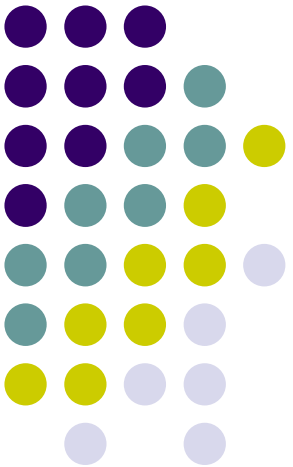
ME451
Kinematics and Dynamics of Machine Systems

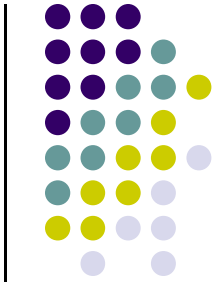


Dynamics of Planar Systems

December 9, 2010

Solving Index 3 DAEs using Newmark Method

Before we get started…



- Last Time:
  - Finish the discussion started last time
    - Marked the end of the Dynamics Analysis of Mechanical Systems chapter

- Today
  - Discuss the final exam
  - Inverse Dynamics Analysis
  - Equilibrium Analysis
  - We hug each other and stuff

- Final Exam
  - Tu, Dec. 21 at 5:05 PM, Room: **ME1245**

Done <u>once</u>, at the very beginning of the dynamics simulation

Find initial conditions for general coordinate positions and velocities that satisfy position and velocity constraint equations

Use Newton's equations of motion along with the acceleration constraint equations to find general coordinate accelerations and lambdas at time = 0.

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^T \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^A \\ \gamma \end{bmatrix}$$

At each time step (solving for the state of the system at t_{n+1}…)

Increment $t_{n+1} = t_n + h$
Define initial guess for accelerations q_dotdot and lambdas (take values from previous time step t_n)

Iterate to solve nonlinear system obtained after Newmark discretization of DAEs.

Update t_{n+1} position and velocity using Newmark's formulas and the most recent values for acceleration and lambdas.

$$\begin{cases} \mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\left[(1-2\beta)\,\ddot{\mathbf{q}}_n + 2\beta\ddot{\mathbf{q}}_{n+1}\right] \\ \dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h\left[(1-\gamma)\,\ddot{\mathbf{q}}_n + \gamma\ddot{\mathbf{q}}_{n+1}\right] \end{cases}$$

Get Jacobian of discretized equation. Use most recent q, q_dot, q_dotdot, and lambda at n+1

$$\tilde{\mathbf{J}} \equiv \begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^T \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}$$

Use most recent values at n+1 to compute what each equation equals. Each equation will probably not equal zero. These values are called the residuals.

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^T(\mathbf{q})\lambda - \mathbf{Q}^A(\dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{0} \\ \frac{1}{\beta h^2}\mathbf{\Phi}\,(\mathbf{q}, t) = \mathbf{0} \end{cases}$$

Yes. Need to refine current acc/Lagr. Multiplier values

Compute the correction vector.

$$\begin{bmatrix} \delta\ddot{\mathbf{q}} \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^T \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}^{-1} \cdot \mathbf{\Psi}(\ddot{\mathbf{q}}^{(old)}, \lambda^{old})$$

residual

Correct the most recent acceleration and lambda to get better approximations for accelerations and lambdas.

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix}^{(new)} = \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix}^{(old)} - \begin{bmatrix} \delta\ddot{\mathbf{q}} \\ \delta\lambda \end{bmatrix}$$

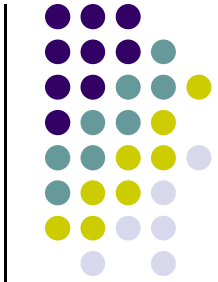
Compute the largest value present in the correction factor matrix and compare to a specified limit. This is the convergence check (largest correction is easily computed as infinity norm in MATLAB)

Is $||\delta\ddot{\mathbf{q}}|| > 10^{-3}$?
(You might have to play with the value 10^{-3} a bit. 10^{-3} is just a good rule of thumb.)

No.

Store acceleration and lambda for time t_{n+1}. Use this accelerations to compute the position and velocity at t_{n+1} and store these quantities as well.

Final Exam Info



- Tuesday, December 21, 5:05 PM. Room: **ME1245**

- For simEngine2D, please make sure you support all the modeling elements that you were assigned to handle in the MATLAB assignment 3:

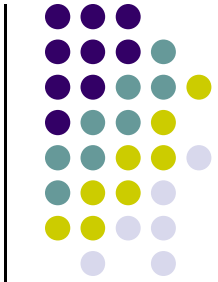
http://sbel.wisc.edu/Courses/ME451/2010/Documents/MATLAB/matlabAssignment03.pdf

- In terms of forces/torques, please make sure you support the ones you were assigned to handle in the MATLAB assignment 5:

http://sbel.wisc.edu/Courses/ME451/2010/Documents/MATLAB/matlabAssignment05.pdf

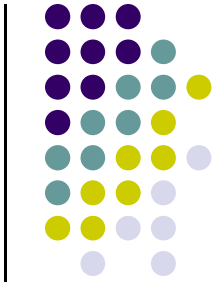
- Note: only the last 15 points (out of 100) of the final exam are tied to simEngine2D. In other words, if you don't even have a simEngine2D and answer all the other questions right you'll end up with a score of 85.

Final Exam, Rules of Engagement



- Define an acf and adm pair of files associated with the type of analysis that you are supposed to carry out and the model that you were given.
- Run simulations (Kinematics, Dynamics, and Equilibrium) using the simulation engine.
- Generate a set of plots that show the time evolution of an attribute of the model (the motion of a point, the value of a reaction force as a function of time, etc.)
- Answer questions that are more theoretical in nature. For instance, "How has been the Lagrange Multiplier Theorem used in deriving the equations of motion?", "Why are initial conditions important in the context of Dynamics analysis?", etc.
- Email the TA and class instructor a zipped directory that contains your code, adm/acf files, png plots of your results, and the answer to the theoretical questions. The naming convention for this directory should be "LastNameME451.zip". For instance, "NegrutME451.zip". The answers to the theoretical questions should be typed in MS-Word in a file called FinalExam.doc[x].

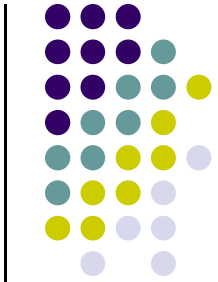
Final Exam, Comments



- If your simEngine2D does not use MATLAB, it will be your responsibility to have the compiler support allowing you to run the software during the Final Exam.
- Running your simEngine2D code should also report the amount of time it took for completing a simulation. This information should also be included in the email to the TA & instructor.
- I will not insist on having simEngine2D that you use during the exam be implemented exclusively by you. However, in good faith, you will have to indicate in the email that you will be sending to the TA & the instructor the percentage of your contribution to the simEngine2D code that you are using for the exam. I will then understand that the remaining percent came from code written by other ME451 colleague[s]. This is fine, but should be acknowledged.
- If you contributed more than 66% to your simEngine2D, you qualify for entering the race for the fastest solver. Winning that race translates into an automatic A-grade in the course.
- One other automatic A grade *might* be assigned for the most general, flexible, and neatly organized simEngine2D code.

[New Topic]

Inverse Dynamics: The idea

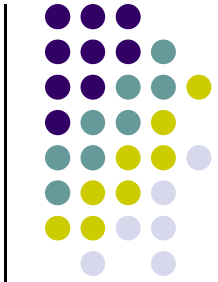


- First of all, what does dynamics analysis mean?
  - You apply some forces/torques on a mechanical system and look at how the configuration of the mechanism changes in time
  - How it moves also depends on the ICs associated with that mechanical system

- In *inverse* dynamics, the situation is quite the opposite:
  - You specify a motion of the mechanical system and you are interested in finding out the set of forces/torques that were actually applied to the mechanical system to lead to this motion

- When is *inverse* dynamics useful?
  - It's useful in controls. For instance in controlling the motion of a robot: you know how you want this robot to move, but you need to figure out what joint torques you should apply to make it move the way it should

Inverse Dynamics: The Math



- When can one talk about Inverse Dynamics?
  - Given a mechanical system, a prerequisite for Inverse Dynamics is that the number of degrees of freedom associated with the system is **zero**
    - You have as many generalized coordinates as constraints (THIS IS KEY)
  - This effectively makes the problem a Kinematics problem

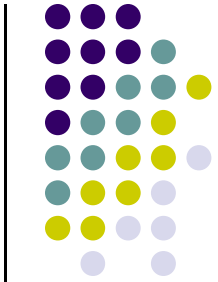
  - The two stages of the Inverse Dynamics analysis
    - First solve for accelerations (recall the acceleration equation):

$$\mathbf{\Phi_q}\ddot{\mathbf{q}} = \gamma$$

    - Next you solve for the reaction forces:

$$\mathbf{\Phi_q}^T\lambda = \mathbf{Q}^A - \mathbf{M}\ddot{\mathbf{q}}$$

Inverse Dynamics: Closure



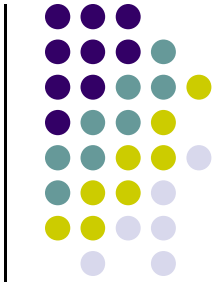
- Are we done once we computed the reaction forces?
  - Yes, because among the forces you computed, you get all the forces/torques that are necessary to impose the driving constraints Φ^D that you imposed on the system

$$\mathbf{F}^D = -\left[\mathbf{\Phi}^D_{\mathbf{r}_i}\right]^T \lambda$$

$$T^D = [(\mathbf{s'}_i^P)^T \mathbf{B}_i^T \left[\mathbf{\Phi}^D_{\mathbf{r}_i}\right]^T - \left[\mathbf{\Phi}^D_{\phi_i}\right]^T]\lambda$$

Here constraint Φ^D acts between body i and some other body. Reaction forces are computed as "felt" by body i

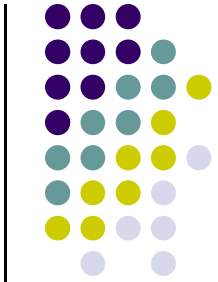
  - This gives you the forces/torques that you need to apply to get the prescribed motion



End Inverse Dynamics
Beginning Equilibrium Analysis

[New Topic]

Equilibrium Analysis: The Idea



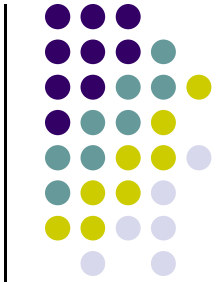
- A mechanical system is in equilibrium if the following conditions hold:

$$\dot{\mathbf{q}} = \mathbf{0} \quad \& \quad \ddot{\mathbf{q}} = \mathbf{0}$$

- Equivalently, the system is at rest, with zero acceleration

- So what does it take to be in this state of equilibrium?
  - You need to be in a certain configuration **q**
  - The reaction forces, that is, Lagrange Multipliers, should assume certain values
  - What does “certain” mean?

Equilibrium Analysis: The Math



- Equations of Motion:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{T}\lambda = \mathbf{Q}^{A} \quad \Rightarrow \quad \mathbf{\Phi}_{\mathbf{q}}^{T}\lambda = \mathbf{Q}^{A}$$

- Position Constraint Equations:

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

- Velocity Constraint Equations:

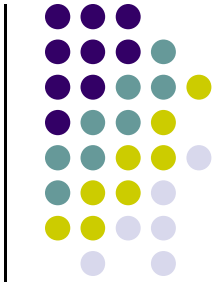
$$\mathbf{\Phi}_{\mathbf{q}} \cdot \dot{\mathbf{q}} = -\mathbf{\Phi}_{t} \quad \Rightarrow \quad \mathbf{0} = -\mathbf{\Phi}_{t}$$

- Acceleration Constraint Equations:

$$\mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = -(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} - \mathbf{\Phi}_{tt} \quad \Rightarrow \quad \mathbf{0} = -\mathbf{\Phi}_{tt}$$

[Cntd.]

Equilibrium Analysis: The Math



- To conclude, one needs a configuration **q** and the Lagrange multipliers λ should be such that

$$\begin{aligned} \mathbf{\Phi}_{\mathbf{q}}^{T}\lambda &= \mathbf{Q}^{A} \\ \mathbf{\Phi}(\mathbf{q},t) &= \mathbf{0} \\ \mathbf{\Phi}_{t}(\mathbf{q},t) &= \mathbf{0} \end{aligned}$$

- How can you go about finding such a configuration?
  - Approach 1 (dumb, but powerful)
    - Add damping in a system and watch it move till it stops
  - Approach 2 (OK, but you need a good starting point)
    - Simply solve the nonlinear system to find **q** and $\mathbf{Q}^A$

$$\begin{aligned} \mathbf{\Phi}_{\mathbf{q}}^{T}\lambda &= \mathbf{Q}^{A} \\ \mathbf{\Phi}(\mathbf{q}) &= \mathbf{0} \end{aligned}$$

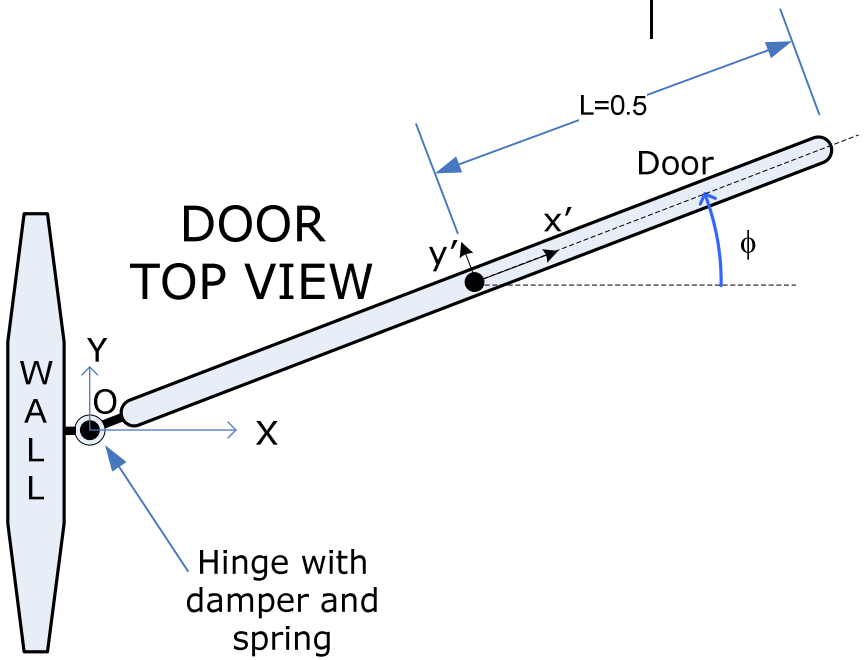
  - Approach 3 (not that common)
    - Cast it as an optimization problem
    - Works for conservative systems only

[AO]

Example: Inverse Dynamics

- Door Mass m = 30
- Mass Moment of Inertia J' = 2.5
- Spring/damping coefficients:

  K = 8 C = 1
- All units are SI.



- Zero Tension Angle of the spring:

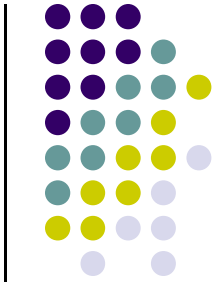
$$\phi_{free} = 0$$

- Compute torque that electrical motor applies to *open* handicapped door
  - Apply motion for two seconds to open the door like

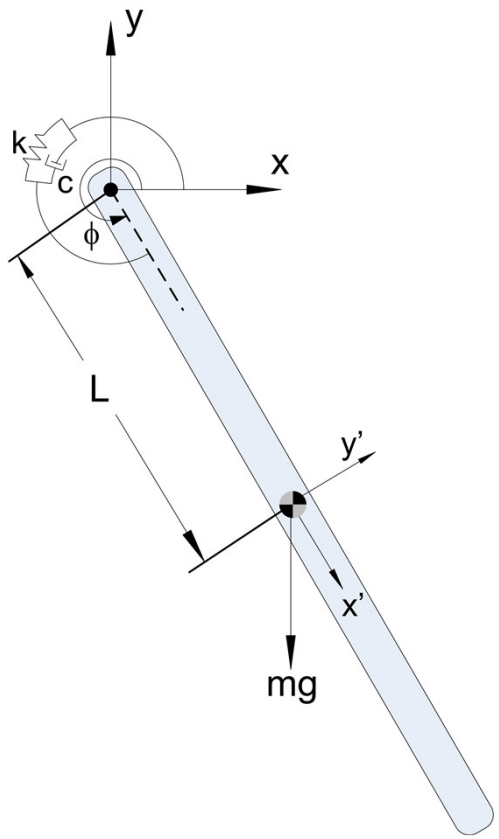
$$\Phi^D(t) = \phi - \frac{\pi}{2}\sin(\frac{\pi}{4}t)$$

[AO]

Example: Equilibrium Analysis



- Find the equilibrium configuration of the pendulum below
  - Pendulum connected to ground through a revolute joint and rotational spring-damper element



- Free angle of the spring:

$$\phi_{free} = 0$$

- Spring constant: k=25
- Mass m = 10
- Length L=1
- All units are SI.