Robot Dynamics & Control

METR 4202: Advanced Control & Robotics

Drs Surya Singh, Paul Pounds, and Hanna Kurniawati

Lecture # 4 August 13, 2012

metr4202@itee.uq.edu.au

http://itee.uq.edu.au/~metr4202/

RoboticsCourseWare Contributor

Schedule

Week	Date	Lecture (M: 12-1:30, 43-102)
1	23-Jul	Introduction
2	30-Jul	Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations)
3	6-Aug	Robot Kinematics and Dynamics
4	13-Aug	Robot Dynamics & Control
5	20-Aug	Obstacle Avoidance & Motion Planning
6	27-Aug	Sensors, Measurement and Perception
7	3-Sep	Localization and Navigation
8	10-Sep	State-space modelling & Controller Design
9	17-Sep	Vision-based control
	24-Sep	*Study break*
10	1-Oct	Uncertainty/POMDPs
11	8-Oct	Robot Machine Learning
12	15-Oct	Guest Lecture
13	22-Oct	Wrap-up & Course Review

Recap from Last Week [1]:
DH: Where to place frame?

1. Align an axis along principal motion
   1. Rotary (**R**): align rotation axis along the z axis
   2. Prismatic (**P**): align slider travel along x axis

2. Orient so as to position x axis towards next frame

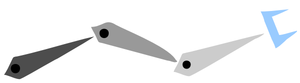
3. $\theta_{\text{(rot } z)} \rightarrow d_{\text{(trans } z)} \rightarrow a_{\text{(trans } x)} \rightarrow \alpha_{\text{(rot } x)}$

Recap from Last Week [2]:
Inverse Kinematics

- Forward: angles → position

  $\mathbf{x} = f(\boldsymbol{\theta})$
- Inverse: position → angles

  $\boldsymbol{\theta} = f^{-1}(\mathbf{x})$
- Analytic Approach



- Numerical Approaches:
  - Jacobian: $J = \frac{\delta x}{\delta q} \rightarrow \delta q \approx J^{-1}\delta x$
  - J^T Approximation: $\tau = J^T \cdot \mathbf{F} \rightarrow \Delta q \approx J^T \Delta x$
    - Slotine & Sheridan method
  - Cyclical Coordinate Descent

Recap from Last Week [3]: Inverse Kinematics

- Inverse Kinematics is the problem of finding the joint parameters given only the values of the homogeneous transforms which model the mechanism
(i.e., the pose of the end effector)

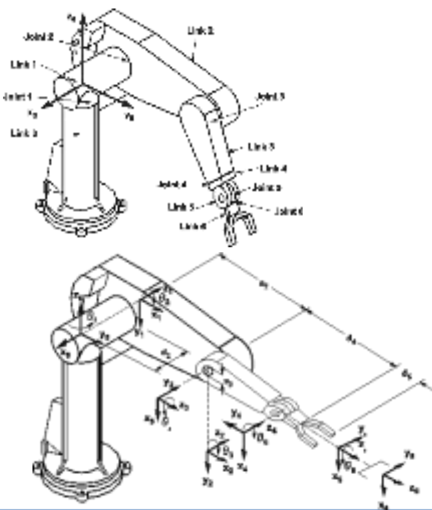
- Solves the problem of where to drive the joints in order to get the hand of an arm or the foot of a leg in the right place

- In general, this involves the solution of a set of simultaneous, non-linear equations

- Hard for serial mechanisms, easy for parallel

Inverse kinematics

- What about a more complicated mechanism?

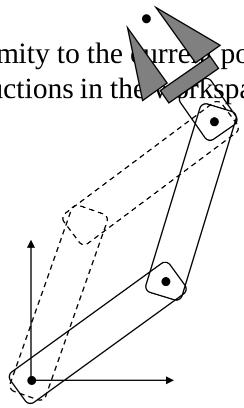


$$ {}^0T_6 = {}^0T_1\,{}^1T_2\,{}^2T_3\,{}^3T_4\,{}^4T_5\,{}^5T_6 = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$
\begin{aligned}
n_x &= c_1(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6) - s_1(s_4c_5c_6 + c_4s_6) \\
n_y &= s_1(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6) + c_1(s_4c_5c_6 + c_4s_6) \\
n_z &= -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6 \\
s_x &= c_1(-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6) - s_1(-s_4c_5s_6 + c_4c_6) \\
s_y &= s_1(-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6) + c_1(-s_4c_5s_6 + c_4c_6) \\
s_z &= s_{23}(c_4c_5s_6 + s_4c_6) - c_{23}s_5s_6 \\
a_x &= c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5 \\
a_y &= s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5 \\
a_z &= -s_{23}c_4s_5 + c_{23}c_5) \\
p_x &= c_1(d_6(c_{23}c_4s_5 + s_{23}c_5) + s_{23}d_4 + a_3c_{23} + a_2c_2) - s_1(d_6s_4s_5 + d_2) \\
p_y &= s_1(d_6(c_{23}c_4s_5 + s_{23}c_5) + s_{23}d_4 + a_3c_{23} + a_2c_2) + c_1(d_6s_4s_5 + d_2) \\
p_z &= d_6(c_{23}c_5 - s_{23}c_4s_5) + c_{23}d_4 - a_3s_{23} - a_2s_2
\end{aligned}
$$

Multiple Solutions

- There will often be multiple solutions for a particular inverse kinematic analysis
- Consider the three link manipulator shown. Given a particular end effector pose, two solutions are possible
- The choice of solution is a function of proximity to the current pose, limits on the joint angles and possible obstructions in the workspace

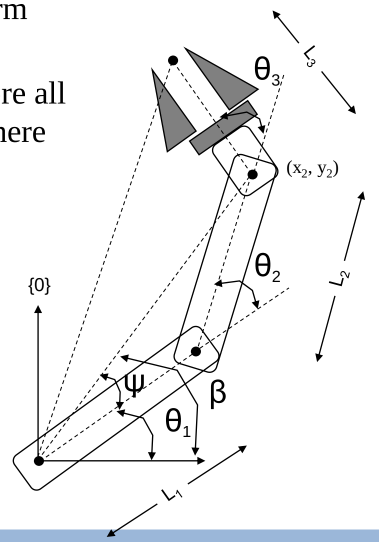


Solution Methods

- Unlike with systems of linear equations, there are no general algorithms that may be employed to solve a set of nonlinear equation
- **Closed-form** and **numerical** methods exist
- We will concentrate on analytical, closed-form methods
- These can be characterized by two methods of obtaining a solution: **algebraic** and **geometric**

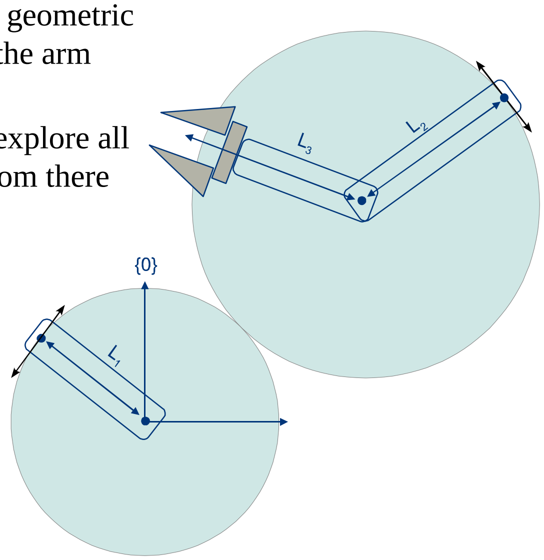
Inverse Kinematics: Geometrical Approach

- We can also consider the geometric relationships defined by the arm
- Start with what is fixed, explore all geometric possibilities from there



Inverse Kinematics: Geometrical Approach [2]

- We can also consider the geometric relationships defined by the arm
- Start with what is fixed, explore all geometric possibilities from there



Inverse Kinematics: Algebraic Approach

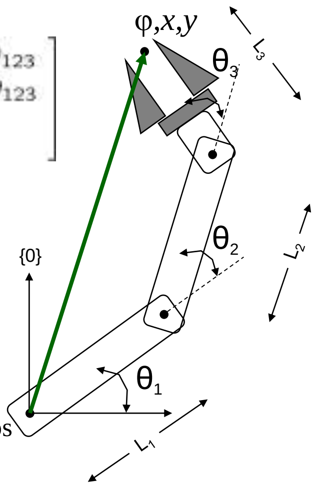
- We have a series of equations which define this system
- Recall, from Forward Kinematics:

$$^{0}T_3 = \begin{bmatrix} c_{\theta_{123}} & -s_{\theta_{123}} & 0 & L_1 c_{\theta_1} + L_2 c_{\theta_{12}} + L_3 c_{\theta_{123}} \\ s_{\theta_{123}} & c_{\theta_{123}} & 0 & L_1 s_{\theta_1} + L_2 s_{\theta_{12}} + L_3 s_{\theta_{123}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The end-effector pose is given by

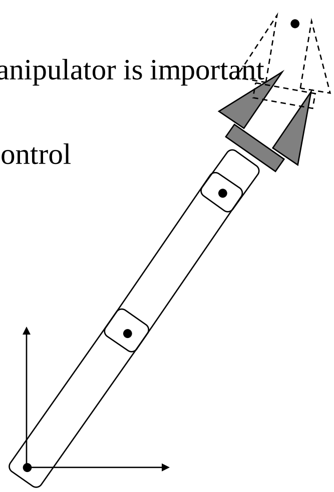
$$^{0}T_3 = \begin{bmatrix} c_{\phi} & -s_{\phi} & 0 & x \\ s_{\phi} & c_{\phi} & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Equating terms gives us a set of algebraic relationships



No Solution - Singularity

- Singular positions:
- An understanding of the workspace of the manipulator is important
- There will be poses that are not achievable
- There will be poses where there is a loss of control
- Singularities also occur when the manipulator loses a DOF
  - This typically happens when joints are aligned
  - **det[Jacobian]=0**



Velocity

- Recall that we can specify a point in one frame relative to another as

$$^{A}\mathbf{P} = {}^{A}\mathbf{P}_{B} + {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{P}$$

- Differentiating w/r/t to **t** we find

$$^{A}\mathbf{V}_{P} = \frac{d}{dt}{}^{A}\mathbf{P} = \lim_{\Delta t \to 0} \frac{{}^{A}\mathbf{P}(t+\Delta t) - {}^{A}\mathbf{P}(t)}{\Delta t}$$
$$= {}^{A}\dot{\mathbf{P}}_{B} + {}^{A}_{B}\mathbf{R}\,{}^{B}\dot{\mathbf{P}} + {}^{A}_{B}\dot{\mathbf{R}}\,{}^{B}\mathbf{P}$$

- This can be rewritten as

$$^{A}\mathbf{V}_{P} = {}^{A}\mathbf{V}_{BORG} + {}^{A}\mathbf{R}_{B}\,{}^{B}\mathbf{V}_{P} + {}^{A}\Omega_{B} \times {}^{A}\mathbf{R}_{B}\,{}^{B}\mathbf{P}$$

Skew – Symmetric Matrix

$$\mathbf{V} = \omega \times \mathbf{r}$$

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

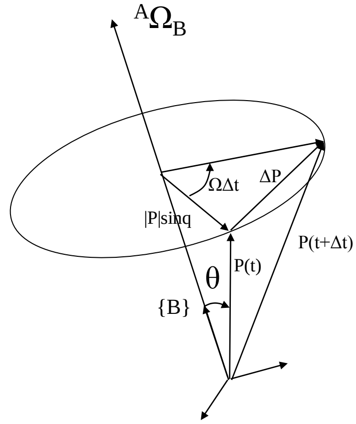
$$\rightarrow \mathbf{V} = \Omega\mathbf{r}$$

Angular Velocity

- If we look at a small timeslice as a frame rotates with a moving point, we find

$$|\Delta \mathbf{P}| = \left(\left|{}^{A}\mathbf{P}\right| \sin\theta\right)\left(\left|{}^{A}\Omega_B\right| \Delta t\right)$$
$$\frac{|\Delta \mathbf{P}|}{\Delta t} = \left(\left|{}^{A}\mathbf{P}\right| \sin\theta\right)\left(\left|{}^{A}\Omega_B\right|\right)$$
$$= {}^{A}\Omega_B \times {}^{A}\mathbf{P}$$

$${}^{A}\mathbf{V}_P = {}^{A}\Omega_B \times {}^{A}R_B\,{}^{B}\mathbf{P}$$



Velocity Representations

- Euler Angles
  - For Z-Y-X (α,β,γ):

$$\begin{pmatrix} \dot\alpha \\ \dot\beta \\ \dot\gamma \end{pmatrix} = \begin{pmatrix} -S\beta & 0 & 1 \\ C\beta S\gamma & C\gamma & 0 \\ C\beta C\gamma & -S\beta & 0 \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

- Quaternions

$$\begin{pmatrix} \dot\varepsilon_0 \\ \dot\varepsilon_1 \\ \dot\varepsilon_2 \\ \dot\varepsilon_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \varepsilon_1 & -\varepsilon_2 & -\varepsilon_3 \\ \varepsilon_0 & \varepsilon_3 & -\varepsilon_2 \\ -\varepsilon_3 & \varepsilon_0 & \varepsilon_1 \\ \varepsilon_2 & -\varepsilon_1 & \varepsilon_0 \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

Manipulator Velocities

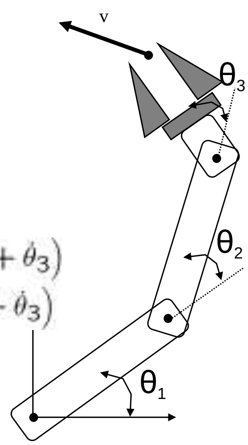
- Consider again the schematic of the planar manipulator shown. We found that the end effector position is given by

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

- Differentiating w/r/t to t

$$\dot{x} = -L_1 \mathsf{s}_1 \dot{\theta}_1 - L_2 \mathsf{s}_{12} \left(\dot{\theta}_1 + \dot{\theta}_2\right) - L_3 \mathsf{s}_{123} \left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3\right)$$
$$\dot{y} = L_1 \mathsf{c}_1 \dot{\theta}_1 + L_2 \mathsf{c}_{12} \left(\dot{\theta}_1 + \dot{\theta}_2\right) + L_3 \mathsf{c}_{123} \left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3\right)$$

- This gives the end effector velocity as a function of pose and joint velocities



Manipulator Velocities [2]

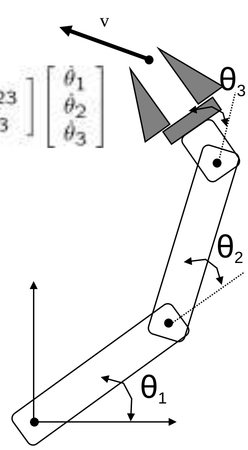
- Rearranging, we can recast this relation in matrix form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -L_1 \mathsf{s}_1 - L_2 \mathsf{s}_{12} - L_3 \mathsf{s}_{123} & -L_2 \mathsf{s}_{12} - L_3 \mathsf{s}_{123} & -L_3 \mathsf{s}_{123} \\ L_1 \mathsf{c}_1 + L_2 \mathsf{c}_{12} + L_3 \mathsf{c}_{123} & L_2 \mathsf{c}_{12} + L_3 \mathsf{c}_{123} & L_3 \mathsf{c}_{123} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

- Or

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

- The resulting matrix is called the Jacobian and provides us with a mapping from Joint Space to Cartesian Space.



Manipulator Velocities [3]: The Jacobian

- In general, the Jacobian takes the form (for example, **j joints** and in **i operational space**)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial \theta_1} & \frac{\partial x_1}{\partial \theta_2} & \cdots & \frac{\partial x_1}{\partial \theta_j} \\ \frac{\partial x_2}{\partial \theta_1} & \frac{\partial x_2}{\partial \theta_2} & \cdots & \frac{\partial x_2}{\partial \theta_j} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_i}{\partial \theta_1} & \frac{\partial x_i}{\partial \theta_2} & \cdots & \frac{\partial x_i}{\partial \theta_j} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_j \end{bmatrix}$$

- Or more succinctly

$$\dot{\mathbf{X}} = \mathbf{J}(\theta)\dot{\theta}$$

Moving On…Differential Motion

- Transformations also encode differential relationships
- Consider a manipulator (say 2DOF, RR)

$$x(\theta_1, \theta_2) = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$
$$y(\theta_1, \theta_2) = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

- Differentiating with respect to the **angles** gives:

$$dx = \frac{\partial x(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial x(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2$$

$$dy = \frac{\partial y(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial y(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2$$

Differential Motion [2]

- Viewing this as a matrix → Jacobian

$$d\mathbf{x} = J d\theta$$

$$J = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

$$J = \begin{bmatrix} [J_1] & [J_2] \end{bmatrix}$$

$$v = J_1 \dot{\theta}_1 + J_2 \dot{\theta}_2$$

Infinitesimal Rotations

- $\cos(d\phi) = 1, \sin(d\phi) = d\phi$

$$\mathbf{R}_x(d\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cd\phi & -sd\phi \\ 0 & sd\phi & cd\phi \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -d\phi_x \\ 0 & d\phi_x & 1 \end{bmatrix}$$

$$\mathbf{R}_y(d\phi) = \begin{bmatrix} cd\phi & 0 & sd\phi \\ 0 & 1 & 0 \\ -sd\phi & 0 & cd\phi \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & d\phi_y \\ 0 & 1 & 0 \\ -d\phi_y & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_z(d\phi) = \begin{bmatrix} cd\phi & -sd\phi & 0 \\ sd\phi & cd\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -d\phi_z & 0 \\ d\phi_z & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Note that:

$$R_x(d\varphi) R_y(d\varphi) = R_y(d\varphi) R_x(d\varphi)$$

→ Therefore … they **commute**

Jacobian [2]

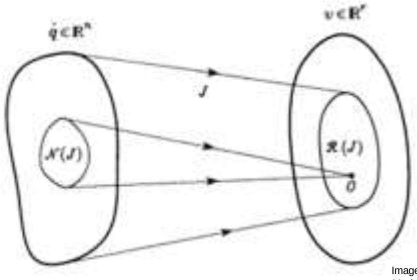


Image:. Sciavicco and Siciliano, *Modelling and Control of Robot Manipulators*, 2nd *ed*, 2000

- Jacobian can be viewed as a mapping from Joint velocity space ($\dot{q}$) to Operational velocity space (*v*)

Inverse Jacobian

- In many instances, we are also interested in computing the set of joint velocities that will yield a particular velocity at the end effector

$$\dot{\theta} = \mathbf{J}(\theta)^{-1}\dot{\mathbf{X}}$$

- We must be aware, however, that the inverse of the Jacobian may be undefined or singular. The points in the workspace at which the Jacobian is undefined are the *singularities* of the mechanism.
- Singularities typically occur at the workspace boundaries or at interior points where degrees of freedom are lost

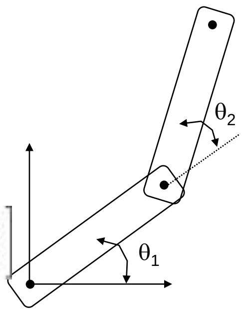
Inverse Jacobian Example

- For a simple two link RR manipulator:

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

- The Jacobian for this is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$



- Taking the inverse of the Jacobian yields

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \frac{1}{L_1 L_2 s_2} \begin{bmatrix} L_2 c_{12} & L_2 s_{12} \\ -L_1 c_1 - L_2 c_{12} & -L_1 s_1 - L_2 s_{12} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

- Clearly, as θ_2 approaches 0 or π this manipulator becomes singular

Static Forces

- We can also use the Jacobian to compute the joint torques required to maintain a particular force at the end effector
- Consider the concept of virtual work

$$F \cdot \delta\mathbf{X} = \tau \cdot \delta\theta$$

- Or

$$F^T \delta\mathbf{X} = \tau^T \delta\theta$$

- Earlier we saw that

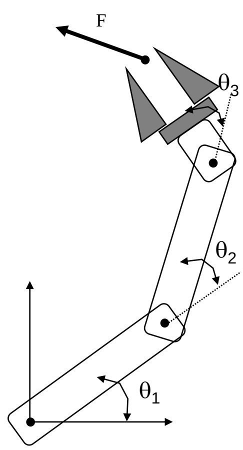
$$\delta\mathbf{X} = \mathbf{J}\delta\theta$$

- So that

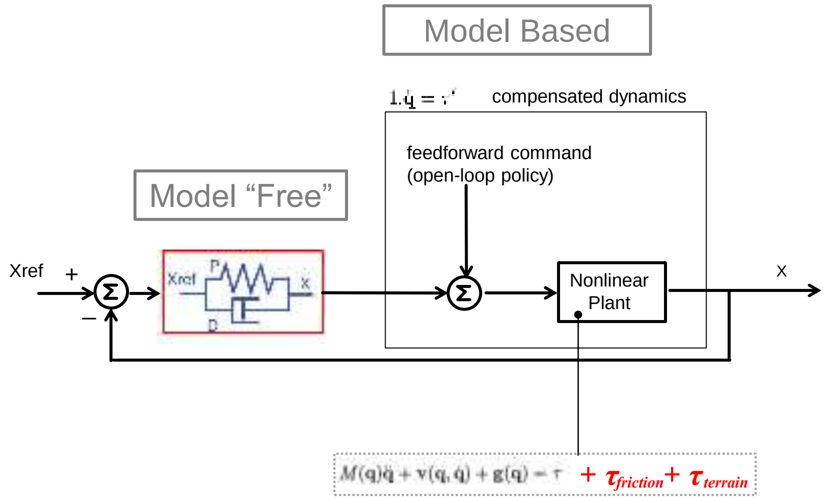
$$F^T \mathbf{J} = \tau^T$$

- Or

$$\tau = \mathbf{J}^T F$$



Operation Space (Computed Torque)



Compensated Manipulation

Dynamics

- We can also consider the forces that are required to achieve a particular motion of a manipulator or other body
- Understanding the way in which motion arises from torques applied by the actuators or from external forces allows us to control these motions
- There are a number of methods for formulating these equations, including
  - Newton-Euler Dynamics
  - Langrangian Mechanics

Dynamics

- For Manipulators, the general form is

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta)$$

where

  - τ is a vector of joint torques
  - Θ is the nx1 vector of joint angles
  - $M(\Theta)$ is the nxn mass matrix
  - $V(\Theta, \dot{\Theta})$is the nx1 vector of centrifugal and Coriolis terms
  - $G(\Theta)$ is an nx1 vector of gravity terms

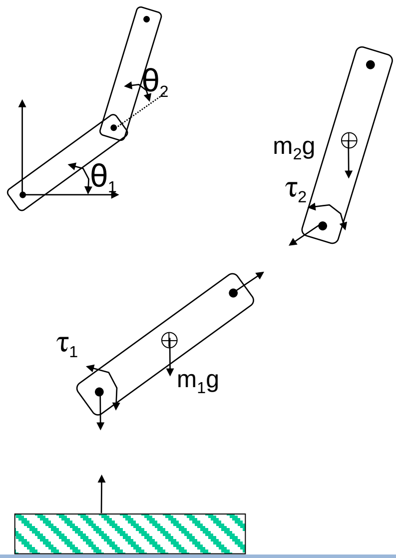
- Notice that all of these terms depend on Θ so the dynamics varies as the manipulator move

Dynamics – Newton-Euler

- In general, we could analyse the dynamics of robotic systems using classical Newtonian mechanics

$$\sum F = m\ddot{x}$$
$$\sum T = J\ddot{\theta}$$

- This can entail iteratively calculating velocities and accelerations for each link and then computing force and moment balances in the system
- Alternatively, closed form solutions may exist for simple configurations



Compensation

Dynamics – Langrangian Mechanics

- Alternatively, we can use Langrangian Mechanics to compute the dynamics of a manipulator (or other robotic system)
- The Langrangian is defined as the difference between the Kinetic and Potential energy in the system
- Using this formulation and the concept of virtual work we can find the forces and torques acting on the system.
- This may seem more involved but is often easier to formulate for complex systems

$$L = K - P$$

$$F_i = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{x}_i}\right) - \frac{\partial L}{\partial x}$$

$$T_i = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta}$$

Summary

- Kinematics is the study of motion without regard to the forces that create it
- Kinematics is important in many instances in Robotics
- The study of dynamics allows us to understand the forces and torques which act on a system and result in motion
- Understanding these motions, and the required forces, is essential for designing these systems

Cool Robotics Share