ME451
Kinematics and Dynamics of Machine Systems

Composite Joints – 3.3

Gears and Cam Followers – 3.4

October 5, 2010

Before we get started…

- Last Time: We looked at several relative constraints
  - x, y, ϕ, and distance relative constraints
  - Recall the five step procedure:
    - Identify and analyze the physical joint
    - Derive the constraint equations associated with the joint
    - Compute constraint Jacobian $\mathbf{\Phi_q}$
    - Get $\boldsymbol{\nu}$ (RHS of velocity equation)
    - Get $\boldsymbol{\gamma}$ (RHS of acceleration equation, this is challenging in some cases)

- Today: Move on and cover a couple of other constraints
    - Revolute Joint
    - Translational Joint
    - Cam-follower

- HW:
  - ADAMS component, emailed by the TA, due on Tu, Oct 12
  - MATLAB assignment, available online, due in *two* weeks, on Oct. 19
  - Problems 3.3.2, 3.3.4, 3.3.5, due on Tu, Oct 12

- Next week, entire week (I'll be travelling):
  - Using ADAMS, part two
  - Focus is on ADAMS co-simulation and interfacing to MATLAB for mechatronics

MATLAB Assignment

- Posted online, at the class website

```
function [phiVal,Phi_q,nuVal,gammaVal]=revoluteJ(qi,qiDot,sBarPi,qj,qjDot,sBarPj,flagC)
% Computes Kinematics Analysis required quantities associated with the
% presenence of a revolute joint. Based on the value of the flagC, one
% could compute only phiVal; i.e., the violation of the algebraic
% constraint, or Phi_q (the sensitivity matrix; i.e., the Jacobian), or
% nuVal, the right hand side of the Velocity Equation, or gammaVal, the
% right hand side of the Acceleration Equation

if flagC == 1
    % compute the expression of phiVal only
    r_i = qi(1:2,1);
    phi_i = qi(3,1);
    r_j = qj(1:2,1);
    phi_j = qj(3,1);
    % use r_i,..., phi_j to compute phiVal below...
    blah blah...
elseif flagC == 2
    % compute the expression of phiVal only
    % code here to compute a matrix that is 2 by 6
    % The first 2X3 is sensitivity wrt qi, the last 2X3 is wrt qj
    blah blah...
elseif flagC == 3
    % compute the expression of nu; this is easy
    nuVal(1) = 0;
    nuVal(2) = 0;
elseif flagC == 4
    % compute the expression of gamma; this is messy
    % needs to use also time derivatives qiDot and qjDot
    blah blah...
end
```

Revolute Joint

- Step 1: Physically imposes the condition that point P on body *i* and a point P on body *j* are coincident at all times
- Step 2: Constraint Equations $\Phi(\mathbf{q},t) = ?$
- Step 3: Constraint Jacobian $\Phi_{\mathbf{q}} = ?$
- Step 4: $\nu = ?$
- Step 5: $\gamma = ?$

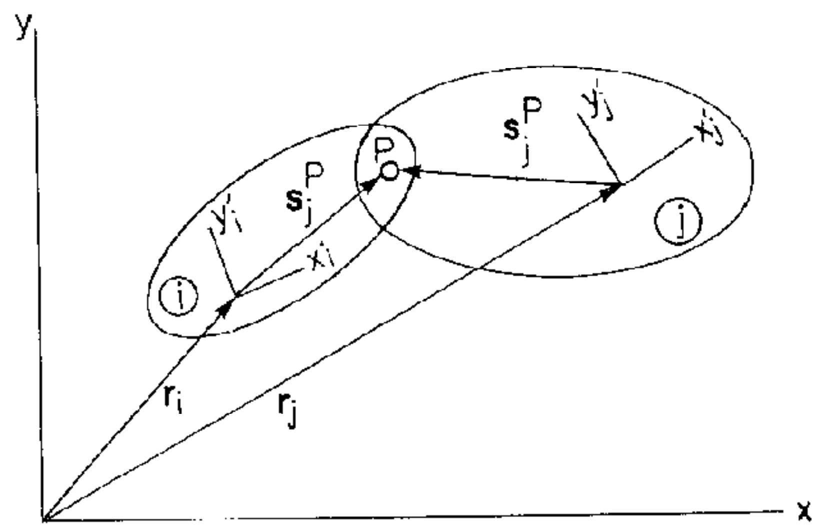


Figure 3.3.4 Revolute joint.

A Couple of Useful Formulas
[Short Detour]

- Recall that

$$\mathbf{R} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

- Therefore, we have that

$$\mathbf{B} \equiv \mathbf{R}\mathbf{A} = \mathbf{A}\mathbf{R}$$

$$\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}_{2\times 2}$$

$$\mathbf{R}\mathbf{R} = -\mathbf{I}_{2\times 2}$$

$$\mathbf{R}\mathbf{B} = \mathbf{B}\mathbf{R} = -\mathbf{A}$$

Translational Joint

- Step 1: Physically, it allows relative translation between two bodies along a *common axis*. No relative rotation is allowed.
- Step 2: Constraint Equations $\Phi(\mathbf{q},t)$ = ?
- Step 3: Constraint Jacobian $\Phi_{\mathbf{q}}$ = ?
- Step 4: ν = ?
- Step 5: γ = ?

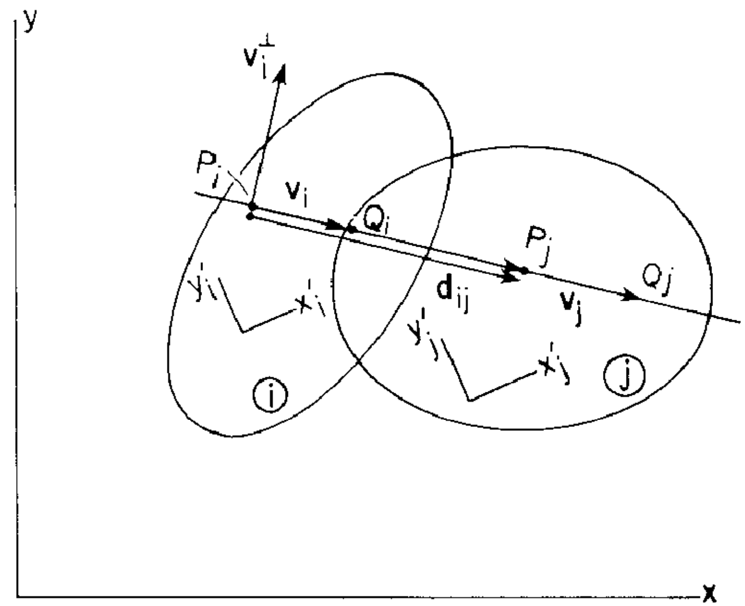


Figure 3.3.5 Translational joint.

NOTE: text uses notation $\mathbf{A}_{ij} = \mathbf{A}_i^T \mathbf{A}_j$

The Attributes of a Constraint

- Attributes of a Constraint: That information that you are supposed to know by inspecting the mechanism

- It represents the parameters associated with the specific constraint that you are considering

- When you are dealing with a constraint, make sure you understand
  - What the input is
  - What the defining attributes of the constraint are
  - What constitutes the output (the algebraic equation[s], Jacobian, γ, ν, etc.)

The Attributes of a Constraint
[Cntd.]

- Examples of constraint attributes:

  - For a revolute joint: You know where the joint is located, so therefore you know the coordinates of $\mathbf{P}_i$ and $\mathbf{P}_j$, that is, $\bar{\mathbf{s}}_i^P$ and $\bar{\mathbf{s}}_j^P$, respectively.

  - For a translational join: You know what the point $\mathbf{P}$ on body i is, and you know the direction of relative translation $\bar{\mathbf{v}}_i$ is. You have the same information for body j: position of point $\mathbf{P}_j$ and direction of relative translation $\bar{\mathbf{v}}_j$.

  - For a distance constraint: You know the distance C_4

  - Etc.

Example 3.3.2 – Different ways of modeling the same mechanism for Kinematic Analysis

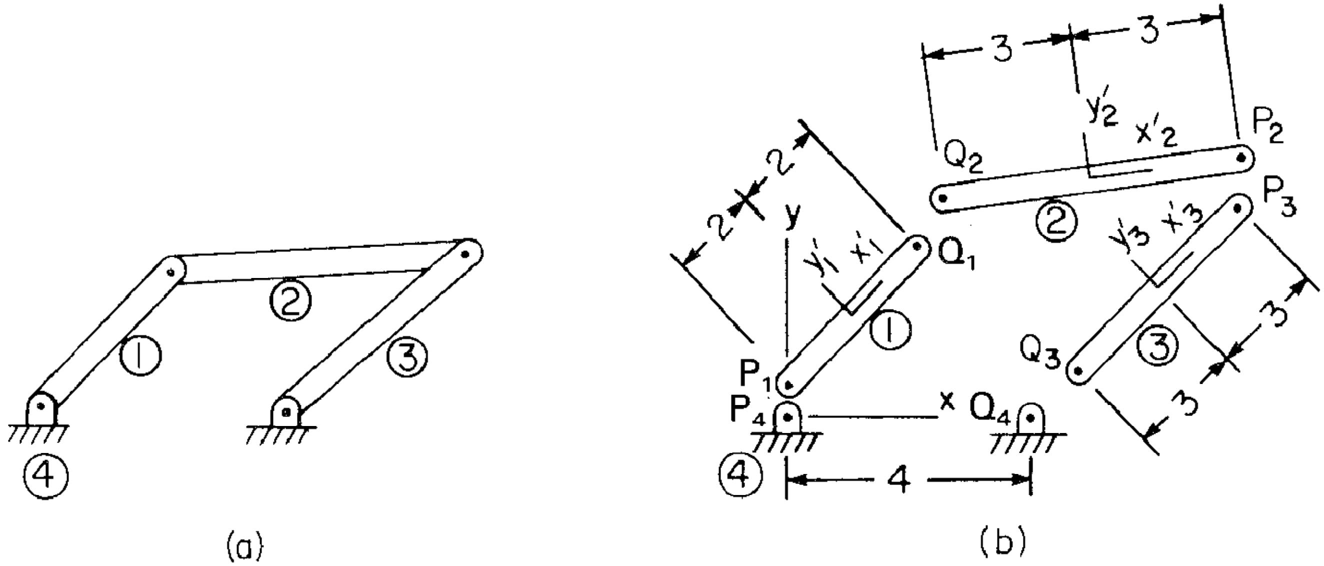


Figure 3.3.2 Four-bar mechanism. (a) Assembled. (b) Dissected.

- Approach 1: bodies 1, 2, and 3
- Approach 2: bodies 1 and 3
- Approach 3: bodies 1 and 2
- Approach 4: body 2

Errata:

- Page 68 (unbalanced parentheses, and text)

Using Eqs. 2.4.12 and 2.6.8,

$$\gamma^{t(i,j)} = -\begin{bmatrix} \mathbf{v}_i'^T[\mathbf{B}_{ij}\mathbf{s}_j'^P(\dot\phi_j - \dot\phi_i)^2 - \mathbf{B}_i^T(\mathbf{r}_j - \mathbf{r}_i)\dot\phi_i^2 - 2\mathbf{A}_i^T(\dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i)\dot\phi_i \\ 0 \end{bmatrix}$$

where the second term on the right is zero, because of Eq. 3.3.13.

- Page 73 (transpose and sign)

$$\begin{aligned}\Phi^{g(i,j)} &= (\mathbf{r}_j^P - \mathbf{r}_i^P)\mathbf{u}^\perp \\ &= (x_j^P - x_i^P)\sin\theta - (y_j^P - y_i^P)\cos\theta = 0 \end{aligned} \qquad \textbf{(3.4.3)}$$

where θ is given by Eq. 3.4.2 and $\mathbf{u}^\perp \equiv [-\sin\theta, \cos\theta]^T$; that is, $\mathbf{u} = [\cos\theta, \sin\theta]^T$ is a unit vector along the line from P_i to P_j in Fig. 3.4.2.

- Page 67 (sign)

For the translational be specified on a line tha between bodies i and j. No P_j and Q_j are located on bc vector $\mathbf{v}_j$ in body j con $\mathbf{v}_i' = [x_i^P - x_i^Q, y_i^P - y_i^Q]^T$ an on body j. The vector $\mathbf{d}_i$ Vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ must rem collinear, it is necessary perpendicular to $\mathbf{v}_i$. Using

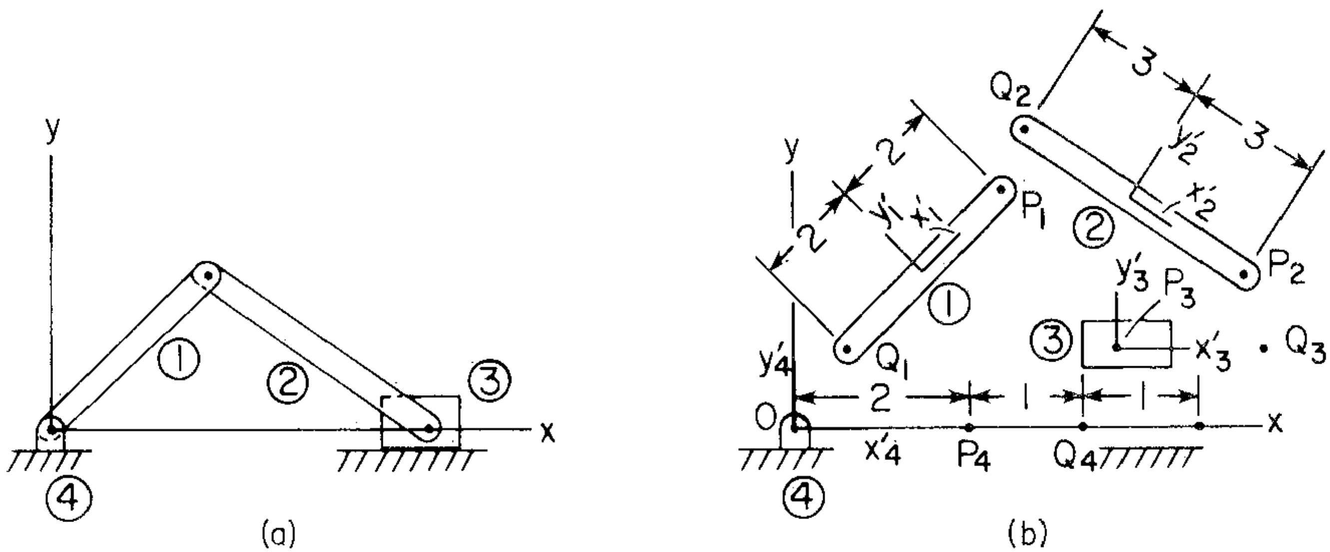
$$\Phi^{t(i,j)} = \begin{bmatrix} (\mathbf{v}_i^\perp)^T\mathbf{d}_{ij} \\ (\mathbf{v}_i^\perp)^T\mathbf{v}_i \end{bmatrix}$$

- Page 73 (perpendicular sign, both equations)

$$\begin{aligned}\Phi_{\mathbf{q}_i}^{g(i,j)} &= \left[-\mathbf{u}^T,\ -\mathbf{s}_i'^{P^T}\mathbf{B}_i^T\mathbf{u} + (\mathbf{r}_j^P - \mathbf{r}_i^P)^T\mathbf{u}^\perp\left(\frac{R_i}{R_i + R_j}\right)\right] \\ \Phi_{\mathbf{q}_j}^{g(i,j)} &= \left[\mathbf{u}^T,\ \mathbf{s}_j'^{P^T}\mathbf{B}_j^T\mathbf{u} + (\mathbf{r}_j^P - \mathbf{r}_i^P)^T\mathbf{u}^\perp\left(\frac{R_j}{R_i + R_j}\right)\right]\end{aligned} \qquad \textbf{(3.4.4)}$$

Example 3.3.4

- Consider the slider-crank below. Come up with the set of kinematic constraint equations to kinematically model this mechanism

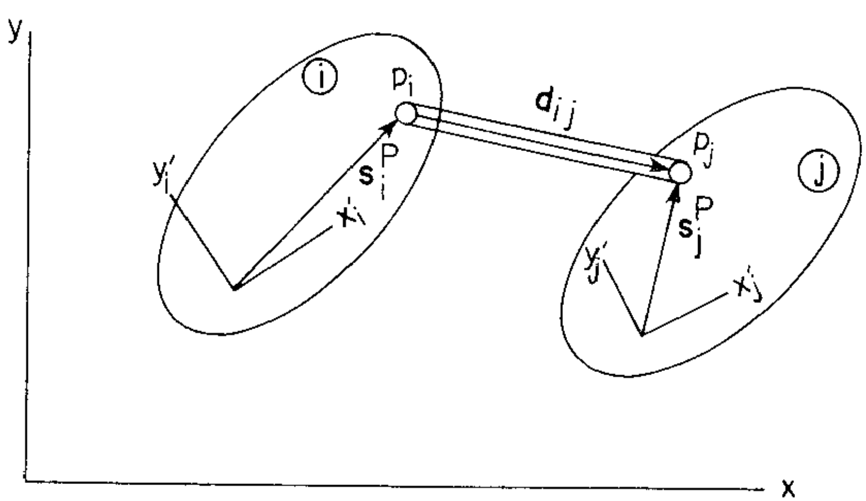


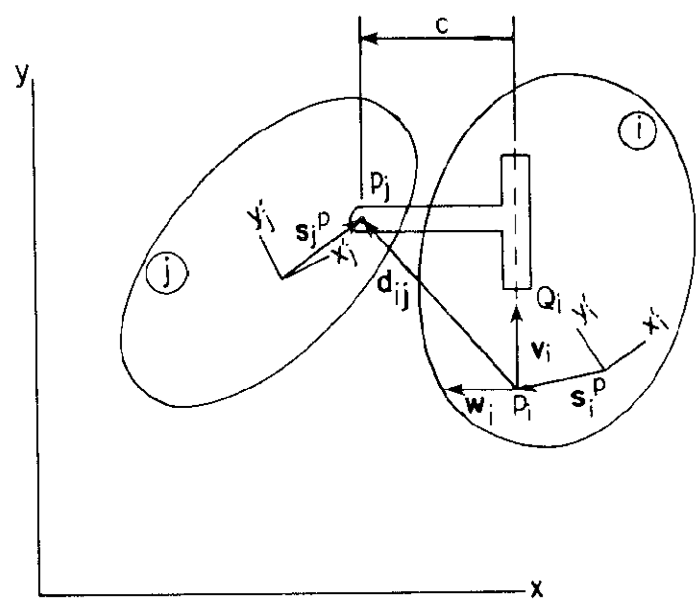
Composite Joints (CJ)

- Just a means to eliminate one intermediate body whose kinematics you are not interested in

- Revolute-Revolute CJ
  - Also called a coupler
  - Practically eliminates need of connecting rod
  - Given to you (joint attributes):
    - Location of points P_i and P_j
    - Distance d_{ij} of the massless rod

- Revolute-Translational CJ
  - Given to you (joint attributes):
    - Distance c
    - Point P_j (location of revolute joint)
    - Axis of translation $\mathbf{v}_i$'





Composite Joints

- One follows exactly the same steps as for any joint:
  - Step 1: Physically, what type of motion does the joint allow?
  - Step 2: Constraint Equations $\Phi(\mathbf{q},t) = ?$
  - Step 3: Constraint Jacobian $\Phi_{\mathbf{q}} = ?$
  - Step 4: $\nu = ?$
  - Step 5: $\gamma = ?$