ME451
Kinematics and Dynamics of Machine Systems

Elements of 2D Kinematics

September 18, 2014

Dan Negrut
ME451, Fall 2014
University of Wisconsin-Madison

- Quote of the day: “Better to fight for something than live for nothing.” – George Patton

Before we get started…

- Last time
  - Wrapped up partial derivatives (sensitivity computation, or Jacobian computation)
    - Focus was on chain rule
  - Started quick overview of MATLAB
  - Experienced the great excitement and thrill of a fire drill

- Today
  - Wrap up MATLAB overview
  - Start discussion on the Kinematics of 2D mechanisms

- HW:
  - Haug's book: 2.5.11, 2.5.12, 2.6.1, 3.1.1, 3.1.2
  - MATLAB (emailed to you later today)
    - Due on Th, 9/25, at 9:30 am
    - Post questions on the forum
    - Drop MATLAB assignment in learn@uw dropbox

2.6

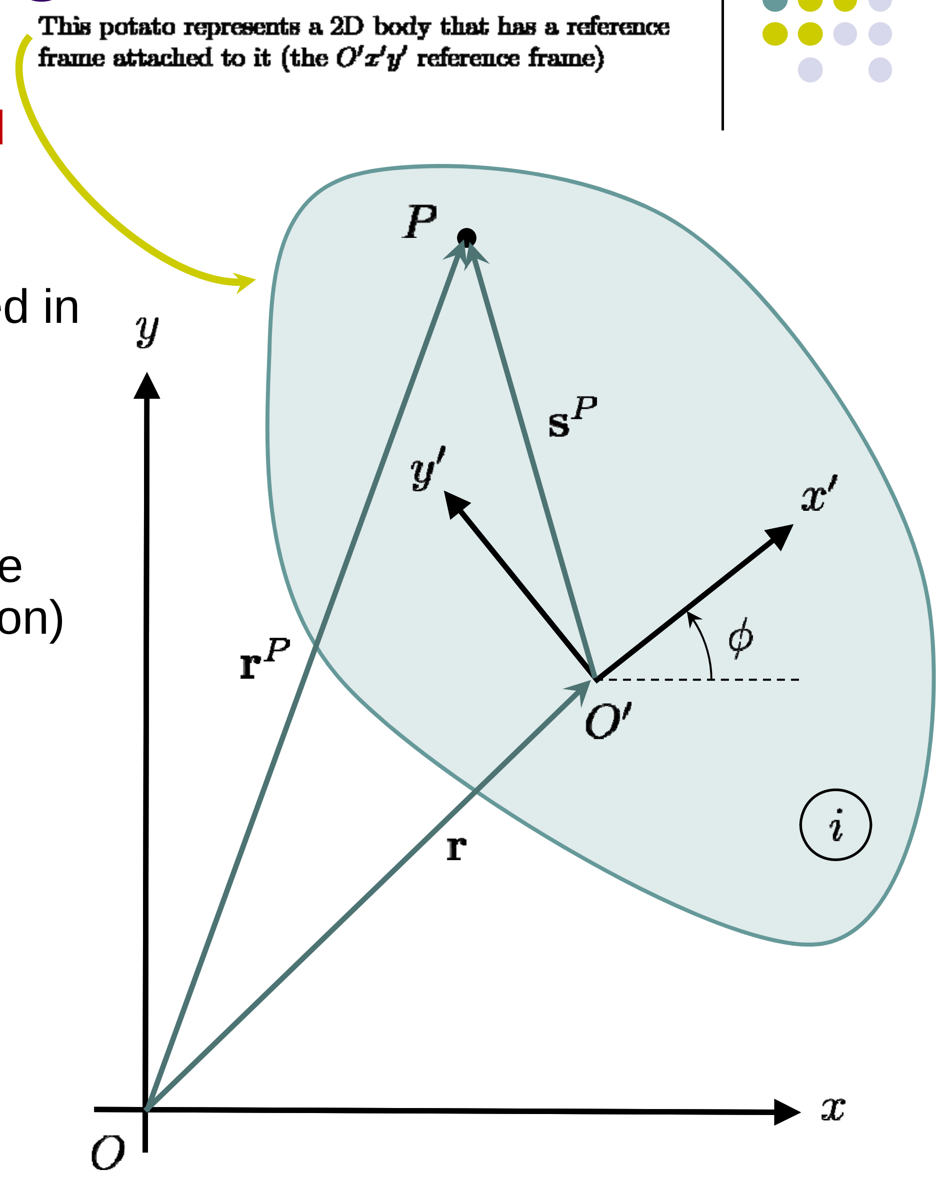
VELOCITY AND ACCELERATION OF A POINT FIXED IN A MOVING FRAME

Velocity and Acceleration of a Point Fixed in a Moving Frame

- A moving rigid body and a point P, fixed (rigidly attached) to the body
- The position vector of point P, expressed in the GRF is:

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

and changes in time because both $\mathbf{r}$ (the body position) and $\mathbf{A}$ (the body orientation) change.



Questions:

- What is the velocity of P?
  That is, what is $\dot{\mathbf{r}}^P$?
- What is the acceleration of P?
  That is, what is $\ddot{\mathbf{r}}^P$?

Keep This in Mind…

- The body (the potato on the previous slide) moves around
  - Translation (it slides)
  - Rotation (it tumbles)

- Therefore,
  - The location of the $O'x'y'$ frame will be a function of time t:

$$\mathbf{r} = \mathbf{r}(t)$$

  - The orientation of the $O'x'y'$ frame will be a function of time t. That is, ϕ in $\mathbf{A}(\phi)$ is a function of time:

$$\phi = \phi(t)$$

Some Preliminaries

- Orthogonal Rotation Matrix $\mathbf{R} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{R} = \mathbf{A}\left(\frac{\pi}{2}\right)$
  - Note that, when applied to a vector, this rotation matrix produces a new vector that is perpendicular to the original vector (counterclockwise rotation)

$$\forall \mathbf{v} \in \mathbb{R}^2 \quad \Rightarrow \quad \mathbf{v}^{\perp} = \mathbf{R}\mathbf{v}$$

- The matrix $\mathbf{B}(\phi) = \begin{bmatrix} -\sin\phi & -\cos\phi \\ \cos\phi & -\sin\phi \end{bmatrix}, \qquad \mathbf{B}(\phi) \triangleq \frac{\partial \mathbf{A}(\phi)}{\partial \phi}$
  - The **B** matrix is always associated with a rotation matrix **A**.
- Important relations (easy to check):

$$\mathbf{R}^2 = \mathbf{R}\mathbf{R} = -\mathbf{I}_2 \quad \Rightarrow \quad \mathbf{R}^2\mathbf{v} = -\mathbf{v}$$

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

$$\dot{\mathbf{A}} = \mathbf{A}_{\phi}\dot{\phi} \equiv \dot{\phi}\mathbf{B}$$

$$\dot{\mathbf{B}} = \mathbf{B}_{\phi}\dot{\phi} = -\dot{\phi}\mathbf{A}$$

Velocity of a Point Fixed in a Moving Frame

- Something to keep in mind: we'll manipulate quantities that depend on the generalized coordinates, which in turn depend on time
  - Specifically, the orientation matrix **A** depends on the generalized coordinate ϕ , which is itself a function of *t*

- This is where the [time and partial] derivatives discussed before come into play

$$\mathbf{r}^P(t) = \mathbf{r}(t) + \mathbf{A}(\phi(t))\mathbf{s}'^P$$
$$\Downarrow$$
$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \dot{\mathbf{A}}\mathbf{s}'^P = \dot{\mathbf{r}} + \dot{\phi}\mathbf{B}\mathbf{s}'^P$$

Acceleration of a Point Fixed in a Moving Frame

- Same idea as for velocity, except that you need two time derivatives to get accelerations

$$\mathbf{r}^P(t) = \mathbf{r}(t) + \mathbf{A}(\phi(t))\mathbf{s}'^P$$

$$\Downarrow$$

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \dot{\mathbf{A}}\mathbf{s}'^P = \dot{\mathbf{r}} + \dot{\phi}\mathbf{B}\mathbf{s}'^P$$

$$\Downarrow$$

$$\ddot{\mathbf{r}}^P = \ddot{\mathbf{r}} + \ddot{\phi}\mathbf{B}\mathbf{s}'^P + \dot{\phi}\dot{\mathbf{B}}\mathbf{s}'^P = \ddot{\mathbf{r}} + \ddot{\phi}\mathbf{B}\mathbf{s}'^P - \dot{\phi}^2\mathbf{A}\mathbf{s}'^P$$

Example

Given:

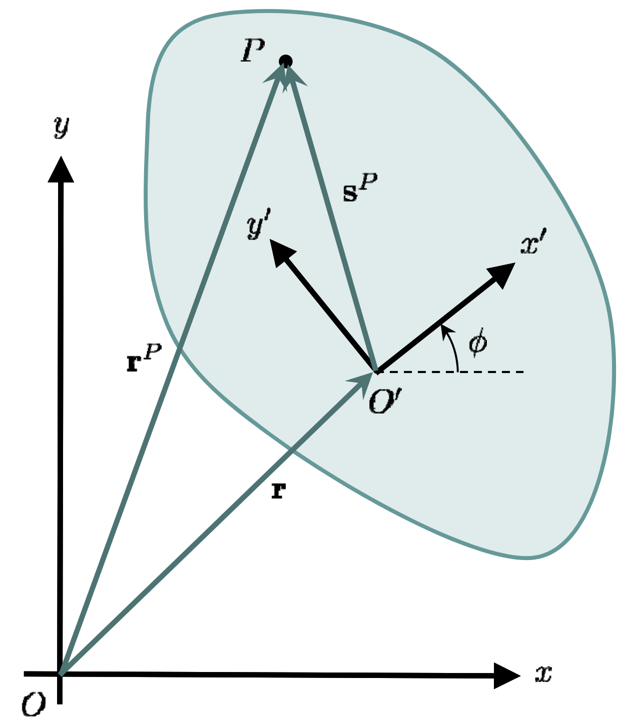
Position of P in LRF $\mathbf{s}'^P = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$

Position and orientation of LRF $\mathbf{r} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$ $\phi = \frac{\pi}{3}$

Linear and angular vel. of LRF $\dot{\mathbf{r}} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$ $\omega = 3$

Linear and angular acc. of LRF $\ddot{\mathbf{r}} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$ $\dot{\omega} = 5$

Find:	Position of P in GRF	$\mathbf{r}^P = ?$
	Velocity of P in GRF	$\dot{\mathbf{r}}^P = ?$
	Acceleration of P in GRF	$\ddot{\mathbf{r}}^P = ?$



ABSOLUTE (CARTESIAN) VS. RELATIVE GENERALIZED COORDINATES

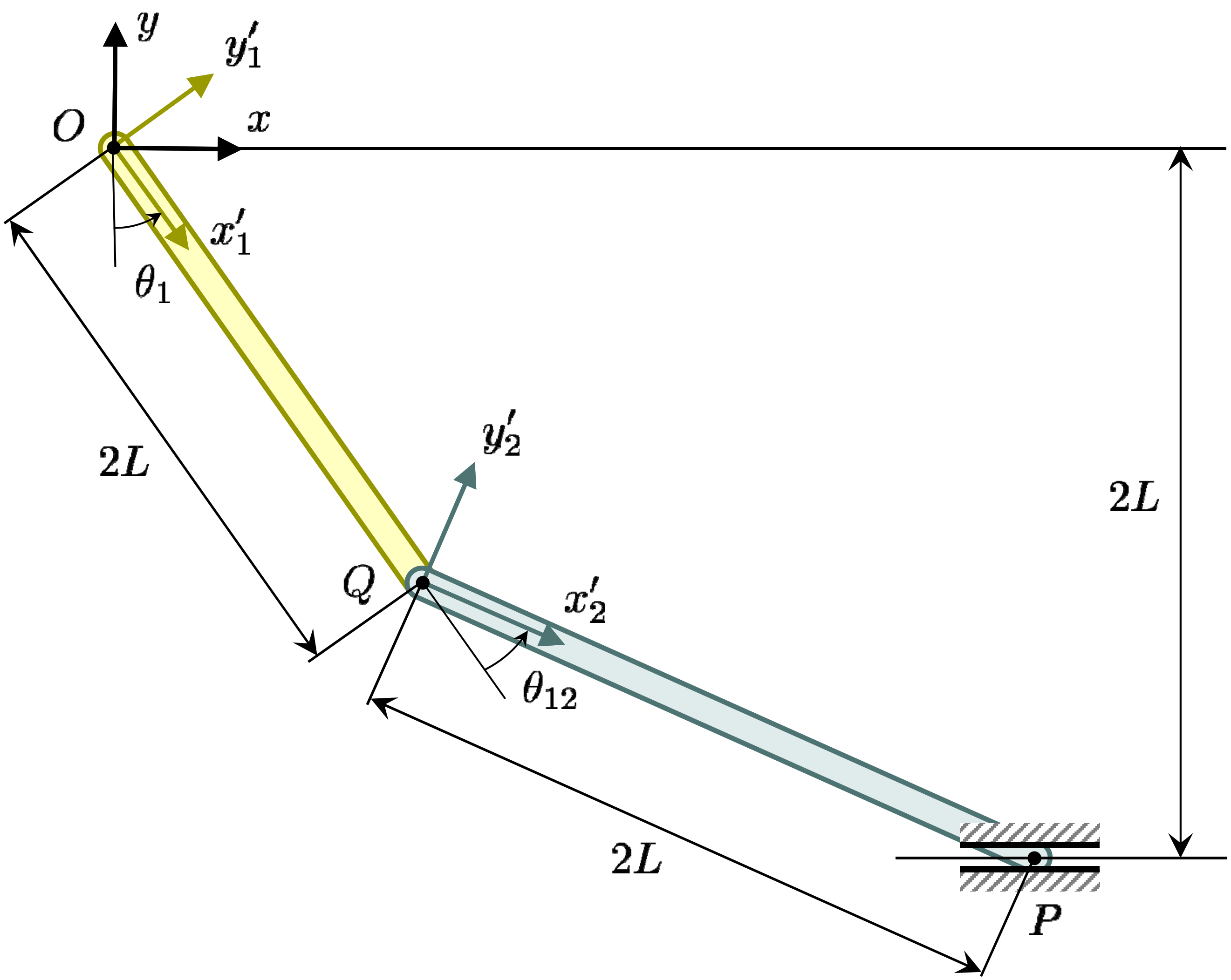
Generalized Coordinates
General Comments

- What are Generalized Coordinates (GC)?
  - A set of quantities (variables) that uniquely determine the state of the mechanism
    - the location of each body
    - the orientation of each body
    - (and from these, the position of any point on any body)
  - These quantities change in time since a mechanism changes/evolves in time
    - In other words, the generalized coordinates are functions of time
  - The rate at which the generalized coordinates change define the set of generalized velocities
    - Most often, obtained as the straight time derivative of the generalized coordinates
    - Sometimes this is not the case though

      Example: in 3D Kinematics, there is no generalized coordinate whose time derivative is the angular velocity
- Important remark: there are multiple ways of choose the set of generalized coordinates that describe the state of your mechanism

[handout]

Example (Relative GC)

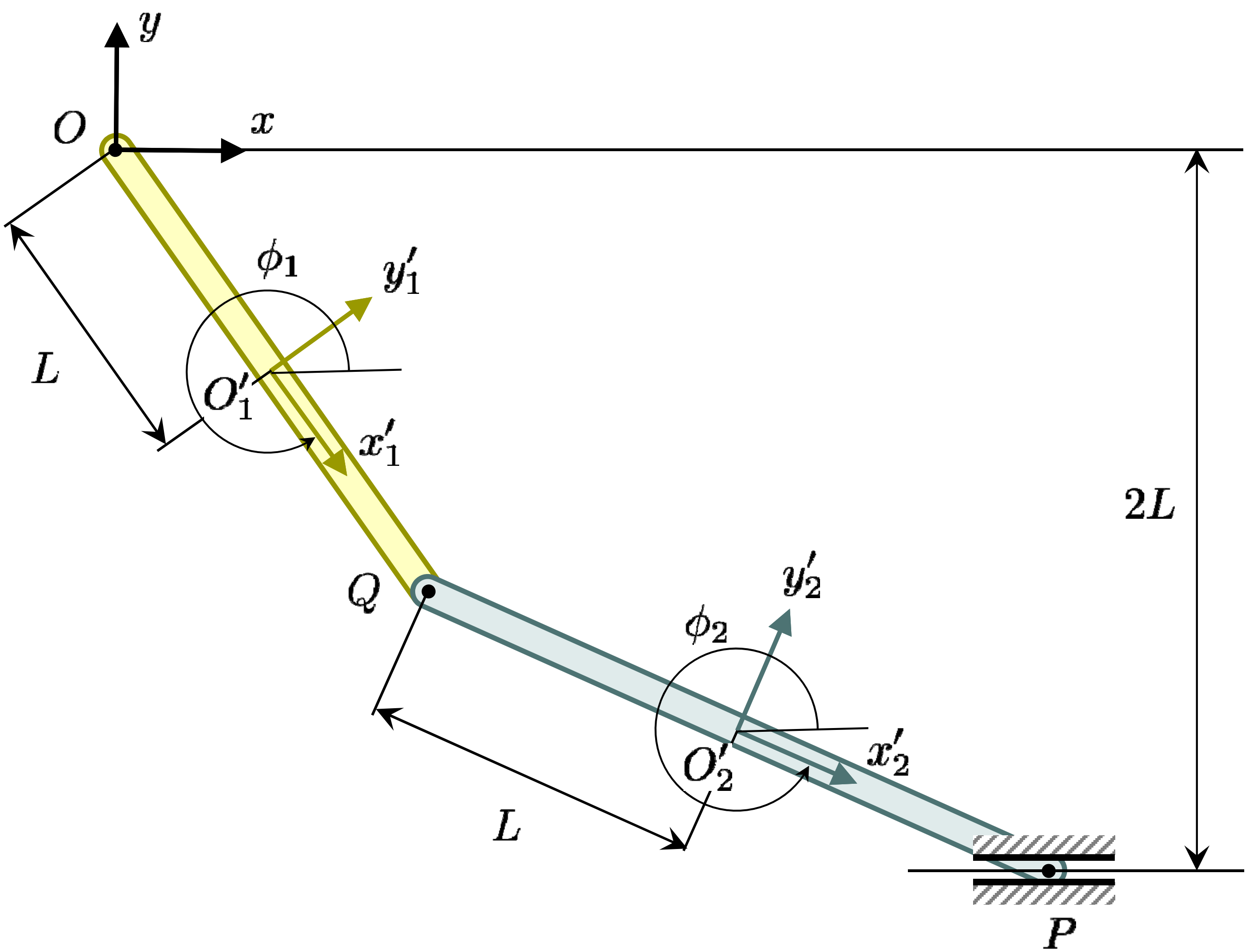


Use the array **q** of generalized coordinates to locate the point P in the GRF (Global Reference Frame)

$$\mathbf{q} = \begin{bmatrix} \theta_1 \\ \theta_{12} \end{bmatrix}$$

[handout]

Example (Absolute GC)



Use the array **q** of generalized coordinates to locate the point P in the GRF (Global Reference Frame)

$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ \phi_1 \\ x_2 \\ y_2 \\ \phi_2 \end{bmatrix}$$

Relative vs. Absolute GCs [1/3]

- A consequential question:
  - When is it easier to come up with position of point P?
    - When using Relative GCs or Absolute GCs?

Relative vs. Absolute GCs [2/3]

- Relative GCs:
  - Angle θ_1 uniquely specified both position and orientation of body 1
  - Angle θ_{12} uniquely specified the position and orientation of body 2 with respect to body 1
    - To locate point P on body 2 w.r.t. the GRF, we need to first position body 1 w.r.t. the GRF (based on θ_1), then position body 2 w.r.t. to body 1 (based on θ_{12})
    - Note that if there were 100 bodies, I would have to position body 1 w.r.t. to GRF, then body 2 w.r.t. body 1, then body 3 w.r.t. body 2, and so on, until we can position body 100 w.r.t. body 99

Relative vs. Absolute GCs [3/3]

- Second Approach (Example AGC) – relies on absolute (Cartesian) generalized coordinates:
  - x_1, y_1, θ_1 define the position and orientation of body 1 w.r.t. the GRF
  - x_2, y_2, θ_2 define the position and orientation of body 2 w.r.t. the GRF
    - To express the location of P is then straightforward and uses only x_2, y_2, θ_2 and local information (local position of B in body 2): in other words, use only information associated with body 2.
  - For AGC, you handle many generalized coordinates
    - 3 for each body in the system (six for this example)

Relative vs. Absolute GCs

And [in ME451] the winner is …

- Absolute GC formulation:
  - Straightforward to express the position of a point on a given body (and only involves the GCs corresponding to the appropriate body and the position of the point in the LRF)…
  - …but requires many GCs (and therefore many equations)
  - Common in multibody dynamics (major advantage: easy to remove/add bodies and/or constraints)

- Relative GC formulation:
  - Requires a minimal set of GCs
  - …but expressing the position of a point on a given body is complicated (and involves GCs associated with an entire chain of bodies)
  - Common in robotics, molecular dynamics, real-time applications

- **We will use AGC** - the math and therefore the software is simpler
  - Let the computer keep track of the multitude of GCs…