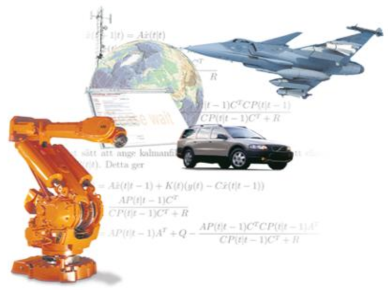
Robot control



Lecture 4
Mikael Norrlöf

Substantial contribution by Stig Moberg

Up till now

- Lecture 1
  - Rigid body motion
  - Representation of rotation
  - Homogenous transformation
- Lecture 2
  - Kinematics
    - Position
    - Velocity via Jacobian
    - DH parameterization
- Lecture 3
  - Dynamics
  - Lagrange's equation
  - Newton Euler

Next lecture 15/1
10-12 Algoritmen

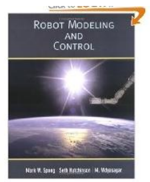
Robot control - examples

- Fanta challenge
  http://www.youtube.com/watch?v=PSKdHsqtok0

- Fanta challenge II
  http://www.youtube.com/watch?v=SOESSCXGhFo

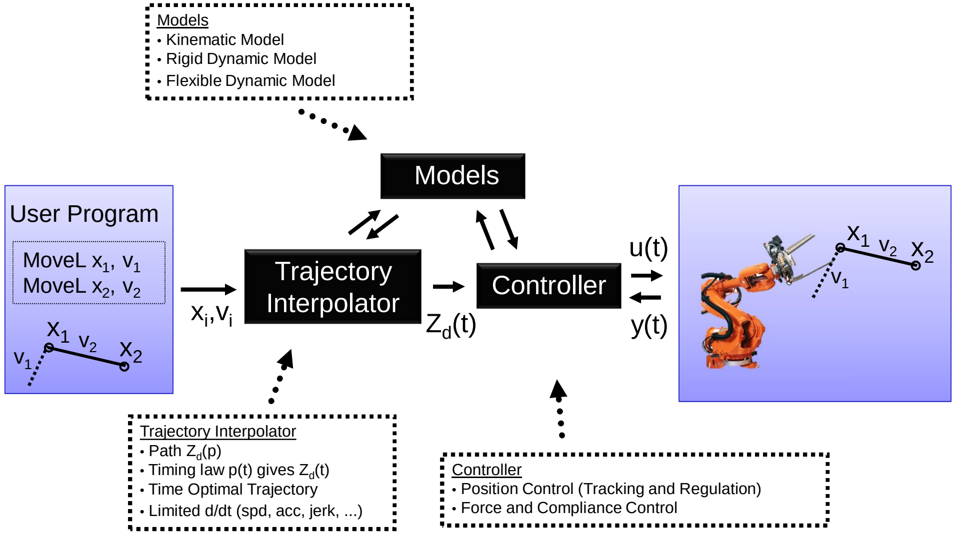
Outline

- Robot Motion Control Overview
- Current and Torque Control
- Control Methods for Rigid Robots
- Control Methods for Flexible Robots
- Interaction with the environment

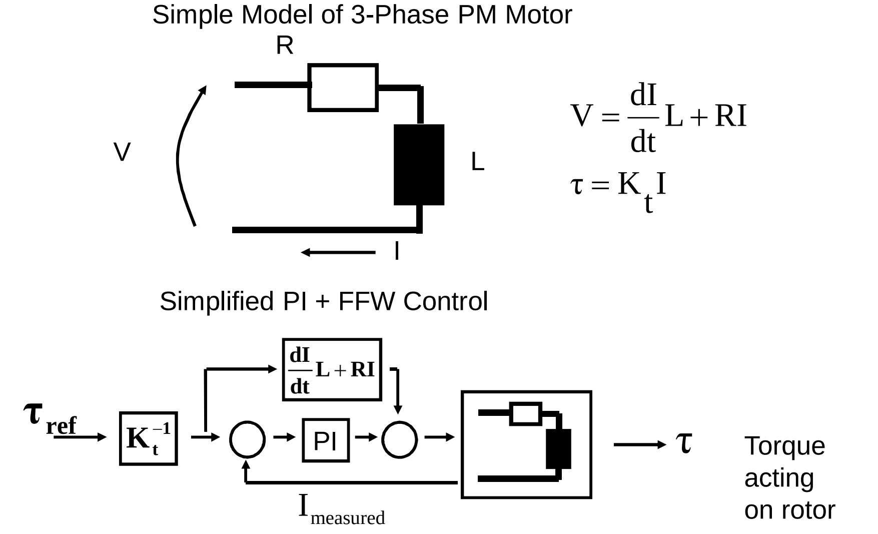


Chapters 6, 8, 9

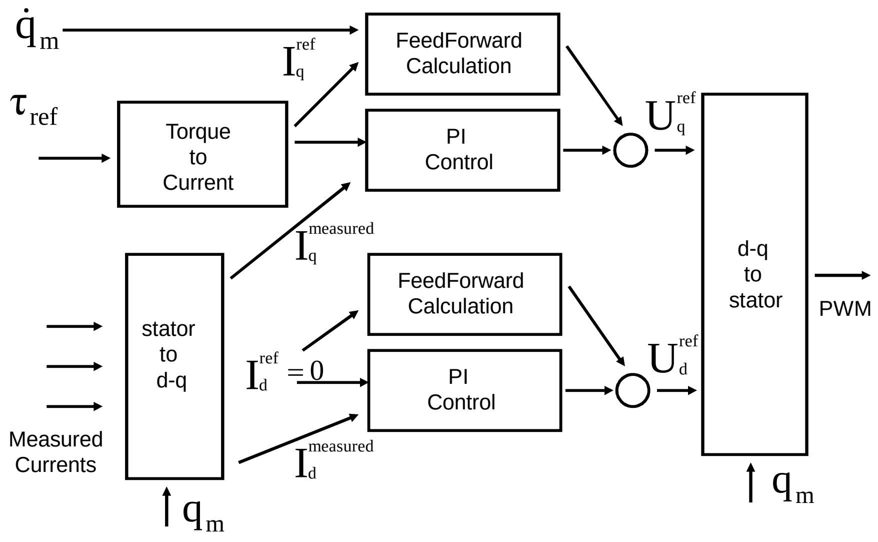
Robot motion control



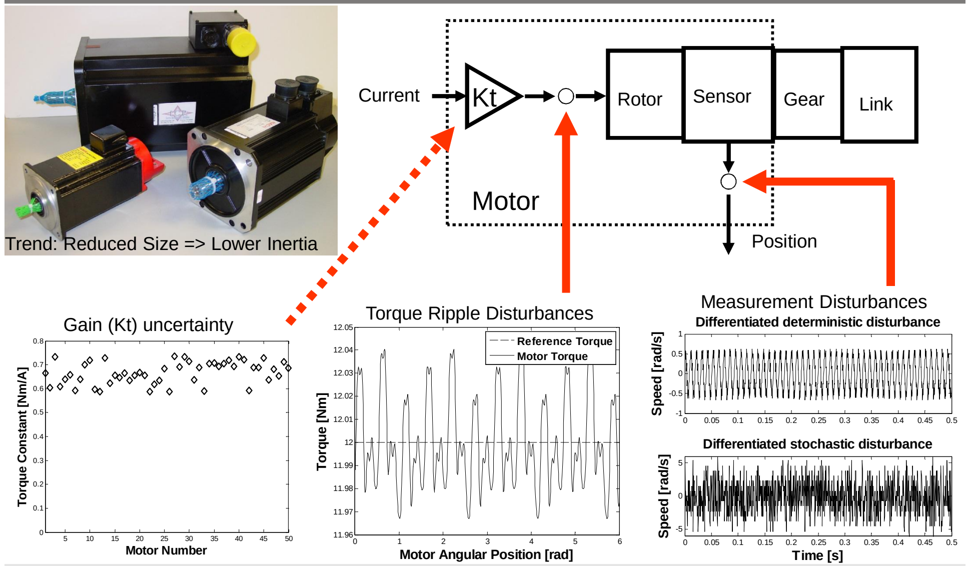
Torque and Current Control



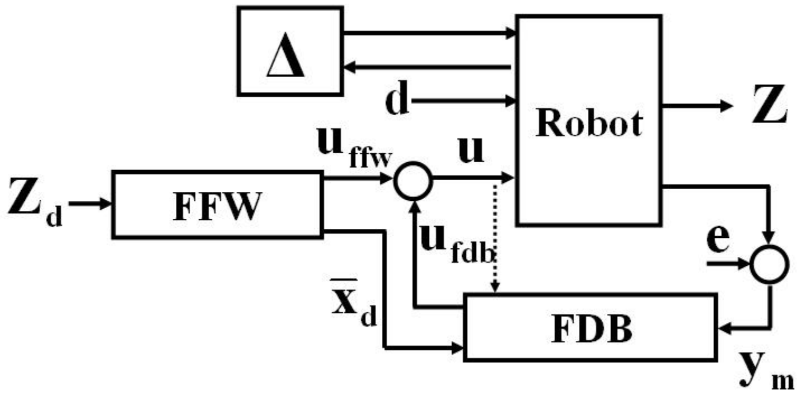
Torque and Current Control



Torque and Current Control

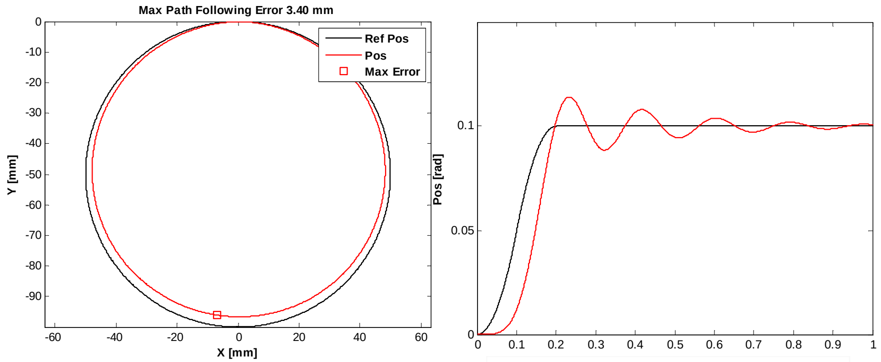


The Robot Control Problem



The user-specified path Z_d must be followed with specified precision even under the influence of different uncertainties. These uncertainties are disturbances acting on the robot and on the measurements, as well as uncertainties in the models used by the motion control system.

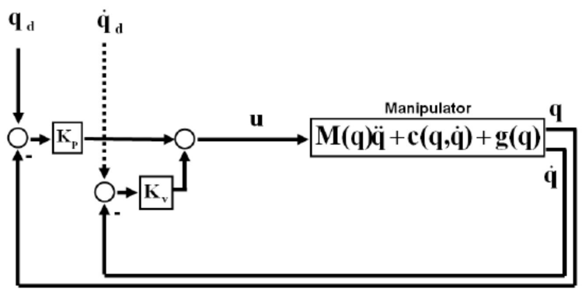
The Robot Control Problem



Typical Requirements:
- Settling Time 0.1 s
- Path Error < 0.1 mm @ 20 mm/s
- Path Error 1 mm @ 1000 mm/s
- Speed Accuracy 5 %
- Absolute Accuracy 1 mm
- Repeatability < 0.1 mm

Independent Joint PD Control

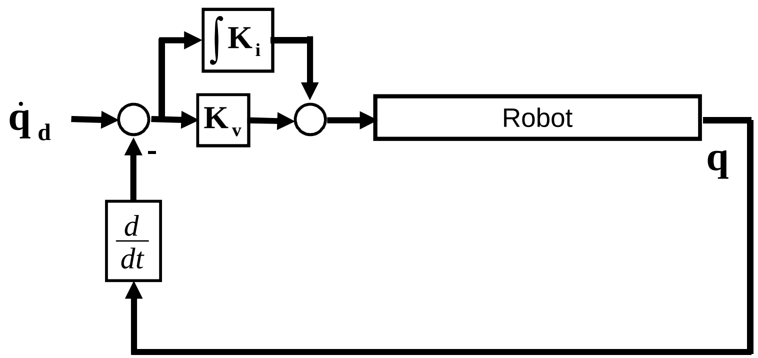
$$u = K_p(q_d - q) + K_v(\dot{q}_d - \dot{q})$$



- Integral part added in most cases
- Speed normally not measured, estimated from position
- D-part on reference can be removed (gives overshoot if I-part is added!)

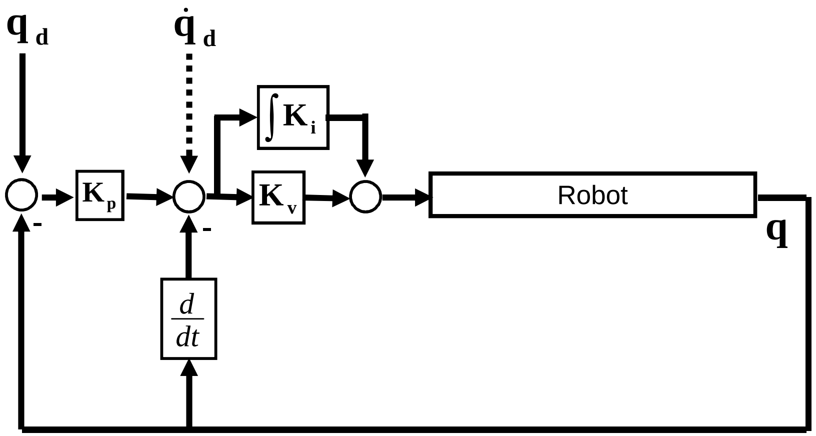
Cascade Controller

Inner Loop: Speed (PI) Controller



Cascade Controller

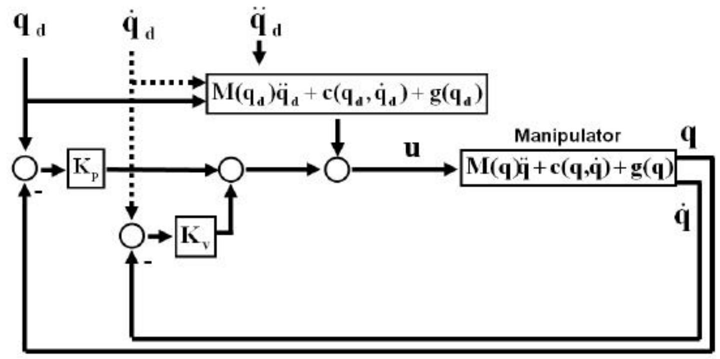
Outer Loop: Position Controller (P) – Inner Loop: Speed (PI) Controller



Feedforward + PD Control

$$u_{ffw} = M(q_d)\ddot{q}_d + c(q_d, \dot{q}_d) + g(q_d)$$

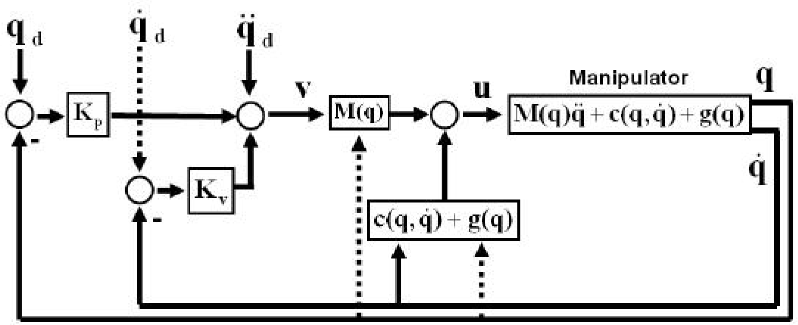
$$u = u_{ffw} + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q})$$



Feedback Linearization + PD Control

$$v = \ddot{q}_d + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q})$$

$$u = M(q)v + c(q, \dot{q}) + g(q)$$

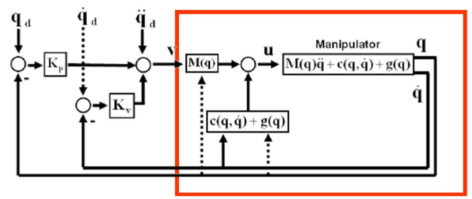


Feedback Linearization + PD Control

$$\mathrm{M(q)\ddot{q} + c(q,\dot{q}) + g(q) = u}$$

$$\mathrm{\hat{M}(q)v + \hat{c}(q,\dot{q}) + \hat{g}(q) = u}$$

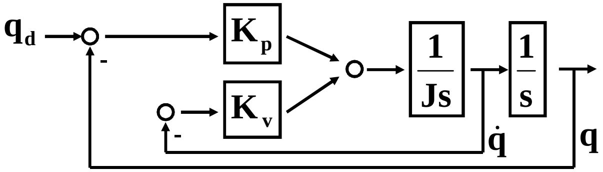
$$(\hat{*}) = (*) \Rightarrow \ddot{q} = v \Rightarrow q = \frac{1}{s^2}v$$

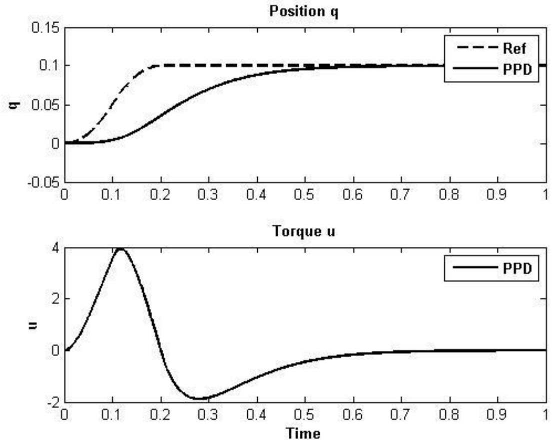


Decoupled system of double integrators!

Feedback Linearization is sometimes called Computed Torque or Inverse Dynamics, so is Feedforward Control!

Simulation Example – Simple Robot



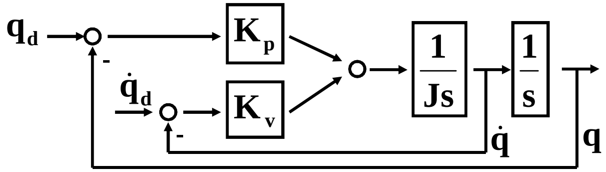


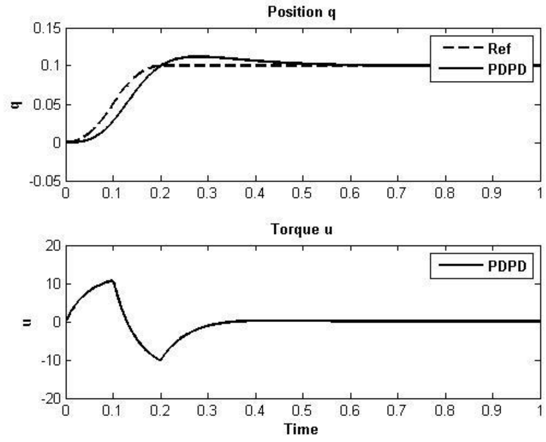
$J = 1$
$Kp = 158$
$Kv = 25$

Closed Loop
Poles -12.6, -12.6

i.e. damping = 1
=> no overshoot

Simulation Example – Simple Robot

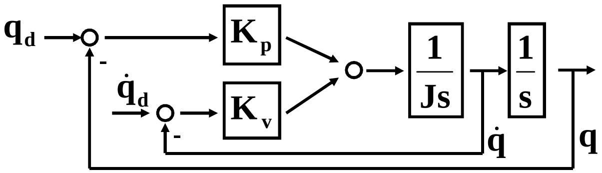


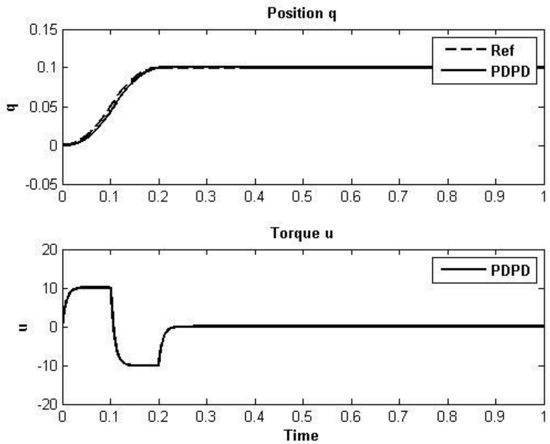


$J = 1$
$Kp = 158$
$Kv = 25$

Closed Loop Poles
-12.6, -12.6
Zero -6

Simulation Example – Simple Robot

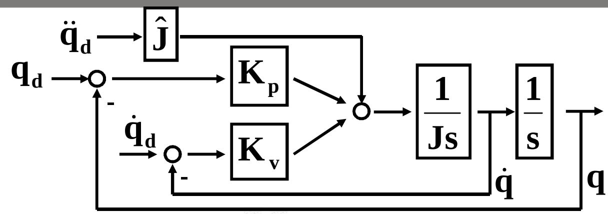


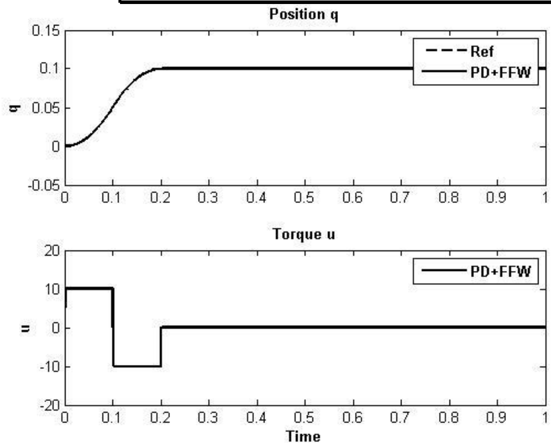


$J = 1$
$Kp = 158$
$Kv = 126$

Closed Loop
Poles -124, -1.3
Zero -1.3

Simulation Example – Simple Robot

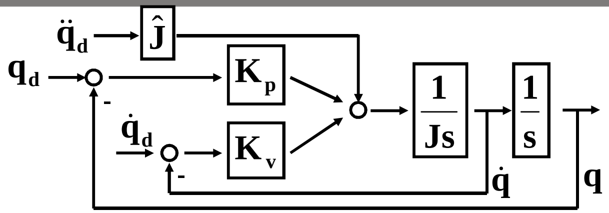


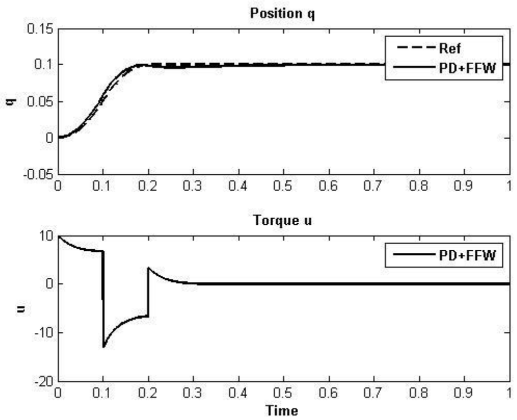


$J = 1$
$\hat{J} = 1$
$Kp = 158$
$Kv = 25$

”2-DOF” Controller:
Tracking &
Regulation/Robustness
”decoupled”

Simulation Example – Simple Robot





$\mathbf{J = 0.7}$
$\hat{\mathbf{J}} = \mathbf{1}$
$\mathbf{Kp = 158}$
$\mathbf{Kv = 25}$

Robustification of Feedback Linearization

Robust outer loop design (Second Method of Lyapunov)

$$M(q)\ddot{q} + c(q,\dot{q}) + g(q) = u$$

$$\hat{M}(q)a_q + \hat{c}(q,\dot{q}) + \hat{g}(q) = u$$

$$e = q_d - q$$

$$a_q = \ddot{q}_d + K_p e + K_v \dot{e}$$

$$(\hat{*}) \neq (*) \Rightarrow \ddot{q} = a_q + \eta(v,e,\dot{e})$$

Worst case estimation of η => discontinuous control term Δa can be added to outer loop control:

$$a_q = \ddot{q}_d + K_p e + K_v \dot{e} + \Delta a(e,\dot{e},t)$$

- High sample rate required
- Must know max acceleration & worst case error in M, c, and g
- Approximate continuous version

Passivity Based Robust Control

Control Input:

$$u = \hat{M}(q)a + \hat{c}(q,\dot{q})v + \hat{g}(q) + Kr$$

$$v = \dot{q}_d + \Lambda(q_d - q)$$

$$a = \dot{v} = \ddot{q}_d + \Lambda(\dot{q}_d - \dot{q})$$

$$r = \dot{q}_d - \dot{q} + \Lambda(q_d - q)$$

- K and Λ are diagonal matrices.
- Closed loop system is coupled and nonlinear
- Linear parametrization of dynamic model is used
- Bound ρ does not depend on trajectory or state

$$\|\theta_0 - \theta\| \leq \rho$$

$$u = Y(q,\dot{q},a,v)(\theta_0 + \delta\theta) + Kr$$

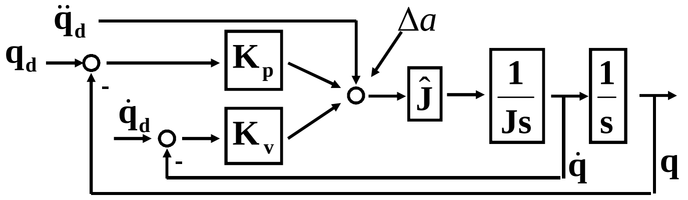
$$\delta\theta = \begin{cases} \rho \dfrac{Y^T r}{\|Y^T r\|}; Y^T r > \varepsilon \\ \rho \dfrac{Y^T r}{\varepsilon}; Y^T r \leq \varepsilon \end{cases}$$

Passivity Based Adaptive Control

Control Input: $u = Y(q,\dot{q},a,v)\hat{\theta} + Kr$

Parameter update: $\dot{\hat{\theta}} = -\Gamma^{-1} Y^T(q,\dot{q},v,a) r$

Robustified Feedback Linearization (Lyap 2)



$$w = p_{21}e + p_{22}e$$

$J = 0.7$

$\hat{J} = 1$

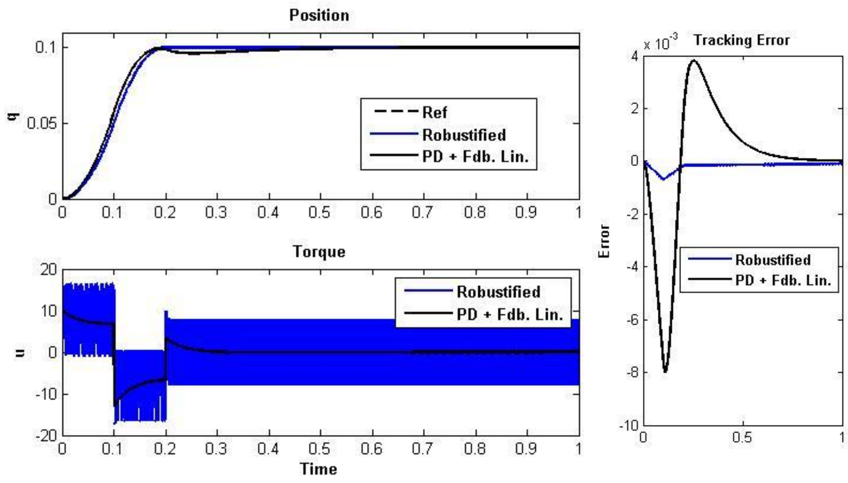
$Kp = 158$

$Kv = 25$

$$\Delta a = \begin{cases} -(\gamma_1 + \gamma_1\sqrt{e^2 + \dot{e}^2})\frac{w}{\|w\|} & ;\|w\| \neq 0 \\ 0 & ;\|w\| = 0 \end{cases}$$

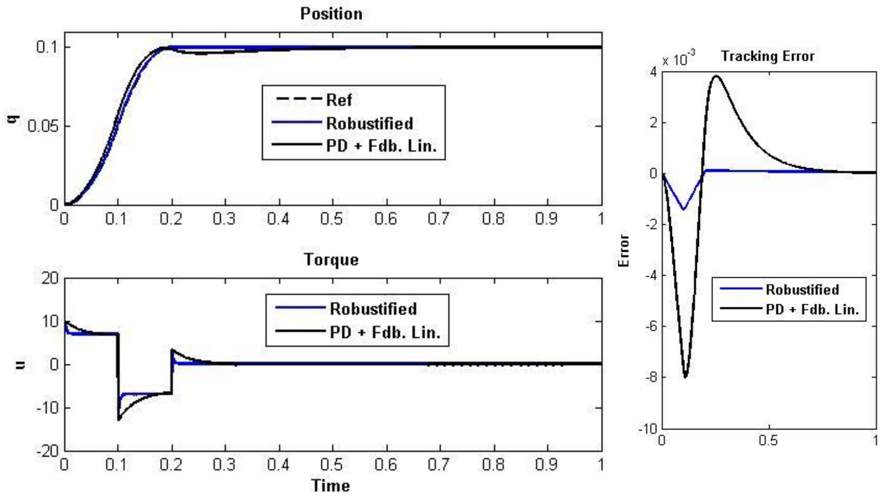
- p_{21}, p_{22}, γ_1, γ_2, computed from J uncertainty, K_v, K_p, and $\ddot{q}_d^{max}$
- Lyapunov equation => p_{21}, p_{22}

Robustified Feedback Linearization (Lyap 2)



Discontinuous Δa

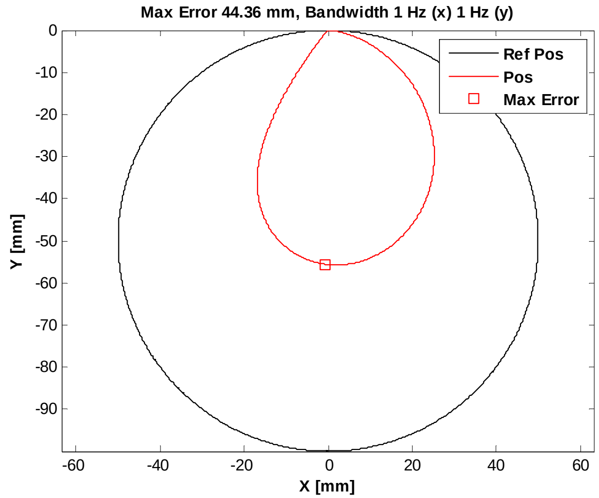
Robustified Feedback Linearization (Lyap 2)



Continuous Δa

Path Accuracy: Circle r = 50 mm @ 1 s

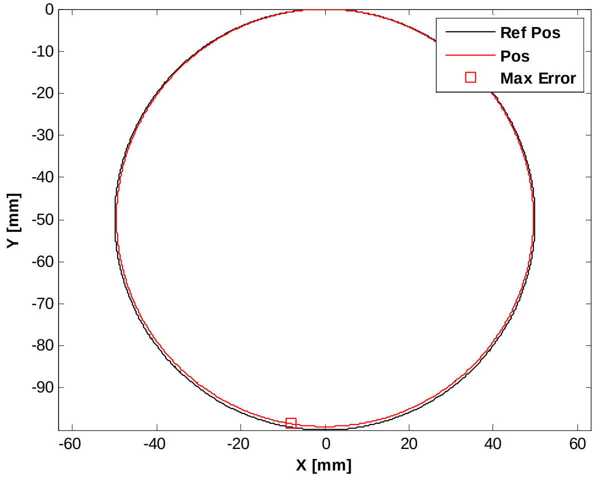
XY rigid robot with PD (fdb) P (ref) control



Path Accuracy: Circle r = 50 mm @ 1 s

XY rigid robot with PD (fdb) P (ref) control

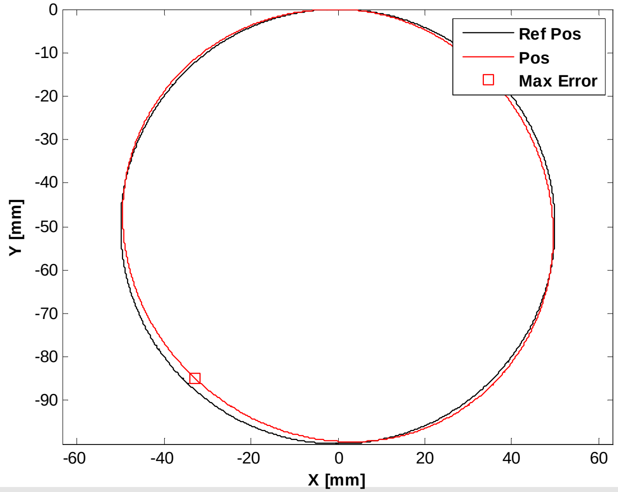
Max Error 0.84 mm, Bandwidth 15 Hz (x) 15 Hz (y)



Path Accuracy: Circle r = 50 mm @ 1 s

XY rigid robot with PD (fdb) P (ref) control

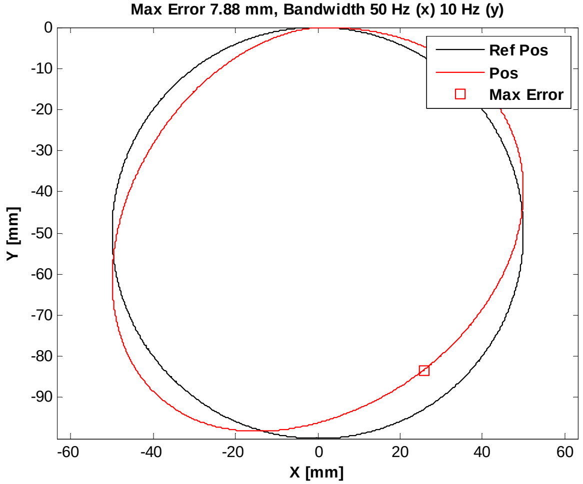
Max Error 2.01 mm, Bandwidth 15 Hz (x) 20 Hz (y)



Path Accuracy: Circle r = 50 mm @ 1 s

XY rigid robot with PD (fdb) P (ref) control

Max Error 7.88 mm, Bandwidth 50 Hz (x) 10 Hz (y)



Bandwidth must be sufficiently high relative frequency content of trajectory

All channels should be equal!

Flexible Joint Model

Complete Model:

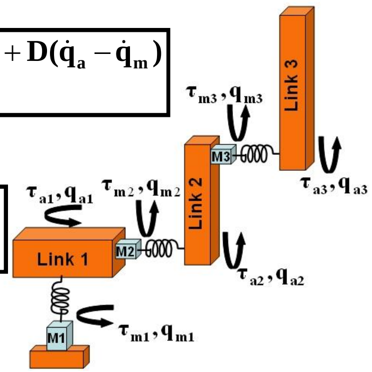
$$\begin{aligned}0 &= \mathbf{M}_a(\mathbf{q}_a)\ddot{\mathbf{q}}_a + \mathbf{S}\ddot{\mathbf{q}}_m + \mathbf{c}_1(\mathbf{q}_a,\dot{\mathbf{q}}_a,\dot{\mathbf{q}}_m) + \mathbf{g}(\mathbf{q}_a) + \mathbf{K}(\mathbf{q}_a - \mathbf{q}_m) + \mathbf{D}(\dot{\mathbf{q}}_a - \dot{\mathbf{q}}_m)\\ \boldsymbol{\tau} &= \mathbf{M}_m\ddot{\mathbf{q}}_m + \mathbf{S}^T\ddot{\mathbf{q}}_m + \mathbf{c}_2(\mathbf{q}_a,\dot{\mathbf{q}}_a) - \mathbf{K}(\mathbf{q}_a - \mathbf{q}_m) - \mathbf{D}(\dot{\mathbf{q}}_a - \dot{\mathbf{q}}_m) + \mathbf{f}(\dot{\mathbf{q}}_m)\end{aligned}$$

Simplified Model I (high gear ratio):

$$\begin{aligned}0 &= \mathbf{M}_a(\mathbf{q}_a)\ddot{\mathbf{q}}_a + \mathbf{c}(\mathbf{q}_a,\dot{\mathbf{q}}_a) + \mathbf{g}(\mathbf{q}_a) + \mathbf{K}(\mathbf{q}_a - \mathbf{q}_m) + \mathbf{D}(\dot{\mathbf{q}}_a - \dot{\mathbf{q}}_m)\\ \boldsymbol{\tau} &= \mathbf{M}_m\ddot{\mathbf{q}}_m - \mathbf{K}(\mathbf{q}_a - \mathbf{q}_m) - \mathbf{D}(\dot{\mathbf{q}}_a - \dot{\mathbf{q}}_m) + \mathbf{f}(\dot{\mathbf{q}}_m)\end{aligned}$$

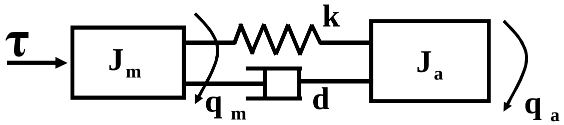
Simplified Model II (friction and damping low):

$$\begin{aligned}0 &= \mathbf{M}_a(\mathbf{q}_a)\ddot{\mathbf{q}}_a + \mathbf{c}(\mathbf{q}_a,\dot{\mathbf{q}}_a) + \mathbf{g}(\mathbf{q}_a) + \mathbf{K}(\mathbf{q}_a - \mathbf{q}_m)\\ \boldsymbol{\tau} &= \mathbf{M}_m\ddot{\mathbf{q}}_m - \mathbf{K}(\mathbf{q}_a - \mathbf{q}_m)\end{aligned}$$



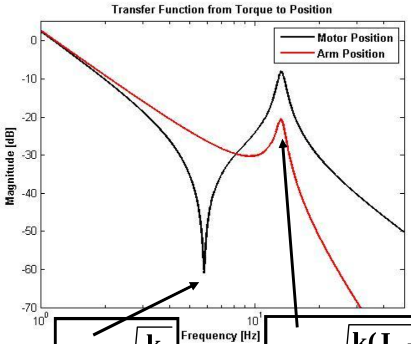
For N links and N motors: 2N d.o.f.

Linear One Axis Model



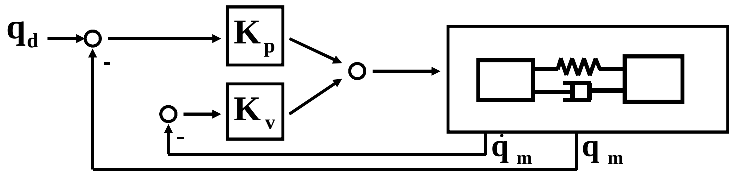
$$G_{\tau \to q_m} = \frac{\frac{s^2}{\omega_z^2} + \frac{2\varsigma_z s}{\omega_z} + 1}{s^2 (J_a + J_m)\left(\frac{s^2}{\omega_p^2} + \frac{2\varsigma_p s}{\omega_p} + 1\right)}$$

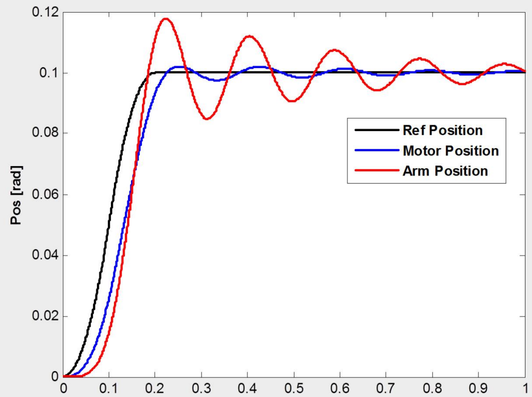
$$G_{\tau \to q_a} = \frac{sd + k}{s^2 (J_a + J_m)\left(\frac{s^2}{\omega_p^2} + \frac{2\varsigma_p s}{\omega_p} + 1\right)}$$



$$\varsigma_z = \frac{d}{2}\sqrt{\frac{1}{kJ_a}} \qquad \varsigma_p = \frac{d}{2}\sqrt{\frac{J_a + J_m}{kJ_aJ_m}} \qquad \omega_z = \sqrt{\frac{k}{J_a}} \qquad \omega_p = \sqrt{\frac{k(J_a + J_m)}{J_aJ_m}}$$

PD Control

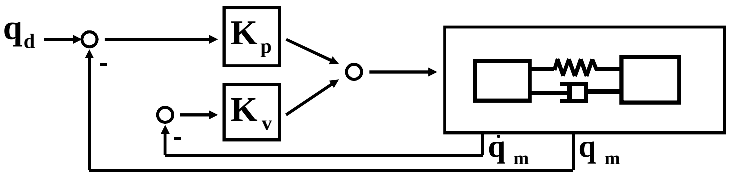


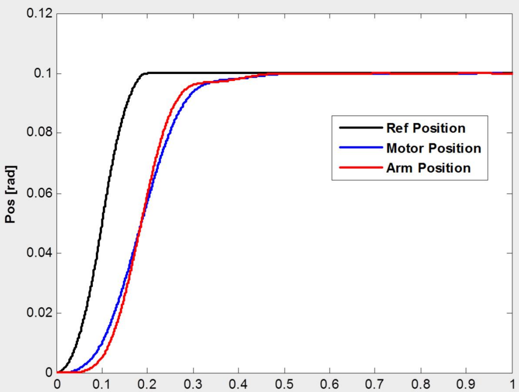


Closed Loop Bandwidth:
10 Hz
Mechanical Resonance:
6 Hz

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

PD Control – Motor Feedback

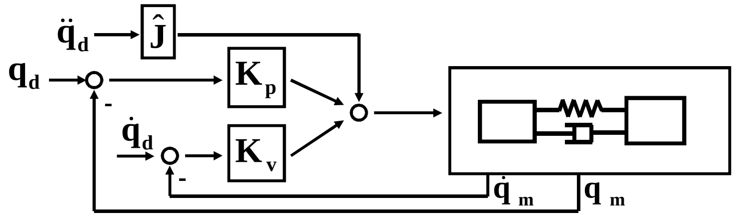


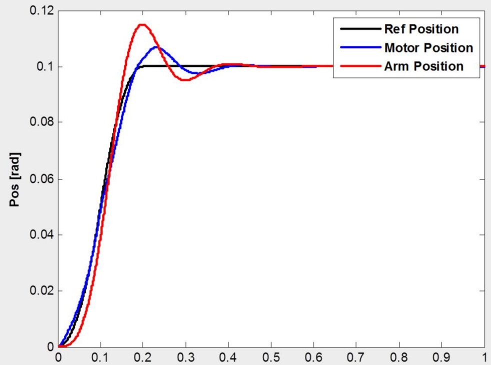


Closed Loop Bandwidth:
3.5 Hz
Mechanical Resonance:
6 Hz

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

PD + FFW Control – Motor Feedback

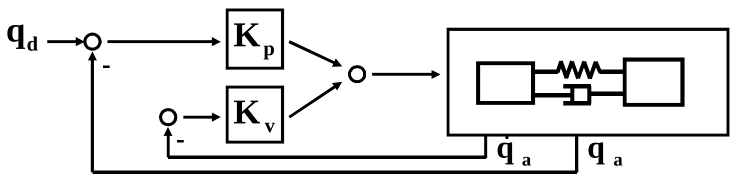


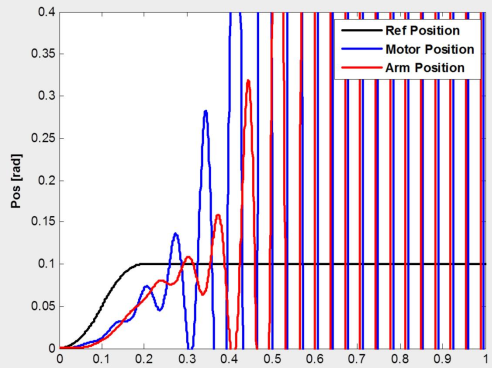


Same PD Tuning as on
previous slide
(Poles @ 3.5 Hz)

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

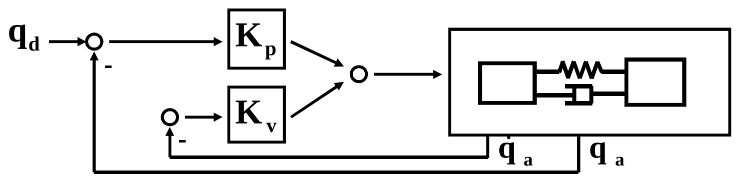
PD Control – Arm Feedback

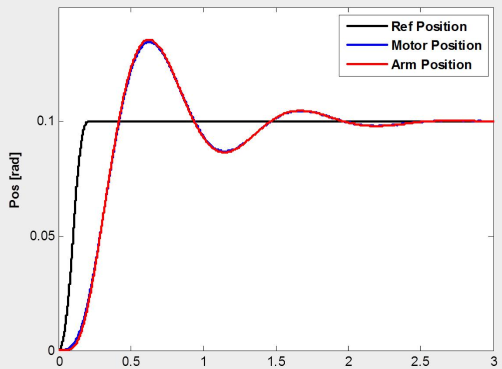




Same PD Tuning as on previous slide => unstable

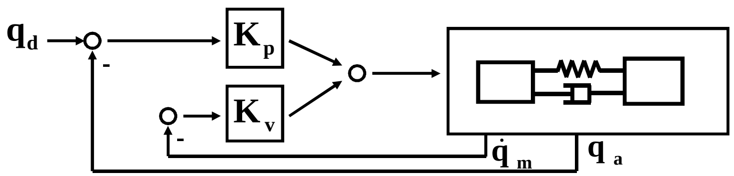
PD Control – Arm Feedback

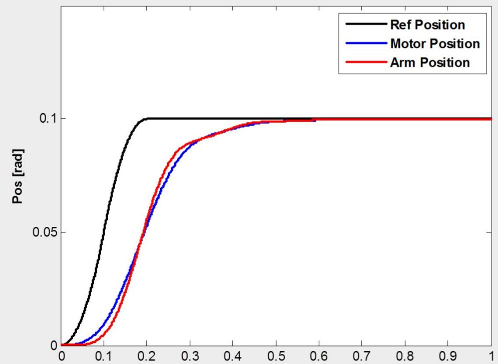




Low Gain Controller

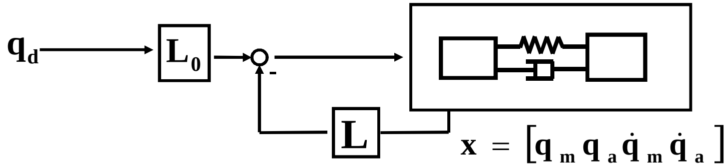
PD Control – Arm Pos / Motor Speed Feedback

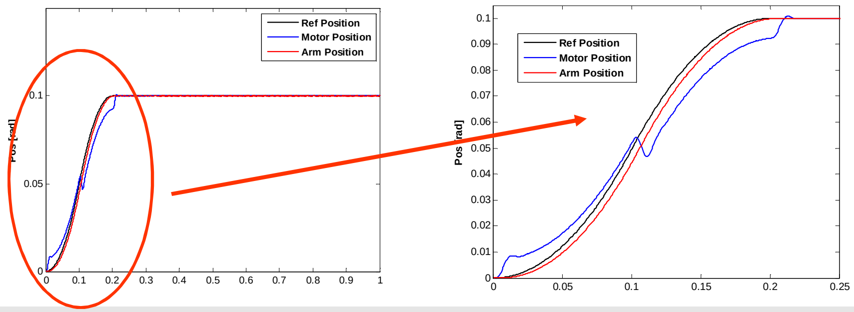




Motor Feedback preferred for flexible joint robot if only two states are measured

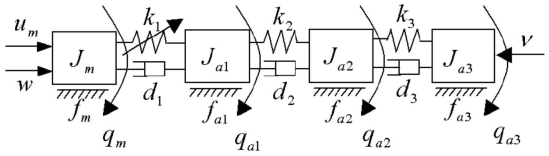
LQ Control / State Feedback – Full State Measurement



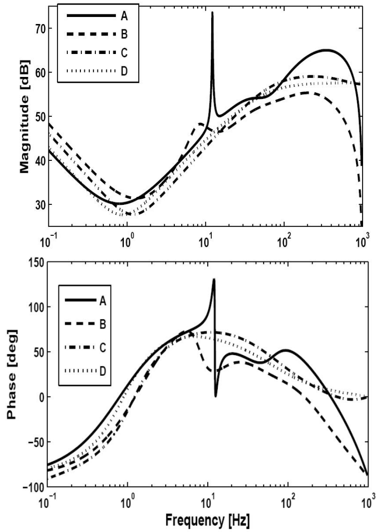


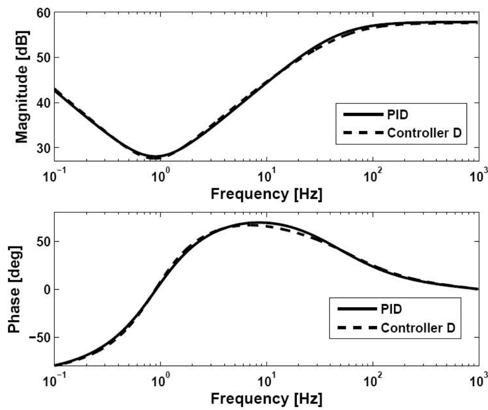
Control of a flexible robot arm: A benchmark problem

- What is the limit for control using motor measurements only?
- Design of a robust digital controller for optimal disturbance rejection
- One axis uncertain model of an industrial robot



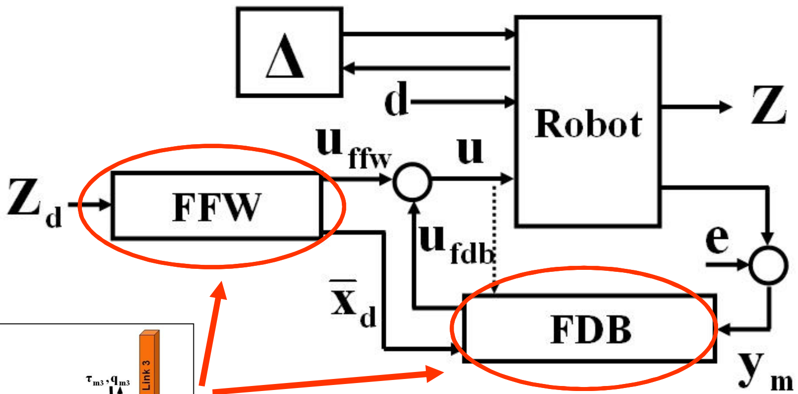
Swedish Open Championships in Robot Control 2005

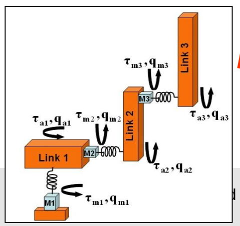




www.robustcontrol.org

Nonlinear MIMO Flexible Joint Control



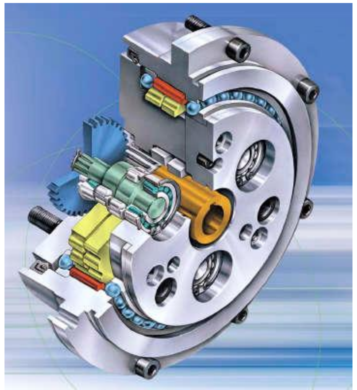


Goal: Perfect tracking, i.e $\mathbf{Z(t) = Z_d(t)}$

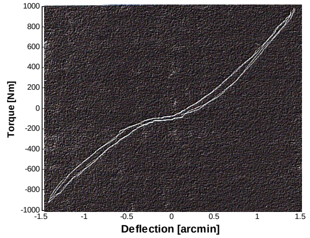
Suggested Control Methods

Feedback Linearization
Passivity-Based Control
Backstepping
Adaptive Control
Neural Networks
Singular Perturbations
Composite Control
Input Shaping
Robust Control based on Lyaponov 2nd Method
Sliding Mode Control
Iterative Learning Control
Feedforward Control
Linear MIMO Control (Pole Placement, LQG, H infinity, …)
Linear Diagonal Control (PID, Pole Placement, …)
…
…

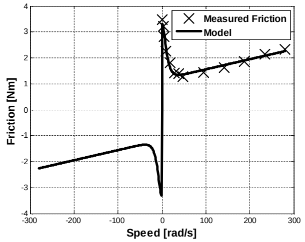
Compact Gear Transmission



Elasticity with hysteresis in rotational direction

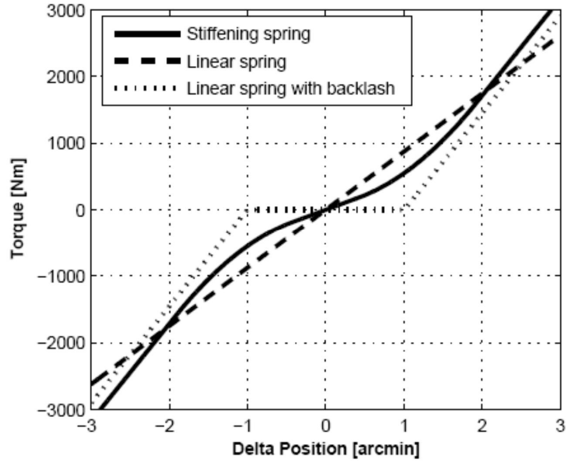


Friction



Bearing Elasticity
Torque Ripple

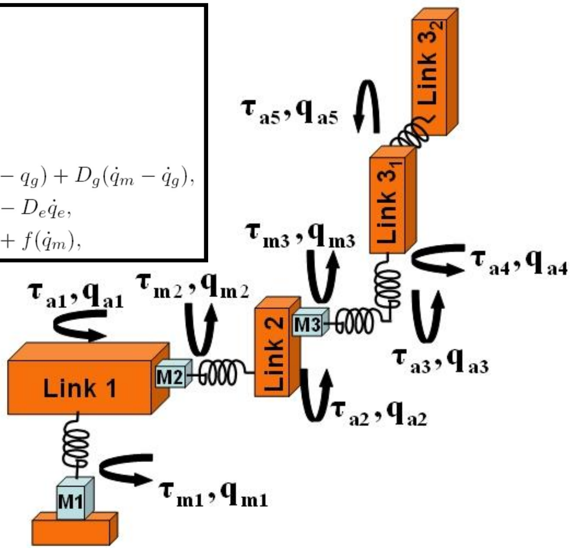
Nonlinear Flexibility



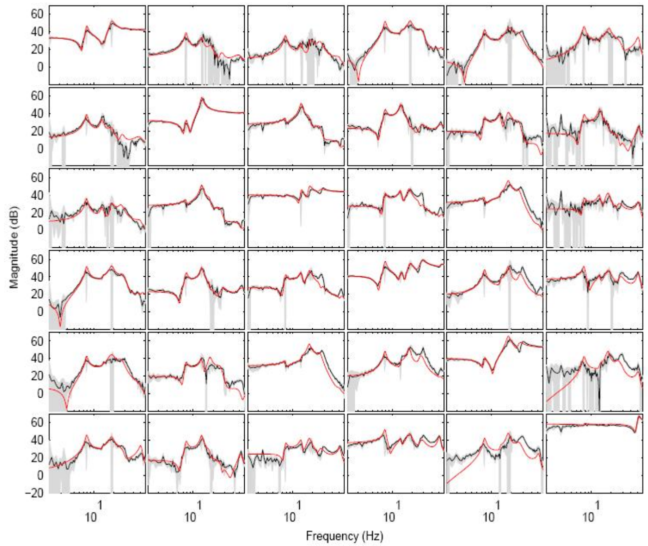
Extended Flexible Joint Model

$$
\begin{aligned}
M_a(q_a)\ddot{q}_a + c(q_a,\dot{q}_a) + g(q_a) &= \tau_a,\\
\tau_a &= \begin{bmatrix}\tau_g\\ \tau_e\end{bmatrix},\\
q_a &= \begin{bmatrix}q_g\\ q_e\end{bmatrix},\\
\tau_g &= K_g(q_m - q_g) + D_g(\dot{q}_m - \dot{q}_g),\\
\tau_e &= -K_e q_e - D_e \dot{q}_e,\\
\tau_m - \tau_g &= M_m \ddot{q}_m + f(\dot{q}_m),
\end{aligned}
$$

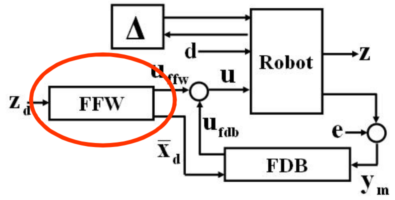
For N links, N motors and M unactuated joints: 2N + M d.o.f.



Extended Flexible Joint Model



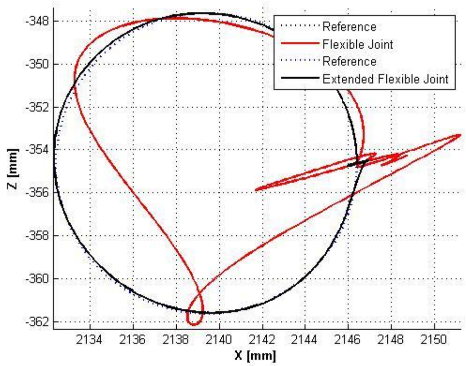
Feedforward Control of Extended Flexible Joint



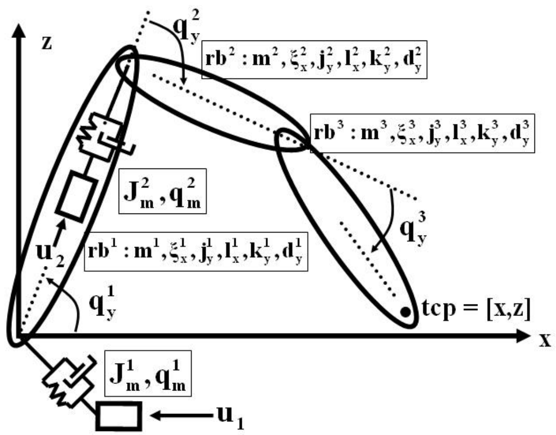
$$\mathbf{M_a(q_a)\dot{v}_a = c(q_a, v_a) + g(q_a) + \tau_a(q_a, q_m, v_a, v_m)}$$
$$\mathbf{M_m \dot{v}_m = \tau_m(q_a, q_m, v_a, v_m) + u}$$
$$\mathbf{\dot{q}_a = v_a}$$
$$\mathbf{\dot{q}_m = v_m}$$
$$\mathbf{Z_d(t) = \Gamma(q_a)}$$

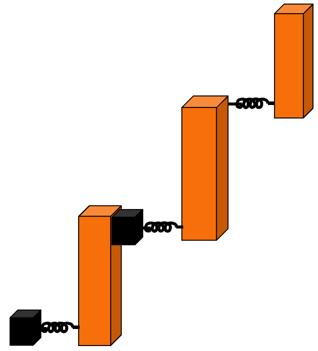
Inverse Dynamics

- Perfect tracking & Point-To-Point
  - A High Index DAE must be solved
  - Solution for M=2, N=3 & M=3, N=9
  - Non-minimum Phase?



Nonlinear Simulation Model





- 2 actuated DOF (N = 2)
- 1 unactuated DOF (M = 1)
- 5 DOF

Simulation Model – The Equations

The inertia matrices M_a and M_m are defined and computed as:

$$M_m = \begin{bmatrix} J_m^1 & 0 \\ 0 & J_m^2 \end{bmatrix}$$

$$M_a = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

$$\begin{aligned} M_{11} = {} & (\xi_x^1)^2 m_1 + (s_y^2)^2 (l_x^1)^2 m_2 - (-c_y^2 l_x^1 - \xi_x^2) m_2 (c_y^2 l_x^1 \\ & + \xi_x^2) + j_{yy}^1 - (s_y^3 c_y^2 l_x^1 + s_y^3 l_x^2 + c_y^3 s_y^2 l_x^1) m_3 (-s_y^3 c_y^2 \\ & l_x^1 - c_y^3 s_y^2 l_x^1 - s_y^3 l_x^2) + j_{yy}^2 - (-\xi_x^3 + s_y^3 s_y^2 l_x^1 - c_y^3 c_y^2 l_x^1 \\ & - c_y^3 l_x^2) m_3 (c_y^3 c_y^2 l_x^1 + c_y^3 l_x^2 - s_y^3 s_y^2 l_x^1 + \xi_x^3) + j_{yy}^3 \end{aligned}$$

$$\begin{aligned} M_{12} = {} & j_{yy}^3 + j_{yy}^2 - (-c_y^2 l_x^1 - \xi_x^2) m_2 \xi_x^2 + (s_y^3 c_y^2 l_x^1 + s_y^3 l_x^2 \\ & + c_y^3 s_y^2 l_x^1) m_3 s_y^3 l_x^2 - (-\xi_x^3 + s_y^3 s_y^2 l_x^1 - c_y^3 c_y^2 l_x^1 - \\ & c_y^3 l_x^2) m_3 (c_y^3 l_x^2 + \xi_x^3) \end{aligned}$$

$$M_{13} = j_{yy}^3 - (-\xi_x^3 + s_y^3 s_y^2 l_x^1 - c_y^3 c_y^2 l_x^1 - c_y^3 l_x^2) m_3 \xi_x^3$$

$$\begin{aligned} M_{21} = {} & \xi_x^2 m_2 (c_y^2 l_x^1 + \xi_x^2) - s_y^3 l_x^2 m_3 (-s_y^3 c_y^2 l_x^1 - c_y^3 s_y^2 l_x^1 \\ & - s_y^3 l_x^2) - (-\xi_x^3 - c_y^3 l_x^2) m_3 (c_y^3 c_y^2 l_x^1 + c_y^3 l_x^2 - s_y^3 s_y^2 l_x^1 \\ & + \xi_x^3) + j_{yy}^2 + j_{yy}^3 \end{aligned}$$

$$\begin{aligned} M_{22} = {} & (\xi_x^2)^2 m_2 - (-\xi_x^3 - c_y^3 l_x^2) m_3 (c_y^3 l_x^2 + \xi_x^3) + j_{yy}^2 \\ & + (s_y^3)^2 (l_x^2)^2 m_3 + j_{yy}^3 \end{aligned}$$

$$M_{23} = j_{yy}^3 - (-\xi_x^3 - c_y^3 l_x^2) m_3 \xi_x^3$$

$$M_{31} = \xi_x^3 m_3 (c_y^3 c_y^2 l_x^1 + c_y^3 l_x^2 - s_y^3 s_y^2 l_x^1 + \xi_x^3) + j_{yy}^3$$

$$M_{32} = \xi_x^3 m_3 (c_y^3 l_x^2 + \xi_x^3) + j_{yy}^3$$

$$M_{33} = (\xi_x^3)^2 m_3 + j_{yy}^3$$

The kinematics is computed as:

$$\Gamma = \begin{bmatrix} x \\ z \end{bmatrix}$$

$$\begin{aligned} x = {} & c_y^3 c_y^1 c_y^2 l_x^3 + c_y^1 l_x^1 - s_y^3 s_y^1 c_y^2 l_x^3 - s_y^3 c_y^1 s_y^2 l_x^3 - c_y^3 s_y^1 s_y^2 l_x^3 \\ & - s_y^1 s_y^2 l_x^2 + c_y^1 c_y^2 l_x^2 \end{aligned}$$

$$\begin{aligned} z = {} & -c_y^3 c_y^1 s_y^2 l_x^3 - c_y^3 s_y^1 c_y^2 l_x^3 - s_y^1 l_x^1 - s_y^3 c_y^1 c_y^2 l_x^3 + s_y^3 s_y^1 s_y^2 l_x^3 \\ & - c_y^1 s_y^2 l_x^2 - s_y^1 c_y^2 l_x^2 \end{aligned}$$

Finally, the position and speed dependent terms are computed as:

$$\begin{aligned} \gamma_{a1} = {} & s_y^2 l_x^1 m_2 ((v_y^2)^2 \xi_x^2 + c_y^2 (v_y^1)^2 l_x^1 + 2 v_y^1 v_y^2 \xi_x^2 + (v_y^1)^2 \xi_x^2) \\ & + (-c_y^2 l_x^1 - \xi_x^2) m_2 s_y^2 (v_y^1)^2 l_x^1 + (s_y^3 c_y^2 l_x^1 + s_y^3 l_x^2 \\ & + c_y^3 s_y^2 l_x^1) m_3 (2 v_y^1 v_y^2 \xi_x^3 + 2 v_y^3 v_y^1 \xi_x^3 + (v_y^1)^2 \xi_x^3 + (v_y^3)^2 \xi_x^3 \\ & + 2 v_y^3 v_y^2 \xi_x^3 - s_y^3 s_y^2 (v_y^1)^2 l_x^1 + 2 c_y^3 v_y^1 v_y^2 l_x^2 + c_y^3 (v_y^2)^2 l_x^2 \\ & + (v_y^2)^2 \xi_x^3 + c_y^3 (v_y^1)^2 l_x^2 + c_y^3 c_y^2 (v_y^1)^2 l_x^1) + (-\xi_x^3 \\ & + s_y^3 s_y^2 l_x^1 - c_y^3 c_y^2 l_x^1 - c_y^3 l_x^2) m_3 (c_y^3 s_y^2 (v_y^1)^2 l_x^1 \\ & + s_y^3 c_y^2 (v_y^1)^2 l_x^1 + 2 s_y^3 v_y^1 v_y^2 l_x^2 + s_y^3 (v_y^2)^2 l_x^2 + s_y^3 (v_y^1)^2 l_x^2) \\ & + \xi_x^1 c_y^1 m_1 g - s_y^2 l_x^1 (-s_y^1 c_y^2 - c_y^1 s_y^2) m_2 g - (-c_y^2 l_x^1 - \xi_x^2) \\ & (c_y^1 c_y^2 - s_y^1 s_y^2) m_2 g - (s_y^3 c_y^2 l_x^1 + s_y^3 l_x^2 + c_y^3 s_y^2 l_x^1)((-s_y^1 c_y^2 \\ & - c_y^1 s_y^2) c_y^3 - (c_y^1 c_y^2 - s_y^1 s_y^2) s_y^3) m_3 g - (-\xi_x^3 + s_y^3 s_y^2 l_x^1 \\ & - c_y^3 c_y^2 l_x^1 - c_y^3 l_x^2)((-s_y^1 c_y^2 - c_y^1 s_y^2) s_y^3 + (c_y^1 c_y^2 - s_y^1 s_y^2) c_y^3) \\ & m_3 g - k_y^1 (q_y^1 - q_m^1) - d_y^1 (v_y^1 - v_m^1) \end{aligned}$$

$$\begin{aligned} \gamma_{a2} = {} & -\xi_x^2 m_2 s_y^2 (v_y^1)^2 l_x^1 + s_y^3 l_x^2 m_3 (2 v_y^1 v_y^2 \xi_x^3 + 2 v_y^3 v_y^1 \xi_x^3 \\ & + (v_y^1)^2 \xi_x^3 + (v_y^3)^2 \xi_x^3 + 2 v_y^3 v_y^2 \xi_x^3 - s_y^3 s_y^2 (v_y^1)^2 \\ & l_x^1 + 2 c_y^3 v_y^1 v_y^2 l_x^2 + c_y^3 (v_y^2)^2 l_x^2 + (v_y^2)^2 \xi_x^3 + c_y^3 (v_y^1)^2 \\ & l_x^2 + c_y^3 c_y^2 (v_y^1)^2 l_x^1) + (-\xi_x^3 - c_y^3 l_x^2) m_3 (c_y^3 s_y^2 (v_y^1)^2 l_x^1 \\ & + s_y^3 c_y^2 (v_y^1)^2 l_x^1 + 2 s_y^3 v_y^1 v_y^2 l_x^2 + s_y^3 (v_y^2)^2 l_x^2 + s_y^3 (v_y^1)^2 l_x^2) \\ & + \xi_x^2 (c_y^1 c_y^2 - s_y^1 s_y^2) m_2 g - s_y^3 l_x^2 ((-s_y^1 c_y^2 - c_y^1 s_y^2) c_y^3 \\ & - (c_y^1 c_y^2 - s_y^1 s_y^2) s_y^3) m_3 g - (-\xi_x^3 - c_y^3 l_x^2)((-s_y^1 c_y^2 \\ & - c_y^1 s_y^2) s_y^3 + (c_y^1 c_y^2 - s_y^1 s_y^2) c_y^3) m_3 g \\ & - k_y^2 (q_y^2 - q_m^2) - d_y^2 (v_y^2 - v_m^2) \end{aligned}$$

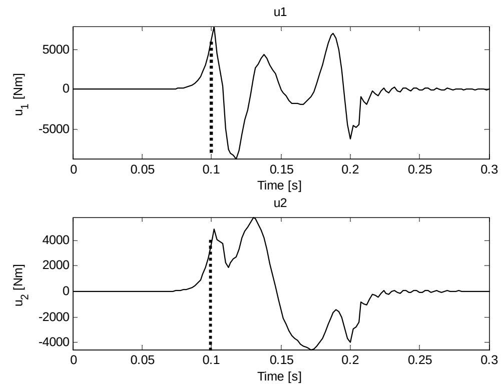
$$\begin{aligned} \gamma_{a3} = {} & -\xi_x^3 m_3 (c_y^3 s_y^2 (v_y^1)^2 l_x^1 + s_y^3 c_y^2 (v_y^1)^2 l_x^1 + 2 s_y^3 v_y^1 v_y^2 l_x^2 \\ & + s_y^3 (v_y^2)^2 l_x^2 + s_y^3 (v_y^1)^2 l_x^2) + \xi_x^3 ((-s_y^1 c_y^2 - c_y^1 s_y^2) s_y^3 \\ & + (c_y^1 c_y^2 - s_y^1 s_y^2) c_y^3) m_3 g - k_y^3 q_y^3 - d_y^3 v_y^3 \end{aligned}$$

$$\gamma_{m1} = k_y^1 (q_y^1 - q_m^1) + d_y^1 (v_y^1 - v_m^1)$$

$$\gamma_{m2} = k_y^2 (q_y^2 - q_m^2) + d_y^2 (v_y^2 - v_m^2)$$

Non-minimum Phase

- Extended Flexible Joint & Flexible Link is normally NMP for perfect tracking
- Example: 5 DOF 2 Axis Model:
  Non-causal solution, movement starts at t = 0.1 s



Control when in contact with the environment

- Position and speed control not enough
- Why?
- Sensor configuration
  - Wrist torque/force sensor
  - Joint torque/force sensor
  - Tactile

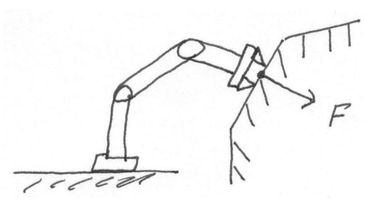


Figure 9.2 in Spong etal

Force and torque when in contact with environment

Virtual displacement (again)

$$\delta X = J(q)\delta q$$

virtual work

$$\delta w = F^T \delta X - \tau^T \delta q$$

$$= \left(F^T J(q) - \tau^T\right)\delta q$$

$$\Rightarrow \tau = J^T(q)F$$

Complete dynamics

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + J^T(q)F_e = u$$

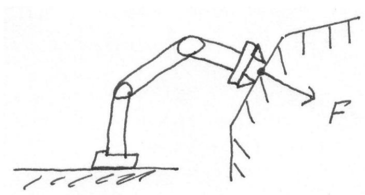


Figure 9.2 in Spong etal

Coordinate frames and constraints

Let the velocity (Twist) be defined $\xi = \begin{bmatrix} v \\ \omega \end{bmatrix}$

and the force (Wrenche) $F = \begin{bmatrix} f \\ n \end{bmatrix}$

Reciprocity condition $\xi^T F = v^T f + \omega^T n = 0$

Task constraints

Natural constraints. Constraints imposed by the task.

$$\xi^T F = v_x f_x + v_y f_y + v_z f_z + \omega_x n_x + \omega_y n_y + \omega_z n_z = 0$$

Example:

Natural constraints	Artificial constraints
$v_x = 0$	$f_x = 0$
$v_y = 0$	$f_y = 0$
$f_z = 0$	$v_z = v_d$
$\omega_x = 0$	$n_x = 0$
$\omega_y = 0$	$n_y = 0$
$n_z = 0$	$\omega_z = 0$

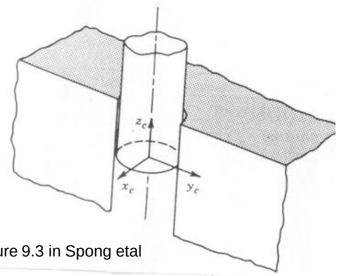
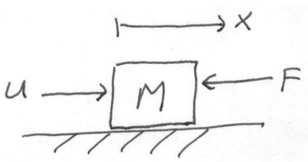


Figure 9.3 in Spong etal

Impedance control

$$M\ddot{x} = u - F$$



Let

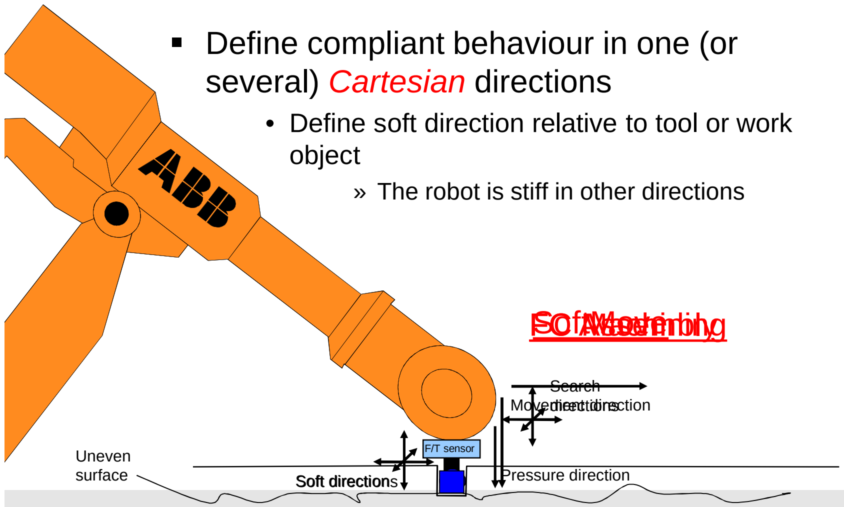
$$u = -(\frac{M}{m} + 1)F$$

The closed loop system is

$$M\ddot{x} = -(\frac{M}{m} + 1)F + F \Rightarrow m\ddot{x} = -F$$

Force Control and SoftMove
General idea – mechanical compliance

- Define compliant behaviour in one (or several) *Cartesian* directions
  - Define soft direction relative to tool or work object
    - » The robot is stiff in other directions



Force Control and SoftMove
Current important applications

- Force Control
  - Machining
    - Milling, grinding, fettling, polishing, etc.
    - FC SpeedChange functionality can be used to control path speed
  - Assembly
    - Complex, multi-stage assembly operations
  - Product testing
- SoftMove
  - Ejector machines, extracting (die casting)
  - Handling workpiece variations

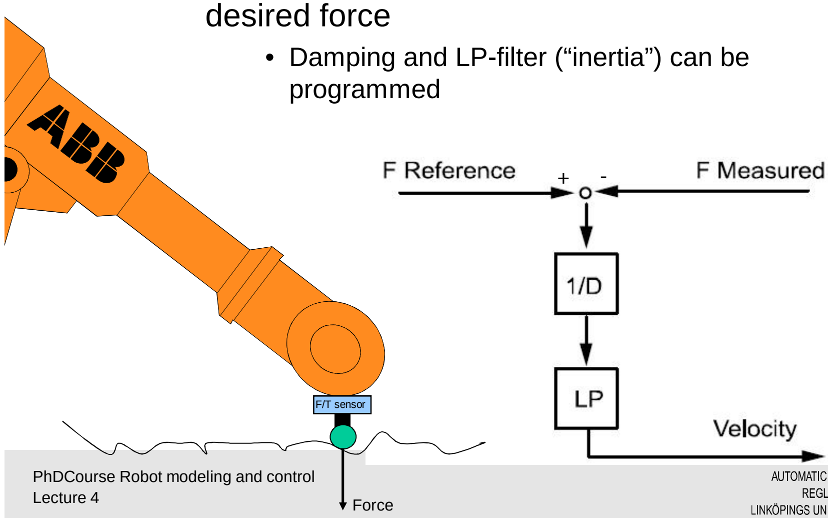
Force Control Assembly

Force Control Machining

SoftMove

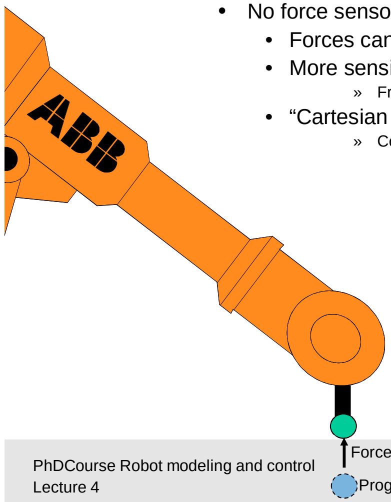
Force Control technology
Force feedback

- Speed reference is generated in the soft directions based on current deviation from desired force
  - Damping and LP-filter ("inertia") can be programmed



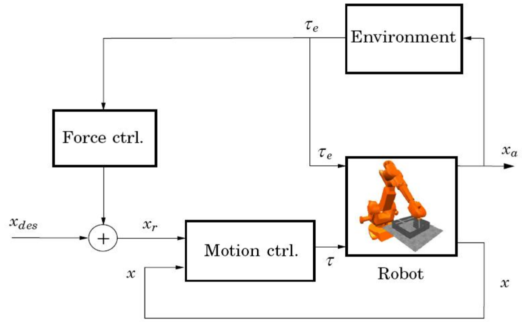
SoftMove technology
Servo controller modification

- Make servo control “softer” in a chosen *Cartesian* direction
  - No force sensor needed
    - Forces can not be controlled
    - More sensitive to friction
      - » Friction can be compensated by adding a force offset
    - “Cartesian Soft Servo”
      - » Compare ordinary (joint) Soft Servo functionality



Force Control and SoftMove
Important aspects to remember

- Forces are unpredictable
  No feed-forward or path planning to improve performance
- A robot in contact behaves very differently depending on contact stiffness and geometry



Summary

- Robot Motion Control Overview
  - Inner/outer loop control architecture
- Current and Torque Control
- Control Methods for Rigid Robots
  - Computed torque
  - Feedback linearization
- Control Methods for Flexible Robots
  - Feed-forward control
  - State feedback
- Interaction with the environment