ME451
Kinematics and Dynamics of Machine Systems

Elements of 2D Kinematics

September 23, 2014

Dan Negrut
ME451, Fall 2014
University of Wisconsin-Madison

- Quote of the day: "Success is stumbling from failure to failure with no loss of enthusiasm." –Winston Churchill

Before we get started…

- Last time
  - Wrapped up MATLAB overview
  - Understanding how to compute the velocity and acceleration of a point P
  - Relative vs. Absolute Generalized Coordinates
- Today
  - What it means to carry out Kinematics analysis of 2D mechanisms
  - Position, Velocity, and Acceleration Analysis
- HW:
  - Haug’s book: 2.5.11, 2.5.12, 2.6.1, 3.1.1, 3.1.2
  - MATLAB (emailed to you on Th)
    - Due on Th, 9/25, at 9:30 am
    - Post questions on the forum
    - Drop MATLAB assignment in learn@uw dropbox

What comes next…

- Planar Cartesian Kinematics (Chapter 3)
  - Kinematics modeling: deriving the equations that describe motion of a mechanism, independent of the forces that produce the motion.
  - We will be using an Absolute (Cartesian) Coordinates formulation
  - Goals:
    - Develop a general library of constraints (the mathematical equations that model a certain physical constraint or joint)
    - Pose the Position, Velocity and Acceleration analysis problems

- Numerical Methods in Kinematics (Chapter 4)
  - Kinematics simulation: solving the equations that govern position, velocity and acceleration analysis

Nomenclature

- Modeling
  - Starting with a physical systems and abstracting the physics of interest to a set of equations
- Simulation
  - Given a set of equations, solve them to understand how the physical system moves in time
- NOTE: People sometimes simply use “simulation” and include the modeling part under the same term

Example 2.4.3
Slider Crank

- Purpose: revisit several concepts – Cartesian coordinates, modeling, simulation, etc.
- Based on information provided in figure (b), derive the position vector associated with point P (that is, find position of point P in the global reference frame OXY)

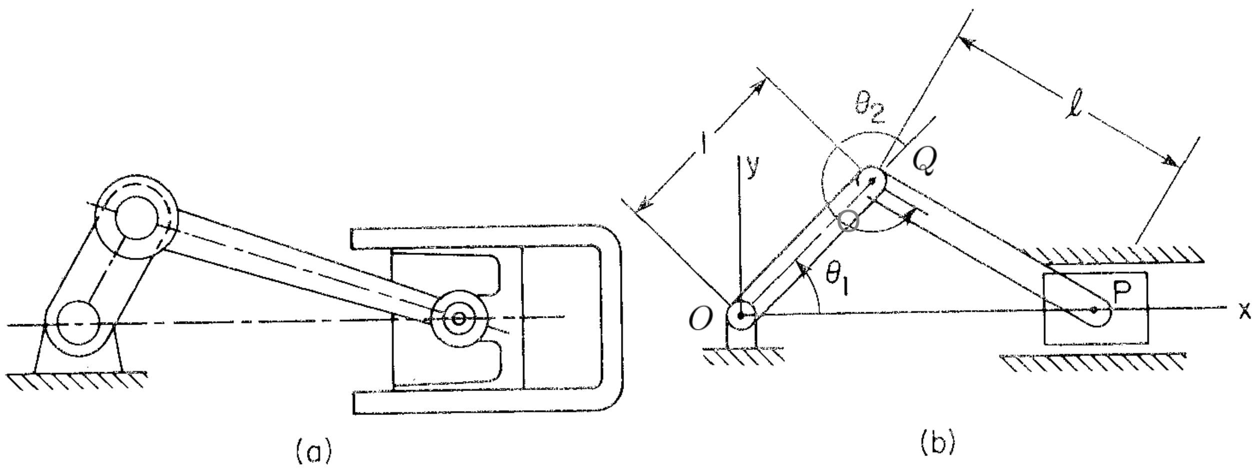


Figure 2.4.6 Slider–crank mechanism. (a) Physical system. (b) Kinematic model.

3.1

BASIC CONCEPTS IN PLANAR KINEMATICS

What is Kinematics?

- Study of the position, velocity, and acceleration of a system of interconnected bodies that make up a mechanism, independent of the forces that produce the motion
- κινημα (kinema): movement, motion

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: we only focus on Kinematic Analysis
- It is also an essential ingredient when formulating the Kinetic problem (so called Dynamics Analysis, discussed in Chapter 6)
- In general, people are more interested in the Dynamic Analysis rather than in the Kinematic Analysis

First Things First

- Before getting lost in the details of Kinematics Analysis:
  - Present a collection of terms that will help understand the “language” of Kinematics
  - Give a high-level overview of things to come and justify the need for the material presented over the next few lectures
- Among the concepts introduced today, here are the more important ones:
  - Constraint equations (as a means to defining the geometry associated with the motion of a mechanism)
  - Jacobian matrix (or simply, the Jacobian)

Nomenclature

- Rigid body
- Body-fixed Reference Frame (also called Local Reference Frame, LRF)
- Generalized coordinates $\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{nc} \end{bmatrix} \in \mathbb{R}^{nc}$
- Cartesian generalized coordinates $\mathbf{q}_i = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix}_i \equiv \begin{bmatrix} x_i \\ y_i \\ \phi_i \end{bmatrix} \in \mathbb{R}^3$

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

$$nc = 3 \cdot nb$$

Constraints (1)

- What are they, and what role do they play?
  - A collection of equations which, if satisfied, coerce the bodies in the model to move like the bodies of the mechanism
- Most important thing in relation to constraints:
  - For each joint in the model, the constraint equations you use must imply the relative motion allowed by the joint
  - Keep in mind: the way you **model** should resemble and imply the **physical system**
- Taxonomy of constraints:
  - Holonomic vs. Nonholonomic

    ολοζ (holos): all, the whole, entire
  - Scleronomic (aka Kinematic) vs. Rheonomic (aka Driving)

    σκλεροζ (skleros): hard

    ρεοσ (rheos): stream, flow, current

Constraints (2)

- Holonomic constraints are expressed as algebraic equations in terms of the generalized coordinates
  - Most constraints fall under this category
  - All constraints we'll deal with in ME451

- Nonholonomic constraints: are more general constraints and involve generalized velocities
  - Example: roll without slip constraint

- Scleronomic (kinematic) constraints are time-independent constraints (meaning they do not depend explicitly on time)

$$\mathbf{\Phi}^K(\mathbf{q})$$

- Rheonomic (driver) constraints depend explicitly on time; they are typically used to prescribe motion

$$\mathbf{\Phi}^D(\mathbf{q}, t)$$

Degrees of Freedom

- The number of degrees of freedom of a mechanism is qualitatively related to the difference between the number of generalized coordinates and the number of constraints that these coordinates must satisfy
  - Kinematic Degrees of Freedom (KDOF): the difference between the number of generalized coordinates and the number of Kinematic (Scleronomic) constraints
    - It is an attribute of the model, and it is independent of generalized coordinates used to represent the time evolution of the mechanism
  - Net Degrees of Freedom (NDOF): the difference between the number of generalized coordinates and the total number of constraints, be them Kinematic (Scleronomic) or Driving (Rheonomic)
    - Depends on how many motions you decide to specify for the parts of the mechanism
- IMPORTANT OBSERVATION: For carrying out Kinematic Analysis, a number of KDOF motions should be specified so that in the end we have NDOF=0

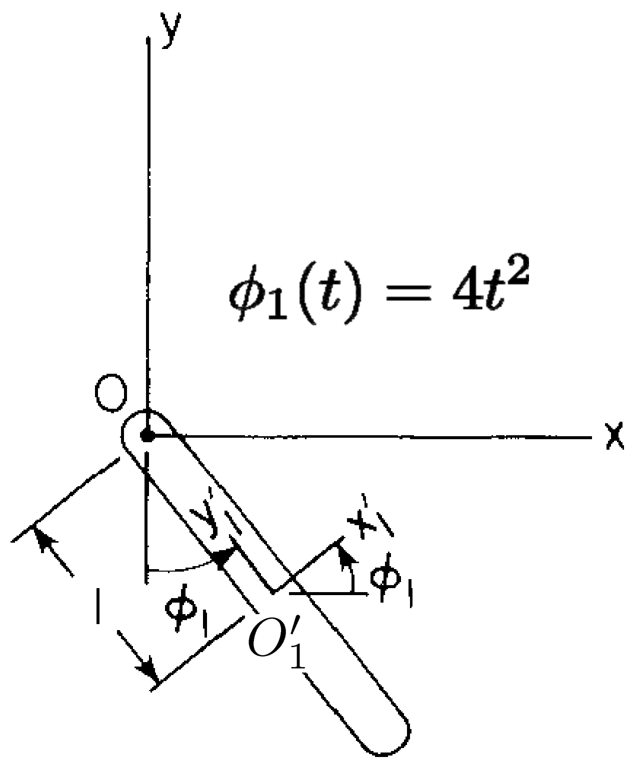
Motion: Causes

- How can one set a mechanical system in motion?
  - For a system with KDOF degrees of freedom, specify KDOF additional driving constraints (one per degree of freedom) that uniquely determine **q**(t) as the solution of an algebraic problem (Kinematic Analysis)
  - Specify/Apply a set of forces acting upon the mechanism, in which case **q**(t) is found as the solution of a differential problem (Dynamic Analysis)

Ignore this for now…

Example 3.1.1

- A pin (revolute) joint present at point O
- A motion is applied to the pendulum
- Specify the set of constraints associated with this model
  - Use Cartesian coordinates
- Write down the Kinematic and Driving constraints
  - Specify the value of KDOF and NDOF
- Formulate the velocity analysis problem
- Formulate the acceleration analysis problem



Kinematics: The Big Picture

- Imagine that just like on the previous slide, you have a motion applied to body 1 in a system with say seven bodies (1 through 7)
- Motion of body 1 dictates motion of all other bodies in the system
- Example:
  - Think of a crank and how you can move all the parts in an engine by turning the crank to start the engine
  - The motion of the crank dictates the motion of each part in the engine
    - We are interested in figuring out how each part moves because of the crank

Kinematic Analysis Stages

- Position Analysis Stage
  - Challenging
- Velocity Analysis Stage
  - Simple
- Acceleration Analysis Stage
  - OK

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

Once you have the constraints…

- Each of the three stages of Kinematics Analysis: position analysis, velocity analysis, and acceleration analysis, follow **very** similar recipes for finding the position, velocity and acceleration, respectively, of every body in the system.

- All stages rely crucially on the Jacobian matrix $\mathbf{\Phi_q}$
  - $\mathbf{\Phi_q}$ – the partial derivative of the constraints w.r.t. the generalized coordinates

- All stages require the solution of linear systems of equations of the form:

$$\mathbf{\Phi_q x = b}$$

- What is different between the three stages is the expression for the RHS **b**.

The Details…

- It all boils down to this:
  - Step 1: Write down the constraint equations associated with the model
  - Step 2: For each stage, construct Φ_q and the specific **b**, then solve for **x**

- Getting the position configuration of the mechanism
  - Kinematic Analysis key observation: The number of constraints (kinematic and driving) should be equal to the number of generalized coordinates

    In other words, NDOF=0 is a prerequisite for Kinematic Analysis

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

$$\mathbf{\Phi} : \mathbb{R}^{nc} \times \mathbb{R} \rightarrow \mathbb{R}^{nc}$$

IMPORTANT:

This is a nonlinear systems with:

nc equations

and

nc unknowns

that must be solved for **q**

Example, Position Analysis
[watered down]

- Here's a possible scenario

$$\mathbf{\Phi}(\mathbf{q},t) = \begin{bmatrix} x(t)+y(t) & - & 2 \\ x(t)-y(t) & - & \sin(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Going back to the previous slide in this example we have, $\mathbf{q} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$, $nc = 2$, $\mathbf{\Phi}^K(\mathbf{q}) = x(t) + y(t) - 2$, and $\mathbf{\Phi}^D(\mathbf{q},t) = x(t) - y(t) - \sin(t)$
- In Kinematics, you are in the business of finding $x(t)$ and $y(t)$

Velocity and Acceleration Analysis

- Position analysis: The generalized coordinates (positions) are solution of the nonlinear system:

$$\mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0}$$

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

$$\dot{\mathbf{\Phi}} \equiv \frac{d}{dt}\mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0} \quad \Rightarrow \quad \mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} + \mathbf{\Phi}_t = \mathbf{0}$$

$$\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = \underbrace{-\mathbf{\Phi}_t}_{\boldsymbol{\nu} \triangleq -\mathbf{\Phi}_t}$$

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

$$\ddot{\mathbf{\Phi}} \equiv \frac{d}{dt}\dot{\mathbf{\Phi}}(\mathbf{q}, t) = \mathbf{0} \quad \Rightarrow \quad \mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} + \left(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}\right)_{\mathbf{q}}\dot{\mathbf{q}} + 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} + \mathbf{\Phi}_{tt} = \mathbf{0}$$

$$\mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = \underbrace{-\left(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}\right)_{\mathbf{q}}\dot{\mathbf{q}} - 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} - \mathbf{\Phi}_{tt}}_{\boldsymbol{\gamma} \triangleq -\left(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}\right)_{\mathbf{q}}\dot{\mathbf{q}} - 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} - \mathbf{\Phi}_{tt}}$$

Producing the RHS of the Acceleration Equation

- The RHS of the acceleration equation was shown to be:

$$\mathbf{\Phi_q \ddot{q}} = \gamma \quad \text{where} \quad \gamma \triangleq -\left(\mathbf{\Phi_q \dot{q}}\right)_\mathbf{q} \mathbf{\dot{q}} - 2\mathbf{\Phi}_{\mathbf{q}_t}\mathbf{\dot{q}} - \mathbf{\Phi}_{tt}$$

- The terms in γ are pretty tedious to calculate by hand.
- Note that the RHS contains (is made up of) everything that does not depend on the generalized accelerations
- Implication:
  - When doing small examples in class, don't bother to compute the RHS using expression above
    - You will do this in **`simEngine2D`**, where you aim for a uniform approach to all problems
  - Simply take two time derivatives of the (simple) constraints and move everything that does not depend on acceleration to the RHS