Robotics 2

Dynamic model of robots:

Analysis, properties, extensions, parametrization, identification, uses

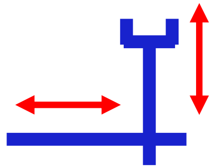
Prof. Alessandro De Luca

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

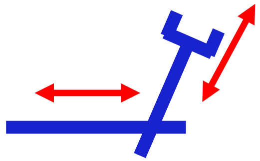
Analysis of inertial couplings

- Cartesian robot



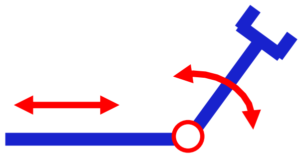
$$M = \begin{pmatrix} m_{11} & 0 \\ 0 & m_{22} \end{pmatrix}$$

- Cartesian "skew" robot



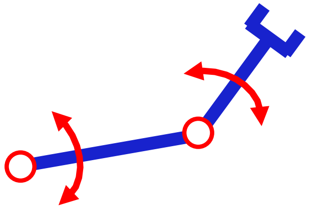
$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{pmatrix}$$

- PR robot



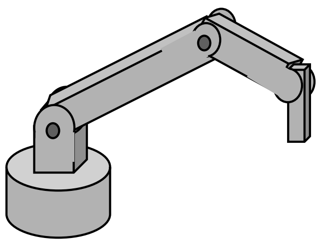
$$M = \begin{pmatrix} m_{11} & m_{12}(q_2) \\ m_{12}(q_2) & m_{22} \end{pmatrix}$$

- 2R robot



$$M = \begin{pmatrix} m_{11}(q_2) & m_{12}(q_2) \\ m_{12}(q_2) & m_{22} \end{pmatrix}$$

- 3R articulated robot
  (under simplifying assumptions on the CoM)



$$M = \begin{pmatrix} m_{11}(q_2, q_3) & 0 & 0 \\ 0 & m_{22}(q_3) & m_{23}(q_3) \\ 0 & m_{23}(q_3) & m_{33} \end{pmatrix}$$

- dynamic model turns out to be linear if
  - $g \equiv 0$
  - M constant $\Rightarrow c \equiv 0$

$d_{c2} = 0$ in PR
$d_2 = 0$ in 2R

center of mass
of link 2
on joint 2 axis

Analysis of gravity term

- static balancing
  - distribution of masses (including motors)
- mechanical compensation
  - articulated system of springs
  - closed kinematic chains
- absence of gravity
  - applications in space
  - constant U_g (motion on horizontal plane)

$$\Rightarrow \quad g(q) \approx 0$$

Bounds on dynamic terms

- for an open-chain (serial) manipulator, there always exist positive real constants k_0 to k_7 such that, for any value of q and $\dot{q}$

$$k_0 \le \|M(q)\| \le k_1 + k_2\|q\| + k_3\|q\|^2$$ inertia matrix

$$\|S(q,\dot{q})\| \le (k_4 + k_5\|q\|)\ \|\dot{q}\|$$ factorization matrix of Coriolis/centrifugal terms

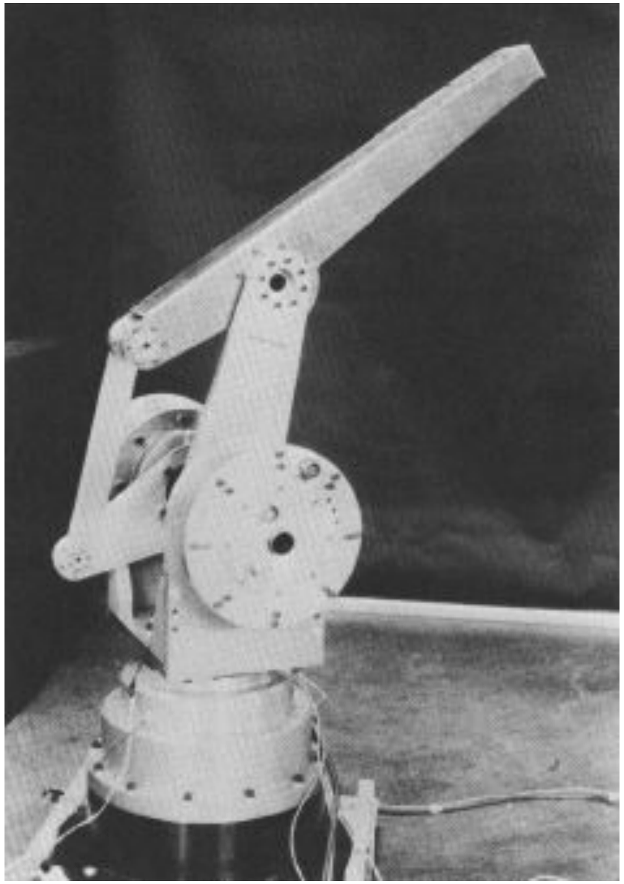
$$\|g(q)\| \le k_6 + k_7\|q\|$$ gravity vector

- if the robot has only revolute joints, these simplify to

$$\boxed{k_0 \le \|M(q)\| \le k_1 \quad \|S(q,\dot{q})\| \le k_4\|\dot{q}\| \quad \|g(q)\| \le k_6}$$

NOTE: norms are either for vectors or for matrices (induced norms)

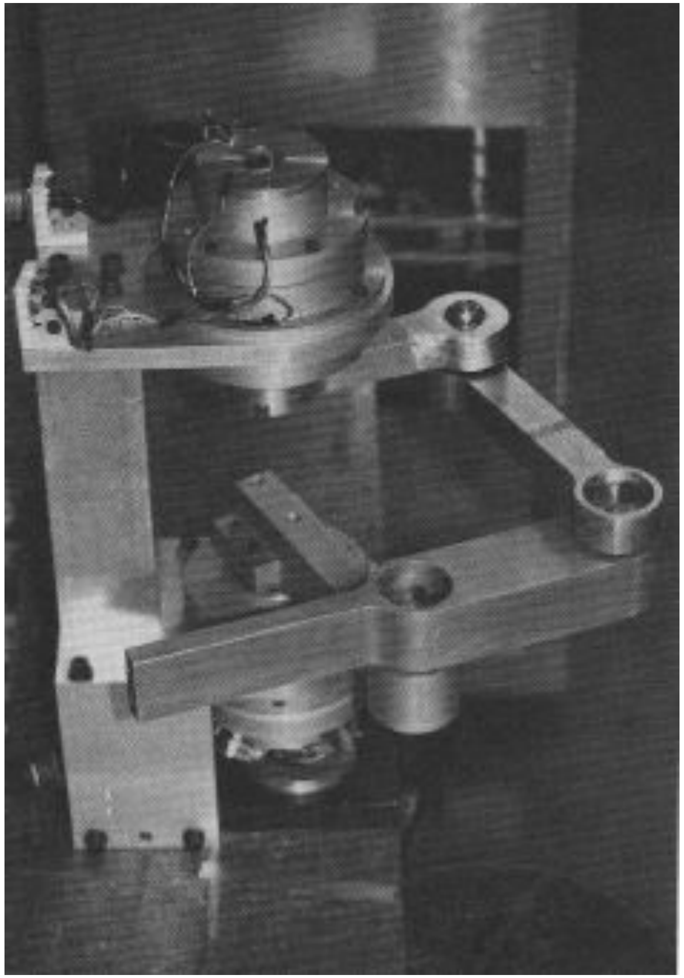
Robots with closed kinematic chains - 1



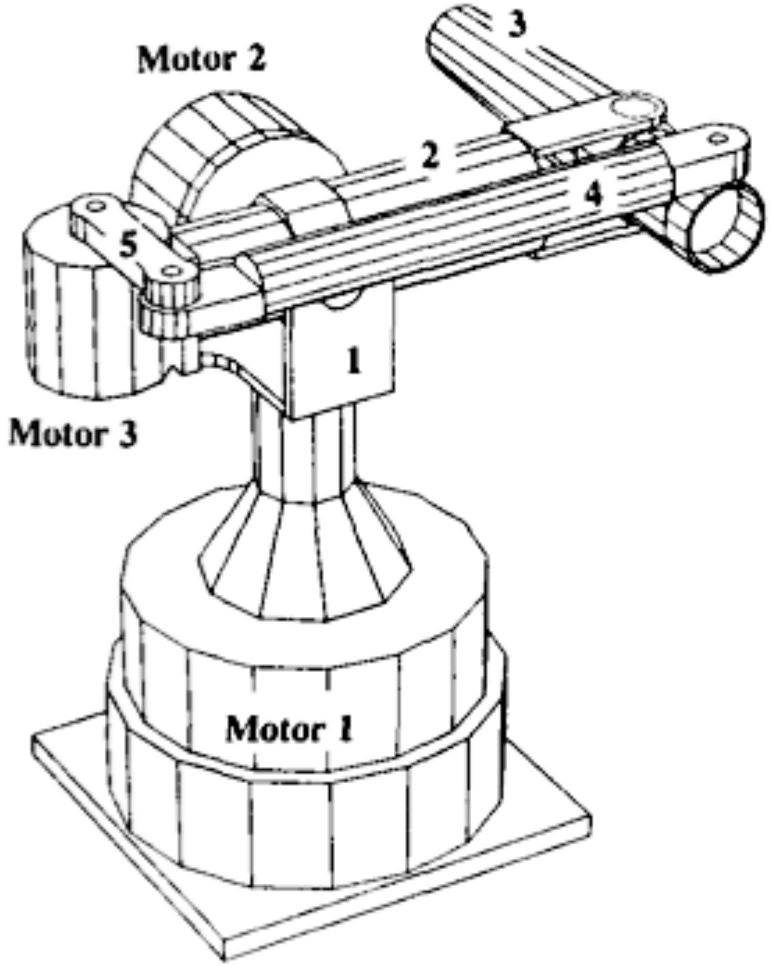
Comau Smart NJ130

MIT Direct Drive Mark II and Mark III

Robots with closed kinematic chains - 2



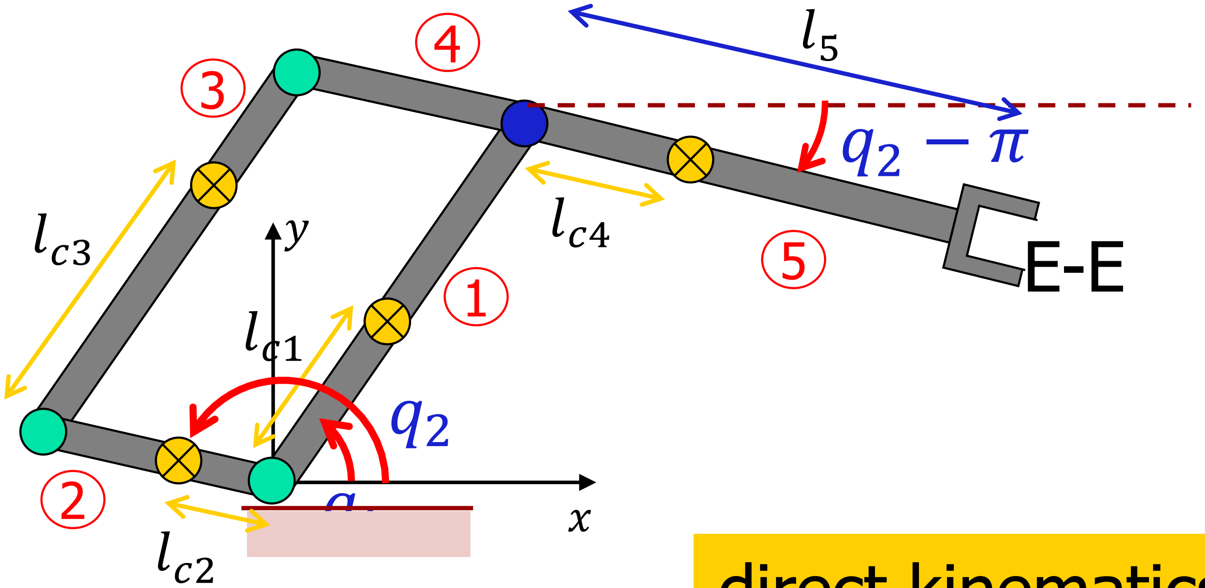
MIT Direct Drive Mark IV
(planar five-bar linkage)



UMinnesota Direct Drive Arm
(spatial five-bar linkage)

Robot with parallelogram structure

(planar) kinematics and dynamics



center of mass: arbitrary l_{ci}

parallelogram:

$l_1 = l_3$

$l_2 = l_4$

direct kinematics

$$p_{EE} = \begin{pmatrix} l_1 c_1 \\ l_1 s_1 \end{pmatrix} + \begin{pmatrix} l_5 \cos(q_2 - \pi) \\ l_5 \sin(q_2 - \pi) \end{pmatrix} = \begin{pmatrix} l_1 c_1 \\ l_1 s_1 \end{pmatrix} - \begin{pmatrix} l_5 c_2 \\ l_5 s_2 \end{pmatrix}$$

position of center of masses

$$p_{c1} = \begin{pmatrix} l_{c1} c_1 \\ l_{c1} s_1 \end{pmatrix} \quad p_{c2} = \begin{pmatrix} l_{c2} c_2 \\ l_{c2} s_2 \end{pmatrix} \quad p_{c3} = \begin{pmatrix} l_2 c_2 \\ l_2 s_2 \end{pmatrix} + \begin{pmatrix} l_{c3} c_1 \\ l_{c3} s_1 \end{pmatrix} \quad p_{c4} = \begin{pmatrix} l_1 c_1 \\ l_1 s_1 \end{pmatrix} - \begin{pmatrix} l_{c4} c_2 \\ l_{c4} s_2 \end{pmatrix}$$

Kinetic energy

linear/angular velocities

$$v_{c1} = \begin{pmatrix} -l_{c1}s_1 \\ l_{c1}c_1 \end{pmatrix} \dot{q}_1 \qquad v_{c3} = \begin{pmatrix} -l_{c3}s_1 \\ l_{c3}c_1 \end{pmatrix} \dot{q}_1 + \begin{pmatrix} -l_2 s_2 \\ l_2 c_2 \end{pmatrix} \dot{q}_2 \qquad \omega_1 = \omega_3 = \dot{q}_1$$

$$v_{c2} = \begin{pmatrix} -l_{c2}s_2 \\ l_{c2}c_2 \end{pmatrix} \dot{q}_2 \qquad v_{c4} = \begin{pmatrix} -l_1 s_1 \\ l_1 c_1 \end{pmatrix} \dot{q}_1 + \begin{pmatrix} l_{c4}s_2 \\ -l_{c4}c_2 \end{pmatrix} \dot{q}_2 \qquad \omega_2 = \omega_4 = \dot{q}_2$$

Note: a (planar) 2D notation is used here!

T_i

$$T_1 = \frac{1}{2} m_1 l_{c1}^2 \dot{q}_1^2 + \frac{1}{2} I_{c1,zz} \dot{q}_1^2 \qquad T_2 = \frac{1}{2} m_2 l_{c2}^2 \dot{q}_2^2 + \frac{1}{2} I_{c2,zz} \dot{q}_2^2$$

$$T_3 = \frac{1}{2} m_3 (l_2^2 \dot{q}_2^2 + l_{c3}^2 \dot{q}_1^2 + 2 l_2 l_{c3} c_{2-1} \dot{q}_1 \dot{q}_2) + \frac{1}{2} I_{c3,zz} \dot{q}_1^2$$

$$T_4 = \frac{1}{2} m_4 (l_1^2 \dot{q}_1^2 + l_{c4}^2 \dot{q}_2^2 - 2 l_1 l_{c4} c_{2-1} \dot{q}_1 \dot{q}_2) + \frac{1}{2} I_{c4,zz} \dot{q}_2^2$$

Robot inertia matrix

$$T = \sum_{i=1}^{4} T_i = \frac{1}{2}\dot{q}^T M(q)\dot{q}$$

$$M(q) = \begin{pmatrix} I_{c1,zz} + m_1 l_{c1}^2 + I_{c3,zz} + m_3 l_{c3}^2 + m_4 l_1^2 & \text{symm} \\ (m_3 l_2 l_{c3} - m_4 l_1 l_{c4}) c_{2-1} & I_{c2,zz} + m_2 l_{c2}^2 + I_{c4,zz} + m_4 l_{c4}^2 + m_3 l_2^2 \end{pmatrix}$$

structural condition
in mechanical design

$$m_3 l_2 l_{c3} = m_4 l_1 l_{c4} \quad (*)$$

$\Downarrow$

$M(q)$ diagonal and constant $\Rightarrow$ centrifugal and Coriolis terms $\equiv 0$

mechanically DECOUPLED and LINEAR
dynamic model (up to the gravity term $g(q)$)

big advantage for the design of a motion control law!

Potential energy and gravity terms

from the y-components of vectors p_{ci}

U_i

$$U_1 = m_1 g_0 l_{c1} s_1 \qquad U_2 = m_2 g_0 l_{c2} s_2$$

$$U_3 = m_3 g_0 (l_2 s_2 + l_{c3} s_1) \qquad U_4 = m_4 g_0 (l_1 s_1 - l_{c4} s_2)$$

$$U = \sum_{i=1}^{4} U_i$$

$$g(q) = \left(\frac{\partial U}{\partial q}\right)^T = \begin{pmatrix} g_0 (m_1 l_{c1} + m_3 l_{c3} + m_4 l_1) c_1 \\ g_0 (m_2 l_{c2} + m_3 l_2 - m_4 l_{c4}) c_2 \end{pmatrix} = \begin{pmatrix} g_1(q_1) \\ g_2(q_2) \end{pmatrix}$$

gravity components are always "decoupled"

in addition, when (*) holds

$$\boxed{\begin{aligned} m_{11}\ddot{q}_1 + g_1(q_1) &= u_1 \\ m_{22}\ddot{q}_2 + g_2(q_2) &= u_2 \end{aligned}}$$

u_i are (non-conservative) torques performing work on q_i

further structural conditions in the mechanical design lead to $g(q) \equiv 0$!!

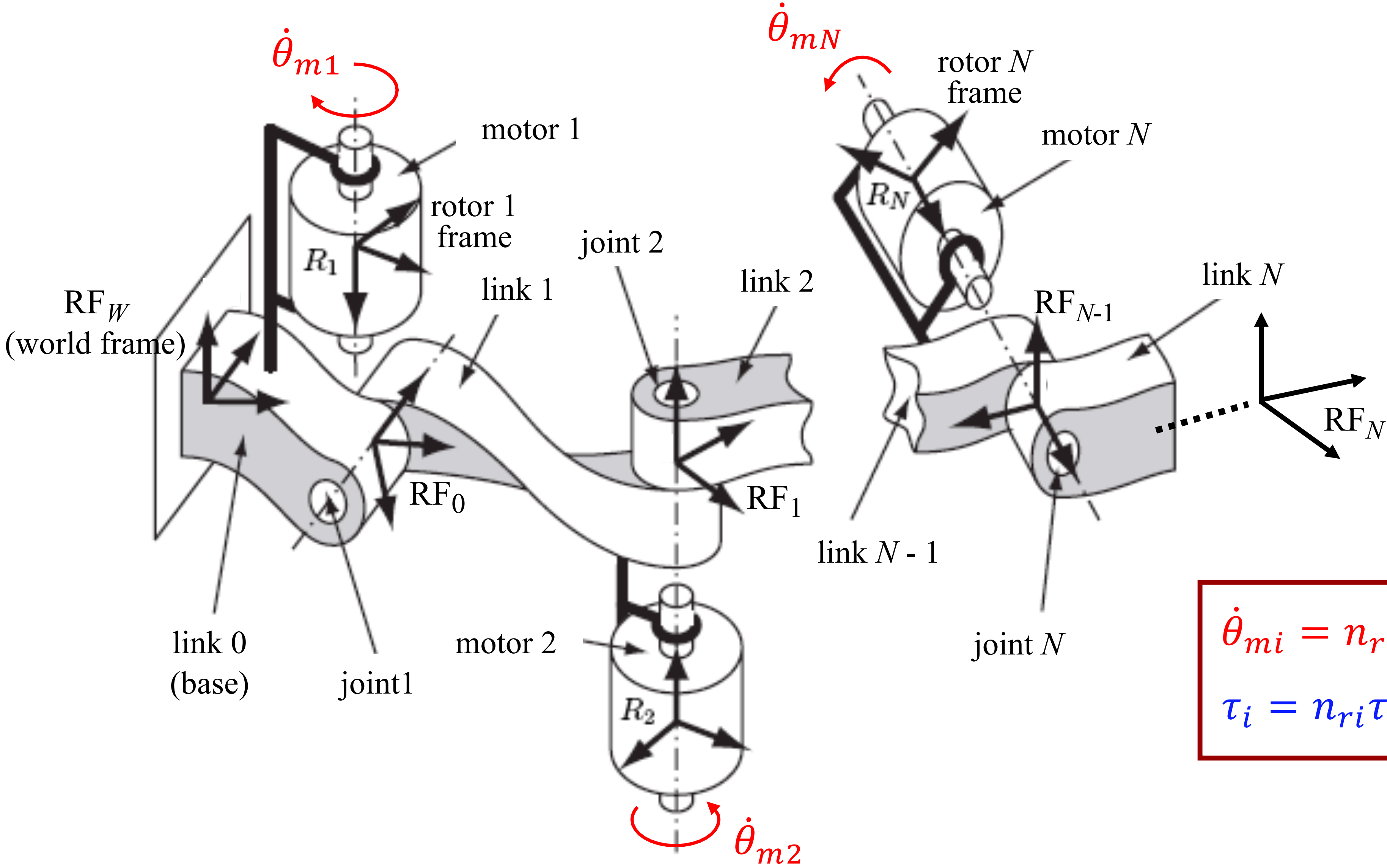
Adding dynamic terms ...

- dissipative phenomena due to friction at the joints/transmissions
  - viscous, dry, Coulomb, ...
  - local effects at the joints
  - difficult to model in general, except for:

$$\boxed{u_{V,i} = -F_{V,i}\,\dot{q}_i} \qquad \boxed{u_{S,i} = -F_{S,i}\,\mathrm{sgn}(\dot{q}_i)}$$

- inclusion of electrical motors (as additional rigid bodies)
  - motor i mounted on link $i-1$ (or before), typically with its motion (spinning) axis aligned with joint i axis
  - (balanced) mass of motor included in total mass of carrying link
  - (rotor) inertia has to be added to robot kinetic energy
  - transmissions with reduction gears (often, large reduction ratios)
  - in some cases, multiple motors cooperate in moving multiple links: use a transmission coupling matrix Γ (with off-diagonal elements)

Placement of motors along the chain



$$\dot{\theta}_{mi} = n_{ri}\dot{\theta}_i$$

$$\tau_i = n_{ri}\tau_{mi}$$

Resulting dynamic model

- simplifying assumption: in the rotational part of kinetic energy, only the "spinning" rotor velocity is considered

$$T_{mi} = \frac{1}{2} I_{mi} \dot{\theta}_{mi}^2 = \frac{1}{2} I_{mi} n_{ri}^2 \dot{q}_i^2 = \frac{1}{2} B_{mi} \dot{q}_i^2 \qquad T_m = \sum_{i=1}^{N} T_{mi} = \frac{1}{2} \dot{q}^T \underset{\text{diagonal}, > 0}{B_m} \dot{q}$$

- including all added terms, the robot dynamics becomes

$$(M(q) + \underset{\text{constant}}{B_m})\ddot{q} + \underset{\text{does NOT contribute to } c}{c(q,\dot{q})} + g(q) + \underbrace{F_V \dot{q} + F_S \operatorname{sgn}(\dot{q})}_{F_V > 0,\, F_S > 0 \text{ diagonal}} = \tau$$

moved to the left ...

motor torques (**after** reduction gears)

- scaling by the reduction gears, looking from the motor side

diagonal

except the terms m_{jj}

$$\left(I_m + \operatorname{diag}\left\{ \frac{m_{ii}(q)}{n_{ri}^2} \right\} \right) \ddot{\theta}_m + \operatorname{diag}\left\{ \frac{1}{n_{ri}} \right\} \left(\sum_{j=1}^{N} \bar{M}_j(q) \ddot{q}_j + f(q,\dot{q}) \right) = \tau_m$$

motor torques (**before** reduction gears)

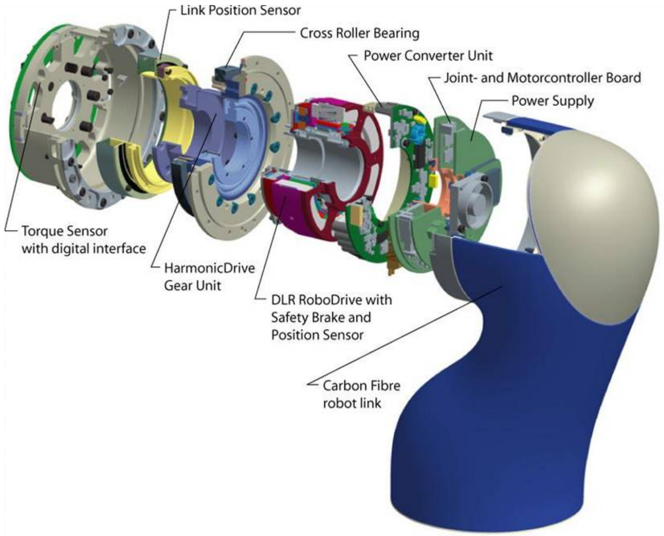
Including joint elasticity

- in industrial robots, use of motion transmissions based on
  - belts
  - harmonic drives
  - long shafts

  introduces flexibility between actuating motors (input) and driven links (output)
- in research robots for human cooperation, compliance in the transmissions is introduced on purpose for safety
  - actuator relocation by means of (compliant) cables and pulleys
  - harmonic drives and lightweight (but rigid) link design
  - redundant (macro-mini or parallel) actuation, with elastic couplings
- in both cases, flexibility is modeled as concentrated at the joints
- most of the times, assuming small joint deformation (elastic domain)

Robots with joint elasticity

Dexter
with cable transmissions



DLR LWR-III
with harmonic drives

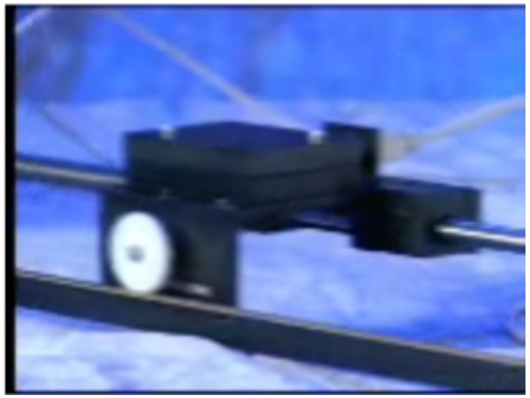
motor

elastic spring

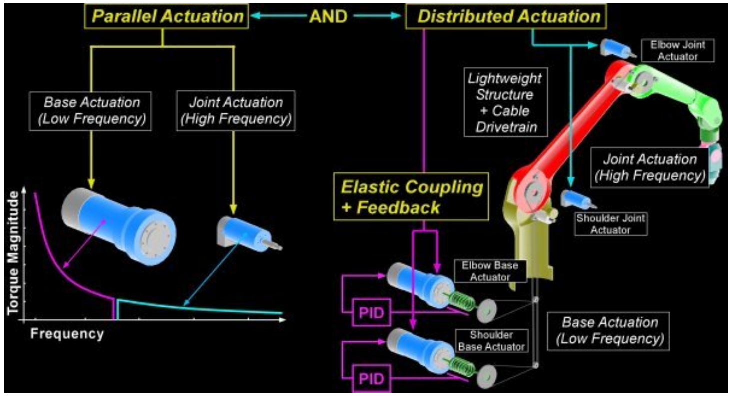
(stiffness K)

load/link

Quanser Flexible Joint
(1-dof linear, educational)



video



Stanford DECMMA
with micro-macro actuation

Dynamic model of robots with elastic joints

- introduce $2N$ generalized coordinates
  - $q = N$ link positions
  - $\theta = N$ motor positions (after reduction, $\theta_i = \theta_{mi}/n_{ri}$)
- add motor kinetic energy T_m to that of the links $T_q = \frac{1}{2}\dot{q}^T M(q)\dot{q}$

$$T_{mi} = \frac{1}{2} I_{mi}\dot{\theta}_{mi}^2 = \frac{1}{2} I_{mi} n_{ri}^2 \dot{\theta}_i^2 = \frac{1}{2} B_{mi}\dot{\theta}_i^2 \qquad T_m = \sum_{i=1}^{N} T_{mi} = \frac{1}{2}\dot{\theta}^T B_m \dot{\theta}$$

diagonal, > 0

- add elastic potential energy U_e to that due to gravity $U_g(q)$
  - $K =$ matrix of joint stiffness (diagonal, > 0)

$$U_{ei} = \frac{1}{2} K_i \left(q_i - \left(\frac{\theta_{mi}}{n_{ri}} \right) \right)^2 = \frac{1}{2} K_i (q_i - \theta_i)^2 \quad U_e = \sum_{i=1}^{N} U_{ei} = \frac{1}{2}(q-\theta)^T K (q-\theta)$$

- apply Euler-Lagrange equations w.r.t. (q, θ)

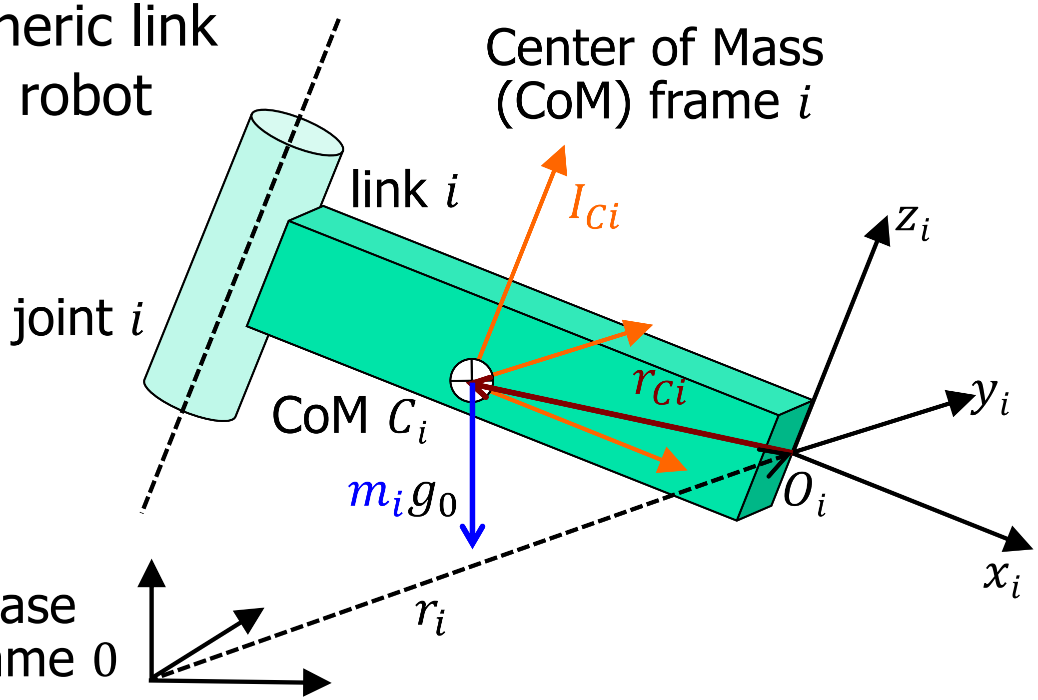
$2N$ 2nd-order differential equations

$$\begin{cases} M(q)\ddot{q} + c(q,\dot{q}) + g(q) + K(q-\theta) = 0 \\ B_m\ddot{\theta} + K(\theta - q) = \tau \end{cases}$$

no external torques performing work on q

Dynamic parameters of robots - 1

- consider a generic link of a fully rigid robot



- each link is characterized by 10 dynamic parameters

$$\left\{ m_i \quad r_{Ci} = \begin{pmatrix} r_{xi} \\ r_{yi} \\ r_{zi} \end{pmatrix} \quad I_{Ci} = \begin{pmatrix} I_{Ci,xx} & I_{Ci,xy} & I_{Ci,xz} \\ & I_{Ci,yy} & I_{Ci,yz} \\ \text{symm} & & I_{Ci,zz} \end{pmatrix} \right.$$

- however, the robot dynamics depends in a nonlinear way on some of these parameters (e.g., through the combination $I_{ci,zz} + m_i r_{xi}^2$)

Dynamic parameters of robots - 2

- kinetic energy and gravity potential energy can both be rewritten so that a new set of dynamic parameters appears only in a linear way
  - need to re-express link inertia and CoM position in (any) known kinematic frame attached to the link (same orientation as the barycentric frame)
- fundamental kinematic relation

$$v_{ci} = v_i + \omega_i \times r_{Ci} = v_i + S(\omega_i)\, r_{Ci} = v_i - S(r_{Ci})\, \omega_i$$

- kinetic energy of link i

$$\begin{aligned} T_i &= \frac{1}{2} m_i v_{Ci}^T v_{Ci} + \frac{1}{2} \omega_i^T I_{Ci} \omega_i \\ &= \frac{1}{2} m_i (v_i - S(r_{Ci})\omega_i)^T (v_i - S(r_{Ci})\omega_i) + \frac{1}{2} \omega_i^T I_{Ci} \omega_i \\ &= \frac{1}{2} m_i v_i^T v_i + \omega_i^T \big(\underbrace{I_{Ci} + m_i S^T(r_{Ci}) S(r_{Ci})}_{\text{Steiner theorem}}\big) \omega_i - v_i^T S(m_i r_{Ci}) \omega_i \end{aligned}$$

$$\rightarrow I_i = \begin{pmatrix} I_{i,xx} & I_{i,xy} & I_{i,xz} \\ & I_{i,yy} & I_{i,yz} \\ \text{symm} & & I_{i,zz} \end{pmatrix}$$

Dynamic parameters of robots - 3

- gravitational potential energy of link i

$$U_i = -m_i g_0^T r_{0,Ci} = -m_i g_0^T (r_i + r_{Ci}) = -m_i g_0^T r_i - g_0^T (m_i r_{Ci})$$

- by expressing vectors and matrices in frame i, both T_i and U_i are linear in the set of 10 (constant) standard parameters $\pi_i \in \mathbb{R}^{10}$

$$T_i = \frac{1}{2} m_i \, {}^i v_i^T \, {}^i v_i + m_i \, {}^i r_{Ci}^T S({}^i v_i) \, {}^i \omega_i + {}^i \omega_i^T \, {}^i I_i \, {}^i \omega_i$$

$$U_i = -m_i g_0^T r_i - g_0^T \, {}^0 R_i \, (m_i \, {}^i r_{Ci})$$

$$\pi_i = \begin{pmatrix} m_i \\ m_i {}^i r_{Ci} \\ vect\{{}^i I_i\} \end{pmatrix} = (\underbrace{m_i}_{\substack{\text{mass of link } i \\ (0\text{-th order moment})}} \quad \underbrace{m_i {}^i r_{Ci,x} \quad m_i {}^i r_{Ci,y} \quad m_i {}^i r_{Ci,z}}_{\substack{\text{mass} \times \text{CoM position of link } i \\ (1\text{-st order moment})}} \quad \underbrace{{}^i I_{i,xx} \quad {}^i I_{i,xy} \quad {}^i I_{i,xz} \quad {}^i I_{i,yy} \quad {}^i I_{i,yz} \quad {}^i I_{i,zz}}_{\substack{\text{inertia of link } i \\ (2\text{-nd order moment})}})^T$$

- since the E-L equations involve only linear operations on T and U, also the robot dynamic model is linear in the standard parameters $\pi \in \mathbb{R}^{10N}$

Dynamic parameters of robots - 4

- using a $N \times 10N$ regression matrix Y_π that depends only on kinematic quantities, the robot dynamic equations can always be rewritten linearly in the standard dynamic parameters as

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

$$\pi^T = (\pi_1^T \quad \pi_2^T \quad \cdots \quad \pi_N^T)$$

- the open kinematic chain structure of the manipulator implies that the i-th dynamic equation can depend only on the standard dynamic parameters of links i to N $\Rightarrow$ Y_π has a block upper triangular structure

$$Y_\pi(q,\dot{q},\ddot{q}) = \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{1N} \\ 0 & Y_{22} & \cdots & Y_{2N} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & Y_{NN} \end{pmatrix}$$

with row vectors $Y_{i,j}$ of size 1×10

Property: element m_{ij} of $M(q)$ is function of at most $(q_{k+1},\cdots,q_N)$, for $k = \min\{i,j\}$, and of the inertial parameters of at most links r to N, with $r = \max\{i,j\}$

Dynamic parameters of robots - 5

- many standard parameters do not appear ("play no role") in the dynamic model of the given robot $\Rightarrow$ the associated columns of Y_π are 0!
- some standard parameters may appear only in fixed combinations with others $\Rightarrow$ the associated columns of Y_π are linearly dependent!
- one can isolate $p \ll 10N$ independent groups of parameters π (associated to p functionally independent columns Y_{indep} of Y_π) and partition matrix Y_π in two blocks, the second containing dependent (or zero) columns as $Y_{dep} = Y_{indep}T$, for a suitable constant $p \times (10N - p)$ matrix T

$$Y_\pi(q,\dot{q},\ddot{q})\,\pi = \begin{pmatrix} Y_{indep} & Y_{dep} \end{pmatrix} \begin{pmatrix} \pi_{indep} \\ \pi_{dep} \end{pmatrix} = \begin{pmatrix} Y_{indep} & Y_{indep}T \end{pmatrix} \begin{pmatrix} \pi_{indep} \\ \pi_{dep} \end{pmatrix}$$

$$= Y_{indep}\left(\pi_{indep} + T\,\pi_{dep}\right) = \boxed{Y(q,\dot{q},\ddot{q})\,a}$$

- these grouped parameters are called dynamic coefficients $a \in \mathbb{R}^p$, "the only that matter" in robot dynamics (= base parameters by W. Khalil)
- the minimal number p of dynamic coefficients that is needed can also be checked numerically (see later → Identification)

Linear parameterization of robot dynamics

it is always possible to rewrite the dynamic model in the form

regression matrix

a = vector of dynamic coefficients

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

$N \times p$

$p \times 1$

e.g., the heuristic grouping (found by inspection) on a 2R planar robot

$$\begin{pmatrix} \ddot{q}_1 & c_2(2\ddot{q}_1+\ddot{q}_2) - s_2(\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2) & \ddot{q}_2 & c_1 & c_{12} \\ 0 & c_2\ddot{q}_1 + s_2\dot{q}_1^2 & \ddot{q}_1+\ddot{q}_2 & 0 & c_{12} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

$$\begin{aligned} a_1 &= I_{c1,zz} + m_1 d_1^2 + I_{c2,zz} + m_2 d_2^2 + m_2 l_1^2 \\ a_2 &= m_2 l_1 d_2 \\ a_3 &= I_{c2,zz} + m_2 d_2^2 \\ a_4 &= g_0(m_1 d_1 + m_2 l_1) \\ a_5 &= g_0 m_2 d_2 \end{aligned}$$

NOTE: 4 more coefficients are added when including the coefficients $F_{V,i}$ and $F_{S,i}$ of viscous and dry friction at joints ("decoupled" terms appearing only in the respective equations $i = 1,2$)

Linear parametrization
of a 2R planar robot ($N = 2$)

- being the kinematics known (i.e., l_1 and g_0), the number of dynamic coefficients can be reduced since we can merge the two coefficients

  $a_2 = m_2 l_1 d_2$ & $a_5 = g_0 m_2 d_2 \quad \Rightarrow \quad a_2 = m_2 d_2$ (factoring out l_1 and g_0)

- therefore, after regrouping, $\boldsymbol{p = 4}$ dynamic coefficients are sufficient

$$\begin{pmatrix} \ddot{q}_1 & l_1 c_2 (2\ddot{q}_1 + \ddot{q}_2) - l_1 s_2 (\dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) + g_0 c_{12} & \ddot{q}_2 & g_0 c_1 \\ 0 & l_1 (c_2 \ddot{q}_1 + s_2 \dot{q}_1^2) + g_0 c_{12} & \ddot{q}_1 + \ddot{q}_2 & 0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = Y\, a = u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

$$a_1 = I_{c1,zz} + m_1 d_1^2 + I_{c2,zz} + m_2 d_2^2 + m_2 l_1^2 \qquad a_3 = I_{c2,zz} + m_2 d_2^2$$

$$a_2 = m_2 d_2 \qquad a_4 = m_1 d_1 + m_2 l_1$$

- this (minimal) linear parametrization of robot dynamics is not unique, both in terms of the chosen set of dynamic coefficients a and for the associated regression matrix Y
  - a systematic procedure for its derivation would be preferable

Linear parametrization
of a 2R planar robot ($N = 2$)

- as alternative to the previous heuristic method, apply the general procedure
  - $10N = 20$ standard parameters are defined for the two links
  - from the assumptions made on CoM locations, only 5 such parameters actually appear, namely (with $d_i = r_{ci,x}$)

link 1: $m_1 d_1 \quad I_{1,zz} = I_{c1,zz} + m_1 d_1^2$ link 2: $m_2 \quad m_2 d_2 \quad I_{2,zz} = I_{c2,zz} + m_2 d_2^2$

  - in the 2×5 matrix Y_π, the 3rd column (associated to $\boxed{m_2}$) is $Y_{\pi 3} = Y_{\pi 1} l_1 + Y_{\pi 2} l_1^2$
  - after regrouping/reordering, $\boldsymbol{p = 4}$ dynamic coefficients are again sufficient

$$\begin{pmatrix} g_0 c_1 & \ddot{q}_1 & l_1 c_2 (2\ddot{q}_1 + \ddot{q}_2) - l_1 s_2 (\dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) + g_0 c_{12} & \ddot{q}_1 + \ddot{q}_2 \\ 0 & 0 & l_1 (c_2 \ddot{q}_1 + s_2 \dot{q}_1^2) + g_0 c_{12} & \ddot{q}_1 + \ddot{q}_2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = Y\, a = u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

$$a_1 = m_1 d_1 + \boxed{m_2} l_1 \qquad a_2 = I_{1,zz} + \boxed{m_2} l_1^2 = \left(I_{c1,zz} + m_1 d_1^2\right) + m_1 l_1^2 \qquad \begin{array}{c} a_3 = m_2 d_2 \\ a_4 = I_{2,zz} = I_{c2,zz} + m_2 d_2^2 \end{array}$$

- determining a minimal parameterization (i.e., minimizing p) is important for
  - experimental identification of dynamic coefficients
  - adaptive/robust control design in the presence of uncertain parameters

Identification of dynamic coefficients

- in order to "use" the model, one needs to know the numeric values of the robot dynamic coefficients
  - robot manufacturers provide at most only a few principal dynamic parameters (e.g., link masses)
- estimates can be found with CAD tools (e.g., assuming uniform mass)
- friction coefficients are (slowly) varying over time
  - lubrication of joints/transmissions
- for an added payload (attached to the E-E)
  - a change in the 10 dynamic parameters of last link
  - this implies a variation of (almost) all robot dynamic coefficients
- preliminary identification experiments are needed
  - robot in motion (dynamic issues, not just static or geometric ones!)
  - only the robot dynamic coefficients can be identified (and not all the link standard parameters!)

Identification experiments

1. choose a motion trajectory that is sufficiently "exciting", i.e.,
   - explores the robot workspace and involves all components in the robot dynamic model
   - is periodic, with multiple frequency components
2. execute this motion (approximately) by means of a control law
   - taking advantage of any available information on the robot model
   - often $u = K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})$ (PD, no model information used)
3. measure q (encoder) and, if available, also $\dot{q}$ in n_c time instants
   - joint velocity and acceleration can be later estimated off line by numerical differentiation (use of non-causal filters is feasible)
4. with such measures/estimates, evaluate the regression matrix Y (on the left) and use the applied commands (on the right) in the expression

$$Y\big(q(t_k), \dot{q}(t_k), \ddot{q}(t_k)\big)\, a = u(t_k) \qquad k = 1, \cdots, n_c$$

Least Squares (LS) identification

- set up the system of linear equations

$$n_c \times N \left(\begin{array}{c} Y\big(q(t_1), \dot{q}(t_1), \ddot{q}(t_1)\big) \\ \vdots \\ Y\left(q(t_{n_c}), \dot{q}(t_{n_c}), \ddot{q}(t_{n_c})\right) \end{array} \right) a = \left(\begin{array}{c} u(t_1) \\ \vdots \\ u(t_{n_c}) \end{array} \right) \Leftrightarrow \boxed{\bar{Y} a = \bar{u}}$$

- sufficiently "exciting" trajectories, large enough number of samples ($n_c \times N \gg p$), and their suitable selection/position, guarantee rank($\bar{Y}$) $= p$ (full column rank)
- solution by pseudoinversion of matrix $\bar{Y}$

$$a = \bar{Y}^{\#} \bar{u} = (\bar{Y}^T \bar{Y})^{-1} \bar{Y}^T \bar{u} \quad (\in \mathbb{R}^p)$$

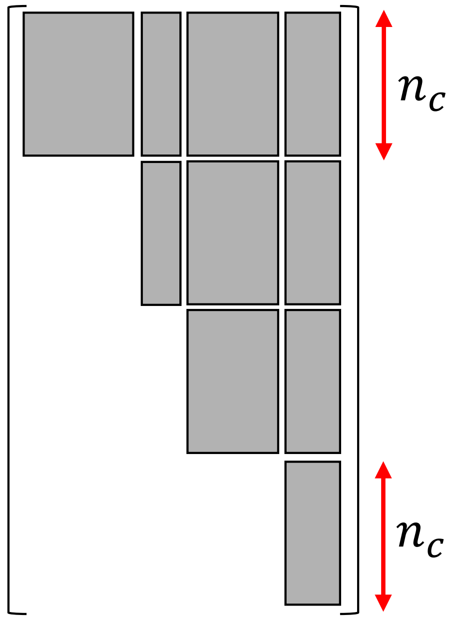
- one can also use a weighted pseudoinverse, to take into account different levels of noise in the collected measures

Additional remarks on LS identification

- it is convenient to preserve the block (upper) triangular structure of the regression matrix, by "stacking" all time evaluations in row by row sequence of the original Y matrix

$$N\times\left\{\begin{pmatrix} Y_1\big(q(t_1),\dot q(t_1),\ddot q(t_1)\big)\\ \vdots\\ Y_1\big(q(t_{n_c}),\dot q(t_{n_c}),\ddot q(t_{n_c})\big)\\ Y_2\big(q(t_1),\dot q(t_1),\ddot q(t_1)\big)\\ \vdots\\ Y_2\big(q(t_{n_c}),\dot q(t_{n_c}),\ddot q(t_{n_c})\big)\\ \vdots\\ Y_N\big(q(t_1),\dot q(t_1),\ddot q(t_1)\big)\\ \vdots\\ Y_N\big(q(t_{n_c}),\dot q(t_{n_c}),\ddot q(t_{n_c})\big)\end{pmatrix}\right. a = \begin{pmatrix} u_1(t_1)\\ \vdots\\ u_1(t_{n_c})\\ u_2(t_1)\\ \vdots\\ u_2(t_{n_c})\\ \vdots\\ u_N(t_1)\\ \vdots\\ u_N(t_{n_c})\end{pmatrix} \Leftrightarrow \boxed{\bar Y a = \bar u}$$

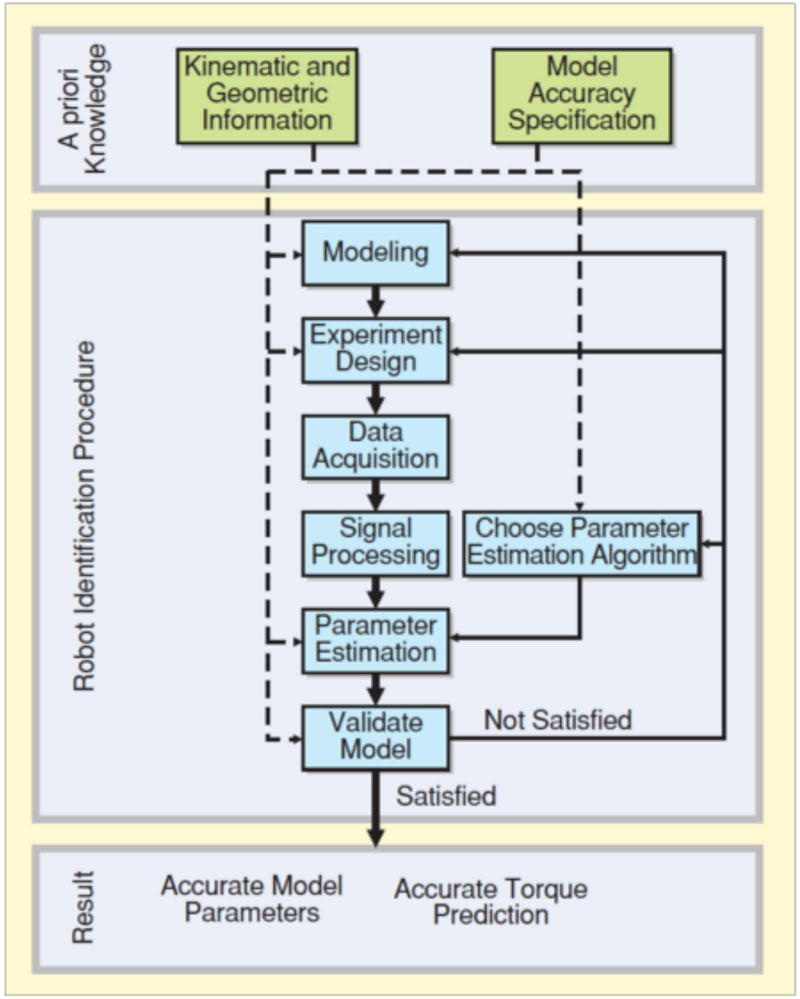
(first and last blocks of rows: n_c rows each)

$\bar Y =$ 

- numerical check of full column rank is more robust $\Leftrightarrow$ rank = p (# of col's)

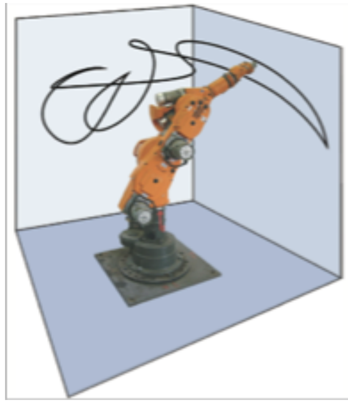
- further practical hints
  - outlier data can be eliminated in advance (when building Y)
  - when complex friction models are not included in Ya, discard the data collected at joint velocities that are close to zero

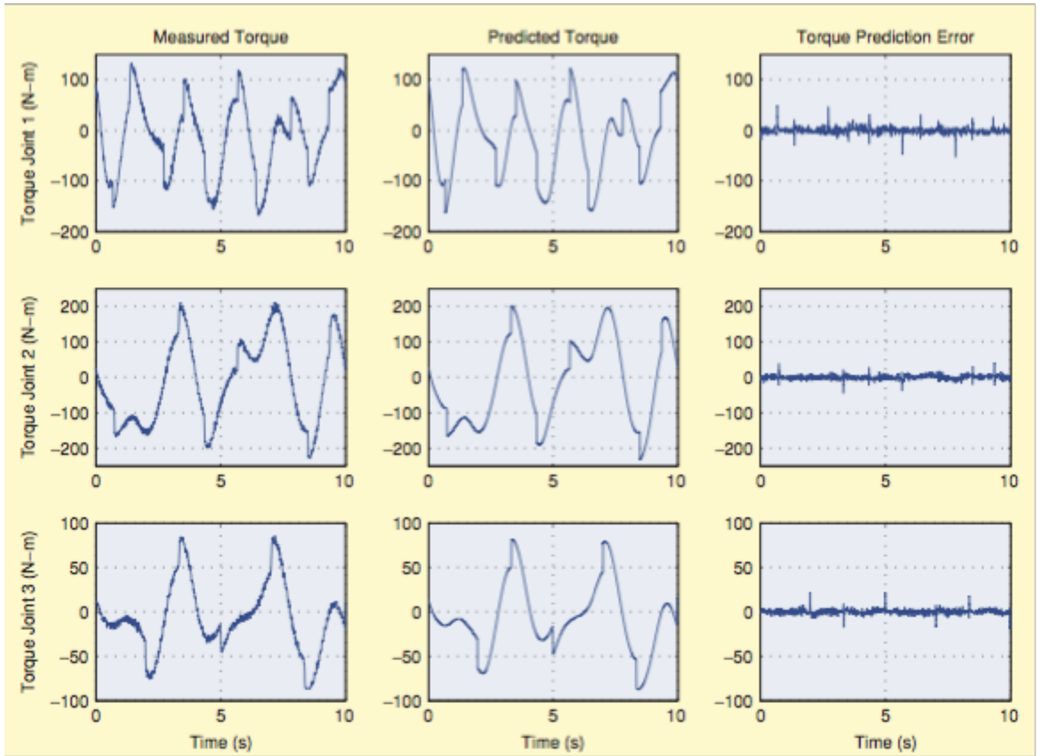
Summary on dynamic identification



J. Swevers, W. Verdonck, and J. De Schutter:
"Dynamic model identification for industrial robots"
IEEE Control Systems Mag., Oct 2007

KUKA IR 361 robot and optimal excitation trajectory





results after identification (first three joints only)

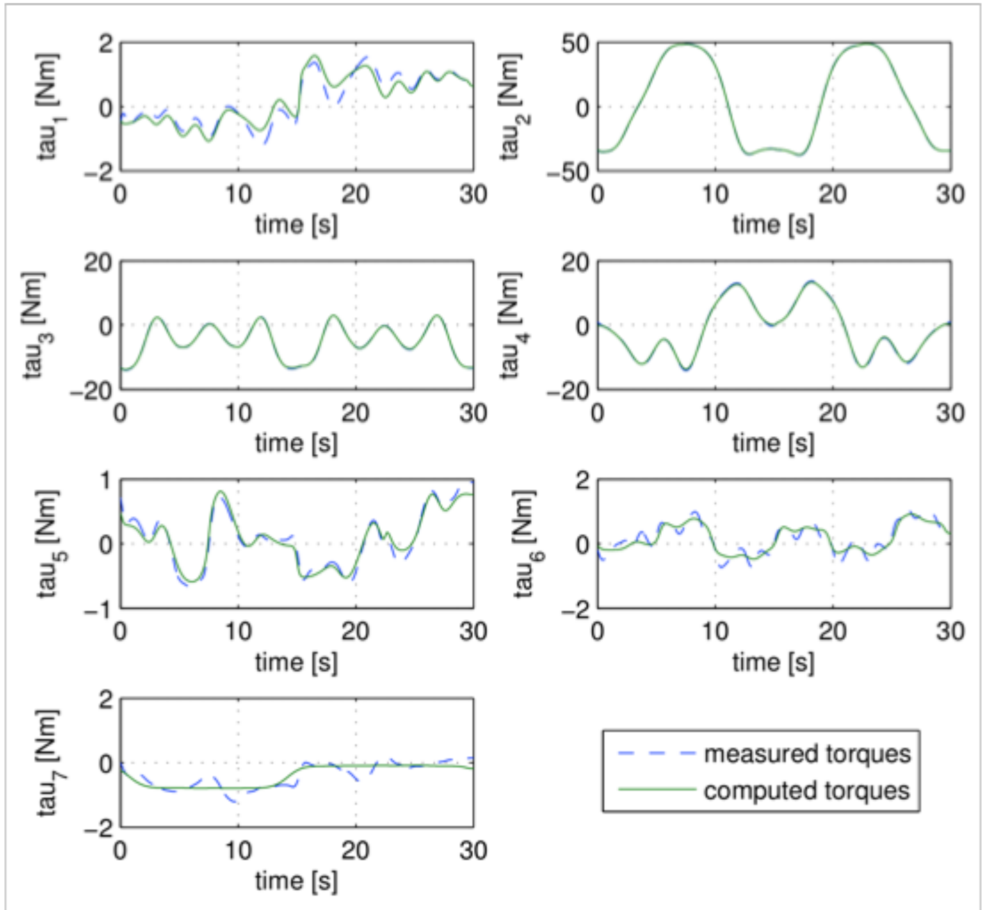
Dynamic identification of KUKA LWR4

video

data acquisition for identification

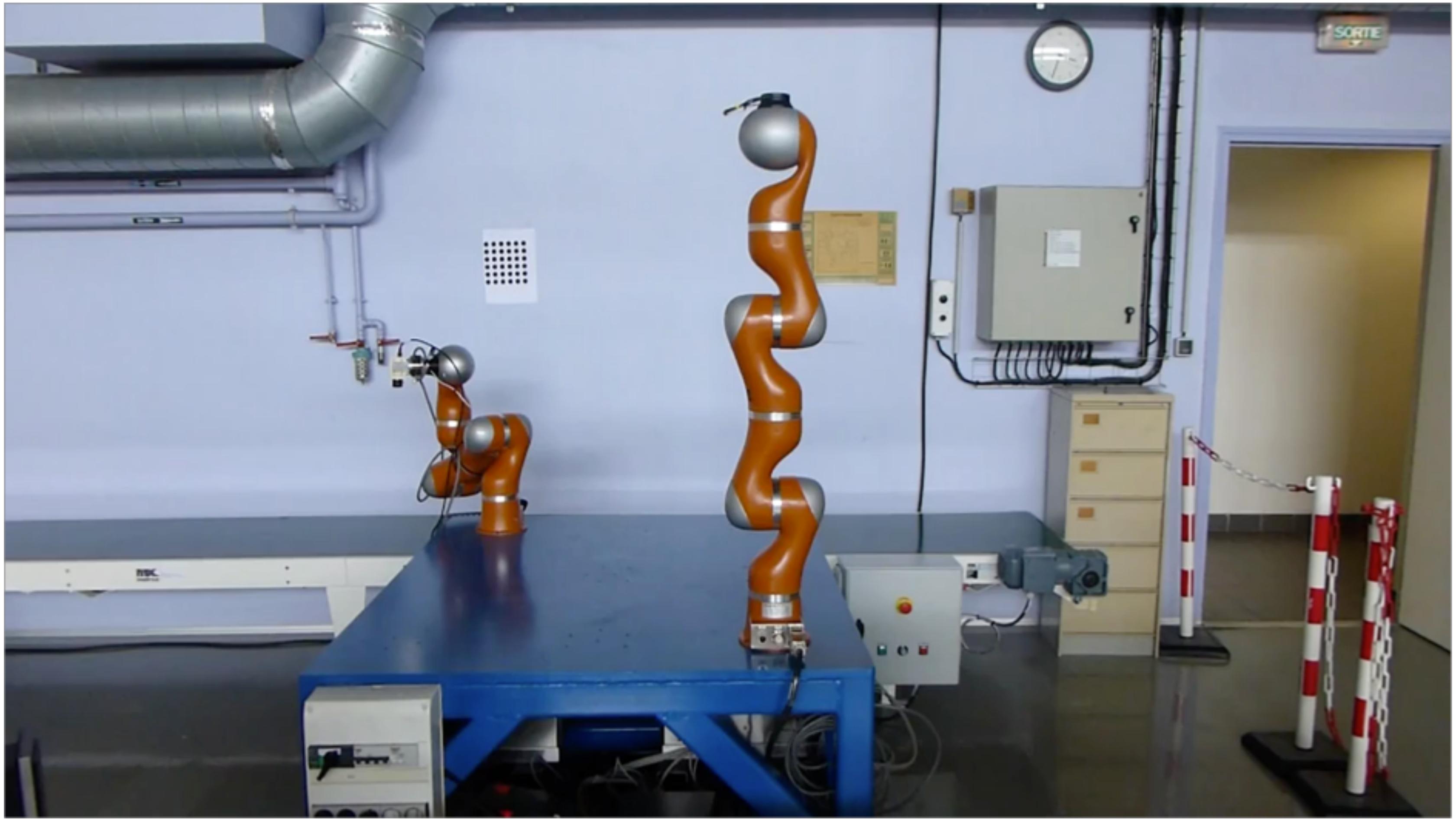
dynamic coefficients: 30 inertial, 12 for gravity

C. Gaz, F. Flacco, A. De Luca:
"Identifying the dynamic model used by the KUKA LWR: A reverse engineering approach"
IEEE ICRA 2014, Hong Kong, June 2014



validation after identification (for all 7 joints):
on new desired trajectories, compare
torques computed with the identified model
and torques measured by joint torque sensors

Dynamic identification of KUKA LWR4



video

using more dynamic robot motions for model identification

J. Hollerbach, W. Khalil, M. Gautier: "Ch. 6: Model Identification", Springer Handbook of Robotics (2nd Ed), 2016
free access to multimedia extension: http://handbookofrobotics.org

Adding a payload to the robot

- in several industrial applications, changes in the robot payload are often needed
  - using different tools for various technological operations such as polishing, welding, grinding, ...
  - pick-and-place tasks of objects having unknown mass
- what is the rule of change for dynamic parameters when there is an additional payload?
  - do we obtain again a linearly parameterized problem?
  - does this property rely on some specific choice of reference frames (e.g., conventional or modified D-H)?

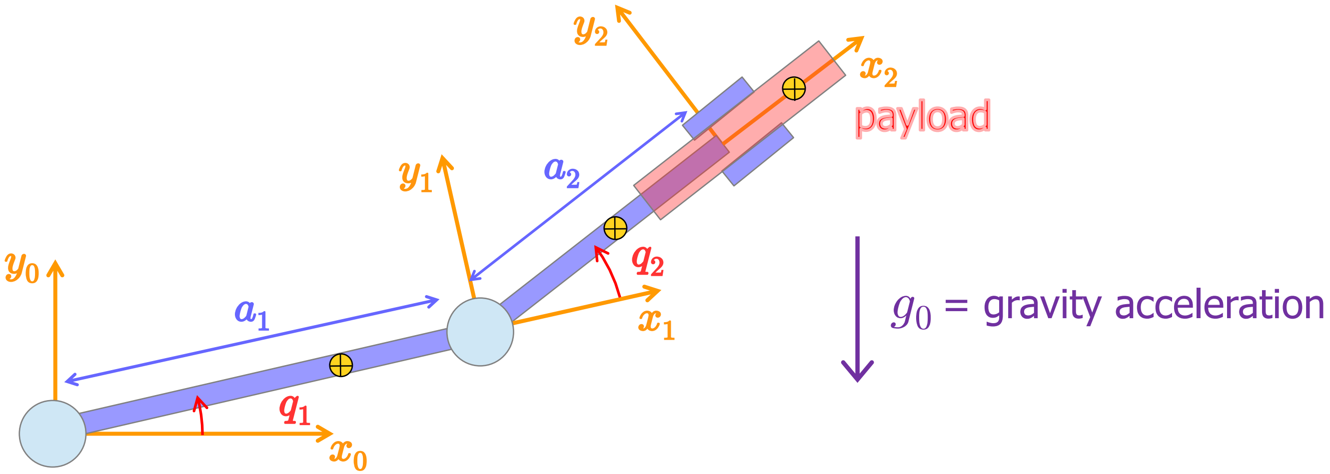
Rule of change in dynamic parameters

- only the dynamic parameters of the link where a load is added will change (typically, added to the last one –link n– as payload)
  - last link dynamic parameters: m_n (mass), $\boldsymbol{c}_n = (c_{nx} c_{ny} c_{nz})^T$ (center of mass), $\boldsymbol{I}_n$ (inertia tensor expressed w.r.t. frame n)
  - payload dynamic parameters: m_L (mass), $\boldsymbol{c}_L = (c_{Lx} c_{Ly} c_{Lz})^T$ (center of mass), $\boldsymbol{I}_L$ (inertia tensor expressed w.r.t. frame n)
- mass $\quad m_n \to m_n + m_L$
- center of mass $\quad c_{ni} m_n \to \dfrac{c_{ni} m_n + c_{Li} m_L}{m_n + m_L}(m_n + m_L) = c_{ni} m_n + c_{Li} m_L$

  (weighted average) where $i = x,\, y,\, z$
- inertia tensor $\quad \boldsymbol{I}_n \to \boldsymbol{I}_n + \boldsymbol{I}_L$

  valid only if tensors are expressed w.r.t. the same reference frame (i.e., frame n)!
- linear parametrization is preserved with any kinematic convention (the parameters of the last link will always appear in the form shown above)

Example: 2R planar robot with payload



unloaded robot dynamics $\boldsymbol{Y\pi} = \boldsymbol{\tau}$

loaded robot dynamics $\boldsymbol{Y\pi}^L = \boldsymbol{\tau}^L$

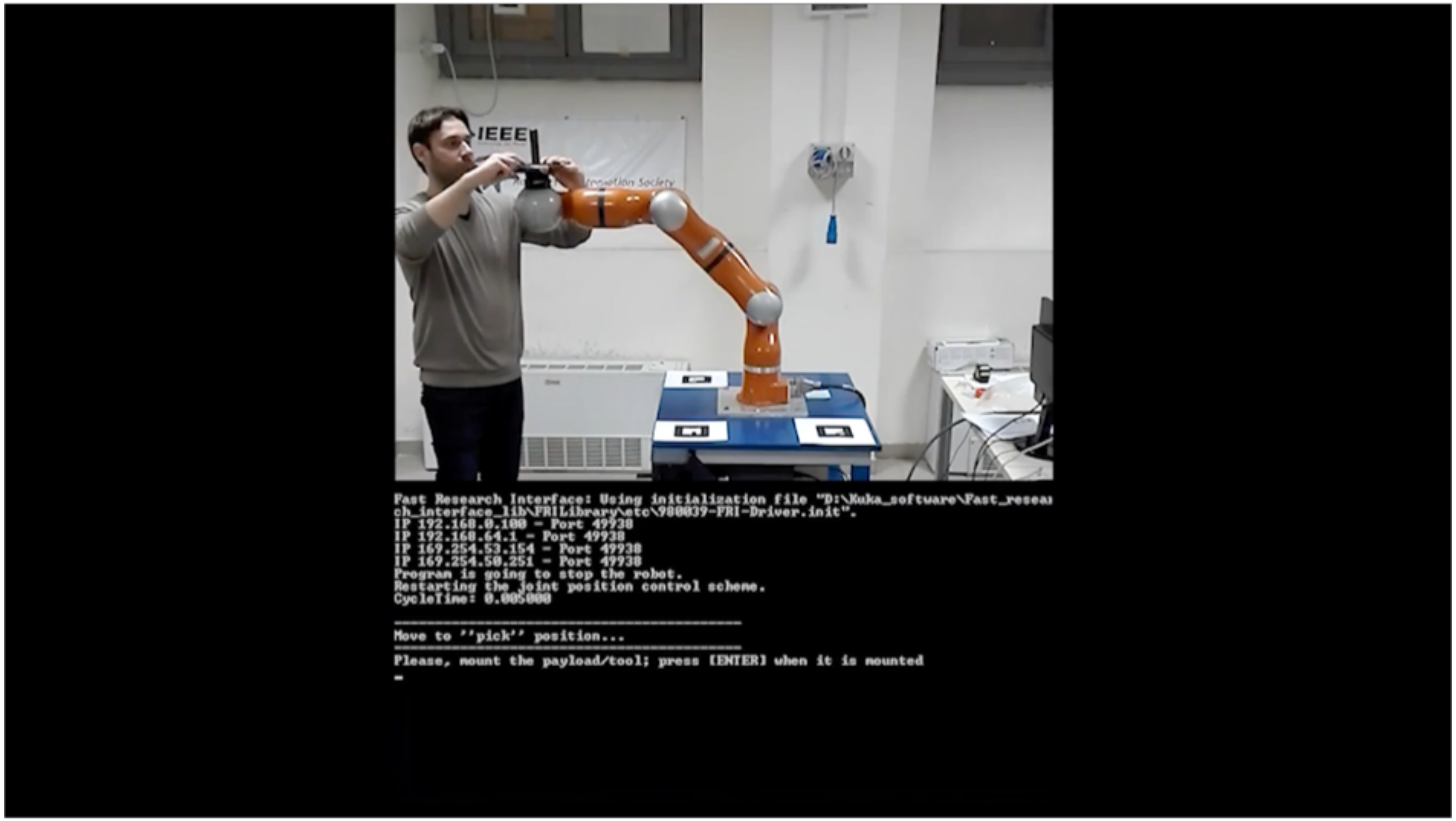
$$\boldsymbol{\pi} = \begin{pmatrix} \frac{1}{2}\left(m_2 a_2^2 + I_{2zz}\right) + a_2 c_{2x} m_2 \\ c_{2x} m_2 + a_2 m_2 \\ c_{2y} m_2 \\ \frac{1}{2}\left(I_{1zz} + a_1^2 m_1 + a_1^2 m_2\right) + a_1 c_{1x} m_1 \\ c_{1x} m_1 + a_1 m_1 + a_1 m_2 \\ c_{1y} m_1 \end{pmatrix} \Rightarrow \boldsymbol{\pi}^L = \begin{pmatrix} \frac{1}{2}\left(a_2^2\left(m_2 + m_L\right) + I_{2zz} + I_{Lzz}\right) + a_2\left(c_{2x} m_2 + c_{Lx} m_L\right) \\ c_{2x} m_2 + c_{Lx} m_L + a_2\left(m_2 + m_L\right) \\ c_{2y} m_2 + c_{Ly} m_L \\ \frac{1}{2}\left(I_{1zz} + a_1^2 m_1 + a_1^2\left(m_2 + m_L\right)\right) + a_1 c_{1x} m_1 \\ c_{1x} m_1 + a_1 m_1 + a_1\left(m_2 + m_L\right) \\ c_{1y} m_1 \end{pmatrix}$$

Note 1: position of the center of mass of the two links and of the payload may also be asymmetric

Note 2: link inertia & center of mass are expressed in the DH kinematic frame attached to the link (e.g., I_{2zz} is the inertia of the second link around the axis z_2)

Validation on the KUKA LWR4 robot

video



C. Gaz, A. De Luca: "Payload estimation based on identified coefficients of robot dynamics – with an application to **collision detection**" IEEE IROS 2017, Vancouver, September 2017

see the block of slides!

Use of the dynamic model
Inverse dynamics

- given a desired trajectory $q_d(t)$
  - twice differentiable ($\exists\ \ddot{q}_d(t)$)
  - possibly obtained from a task/Cartesian trajectory $r_d(t)$, by (differential) kinematic inversion

the input torque needed to execute this motion (in free space) is

$$\tau_d = (M(q_d) + B_m)\ddot{q}_d + c(q_d, \dot{q}_d) + g(q_d) + F_V \dot{q}_d + F_S \operatorname{sgn}(\dot{q}_d)$$

- useful also for control (e.g., nominal feedforward)
- however, this way of performing the algebraic computation ($\forall t$) is not efficient when using the above Lagrangian approach
- symbolic terms grow much longer, quite rapidly for larger N
- in real time, numerical computation is based on Newton-Euler method

Dynamic scaling of trajectories

uniform time scaling of motion

- given a smooth original trajectory $q_d(t)$ of motion for $t \in [0, T]$
  - suppose to rescale time as $t \to r(t)$ (a strictly *increasing* function of t)
  - in the new time scale, the scaled trajectory $q_s(r)$ satisfies

$$q_d(t) = q_s(r(t)) \quad \Rightarrow \quad \dot{q}_d(t) = \frac{dq_d}{dt} = \frac{dq_s}{dr}\frac{dr}{dt} = q_s'\,\dot{r}$$

same path executed
(at different instants of time)

$$\ddot{q}_d(t) = \frac{d\dot{q}_d}{dt} = \left(\frac{dq_s'}{dr}\frac{dr}{dt}\right)\dot{r} + q_s'\ddot{r} = q_s''\dot{r}^2 + q_s'\ddot{r}$$

  - uniform scaling of the trajectory occurs when $r(t) = k\,t$

$$\dot{q}_d(t) = k q_s'(kt) \qquad \ddot{q}_d(t) = k^2 q_s''(kt)$$

Q: what is the new input torque needed to execute the scaled trajectory?
(suppose dissipative terms can be neglected)

Dynamic scaling of trajectories

inverse dynamics under uniform time scaling

- the new torque could be recomputed through the inverse dynamics, for every $r = kt \in [0, T'] = [0, kT]$ along the scaled trajectory, as

$$\tau_s(kt) = M(q_s)q_s'' + c(q_s, q_s') + g(q_s)$$

- however, being the dynamic model linear in the acceleration and quadratic in the velocity, it is

$$\tau_d(t) = M(q_d)\ddot{q}_d + c(q_d, \dot{q}_d) + g(q_d) = M(q_s)k^2 q_s'' + c(q_s, kq_s') + g(q_s)$$
$$= k^2\big(M(q_s)q_s'' + c(q_s, q_s')\big) + g(q_s) = k^2\big(\tau_s(kt) - g(q_s)\big) + g(q_s)$$

- thus, saving separately the total torque $\tau_d(t)$ and gravity torque $g_d(t)$ in the inverse dynamics computation along the original trajectory, the new input torque is obtained directly as

$$\boxed{\tau_s(kt) = \frac{1}{k^2}\big(\tau_d(t) - g(q_d(t))\big) + g(q_d(t))}$$

$k > 1$: slow down $\Rightarrow$ reduce torque

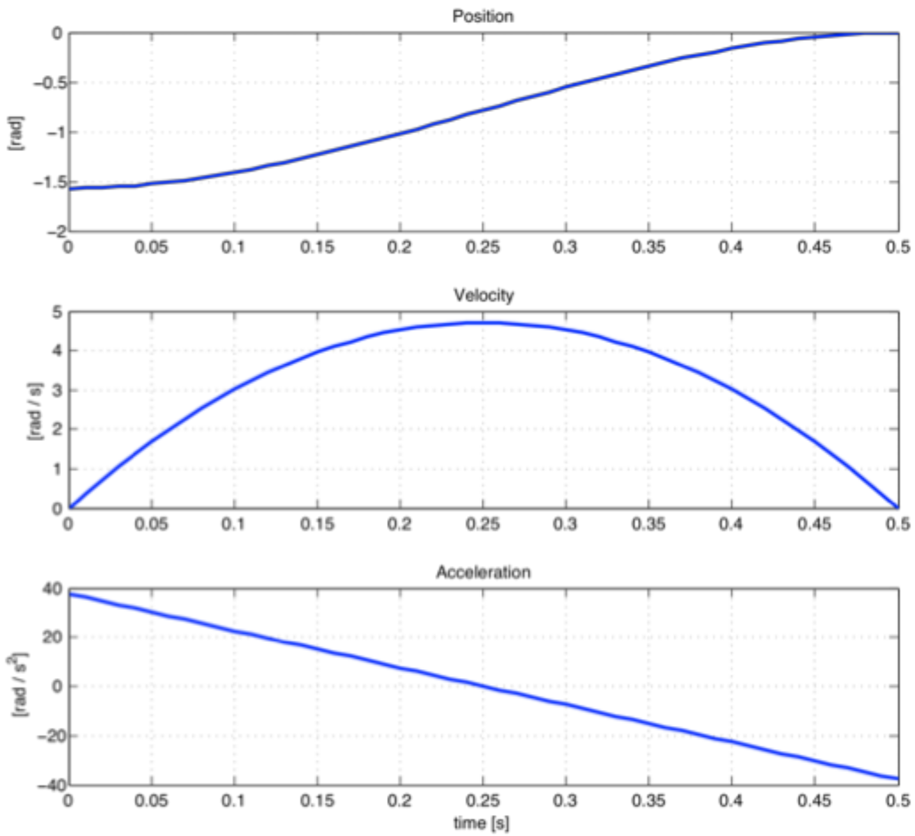
$k < 1$: speed up $\Rightarrow$ increase torque

gravity term (only position-dependent): does NOT scale!

Dynamic scaling of trajectories

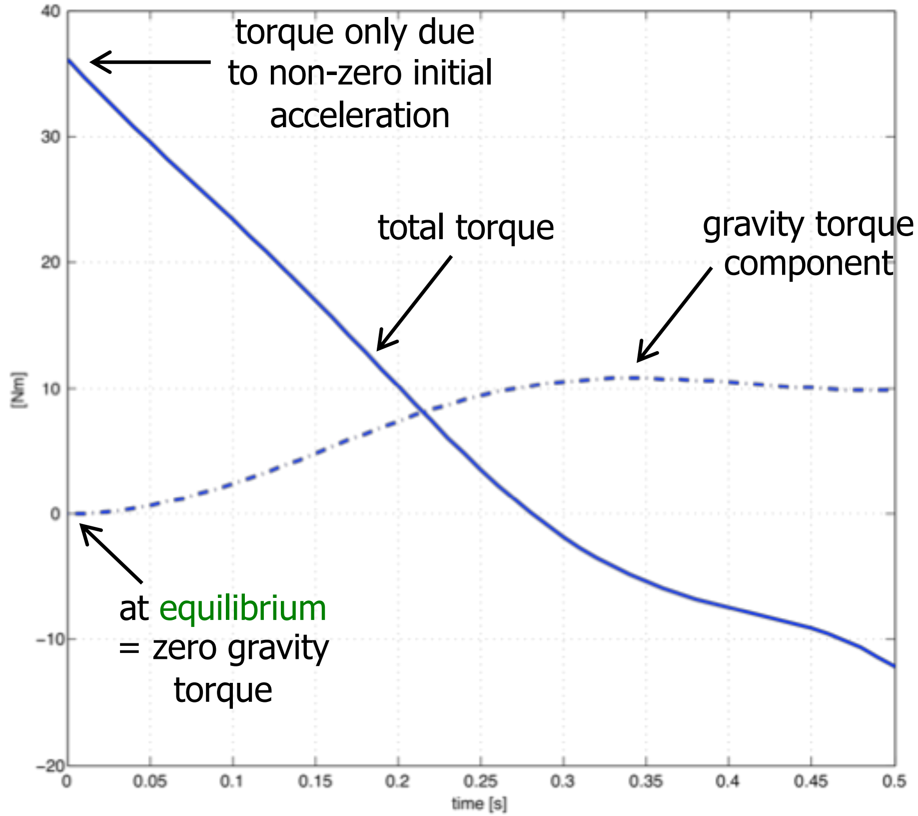
numerical example

- rest-to-rest motion with cubic polynomials for planar 2R robot under gravity (from downward equilibrium to horizontal link 1 & upward vertical link 2)
- original trajectory lasts $T = 0.5$ s (but maybe violates the torque limit at joint 1)



for both joints

only joint 1 torque is shown

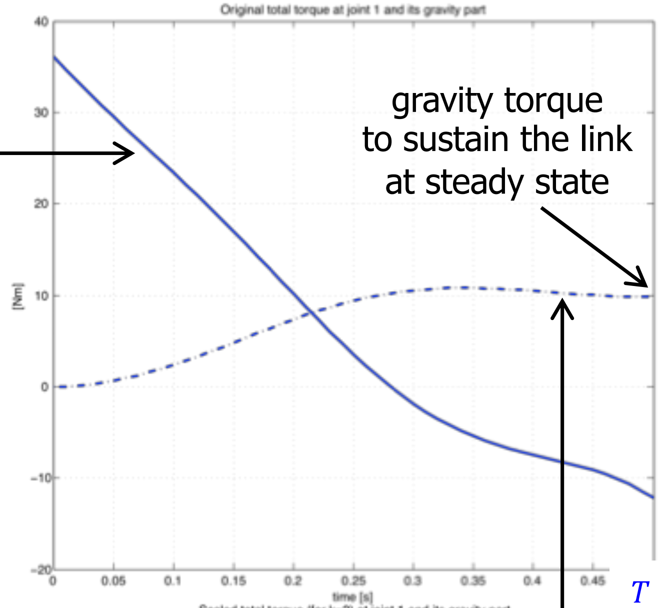


Dynamic scaling of trajectories

numerical example

original total torque

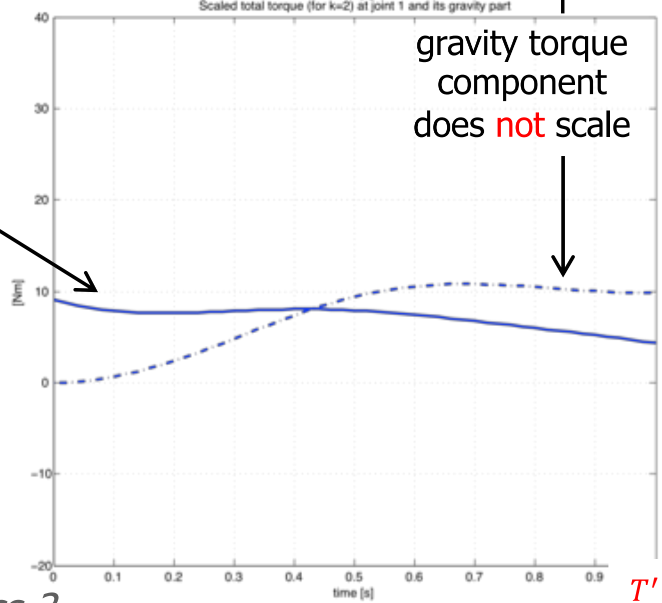
gravity torque to sustain the link at steady state



$T = 0.5$ s

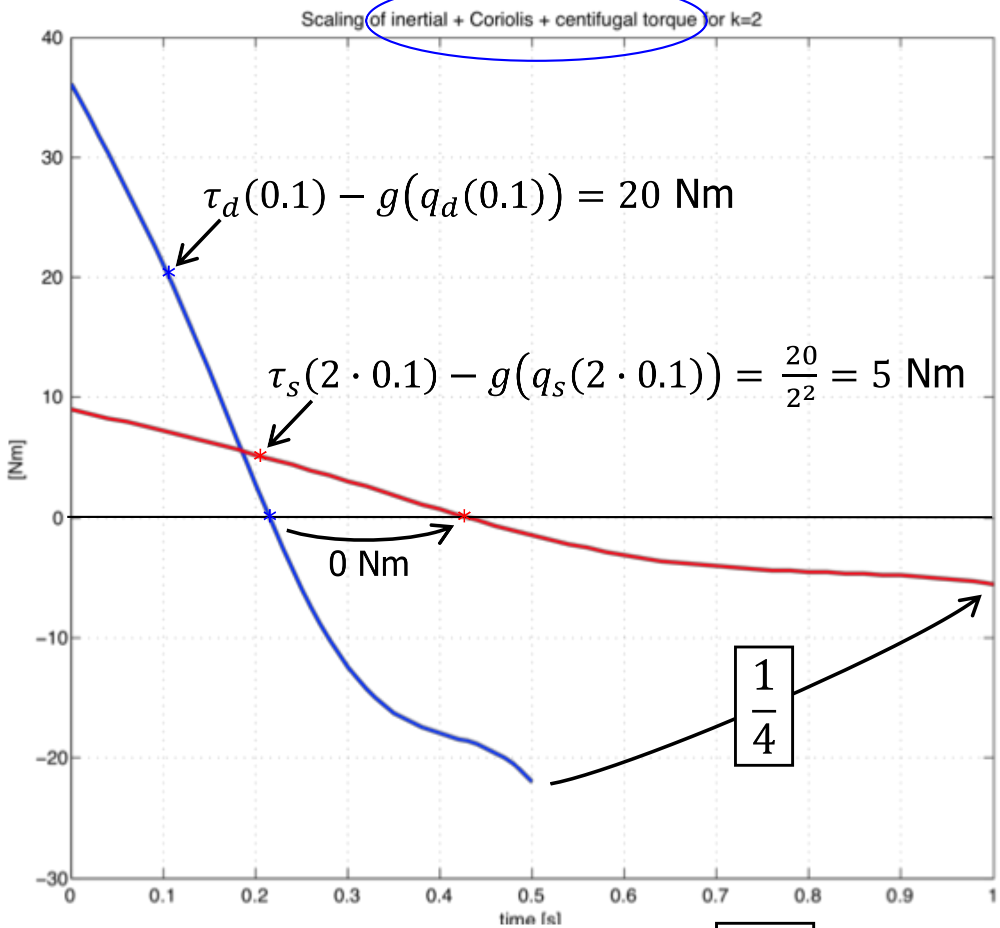
gravity torque component does not scale

scaled total torque



$T' = 1$ s

- scaling with $k = 2$ (slower) $\rightarrow T' = 1$ s



$\tau_d(0.1) - g\big(q_d(0.1)\big) = 20$ Nm

$\tau_s(2 \cdot 0.1) - g\big(q_s(2 \cdot 0.1)\big) = \frac{20}{2^2} = 5$ Nm

0 Nm

$\boxed{\frac{1}{4}}$

$T = 0.5$ s $\longrightarrow \boxed{k = 2} \longrightarrow T = 1$ s

State equations
Direct dynamics

Lagrangian dynamic model

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

N differential 2nd order equations

defining the vector of state variables as $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} \in \mathbb{R}^{2N}$

state equations

$$\boxed{\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -M^{-1}(x_1)[c(x_1,x_2) + g(x_1)] \end{pmatrix} + \begin{pmatrix} 0 \\ M^{-1}(x_1) \end{pmatrix} u}$$

$$= \underbrace{f(x)}_{2N \times 1} + \underbrace{G(x)}_{2N \times N} u$$

$2N$ differential 1st order equations

another choice... $\boxed{\tilde{x} = \begin{pmatrix} q \\ M(q)\dot{q} \end{pmatrix}}$ $\longleftarrow$ generalized momentum

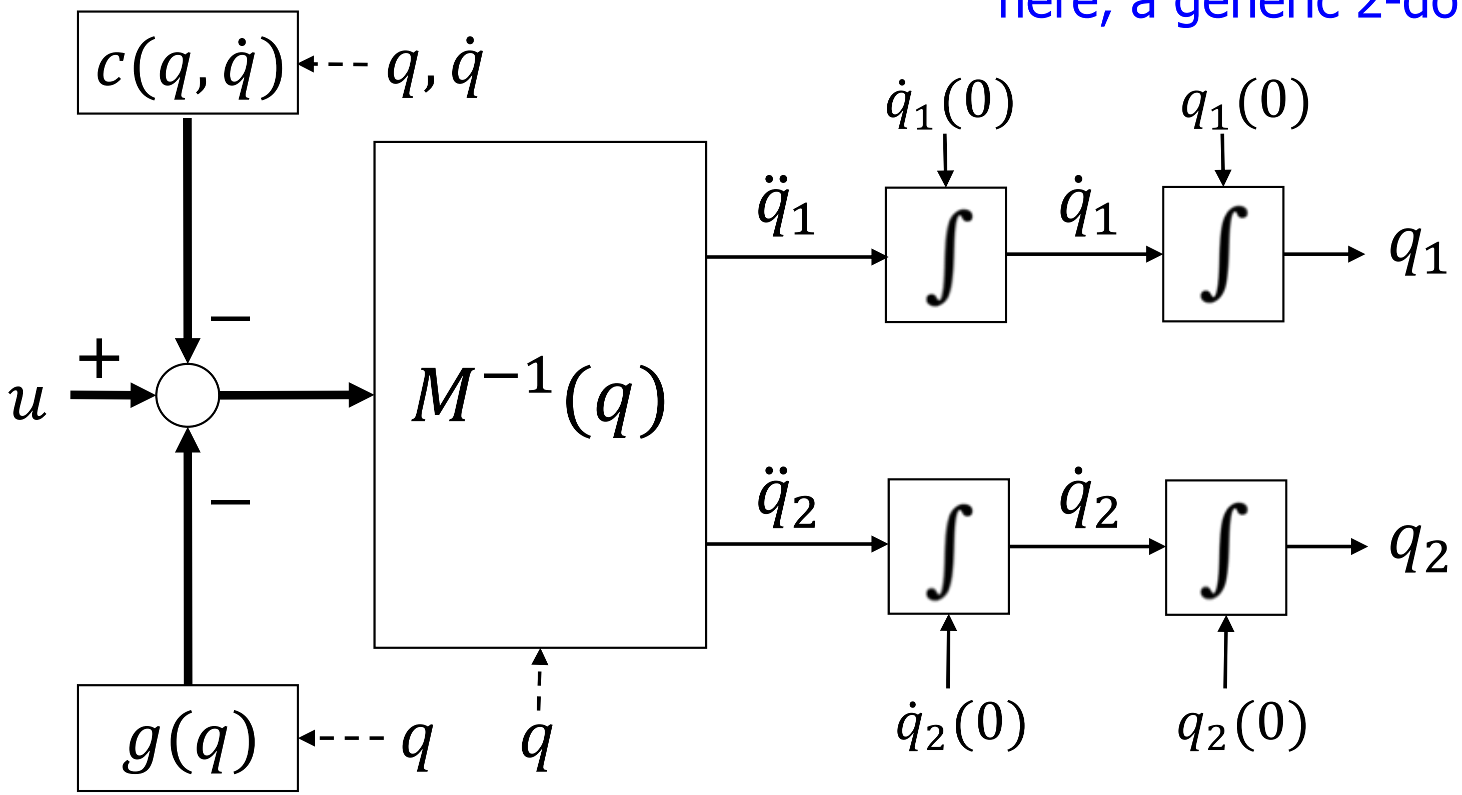
$\dot{\tilde{x}} = \ldots$ (do it as exercise)

Dynamic simulation

Simulink block scheme

here, a generic 2-dof robot

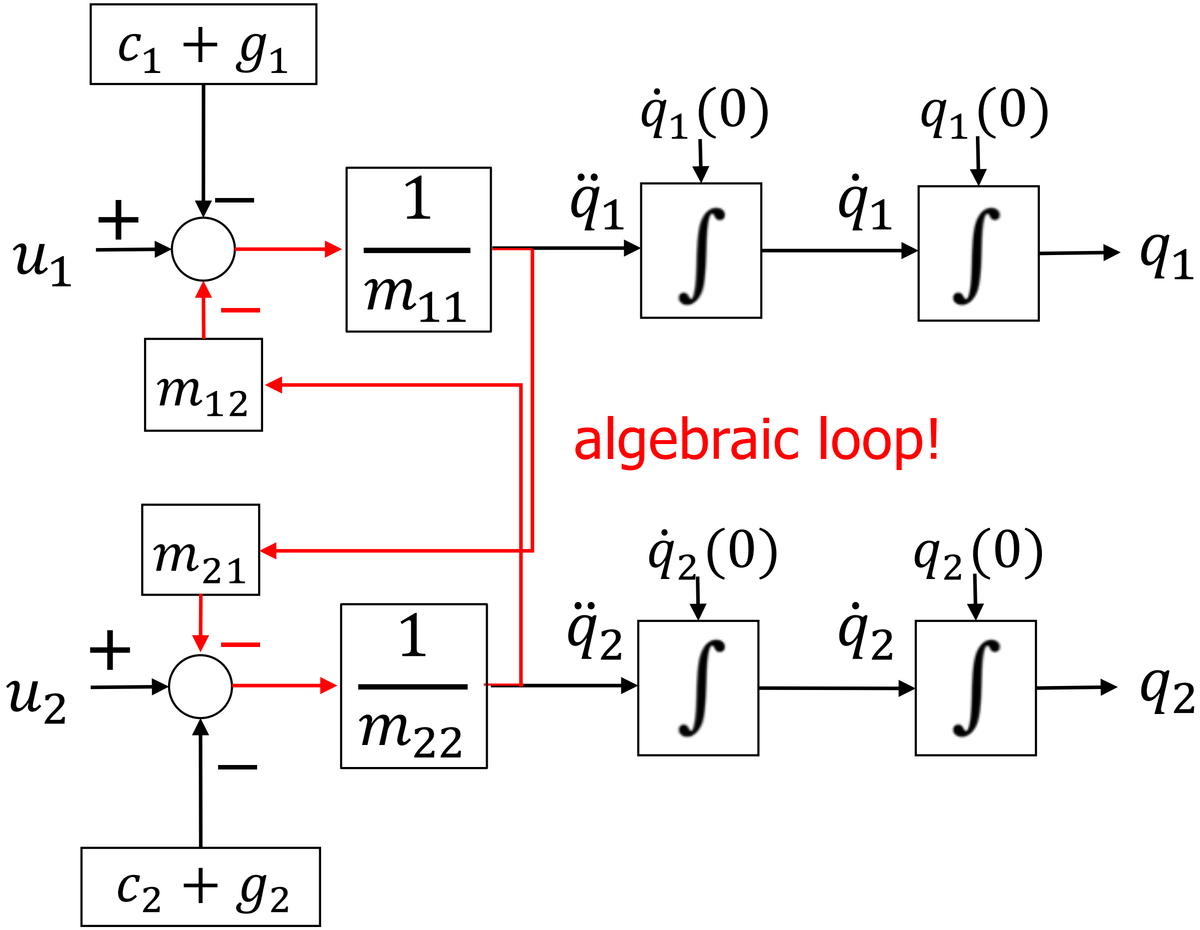
input torque command (open-loop or in feedback)



including "inv(M)"

- initialization (dynamic coefficients and initial state)
- calls to (user-defined) Matlab functions for the evaluation of model terms
- choice of a numerical integration method (and of its parameters)

… an incorrect realization



causality principle is violated!!

inversion of the robot inertia matrix *in toto* is needed
(and not *only* of its elements on the diagonal...)

Approximate linearization

- we can derive a linear dynamic model of the robot, which is valid locally around a given operative condition
  - useful for analysis, design, and gain tuning of linear (or, the linear part of) control laws
  - approximation by Taylor series expansion, up to the first order
  - linearization around a (constant) equilibrium state or along a (nominal, time-varying) equilibrium trajectory
  - usually, we work with (nonlinear) state equations; for mechanical systems, it is more convenient to directly use the 2nd order model
    - same result, but easier derivation

equilibrium state $(q, \dot{q}) = (q_e, 0)$ $[\, \ddot{q} = 0 \,]$ $\Rightarrow$ $g(q_e) = u_e$

equilibrium trajectory $(q, \dot{q}) = (q_d(t), \dot{q}_d(t))$ $[\, \ddot{q} = \ddot{q}_d(t) \,]$

$$\Rightarrow \quad M(q_d)\ddot{q}_d + c(q_d, \dot{q}_d) + g(q_d) = u_d$$

Linearization at an equilibrium state

- variations around an equilibrium state

$$q = q_e + \Delta q \quad \dot{q} = \dot{q}_e + \dot{\Delta q} = \dot{\Delta q} \quad \ddot{q} = \ddot{q}_e + \ddot{\Delta q} = \ddot{\Delta q} \quad u = u_e + \Delta u$$

- keeping into account the quadratic dependence of c terms on velocity (thus, neglected around the zero velocity)

$$M(q_e)\ddot{\Delta q} + \cancel{g(q_e)} + \underbrace{\left.\frac{\partial g}{\partial q}\right|_{q=q_e}}_{G(q_e)} \Delta q + \cancel{\mathrm{o}\left(\|\Delta q\|, \|\dot{\Delta q}\|\right)} = \cancel{u_e} + \Delta u$$

infinitesimal terms
of second or higher order

- in state-space format, with $\Delta x = \begin{pmatrix} \Delta q \\ \dot{\Delta q} \end{pmatrix}$

$$\dot{\Delta x} = \begin{pmatrix} 0 & I \\ -M^{-1}(q_e)G(q_e) & 0 \end{pmatrix} \Delta x + \begin{pmatrix} 0 \\ M^{-1}(q_e) \end{pmatrix} \Delta u = A\,\Delta x + B\,\Delta u$$

Linearization along a trajectory

- variations around an equilibrium trajectory

$$q = q_d + \Delta q \qquad \dot{q} = \dot{q}_d + \Delta\dot{q} \qquad \ddot{q} = \ddot{q}_d + \Delta\ddot{q} \qquad u = u_d + \Delta u$$

- developing to 1st order the terms in the dynamic model ...

$$M(q_d + \Delta q)\left(\ddot{q}_d + \Delta\ddot{q}\right) + c(q_d + \Delta q, \dot{q}_d + \Delta\dot{q}) + g(q_d + \Delta q) = u_d + \Delta u$$

$$M(q_d + \Delta q) \cong M(q_d) + \sum_{i=1}^{N} \left.\frac{\partial M_i}{\partial q}\right|_{q=q_d} e_i^T \Delta q$$

i-th row of the identity matrix

$$g(q_d + \Delta q) \cong g(q_d) + G(q_d)\Delta q$$

$$c\left(q_d + \Delta q, \dot{q}_d + \Delta\dot{q}\right) \cong c(q_d, \dot{q}_d) + \overbrace{\left.\frac{\partial c}{\partial q}\right|_{\substack{q=q_d\\ \dot{q}=\dot{q}_d}}}^{C_1(q_d,\dot{q}_d)} \Delta q + \underbrace{\left.\frac{\partial c}{\partial \dot{q}}\right|_{\substack{q=q_d\\ \dot{q}=\dot{q}_d}}}_{C_2(q_d,\dot{q}_d)} \Delta\dot{q}$$

Linearization along a trajectory (cont)

- after simplifications …

$$M(q_d)\Delta\ddot{q} + C_2(q_d,\dot{q}_d)\Delta\dot{q} + D(q_d,\dot{q}_d,\ddot{q}_d)\Delta q = \Delta u$$

with

$$D(q_d,\dot{q}_d,\ddot{q}_d) = G(q_d) + C_1(q_d,\dot{q}_d) + \sum_{i=1}^{N} \left.\frac{\partial M_i}{\partial q}\right|_{q=q_d} \ddot{q}_d e_i^T$$

- in state-space format

$$\Delta\dot{x} = \begin{pmatrix} 0 & I \\ -M^{-1}(q_d)D(q_d,\dot{q}_d,\ddot{q}_d) & -M^{-1}(q_d)C_2(q_d,\dot{q}_d) \end{pmatrix}\Delta x + \begin{pmatrix} 0 \\ M^{-1}(q_d) \end{pmatrix}\Delta u = A(t)\,\Delta x + B(t)\,\Delta u$$

a linear, but time-varying system!!

Coordinate transformation

$$q \in \mathbb{R}^N \quad \boxed{M(q)\ddot{q} + c(q,\dot{q}) + g(q) = M(q)\ddot{q} + n(q,\dot{q}) = u_q} \quad (1)$$

if we wish/need to use a new set of generalized coordinates p

$$p \in \mathbb{R}^N \quad \boxed{p = f(q)} \Rightarrow \boxed{q = f^{-1}(p)}$$

$$\boxed{\dot{p} = \frac{\partial f}{\partial q}\dot{q} = J(q)\dot{q}} \Rightarrow \boxed{\dot{q} = J^{-1}(q)\dot{p}} \quad \boxed{u_q = J^T(q)u_p}$$

$$\boxed{\ddot{p} = J(q)\ddot{q} + \dot{J}(q)\dot{q}} \Rightarrow \boxed{\ddot{q} = J^{-1}(q)\left(\ddot{p} - \dot{J}(q)J^{-1}(q)\dot{p}\right)}$$

$\Rightarrow (1)$

$$\boxed{M(q)J^{-1}(q)\ddot{p} - M(q)J^{-1}(q)\dot{J}(q)J^{-1}(q)\dot{p} + n(q,\dot{q}) = J^T(q)u_p}$$

$J^{-T}(q)\cdot$ pre-multiplying the whole equation...

Robot dynamic model
after coordinate transformation

$$\boxed{J^{-T}(q)M(q)J^{-1}(q)\ddot{p} + J^{-T}(q)\left(n(q,\dot{q}) - M(q)J^{-1}(q)\dot{J}(q)J^{-1}(q)\dot{p}\right) = u_p}$$

$\boxed{q \rightarrow p}$

for actual computation, these inner substitutions are not necessary

$\boxed{(q,\dot{q}) \rightarrow (p,\dot{p})}$

non-conservative generalized forces performing work on p

$$\Rightarrow \quad \boxed{M_p(p)\ddot{p} + c_p(p,\dot{p}) + g_p(q) = u_p}$$

$$M_p = J^{-T}MJ^{-1}$$ symmetric, positive definite (out of singularities)

$$g_p = J^{-T}g$$

$$c_p = J^{-T}\left(c - MJ^{-1}\dot{J}\,J^{-1}\dot{p}\right) = J^{-T}c - M_p\dot{J}\,J^{-1}\dot{p}$$ quadratic dependence on $\dot{p}$

$$c_p(p,\dot{p}) = S_p(p,\dot{p})\,\dot{p} \qquad \dot{M}_p - 2S_p$$ skew-symmetric

when p = E-E pose, this is the robot dynamic model in Cartesian coordinates

Q: What if the robot is redundant with respect to the Cartesian task?

Optimal point-to-point robot motion

considering the dynamic model

- given the initial and final robot configurations (at rest) and actuator torque bounds, find
  - the minimum-time T_{min} motion
  - the (global/integral) minimum-energy E_{min} motion

  and the associated command torques needed to execute them
- a complex nonlinear optimization problem solved numerically

video

T_{min}= 1.32 s, E = 306

video

T = 1.60 s, E_{min} = 6.14