INDUSTRIAL ROBOT SIMULATION MODELS FOR CONTROL DESIGN AND ANALISYS PURPOSES

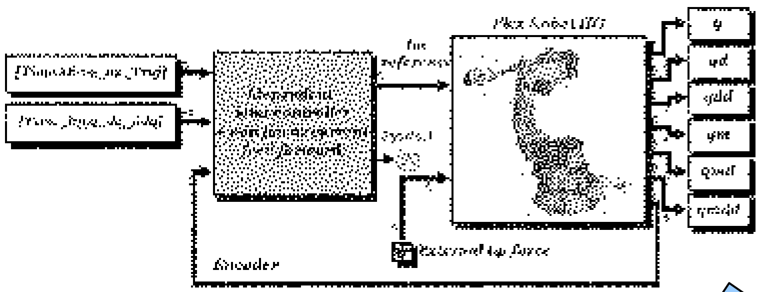
A.Bottero, D.Martinello

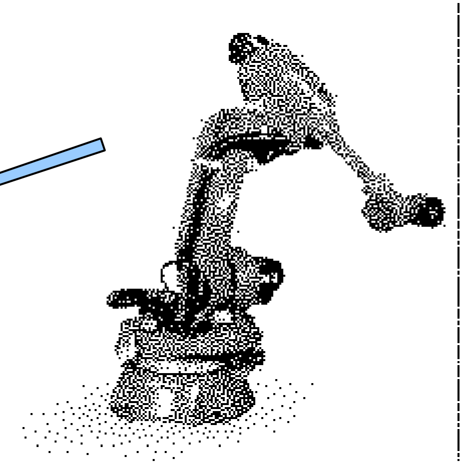
***Control Engineering*, COMAU S.p.A. Robotics, FA & Service**

Simulation: the ACODUASIS CONCEPT

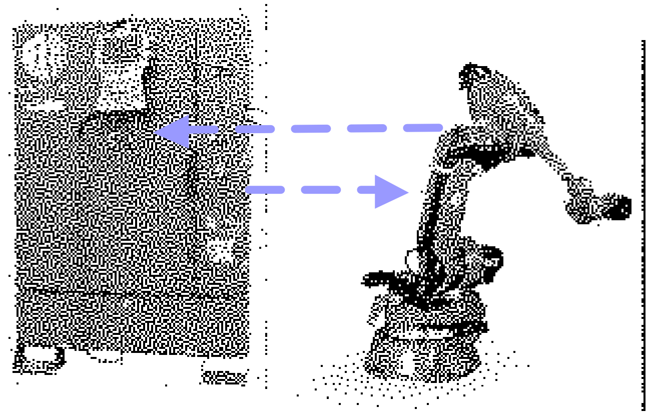
Why industrial robot simulation?

Starting from the robot, we define a "fine" model of the system





with the "fine" model (virtual prototype) we design and test the control system algorithms and parameters, according to the specifications…

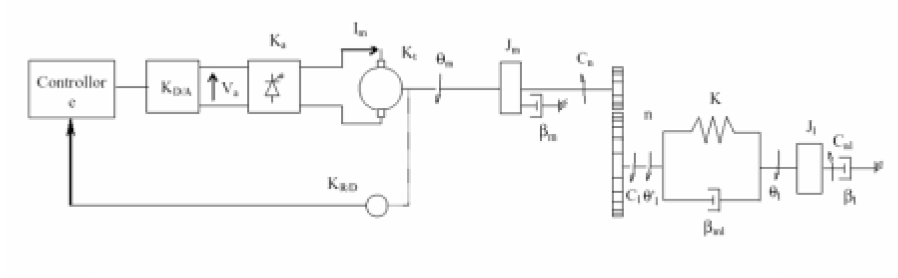


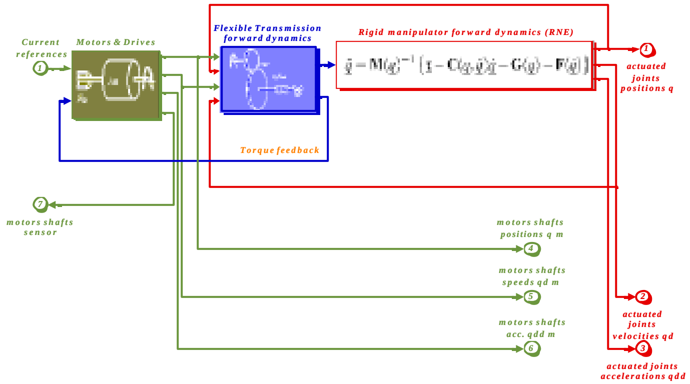
…..and it has to work really on the real robot control system !

Model classes for industrial robot simulation

Three basic classes:

- independent joint axis model:
  - *motor+*
  - *gearbox+*
  - *load inertia*
- Elastic joint (EJ) robot model
  - *rigid multi body chain*
  - *elastic joint transmissions model*
  - *Motors*
- Flexible link robot with accurate transmission model





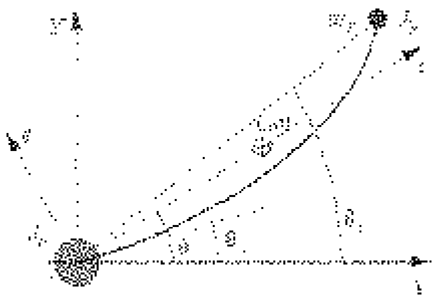
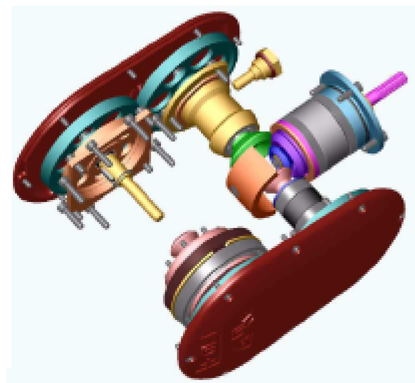
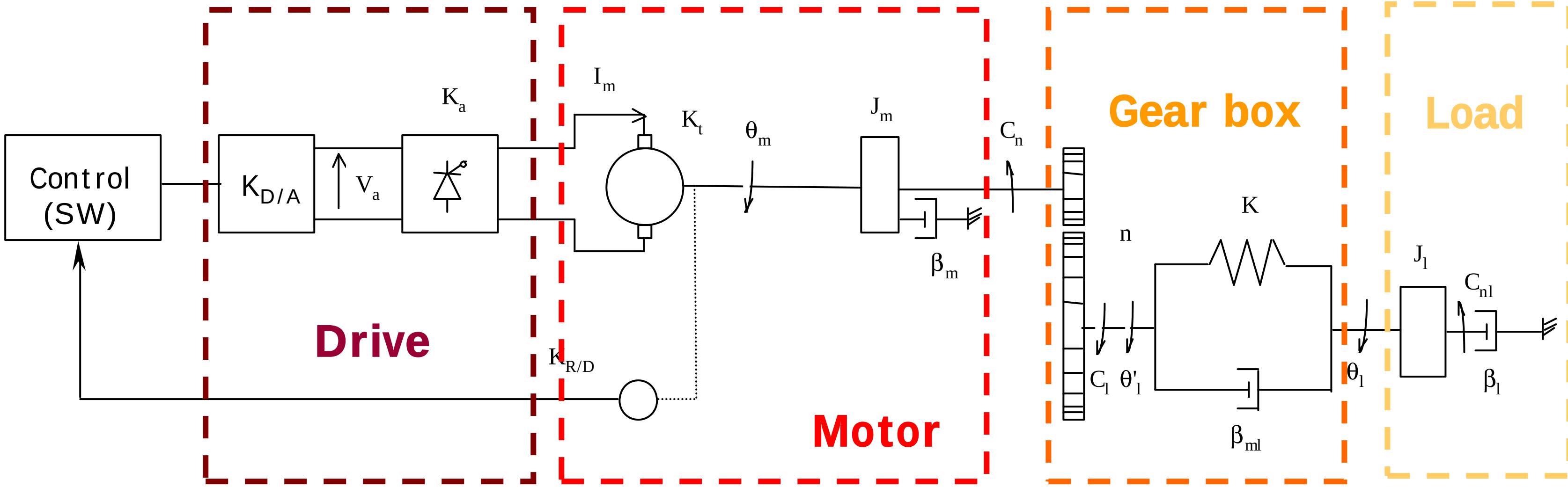


Fig. 1. Euler-Bernoulli beam in rotation



An EJ robot axis



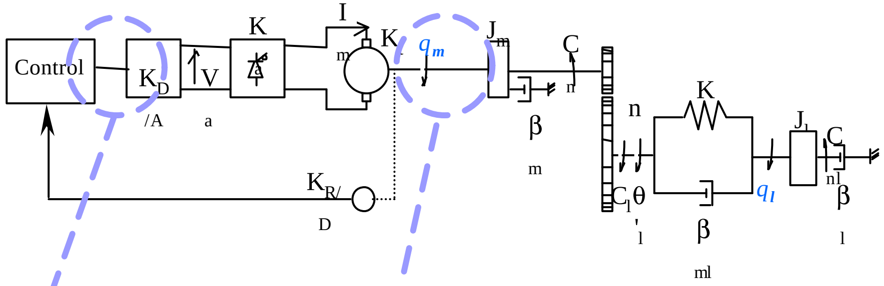
$$\ddot{q}_m = (K_t K_a I_{ref} - \beta_m \dot{q}_m - f(\dot{q}_m) - ntr^{-1} \cdot \tau) / J_m$$

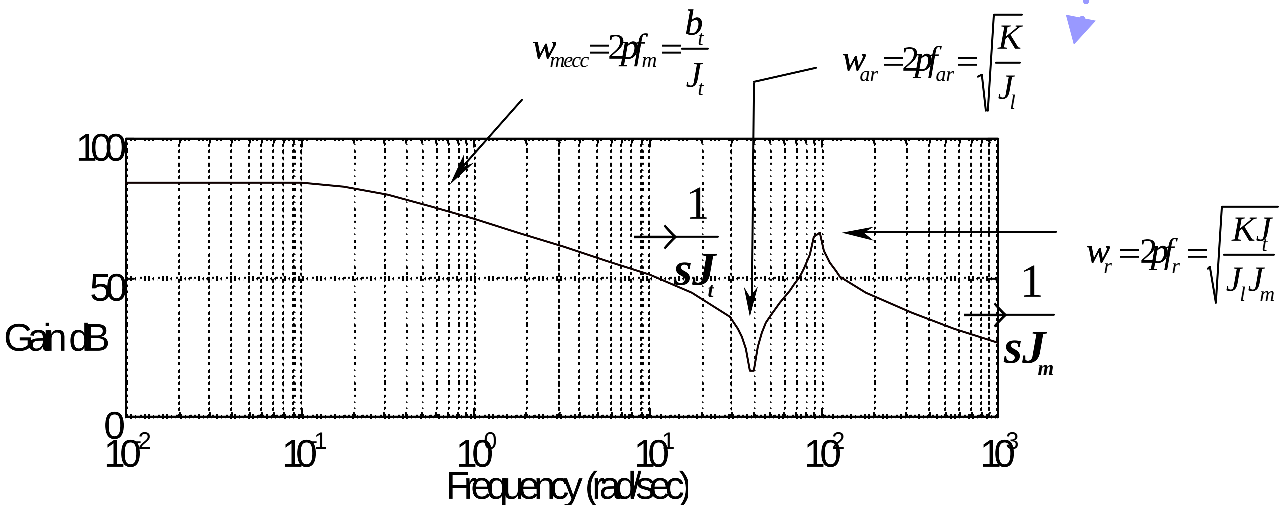
$$\tau = K(ntr^{-1} \cdot q_m - q_l) + \beta_{ml}(ntr^{-1} \cdot \dot{q}_m - \dot{q}_l)$$

$$\ddot{q}_l = (\tau - \beta_l \dot{q}_l + \tau_{ext}) / J_l(q)$$

Motor current to motor speed model

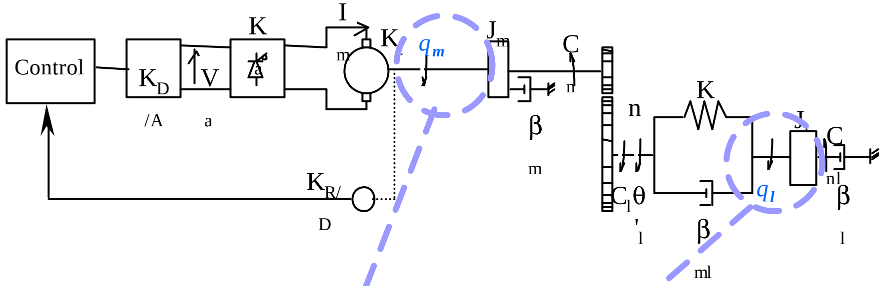
Taking the joint motor current as input variable, the plant shows anti resonance/resonance behaviour on motor velocity



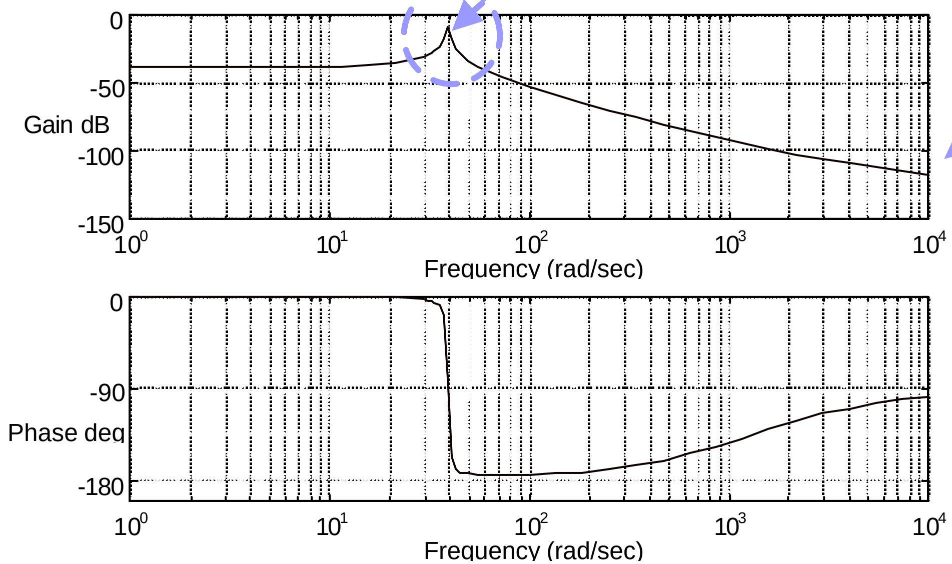


Motor position to joint position model

Taking the motor position
as input variable,
the plant shows resonance
behaviour on joint
position

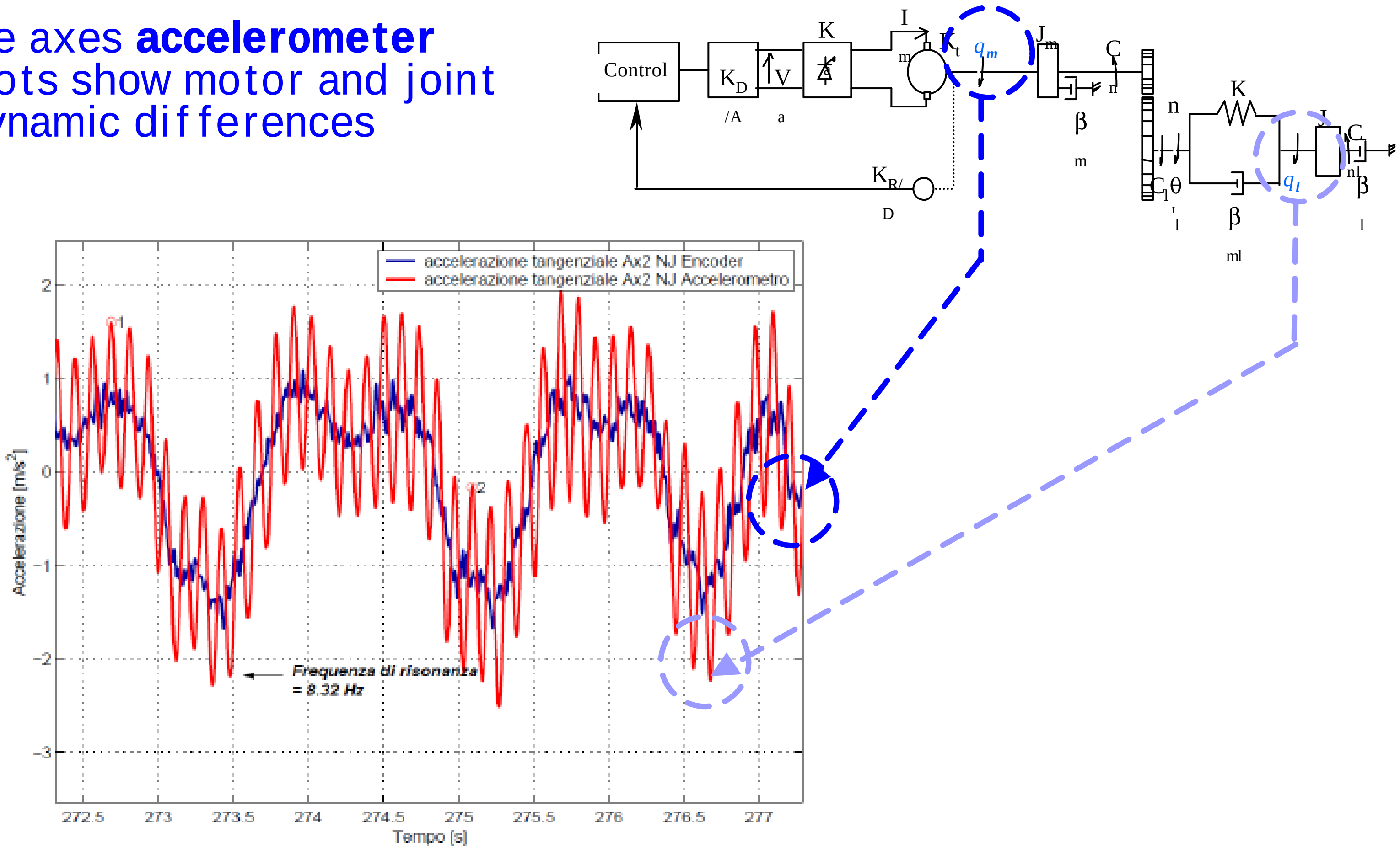


$$w_r = 2pf_r = \sqrt{\frac{K}{J_l}}$$

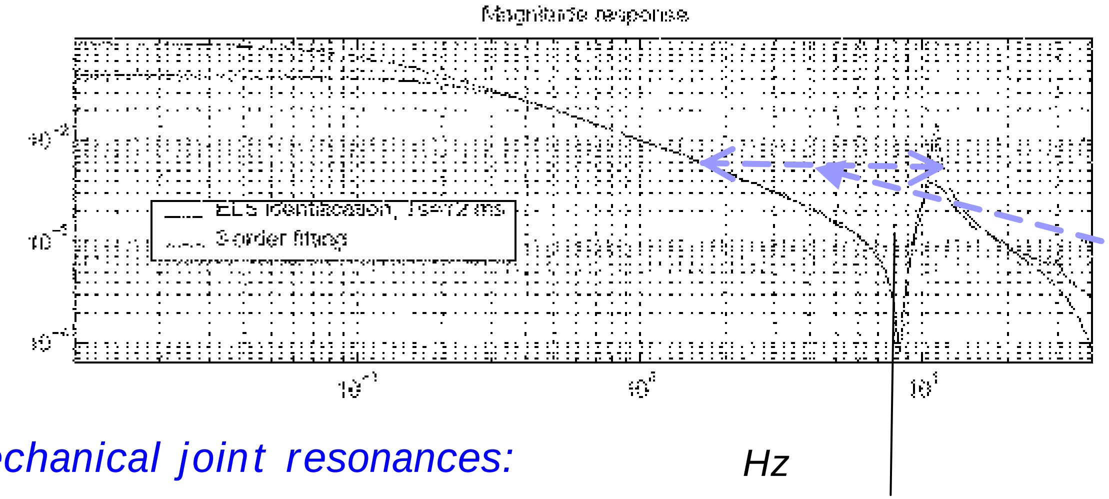


Time domain mesurements of an EJ robot axis

Three axes **accelerometer** plots show motor and joint dynamic differences



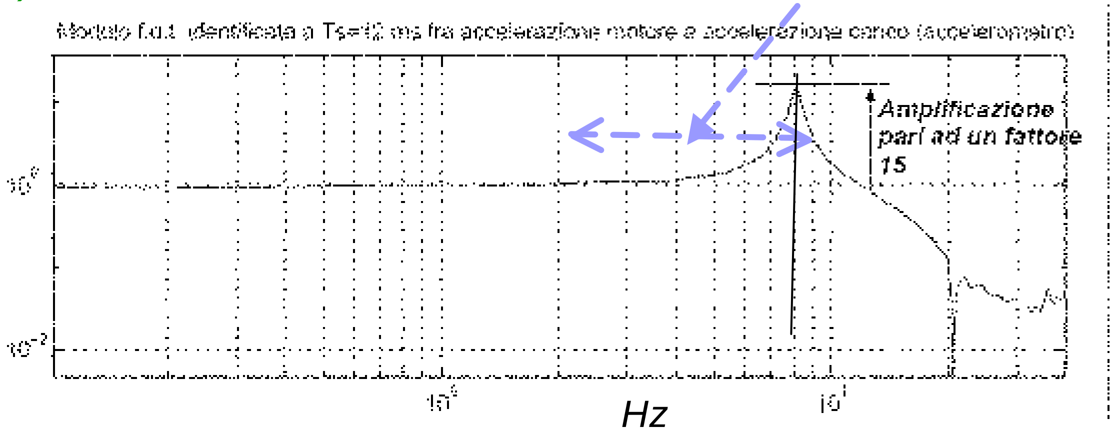
Frequency domain experimental plots



$$w_{antiris} = \sqrt{\frac{K}{J_l}}$$

Mechanical joint resonances:

- *low frequency range (3…20 Hz)*
- *depend on joint stiffness **K***
- *depend on load inertia $\boldsymbol{J_l}$*
- *not fixed but change rapidly with inertia (so with robot position in space)*



Elasticity in real robot

In industrial robots the elasticity effects are introduced by three main causes:

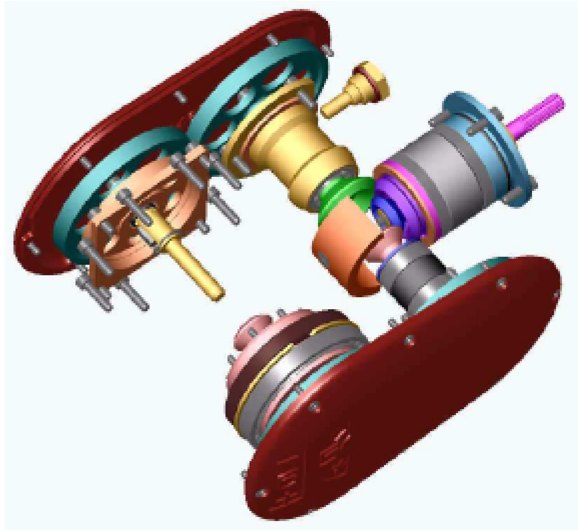
- *the presence of **transmission elements** such as: gearboxes and transmission belts, long shafts (e.g., last 3-dofs of many industrial robots);*

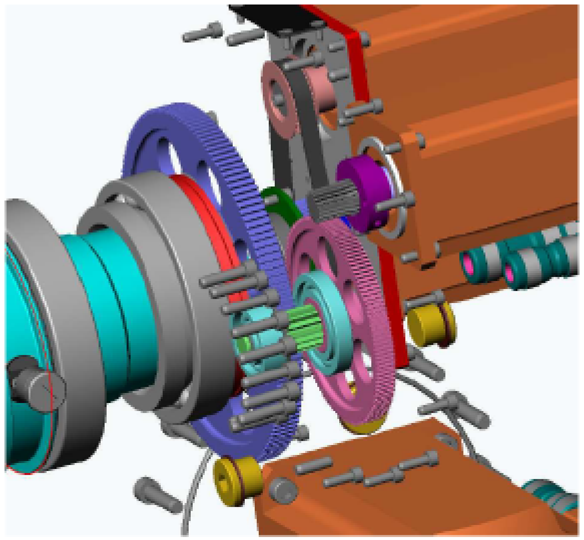
- ***distributed link deformation** : 'link rigidity' is always an ideal assumption and may fail when increasing payload-to-weight ratio, motion speed, control bandwidth;*

- *the presence of **parasitic degrees of freedom**, due to non ideal behaviour of complex mechanical structures, introduce parasitic elastic joints with parasitic resonances effects.*

Elasticity effects in transmissions

Industrial gearboxes and transmissions are complex mechanical systems. The presence of multiple reduction stages can introduce multiple resonances. Since the gear inertias are much smaller than the applied load inertia the induced high frequency resonances may be neglected.





- The gear elasticity usually introduces one dominant low frequency resonance.

- We assume gear elasticity as concentrated in the joint and represented with one spring.

Distributed link deformation

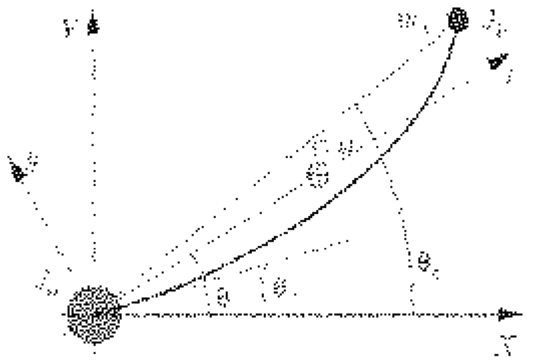


Fig. 1. Euler-Bernoulli beam in rotation

We assume that:

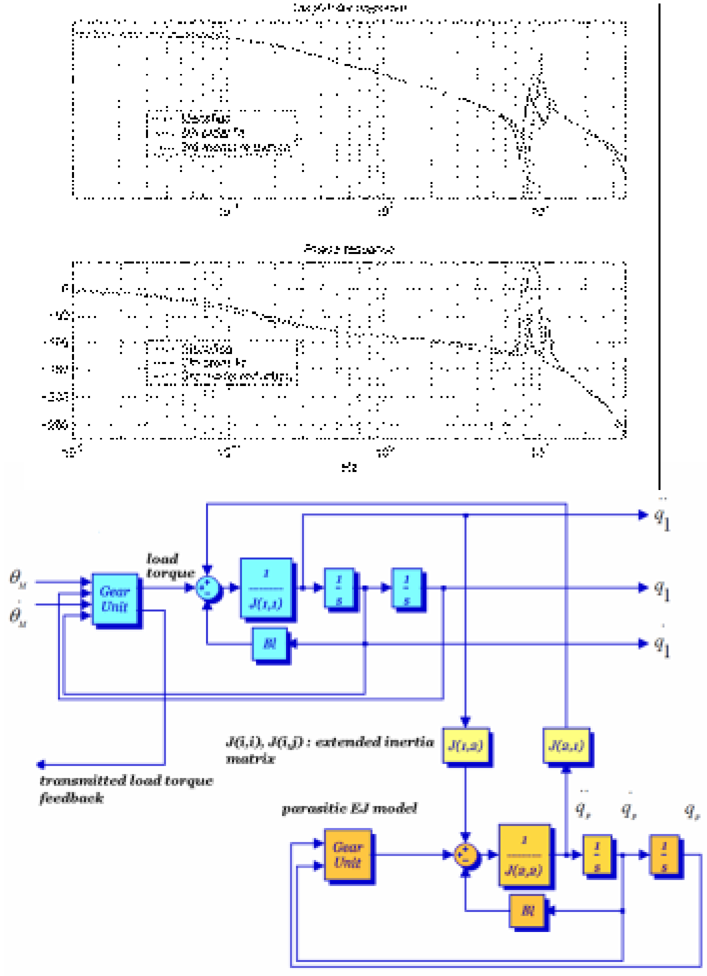
A1) the difference *?t*(*t*)-*?c*(*t*) is “small” and we introduce the rough straight line approximation of the beam bending deflection: the link is still considered as rigid with position *?(t)* of the *CoM*. *?(t)*-*?c*(*t*) is considered tangent to the link base and is added to the transmission elasticity deflection. Link bending and gear deflection will be represented as an elastic joint linking the actuator output shaft to a rigid link.

Parasitic resonances

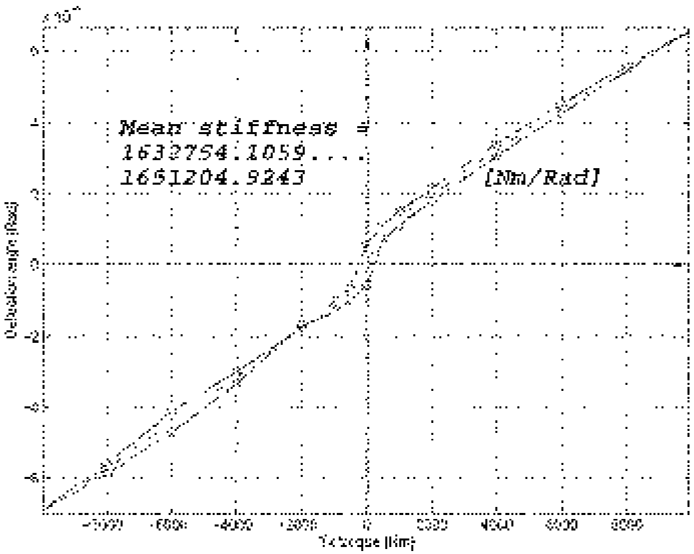
Parasitic joint model represent unexpected elastic deformations concentrated in some part of the robot link chain.

These parasitic EJ are responsible for resonance modes not introduced by actuated EJ.

We introduce the extended higher order dynamic model with both actuated and parasitic EJ.



A typical example: Comau NJ Ax3



Payload flange
(measurement point)

Joint 3 rotation axis (Ax3)

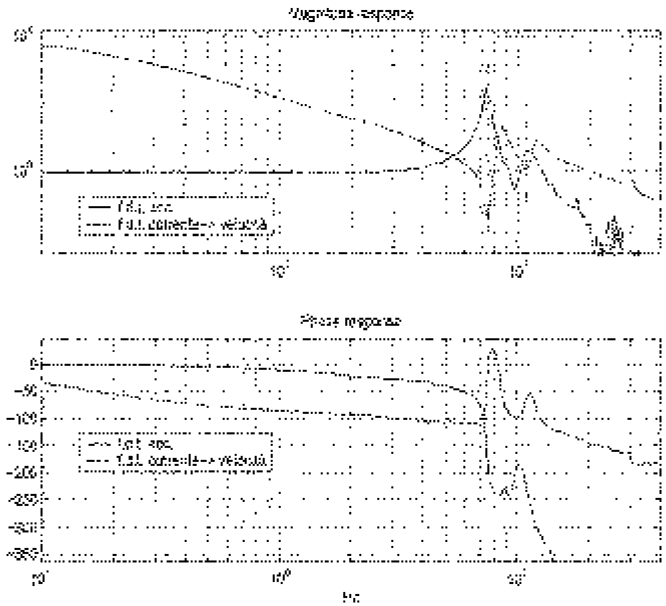
A typical example: Comau NJ Ax3 (2)

Frequency domain experimental plots and stiffness identification:

- ***blue***: *output measurements from an accelerometer on payload flange.*

- ***red***: *output measurements are motor shaft speed (encoder) measurements.*

Two resonances (main and parasitic).



Stiffness estimate result report for Ax3:

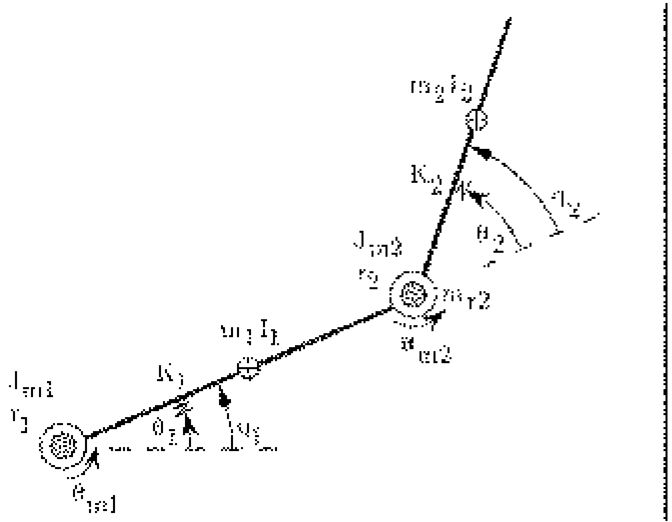
- K_{ax3} = 1651204 Nm/Rad (out of Ax2 effect)
- $K_{gear\ ax3}$ = 8126271 Nm/Rad
- $K_{est\ Ax3}$ = 1723408 Nm/Rad (out of parasitic effects)

EJ Robot dynamic modeling

To introduce the EJ robot dynamic model, we consider:

•an open-chain robot with *N* (rotary or prismatic) elastic joints and *N* rigid links, driven by electrical actuators;

•motor variables $? ?R_N$ (as reflected through reduction ratios) and link variables $q\ ?R_N$ as generalized coordinates; $(q-?)$ represents both the gear and the link bending deflections and *q* defines the positions of link centers of mass;



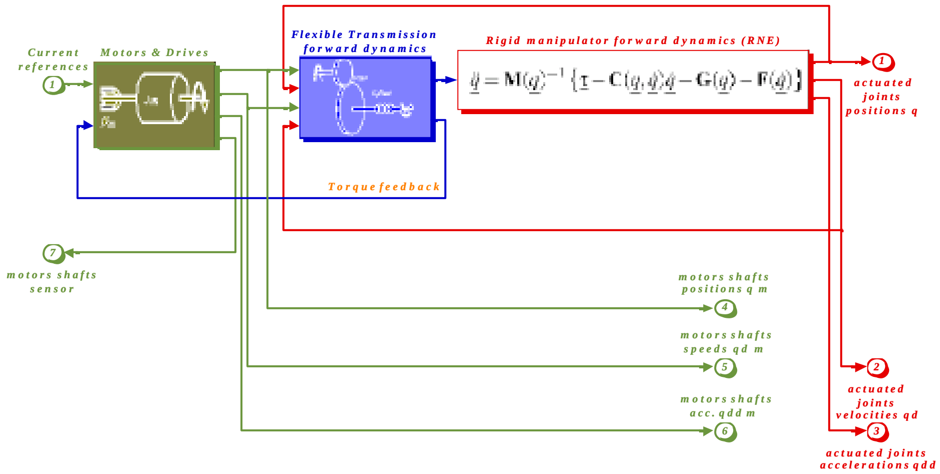
EJ robot dynamic modeling: the ACODUASIS “fine” model

Adding to the model:

- the dissipative terms (friction),
- the elasticity internal damping
- the diagonal matrix *Ntr* for joint reduction ratios,

we obtain three equations:

- *link,*
- *joint transmission and*
- *motor equation.*



This is the formal expression for the **ACODUASIS “fine” robot model definition.**

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + f_l(\dot{q}) + b_l\dot{q} = t_l$$

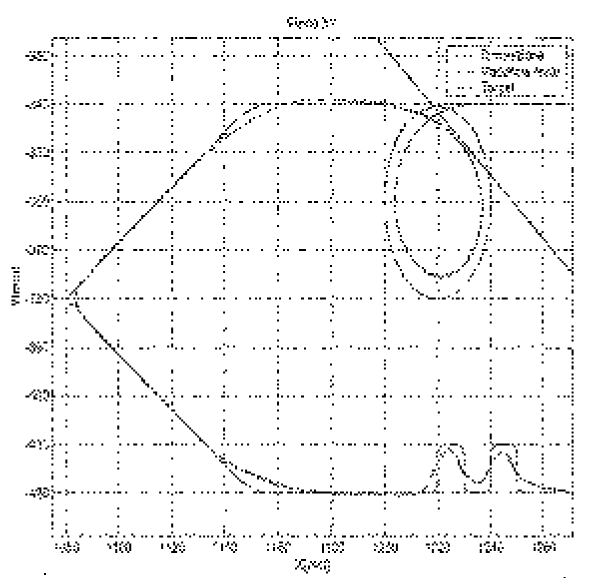
$$t_l = K(q - q) + b_{ml}(\dot{q} - \dot{q}) \qquad q = Ntr^{-1} \cdot q_m$$

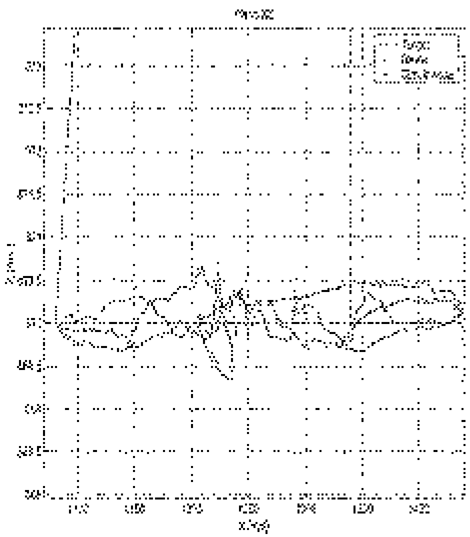
$$J_m\ddot{q}_m + b_m\dot{q}_m + f_m(\dot{q}_m) - Ntr^{-1} \cdot t_l = K_t \cdot I_m$$

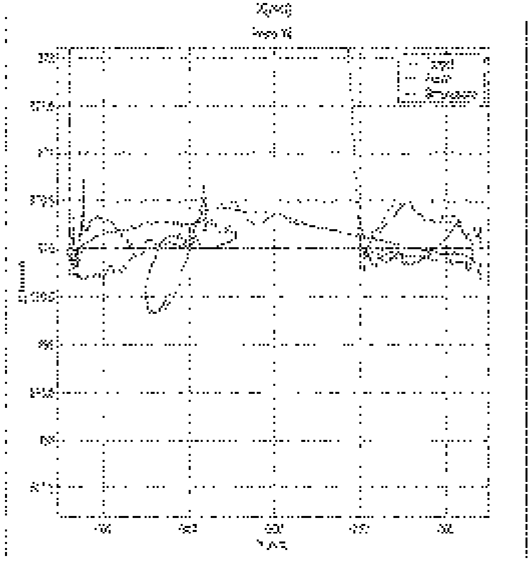
Sample simulation results

The ACODUASIS “fine” robot simulation model, when properly characterized, gives a good simulation of the robot behavior, both in joint and in Cartesian space.

It’s a good model for control design and testing.





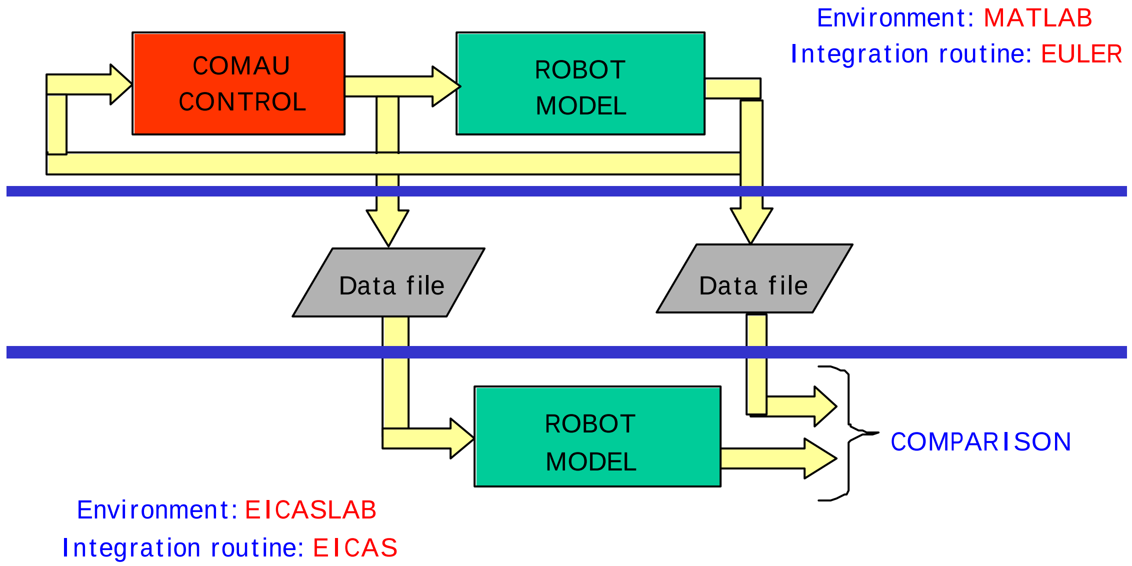


EICASLAB “fine” model implementation and validation (1)

The fine model of the plant has been provided by COMAU Robotics and implemented in the EICASLab software tool.

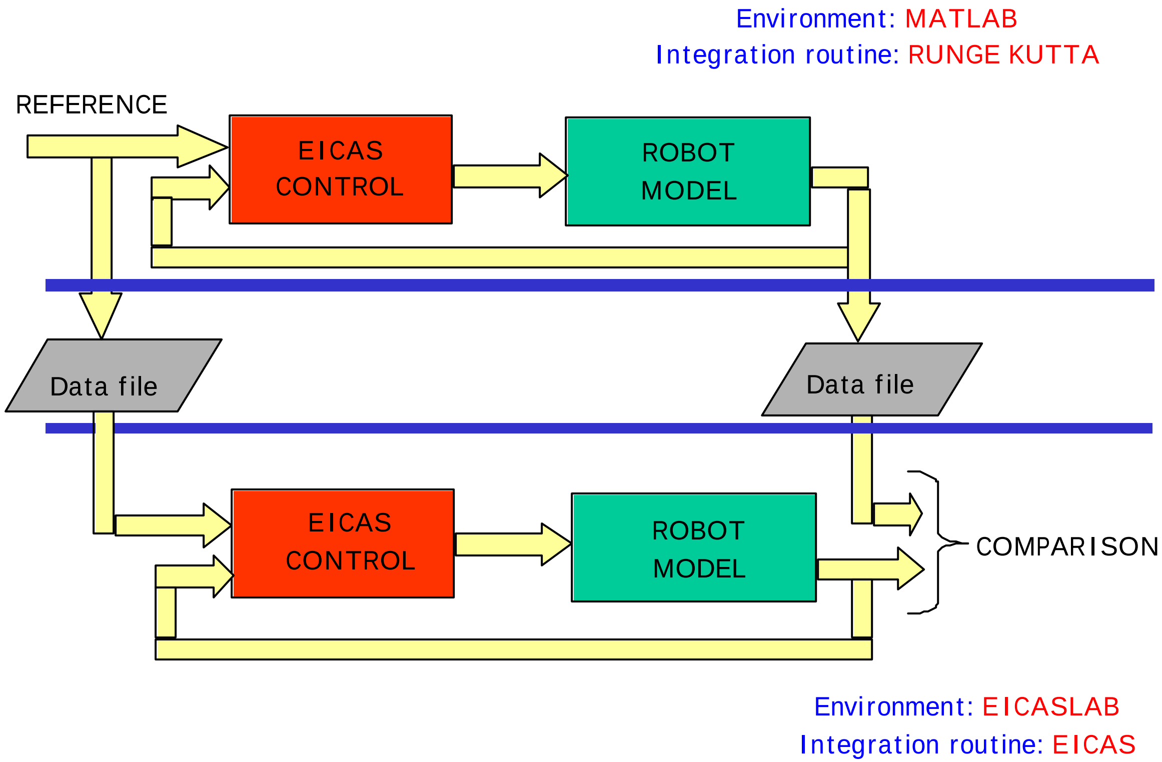
The validation has been performed by comparing the outputs of the EICASLab model with the ones of the COMAU model (Matlab model) when the same inputs were applied to both the models.

This approach pointed out some problems related to the fact that two different numerical integration routines (Euler in Matlab and an EICAS property algorithm in EICASLab) were used. As the model has an intrinsic instability, small numerical errors during the simulations caused drifts. Anyway these tests allowed to validate the static relations in the models.

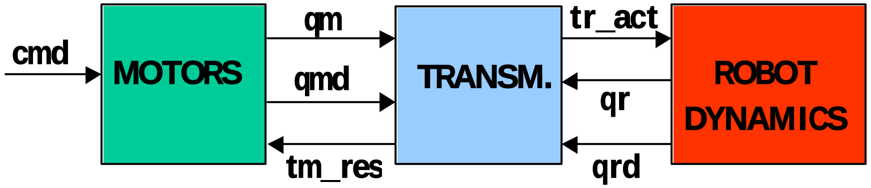


EICASLAB "fine" model implementation and validation (2)

In order to validate the dynamic parts of the EICASLab model, another kind of approach has been used. The principle was to use in each environment (EICASLab and Matlab) the same control algorithm, realised with the EICAS methodology. In this way the stability of the models has been guaranteed and the difference between the two integration routines (Runge Kutta in Matlab and EICAS in EICASLab) can be neglected.



The ACODUASIS "fine" model robotics library



•cinematic and dynamic behaviour of an industrial robot;

•generic chain of links, joined by prismatic or rotational joints, and structured in an open or closed loop chain;

•just a parametrical description (object attributes specifications); no code must be written (object methods already defined);

•it is possible to do simulation and evaluation of robot performance both in the joint coordinates space and in the Cartesian space.