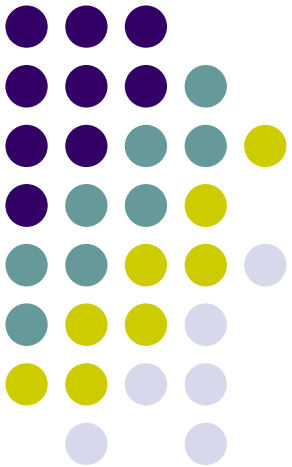
ME751
Advanced Computational Multibody Dynamics

Simulation Visualization “How To”

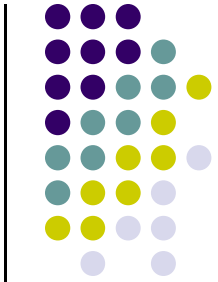
Newton-Euler Form of the EOM

March 11, 2010



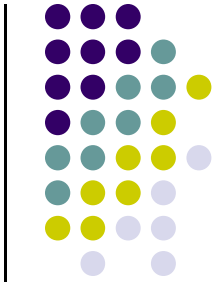
"I am so clever that sometimes I don't understand a single word of what I am saying.”
Oscar Wilde

Before we get started…



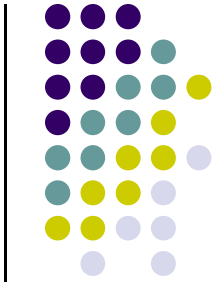
- Last Time:
  - Learn how to obtain EOM for a 3D system: the **r**-ω formulation
- Today:
  - Learn how to visualize the motion of your multibody system (Hammad)
  - Obtain the EOM: Handling the Reaction Forces.
- Visualization discussion topic started on Forum
  - http://sbel.wisc.edu/Forum/index.php?topic=62.0
- New assignment posted online
  - HW 8 due on March 18
- Final Project:
  - I'll provide feedback within one week
  - Topics and your one pager will appear on the class webpage

Simulation Visualization



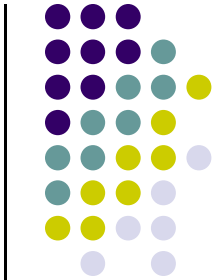
Hammad Mazhar

Goals



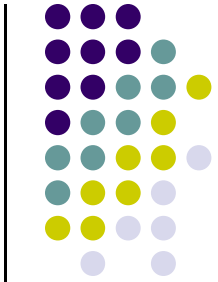
- A nice way to view a simulation
- Provide a means to look at data from different angles
- Post process visuals

Workflow



- Generate Data [Simulation Engine]
- Create Geometry [Solidworks, Maya, Blender…]
- Convert Geometry [Ogre mesh converter, Ogre XMLConverter]
- Create Input File
- View Simulation
- Render Simulation [Virtual Dub, Divx]

Step 1: Data Collection

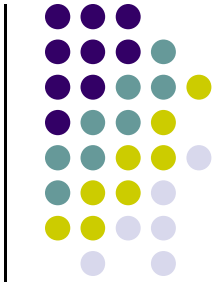


- Store simulation data to file
  - “data” means the collection of values (numbers) assumed by the generalized coordinates at each time step
  - “data” to be generated as an outcome of Kinematics or Dynamics analysis

- Need to meet minimum requirements
  - Positions of all objects
  - Euler Parameters of all objects
    - (Prevents Gimbal lock)

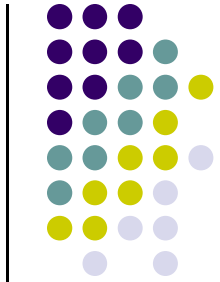
- One file per frame: pos000.dat, pos001.dat, etc
- One object per line

Example: pos23.dat



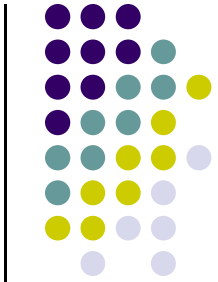
```
1, 20, -0.25, 0, 0, 0, 0, 0,                    [#, x, y, z, e0, e1, e2, e3]
2, 0, 2.5, 0, 0, 0, 0, 0,
3, 40, 2.5, 0, 0, 0, 0, 0,
4, 20, 2.5, -40, 0, 0, 0, 0,
5, 20, 2.5, 40, 0, 0, 0, 0,
6, -33.775, 25, 0, 0.965926, 0, 0, -0.258819,
7, -32.525, 27.1651, 39, 0.965926, 0, 0, -0.258819,
8, -32.525, 27.1651, -39, 0.965926, 0, 0, -0.258819,
9, 33.775, 50, 0, 0.965926, 0, 0, 0.258819,
10, 32.525, 52.1651, 20, 0.965926, 0, 0, 0.258819,
11, 32.525, 52.1651, -20, 0.965926, 0, 0, 0.258819,
12, 5.08655, 1.47155, -6.50457, -0.543056, -0.465413, -0.105751, -0.690878,
13, 10.0644, 0.358719, 28.176, -0.463821, 0.413781, 0.366809, -0.692184,
14, 10.3082, 0.398135, -21.7696, -0.687234, -0.217201, -0.692418, -0.0328985,
15, -0.708364, 6.53418, -18.624, -0.310487, -0.0341572, 0.00743839, -0.949934,
16, -2.05941, 8.52682, -13.7048, -0.743404, 0.308513, -0.370145, -0.463861,
17, 27.2258, 0.316766, -23.067, 0.505163, 0.00571037, 0.858919, -0.0839137,
18, -1.00176, 7.0769, 3.75132, 0.76269, -0.0641002, 0.475516, 0.433679,
19, 17.1133, 0.706114, -13.3631, -0.556282, 0.61313, 0.260976, -0.496504,
20, -5.51988, 9.85426, -13.7024, -0.165998, -0.255, -0.951555, 0.0445564,
```

Step 2: Generate Models



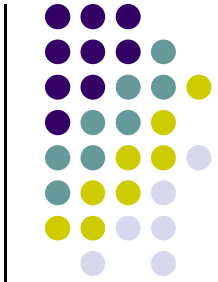
- Using Cad/3D Modeling software create geometry
- Autocad, Solidworks, Maya, 3DS Max … etc
- Any Geometry can be used
  - Not constrained to simulation geometry
  - Purely for visual effect

Step 3: Convert Models



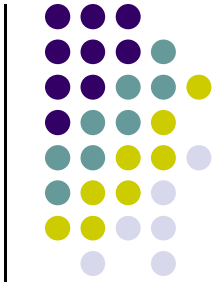
- Export model from modeling software
- Import into Blender
- Set Materials
- Export as .mesh
  - Ogre-Specific geometry file

Step 4:Create simulation input file



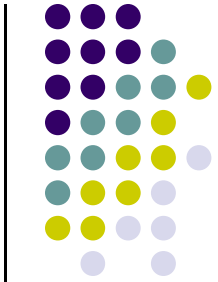
- Define:
  - Data Location
  - Number of frames
  - Start/Stop Times
  - Number of objects
  - Materials
  - Number of each type of object

Example:conf.txt



```
pos .dat
1500 1
1011 12
1      Cube.mesh 40 .5 80       Material1
1      Cube.mesh .5 5 80  Material1
1      Cube.mesh .5 5 80  Material1
1      Cube.mesh 40 5 .5  Material1
1      Cube.mesh 40 5 .5  Material1
1      Cube.mesh 78 .5 78       Material1
1      Cube.mesh 78 5 .5  Material1
1      Cube.mesh 78 5 .5  Material1
1      Cube.mesh 39 .5 39       Material1
1      Cube.mesh 39 5 .5  Material1
1      Cube.mesh 39 5 .5  Material1
1000   Cube.mesh .5 .5 .5       Material1
```

Other parameters



- Materials are in ./resources/materials.material file:

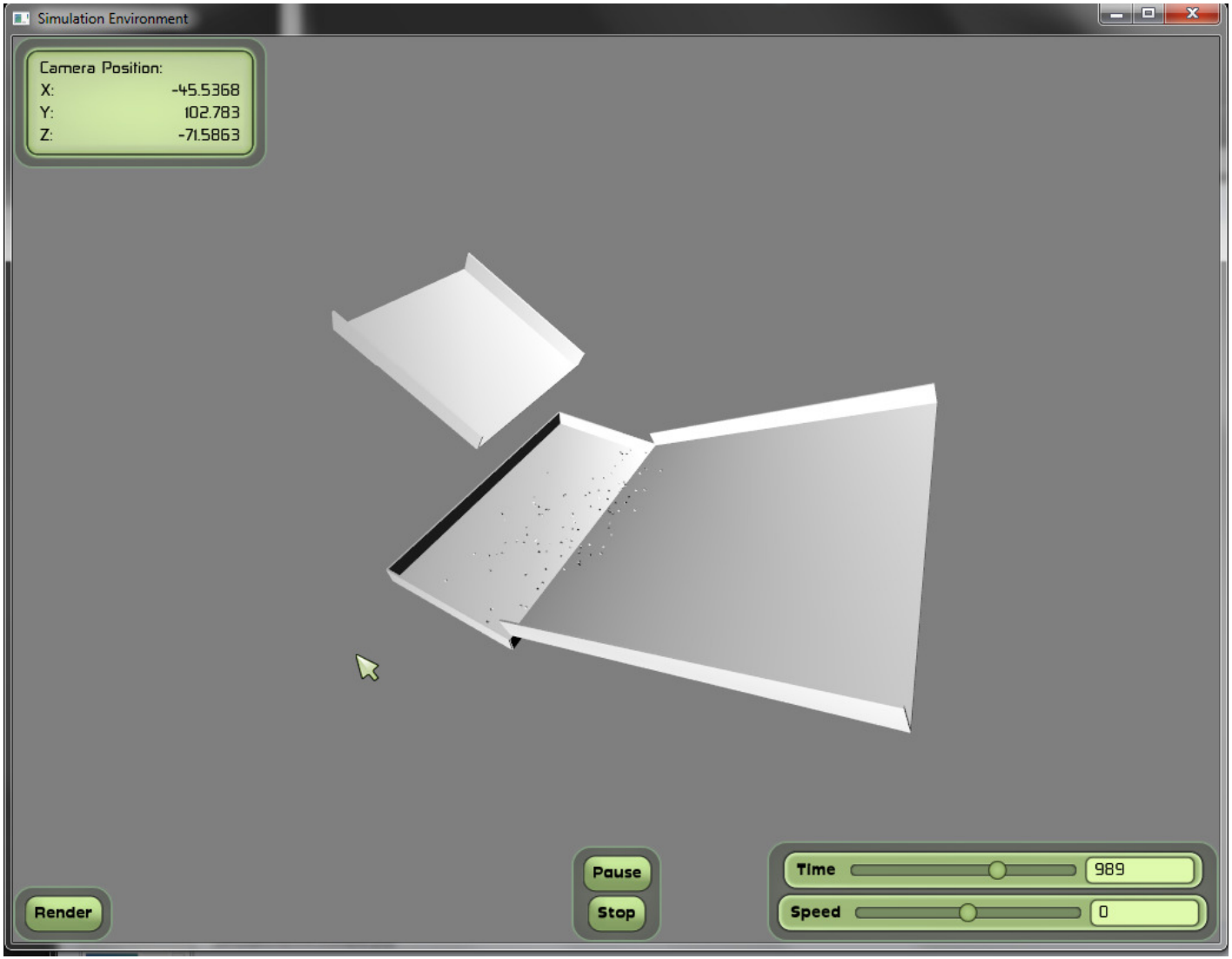
```
material Material1
{
      technique
      {
               pass
               {
                              ambient 0.000000 0.000000 0.000000 1.000000
                              diffuse 1.00000 1.00000 1.00000 1.000000
                              specular 0.500000 0.500000 0.500000 1.000000 12.500000
                              emissive 0.000000 0.000000 0.000000 1.000000
               }
      }
}
```

- File “resources.cfg” contains paths to model and data zips

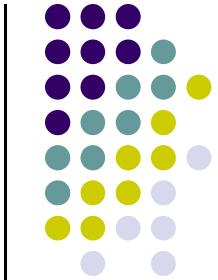
```
# Resources required
[General]
Zip=resources/SdkTrays.zip
Zip=resources/models.zip
FileSystem=./resources

[Data]
Zip=resources/data.zip
```

Viewing Simulation:



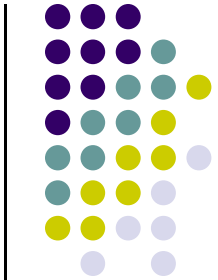
Rendering



- Click Render to store each frame to an image file
  - Saved in the ./render directory

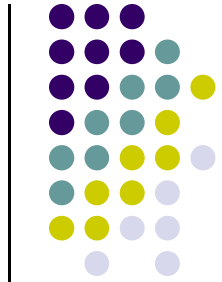
- Use Virtual Dub to join images together
  - Open up first image and the rest will be loaded
  - Select compression from video->compression
  - Save video

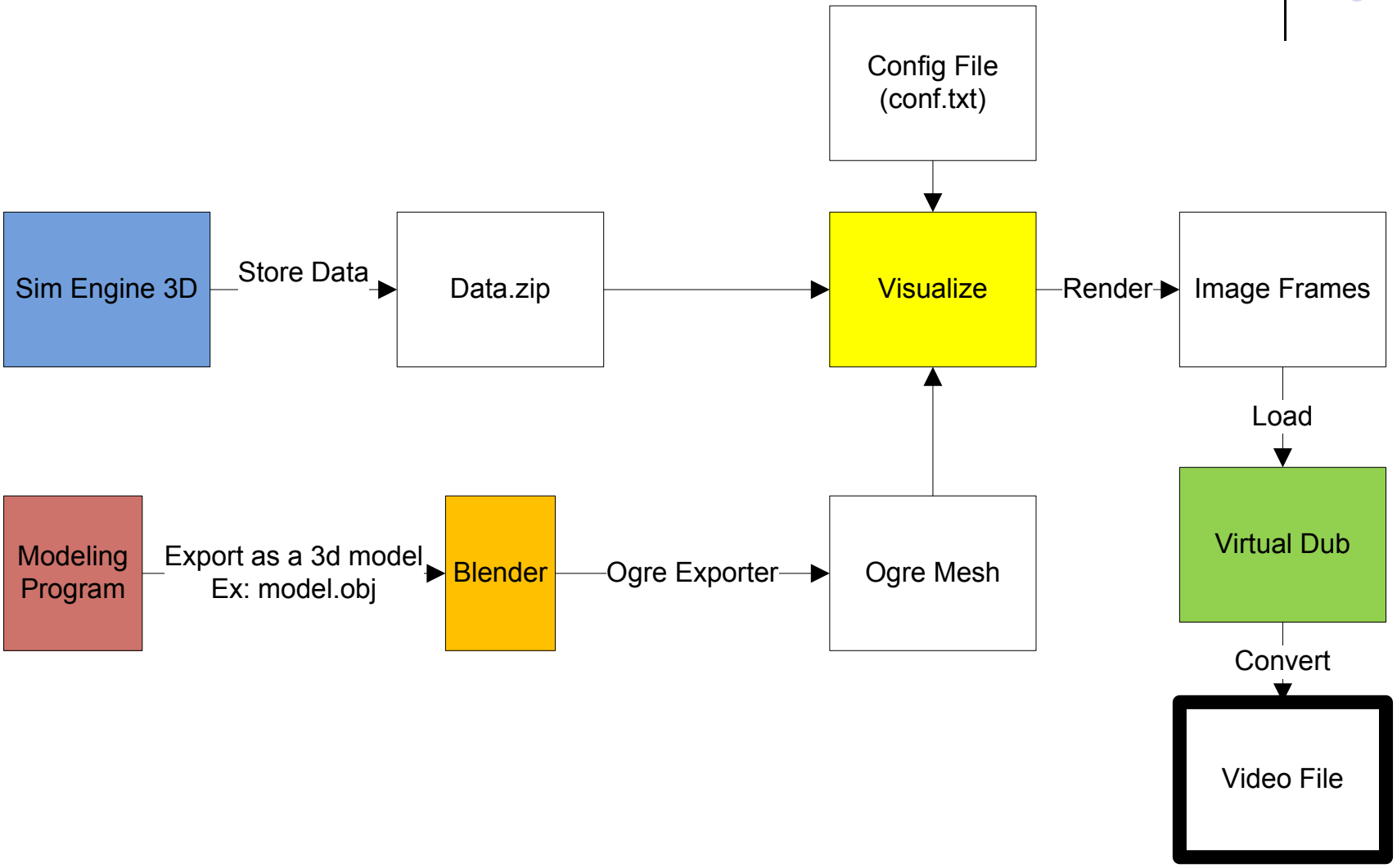
Links



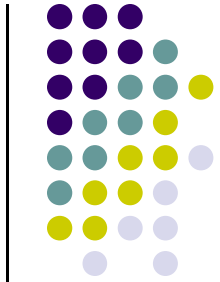
- **Blender**
  - http://www.blender.org/
- **Ogre Exporter for blender**
  - http://www.ogre3d.org/wiki/index.php/Blender_Exporter
- **Python**
  - http://www.python.org/download/
- **Ogre3d**
  - http://www.ogre3d.org/
- **VirtualDub**
  - http://virtualdub.sourceforge.net/

Visualization: The Work Flow



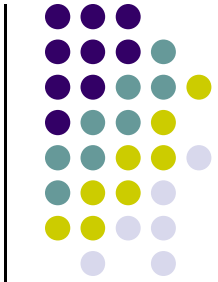


~Fin~



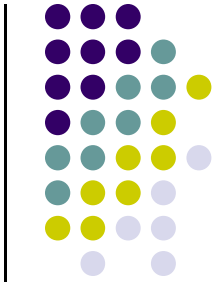
- Yes the code has bugs
- No the code is not idiot proof
- Use xvid to compress video in virtual dub
  - Faster, better quality and smaller size
  - http://www.xvid.org/Downloads.15.0.html

EOM: the $\mathbf{r} - \bar{\omega}$ Formulation



- We now return to our regularly scheduled program…

Virtual Work: Putting Things in Perspective



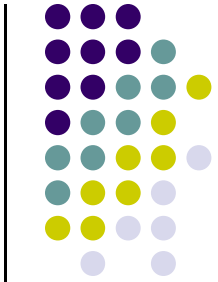
- Recall that last time we showed that the expression of the virtual work assumed the form:

$$\begin{aligned}\delta W &= \sum_{i=1}^{nb} \left[-\delta \mathbf{r}_i^T \ddot{\mathbf{r}}_i m_i - \delta \bar{\pi}_i^T \tilde{\bar{\omega}}_i \bar{\mathbf{J}}_i \bar{\omega}_i - \delta \bar{\pi}_i^T \bar{\mathbf{J}}_i \dot{\bar{\omega}}_i + \delta \mathbf{r}_i^T \cdot \mathbf{F}_i^m + \delta \bar{\pi}_i^T \cdot \bar{\mathbf{n}}_i^m \right. \\ &+ \left. \delta \mathbf{r}_i^T \mathbf{F}_i^a + \delta \bar{\pi}_i^T \bar{\mathbf{n}}_i^a + \delta \mathbf{r}_i^T \mathbf{F}_i^r + \delta \bar{\pi}_i^T \bar{\mathbf{n}}_i^r \right] = 0\end{aligned}$$

- Alternatively,

$$\begin{aligned}\delta W &= \sum_{i=1}^{nb} \left[\delta \mathbf{r}_i^T \left(-\ddot{\mathbf{r}}_i m_i + \mathbf{F}_i^m + \mathbf{F}_i^a + \mathbf{F}_i^r \right) \right. \\ &+ \left. \delta \bar{\pi}_i^T \left(-\tilde{\bar{\omega}}_i \bar{\mathbf{J}}_i \bar{\omega}_i - \bar{\mathbf{J}}_i \dot{\bar{\omega}}_i + \bar{\mathbf{n}}_i^m + \bar{\mathbf{n}}_i^a + \bar{\mathbf{n}}_i^r \right) \right] = 0 \end{aligned} \tag{1}$$

Virtual Work: Putting Things in Perspective



- Since Eq.(1) on previous slide should hold for *any* set of virtual displacements $(\delta\mathbf{r}_1, \delta\bar{\pi}_1)$, $(\delta\mathbf{r}_2, \delta\bar{\pi}_2)$,..., $(\delta\mathbf{r}_{nb}, \delta\bar{\pi}_{nb})$, then we necessarily have that for $i = 1, \ldots, nb$:

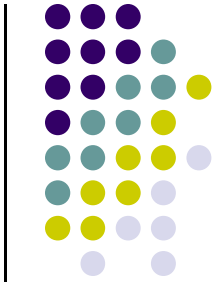
$$\begin{aligned} -\ddot{\mathbf{r}}_i m_i + \mathbf{F}_i^m + \mathbf{F}_i^a + \mathbf{F}_i^r &= \mathbf{0}_3 \\ -\tilde{\bar{\omega}}_i \bar{\mathbf{J}}_i \bar{\omega}_i - \bar{\mathbf{J}}_i \dot{\bar{\omega}}_i + \bar{\mathbf{n}}_i^m + \bar{\mathbf{n}}_i^a + \bar{\mathbf{n}}_i^r &= \mathbf{0}_3 \end{aligned}$$

- Equivalently,

$$\begin{aligned} m_i \ddot{\mathbf{r}}_i &= \mathbf{F}_i^m + \mathbf{F}_i^a + \mathbf{F}_i^r \\ \bar{\mathbf{J}}_i \dot{\bar{\omega}}_i &= \bar{\mathbf{n}}_i^m + \bar{\mathbf{n}}_i^a + \bar{\mathbf{n}}_i^r - \tilde{\bar{\omega}}_i \bar{\mathbf{J}}_i \bar{\omega}_i \end{aligned}$$

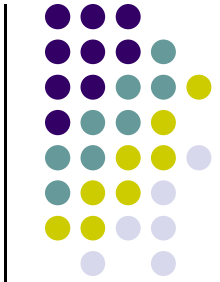
- The set of equations above represent the EOM for the system of nb bodies.

Handling the Reaction Forces



- Unfortunately, we have the EOM but in the current form we can't use them since they contain the reaction forces and torques $\mathbf{F}_i^r$ and $\bar{\mathbf{n}}_i^r$, $i = 1, \ldots, nb$, that we don't know.
  - It turns out that in order to get $\ddot{\mathbf{r}}_i$ and $\dot{\bar{\omega}}_i$, which are quantities of primary interest, we have to find a sensible way to handle the reaction torques and moments, $\mathbf{F}_i^r$ and $\bar{\mathbf{n}}_i^r$, respectively.
- The key observation, and the reason for using the principle of Virtual Work:
  - If we are careful about choosing the virtual displacements $(\delta\mathbf{r}_1, \delta\bar{\pi}_1)$, $(\delta\mathbf{r}_2, \delta\bar{\pi}_2)$,..., $(\delta\mathbf{r}_{nb}, \delta\bar{\pi}_{nb})$, we can get eliminate the contribution of the reaction forces
  - Specifically, if the virtual displacements $(\delta\mathbf{r}_1, \delta\bar{\pi}_1)$, $(\delta\mathbf{r}_2, \delta\bar{\pi}_2)$,..., $(\delta\mathbf{r}_{nb}, \delta\bar{\pi}_{nb})$, are chosen so that they are consistent with the set of constraints that produce the reaction torques and moments, $\mathbf{F}_i^r$ and $\bar{\mathbf{n}}_i^r$, respectively, then their total virtual work is zero

Handling the Reaction Forces



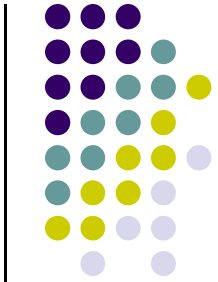
- Recall that the following condition should hold for a set of virtual displacements to be consistent (see two lectures ago):

$$\left[\quad \mathbf{\Phi_r} \qquad \bar{\mathbf{\Pi}}(\mathbf{\Phi}) \quad \right] \cdot \left[\begin{array}{c} \delta\mathbf{r} \\ \delta\bar{\pi} \end{array} \right] = \bar{\mathbf{R}}(\mathbf{\Phi}) \cdot \left[\begin{array}{c} \delta\mathbf{r} \\ \delta\bar{\pi} \end{array} \right] = \mathbf{0}$$

- In what follows we'll use the notation:

$$\left[\begin{array}{c} \delta\mathbf{r} \\ \delta\bar{\pi} \end{array} \right]_{6nb} = \left[\begin{array}{c} \delta\mathbf{r}_1 \\ \vdots \\ \delta\mathbf{r}_{nb} \\ \delta\bar{\pi}_1 \\ \vdots \\ \delta\bar{\pi}_{nb} \end{array} \right]_{6nb} \qquad \mathbf{F}^r = \left[\begin{array}{c} \mathbf{F}^r_1 \\ \vdots \\ \mathbf{F}^r_{nb} \end{array} \right]_{3nb} \qquad \bar{\mathbf{n}}^r = \left[\begin{array}{c} \bar{\mathbf{n}}^r_1 \\ \vdots \\ \bar{\mathbf{n}}^r_{nb} \end{array} \right]_{3nb}$$

Handling the Reaction Forces
[Cntd.]



- Going back to the observation that the *total* virtual work of the reaction forces is zero when the virtual displacements are consistent, we have that

$$\delta\mathbf{r}_1^T\mathbf{F}_1^r + \ldots + \delta\mathbf{r}_{nb}^T\mathbf{F}_{nb}^r + \delta\bar{\pi}_1^T\bar{\mathbf{n}}_1^r + \ldots + \delta\bar{\pi}_{nb}^T\bar{\mathbf{n}}_{nb}^r = 0$$

provided

$$\mathbf{\Phi}_{\mathbf{r}_1}\delta\mathbf{r}_1 + \ldots + \mathbf{\Phi}_{\mathbf{r}_{nb}}\delta\mathbf{r}_{nb} + \bar{\mathbf{\Pi}}_1(\mathbf{\Phi})\delta\bar{\pi}_1 + \ldots + \bar{\mathbf{\Pi}}_{nb}(\mathbf{\Phi})\delta\bar{\pi}_{nb} = \mathbf{0}_{nc}$$

- In matrix form,

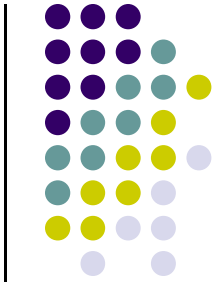
$$\delta\mathbf{r}^T\mathbf{F}^r + \delta\bar{\pi}^T\bar{\mathbf{n}}^r = 0$$

provided

$$\mathbf{\Phi}_{\mathbf{r}}\delta\mathbf{r} + \bar{\mathbf{\Pi}}(\mathbf{\Phi})\delta\bar{\pi} = \mathbf{0}_{nc}$$

- NOTE: I used nc for the number of constraints, rather than m, which is what I used in the past. This was in order to avoid confusion with the superscript m.

Handling the Reaction Forces
[Cntd.]



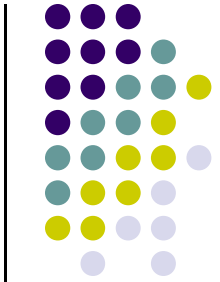
- Going back to the expression of the virtual work, we'll ignore the effect of the reaction forces/torques at the price of not having quite any arbitrary set of virtual displacements. Instead, we will have to ensure that they are consistent.
- Then,

$$\delta W = \sum_{i=1}^{nb} \left[\delta \mathbf{r}_i^T \left(-m_i \ddot{\mathbf{r}}_i + \mathbf{F}_i^m + \mathbf{F}_i^a \right) + \delta \bar{\pi}_i^T \left(-\tilde{\bar{\omega}}_i \bar{\mathbf{J}}_i \bar{\omega}_i - \bar{\mathbf{J}}_i \dot{\bar{\omega}}_i + \bar{\mathbf{n}}_i^m + \bar{\mathbf{n}}_i^a \right) \right] = 0$$

provided

$$\mathbf{\Phi}_{\mathbf{r}_1} \delta \mathbf{r}_1 + \ldots + \mathbf{\Phi}_{\mathbf{r}_{nb}} \delta \mathbf{r}_{nb} + \bar{\mathbf{\Pi}}_1(\mathbf{\Phi}) \delta \bar{\pi}_1 + \ldots + \bar{\mathbf{\Pi}}_{nb}(\mathbf{\Phi}) \delta \bar{\pi}_{nb} = \mathbf{0}_{nc}$$

Handling the Reaction Forces
[Cntd.]

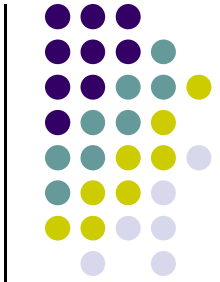


- We will now set the stage for applying the Lagrange Multiplier theorem
- We introduce the following notation ($\mathbf{I}_3$ is the identity matrix of dimension 3):

$$\mathbf{M} = \begin{bmatrix} m_1\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & m_2\mathbf{I}_3 & \dots & \mathbf{0}_{3\times 3} \\ \dots & \dots & \dots & \dots \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & m_{nb}\mathbf{I}_3 \end{bmatrix} \qquad \bar{\mathbf{J}} = \begin{bmatrix} \bar{\mathbf{J}}_1 & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \bar{\mathbf{J}}_2 & \dots & \mathbf{0}_{3\times 3} \\ \dots & \dots & \dots & \dots \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \bar{\mathbf{J}}_{nb} \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1^a + \mathbf{F}_1^m \\ \vdots \\ \mathbf{F}_{nb}^a + \mathbf{F}_{nb}^m \end{bmatrix}_{3nb} \qquad \tau = \begin{bmatrix} \bar{\mathbf{n}}_1^a + \bar{\mathbf{n}}_1^m - \tilde{\bar{\omega}}_1\bar{\mathbf{J}}_1\bar{\omega}_1 \\ \vdots \\ \bar{\mathbf{n}}_{nb}^a + \bar{\mathbf{n}}_{nb}^m - \tilde{\bar{\omega}}_{nb}\bar{\mathbf{J}}_{nb}\bar{\omega}_{nb} \end{bmatrix}_{3nb}$$

Handling the Reaction Forces
[Cntd.]



- The virtual work then assumes the expression (see two slides ago):

$$\delta W = \delta \mathbf{r}^T \left(\mathbf{M}\ddot{\mathbf{r}} - \mathbf{F} \right) + \delta \bar{\pi}^T \left(\bar{\mathbf{J}}\dot{\bar{\omega}} - \tau \right) = 0$$

provided

$$\mathbf{\Phi_r} \delta \mathbf{r} + \bar{\mathbf{\Pi}}(\mathbf{\Phi}) \delta \bar{\pi} = \mathbf{0}_{nc}$$

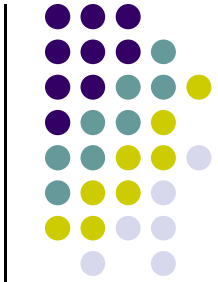
- Other way of posing this is as follows:

If the virtual displacements are consistent; i.e., $\begin{bmatrix} \mathbf{\Phi_r} & \bar{\mathbf{\Pi}}(\mathbf{\Phi}) \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{r} \\ \delta \bar{\pi} \end{bmatrix} = \mathbf{0}_{nc}$

then it follows that

$$\delta W = \begin{bmatrix} \delta \mathbf{r} \\ \delta \bar{\pi} \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{M}\ddot{\mathbf{r}} - \mathbf{F} \\ \bar{\mathbf{J}}\dot{\bar{\omega}} - \tau \end{bmatrix} = 0$$

Handling the Reaction Forces: Applying Lagrange Multiplier Theorem



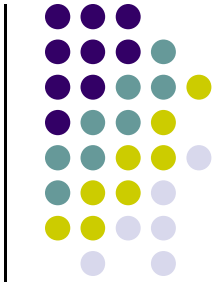
- According to the Lagrange Multiplier theorem, there exists a vector of nc Lagrange Multipliers, $\lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{nc} \end{bmatrix}$, so that

$$\begin{bmatrix} \mathbf{M}\ddot{\mathbf{r}} - \mathbf{F} \\ \bar{\mathbf{J}}\dot{\bar{\omega}} - \tau \end{bmatrix} + \begin{bmatrix} \mathbf{\Phi}_{\mathbf{r}}^T \\ \bar{\mathbf{\Pi}}^T(\mathbf{\Phi}) \end{bmatrix} \lambda = \mathbf{0}_{6nb}$$

- Expression above is the most important equation in ME751: the Newton-Euler form of the EOM. Equivalently expressed as:

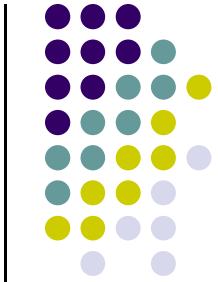
$$\begin{cases} \mathbf{M}\ddot{\mathbf{r}} + \mathbf{\Phi}_{\mathbf{r}}^T \lambda = \mathbf{F} \\ \bar{\mathbf{J}}\dot{\bar{\omega}} + \bar{\mathbf{\Pi}}^T(\mathbf{\Phi})\lambda = \tau \end{cases}$$

Comments on the Newton-Euler Form of the EOM



- Here is what we know:
  - $\mathbf{M}$ and $\bar{\mathbf{J}}$, attributes of the bodies making up the multibody system
  - $\mathbf{\Phi_r}$ and $\bar{\mathbf{\Pi}}(\mathbf{\Phi})$, two quantities that we obtain based on the constraints present in the multibody system
  - $\mathbf{F}$ and τ, force/torque quantities that we obtain based on the forces and torques acting on each body in the system
- Here is what we do not know:
  - $\ddot{\mathbf{r}}$ and $\dot{\omega}$, the translational and angular acceleration
  - λ, the set of Lagrange Multipliers associated with the constraints present in the system
- To summarize,
  - The number of equations on previous slide: $6nb$
  - The number of unknowns in those equations $3nb$ for $\ddot{\mathbf{r}}$, $3nb$ for $\dot{\omega}$, and nc for λ $\Rightarrow$ $6nb+nc$ unknowns
  - We have more unknowns than equation...

Augmenting the EOM. The Complete Picture.



- What saves the day is the fact that we can also use the acceleration kinematic constraint equations:

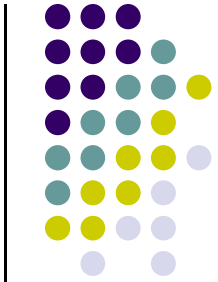
$$\mathbf{\Phi_r}\ddot{\mathbf{r}} + \bar{\mathbf{\Pi}}(\mathbf{\Phi})\dot{\bar{\omega}} = \gamma \qquad \Rightarrow \qquad nc \text{ more equations.}$$

- We can assemble everything in matrix form and produce the following *system of linear equations* of dimension $6nb + nc$, whose solution will provide the unknowns $\ddot{\mathbf{r}}$, $\dot{\bar{\omega}}$, and λ:

$$\begin{bmatrix} \mathbf{M} & \mathbf{0}_{3nb\times 3nb} & \mathbf{\Phi}_\mathbf{r}^T \\ \mathbf{0}_{3nb\times 3nb} & \bar{\mathbf{J}} & \bar{\mathbf{\Pi}}^T(\mathbf{\Phi}) \\ \mathbf{\Phi_r} & \bar{\mathbf{\Pi}}(\mathbf{\Phi}) & \mathbf{0}_{nc\times nc} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{r}} \\ \dot{\bar{\omega}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \tau \\ \gamma \end{bmatrix}_{6nb+nc}$$

- Remark: the derivation was kind of tedious, but the problem we have to solve to get the quantities of interest looks simple since it is formulated in terms of quantities that we are already familiar with

Recovering The Reaction Forces/Torques

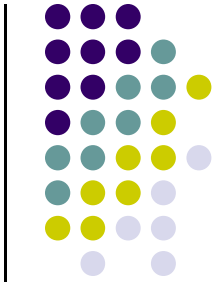


- In order to understand the role played by the Lagrange Multipliers λ, we express the Newton-Euler form of the EOM at the body level; i.e., for each body separately:

$$\left.\begin{array}{rcl} m_1\ddot{\mathbf{r}}_1 + \mathbf{\Phi}_{\mathbf{r}_1}^T\lambda &=& \mathbf{F}_1^a + \mathbf{F}_1^m \\ \cdots & & \cdots \\ m_i\ddot{\mathbf{r}}_i + \mathbf{\Phi}_{\mathbf{r}_i}^T\lambda &=& \mathbf{F}_i^a + \mathbf{F}_i^m \\ \cdots & & \cdots \\ m_{nb}\ddot{\mathbf{r}}_{nb} + \mathbf{\Phi}_{\mathbf{r}_{nb}}^T\lambda &=& \mathbf{F}_{nb}^a + \mathbf{F}_{nb}^m \end{array}\right\} \Rightarrow \quad \begin{array}{l} m_i\ddot{\mathbf{r}}_i + \mathbf{\Phi}_{\mathbf{r}_i}^T\lambda = \mathbf{F}_i^a + \mathbf{F}_i^m \\ (i = 1, \ldots, nb) \end{array}$$

$$\left.\begin{array}{rcl} \bar{\mathbf{J}}_1\dot{\bar{\omega}}_1 + \bar{\mathbf{\Pi}}_1^T(\mathbf{\Phi})\lambda &=& \bar{\mathbf{n}}_1^a + \bar{\mathbf{n}}_1^m - \tilde{\bar{\omega}}_1\bar{\mathbf{J}}_1\bar{\omega}_1 \\ \cdots & & \cdots \\ \bar{\mathbf{J}}_i\dot{\bar{\omega}}_i + \bar{\mathbf{\Pi}}_i^T(\mathbf{\Phi})\lambda &=& \bar{\mathbf{n}}_i^a + \bar{\mathbf{n}}_i^m - \tilde{\bar{\omega}}_i\bar{\mathbf{J}}_i\bar{\omega}_i \\ \cdots & & \cdots \\ \bar{\mathbf{J}}_{nb}\dot{\bar{\omega}}_{nb} + \bar{\mathbf{\Pi}}_{nb}^T(\mathbf{\Phi})\lambda &=& \bar{\mathbf{n}}_{nb}^a + \bar{\mathbf{n}}_{nb}^m - \tilde{\bar{\omega}}_{nb}\bar{\mathbf{J}}_{nb}\bar{\omega}_{nb} \end{array}\right\} \Rightarrow \begin{array}{l} \bar{\mathbf{J}}_i\dot{\bar{\omega}}_i + \bar{\mathbf{\Pi}}_i^T(\mathbf{\Phi})\lambda = \bar{\mathbf{n}}_i^a + \bar{\mathbf{n}}_i^m - \tilde{\bar{\omega}}_i\bar{\mathbf{J}}_i\bar{\omega}_i \\ (i = 1, \ldots, nb) \end{array}$$

Recovering The Reaction Forces/Torques [Cntd.]



- If we compare the expression of the EOM on previous slide with the one we obtained back when we still had $\mathbf{F}_i^r$ and $\bar{\mathbf{n}}_i^r$ around (slide 19), it is easy to see that

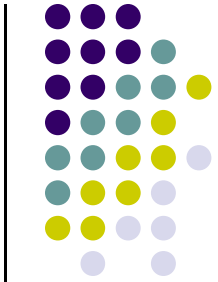
$$\mathbf{F}_i^r = -\boldsymbol{\Phi}_{\mathbf{r}_i}^T \lambda \qquad \bar{\mathbf{n}}_i^r = -\bar{\boldsymbol{\Pi}}_i^T(\boldsymbol{\Phi})\lambda$$

- In other words, the Lagrange multipliers, obtained along with $\ddot{\mathbf{r}}$ and $\dot{\bar{\omega}}$ as the solution of a linear system, are the key ingredients needed to compute the reaction forces/torques produced by the constraints $\boldsymbol{\Phi}(\mathbf{q}, t)$ present in the system

- The way the reaction forces and torques should be interpreted is like this: the quantities $-\boldsymbol{\Phi}_{\mathbf{r}_i}^T \lambda$ and $-\bar{\boldsymbol{\Pi}}_i^T(\boldsymbol{\Phi})\lambda$ represent the net reaction force applied at the center of the L-RF$_i$ and the net reaction torque applied to the rigid body, respectively, them being a consequence of the presence of a set of constraints $\boldsymbol{\Phi}(\mathbf{q}, t)$

- Moreover, one can go to a finer level of granularity and investigate the contribution of a certain constraint Φ^α (with Lagrange Multiplier λ_α) on body i. The reaction force is simply $-[\Phi_{\mathbf{r}_i}^\alpha]^T \lambda_\alpha$, while the reaction torque expressed in L-RF$_i$ is $-\bar{\boldsymbol{\Pi}}_i^T(\Phi^\alpha)\lambda_\alpha$

Comments on the Reaction Forces/Torques



- For every constraint there is a Lagrange Multiplier. Once you have the Lagrange Multiplier you can compute the reaction force and reaction torque produced by the said constraint
  - Just to reinforce this, if your mechanical system has a collection of joints that say leads to 13 GCons, you will have 13 Lagrange Multipliers, with them computing the reaction forces and torques produced by the 13 GCons
- We said that the reaction force and torque associated with a basic GCon Φ^α that has the Lagrange Multiplier λ_α are

$$\mathbf{F}_i^{r,\alpha} = -[\Phi^\alpha_{\mathbf{r}_i}]^T \lambda_\alpha \qquad \bar{\mathbf{n}}_i^{r,\alpha} = -\bar{\mathbf{\Pi}}_i^T(\Phi^\alpha)\lambda_\alpha$$

  - Note that if $\mathbf{r}_i$ does not enter the expression of Φ^α then $\mathbf{F}_i^{r,\alpha} = \mathbf{0}_{3\times 1}$ since $\Phi^\alpha_{\mathbf{r}_i} = \mathbf{0}_{1\times 3}$
  - Likewise, if Φ^α has no explicit dependence on orientation, then $\bar{\mathbf{\Pi}}_i(\Phi^\alpha) = \mathbf{0}_{1\times 3}$ and therefore $\bar{\mathbf{n}}_i^{r,\alpha} = \mathbf{0}_{3\times 1}$ (this scenario is not that common)