# Robot Motion & Sensing

METR 4202: Advanced Control & **Robotics**

Dr Surya Singh -- Lecture # 5

**August 27, 2014**

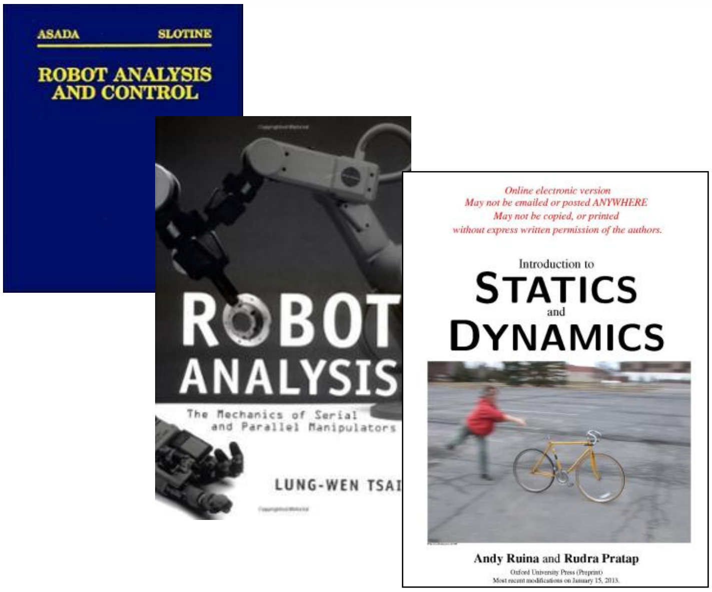
**metr4202@itee.uq.edu.au**
http://robotics.itee.uq.edu.au/~metr4202/



## Schedule

| Week | Date | Lecture (W: 11:10-12:40, 24-402) |
|---|---|---|
| 1 | 30-Jul | Introduction |
| 2 | 6-Aug | Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations) |
| 3 | 13-Aug | Robot Kinematics (& Ekka Day) |
| 4 | 20-Aug | Robot Dynamics & Control |
| 5 | 27-Aug | **Robot Motion & Sensing** |
| 6 | 3-Sep | Perception (Computer Vision) |
| 7 | 10-Sep | Multiple View Geometry (Computer Vision) |
| 8 | 17-Sep | Navigation & Localization (+ Prof. M. Srinivasan) |
| 9 | 24-Sep | Motion Planning + Control |
| | 1-Oct | *Study break* |
| 10 | 8-Oct | State-Space Modelling |
| 11 | 15-Oct | Shaping the Dynamic Response |
| 12 | 22-Oct | Linear Observers & LQR |
| 13 | 29-Oct | Applications in Industry & Course Review |

## Reference Material






## Outline

- Newton-Euler Formulation
- Lagrange Formulation

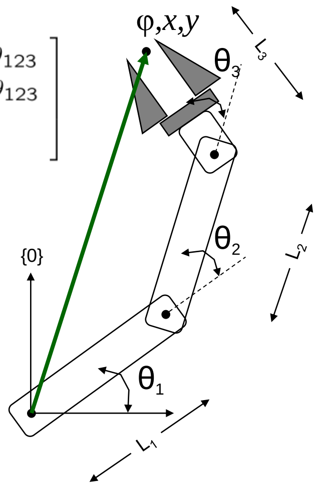
---

- Sensing & Perception

# 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

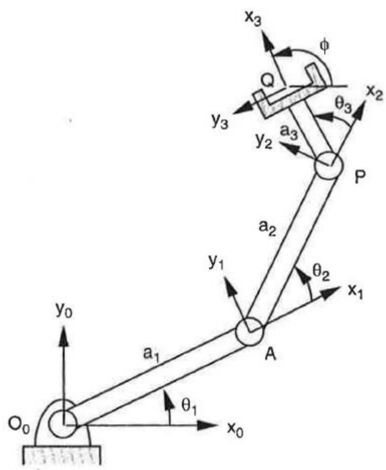
$$
{}^{0}T_{3} = \begin{bmatrix} c_{\phi} & -s_{\phi} & 0 & x \\ s_{\phi} & c_{\phi} & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

- Equating terms gives us a set of algebraic relationships



# Inverse Kinematics: Example I

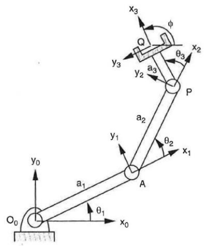
Planar Manipulator:



## Inverse Kinematics: Example I

- Forward Kinematics:

[For the Frame {Q} at the end effector]:




$$\begin{bmatrix} q_x \\ q_y \\ q_z \\ 1 \end{bmatrix} = {}^0A_3 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \\ 1 \end{bmatrix}$$

$\because$

$${}^0A_3 = \begin{bmatrix} c\theta_{123} & -s\theta_{123} & 0 & a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ s\theta_{123} & c\theta_{123} & 0 & a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- For an arbitrary point **G** in the end effector: ${}^3\mathbf{g} = [g_u, g_v, 0, 1]^T$

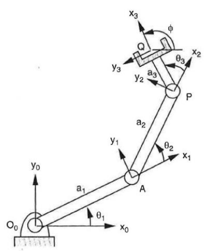
$$\begin{bmatrix} g_x \\ g_y \\ g_z \\ 1 \end{bmatrix} = {}^0A_3 \begin{bmatrix} g_u \\ g_v \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} g_u c\theta_{123} - g_v s\theta_{123} + a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ g_u s\theta_{123} + g_v c\theta_{123} + a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \\ 1 \end{bmatrix}$$



## Inverse Kinematics: Example I

- Forward Kinematics:

[For the Frame {Q} at the end effector]:




$$\begin{bmatrix} q_x \\ q_y \\ q_z \\ 1 \end{bmatrix} = {}^0A_3 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \\ 1 \end{bmatrix}$$

$\because$

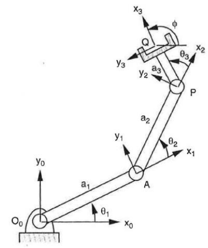
$${}^0A_3 = \begin{bmatrix} c\theta_{123} & -s\theta_{123} & 0 & a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ s\theta_{123} & c\theta_{123} & 0 & a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- For an arbitrary point **G** in the end effector: ${}^3\mathbf{g} = [g_u, g_v, 0, 1]^T$

$$\begin{bmatrix} g_x \\ g_y \\ g_z \\ 1 \end{bmatrix} = {}^0A_3 \begin{bmatrix} g_u \\ g_v \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} g_u c\theta_{123} - g_v s\theta_{123} + a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ g_u s\theta_{123} + g_v c\theta_{123} + a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \\ 1 \end{bmatrix}$$

## Inverse Kinematics: Example I

- Inverse Kinematics:
  - Set the final position equal to the Forward Transformation Matrix ${}^{0}\mathbf{A}_3$:




$$
{}^{0}A_3 = \begin{bmatrix} c\phi & -s\phi & 0 & q_x \\ s\phi & c\phi & 0 & q_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

- The solution strategy is to equate the elements of ${}^{0}\mathbf{A}_3$ to that of the given position $(q_x, q_y)$ and orientation $\phi$



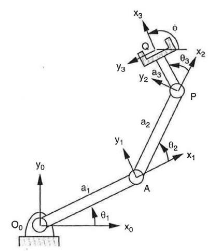
## Inverse Kinematics: Example I

- Orientation ($\phi$):

$$c\theta_{123} = c\phi,$$

$$s\theta_{123} = s\phi.$$

$$\theta_{123} = \theta_1 + \theta_2 + \theta_3 = \phi.$$




- Now Position of the 2DOF point **P**:

$$p_x = a_1 c\theta_1 + a_2 c\theta_{12},$$

$$p_y = a_1 s\theta_1 + a_2 s\theta_{12},$$

$\therefore \quad p_x = q_x - a_3 c\phi \qquad p_y = q_y - a_3 s\phi$

- Substitute: $\theta_3$ disappears and now we can eliminate $\theta_1$:

$$p_x^2 + p_y^2 = a_1^2 + a_2^2 + 2a_1 a_2 c\theta_2.$$

# Inverse Kinematics: Example I

- we can eliminate $\theta_1$…

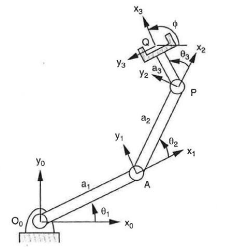
  $$p_x^2 + p_y^2 = a_1^2 + a_2^2 + 2a_1a_2c\theta_2.$$

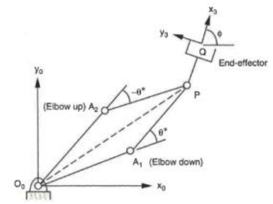
- Then solve for $\theta_{12}$:

  $$\theta_2 = \cos^{-1}\kappa, \qquad \kappa = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1a_2}$$

  – This gives 2 real ($\mathbb{R}$) roots if $|\kappa| < 1$
  – One double root if $|\kappa| = 1$
  – No real roots if $|\kappa| > 1$

- Elbow up/down:
  – In general, **if** $\theta_2$ is a solution **then** $-\theta_2$ is a solution








# Inverse Kinematics: Example I

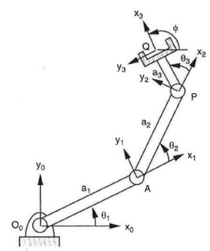
- Solving for $\theta_1$…
  – Corresponding to each $\theta_2$, we can solve $\theta_1$

$$(a_1 + a_2c\theta_2)c\theta_1 - (a_2s\theta_2)s\theta_1 = p_x$$
$$(a_2s\theta_2)c\theta_1 + (a_1 + a_2c\theta_2)s\theta_1 = p_y$$

$$c\theta_1 = \frac{p_x(a_1 + a_2c\theta_2) + p_y a_2 s\theta_2}{\Delta},$$
$$s\theta_1 = \frac{-p_x a_2 s\theta_2 + p_y(a_1 + a_2c\theta_2)}{\Delta}$$

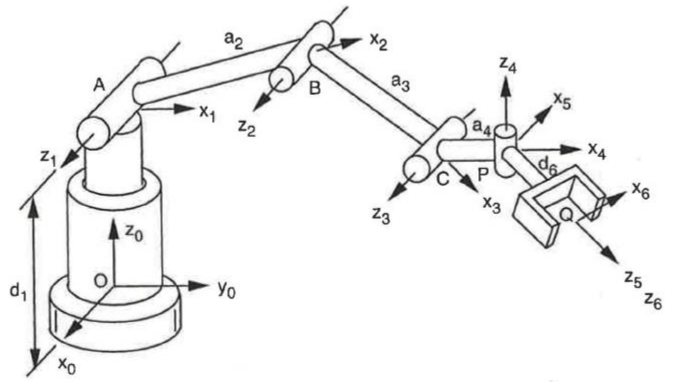
$$\Delta = a_1^2 + a_2^2 + 2a_1a_2c\theta_2$$

$$\theta_1 = \text{Atan2}(s\theta_1, c\theta_1).$$



# Inverse Kinematics: Example II

Elbow Manipulator:



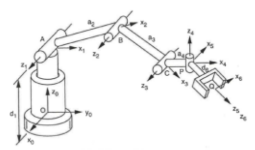



# Inverse Kinematics: Example II

- Target Position:

  $\mathbf{u} = [u_x, u_y, u_z]^{\mathrm{T}}$, $\mathbf{v} = [v_x, v_y, v_z]^{\mathrm{T}}$, $\mathbf{w} = [w_x, w_y, w_z]^{\mathrm{T}}$, and

  $\mathbf{p} = [p_x, p_y, p_z]^{\mathrm{T}}$.



- Transformation Matrices:

$$A_1 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad ({}^0A_1)^{-1} = \begin{bmatrix} c\theta_1 & s\theta_1 & 0 & 0 \\ -s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

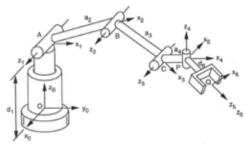
$$A_2 = \begin{bmatrix} c\theta_2 & 0 & -s\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ s\theta_2 & 0 & c\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} c\theta_3 & 0 & -s\theta_3 & a_2(1 - c\theta_3) \\ 0 & 1 & 0 & 0 \\ s\theta_3 & 0 & c\theta_3 & -a_2 s\theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 & (a_2 + a_3)(1 - c\theta_4) \\ 0 & 1 & 0 & 0 \\ s\theta_4 & 0 & c\theta_4 & -(a_2 + a_3)s\theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_5 = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & (a_2 + a_3 + a_4)(1 - c\theta_5) \\ s\theta_5 & c\theta_5 & 0 & -(a_2 + a_3 + a_4)s\theta_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## Inverse Kinematics: Example II

- Key Matrix Products:



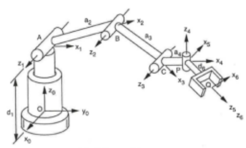
$$A_2A_3A_4 = \begin{bmatrix} c\theta_{234} & 0 & -s\theta_{234} & a_2c\theta_2 + a_3c\theta_{23} - (a_2+a_3)c\theta_{234} \\ 0 & 1 & 0 & 0 \\ s\theta_{234} & 0 & c\theta_{234} & a_2s\theta_2 + a_3s\theta_{23} - (a_2+a_3)s\theta_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1A_2A_3A_4 = \begin{bmatrix} c\theta_1c\theta_{234} & -s\theta_1 & -c\theta_1s\theta_{234} & c\theta_1[a_2c\theta_2 + a_3c\theta_{23} - (a_2+a_3)c\theta_{234}] \\ s\theta_1c\theta_{234} & c\theta_1 & -s\theta_1s\theta_{234} & s\theta_1[a_2c\theta_2 + a_3c\theta_{23} - (a_2+a_3)c\theta_{234}] \\ s\theta_{234} & 0 & c\theta_{234} & [a_2s\theta_2 + a_3s\theta_{23} - (a_2+a_3)s\theta_{234}] \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



## Inverse Kinematics: Example II

- Inverse Kinematics:



$$\mathbf{p} = A_1A_2A_3A_4\mathbf{p}_0.$$

$$A_1^{-1}\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = A_2A_3A_4\begin{bmatrix} a_2+a_3+a_4 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

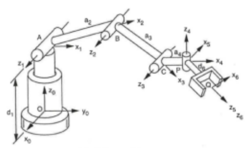
$$p_xc\theta_1 + p_ys\theta_1 = a_2c\theta_2 + a_3c\theta_{23} + a_4c\theta_{234},$$

$$-p_xs\theta_1 + p_yc\theta_1 = 0,$$

$$p_z = a_2s\theta_2 + a_3s\theta_{23} + a_4s\theta_{234}.$$

# Inverse Kinematics: Example II

- Solving the System:



$$\theta_1 = \tan^{-1}\frac{p_y}{p_x}.$$

$$\theta_5 = \sin^{-1}(-w_x s\theta_1 + w_y c\theta_1).$$

$$\theta_{234} = \text{Atan2}\left[w_z/c\theta_5, (w_x c\theta_1 + w_y s\theta_1)/c\theta_5\right].$$

$$a_2 c\theta_2 + a_3 c\theta_{23} = k_1, \qquad k_1 = p_x c\theta_1 + p_y s\theta_1 - a_4 c\theta_{234}$$

$$a_2 s\theta_2 + a_3 s\theta_{23} = k_2, \qquad k_2 = p_z - a_4 s\theta_{234}$$

$$a_2^2 + a_3^2 + 2a_2 a_3 c\theta_3 = k_1^2 + k_2^2.$$

$$\theta_3 = \cos^{-1}\frac{k_1^2 + k_2^2 - a_2^2 - a_3^2}{2a_2 a_3}.$$

$$\theta_6 = \text{Atan2}(s\theta_6, c\theta_6).$$



# Dynamics of Serial Manipulators

- Systems that keep on manipulating (the system)

- Direct Dynamics:
  - Find the response of a robot arm with torques/forces applied
- Inverse Dynamics:
  - Find the (actuator) torques/forces required to generate a desired trajectory of the manipulator

# Dynamics

- For Manipulators, the general form is

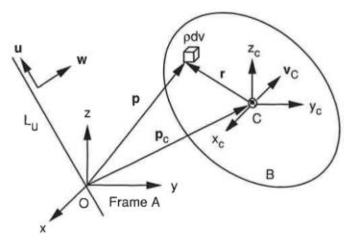
$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta)$$

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

- The moment of inertia (second moment) of a rigid body B relative to a line L that passes through a reference point O and is parallel to a unit vector **u** is given by:




$$I_u^O = \int_V p \times (u \times p)\, \rho dV$$
$$= \int_V \left[p^2 u - \left(p^T u\right) p\right] \rho dV$$

- The scalar product of $I^o_u$ with a second axis (**w**) is called the product of inertia

$$I_{uw}^O = I_u^O \cdot w = \int_V \left[\left(u^T w\right) p^2 - \left(p^T u\right)\left(p^T w\right)\right] \rho dV$$

- If u=w, then we get the moment of inertia:

$$I_{uu} = \int_V \left[p^2 - \left(p^T u\right)^2\right] \rho dV = m r_g^2$$

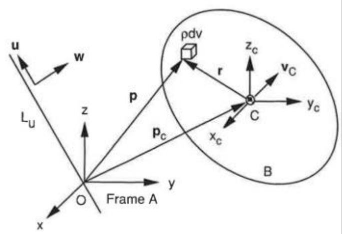
Where: $\mathbf{r_g}$: *radius of gyration* of B w/r/t to L

$$r_g = p^2 - \left(p^T u\right)^2 = (u \times p)^2$$

# Dynamics: Mass Matrix & Inertia Matrix

- This can be written in a Matrix form as:

$$\mathbf{I}_u^O = I_B^O \mathbf{u}$$




- Where $I^O_B$ is the inertial matrix or inertial tensor of the body B about a reference point O

$$I_B^O = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yz} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

- Where to get $I_{xx}$, etc? ➔ Parallel Axis Theorem

  If CM is the center of mass, then:

$$I_{xx}^O = I_{xx}^{CM} + m\left(y_c^2 + z_c^2\right) \qquad I_{xy}^O = I_{xx}^{CM} + m x_c y_c$$
$$I_{yy}^O = I_{yy}^{CM} + m\left(x_c^2 + z_c^2\right) \qquad I_{yz}^O = I_{xx}^{CM} + m y_c z_c$$
$$I_{zz}^O = I_{zz}^{CM} + m\left(x_c^2 + y_c^2\right) \qquad I_{zx}^O = I_{xx}^{CM} + m z_c x_c$$



# Dynamics: Mass Matrix

- The Mass Matrix: Determining via the Jacobian!

$$K = \sum_{i=1}^{N} K_i$$

$$K_i = \frac{1}{2}\left(m_i v_{C_i}^T v_{C_i} + \omega_i^T I_{C_i} \omega_i\right)$$

$$v_{C_i} = J_{v_i}\dot{\theta} \qquad J_{v_i} = \left[\begin{array}{cccccc} \frac{\partial \mathbf{p}_{C_1}}{\partial \theta_1} & \cdots & \frac{\partial \mathbf{p}_{C_i}}{\partial \theta_i} & \underbrace{0}_{i+1} & \cdots & \underbrace{0}_{n} \end{array}\right]$$

$$\omega_i = J_{\omega_i}\dot{\theta} \qquad J_{\omega_i} = \left[\begin{array}{cccccc} \bar{\varepsilon}_1 Z_1 & \cdots & \bar{\varepsilon}_i Z_i & \underbrace{0}_{i+1} & \cdots & \underbrace{0}_{n} \end{array}\right]$$

$$\therefore M = \sum_{i=1}^{N}\left(m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^T I_{C_i} J_{\omega_i}\right)$$

! M is symmetric, positive definite $\therefore m_{ij} = m_{ji}, \dot{\boldsymbol{\theta}}^T M \dot{\boldsymbol{\theta}} > 0$

## 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

$$L = K - P$$

$$\mathbf{F} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{x}}}\right) - \frac{\partial L}{\partial \mathbf{x}}$$

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta}$$



## Dynamics – Langrangian Mechanics [2]

$L = K - P, \dot{\boldsymbol{\theta}}$ : Generalized Velocities, $M$ : Mass Matrix

$$\boldsymbol{\tau} = \sum_{i=1}^{N} \tau_i = \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\boldsymbol{\theta}}}\right) - \frac{\partial K}{\partial \boldsymbol{\theta}} + \frac{\partial P}{\partial \boldsymbol{\theta}}$$

$$K = \tfrac{1}{2}\dot{\theta}^T M(\theta)\,\dot{\theta}$$

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}}\right) = \frac{d}{dt}\left(\frac{\partial}{\partial \dot{\theta}}\left(\tfrac{1}{2}\dot{\theta}^T M(\theta)\,\dot{\theta}\right)\right) = \frac{d}{dt}\left(M\dot{\theta}\right) = M\ddot{\theta} + \dot{M}\dot{\theta}$$

$$\rightarrow \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}}\right) - \frac{\partial K}{\partial \theta} = \left[M\ddot{\theta} + \dot{M}\dot{\theta}\right] - \left[\tfrac{1}{2}\dot{\theta}^T M(\theta)\,\dot{\theta}\right] = M\ddot{\theta} + \underbrace{\left\{\dot{M}\dot{\theta} - \tfrac{1}{2}\begin{bmatrix}\dot{\theta}^T \frac{\partial M}{\partial \theta_1}\dot{\theta} \\ \vdots \\ \dot{\theta}^T \frac{\partial M}{\partial \theta_n}\dot{\theta}\end{bmatrix}\right\}}_{\mathbf{v}(\theta,\dot{\theta})}$$

$$\mathbf{v}\left(\theta,\dot{\theta}\right) = \underbrace{C(\theta)\left[\dot{\theta}^2\right]}_{\text{Centrifugal}} + \underbrace{B(\theta)\left[\dot{\theta}\dot{\theta}\right]}_{\text{Coriolis}}$$

$$\Rightarrow \boldsymbol{\tau} = M(\theta)\,\ddot{\boldsymbol{\theta}} + \mathbf{v}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right) + \mathbf{g}(\boldsymbol{\theta})$$

## Dynamics – Langrangian Mechanics [3]

- The Mass Matrix: Determining via the Jacobian!

$$K = \sum_{i=1}^{N} K_i$$

$$K_i = \frac{1}{2}\left(m_i v_{C_i}^T v_{C_i} + \omega_i^T I_{C_i} \omega_i\right)$$

$$v_{C_i} = J_{v_i}\dot{\theta} \qquad J_{v_i} = \begin{bmatrix} \frac{\partial \mathbf{p}_{C_1}}{\partial \theta_1} & \cdots & \frac{\partial \mathbf{p}_{C_i}}{\partial \theta_i} & \underbrace{0}_{i+1} & \cdots & \underbrace{0}_{n} \end{bmatrix}$$

$$\omega_i = J_{\omega_i}\dot{\theta} \qquad J_{\omega_i} = \begin{bmatrix} \bar{\varepsilon}_1 Z_1 & \cdots & \bar{\varepsilon}_i Z_i & \underbrace{0}_{i+1} & \cdots & \underbrace{0}_{n} \end{bmatrix}$$

$$\therefore M = \sum_{i=1}^{N}\left(m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^T I_{C_i} J_{\omega_i}\right)$$

! M is symmetric, positive definite $\therefore m_{ij} = m_{ji}, \dot{\boldsymbol{\theta}}^T M \dot{\boldsymbol{\theta}} > 0$



## Generalized Coordinates

- A significant feature of the Lagrangian Formulation is that any convenient coordinates can be used to derive the system.
- Go from Joint → Generalized
  - Define **p**: $d\mathbf{p} = \mathbf{J}d\mathbf{q}$

$$\mathbf{q} = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix} \rightarrow \mathbf{p} = \begin{bmatrix} p_1 & \dots & p_n \end{bmatrix}$$

➔Thus: the kinetic energy and gravity terms become

$$KE = \tfrac{1}{2}\dot{\mathbf{p}}^T \mathbf{H}^* \dot{\mathbf{p}} \qquad \mathbf{G}^* = \left(\mathbf{J}^{-1}\right)^T \mathbf{G}$$

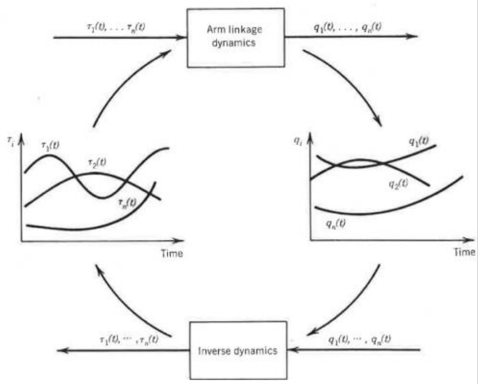
where: $\mathbf{H}^* = \left(\mathbf{J}^{-1}\right)^T \mathbf{H}\mathbf{J}^{-1}$

# Inverse Dynamics

- Forward dynamics governs the dynamic responses of a manipulator arm to the input torques generated by the actuators.
- The inverse problem:
  - Going from joint angles to torques
  - Inputs are desired trajectories described as functions of time

    $\mathbf{q} = [q_1 \quad \dots \quad q_n] \rightarrow [\theta_1(t) \quad \theta_2(t) \quad \theta_3(t)]$
  - Outputs are joint torques to be applied at each instance

    $\boldsymbol{\tau} = [\tau_1 \quad \dots \quad \tau_n]$
- Computation "big" (6DOF arm: 66,271 multiplications), but not scary (4.5 ms on PDP11/45)




Graphic from Asada & Slotine p. 119



# Also: 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:

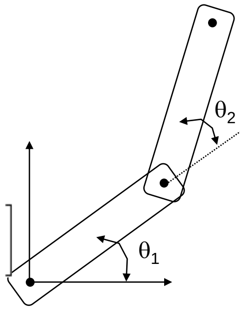
$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

- 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 $\theta_2$ approaches 0 or $\pi$ 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

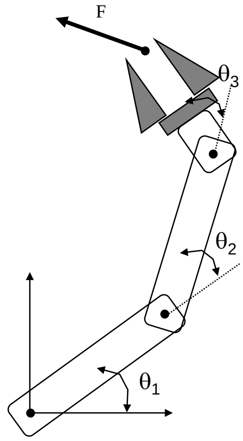
$$\delta\mathbf{X} = \mathbf{J}\delta\theta$$

- So that

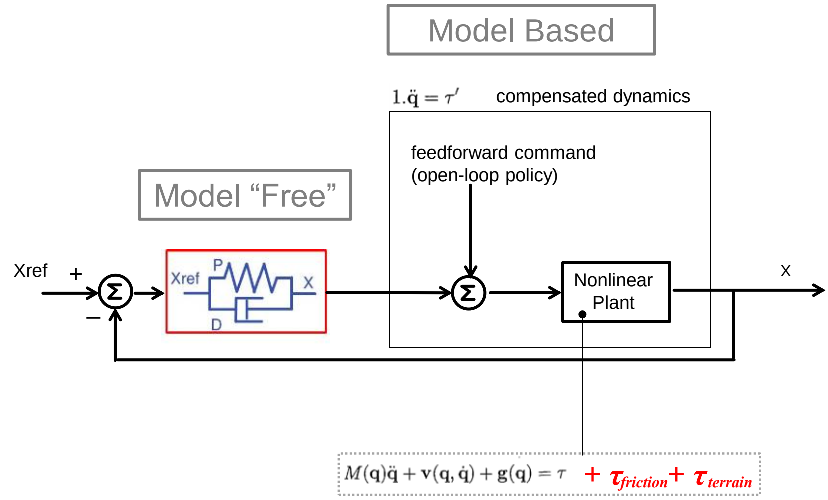
$$F^T \mathbf{J} = \tau^T$$

- Or

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



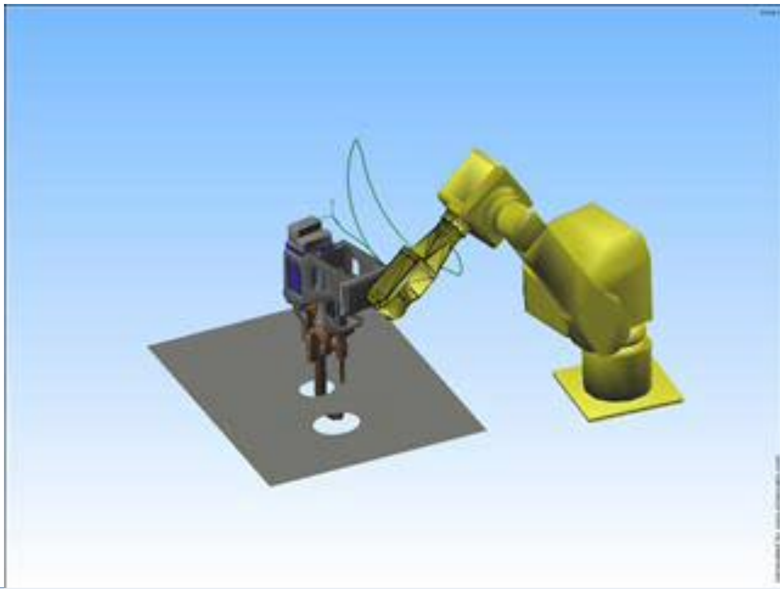
# Operation Space (Computed Torque)






# Compensated Manipulation

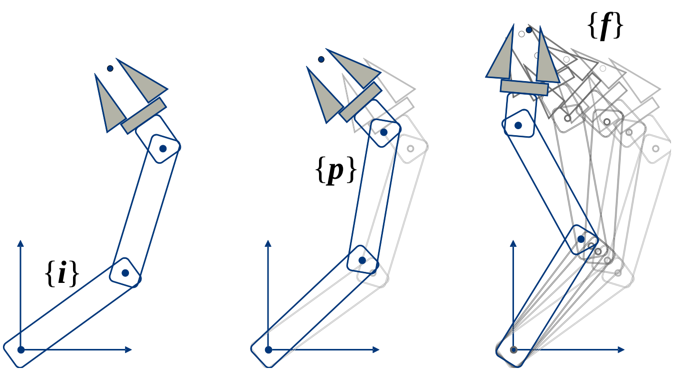
## Trajectory Generation & Planning





## Trajectory Generation

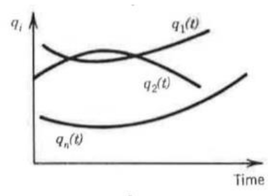
- The goal is to get from an initial position $\{\boldsymbol{i}\}$ to a final position $\{\boldsymbol{f}\}$ via a path points $\{\boldsymbol{p}\}$



## Joint Space

Consider only the **joint positions** as a function of time

- + Since we control the joints, this is more direct
- -- If we want to follow a particular trajectory, not easy
  - at best lots of intermediate points
  - No guarantee that you can solve the Inverse Kinematics for all path points

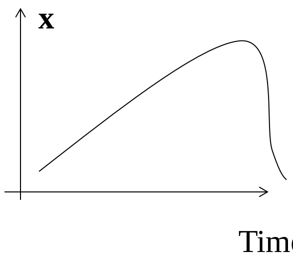





## Cartesian Workspace

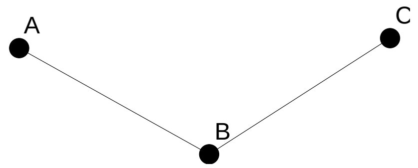
Consider the **Cartesian positions** as a function of time

- + Can track shapes exactly
- -- We need to solve the inverse kinematics and dynamics



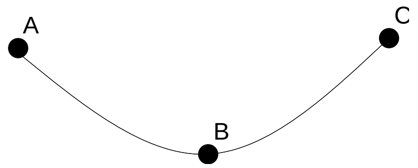
## Polynomial Trajectories

- Straight line Trajectories




- Simpler

- Polynomial Trajectories




$$u(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

- Parabolic blends are smoother
- Use “pseudo via points”



## 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




# Dynamic Simulation Software

- v-rep virtual robot experimentation platform
- Reflexxes

http://www.coppeliarobotics.com/ http://www.reflexxes.com/