Kinematically Driven Systems

Chandramouli P

Indian Institute of Technology Madras

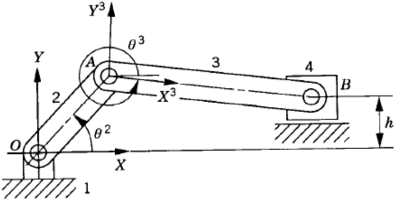
August 9, 2023

Introduction

- Recall Kinematics is study of motion
  - Without consideration of forces
- For a given kinematic input find position/ velocity/ acceleration
- The first step is to carry out position analysis
  - Non-linear functions of system co-ordinates
  - Requires use of methods such as Newton-Raphson iteration
- Following this one does velocity analysis and then acceleration analysis
  - These are obtained by time differentiation of kinematic relations
  - Involves solving linear equations
- We will start with the classical approach

Classical Approach

- Assuming crank angle θ^2 is given one can use loop closure to write
  - $l^2 \cos\theta^2 + l^3 \cos\theta^3 - x_B^4 = 0$
  - $l^2 \sin\theta^2 + l^3 \sin\theta^3 - h = 0$
- Two unknowns are θ^3 and x_B^4
- Easiest way is by elimination
  - $\theta^3 = \sin^{-1}\left(\frac{h - l^2 \sin\theta^2}{l^3}\right)$
  - $x_B^4 = l^2 \cos\theta^2 + \sqrt{(l^3)^2 - (h - l^2 \sin\theta^2)^2}$
- A more general way is shown next



Loop Closure Kinematics

- The two equations in the previous slide are non-linear algebraic equations
  - $l^2 \cos\theta^2 + l^3 \cos\theta^3 - x_B^4 = f_1(\theta^3, x_B^4) = 0$
  - $l^2 \sin\theta^2 + l^3 \sin\theta^3 - h = f_2(\theta^3, x_B^4) = 0$
- Given θ^2 these can be solved by Newton-Raphson iteration

$$\boxed{\begin{Bmatrix} \theta^3 \\ x_B^4 \end{Bmatrix}^{k+1} = \begin{Bmatrix} \theta^3 \\ x_B^4 \end{Bmatrix}^{k} - (\mathbf{J}^k)^{-1} \begin{Bmatrix} f_1(\theta^3,\, x_B^4) \\ f_2(\theta^3,\, x_B^4) \end{Bmatrix}^{k}}$$

- The Jacobian matrix $\mathbf{J}^k$ at the k^{th} iteration is calculated as shown next

Jacobian Matrix

$$\boxed{\mathbf{J}^k = \begin{bmatrix} \frac{\partial f_1}{\partial \theta^3} & \frac{\partial f_1}{\partial x_B^4} \\ \frac{\partial f_2}{\partial \theta^3} & \frac{\partial f_2}{\partial x_B^4} \end{bmatrix}^k = \begin{bmatrix} -l^3 \sin\theta^3 & -1 \\ l^3 \cos\theta^3 & 0 \end{bmatrix}^k}$$

- Note that the Jacobian will be singular when $\theta^3 = 270^\circ$
- This does not happen in most practical situations as l^3 is at least 3-4 times l^2
  - Can happen for $l^2 = l^3$ when $\theta^2 = 90^\circ$ when offset $h = 0$
- Such configurations are called singular configurations
- Then two possibilities exist Singular
  - Both links move together like a pendulum
  - Slider moves to right or left

Velocity Analysis

- Taking the time derivative of the velocity equations yield
  - $-l^2\dot{\theta}^2 \sin\theta^2 - l^3\dot{\theta}^3 \sin\theta^3 - \dot{x}_B^4 = 0$
  - $l^2\dot{\theta}^2 \cos\theta^2 + l^3\dot{\theta}^3 \cos\theta^3 = 0$
- This can be re-arranged assuming $\dot{\theta}^2$ as known input

$$\boxed{\begin{bmatrix} -l^3 \sin\theta^3 & -1 \\ l^3 \cos\theta^3 & 0 \end{bmatrix} \begin{Bmatrix} \dot{\theta}^3 \\ \dot{x}_B^4 \end{Bmatrix} = \dot{\theta}^2 \begin{Bmatrix} l^2 \sin\theta^2 \\ -l^2 \cos\theta^2 \end{Bmatrix}}$$

- This is a linear equation once positions have been found
- Note that the matrix is the Jacobian $\mathbf{J}$

Acceleration Analysis

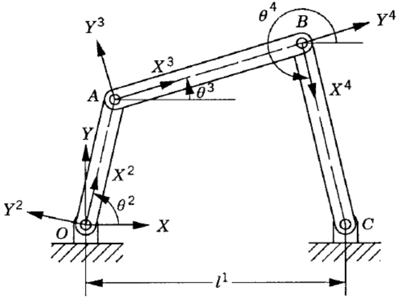
- Taking the time derivative of the loop closure equations yield
  - $-l^2\ddot{\theta}^2 \sin\theta^2 - l^3\ddot{\theta}^3 \sin\theta^3 - \ddot{x}_B^4 - l^2(\dot{\theta}^2)^2 \cos\theta^2 - l^3(\dot{\theta}^3)^2 \cos\theta^3 = 0$
  - $l^2\ddot{\theta}^2 \cos\theta^2 + l^3\ddot{\theta}^3 \cos\theta^3 - l^2(\dot{\theta}^2)^2 \sin\theta^2 - l^3(\dot{\theta}^3)^2 \sin\theta^3 = 0$
- Assuming $\dot{\theta}^2$ and $\ddot{\theta}^2$are known inputs we get

$$\begin{bmatrix} -l^3\sin\theta^3 & -1 \\ l^3\cos\theta^3 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{\theta}^3 \\ \ddot{x}_B^4 \end{Bmatrix} = (\dot{\theta}^2)^2 \begin{Bmatrix} l^2\cos\theta^2 \\ l^2\sin\theta^2 \end{Bmatrix} + \ddot{\theta}^2 \begin{Bmatrix} l^2\sin\theta^2 \\ -l^2\cos\theta^2 \end{Bmatrix} + (\dot{\theta}^3)^2 \begin{Bmatrix} l^3\cos\theta^3 \\ l^3\sin\theta^3 \end{Bmatrix}$$

- Linear equation once positions and velocities found
- Note matrix is Jacobian $\mathbf{J}$

Body Co-ordinate System

- To describe motion introduce a co-ordinate system for each body
- Co-ordinates of P^i on body i in X^iY^i system
  - $\bar{\mathbf{u}}_P^i = \begin{Bmatrix} \bar{x}_P^i \\ \bar{y}_P^i \end{Bmatrix}$
- Alternately $\boxed{\bar{\mathbf{u}}_P^i = \bar{x}_P^i \mathbf{i}^i + \bar{y}_P^i \mathbf{j}^i}$
  - $\mathbf{i}^i$ and $\mathbf{j}^i$ unit vectors along X^i & Y^i
- X^iY^i makes an angle θ^i with XY the fixed frame



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

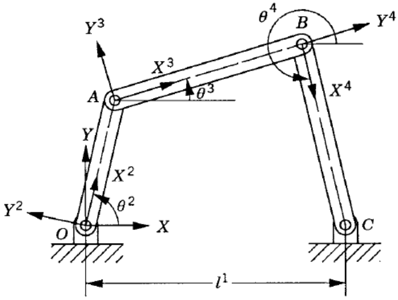
Transformation

- $\mathbf{i} = \cos\theta^i \mathbf{i}^i - \sin\theta^i \mathbf{j}^i$
- $\mathbf{j} = \sin\theta^i \mathbf{i}^i + \cos\theta^i \mathbf{j}^i$
- Reversing the relation leads to
- $\mathbf{i}^i = \cos\theta^i \mathbf{i} + \sin\theta^i \mathbf{j}$
- $\mathbf{j}^i = -\sin\theta^i \mathbf{i} + \cos\theta^i \mathbf{j}$
- Using above one can write

  $$\mathbf{u}_P^i = (\bar{x}_p^i \cos\theta^i - \bar{y}_P^i \sin\theta^i)\mathbf{i} + (\bar{x}_p^i \sin\theta^i + \bar{y}_P^i \cos\theta^i)\mathbf{j}$$

  $$\mathbf{u_p^i} = \mathbf{A}^i \bar{\mathbf{u}}_P^i$$

- $\mathbf{A}^i$ is the transformation matrix



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Transformation 2

- Transformation matrix

$$\boxed{\mathbf{A}^i = \begin{bmatrix} \cos\theta^i & -\sin\theta^i \\ \sin\theta^i & \cos\theta^i \end{bmatrix}}$$

- This has the following property
  - $\mathbf{A}^i\mathbf{A}^{i^T} = \mathbf{A}^{i^T}\mathbf{A}^i = \mathbf{I}$; $\mathbf{I}$ is Identity matrix
  - $\mathbf{A}^i$ is called a normal matrix
- Next we represent the global co-ordinates of P^i
  - $\boxed{\mathbf{r}_P^i = \mathbf{R}^i + \mathbf{u}_P^i}$
  - $\mathbf{R}^i$ is position vector of origin O^i of X^iY^i
- Using the transformation done earlier we have
  - $\boxed{\mathbf{r}_P^i = \mathbf{R}^i + \mathbf{A}^i\bar{\mathbf{u}}_P^i}$

Velocity

- Displacement of arbitrary point P^i on body i in terms of
  - Translation $\mathbf{R}^i$ of origin O^i of body co-ordinate system X^iY^i
  - Rotation θ^i of X^iY^i with respect to XY
- The next step is to find the velocity of this point P^i
  - $\boxed{\dot{\mathbf{r}}_P^i = \dot{\mathbf{R}}^i + \dot{\mathbf{A}}^i \bar{\mathbf{u}}_P^i}$
  - Note that generally $\dot{\bar{\mathbf{u}}}_P^i = 0$
- $\dot{\mathbf{A}}^i = \dot{\theta}^i \mathbf{A}_\theta^i$; $\mathbf{A}_\theta^i = \begin{bmatrix} -\sin\theta^i & -\cos\theta^i \\ \cos\theta^i & -\sin\theta^i \end{bmatrix}$
- This implies then $\boxed{\dot{\mathbf{r}}_P^i = \dot{\mathbf{R}}^i + \dot{\theta}^i \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i}$

Equivalence

- Recall from classical vector based approach
  - $\mathbf{v}_{P^i} = \mathbf{v}_{O^i} + \mathbf{v}_{P^i/O^i} = \mathbf{v}_{O^i} + \boldsymbol{\omega}^i \times \mathbf{u}_P^i$
- The skew-symmetric form for the cross product will become
  - $\boldsymbol{\omega}^i \times \mathbf{u}_P^i = \tilde{\boldsymbol{\omega}}^i \mathbf{A}^i \bar{\mathbf{u}}_P^i$; $\tilde{\boldsymbol{\omega}}^i = \dot{\theta}^i \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
- From this we can show
  - $\mathbf{A}_\theta^i = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{A}^i$

Acceleration

- The next step is to find the acceleration of this point P^i
  - $\boxed{\ddot{\mathbf{r}}^i_P = \ddot{\mathbf{R}}^i + \dot{\theta}^i \dot{\mathbf{A}}^i_\theta \bar{\mathbf{u}}^i_P + \ddot{\theta}^i \mathbf{A}^i_\theta \bar{\mathbf{u}}^i_P}$
  - Note that generally $\dot{\bar{\mathbf{u}}}^i_P = \ddot{\bar{\mathbf{u}}}^i_P = 0$
- $\dot{\mathbf{A}}^i_\theta = \dot{\theta}^i \mathbf{A}^i_{\theta\theta}$; $\mathbf{A}^i_{\theta\theta} = \begin{bmatrix} -\cos\theta^i & \sin\theta^i \\ -\sin\theta^i & -\cos\theta^i \end{bmatrix} = -\mathbf{A}^i$
- This implies then $\boxed{\ddot{\mathbf{r}}^i_P = \ddot{\mathbf{R}}^i + \ddot{\theta}^i \mathbf{A}^i_\theta \bar{\mathbf{u}}^i_P - (\dot{\theta}^i)^2 \mathbf{A}^i \bar{\mathbf{u}}^i_P}$

Equivalence

- The classical vector based approach gives
  - $\mathbf{a}_{P^i} = \mathbf{a}_{O^i} + \mathbf{a}_{P^i/O^i} = \mathbf{a}_{O^i} + \boldsymbol{\alpha}^i \times \mathbf{u}_P^i + \boldsymbol{\omega}^i \times \boldsymbol{\omega}^i \times \mathbf{u}_P^i$
- Based on the earlier equivalence for $\boldsymbol{\omega}^i \times \bar{\mathbf{u}}_P^i$ one can see that
  - $\boldsymbol{\alpha}^i \times \mathbf{u}_P^i = \ddot{\theta}^i \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i$
- The skew-symmetric form for the cross-products of the next term
  - $\boldsymbol{\omega}^i \times \boldsymbol{\omega}^i \times \mathbf{u}_P^i = \tilde{\boldsymbol{\omega}}^i \tilde{\boldsymbol{\omega}}^i \mathbf{A}^i \bar{\mathbf{u}}_P^i$;

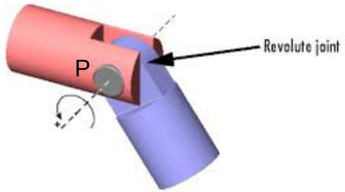
$$\tilde{\boldsymbol{\omega}}^i \tilde{\boldsymbol{\omega}}^i = (\dot{\theta}^i)^2 \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = -(\dot{\theta}^i)^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Moving Point on Rigid Body

- If point P^i moving relative to body i then $\bar{\mathbf{u}}_P^i$ not constant
  - Block in a slot is an example MovingPoint
- Then the velocity expression becomes
  - $\boxed{\dot{\mathbf{r}}_P^i = \dot{\mathbf{R}}^i + \dot{\theta}^i \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i + \mathbf{A}^i \dot{\bar{\mathbf{u}}}_P^i}$
- Equivalent classical expression
  - $\mathbf{v}_{P^i} = \mathbf{v}_{O^i} + \boldsymbol{\omega}^i \times \mathbf{u}_P^i + (\mathbf{v}_{P^i/O^i})_r$
- Acceleration expression
  - $\boxed{\ddot{\mathbf{r}}_P^i = \ddot{\mathbf{R}}^i + \ddot{\theta}^i \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i - (\dot{\theta}^i)^2 \mathbf{A}^i \bar{\mathbf{u}}_P^i + 2\dot{\theta}^i \mathbf{A}_\theta^i \dot{\bar{\mathbf{u}}}_P^i + \mathbf{A}^i \ddot{\bar{\mathbf{u}}}_P^i}$
- Equivalent classical expression
  - $\mathbf{a}_{P^i} = \mathbf{a}_{O^i} + \boldsymbol{\alpha}^i \times \mathbf{u}_P^i + \boldsymbol{\omega}^i \times \boldsymbol{\omega}^i \times \mathbf{u}_P^i + 2\boldsymbol{\omega}^i \times (\mathbf{v}_{P^i/O^i})_r + (\mathbf{a}_{P^i/O^i})_r$

Revolute Joint Constraint

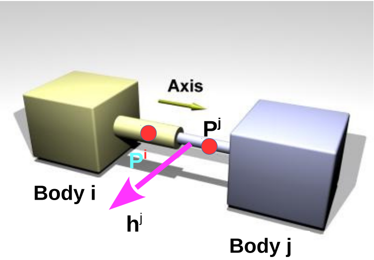
- Suppose now that body i and j are connected by a revolute joint
- Point P is common to both bodies i (pink) and j (blue)
- Revolute joint constraint is point P^i and P^j remain in contact throughout motion
  - $\mathbf{r}_P^i = \mathbf{r}_P^j$
- Two bodies are free to rotate with respect to each other



Courtesy: www.mathworks.com

Prismatic Joint Constraint

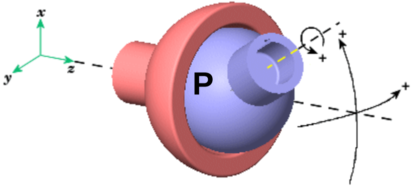
- No relative rotation between the bodies
  - $\theta^i - \theta^j = \theta_0^{ij}$
- No relative translation along axis $\perp$ to prismatic joint axis
- P^i and P^j on bodies i and j along joint axis
- $\mathbf{h}^j$ or $\mathbf{h}^i$ defined $\perp$ to joint axis
  - $(\mathbf{r}_P^i - \mathbf{r}_P^j)^{\mathrm{T}}\mathbf{h}^j = 0$ or
  - $(\mathbf{r}_P^i - \mathbf{r}_P^j)^{\mathrm{T}}\mathbf{h}^i = 0$



Courtesy: https://www.ode-wiki.org/wiki/index.php?title=Manual:_Joint_Types_and_Functions#Slider

Spherical Joint Constraint

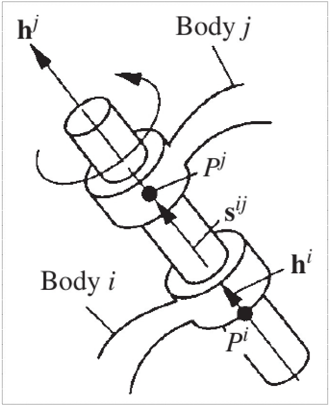
- Suppose now that body i and j are connected by a spherical joint
- Point P is common to both bodies i (pink) and j (blue)
- Constraint is point P^i and P^j remain in contact throughout motion
  - $\mathbf{r}^i_P = \mathbf{r}^j_P$
- Two bodies are free to rotate with respect to each other about 3 mutually perpendicular axes



Courtesy: https://www.mathworks.com/

Cylindrical Joint Constraint

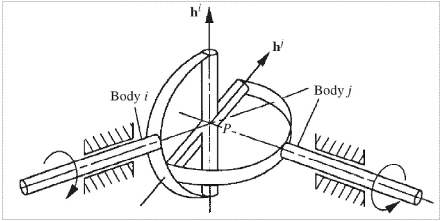
- $\mathbf{h}^i$ and $\mathbf{h}^j$ are vectors along joint axis on body i and j
- $\mathbf{s}^{ij} = \mathbf{r}_P^i - \mathbf{r}_P^j$ is the varying distance between P^i and P^j
- $\mathbf{h}^i$ must remain collinear to $\mathbf{h}^j$ and $\mathbf{s}^{ij}$
  - $\mathbf{h}^i \times \mathbf{h}^j = \mathbf{0}$
  - $\mathbf{h}^i \times \mathbf{s}^{ij} = \mathbf{0}$
- In each cross-product only 2 equations independent leading to 4 constraint equations



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Universal Joint Constraint

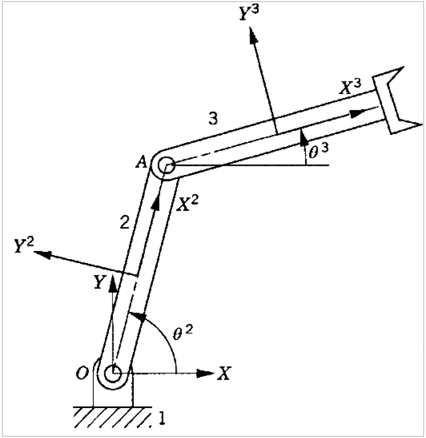
- Allows relative rotation about 2 $\perp$ axes
- In each cross-product only 2 equations independent leading to 4 constraint equations
- Point P at the centre of the cross-piece common to both body i and j
- $\mathbf{h}^i$ and $\mathbf{h}^j$ along the bars of the cross
  - $\mathbf{r}_P^i = \mathbf{r}_P^j$; $\mathbf{h}^{i\mathrm{T}}\mathbf{h}^j = 0$



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Example: Two Link Manipulator

- Body i has 3 co-ordinates to describe motion
  - R_x^i, R_y^i and θ^i
- Assume for body 1 body co-ordinate system coincides with global XY
- Body co-ordinate system for 2 is at distance of $\frac{l^2}{2}$ from O
- Body co-ordinate system for 3 at $\frac{l^3}{2}$ from A
- 2 revolute joints at O and A and body 1 fixed



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Constraint Equations

- For body 1 we have
  - $\boxed{R_x^1 = 0;\ R_y^1 = 0;\ \theta^1 = 0}$
- For revolute joint at O as part of body 1 ▸ 2linkManip
  - $\mathbf{r}_O^1 = \mathbf{R}^1 + \mathbf{A}^1\bar{\mathbf{u}}_O^1$
- For revolute joint at O as part of body 2 ▸ 2linkManip
  - $\mathbf{r}_O^2 = \mathbf{R}^2 + \mathbf{A}^2\bar{\mathbf{u}}_O^2$
- We have for revolute joint $\boxed{\mathbf{r}_O^1 - \mathbf{r}_O^2 = \mathbf{0}}$
- In component form
  - $\boxed{R_x^1 - R_x^2 + \frac{l^2}{2}\cos\theta^2 = 0}$
  - $\boxed{R_y^1 - R_y^2 + \frac{l^2}{2}\sin\theta^2 = 0}$

Constraint Equations 2

- For revolute joint at A as part of body 2 ‣ 2linkManip
  - $\mathbf{r}_A^2 = \mathbf{R}^2 + \mathbf{A}^2\bar{\mathbf{u}}_A^2$
- For revolute joint at A as part of body 3 ‣ 2linkManip
  - $\mathbf{r}_A^3 = \mathbf{R}^3 + \mathbf{A}^3\bar{\mathbf{u}}_A^3$
- We have for revolute joint $\boxed{\mathbf{r}_A^2 - \mathbf{r}_A^3 = \mathbf{0}}$
- In component form
  - $R_x^2 + \frac{l^2}{2}\cos\theta^2 - R_x^3 + \frac{l^3}{2}\cos\theta^3 = 0$
  - $R_y^2 + \frac{l^2}{2}\sin\theta^2 - R_y^3 + \frac{l^3}{2}\sin\theta^3 = 0$
- So we have developed 7 constraint equations relating 9 co-ordinates

Driving Constraints

- Joint constraints only depend on system co-ordinates
- Driving constraints define specific motion trajectories
  - Depends on both system co-ordinates and time
- For example recall slider-crank mechanism discussed
  - We assume $\dot{\theta}^2 = \omega^2$ is constant
  - Integrating this ODE yields $\theta^2 = \omega^2 t + \theta_0^2$
  - The above is called simple driving constraint
- Specify trajectory of a point P^i in the XY plane
  - $R_x^i + \bar{x}_P^i \cos\theta^i - \bar{y}_P^i \sin\theta^i = f_1(t)$
  - $R_y^i + \bar{x}_P^i \sin\theta^i + \bar{y}_P^i \cos\theta^i = f_2(t)$
  - A complex driving constraint as more than one co-ordinate involved

Driving Constraints for 2 Link Manipulator

- Number of driving constraints equals the number of degrees-of-freedom of multi-body system
- For 2 link manipulator then we specify two
  - $\dot{\theta}^2 - \omega^2 = 0$; $\dot{\theta}^3 - \omega^3 = 0$
- Integrating these yield the following
  - $\theta^2 - \omega^2 t - \theta_0^2 = 0$; $\theta^3 - \omega^3 t - \theta_0^3 = 0$
- So now we have 9 equations for 9 system co-ordinates
- The general form of the constraints is $\mathbf{C}(\mathbf{q}, t) = \mathbf{0}$
  - For 2 link manipulator

$$\mathbf{q} = \begin{bmatrix} R_x^1 & R_y^1 & \theta^1 & R_x^2 & R_y^2 & \theta^2 & R_x^3 & R_y^3 & \theta^3 \end{bmatrix}^{\mathrm{T}}$$

Full Set of Constraints

$$
\begin{aligned}
R_x^1 &= 0 \\
R_y^1 &= 0 \\
\theta^1 &= 0 \\
R_x^1 - R_x^2 + \frac{l^2}{2}\cos\theta^2 &= 0 \\
R_y^1 - R_y^2 + \frac{l^2}{2}\sin\theta^2 &= 0 \\
R_x^2 + \frac{l^2}{2}\cos\theta^2 - R_x^3 + \frac{l^3}{2}\cos\theta^3 &= 0 \\
R_y^2 + \frac{l^2}{2}\sin\theta^2 - R_y^3 + \frac{l^3}{2}\sin\theta^3 &= 0 \\
\theta^2 - \omega^2 t - \theta_0^2 &= 0 \\
\theta^3 - \omega^3 t - \theta_0^3 &= 0
\end{aligned}
$$

Position Analysis

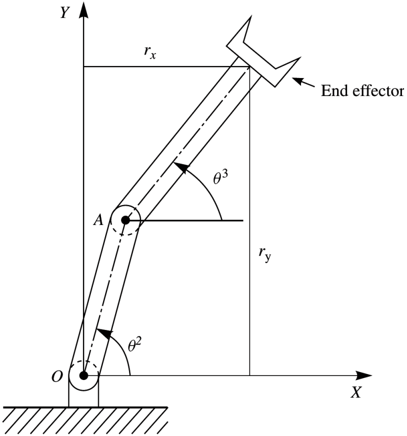
- System of non-linear algebraic equations solved numerically
  - Newton-Raphson iteration method at each t
- Taylor series expansion assuming $\mathbf{q}_i + \Delta\mathbf{q}_i$ is exact solution
  - $\mathbf{C}(\mathbf{q}_i + \Delta\mathbf{q}_i, t) \approx \mathbf{C}(\mathbf{q}_i, t) + \mathbf{C}_{\mathbf{q}_i}\Delta\mathbf{q}_i = 0$
  - $\boxed{\mathbf{C}_{\mathbf{q}_i}\Delta\mathbf{q}_i = -\mathbf{C}(\mathbf{q}_i, t)}$
- $\mathbf{C}_{\mathbf{q}_i}$ is the Jacobian matrix ▸ MatrixElem
- Assuming $\mathbf{C}_{\mathbf{q}_i}$ is non-singular one can find $\Delta\mathbf{q}_i$
  - Updated vector of co-ordinates $\mathbf{q}_{i+1} = \mathbf{q}_i + \Delta\mathbf{q}_i$
- Iteration continues until $|\Delta\mathbf{q}_k| < \epsilon$
- ϵ is a pre-specified tolerance and k is iteration number

Convergence Issues

- Sometimes iterations may not converge
  - Initial estimate of desired solution not close enough to exact solution
  - Error in definition of constraints
  - System close to a singular configuration
- Always a wise idea to specify upper limit for number of iterations

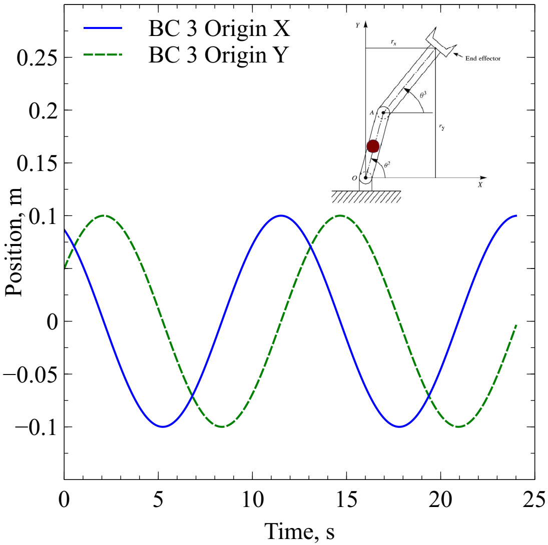
Solving 2 Link Manipulator

- $l^2 = 0.2$ m; $l^3 = 0.35$ m
- $\omega^2 = 0.5$ rad/s; $\omega^3 = 0.75$ rad/s
- $\theta^2_0 = \frac{\pi}{6}$; $\theta^3_0 = \frac{\pi}{12}$
- Position analysis done from $t = 0$ to $t = 24$ s
  - Using Newton-Raphson iteration

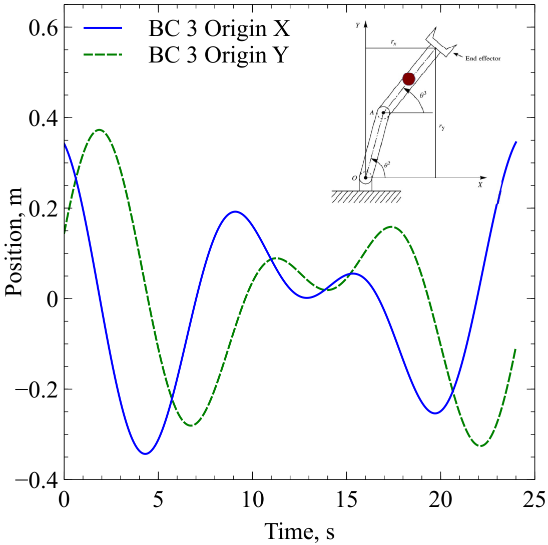


Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

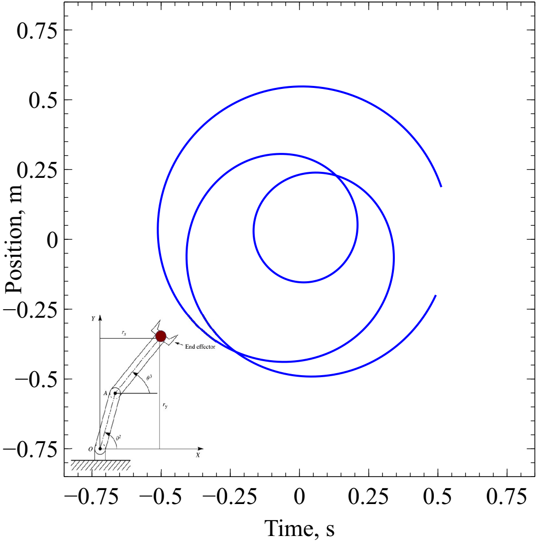
Link 2 Body Co-ordinate Origin



Link 3 Body Co-ordinate Origin



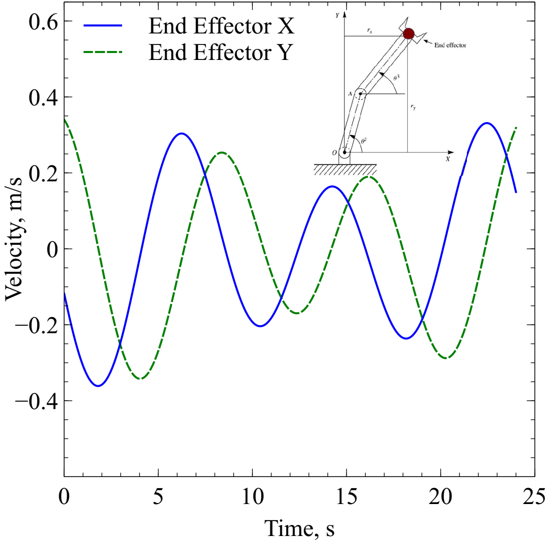
End Effector Trajectory



Velocity Analysis

- If we differentiate the constraint equations
  - $\mathbf{C_q}\dot{\mathbf{q}} + \mathbf{C}_t = \mathbf{0}$
  - $\mathbf{C}_t = \begin{bmatrix} \frac{\partial C_1}{\partial t} & \frac{\partial C_2}{\partial t} & \frac{\partial C_3}{\partial t} & \dots & \frac{\partial C_{nc}}{\partial t} \end{bmatrix}^{\mathrm{T}}$
- $\mathbf{C_q}$ is Jacobian matrix available from position analysis
- Finding $\dot{\mathbf{q}}$ is a linear equation solution
  - $\mathbf{C_q}\dot{\mathbf{q}} = -\mathbf{C}_t$
- For the 2-link manipulator example
  - $\mathbf{C}_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\omega^2 & -\omega^3 \end{bmatrix}^{\mathrm{T}}$

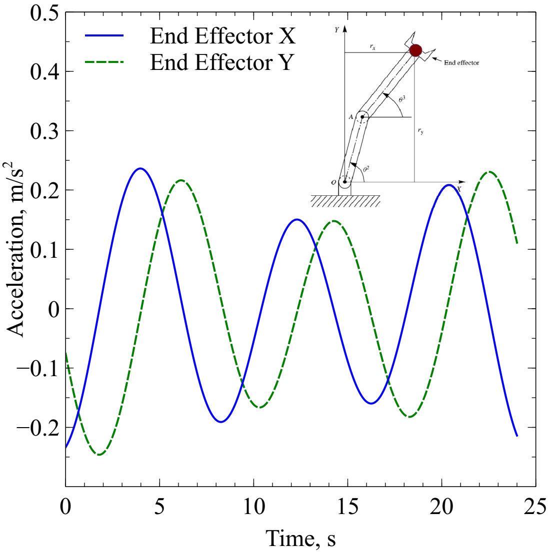
End Effector Velocities



Acceleration Analysis

- Differentiate the velocity equation once more
  - $\boxed{\mathbf{C_q\ddot{q}} + (\mathbf{C_q\dot{q}})_\mathbf{q}\mathbf{\dot{q}} + 2\mathbf{C}_{\mathbf{q}t}\mathbf{\dot{q}} + \mathbf{C}_{tt} = \mathbf{0}}$
- Can be re-written as
  - $\mathbf{C_q\ddot{q}} = \mathbf{Q}_d$; $\mathbf{Q}_d = -(\mathbf{C_q\dot{q}})_\mathbf{q}\mathbf{\dot{q}} - 2\mathbf{C}_{\mathbf{q}t}\mathbf{\dot{q}} - \mathbf{C}_{tt}$
- The RHS term depends on position and velocity which are already known
  - For the 2-link example $\mathbf{C}_{tt} = \mathbf{0}$ as angular velocity driving inputs constant
  - Hence $\mathbf{C}_{\mathbf{q}t} = \mathbf{0}$ as $\mathbf{C}_t$ does not depend on $\mathbf{q}$
- So how does $\mathbf{C_q\dot{q}}$ look like? Cqqdot

End Effector Accelerations



Constraint Library: Revolute

- The revolute joint constraint in general form
  - $\mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}_P^i - \mathbf{R}^j - \mathbf{A}^j \bar{\mathbf{u}}_P^j = \mathbf{0}$
- To generate the Jacobian elements for these equations
  - $\mathbf{C}_\mathbf{q} = \left[\frac{\partial \mathbf{C}}{\partial \mathbf{R}^i} \quad \frac{\partial \mathbf{C}}{\partial \theta^i} \quad -\frac{\partial \mathbf{C}}{\partial \mathbf{R}^j} \quad -\frac{\partial \mathbf{C}}{\partial \theta^j} \right]$
  - $= \left[\mathbf{I} \quad \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i \quad -\mathbf{I} \quad -\mathbf{A}_\theta^j \bar{\mathbf{u}}_P^j \right]$
  - Note that this is a matrix of size 2×6
- Also $\mathbf{C}_t$ as well as $\mathbf{C}_{\mathbf{q}t}$ and $\mathbf{C}_{tt}$ are zero
- $(\mathbf{C}_\mathbf{q} \dot{\mathbf{q}})_\mathbf{q} \dot{\mathbf{q}} = (\dot{\theta}^j)^2 \mathbf{A}^j \bar{\mathbf{u}}_P^j - (\dot{\theta}^i)^2 \mathbf{A}^i \bar{\mathbf{u}}_P^i$

Constraint Library: Prismatic

- $\mathbf{C}(\mathbf{q},t) = \begin{bmatrix} \theta^i - \theta^j - \theta_0^{ij} & {\mathbf{h}^i}^{\mathrm{T}} \mathbf{r}_P^{ij} \end{bmatrix}^{\mathrm{T}}$
- Note $\mathbf{r}_P^{ij} = \mathbf{r}_P^i - \mathbf{r}_P^j = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}_P^i - \mathbf{R}^j - \mathbf{A}^j \bar{\mathbf{u}}_P^j$
- Also $\mathbf{h}^i = \mathbf{A}^i (\bar{\mathbf{u}}_P^i - \bar{\mathbf{u}}_Q^i) = \mathbf{A}^i \bar{\mathbf{h}}^i$
- From these the Jacobian becomes

$$\boxed{\mathbf{C}_{\mathbf{q}} = \begin{bmatrix} \mathbf{0}^{\mathrm{T}} & 1 & \mathbf{0}^{\mathrm{T}} & -1 \\ {\mathbf{h}^i}^{\mathbf{T}} & {\mathbf{r}_P^{ij}}^{\mathrm{T}} \mathbf{A}_\theta^i \bar{\mathbf{h}}^i + {\mathbf{h}^i}^{\mathrm{T}} \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i & -{\mathbf{h}^i}^{\mathbf{T}} & -{\mathbf{h}^i}^{\mathrm{T}} \mathbf{A}_\theta^j \bar{\mathbf{u}}_P^j \end{bmatrix}}$$

- The terms $\mathbf{C}_t$, $\mathbf{C}_{tt}$ and $\mathbf{C}_{\mathbf{q}t}$ are all $\mathbf{0}$

Prismatic 2

- The product $\mathbf{C_q}\dot{\mathbf{q}}$ is

$$\mathbf{C_q}\dot{\mathbf{q}} = \left\{ \begin{matrix} \dot{\theta}^i - \dot{\theta}^j \\ \mathbf{h}^{i^{\mathrm{T}}}(\dot{\mathbf{R}}^i - \dot{\mathbf{R}}^j) + \dot{\theta}^i(\mathbf{r}_P^{ij^{\mathrm{T}}}\mathbf{A}_\theta^i\bar{\mathbf{h}}^i + \mathbf{h}^{i^{\mathrm{T}}}\mathbf{A}_\theta^i\bar{\mathbf{u}}_P^i) - \\ \dot{\theta}^j\mathbf{h}^{i^{\mathrm{T}}}\mathbf{A}_\theta^j\bar{\mathbf{u}}_P^j \end{matrix} \right\}$$

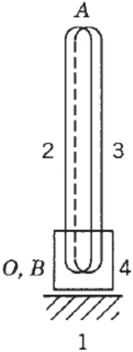
- Then $(\mathbf{C_q}\dot{\mathbf{q}})_\mathbf{q}$ is

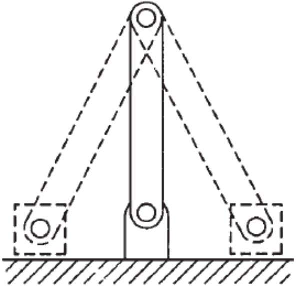
$$(\mathbf{C_q}\dot{\mathbf{q}})_\mathbf{q} = \begin{bmatrix} \mathbf{0}^T & 0 & \mathbf{0}^T & 0 \\ \mathbf{D}_1^T & D_2 & \mathbf{D}_3^T & D_4 \end{bmatrix}$$

Prismatic 3

- $\mathbf{D}_1 = \dot{\theta}^i \mathbf{A}_\theta^i \bar{\mathbf{h}}^i ; \mathbf{D}_3 = -\dot{\theta}^i \mathbf{A}_\theta^i \bar{\mathbf{h}}^i$
- $D_2 = (\mathbf{A}_\theta^i \bar{\mathbf{h}}^i)^T (\dot{\mathbf{R}}^i - \dot{\mathbf{R}}^j) + \dot{\theta}^i (\bar{\mathbf{u}}_P^i \bar{\mathbf{h}}^i - {\mathbf{r}_P^{ij}}^T \mathbf{h}^i) - \dot{\theta}^j (\mathbf{A}_\theta^i \bar{\mathbf{h}}^i)^T \mathbf{A}_\theta^j \bar{\mathbf{u}}_P^j$
- $D_4 = -\dot{\theta}^i (\mathbf{A}_\theta^j \bar{\mathbf{u}}_P^j)^T (\mathbf{A}_\theta^i \bar{\mathbf{h}}^i) + \dot{\theta}^j {\mathbf{h}^i}^T \mathbf{u}_P^j$
- $\mathbf{Q}_d = \begin{bmatrix} 0 & Q_{d2} \end{bmatrix}^T$
  - $Q_{d2} = -(\mathbf{D}_1^T \dot{\mathbf{R}}^i + \dot{\theta}^i D_2 + \mathbf{D}_3^T \dot{\mathbf{R}}^j + \dot{\theta}^j D_4)$

Singular Configurations

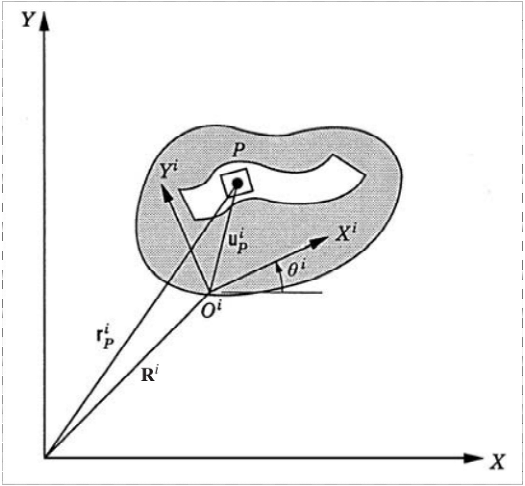




Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Return

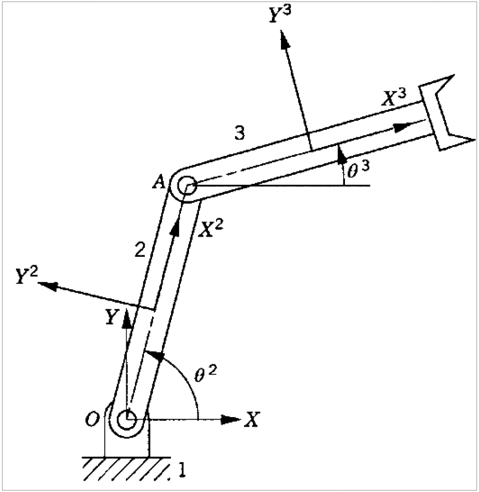
Moving Point on Body



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

◂ Back

Two Link Manipulator



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Return Return2

Jacobian Matrix

$$\mathbf{C}_{\mathbf{q}_i} = \begin{bmatrix} \dfrac{\partial C_1}{\partial q_1} & \dfrac{\partial C_1}{\partial q_2} & \dfrac{\partial C_1}{\partial q_3} & \cdots & \dfrac{\partial C_1}{\partial q_n} \\[2ex] \dfrac{\partial C_2}{\partial q_1} & \dfrac{\partial C_2}{\partial q_2} & \dfrac{\partial C_2}{\partial q_3} & \cdots & \dfrac{\partial C_2}{\partial q_n} \\[2ex] \dfrac{\partial C_3}{\partial q_1} & \dfrac{\partial C_3}{\partial q_2} & \dfrac{\partial C_3}{\partial q_3} & \cdots & \dfrac{\partial C_3}{\partial q_n} \\[2ex] \vdots & \vdots & \vdots & \ddots & \vdots \\[2ex] \dfrac{\partial C_{n_c}}{\partial q_1} & \dfrac{\partial C_{n_c}}{\partial q_2} & \dfrac{\partial C_{n_c}}{\partial q_3} & \cdots & \dfrac{\partial C_{n_c}}{\partial q_n} \end{bmatrix}$$

- For kinematically driven systems $n_c = n$

Back

Jacobian Matrix

$$\mathbf{C_q}\dot{\mathbf{q}} = \left\{ \begin{array}{c} \dot{R}_x^1 \\ \dot{R}_y^1 \\ \dot{\theta}_1 \\ \dot{R}_x^1 - \dot{R}_x^2 - \frac{l^2}{2}\dot{\theta}^2 \sin\theta^2 \\ \dot{R}_y^1 - \dot{R}_y^2 + \frac{l^2}{2}\dot{\theta}^2 \cos\theta^2 \\ \dot{R}_x^2 - \dot{R}_x^3 - \frac{l^2}{2}\dot{\theta}^2 \sin\theta^2 - \frac{l^3}{2}\dot{\theta}^3 \sin\theta^3 \\ \dot{R}_y^2 - \dot{R}_y^3 + \frac{l^2}{2}\dot{\theta}^2 \cos\theta^2 + \frac{l^3}{2}\dot{\theta}^3 \cos\theta^3 \\ \dot{\theta}^2 \\ \dot{\theta}^3 \end{array} \right\}$$

- For kinematically driven systems $n_c = n$

Return