Analysis:

Remember that the velocity equation is:

\[ \dot{q} = -C_q^{-1} C_t \]

The Jacobian matrix becomes,

\[
C_q = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & \frac{l^2 \sin \theta^2}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -\frac{l^2 \cos \theta^2}{2} & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -\frac{l^2 \sin \theta^2}{2} & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & \frac{l^2 \cos \theta^2}{2} & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\]

In this case, \( C_t \) is simple since only the last two elements are explicit function of time.

\[
C_t = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & -\omega^2 & -\omega^3
\end{bmatrix}^T
\]
**Acceleration Analysis**

Remember that the acceleration equation is:

\[ \ddot{q} = C_q^{-1} Q_d \]

The vector \( Q_d \) is

\[ Q_d = -(C_q \dot{q})_q \dot{q} - 2C_q t \dot{q} - C_{tt} \]

Remember that \( C_t \) is,

\[ C_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -\omega^2 & -\omega^3 \end{bmatrix}^T \]

\( C_t \) matrix does not have any term that is in terms of time explicitly. Therefore,

\( C_{qt} = 0 \)
\( C_{tt} = 0 \)

Similarly, \( \dot{q} \) is not function of \( q \), which means that,

\( \dot{q}_q = 0 \)

The vector \( Q_d \) is reduced to,

\[ Q_d = -(C_q \dot{q})_q \dot{q} \]
Since,

\[
C_q \dot{q} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
\dot{R}_x^1 \\
\dot{R}_y^1 \\
\dot{\theta}_1^1 \\
\dot{R}_x^2 \\
\dot{R}_y^2 \\
\dot{\theta}_2^1 \\
\dot{R}_x^3 \\
\dot{R}_y^3 \\
\dot{\theta}_3^1 \\
\end{bmatrix}
\]

\[
(C_q \dot{q})_q \dot{q} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \begin{bmatrix}
\dot{R}_x^1 \\
\dot{R}_y^1 \\
\dot{\theta}_1^1 \\
\dot{R}_x^2 \\
\dot{R}_y^2 \\
\dot{\theta}_2^1 \\
\dot{R}_x^3 \\
\dot{R}_y^3 \\
\dot{\theta}_3^1 \\
\end{bmatrix}
\]

Therefore,
\[
(C_q \dot{q}) \dot{q} = \begin{bmatrix}
0 \\
0 \\
0 \\
\frac{\dot{\theta}^2 l^2 \cos \theta^2}{2} \\
\frac{\dot{\theta}^2 l^2 \sin \theta^2}{2} \\
-\frac{\dot{\theta}^2 l^2 \cos \theta^2}{2} - \frac{\dot{\theta}^3 l^3 \cos \theta^3}{2} \\
-\frac{\dot{\theta}^2 l^2 \sin \theta^2}{2} - \frac{\dot{\theta}^3 l^3 \sin \theta^3}{2} \\
0 \\
0 
\end{bmatrix}
\]

While we are not concerned with the closed form solutions, it may informative to have a look at the result of multiplying these two matrices.

The results show the normal acceleration form of the constraint equations for joints O and A respectively.

\[
\ddot{q} = C_q^{-1} Q_d = C_q^{-1}
\]

**Homework**
- Based on the kinematic constraint equations that you already derived, develop the velocity and acceleration equations in Figures P3.3 and P3.4.
Special Cases in Kinematic Analysis

Degenerate (Extreme) Configurations

- Let’s start by discussing *degenerate configurations*, which are the ones when a mechanical system loses one or more degrees of freedom.
- Extreme position is associated with a reduction in the order of the geometry of the machines or *degeneracy* (e.g., triangular to straight line or quadrilateral to triangular).
- Extreme positions are associated with the loss of (zero) velocity. This should make sense as the machine is changing direction.
- Here are some examples:

```plaintext
\begin{align*}
\text{General Position} \\
\text{Extreme Positions}
\end{align*}
```
General Position

Extreme Positions
Singular (Lockup) Configurations

Singularities

- Singular (lockup) positions are associated with infinite joint velocities.
- We are also interested in positions that are close to singularity as they are associated with excessive velocities that may damage the machine or result in jerky motion.

If \( \cos \theta^3 = 0 \) i.e. \( \theta^3 = \frac{\pi}{2} \) or \( \frac{3\pi}{2} \)

Then, \( \{\dot{x}^4_B\} = \{\infty\} \)

The figure below shows the only time this case is feasible.