*Robotics 2*

# Hybrid Force/Motion Control

Prof. Alessandro De Luca

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

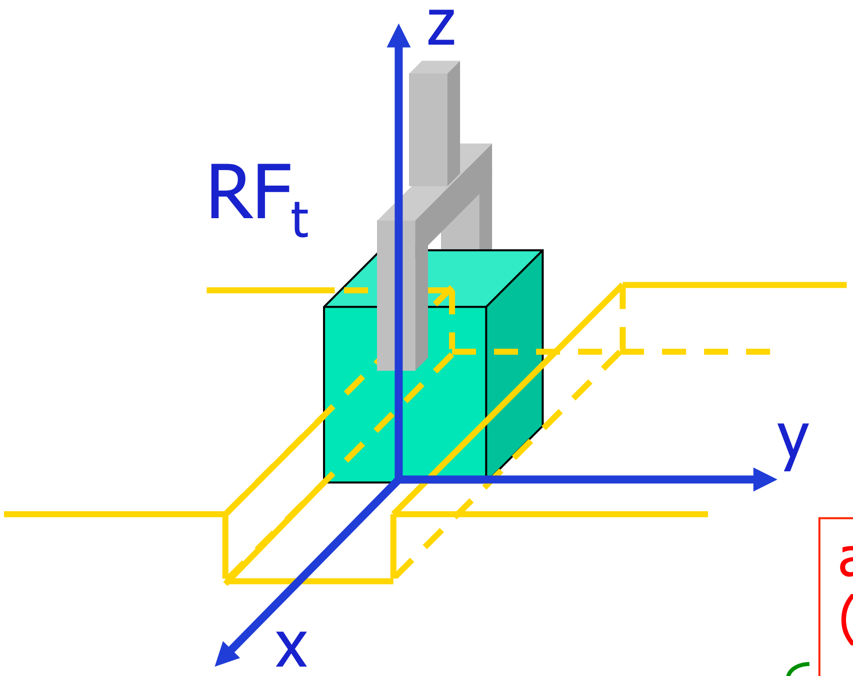
# Natural and artificial constraints

- contact/interaction between the robot and a "purely geometric" (rigid and frictionless) environment naturally constrains the end-effector motion
- in ideal conditions (robot and environment perfectly rigid, frictionless contact), one can define two sets of generalized directions in the task space which are selected so that
  - end-effector motion is feasible in a set of k directions (where the environment cannot react with forces/torques)
  - contact reaction forces/torques arise in a set of 6-k directions (where the environment bars any end-effector motion)

**natural constraints** on force and motion imposed by the task

- these sets of directions are mutually orthogonal (and complementary, namely they cover the 6D task/Cartesian space) and are characterized by a suitable task frame $RF_t$ (typically attached to the robot end-effector)
- for general interaction tasks, position and orientation of the task frame will be time-varying
- the way task execution should be performed can be expressed in terms of **artificial constraints** that specify desired values (to be imposed by the control law) for the velocities, in the k directions feasible for motion, and for the forces, in the remaining 6-k directions feasible for contact reaction

# Task frame and constraints - example 1




task: slide the cube
along the guide

natural (geometric) constraints

$$\left.\begin{array}{l} v_y = v_z = 0 \\ \omega_x = \omega_z = 0 \end{array}\right\} \text{6-k=4}$$

$$\left. F_x = M_y = 0 \right\} \text{k=2}$$

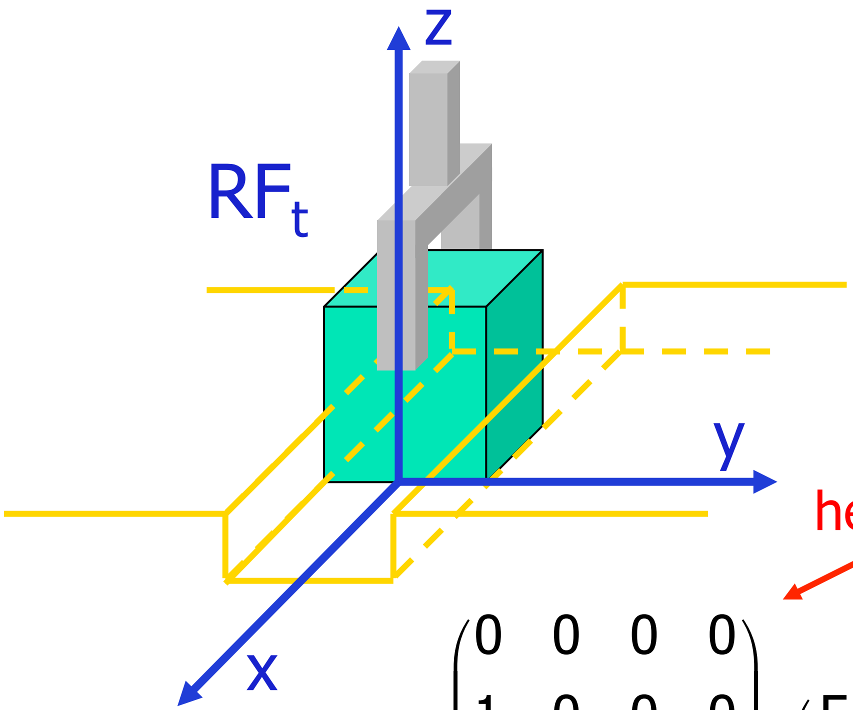
artificial constraints
(to be imposed by the control law)

$$\text{6-k=4} \left\{ \begin{array}{l} F_y = F_{y,des} \ (= 0) \\ M_x = M_{x,des} (= 0), \quad M_z = M_{z,des} \ (= 0) \\ F_z = F_{z,des} \end{array}\right.$$

$$\text{k=2} \left\{ \begin{array}{l} \omega_y = \omega_{y,des} (= 0) \\ v_x = v_{x,des} \end{array}\right.$$

v = linear velocity
$\omega$ = angular velocity
F = force
M = moment

# Selection of directions - example 1




$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} v_x \\ \omega_y \end{pmatrix} = T \cdot \begin{pmatrix} v_x \\ \omega_y \end{pmatrix}$$

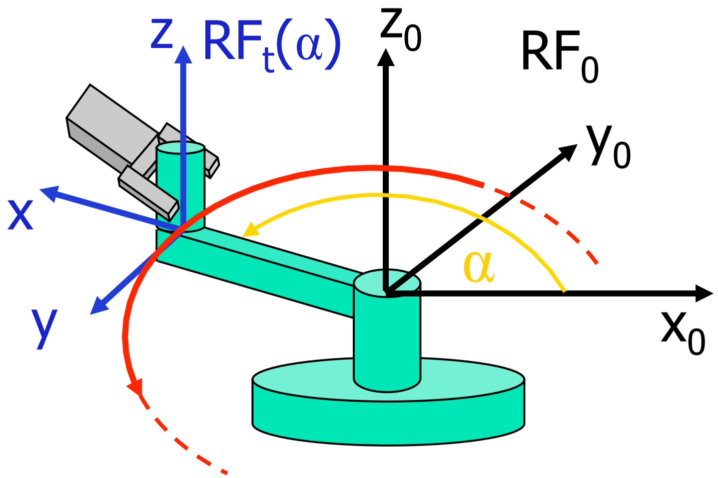
here, constant

$$\begin{pmatrix} F \\ M \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} F_y \\ F_z \\ M_x \\ M_z \end{pmatrix} = Y \cdot \begin{pmatrix} F_y \\ F_z \\ M_x \\ M_z \end{pmatrix}$$

$$T^T \cdot Y = 0$$

constraint forces/torques do not perform work on feasible motion

# Task frame and constraints - example 2




$$RF_0 \xrightarrow{R_z(\alpha)} RF_t(\alpha)$$

task: turning a crank
(free handle)

**natural constraints**

$v_x = v_z = 0$

$\omega_x = \omega_y = 0$

$F_y = M_z = 0$

**artificial constraints**

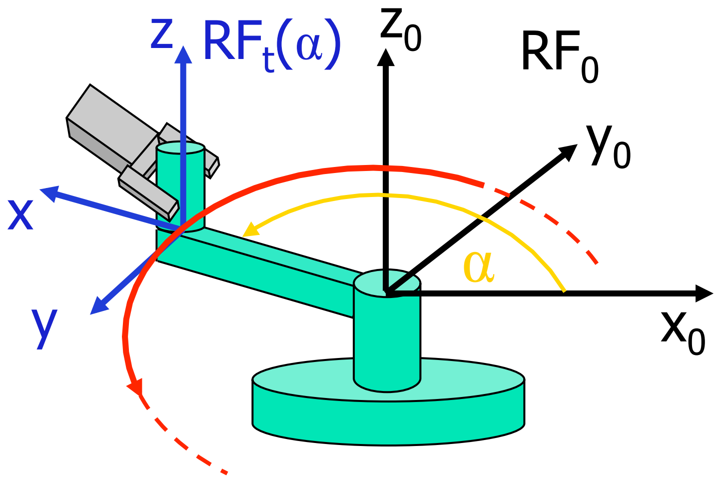
$F_x = F_{x,des}\ (= 0),\quad F_z = F_{z,des}\ (= 0)$

$M_x = M_{x,des}\ (= 0),\quad M_y = M_{y,des}\ (= 0)$

$v_y = v_{y,des}$

$\omega_z = \omega_{z,des}\ (= 0)$

# Selection of directions – example 2

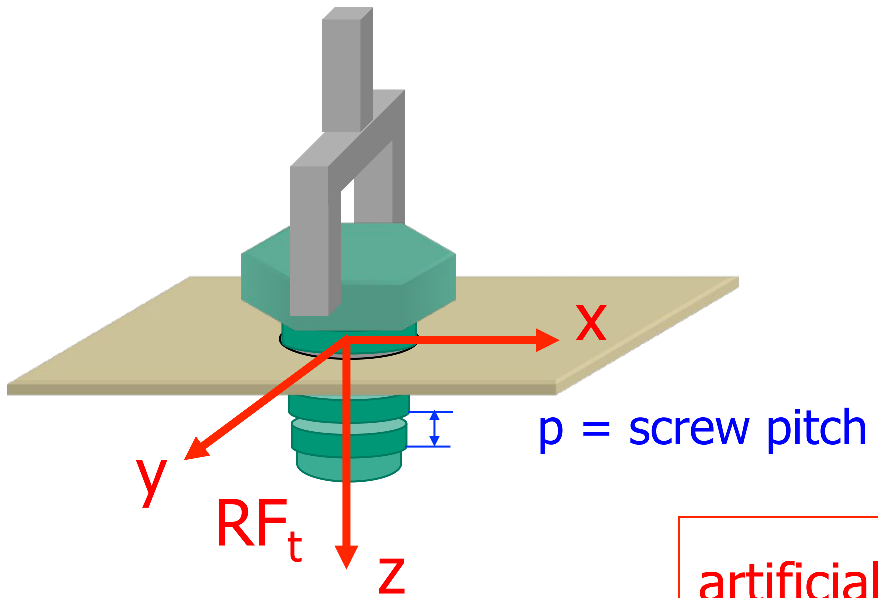



$$\begin{pmatrix} {}^0v \\ {}^0\omega \end{pmatrix} = \begin{pmatrix} R^T(\alpha) & 0 \\ 0 & R^T(\alpha) \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_y \\ \omega_z \end{pmatrix} = T(\alpha) \cdot \begin{pmatrix} v_y \\ \omega_z \end{pmatrix}$$

$$\begin{pmatrix} {}^0F \\ {}^0M \end{pmatrix} = \begin{pmatrix} R^T(\alpha) & 0 \\ 0 & R^T(\alpha) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_x \\ F_z \\ M_x \\ M_y \end{pmatrix} = Y(\alpha) \cdot \begin{pmatrix} F_x \\ F_z \\ M_x \\ M_y \end{pmatrix}$$

$$\boxed{T^T(\alpha) \cdot Y(\alpha) = 0}$$

# Task frame and constraints - example 3




task: insert a screw in a bolt

natural constraints (partial…)

$v_x = v_y = 0$

$\omega_x = \omega_y = 0$

artificial constraints (abundant…)

$F_x = F_{x,des} = 0, \quad F_y = F_{y,des} = 0$

$M_x = M_{x,des} = 0, \quad M_y = M_{y,des} = 0$

$v_z = v_{z,des}\,,\ \omega_z = \omega_{z,des} = (2\pi/p) \cdot v_{z,des}$

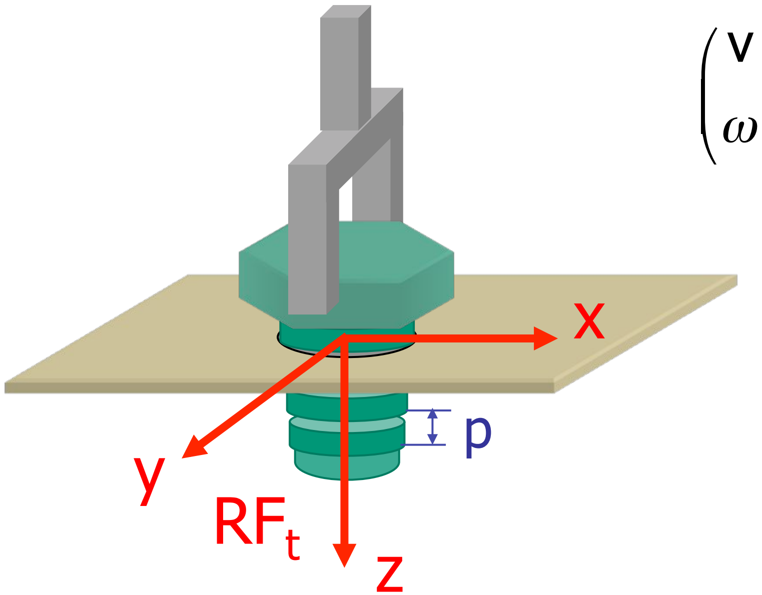
$F_z = F_{z,des}\,,\ M_z = M_{z,des}(F_{z,des})$

the screw proceeds along and around the z-axis, but **not** in an **independent** way! (1 dof)

accordingly, $F_z$ and $M_z$ **cannot** be **independent**

the force/torque direction should be orthogonal to that of motion!

# Selection of directions – example 3




$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & \frac{2\pi}{p} \end{pmatrix}^T \cdot v_z = T \cdot v_z \qquad k=1$$

Y: such that $T^T \cdot Y = 0$

$$F_z = -\frac{2\pi}{p} M_z$$

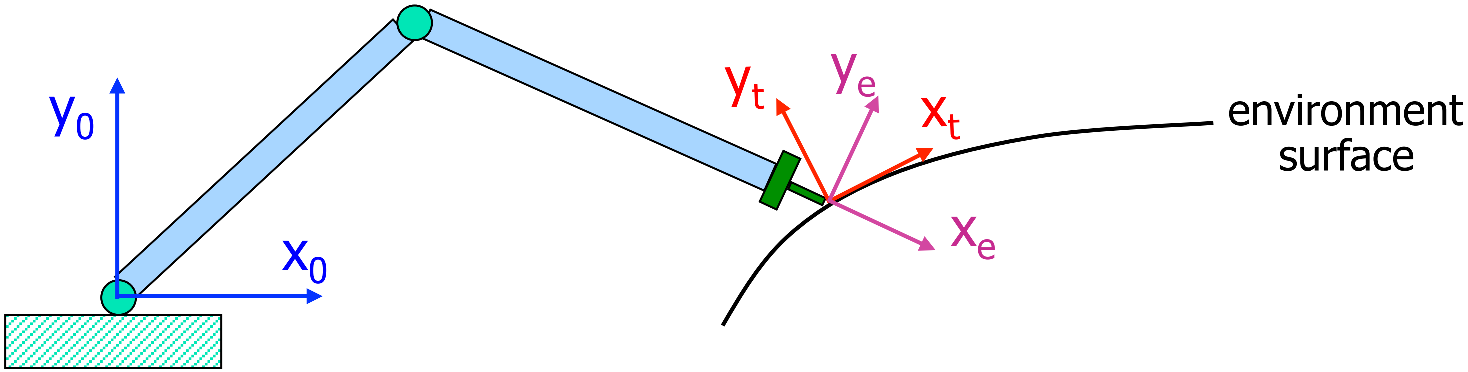
6-k=5

$$\begin{pmatrix} F \\ M \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{2\pi}{p} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} F_x \\ F_y \\ M_x \\ M_y \\ M_z \end{pmatrix} = Y \cdot \begin{pmatrix} F_x \\ F_y \\ M_x \\ M_y \\ M_z \end{pmatrix}$$

the columns of T and Y
do not necessarily coincide
with Cartesian directions
(columns of the identity matrix)
⇨ generalized directions

# Frames of interest – example 4

planar motion of a 2R robot in contact with a surface (M=2)




- task frame $RF_t$ used for an independent definition of the hybrid reference values (here: ${}^t v_{x,des}$ [k=1] and ${}^t F_{y,des}$ [M-k=1]) and for computing the errors driving the feedback control law
- sensor frame $RF_e$ (here = $RF_2$) where the force ${}^e F = ({}^e F_x, {}^e F_y)$ is measured
- base frame $RF_0$ in which the end-effector velocity is expressed (here, ${}^0 v = ({}^0 v_x, {}^0 v_y)$ of $O_2$), computed using robot Jacobian and joint velocities

all quantities (and errors!) should be expressed (rotated) in the same reference frame: the task frame!

# Parameterization of hybrid tasks

a "description" of robot-environment contact type: it defines the task frame

$$\begin{cases} \begin{pmatrix} v \\ \omega \end{pmatrix} = T(s) \cdot \dot{s} & s \in \mathbb{R}^k \text{ parameterizes E-E free motion} \\ \begin{pmatrix} F \\ M \end{pmatrix} = Y(s) \cdot \lambda & \lambda \in \mathbb{R}^{M-k} \text{ parameterizes contact forces/torques} \end{cases}$$

contact forces/torques do not perform work on E-E displacements

$$T^T(s) \cdot Y(s) = 0$$

the generalized directions of the task frame depend in general on s (i.e., on the E-E pose in the environment)

in the previous first three examples, and in general, it is M=6

+

robot dynamics

$$B(q)\ddot{q} + S(q,\dot{q})\dot{q} + g(q) = u + J^T(q)\begin{pmatrix} F \\ M \end{pmatrix}$$

robot kinematics

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = J(q)\dot{q}$$

# Hybrid force/velocity control

- control objective: to impose the desired evolution to the parameters s of motion and to parameters $\lambda$ of force

$$s(t) \rightarrow s_d(t), \quad \lambda(t) \rightarrow \lambda_d(t)$$

- control law is designed in two steps

1. exact linearization and decoupling in the task frame by feedback

$$\boxed{\begin{array}{c}\text{closed-loop}\\ \text{model}\end{array}} \longrightarrow \boxed{\begin{pmatrix} \ddot{s} \\ \lambda \end{pmatrix} = \begin{pmatrix} a_s \\ a_\lambda \end{pmatrix}}$$

2. (linear) design of $a_s$ and $a_\lambda$ so as to impose the desired dynamic behavior to the errors $e_s = s_d - s$ and $e_\lambda = \lambda_d - \lambda$

- assumptions: $N = M$ (= 6, usually), $J(q)$ out of singularity
  (+ $T^T \cdot Y = 0$)

Note: in "simple" cases, $\lambda$ and $\dot{s}$ are just single components of F or M and of v or $\omega$; accordingly, Y and T will be simple 0/1 selection matrices

# Feedback linearization in task space

$$J(q)\cdot\dot{q} = T(s)\cdot\dot{s} \;\Rightarrow\; J\cdot\ddot{q} + \dot{J}\cdot\dot{q} = T\cdot\ddot{s} + \dot{T}\cdot\dot{s} \;\Rightarrow\; \ddot{q} = J^{-1}\left(T\cdot\ddot{s} + \dot{T}\cdot\dot{s} - \dot{J}\cdot\dot{q}\right)$$

$$B(q)\ddot{q} + S(q,\dot{q})\dot{q} + g(q) = u + J^T(q)\begin{pmatrix} F \\ M \end{pmatrix} = u + J^T(q)Y(s)\cdot\lambda$$

$$\boxed{\left(B(q)J^{-1}(q)T(s) \;\vdots\; -J^T(q)Y(s)\right)\begin{pmatrix} \ddot{s} \\ \lambda \end{pmatrix} + B(q)J^{-1}(q)\left(\dot{T}(s)\dot{s} - \dot{J}(q)\dot{q}\right) + S(q,\dot{q})\dot{q} + g(q) = u}$$

nonsingular
N x N matrix
under the
assumptions
made

$$u = \left(BJ^{-1}T \;\vdots\; -J^TY\right)\begin{pmatrix} a_s \\ a_\lambda \end{pmatrix} + BJ^{-1}\left(\dot{T}\dot{s} - \dot{J}\dot{q}\right) + S\cdot\dot{q} + g$$

linearizing and
decoupling
control law

$$\Rightarrow\; \boxed{\begin{pmatrix} \ddot{s} \\ \lambda \end{pmatrix} = \begin{pmatrix} a_s \\ a_\lambda \end{pmatrix}} \begin{matrix} \}\; k \\ \}\; M-k \end{matrix}$$

s has "relative degree" 2

λ has "relative degree" 0

# Stabilization with $a_s$ and $a_\lambda$

as usual, it is sufficient to apply linear control techniques (on each single input-output scalar channel)

$$\boxed{a_s = \ddot{s}_d + K_D(\dot{s}_d - \dot{s}) + K_P(s_d - s)}$$

$$\Rightarrow \quad \boxed{\ddot{e}_s + K_D\dot{e}_s + K_P e_s = 0} \qquad e_s = s_d - s \to 0$$

here $a_\lambda = \lambda_d$ could be enough, but then there would be no "force error" (thus, poor robustness)

$$\boxed{a_\lambda = \lambda_d + K_I \int (\lambda_d - \lambda) d\tau}$$

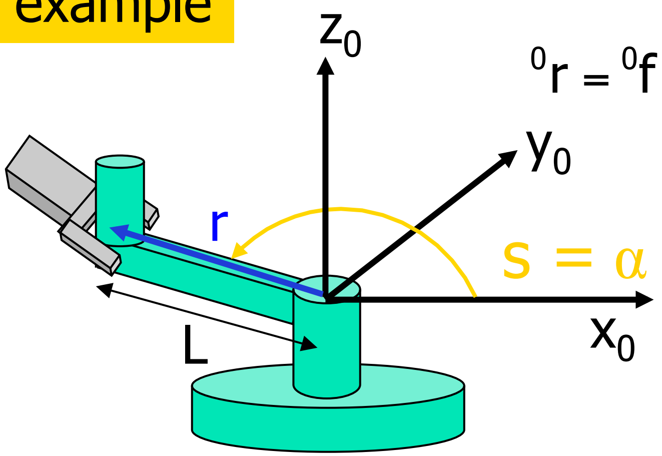
$$\Rightarrow \quad \boxed{\dot{\varepsilon}_\lambda + K_I \varepsilon_\lambda = 0} \qquad \varepsilon_\lambda = \int (\lambda_d - \lambda) d\tau \to 0$$

we need "measures" of s, $\dot{s}$ and $\lambda$!

# "Filtering" position and force measures

- $s$, $\dot{s}$ obtained from measures of $q$ and $\dot{q}$, equating the descriptions of the end-effector pose and velocity "from the robot side" (direct and differential kinematics) and "from the environment side" (function of s)

**example**




$${}^0r = {}^0f(q) = \begin{pmatrix} L\cos s \\ L\sin s \\ 0 \end{pmatrix} \Rightarrow s = \mathrm{ATAN2}\left({}^0f_y(q), {}^0f_x(q)\right)$$

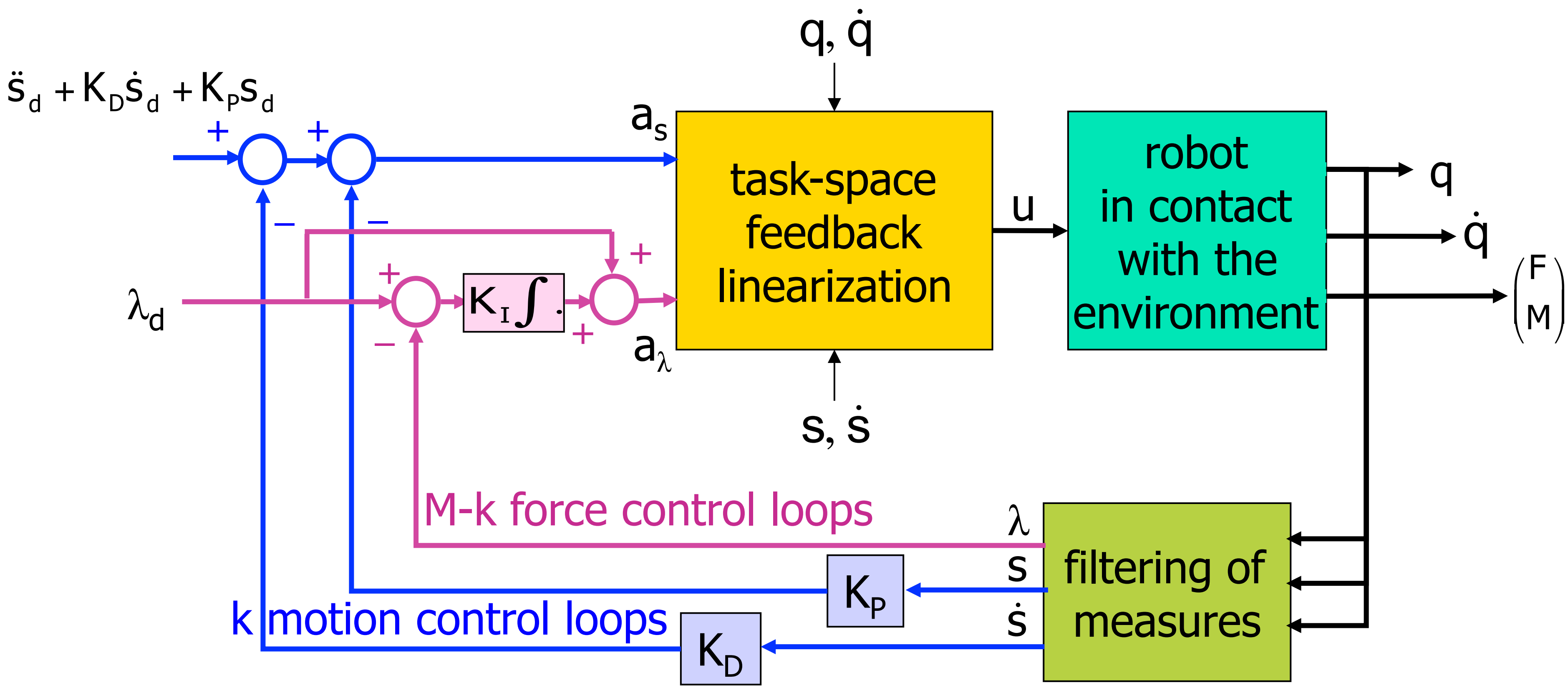
$$J(q)\cdot\dot{q} = T(s)\cdot\dot{s} \Rightarrow \dot{s} = T^{\#}(s)J(q)\cdot\dot{q}$$

- $\lambda$ obtained from force/torque measures at the end-effector

$$\begin{pmatrix} F \\ M \end{pmatrix} = Y(s)\cdot\lambda \Rightarrow \lambda = Y^{\#}(s)\begin{pmatrix} F \\ M \end{pmatrix}$$

pseudoinverses of "tall" matrices with full column rank, e.g., $(T^TT)^{-1}T^T$ (or weighted)

# Block diagram of hybrid control



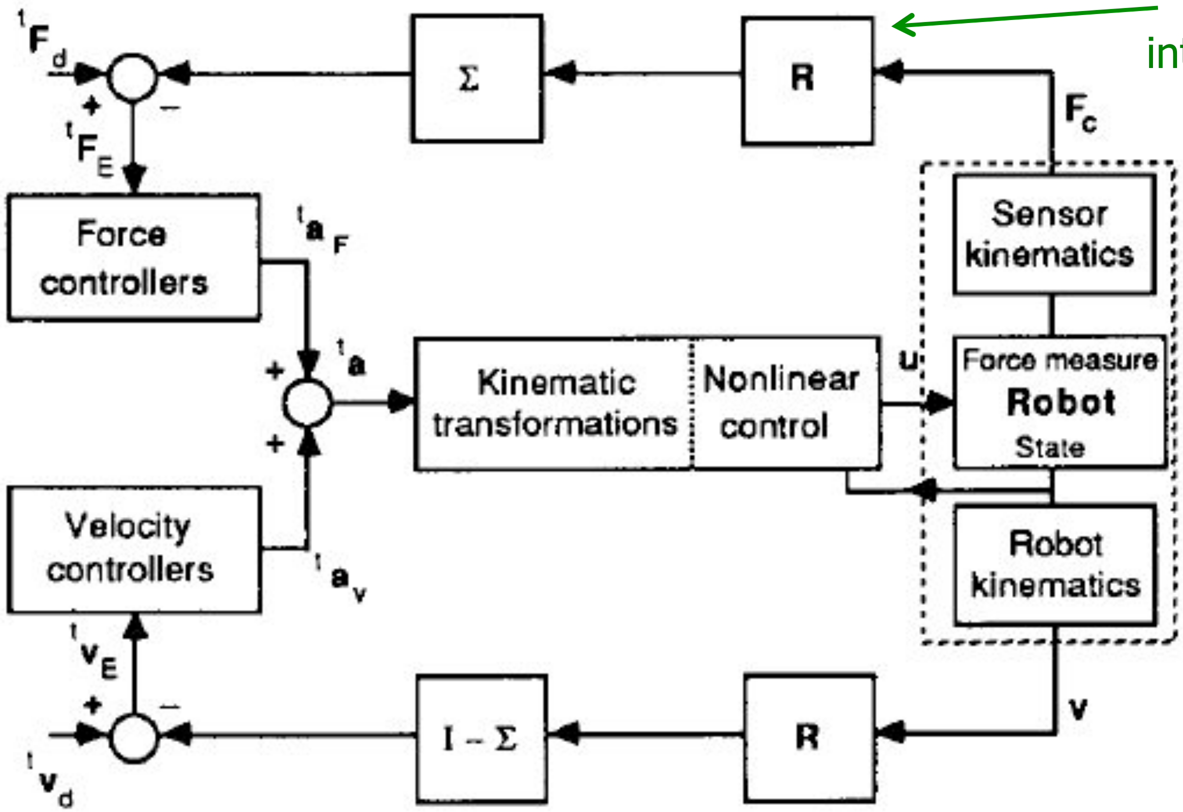

limit cases k=M (free motion): no force control loops, only motion

k=0 (“frozen” robot end-effector): no motion control loops, only force

# Block diagram of hybrid control

## simpler case of 0/1 selection matrices

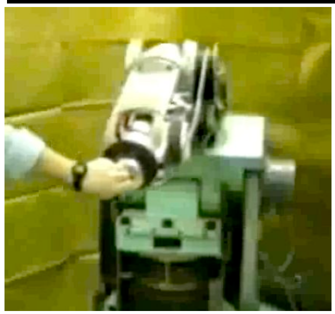
rotation matrix
into the task space




$\lambda$ and $\dot{s}$ are just single components of F (or M) and of v (or $\omega$)

Y and T are replaced by 0/1 selection matrices: $\Sigma$ and I- $\Sigma$

# First experiments with hybrid control



video

video

MIMO-CRF robot
(DIS, Laboratorio di Robotica, 1991)

# Sources of inconsistency
## in force and velocity measurements

1. presence of friction at the contact

   ➔ there is a reaction force component in the "free" motion directions that opposes motion (in case of Coulomb friction, the tangent force intensity depends also from the applied normal force...)

2. compliance in the robot structure and/or at the contact

   ➔ a (small) displacement may result also directions that are nominally "constrained" by the environment

   NOTE: however, if the geometry of the environment at the contact is known with precision, task inconsistencies due to 1. and 2. on the "measures" of s and $\lambda$ are automatically filtered out through the pseudo-inversion of the matrices T and Y

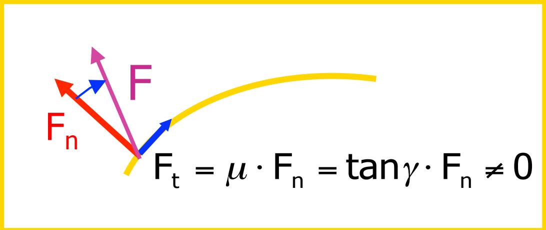
3. uncertainty on the environment geometry at the contact (can be reduced/eliminated by real time estimation processes driven by external sensors: vision, but also force!)

# Identification of an unknown surface

how difficult is to identify the unknown profile of the environment surface, using information from velocity and force measurements at the contact?

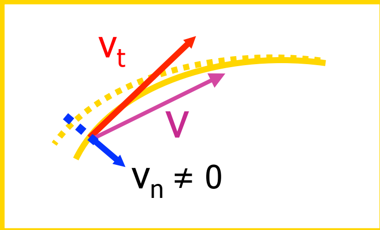
1. normal = nominal direction of measured force

   ... in the presence of contact motion with friction, the measured force F is slightly rotated from the actual normal by an (unknown) angle $\gamma$




2. tangent = nominal direction of measured velocity

   ... compliance in the robot structure (joints) and/or at the contact may lead to a computed velocity v having a small component along the actual normal to the surface

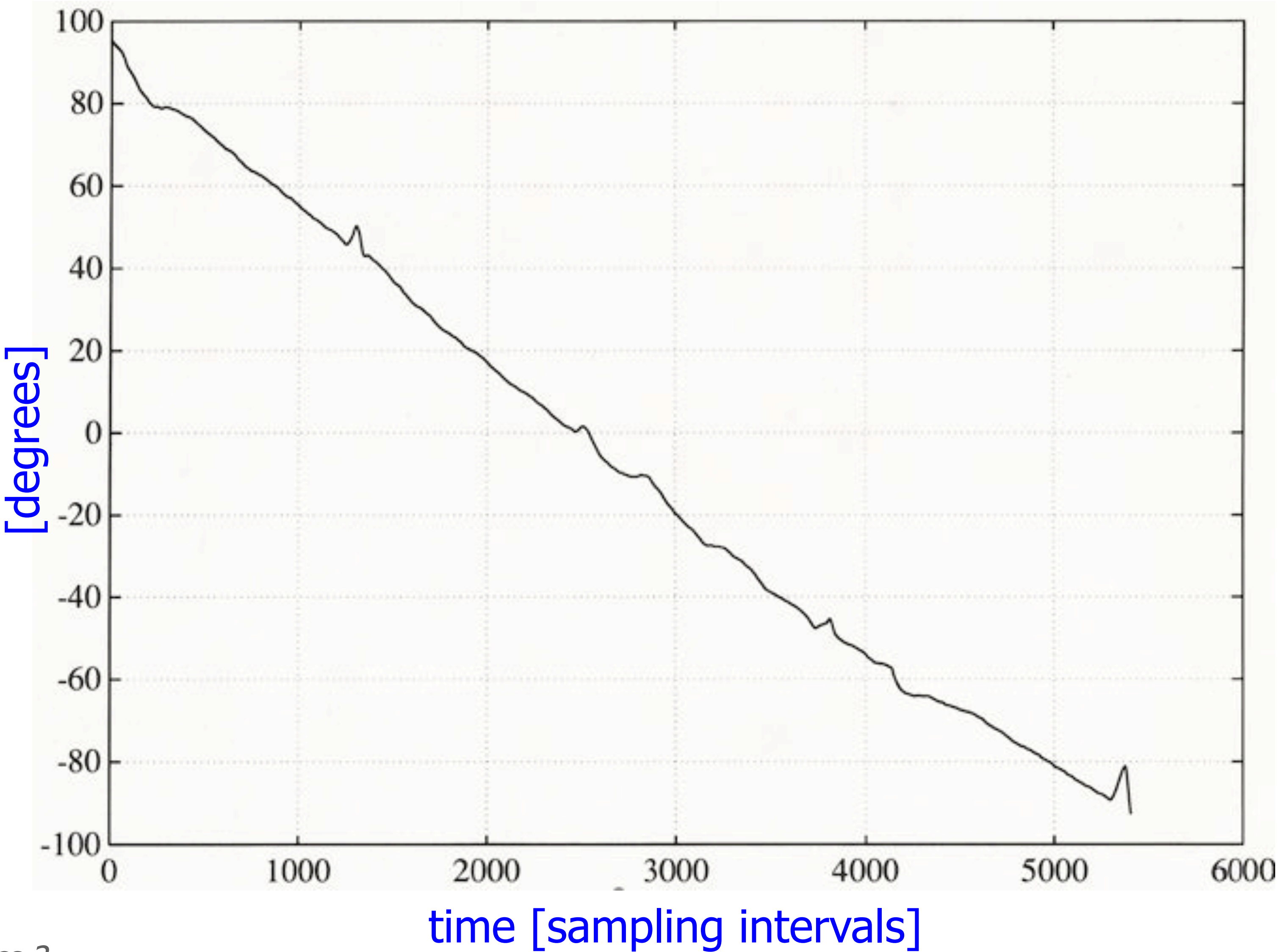



3. mixed method (sensor fusion) with RLS
   a. tangent direction is estimated in a recursive way from position measurements
   b. friction angle is estimated in a recursive way using the current estimate of the tangent and from force measurements

for approaching the unknown surface and for recovering contact (in case of loss), the robot uses a simple exploratory logic

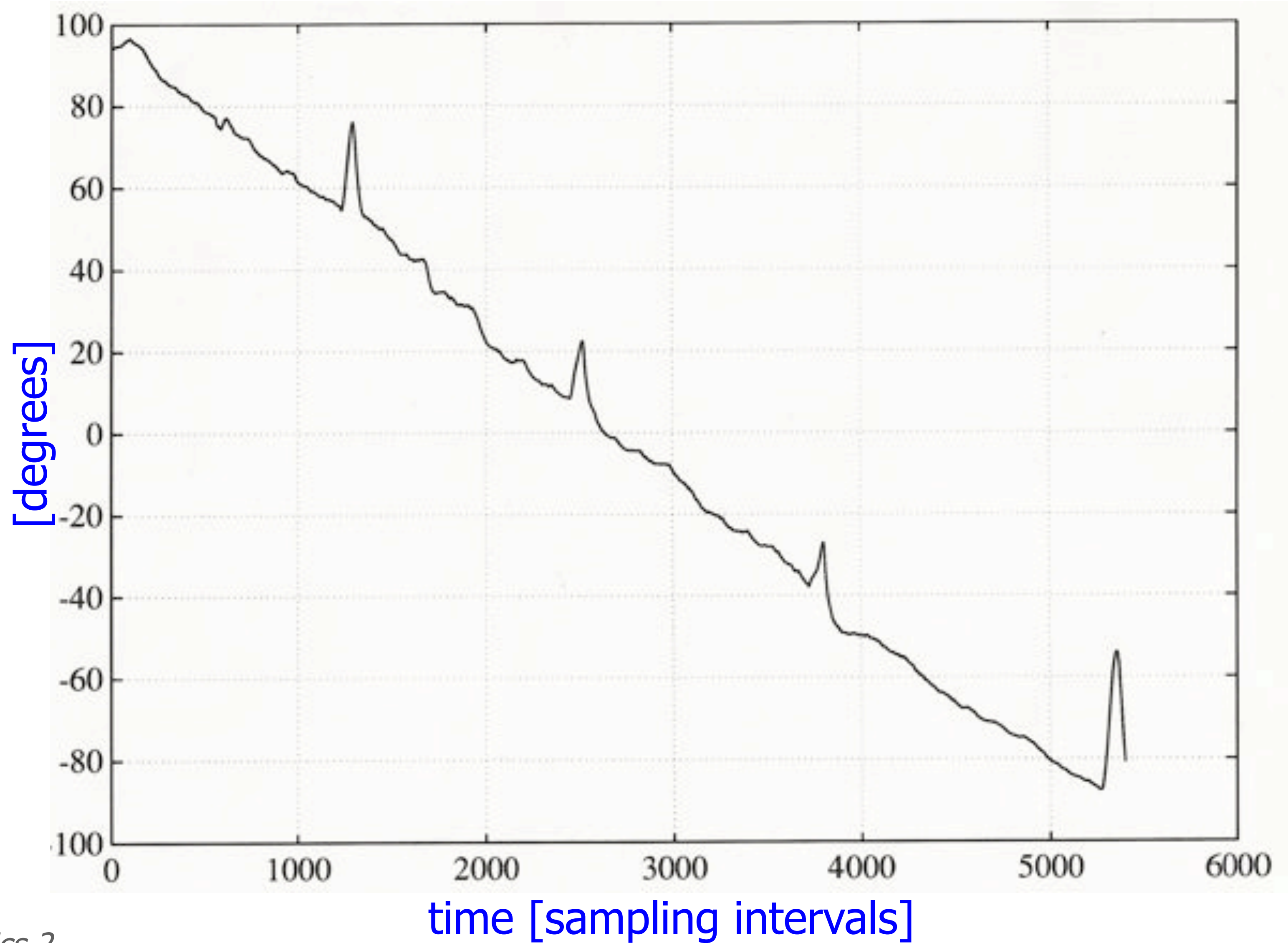
# Position-based estimation of the tangent

(for a circular surface traced at constant speed)

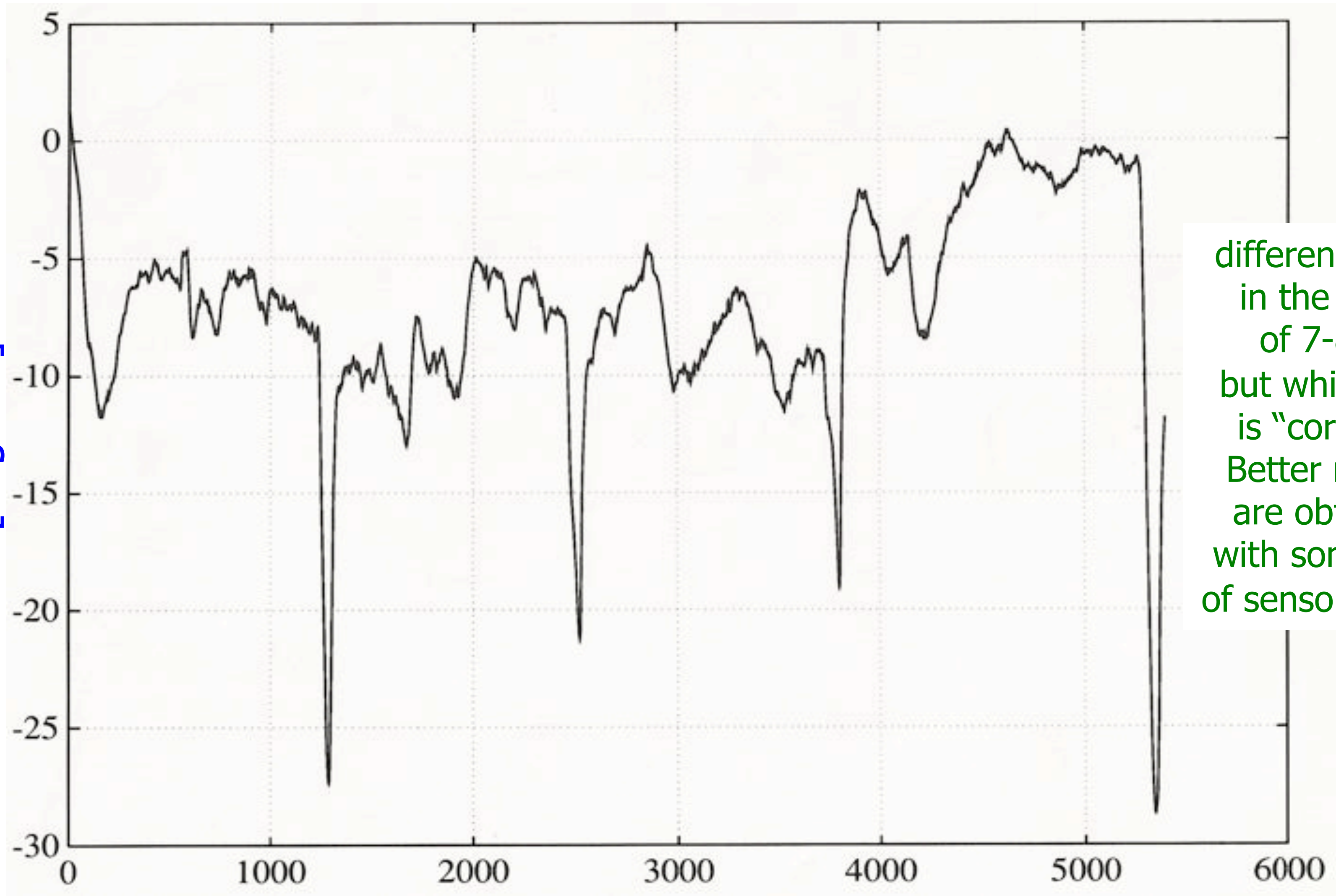


# Force-based estimation of the tangent

(for the same circular surface traced at constant speed)



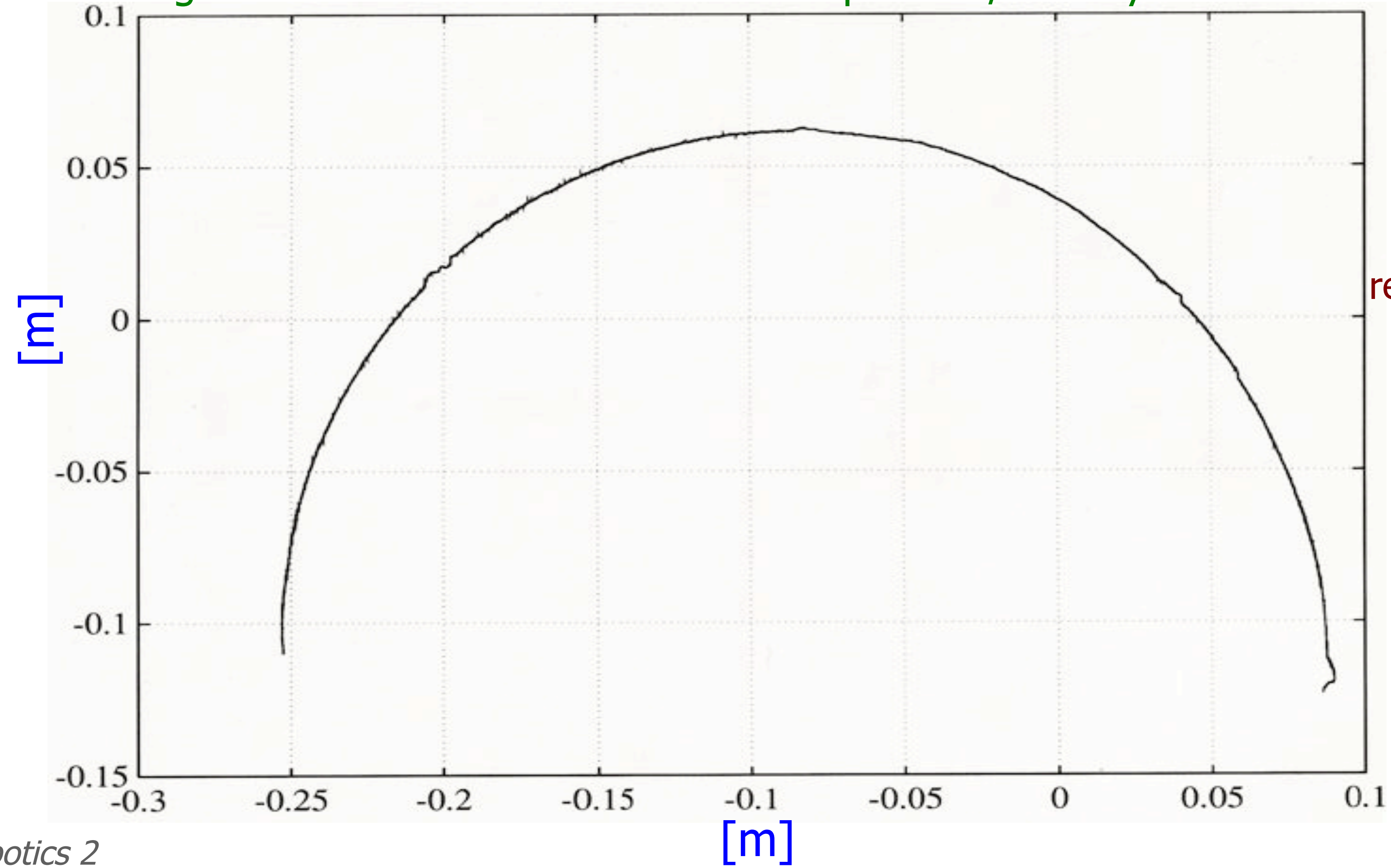
# Difference between estimated tangents




differences are in the order of 7-8°... but which one is "correct"? Better results are obtained with some kind of sensor fusion!

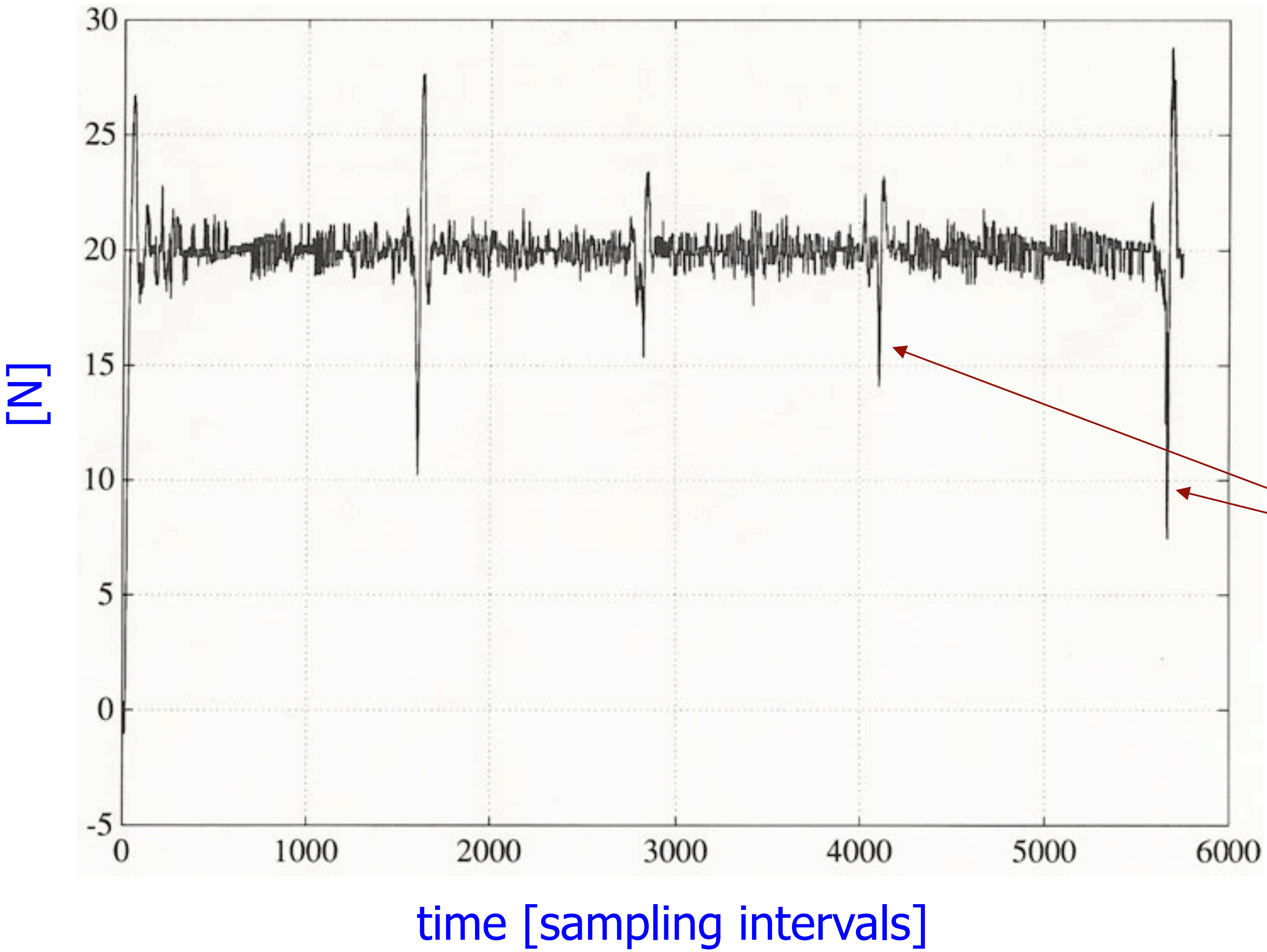
# Reconstructed surface profile

identification with a RLS (Recursive Least Squares) method, which continuously updates the coefficients of two quadratic polynomials fitting locally the unknown contour through data fusion from both force and position/velocity measurements




this is the reconstructed contour of a cinema "film reel" (of radius = 17 cm)

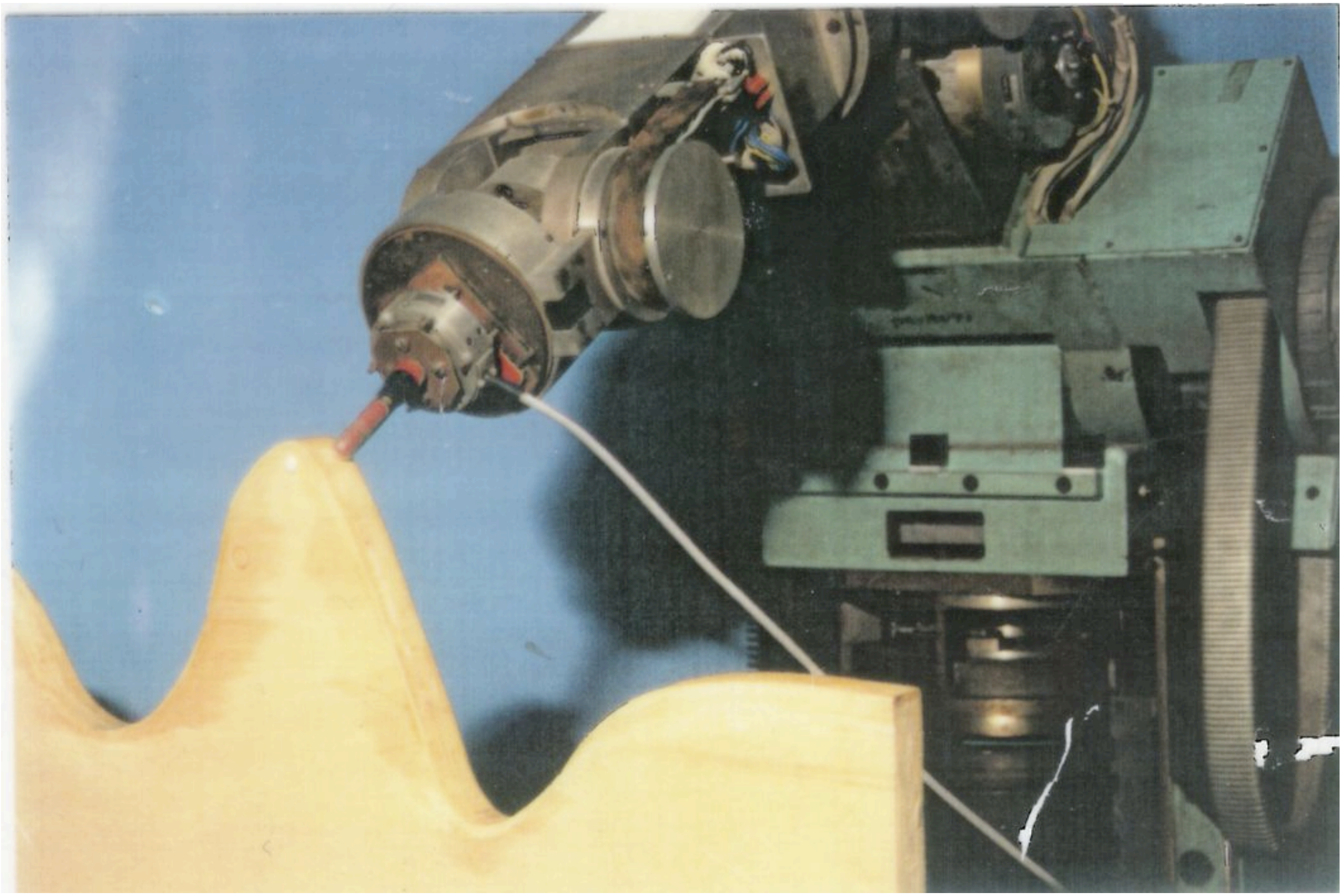
# Normal force




regulated to 20 N during simultaneous motion and identification

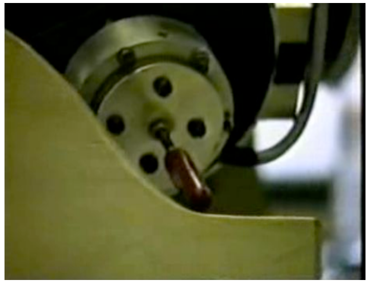
peaks correspond to grooves on the surface contour

# Contour identification and hybrid control
## performed simultaneously



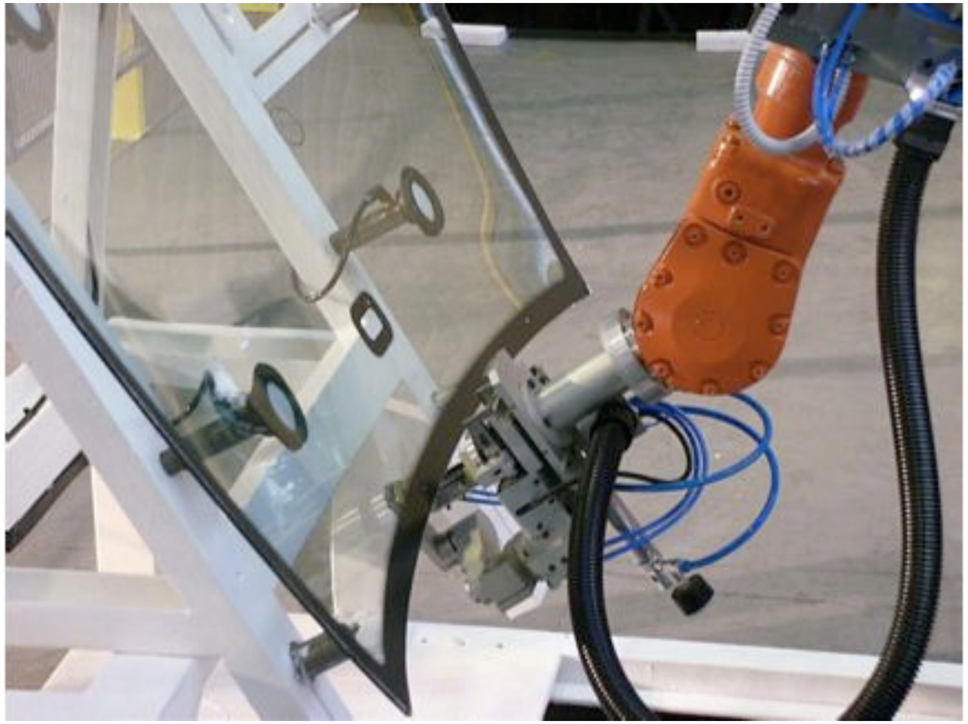
MIMO-CRF robot (DIS, Laboratorio di Robotica, 1992)

# Contour identification and hybrid control



video

# Robotized deburring of car windshields

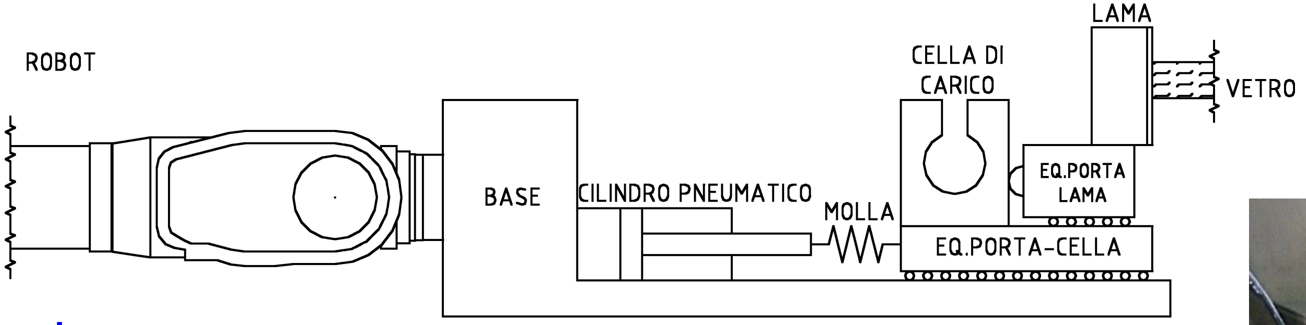


- car windshields with sharp edges and tolerances due to fabrication and excess of gluing material (PVB=Polyvinyl butyral) between glass layers
- robot end-effector follows a pre-programmed path, despite the small errors w.r.t. the nominal windshield profile, thanks to the passive compliance of the deburring work tool
- contact force between blades and work piece can be independently controlled by a pneumatic actuator in the work tool

the deburring robotic worktool contains in particular:

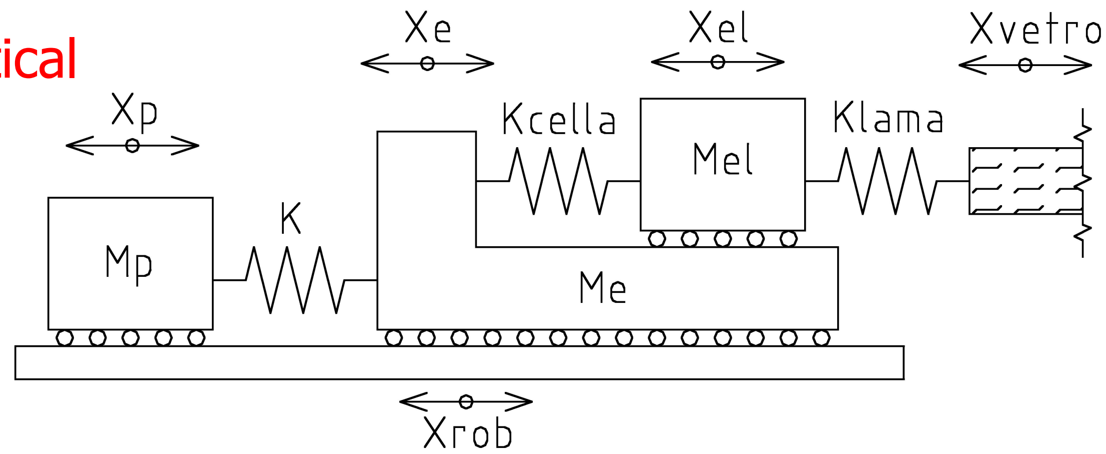
- two blades for cutting the exceeding plastic material (PVB), the first actuated, the second passively pushed by a spring
- a load cell for measuring the 1D applied force
- on-board control system

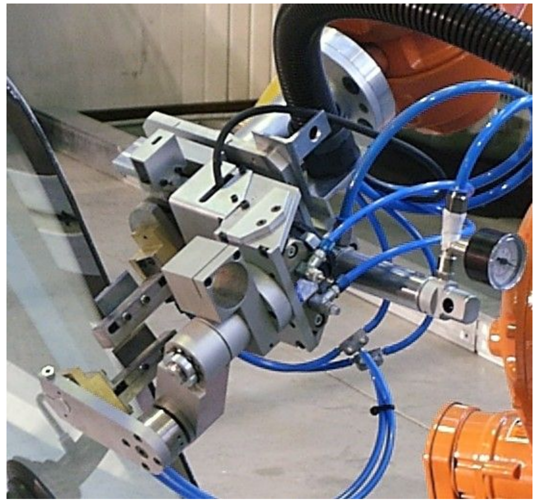
# Model of the deburring work tool




physical

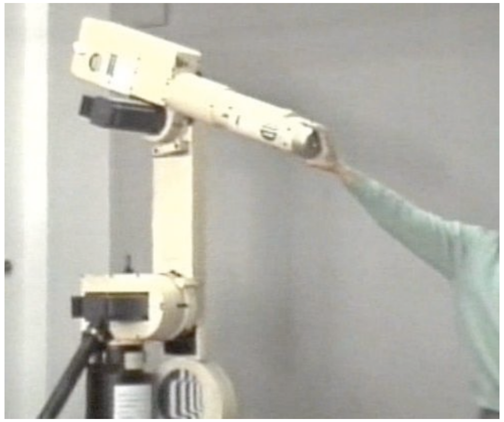
mathematical



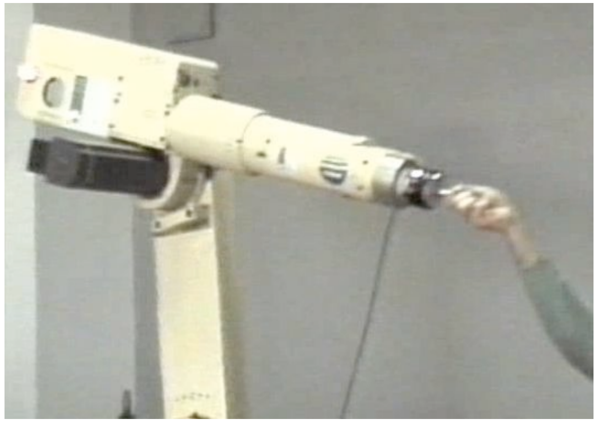



for a stability analysis of a force control loop in a single direction and in presence of multiple masses/springs (based on linear models and root locus techniques),
see again Eppinger & Seering, IEEE CSM, 1987 (material in the course web site)

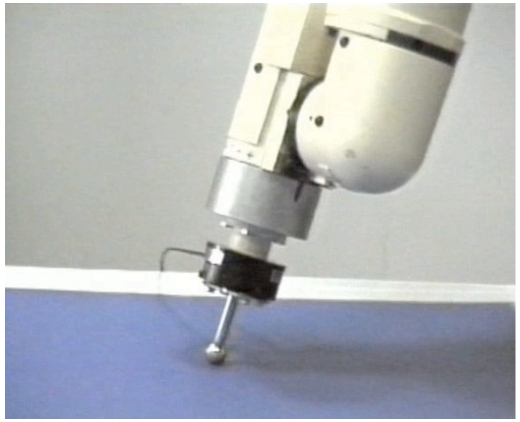
# Summary through video segments



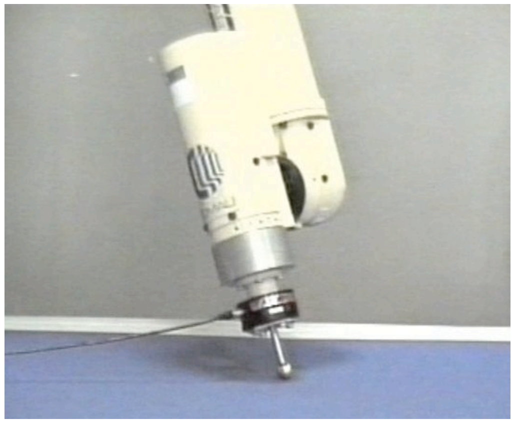
compliance control
(active Cartesian stiffness control without F/T sensor)



impedance control
(with F/T sensor)



force control
(realized as external loop providing the reference to an internal position loop)



hybrid force/position control

COMAU Smart robot
c/o Università di Napoli, 1994
(full video on course web site)