ME751
Advanced Computational Multibody Dynamics

September 23, 2016

Dan Negrut
University of Wisconsin-Madison

Quote of the day

“A nickel ain’t worth a dime anymore.”
--Yogi Berra

Before we get started…

- Last Time:
  - Describing the orientation of a body using Euler parameters

- Today:
  - Wrap up, describing the orientation of a body using Euler parameters
  - Connection between ang. velocity and time derivatives of Euler parameters
  - Start discussion of Kinematic Analysis of mechanisms
    - Kinematic constraints & driving constraints
    - Degrees of freedom
    - Position, Velocity, and Acceleration analysis of a mechanism

- Misc. other issues
  - Third assignment emailed out today, uploaded onto the website
  - Due in one week

[pp.341]

Part 2, Case 1: $e_0 \neq 0$

- Recall that

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = 2 \begin{bmatrix} e_0^2 + e_1^2 - \frac{1}{2} & e_1 e_2 - e_0 e_3 & e_1 e_3 + e_0 e_2 \\ e_1 e_2 + e_0 e_3 & e_0^2 + e_2^2 - \frac{1}{2} & e_2 e_3 - e_0 e_1 \\ e_1 e_3 - e_0 e_2 & e_2 e_3 + e_0 e_1 & e_0^2 + e_3^2 - \frac{1}{2} \end{bmatrix}$$

- Then, with $e_0 \neq 0$ available to you (see previous slide), you get

$$a_{32} - a_{23} = 4e_0 e_1$$

$$a_{13} - a_{31} = 4e_0 e_2$$

$$a_{21} - a_{12} = 4e_0 e_3$$

Singularity-free extraction of a quaternion from a direct
matrix
A. R. KLUMPP
Journal of Spacecraft and Rockets, December, Vol. 13
pp. 754-755

- From where,

$$e_1 = \frac{a_{32} - a_{23}}{4e_0}$$

$$e_2 = \frac{a_{13} - a_{31}}{4e_0}$$

$$e_3 = \frac{a_{21} - a_{12}}{4e_0}$$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = 2 \begin{bmatrix} e_0^2 + e_1^2 - \frac{1}{2} & e_1e_2 - e_0e_3 & e_1e_3 + e_0e_2 \\ e_1e_2 + e_0e_3 & e_0^2 + e_2^2 - \frac{1}{2} & e_2e_3 - e_0e_1 \\ e_1e_3 - e_0e_2 & e_2e_3 + e_0e_1 & e_0^2 + e_3^2 - \frac{1}{2} \end{bmatrix}$$

[pp.341]

Part 2, Case 1: $e_0 \neq 0$

- Step 1: Compute e_0^2 as

$$e_0^2 = \frac{tr(\mathbf{A}) + 1}{4}$$

- Step 2: Compute e_0 from e_0^2; choose any sign you wish for e_0, maybe should be close to some previous value of it that you just computed

- Step 3: For $i = 1, 2, 3$, compute

$$e_i^2 = \frac{1 + 2a_{ii} - tr(\mathbf{A}) + 1}{4}$$

- Step 4: When computing e_i from e_i^2, the sign of e_i should be determined based on the following relations:

$$a_{32} - a_{23} = 4e_0e_1$$
$$a_{13} - a_{31} = 4e_0e_2$$
$$a_{21} - a_{12} = 4e_0e_3$$

“Singularity-free extraction of a quaternion from a direction-cosine matrix,”
A. R. Klumpp, Journal of Spacecraft and Rockets, December, Vol. 13, No. 12 : pp. 754-755

Exercise

- Prove that the four values computed on the previous slide verify the Euler Parameter normalization condition

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = 2 \begin{bmatrix} e_0^2 + e_1^2 - \frac{1}{2} & e_1 e_2 - e_0 e_3 & e_1 e_3 + e_0 e_2 \\ e_1 e_2 + e_0 e_3 & e_0^2 + e_2^2 - \frac{1}{2} & e_2 e_3 - e_0 e_1 \\ e_1 e_3 - e_0 e_2 & e_2 e_3 + e_0 e_1 & e_0^2 + e_3^2 - \frac{1}{2} \end{bmatrix}$$

[pp.341]

Part 2, Case 2: $e_0 = 0$

- Since $e_0 = \cos\frac{\chi}{2} = 0$, it means that $\chi = \pi$.
  - Note that we assume that χ is in between 0 and 2π
- How do you compute e_1, e_2, e_3 next?
- Their computation draws on the following remark: since now $e_1^2 + e_2^2 + e_3^2 = 1$, at least one of e_1, e_2, and e_3 is nonzero. Whichever that one is, you'll use it to solve for the other two using two out of the three following conditions:

$$a_{21} + a_{12} = 4e_1 e_2$$

$$a_{31} + a_{13} = 4e_1 e_3$$

$$a_{32} + a_{23} = 4e_2 e_3$$

  - Note that you'll have the same sign issue like we saw for the $e_0 \neq 0$ case. However, we understand what that means and what its implications are

Euler Parameters:
The One-To-One Mapping to A – Concluding Remarks

- One-To-One Mapping, Euler Parameter to Body Orientation, final word:
  - Things worked out well…

…(with the caveat that you have for each **A** two **p** sets, yet they are far apart and there is no danger to get confused)

[Challenge Homework]
Exercise

- (Did this already for $e_0 \neq 0$) Assume that given $\mathbf{A}$, you have just obtained e_0, e_1, e_2, e_3, as indicated before. Prove that these values satisfy the Euler Parameter normalization constraint. That is,

$$e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$$

- Next, prove that for the $\mathbf{p}$ you just got, if you do $\mathbf{E}\mathbf{G}^T$ you get back $\mathbf{A}$
- Keep in mind that you have two cases: first, when $e_0 \neq 0$; second, when $e_0 = 0$
- NOTE: I couldn't find a proof for this anywhere I looked

[New Topic]

Euler Parameter Identities

- We'll discuss a couple of identities (formulas) that involve the Euler parameters. Why?
  - They are needed for the discussion regarding the angular velocity
  - They are needed when discussing about the concept of virtual displacement
- There is very little intuition behind these identities (at least to me)
  - Take them as they are: some helpers who are going to show up here and there in proving various results that involve the matrix **A** or its time derivatives

[we saw this before]

Position Level Identities

[Level 0 Identities][p.343]

- There are two 3 × 4 matrices that show up quite often:

$$\mathbf{E} \equiv [-\mathbf{e}, \tilde{\mathbf{e}}+e_0\mathbf{I}] = \begin{bmatrix} -e_1 & e_0 & -e_3 & e_2 \\ -e_2 & e_3 & e_0 & -e_1 \\ -e_3 & -e_2 & e_1 & e_0 \end{bmatrix} \qquad \mathbf{G} \equiv [-\mathbf{e}, -\tilde{\mathbf{e}}+e_0\mathbf{I}] = \begin{bmatrix} -e_1 & e_0 & e_3 & -e_2 \\ -e_2 & -e_3 & e_0 & e_1 \\ -e_3 & e_2 & -e_1 & e_0 \end{bmatrix}$$

- The key identity is this [HOMEWORK]:

$$\mathbf{A} = \mathbf{E}\mathbf{G}^T$$

- Also note the following four identities:

1. $\mathbf{E}\mathbf{p} = \mathbf{0}_3$
2. $\mathbf{G}\mathbf{p} = \mathbf{0}_3$
3. $\mathbf{E}\mathbf{E}^T = \mathbf{G}\mathbf{G}^T = \mathbf{I}_3$
4. $\mathbf{G}^T\mathbf{G} = \mathbf{E}^T\mathbf{E} = \mathbf{I}_4 - \mathbf{p}\mathbf{p}^T$

Velocity Level Identities

[Level 1 Identities][p.344]

- Note the following identities that include time derivatives of the Euler Parameters:

  1. $\mathbf{p}^T\dot{\mathbf{p}} = \dot{\mathbf{p}}^T\mathbf{p} = \mathbf{0}$
  2. $\mathbf{E}\dot{\mathbf{p}} = -\dot{\mathbf{E}}\mathbf{p}$
  3. $\mathbf{G}\dot{\mathbf{p}} = -\dot{\mathbf{G}}\mathbf{p}$
  4. $\mathbf{E}\dot{\mathbf{G}}^T = \dot{\mathbf{E}}\mathbf{G}^T$
  5. $\dot{\mathbf{A}} = \dot{\mathbf{E}}\mathbf{G}^T + \mathbf{E}\dot{\mathbf{G}}^T = \mathbf{2}\mathbf{E}\dot{\mathbf{G}}^T$

- A though one to prove is the following (bottom pp.344):

$$\widetilde{\mathbf{G}\dot{\mathbf{p}}} = \mathbf{G}\dot{\mathbf{G}}^T$$

Expressing the Angular Velocity when Using Euler Parameters as GCs

$\dot{\mathbf{p}}$ given; $\quad \omega = ?$

$$\tilde{\bar{\omega}} = \mathbf{A}^T\dot{\mathbf{A}} = 2\mathbf{G}\mathbf{E}^T\mathbf{E}\dot{\mathbf{G}}^T$$

$$\Downarrow$$

$$\tilde{\bar{\omega}} = 2\mathbf{G}\dot{\mathbf{G}}^T$$

$$\Downarrow$$

$$\tilde{\bar{\omega}} = 2(\widetilde{\mathbf{G}\dot{\mathbf{p}}})$$

$$\Downarrow$$

$$\bar{\omega} = 2(\mathbf{G}\dot{\mathbf{p}})$$

$$\omega = \mathbf{A}\bar{\omega} = 2\mathbf{E}\mathbf{G}^T\mathbf{G}\dot{\mathbf{p}}$$

$$\Downarrow$$

$$\omega = 2\mathbf{E}\dot{\mathbf{p}}$$

pp.344, Haug’s book

ω given; $\quad \dot{\mathbf{p}} =?$

$$\begin{tikzcd}
 & \bar{\omega} = 2\mathbf{G}\dot{\mathbf{p}} \arrow[d] & \\
 & \mathbf{G}^T\bar{\omega} = 2\mathbf{G}^T\mathbf{G}\dot{\mathbf{p}} \arrow[dl] & \\
\dot{\mathbf{p}} = \dfrac{1}{2}\mathbf{G}^T\bar{\omega} \arrow[rr] & & \dot{\mathbf{p}} = \dfrac{1}{2}\mathbf{E}^T\omega
\end{tikzcd}$$

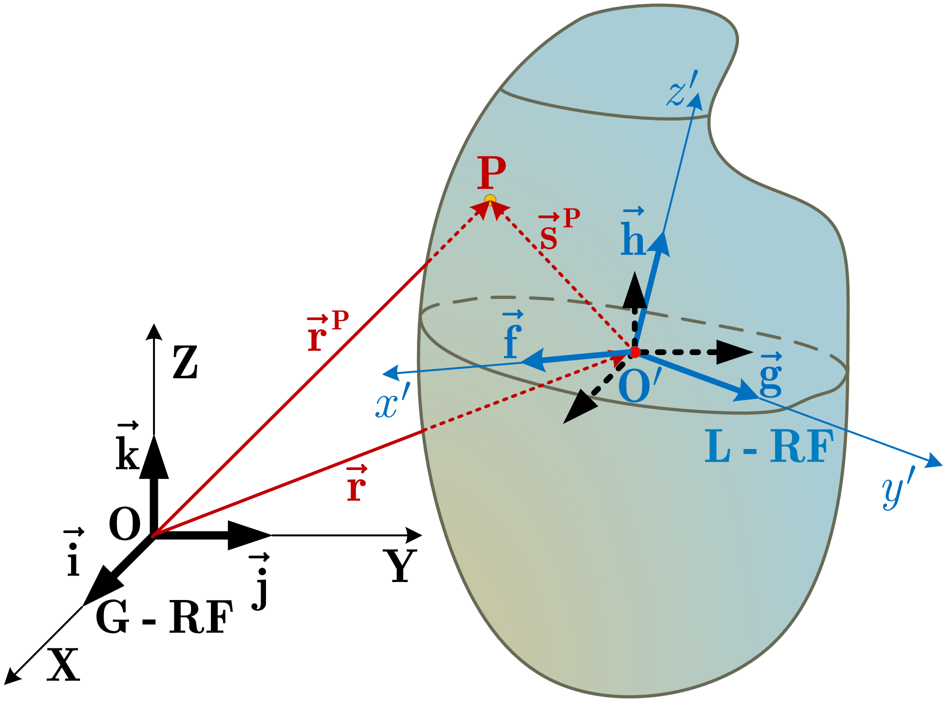
pp.345, Haug’s book

[New Topic: Combining Translation and Rotation]

Full 3D Kinematics of Rigid Bodies

- So far, we focused on the rotation of a rigid body
- Scenario used: the body was connected to ground through a spherical joint that allowed it to experience an arbitrary rotation
- Yet bodies are in general experiencing both translation and rotation

- Framework and Notation Conventions:
  - A L-RF is attached to the rigid body at some location denoted by O'
  - Relative to the G-RF, point O' is located by vector $\vec{\mathbf{r}}$
  - L-RF defined by vectors $\vec{\mathbf{f}}$, $\vec{\mathbf{g}}$, $\vec{\mathbf{h}}$
  - An arbitrary point P of the rigid body is considered. Its location relative to the L-RF is provided through the vector $\vec{\mathbf{s}}^P$



3D Rigid Body Kinematics:

Determining Position of Arbitrary Point P

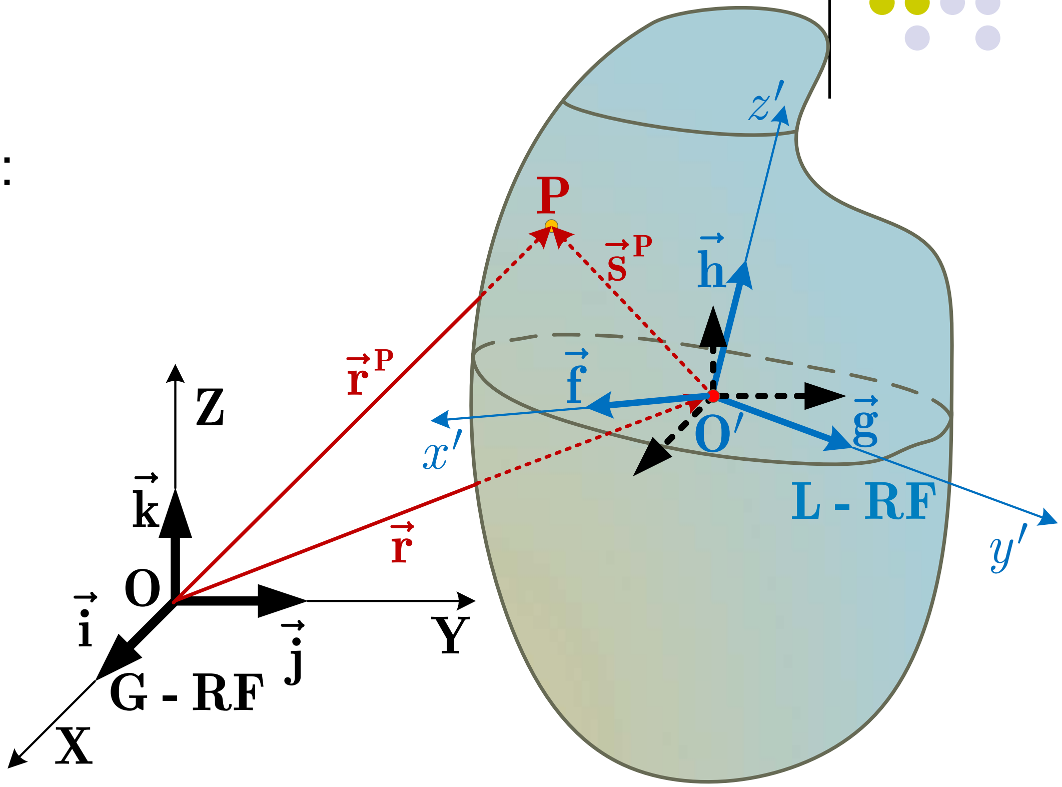
[Very Important to Understand]

- In the Geometric Vector world:

$$\overrightarrow{OP} = \overrightarrow{OO'} + \overrightarrow{O'P}$$

$$\Downarrow$$

$$\vec{\mathbf{r}}^P = \vec{\mathbf{r}} + \vec{\mathbf{s}}^P$$



- Algebraic Vector world:

$$\mathbf{r}^P = \mathbf{r} + \mathbf{s}^P = \mathbf{r} + \mathbf{A}\bar{\mathbf{s}}^P$$

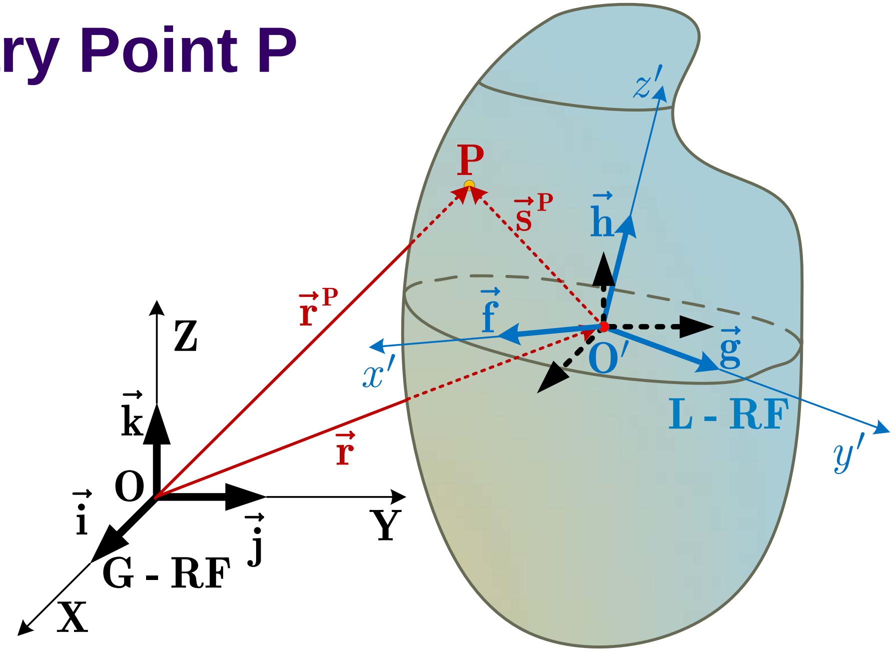
- Important observation:
  - The vector $\bar{\mathbf{s}}^P$ that provides the location of P in the L-RF is a constant vector
    - $*$ True because the body is assumed to be rigid

3D Rigid Body Kinematics:

Determining Velocity of Arbitrary Point P

- In the Geometric Vector world, by definition:

$$\vec{\mathbf{v}}^P = \frac{d\vec{\mathbf{r}}^P}{dt} = \dot{\vec{\mathbf{r}}} + \dot{\vec{\mathbf{s}}}^P = \dot{\vec{\mathbf{r}}} + \vec{\omega} \times \vec{\mathbf{s}}^P$$



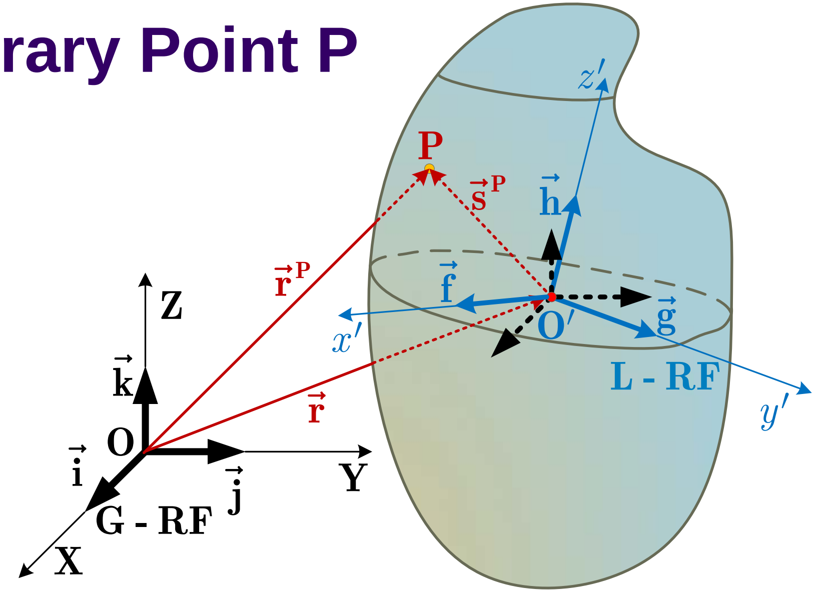
- Using the Algebraic Vector representation:

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \dot{\mathbf{s}}^P = \dot{\mathbf{r}} + \dot{\mathbf{A}}\bar{\mathbf{s}}^P = \dot{\mathbf{r}} + \tilde{\omega}\mathbf{A}\bar{\mathbf{s}}^P = \dot{\mathbf{r}} + \tilde{\omega}\mathbf{s}^P$$

- In plain words: the velocity $\dot{\mathbf{r}}^P$ of a point P is equal to the sum of the velocity $\dot{\mathbf{r}}$ of the point where the L-RF is located and the velocity $\tilde{\omega}\mathbf{s}^P$ due to the rotation with angular velocity ω of the rigid body

3D Rigid Body Kinematics:

Determining Acceleration of Arbitrary Point P



- In the Geometric Vector world, by definition:

$$\vec{\mathbf{a}}^P \equiv \frac{d^2\vec{\mathbf{r}}^P}{dt^2} = \ddot{\vec{\mathbf{r}}} + \vec{\omega} \times \vec{\omega} \times \vec{\mathbf{s}}^P + \dot{\vec{\omega}} \times \vec{\mathbf{s}}^P$$

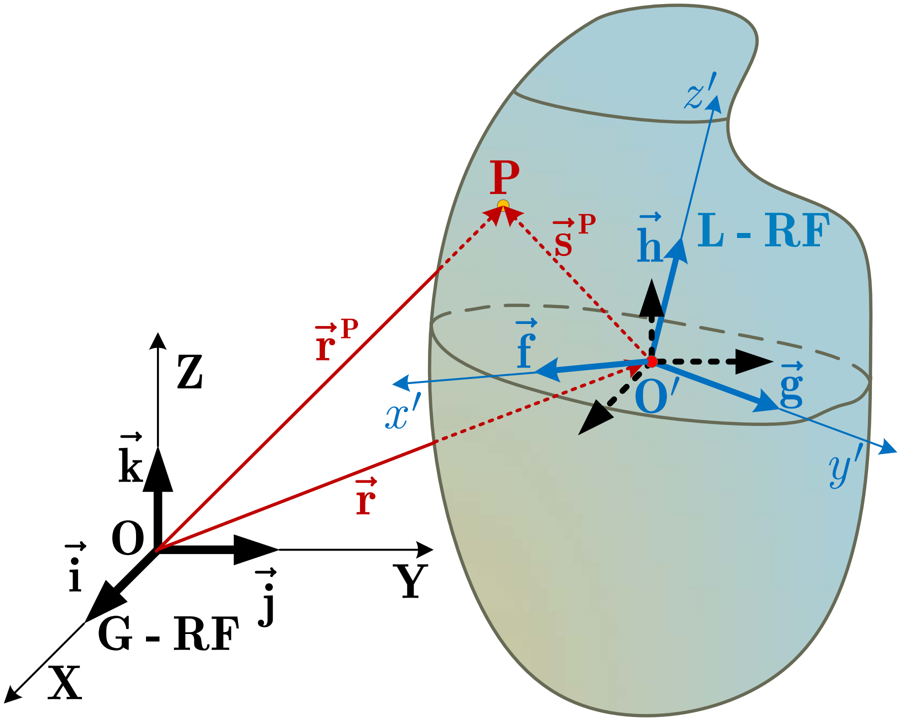
- Using the Algebraic Vector representation:

$$\mathbf{a}^P \equiv \ddot{\mathbf{r}}^P = \ddot{\mathbf{r}} + \ddot{\mathbf{s}}^P = \ddot{\mathbf{r}} + \tilde{\omega}\tilde{\omega}\mathbf{A}\bar{\mathbf{s}}^P + \tilde{\dot{\omega}}\mathbf{A}\bar{\mathbf{s}}^P = \ddot{\mathbf{r}} + \tilde{\omega}\tilde{\omega}\mathbf{s}^P + \tilde{\dot{\omega}}\mathbf{s}^P$$

3D Translation and Rotation OK. Now What?

- Given arbitrary point P on a rigid body, we know how to the following:
  - Capture its position with respect to both a G-RF and a L-RF
  - Compute its velocity
  - Compute its acceleration

- This will become important when we discuss how to express in *mathematical terms* the fact that the motion of a body is constrained by the presence of joints that limit its relative motion to ground or to other bodies in a mechanical system
  - We start with a geometric perspective on the relative motion between two bodies and then formulate a set of equations in terms of algebraic vectors that enforce the kinematics; that is, capture the effect of the joint connecting the two bodies



End: Kinematics of a Rigid Body in 3D

Begin: Kinematics Analysis of Mech. System

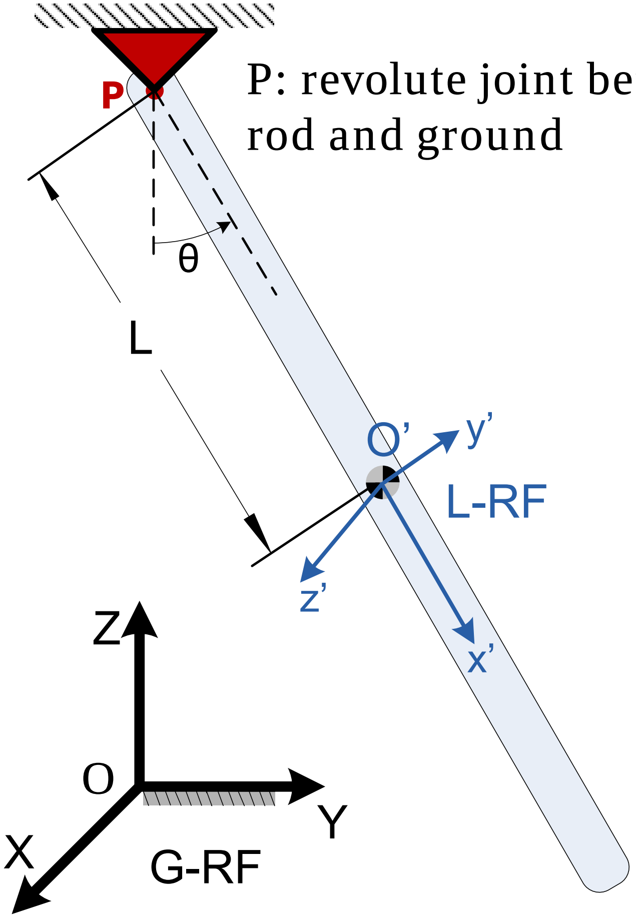
Kinematics Analysis: Definition

- Kinematics Analysis – the process of computing the position, velocity, and acceleration of a system of interconnected bodies that make up a mechanical system <u>independent</u> of the forces that produce its motion

Motivating Example: Motion of Simple Pendulum

- A revolute (hinge) joint present at point **P**
- A motion „(t)=4t^2 is applied to the pendulum
- Find the time evolution of this pendulum

Simple 3D Pendulum
(connected to ground at point P)



P: revolute joint between rod and ground

Kinematic Analysis Stages

- Position Analysis Stage
  - Challenging

- Velocity Analysis Stage
  - Simple

- Acceleration Analysis Stage
  - Kind of OK

- To take care of all these stages, ONE step is critical:
  - Write down the constraint equations associated with the joints present in your mechanism
- Once you have the constraints, the rest is boilerplate

Why is Kinematics Important?

- It can be an end in itself…
  - *Kinematic Analysis* - Interested how components of a certain mechanism move when motion[s] are applied
  - *Kinematic Synthesis* – Interested in finding how to design a mechanism to perform a certain operation in a certain way
    - NOTE: ME751 only covers Kinematic Analysis

- Important to understand Kinematics
  - The building blocks of the Kinematics infrastructure recycled when assembling the infrastructure for the Kinetic problem (“Dynamics Analysis”, discussed in Chapter 11)

- In general, the Dynamic Analysis sees more mileage compared to the Kinematic Analysis of mechanism

Nomenclature & Conventions

[1st out of 2]

- We are dealing with rigid bodies only

- Recall this: L-RF – is a <u>body fixed Reference Frame</u> used to describe the position and orientation of a rigid body in the 3D space

- Body Cartesian generalized coordinates (GCs)
  - Used to define the position and orientation of the L-RF mentioned above
  - We'll use Euler Parameters (less headaches than Euler Angles…)
  - For body "i" the GCs that we'll work with are

$$\mathbf{q}_i = \begin{bmatrix}\mathbf{r}_i\\ \mathbf{p}_i\end{bmatrix} = \left[\begin{array}{c} x_i\\ y_i\\ z_i\\ e_{0,i}\\ e_{1,i}\\ e_{2,i}\\ e_{3,i}\end{array}\right] \begin{array}{l}\left.\vphantom{\begin{array}{c}x\\ y\\ z\end{array}}\right\}\text{Tell us where the body is located}\\ \left.\vphantom{\begin{array}{c}e\\ e\\ e\\ e\end{array}}\right\}\text{Tell us how the body is oriented}\end{array}$$

Note: $\mathbf{q}_i \in \mathbb{R}^7$

Nomenclature & Conventions

[2/2]

- The system GSs – the array of all bodies GCs

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \cdots \\ \mathbf{q}_{nb} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ \vdots \\ e_{2,nb} \\ e_{3,nb} \end{bmatrix} \equiv \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ \vdots \\ q_{nc} \end{bmatrix} \in \mathbb{R}^{nc}$$

- NOTE: for a mechanism with *nb* bodies, the number *nc* of Cartesian generalized coordinates is

$$nc = 7 \cdot nb$$

  - “nc” stands for “number of coordinates”

- Recall we have a number of *nb* “Euler Parameter normalization constraints”:

$$\mathbf{p}_i^T \cdot \mathbf{p}_i = 1 \; , \qquad i = 1, 2, \ldots, nb$$

Putting Things in Perspective

- Before getting lost in the details of the Kinematics Analysis:
  - Recall that we presented a collection of terms that will help understand the “language” of Kinematics
  - We are about to give a 30,000 feet perspective of things to come to justify the need for the material presented over the next two lectures
    - Among the concepts introduced today, here are the more important ones:
      - Constraint equations (as a means to specifying the geometry associated with the motion of a mechanism)
      - Jacobian matrix (or simply, the Jacobian)

Joints (Physical System)
vs.
Constraint Equations (Virtual System)

- Physical Mechanical System (“mechanism”, “assembly”):
  - Uses joints to connect bodies
  - Some of its elements (components) are driven in a predefined fashion

- Virtual System:
  - A set of constraint equations needs to be specified to capture the effect of the joints present in the physical model
  - Some of these equations will be time dependent to capture motions

Joints (Physical System) vs. Constraint Equations (Virtual System)

- Physical Mechanical System (“mechanism”, “assembly”):
  - Uses joints to connect bodies
  - Some of its elements (components) are driven in a predefined fashion

- Virtual Mechanical System (“model”):
  - A set of constraint equations established to capture the effect of the joints present in the physical model
  - Some of these equations will explicitly depend on time to capture motions

- Constraint Equations, taxonomy
  - Holonomic vs. Nonholonomic constraint
    - Holonomic: only depends on generalized coordinates, not on their time derivative
  - Scleronomic (“Kinematic”) vs. Rheonomic (“Driving”) constraints

$$\Downarrow \qquad\qquad\qquad\qquad\qquad\qquad \Downarrow$$

$$\mathbf{\Phi}^K(\mathbf{q}) = \mathbf{0} \qquad\qquad\qquad\qquad \mathbf{\Phi}^D(\mathbf{q}, t) = \mathbf{0}$$

Kinematic Constraints, $\mathbf{\Phi}^K(\mathbf{q}) = \mathbf{0}$

- What are they, and what role do they play?
  - A collection of equations that, if satisfied, coerce the bodies in the model to move like the bodies of the mechanism
    - They enforce the geometry of the motion

- Most important thing in relation to constraints:
  - For each joint in the model, the equations of constraint that you use must imply the relative motion allowed by the joint
    - This is where we'll spend a lecture

  - Keep in mind: the way you **model** should resemble the **physical system** (the geometry of the motion)

- Notation: We'll use m_K to denote the number of kinematic (or scleronomic) constraints present in the model:

$$\mathbf{\Phi}^K(\mathbf{q}) \in \mathbb{R}^{m_K}$$

Driving Constraints, $\mathbf{\Phi}^{D}(\mathbf{q}, t) = \mathbf{0}$

- What are they, and what role do they play?
  - An equation, such as „(t)=` t + `/2, that specifies how a generalized coordinate (GC) changes in time (the explicit dependence of that GC on time)
  - A driving constraint can be set up such that it involves in one equation more than just one GC
  - Bottom line: driving constraints define (“prescribe”, “drive”) the motion

- Notation:
  - We’ll use m_D to denote the number of driving (or rheonomic) constraints present in the model

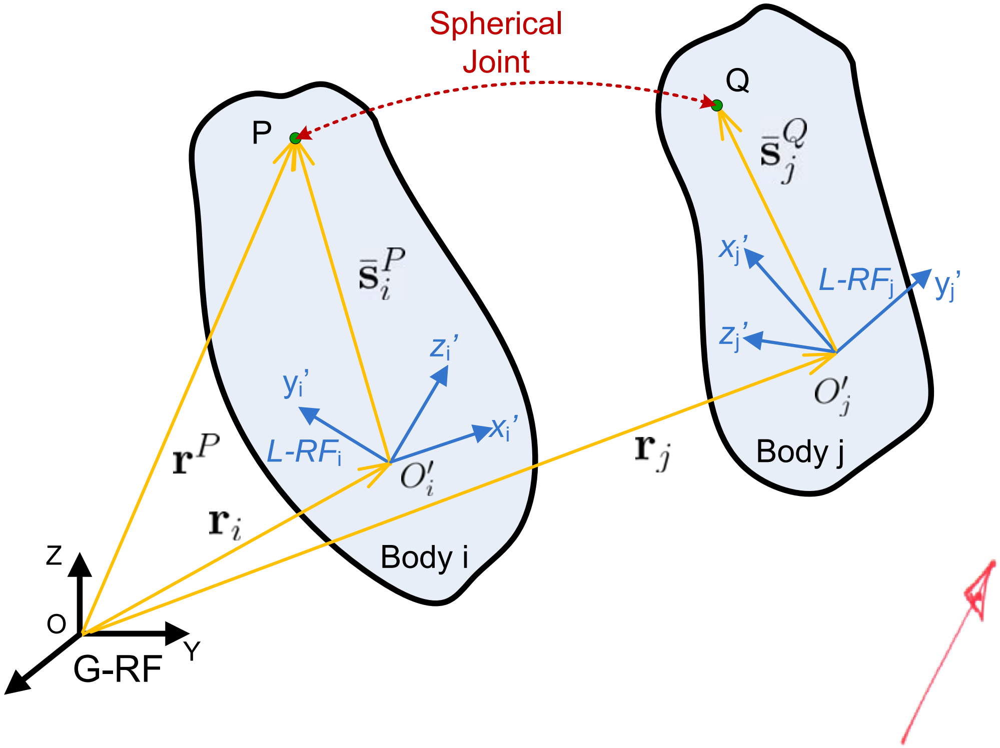
$$\mathbf{\Phi}^{D}(\mathbf{q}, t) \in \mathbb{R}^{m_D}$$

  - We’ll use m to denote the total number of constraints (kinematic and driving) present in the model:

$$m = m_K + m_D$$

Example:
Handling a Spherical Joint

- Define a set of Kinematic Constraints that reflect the existence of a spherical joint between points P on body i and Q on body j



$$\mathbf{r}^P = \mathbf{r}_i + \mathbf{s}_i^P = \mathbf{r}_i + \mathbf{A}_i(\mathbf{p}_i)\bar{\mathbf{s}}_i^P$$

$$\mathbf{r}^Q = \mathbf{r}_j + \mathbf{s}_j^Q = \mathbf{r}_j + \mathbf{A}_j(\mathbf{p}_j)\bar{\mathbf{s}}_j^Q$$

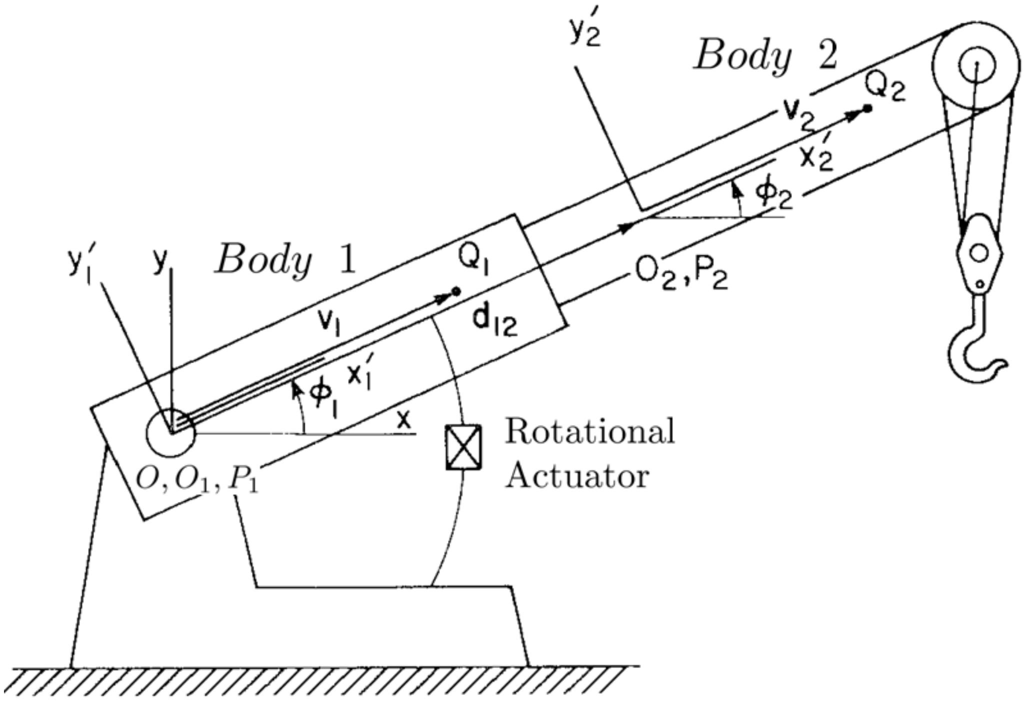
$$\boxed{\mathbf{r}^P = \mathbf{r}^Q} \quad \Rightarrow \quad \mathbf{r}_i + \mathbf{A}_i\bar{\mathbf{s}}_i^P = \mathbf{r}_j + \mathbf{A}_j\bar{\mathbf{s}}_j^Q$$

$$\mathbf{\Phi}(\mathbf{q}_i, \mathbf{q}_j) = \mathbf{r}_i + \mathbf{A}_i\bar{\mathbf{s}}_i^P - \mathbf{r}_j - \mathbf{A}_j\bar{\mathbf{s}}_j^Q = \mathbf{0}_3$$

- Note that the spherical joint condition is enforced by requiring that the points **P** and **Q** coincide at all times

Example: Specifying Motions

- Wrecker boom with two motions prescribed (ME451 example)



- Prescribed motions:

$$||P_1P_2|| = 3 + 0.1t$$

$$\phi_1(t) = 0.1t$$

Degrees of Freedom

- Number of degrees of freedom (NDOF, ndof) is equal to total number of generalized coordinates minus the number of constraints that these coordinates must satisfy
  - Sometimes also called “Gruebler Count”

$$NDOF = nc - m_K - m_D$$

- Quick Remarks:
  - NDOF is an attribute of the model, and it is independent of the set of generalized coordinates used to represent the motion of the mechanism
  - When using Euler Parameters for body orientation, m_K should also include the set of nb normalization constraints

- In general, for carrying out Kinematic Analysis, $NDOF = 0$
  - For Dynamics Analysis, we need $NDOF \geq 0$

Motion: Causes

- How can one set a mechanical system in motion?
  - Approach leading to Kinematic Analysis
    - Prescribe motions for various components of the mechanical system until NDOF=0
    - For a well posed problem, you'll be able to uniquely determine **q**(t) as the solution of an algebraic problem

(Forget this for now…)

  - Approach leading to Dynamics Analysis
    - Apply a set of forces upon the mechanism and specify a number of motions, but when doing the latter make sure you end up with NDOF $\geq$ 0
    - For a well posed problem, **q**(t) found as the solution of a differential problem

Position Analysis

- How do you get the position configuration of the mechanism?
  - Kinematic Analysis key observation: The number of constraints (kinematic and driving) is equal to the number of generalized coordinates: $m = nc$
    - This is a prerequisite condition for Kinematic Analysis

$$\mathbf{\Phi}(\mathbf{q}, t) = \begin{bmatrix} \mathbf{\Phi}^K(\mathbf{q}) \\ \mathbf{\Phi}^D(\mathbf{q}, t) \end{bmatrix}_{nc \times 1} = \mathbf{0}$$

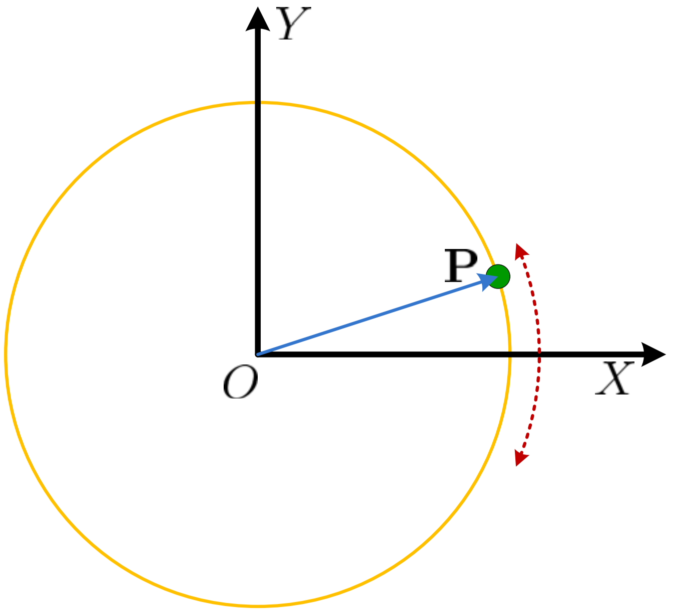
$$\mathbf{q} \in \mathbb{R}^{nc} \qquad \mathbf{\Phi} : \mathbb{R}^{nc+1} \rightarrow \mathbb{R}^{nc}$$

IMPORTANT: This is a nonlinear systems with *nc* equations and *nc* unknowns that you must solve to find **q**

- The solution of the nonlinear system is found by using the so called “Newton-Raphson” algorithm
  - We’ll elaborate on this later, for now just assume that you have a way to solve the above nonlinear system to find the solution **q**(t)

Exercise: Kinematic Analysis

- A particle moves on a circle of radius 1
- The generalized coordinates used are $\mathbf{q} = [x \ , \ y]^T$
- The y coordinate has a prescribed motion: $y(t) = 0.1\sin(50\pi t)$
- Carry our Position Analysis for the given one particle system



Velocity Analysis

- Take one time derivative of constraints Φ(q,t) to obtain the **velocity equation**:

$$\frac{d}{dt}\boldsymbol{\Phi}(\mathbf{q},t) = \mathbf{0} \qquad \Rightarrow \qquad \boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = \underbrace{-\boldsymbol{\Phi}_t}_{\nu}$$

- The Jacobian has as many rows (m) as it has columns (nc) since for Kinematics Analysis, $NDOF = nc - m \ = 0$

- Therefore, you have a linear system that you need to solve to recover $\dot{\mathbf{q}}$

$$\boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = \nu$$

Acceleration Analysis

- Take yet one more time derivative to obtain the **acceleration equation:**

$$\ddot{\mathbf{\Phi}} = \frac{d^2}{dt^2}\mathbf{\Phi}(\mathbf{q},t) = \mathbf{0} \qquad \Rightarrow \qquad \mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = \underbrace{-\left(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}\right)_{\mathbf{q}}\dot{\mathbf{q}} - 2\mathbf{\Phi}_{\mathbf{q}t}\,\dot{\mathbf{q}} - \mathbf{\Phi}_{tt}}_{\gamma}$$

- NOTE: Getting right-hand side of acceleration equation is tedious
  - One observation that simplifies the computation: note that the right side of the above equation is made up of everything in the expression of $\ddot{\mathbf{\Phi}}$ that does *not* depend on second time derivatives (accelerations)
    - This is a very useful observation in the derivation of the RHS of the acceleration equation

- Just like we pointed out for the velocity analysis, you also have to solve a linear system to retrieve the acceleration $\ddot{\mathbf{q}}$

$$\mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = \gamma$$