Introduction to Multi-body Dynamics

Chandramouli P

Indian Institute of Technology Madras

August 2, 2023

1. Overview

2. Math Overview

What is Multi-body Dynamics?

- Mechanical systems usually consist of many components
- They are connected by joints and/or by actuators
- An example is an Excavator ▸ Excavator
  - These are now called multi-body systems
- The relative motion and joint/actuator forces dictate the dynamics

Some Design Questions

- Relation between track-chain motion and forward vehicle velocity?
- Effect of friction between track-chain and ground on vehicle performance?
- Soil-track interaction?
- To answer some of these one needs to understand the dynamics of such a system
- This course will focus on developing the dynamic models for such systems

Classification

- Rigid Multi-body Systems
  - All components of mechanism/machine are assumed rigid
    - Represented by inertia
  - Acutuators or force elements modelled in terms of springs/dampers
- Flexible Multi-body Systems
  - Have rigid and deformable bodies
    - Distributed inertia and elasticity based on deformation
  - Elastic and inertia properties change with time
    - Hence more complex to analyze the dynamics

Motion and Constraints

- To understand the dynamics of multi-body system
  - We need to understand the motion of its components
- Joints are used to control the mobility of the system
- Actuators or force elements used to help multi-body system perform assigned tasks
  - Could be simple or complex
- We look first at unconstrained motion

Unconstrained Motion

- Rigid body motion composed of translation and rotation
- Finite rotation problem not trivial
  - Leads to geometric non-linearities
- We introduce a body-coordinate system $X^iY^iZ^i$ on rigid body i

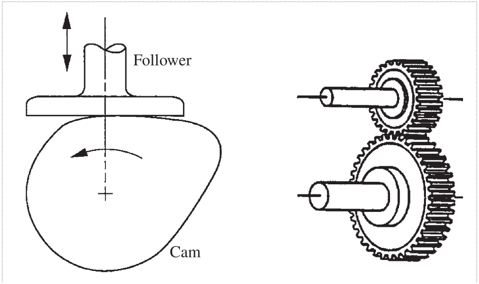
  ▸ Coordinates
  - Rigidly attached to point O^i
- Displacement of body i described by
  - Translation of O^i
  - Orientation of $X^iY^iZ^i$ with respect to inertial axes XYZ
- For planar motion
  - Two translations and rotation about axis $\perp$ to plane

Joints

- The components in a multi-body system not designed for arbitrary motion
  - Mechanical joints are used to constrain motion in certain directions ▸ Joints
  - First joint shown is prismatic joint
    - ★ Eliminates freedom of body i to translate relative to j in any direction other than joint axis
    - ★ Also prevents rotation of i with respect to j
  - Last one shown is spherical joint
    - ★ Does not allow any translation between body i and j
    - ★ Relative rotation about 3 perpendicular axes

Other Joints

- Other joints also are used in mechanical systems
  - Cam-follower and gears
- Cam rotates and constrains the follower to pre-designed translation or oscillation profile
- Gears used to transmit relative rotation about parallel or perpendicular axis



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Degree-of-freedom

- Each multi-body system is a combination of several bodies with joints and force elements or actuators
- Configuration described by co-ordinates of the bodies

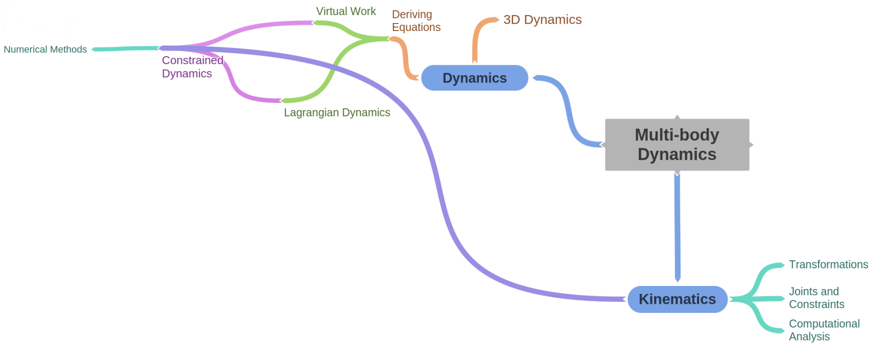
Degree-of-freedom

- Each multi-body system is a combination of several bodies with joints and force elements or actuators
- Configuration described by co-ordinates of the bodies
- Number of independent co-ordinates required to describe configuration is degree-of-freedom
- Let us look at the slider-crank mechanism used in IC engines
  SliderCrank
  - There are 4 bodies with 3 revolute joints and one prismatic joint
  - But only one degree-of-freedom
  - Gas force moving the piston rotates the crank and vice-versa

Multi Degree-of-freedom Systems

- Single degree-of-freedom systems usually are closed chains
- Open chains such as robotic manipulators have multiple degrees-of-freedom ▸ RobotArm
- Usually designed to mimic human functions with more precision
  - Applications to welding/ painting/ assembly tasks
- Remember that the number of independent co-ordinates is unique but not the co-ordinates themselves
  - Rotation of crank or translation of slider can be the degree-of-freedom considered

Mind-map



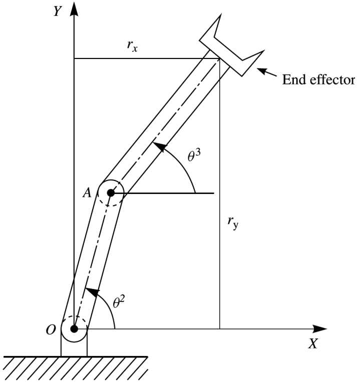
Numerical Methods
Constrained Dynamics
Virtual Work
Lagrangian Dynamics
Deriving Equations
3D Dynamics
Dynamics
Multi-body Dynamics
Kinematics
Transformations
Joints and Constraints
Computational Analysis

Kinematic Analysis

- Geometric aspect of motion of bodies without regard to forces that produce them
- Classical approach begins with identifying system degrees-of-freedom
- This is followed by position analysis
  - Determine location and orientation of bodies in the system
- Next is velocity analysis
  - From time differentiation of kinematic relations used in position analysis
- Final step is acceleration analysis
  - From time differentiation of velocity analysis equations

Example

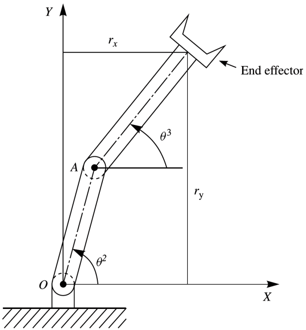
- Manipulator has 2 degrees-of-freedom
- Choose θ^2 and θ^3
- Two link lengths are l^2 and l^3
- Position of the end effector
  - $r_x = l^2 \cos\theta^2 + l^3 \cos\theta^3$
  - $r_y = l^2 \sin\theta^2 + l^3 \sin\theta^3$



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Velocity

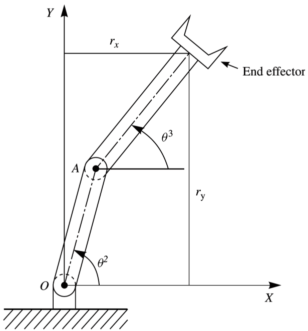
- Velocity of end effector
  - $\dot{r}_x = -l^2\dot{\theta}^2 \sin\theta^2 - l^3\dot{\theta}^3 \sin\theta^3$
  - $\dot{r}_y = l^2\dot{\theta}^2 \cos\theta^2 + l^3\dot{\theta}^3 \cos\theta^3$
- Given θ^2, θ^3 and angular velocities one can find the end effector velocity
- One can also determine velocity of any other point on link 2 or 3



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Acceleration

- Acceleration of end effector
  - $\ddot{r}_x = -l^2(\dot{\theta}^2)^2 \cos\theta^2 - l^3(\dot{\theta}^3)^2 \cos\theta^3 - l^2\ddot{\theta}^2 \sin\theta^2 - l^3\ddot{\theta}^3 \sin\theta^3$
  - $\ddot{r}_y = -l^2(\dot{\theta}^2)^2 \sin\theta^2 - l^3(\dot{\theta}^3)^2 \sin\theta^3 + l^2\ddot{\theta}^2 \cos\theta^2 + l^3\ddot{\theta}^3 \cos\theta^3$
- Given θ^2, θ^3 and their first and second derivatives acceleration can be found
- One can also determine acceleration of any other point on link 2 or 3



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Comments

- If all the kinematic quantities are specified then force equations are not required
- Position, velocity and acceleration at any point on the rigid bodies can be found
- Called a kinematically driven system
- If one or more degrees-of-freedom are unknown then force equations required
  - To find the configuration of the system
  - Such a system is called dynamically driven
- Identifying the appropriate degrees-of-freedom for a large system can become difficult with classical approach
- Require a computer based approach

Computational Approach

- Kinematic constraint equations for mechanical joints and specified motion trajectories formulated
- Leads to a relatively large system of non-linear algebraic equations
  - Solved using computer based numerical methods
- Basis for analyzing large size kinematically driven systems

Force Analysis

- Inertia forces which depend on mass and shape of body as well as acceleration/velocity
- Joint forces arising due to constraints at connections between bodies
- External forces are those that are not the above two
  - Spring/ Damper forces or motor torque or actuator forces or gravity
- Motion now governed by second order differential equations
  - Can be derived using principle of virtual work or Lagrange's equations apart from classical Newtonian approach

Different Forms

- Assuming no friction force one can write the following equation for horizontal motion of a block ▸ Block
  - $m\ddot{x} = F$
  - x is the block co-ordinate and F the external force
- No reaction forces in equation as motion described by degree-of-freedom x
- Instead one could write the following set of equations too
  - $m\ddot{x} = F$
  - $m\ddot{y} = N - mg$
- We have introduced a redundant co-ordinate y

Different Forms

- $m\ddot{x} = F$; $m\ddot{y} = N - mg$
- We have introduced a redundant co-ordinate y which brings in the reaction force N
- Assuming F known need one more equation to solve
  - Unknowns are $\ddot{x}$, $\ddot{y}$ and N
- Done by imposing constraint on motion of block in vertical direction
  - $y = c$ where c is a constant
- We have 2 differential equations plus an algebraic equation that need to be solved together
  - Termed Differential Algebraic Equations (DAE)

Use of Redundant Co-ordinates

- Has computational advantages as demonstrated later in the course
  - Increases generality and flexibility of the formulation
  - Widely used in many commercially available software
- Note that the number of reactions is equal to number of redundant co-ordinates
  - True for all systems of any size

Forward & Inverse Dynamics

- Acceleration using first form
  - $\ddot{x} = \frac{F}{m}$
- On integration one can get velocity and displacement
  - $\dot{x} = \dot{x}_0 + \int_0^t \frac{F}{m} dt$; $x = x_0 + \int_0^t \dot{x} dt$
- We need to specify two initial conditions x_0 and $\dot{x}_0$
- In complex systems the velocity and displacement have to be obtained numerically
- For inverse dynamics kinematic quantities are specified and forces calculated from algebraic equations

Computer Implementations

- In the first form the reactions are eliminated
  - Expressing them in terms of the degrees-of-freedom
  - Usually joint variables used as degrees-of-freedom
- Reduced size of the problem
  - But highly non-linear
- Called embedding techniques
  - Forms the basis of recursive methods used in robotic manipulators

Second Form

- Equations of motion include redundant co-ordinates
- Since not independent requires kinematic constraint equations connecting these co-ordinates
- Constraint equations appear in governing equations
  - Called augmented formulation
- Drawbacks
  - Larger number of equations
  - Complexity of solution algorithms
- Advantage is the sparse matrix structure from simpler form of equations
- Computationally more efficient to solve such systems

Defining Orientation

- In spatial dynamics many sets of orientation co-ordinates used
  - To represent three-dimensional motion
- Some of these lack physical meaning
  - Difficult to define initial configuration using them
- Define 3 points on a rigid body and use vector cross product
  - Location and orientation of body Cartesian co-ordinate system

Defining Orientation

- Consider body i with co-ordinate system $X^iY^iZ^i$ at origin O^i

  CartCoord

- Two other points P^i and Q^i are chosen
  - P^i is along X^i axis
  - Q^i is on the X^iY^i plane
- Position vectors in XYZ are $\mathbf{r}^i_O$, $\mathbf{r}^i_P$ and $\mathbf{r}^i_Q$
  - Define unit vectors $\mathbf{i}^i = \frac{\mathbf{r}^i_P - \mathbf{r}^i_O}{|\mathbf{r}^i_P - \mathbf{r}^i_O|}$ and $\mathbf{i}^i_t = \frac{\mathbf{r}^i_Q - \mathbf{r}^i_O}{|\mathbf{r}^i_Q - \mathbf{r}^i_O|}$
  - Unit vector $\mathbf{k}^i$ along Z^i given by $\mathbf{k}^i = \frac{\mathbf{i}^i \times \mathbf{i}^i_t}{|\mathbf{i}^i \times \mathbf{i}^i_t|}$
  - Unit vector along Y^i is then simply $\mathbf{j}^i = \mathbf{k}^i \times \mathbf{i}^i$

Orientation ...

- Orientation of body co-ordinate system with respect to XYZ
  - $\boxed{\mathbf{A} = \begin{bmatrix} \mathbf{i}^i & \mathbf{j}^i & \mathbf{k}^i \end{bmatrix}}$
  - Direction Cosine matrix
- Location of body co-ordinate system given by $\mathbf{r}_O^i$
- Let us look at an example
  - $\mathbf{r}_O = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^{\mathrm{T}}$; $\mathbf{r}_P = \begin{bmatrix} 0 & 2 & 3 \end{bmatrix}^{\mathrm{T}}$; $\mathbf{r}_Q = \begin{bmatrix} -1 & 1 & 3 \end{bmatrix}^{\mathrm{T}}$
- From this we have
  - $\mathbf{i}^i = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$; $\mathbf{i}_t^i = \begin{bmatrix} -\frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & 0 \end{bmatrix}^{\mathrm{T}}$
  - $\mathbf{k}^i = \frac{\mathbf{i}^i \times \mathbf{i}_t^i}{|\mathbf{i}^i \times \mathbf{i}_t^i|} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$
  - $\mathbf{j}^i = \mathbf{k}^i \times \mathbf{i}^i = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^{\mathrm{T}}$

Parallelism

- We require for certain joint constraints to express fact that two vectors are parallel
- If $\mathbf{a}^i$ is parallel to $\mathbf{a}^j$ then we have
  - $\mathbf{a}^i \times \mathbf{a}^j = \mathbf{0}$
  - We have 3 scalar equations out of which only 2 are independent
- Instead use dot product to express parallelism
  - Form orthogonal triad $\mathbf{a}^i$, $\mathbf{a}_1^i$ and $\mathbf{a}_2^i$ OrthTriad
  - We have $\mathbf{a}_1^{i\,\mathrm{T}}\mathbf{a}^j = 0$; $\mathbf{a}_2^{i\,\mathrm{T}}\mathbf{a}^j = 0$
  - Two independent equations
- How do we generate an orthogonal triad?

Generating Orthogonal Triad

- Determine $\mathbf{a}_d$ not parallel to $\mathbf{a}^i$
- Get $\mathbf{a}_d$ by making one element zero
  - The element corresponding to largest entry in $\mathbf{a}^i$ is set to zero in $\mathbf{a}_d$
  - The other two are set to 1
- Generate vector $\mathbf{a}_1^i$ as $\mathbf{a}^i \times \mathbf{a}_d$
- Now $\mathbf{a}_2^i = \mathbf{a}_1^i \times \mathbf{a}^i$
- We look at an example to demonstrate the procedure

Triad Example

- Let $\mathbf{a}^i = \begin{bmatrix} 1 & 0 & -3 \end{bmatrix}^{\mathrm{T}}$
- Third element is largest in magnitude
  - $\mathbf{a}_d = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$
- From this we have $\mathbf{a}_1^i = \begin{bmatrix} 3 & -3 & 1 \end{bmatrix}^{\mathrm{T}}$
- And finally we have $\mathbf{a}_2^i = \begin{bmatrix} 9 & 10 & 3 \end{bmatrix}^{\mathrm{T}}$
- We now look at a matrix-vector representation of a cross product
  - $\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 & a_3 b_1 - a_1 b_3 & a_1 b_2 - a_2 b_1 \end{bmatrix}^{\mathrm{T}}$

Cross Product

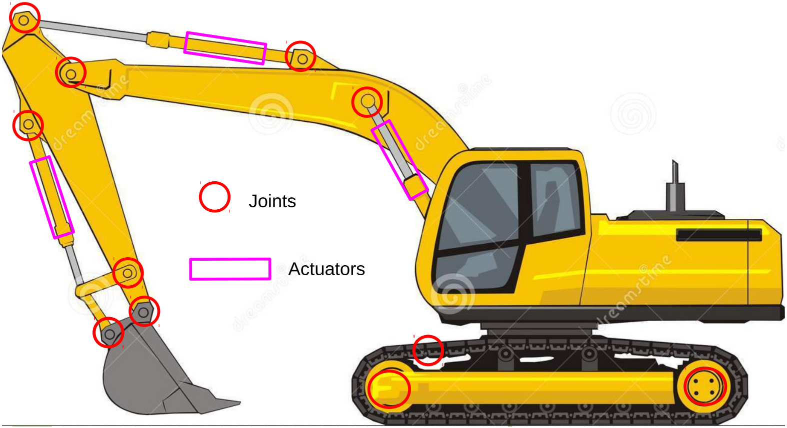
- This can be written in matrix-vector form as

$$\mathbf{a} \times \mathbf{b} = \tilde{\mathbf{a}}\mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

- One can see that $\tilde{\mathbf{a}}$ is a skew-symmetric matrix
- Now $\mathbf{b} \times \mathbf{a} = -\mathbf{a} \times \mathbf{b}$ can be expressed as

$$\mathbf{b} \times \mathbf{a} = \tilde{\mathbf{b}}\mathbf{a} = \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix}$$

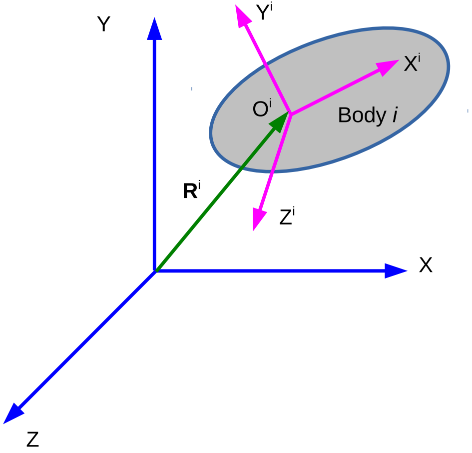
Excavator Example



Courtesy: https://www.dreamstime.com/stock-images-excavator-image15419104

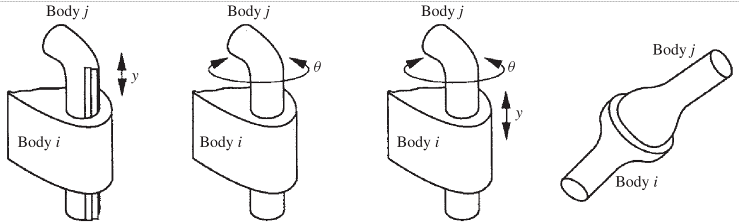
Back

Rigid Body Coordinates



Return

Joints



(a)
Prismatic
joint

(b)
Revolute
joint

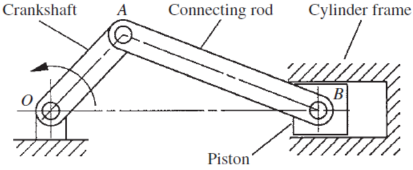
(c)
Cylindrical
joint

(d)
Spherical
joint

Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Back

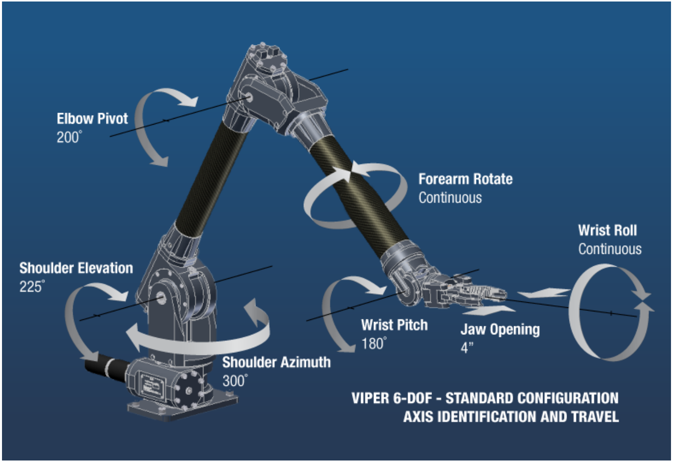
IC Engine Mechanism



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Return

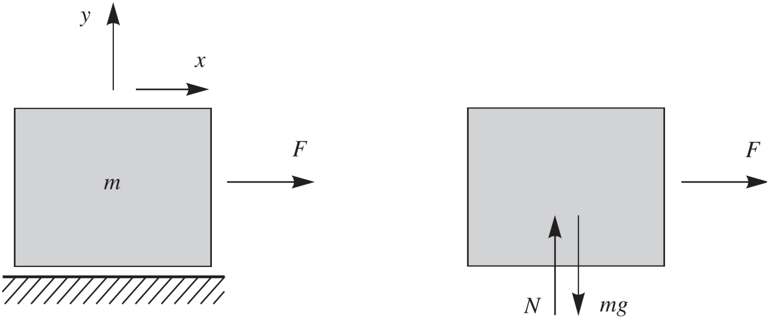
Robotic Arm



Courtesy: http://edge.rit.edu/edge/P14253/public/Home

Back

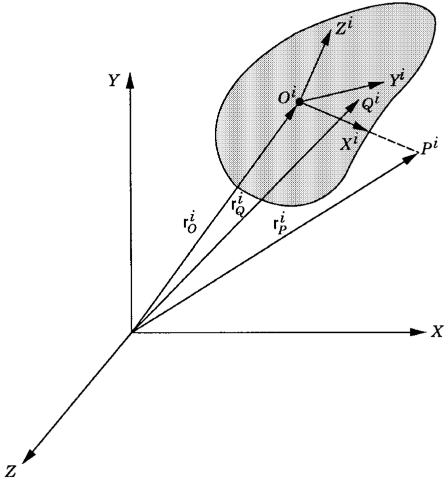
Sliding Block



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

◂ Return

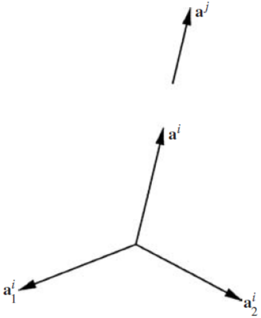
Finding Orientation



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Back

Orthogonal Triad



Courtesy: A. A. Shabana, 2010, *Computational Dynamics*, Third Edition, John Wiley & Sons.

Return