



Robotics 2

Hybrid Force/Motion Control

Prof. Alessandro De Luca

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI





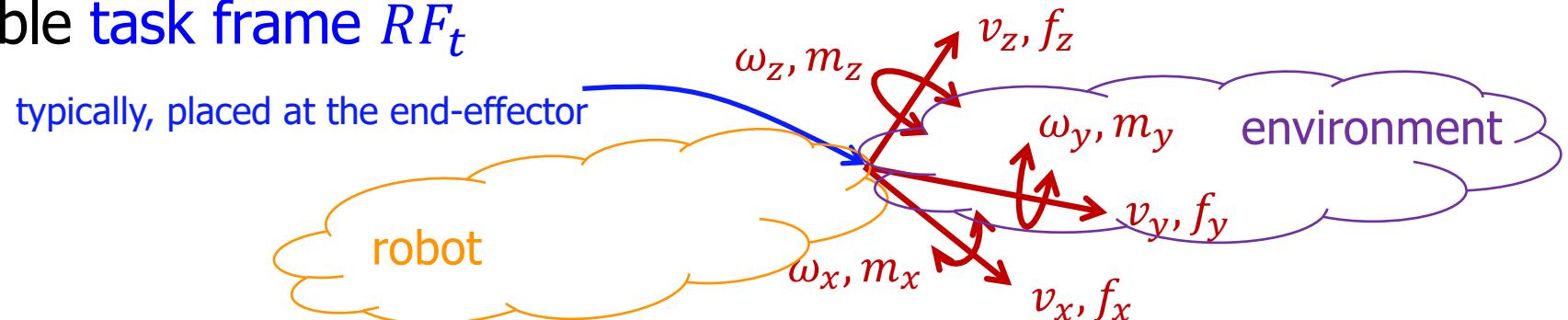
Hybrid force/motion control

- we consider **contacts/interactions** between a robot and a stiff environment that **naturally constrains** the end-effector motion
- **compared** to an approach using the constrained/reduced robot dynamics with (bilateral) **geometric constraints**, the **differences** are
 - the hybrid control law is designed in **ideal conditions**, but now unconstrained directions of motion and constrained force directions are defined in a more direct way using a **task frame formalism**
 - all **non-ideal conditions** (compliant surfaces, friction at the contact, errors in contact surface orientation) are handled explicitly in the control scheme by a **geometric filtering of the measured quantities**
 - considering only signal components that should appear in certain directions based on the nominal task model, and treating those that should not be there as **disturbances** to be rejected
- the hybrid control law avoids to introduce conflicting behaviors (force control vs. motion control) in all task space directions!!



Natural constraints

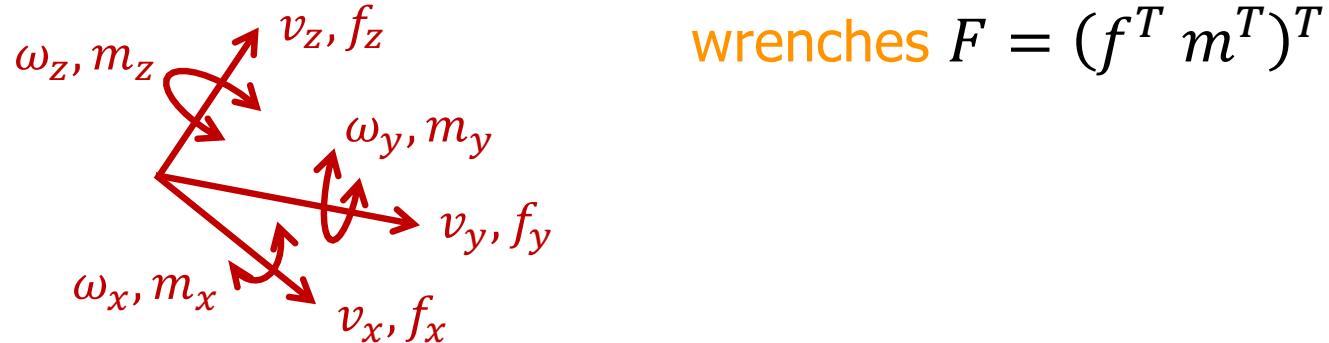
- in **ideal conditions** (robot and environment are perfectly rigid, contact is frictionless), **two sets of generalized directions** can be defined in the **task space**
 - end-effector motion (v/ω) is prohibited along/around **$6 - k$ directions** (since the environment reacts there with forces/torques)
 - reaction forces/torques (f/m) are absent along/around **k directions** (where the environment does not prevent end-effector motions)
- these constraints have been called the **natural constraints** on force and motion associated to the task geometry
- the two sets of directions are characterized through the axes of a suitable **task frame RF_t**





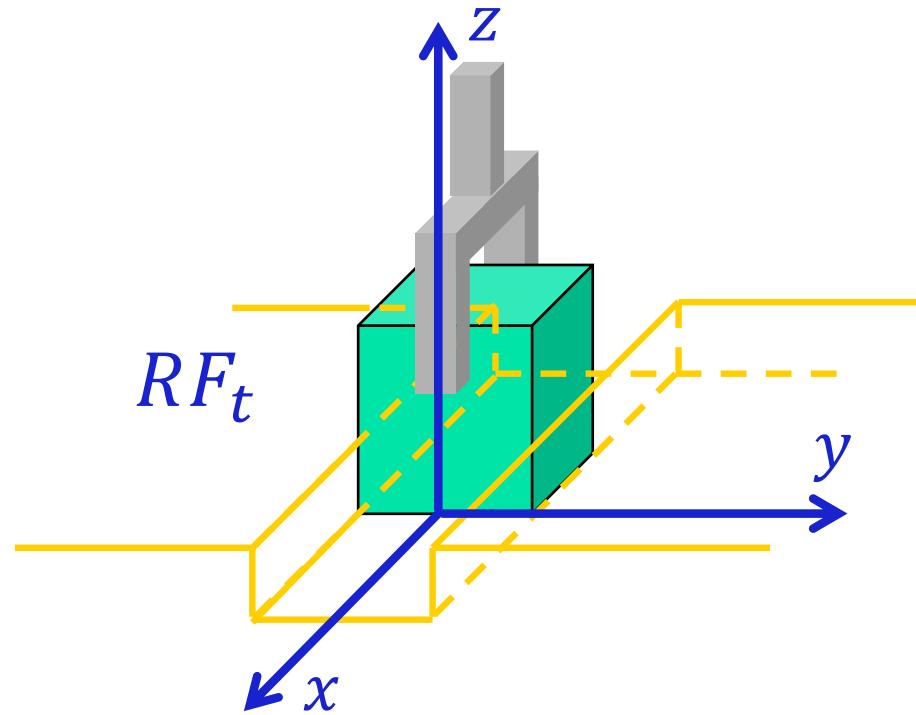
Artificial constraints

- the way **task execution** should be performed can be expressed in terms of so-called **artificial constraints** that specify the desired values (to be imposed by the control law)
 - for the **end-effector velocities** (v/ω) along/around **k directions** where feasible motions can occur
 - for the **contact forces/torques** (f/m) along/around **$6 - k$ directions** where admissible reactions of the environment can occur
- the two sets of directions are **complementary** (they cover the 6D generalized task space) and mutually **orthogonal**, while the **task frame** can be **time-varying** ("moves with task progress")
 - directions are intended as 6D **screws**: twists $V = (v^T \omega^T)^T$ and wrenches $F = (f^T m^T)^T$





Task frame and constraints - example 1



v = linear velocity
 ω = angular velocity
 f = force
 m = moment

$$6 - k = 4 \quad \left\{ \begin{array}{l} \\ \\ \end{array} \right.$$

$$k = 2 \quad \left\{ \begin{array}{l} \\ \\ \end{array} \right.$$

task: slide the cube along a guide

natural (geometric) constraints

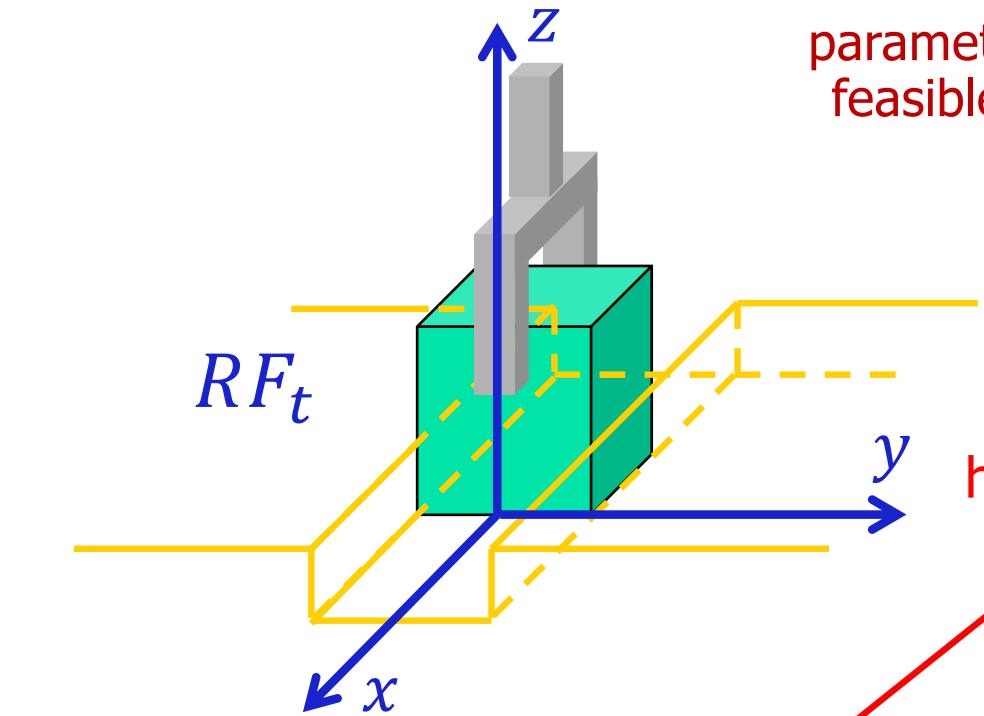
$$\left. \begin{array}{l} v_y = v_z = 0 \\ \omega_x = \omega_z = 0 \\ f_x = m_y = 0 \end{array} \right\} \begin{array}{l} 6 - k = 4 \\ k = 2 \end{array}$$

artificial constraints
(to be imposed by the control law)

$$\left. \begin{array}{l} f_y = f_{y,des} (= 0) \text{ (to avoid internal stress)} \\ m_x = m_{x,des} (= 0), m_z = m_{z,des} (= 0) \\ f_z = f_{z,des} \\ \omega_y = \omega_{y,des} = 0 \text{ (to slide and not to roll !!)} \\ v_x = v_{x,des} \end{array} \right\}$$



Selection of directions - example 1



parametrization of feasible reactions

$$(f_m) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_y \\ f_z \\ m_x \\ m_z \end{pmatrix} = Y \begin{pmatrix} f_y \\ f_z \\ m_x \\ m_z \end{pmatrix}$$

parametrization of feasible motions

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

here, constant and unitary
("selection" of columns from the 6×6 identity matrix)

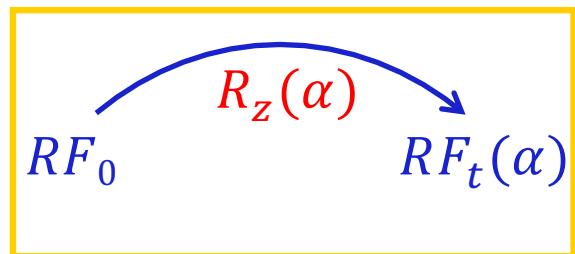
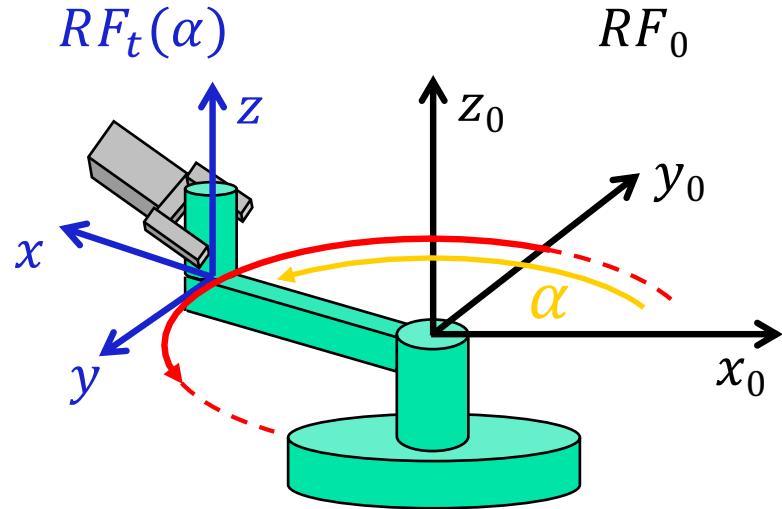
$$T^T Y = 0$$

reaction forces/torques do **not** perform work on feasible motions

$$(f^T \quad m^T) \begin{pmatrix} v \\ \omega \end{pmatrix} = 0$$



Task frame and constraints - example 2



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

$$f_x = f_{x,des} (= 0), f_z = f_{z,des} (= 0)$$

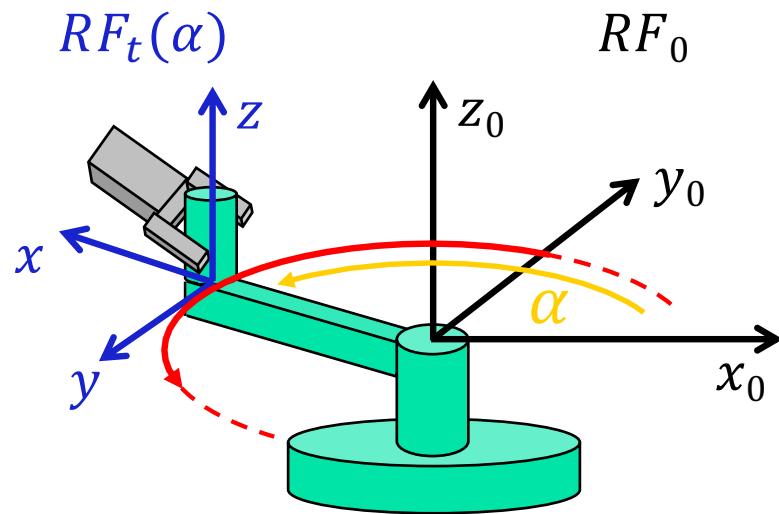
$$m_x = m_{x,des} (= 0), m_y = m_{y,des} (= 0)$$

$$v_y = v_{y,des} \text{ (the tangent speed of rotation)}$$

$$\omega_z = \omega_{z,des} \text{ (= 0 if handle should not spin)}$$



Selection of directions – example 2



parametrization of feasible motions

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

$$= T(\alpha) \begin{pmatrix} v_y \\ \omega_z \end{pmatrix}$$

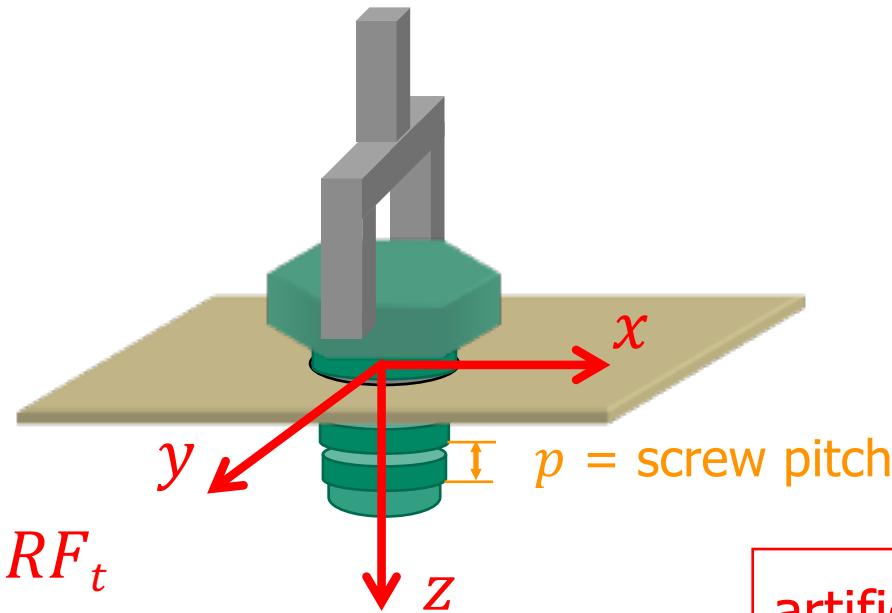
$$T^T(\alpha)Y(\alpha) = 0$$

parametrization of feasible reactions

$$\begin{pmatrix} {}^0f \\ {}^0m \end{pmatrix} = \begin{pmatrix} R^T(\alpha) & 0 \\ 0 & R^T(\alpha) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_x \\ f_z \\ m_x \\ m_y \end{pmatrix} = Y(\alpha) \begin{pmatrix} f_x \\ f_z \\ m_x \\ m_y \end{pmatrix}$$



Task frame and constraints - example 3



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**

task: insert a screw in a bolt

natural constraints (partial...)

$$v_x = v_y = 0$$

$$\omega_x = \omega_y = 0$$

artificial constraints (abundant...)

$$f_x = f_{x,des} = 0, f_y = f_{y,des} = 0$$

$$m_x = m_{x,des} = 0, m_y = m_{y,des} = 0$$

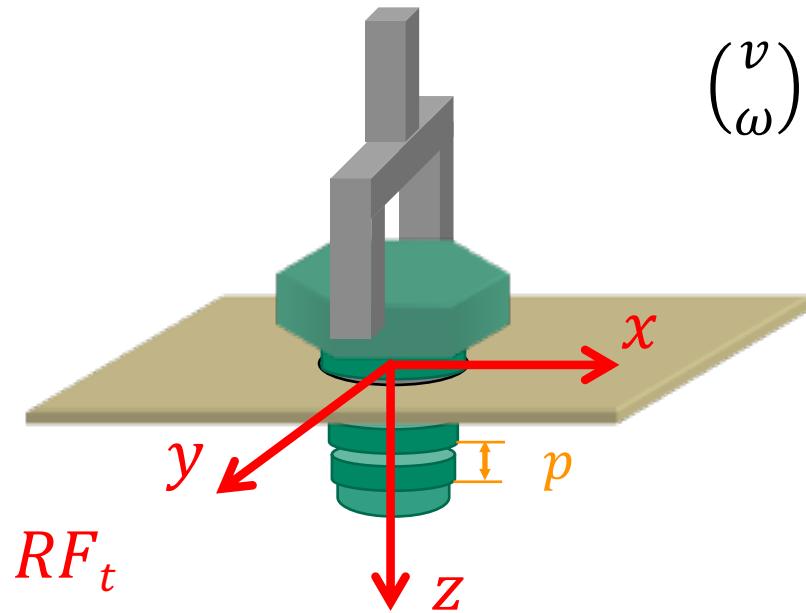
$$v_z = v_{z,des}, \omega_z = \omega_{z,des} = (2\pi/p)v_{z,des}$$

$$f_z = f_{z,des}, m_z = m_{z,des} (\text{a function of } f_{z,des})$$

wrench (force/torque) direction should be **orthogonal** to motion twist!



Selection of directions – example 3

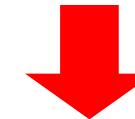


the columns of \mathbf{T} and \mathbf{Y}
do not necessarily coincide
with selected columns
of the 6×6 identity matrix
 \Rightarrow generalized (screw)
directions

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

or $\omega_z = 2\pi \frac{v_z}{p}$

\mathbf{Y} : such that $T^T \mathbf{Y} = 0$



$$f_z = -\frac{2\pi}{p} m_z$$

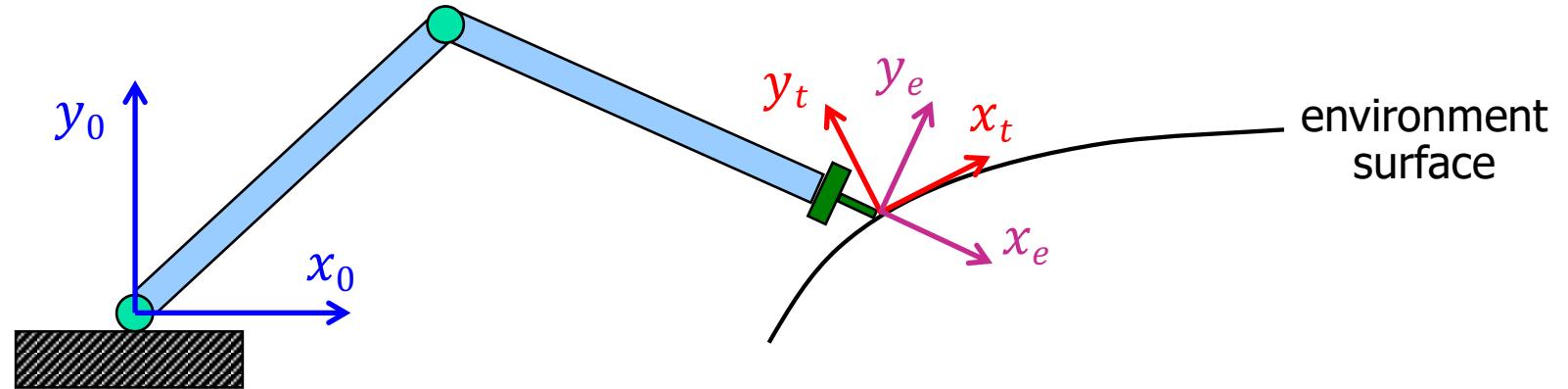
$(6 - k = 5)$

$$\begin{pmatrix} f \\ m \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2\pi/p \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_x \\ f_y \\ m_x \\ m_y \\ m_z \end{pmatrix} = \mathbf{Y} \begin{pmatrix} f_x \\ f \\ f_x \\ f_y \\ f_z \end{pmatrix}$$



Frames of interest – example 4

planar motion of a 2R robot in pointwise 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 that drive 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 \Rightarrow the **task frame!**



General parametrization of hybrid tasks

a “description” of robot-environment contact type:
it implicitly defines the task frame

$$\begin{cases} \begin{pmatrix} v \\ \omega \end{pmatrix} = T(s)\dot{s} & s \in \mathbb{R}^k \\ \text{parametrizes robot E-E free motion} \\ \begin{pmatrix} f \\ m \end{pmatrix} = Y(s)\lambda & \lambda \in \mathbb{R}^{M-k} \\ \text{parametrizes reaction forces/torques} \end{cases}$$

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

+

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



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

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

robot dynamics

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

robot kinematics

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



Hybrid force/velocity control

- **control objective:** to impose desired task evolution to the parameters s of **motion** and to the parameters λ of **force**

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

- the control law is designed in **two steps**

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

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

2. (**linear**) design of a_s and a_λ 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

Note: in “simple” cases, \dot{s} and $\dot{\lambda}$ are just single components of v or ω and of f or m ; accordingly, T and Y will be simple 0/1 selection matrices



Feedback linearization in task space

$$J(q)\dot{q} = \begin{pmatrix} v \\ \omega \end{pmatrix} = T(s)\dot{s} \rightarrow J\ddot{q} + \dot{J}\dot{q} = T\ddot{s} + \dot{T}\dot{s} \rightarrow \ddot{q} = J^{-1}(T\ddot{s} + \dot{T}\dot{s} - \dot{J}\dot{q})$$

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

(under the assumptions made)

$$(M(q)J^{-1}(q)T(s) : -J^T(q)Y(s)) \begin{pmatrix} \ddot{s} \\ \lambda \end{pmatrix} + M(q)J^{-1}(q)(\dot{T}(s)\dot{s} - \dot{J}(q)\dot{q}) + S(q, \dot{q})\dot{q} + g(q) = u$$

$$u = (MJ^{-1}T : -J^TY) \begin{pmatrix} a_s \\ a_\lambda \end{pmatrix} + MJ^{-1}(\dot{T}\dot{s} - \dot{J}\dot{q}) + S\dot{q} + g$$

linearizing and
decoupling
control law

$$\rightarrow \begin{pmatrix} \ddot{s} \\ \lambda \end{pmatrix} = \begin{pmatrix} a_s \\ a_\lambda \end{pmatrix} \quad \left. \begin{array}{l} k \\ M - k \end{array} \right\}$$

s has “relative degree” = 2
λ has “relative degree” = 0



Stabilization with a_s and a_λ

as usual, it is sufficient to apply **linear** control techniques for the exponential stabilization of tracking errors (on each single, input-output decoupled channel)

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

$K_P, K_D > 0$
and diagonal

$$\ddot{e}_s + K_D \dot{e}_s + K_P e_s = 0 \quad \rightarrow \quad e_s = s_d - s \rightarrow 0$$

$K_I \geq 0$
diagonal

$$a_\lambda = \lambda_d + K_I \int (\lambda_d - \lambda) dt$$

$a_\lambda = \lambda_d$ would be enough,
but adding an integral
with the **force error**
gives more robustness
to (constant) disturbances

$$e_\lambda + K_I \int e_\lambda dt = 0 \quad \rightarrow \quad e_\lambda = \lambda_d - \lambda \rightarrow 0$$

we need “values” for s , \dot{s} and λ to be
extracted from actual **measurements** !



“Filtering” position and force measures

- s and \dot{s} are 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, \dot{s})

example

$${}^0r = {}^0f(q) = \begin{pmatrix} L \cos s \\ L \sin s \\ 0 \end{pmatrix} \rightarrow s = \text{atan2}\{{}^0f_y(q), {}^0f_x(q)\}$$

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

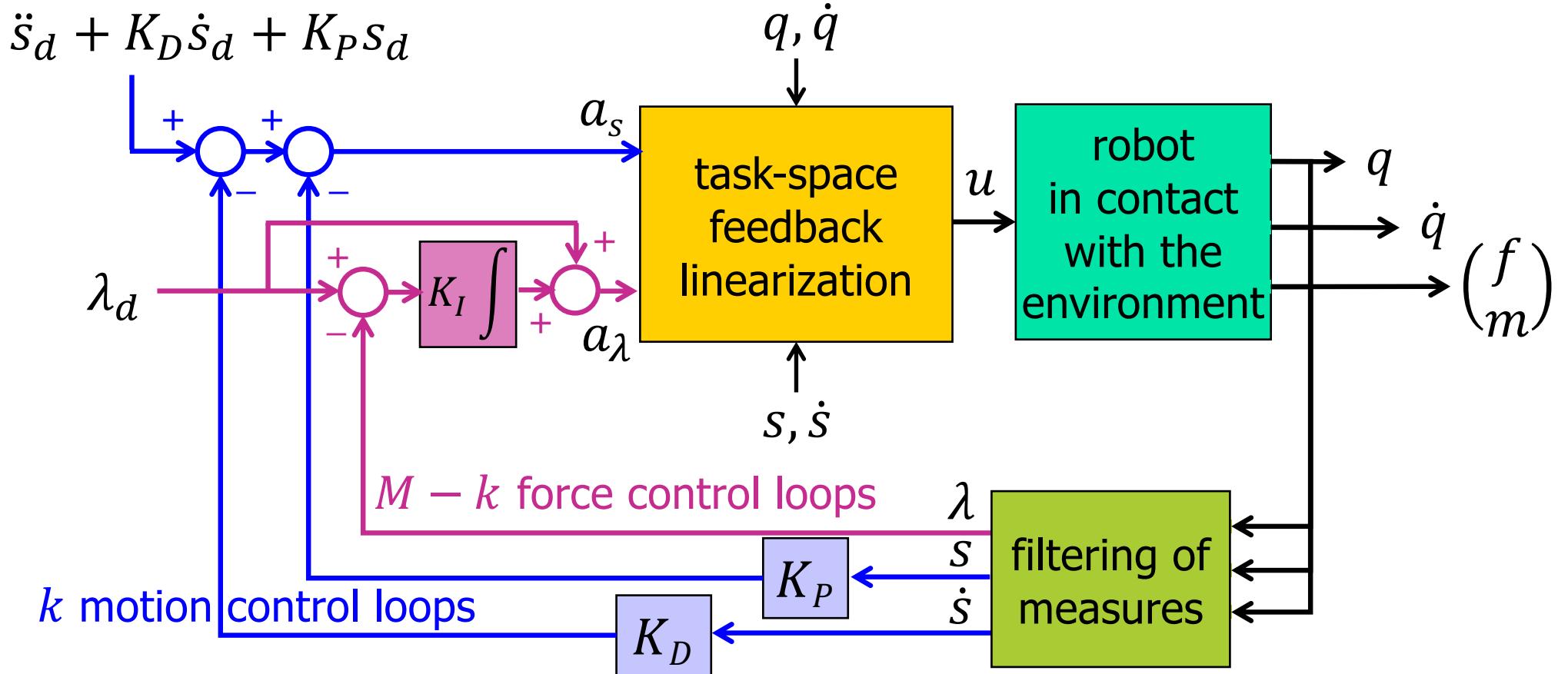
- λ is obtained from force/torque measures at end-effector

$$\begin{pmatrix} f \\ m \end{pmatrix} = Y(s)\lambda \rightarrow \lambda = Y^\#(s) \begin{pmatrix} f \\ m \end{pmatrix}$$

pseudoinverses
of “tall” matrices
having full
column rank, e.g.,
 $T^\# = (T^T T)^{-1} T^T$
(or weighted)



Block diagram of hybrid control



usually $M = 6$ (complete 3D space)

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

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



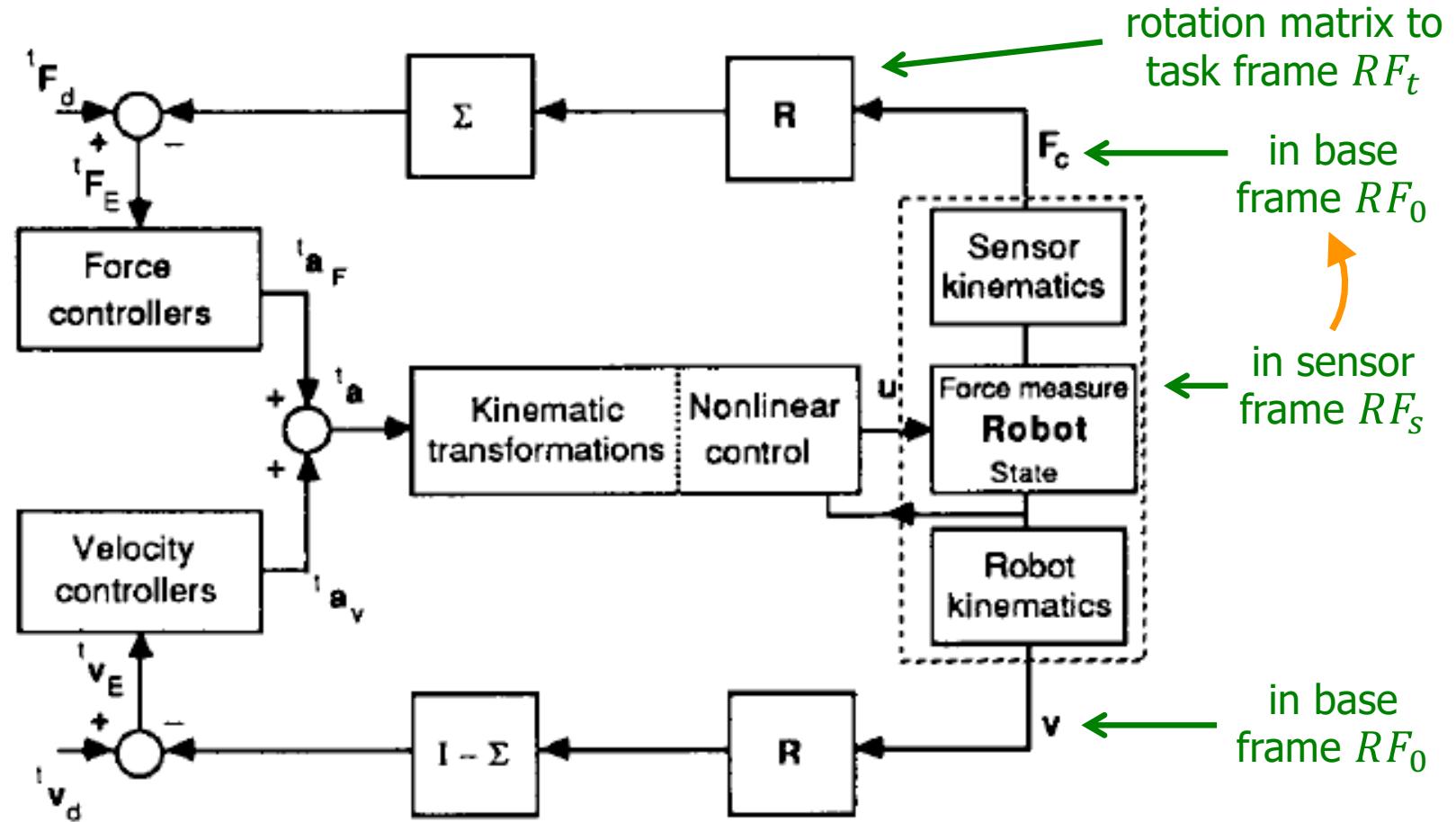
Block diagram of hybrid control

simpler case of 0/1 selection matrices

compact notation
in this slide

$$F = \begin{pmatrix} f \\ m \end{pmatrix}$$

$$V = \begin{pmatrix} v \\ \omega \end{pmatrix}$$



\dot{s} and λ are just single components of v (or ω) and f (or m)

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



Force control via an impedance model

- in a force-controlled direction of the hybrid task space, when the **contact stiffness is limited** (i.e., far from infinite, as assumed in the ideal case), one may use **impedance model ideas** to explicitly **control the contact force**
 - let x be the position of the robot along such a direction, x_d the (constant) contact point, $k_s > 0$ the contact (viz., sensor) stiffness, and $f_d > 0$ the desired contact force
- the impedance model is chosen then as

$$m_m \ddot{x} + d_m \dot{x} + k_s(x - x_d) = f_d$$

where the **force sensor** measures $f_s = k_s(x - x_d)$, and only $m_m > 0$ and $d_m > 0$ are free model parameters

- after feedback linearization ($\ddot{x} = a_x$), the command a_x is designed as

$$a_x = (1/m_m)[(f_d - f_s) - d_m \dot{x}]$$

which is a **P-regulator** of the desired force, **with velocity damping**

- the **same** control law works also before the contact ($f_s = 0$), guaranteeing a steady-state speed $\dot{x}_{ss} = f_d/d_m > 0$ in the **approaching phase**



First experiments with hybrid control

First Experiments with Hybrid Force/Velocity Control

Università di Roma "La Sapienza"
DIS, LabRob
February 1991



video

First Experiments with Hybrid Force/Velocity Control

(part II)

Università di Roma "La Sapienza"
DIS, LabRob
February 1991



video

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

Sources of inconsistency in force and velocity measurements



1. presence of **friction** at the contact
 - a reaction force component appears that opposes motion in a “free” motion direction (in case of Coulomb friction, the tangent force intensity depends also on the applied normal force ...)
 2. **compliance** in the robot structure and/or at the contact
 - a (small) displacement may be present also along directions that are nominally “constrained” by the environment
- NOTE:** if the environment geometry at the contact is perfectly known, the task inconsistencies due to 1. and 2. on parameters s and λ are already **filtered out** by the pseudo-inversion of matrices T and Y
3. uncertainty on **environment geometry** at the contact
 - can be reduced/eliminated by real-time **estimation processes** driven by external sensors (e.g., vision –but also force!)

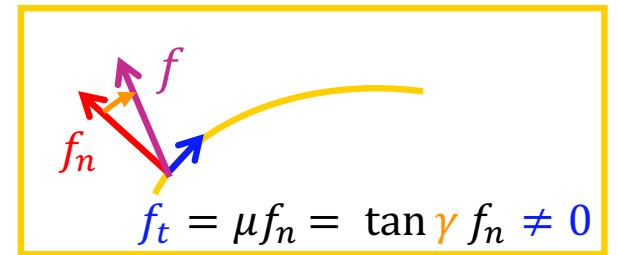


Estimation of an unknown surface

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

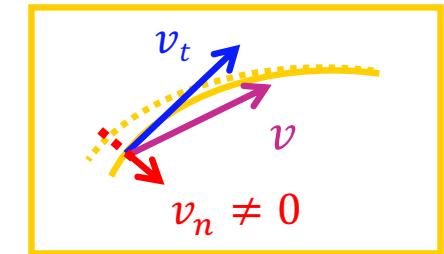
1. normal = nominal direction of measured **force**

... in the presence of contact motion with friction, the **measured** force f is slightly rotated from the actual normal by an (unknown) angle γ



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



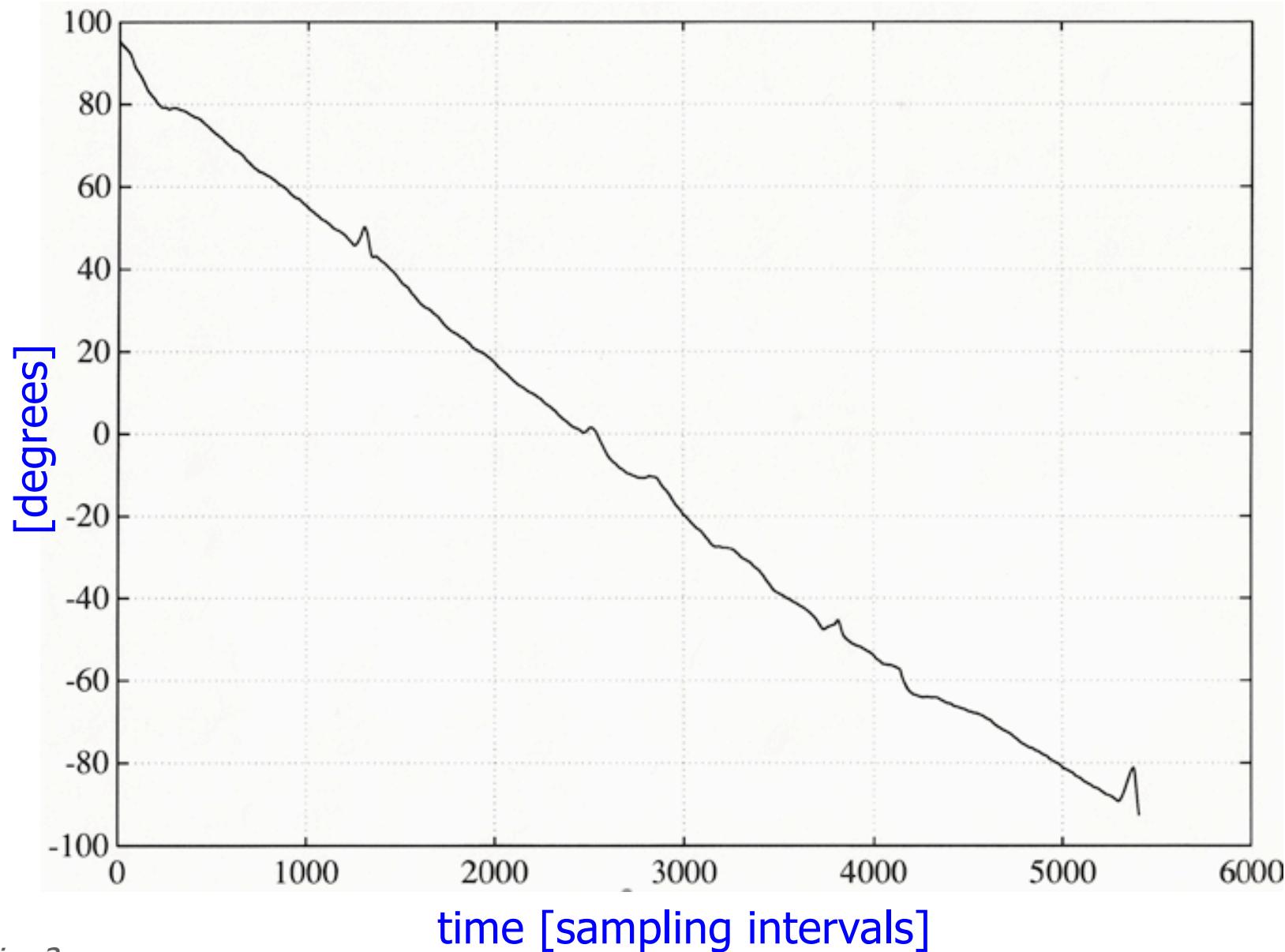
3. mixed method (**sensor fusion**) with RLS

- a. tangent direction is estimated by a **recursive least squares** method from position measurements
- b. friction angle is estimated by a **recursive least squares** method, using the current estimate of the tangent direction and from force measurements

to approach an unknown surface or to recover contact (in case of loss), the robot uses a simple exploratory logic



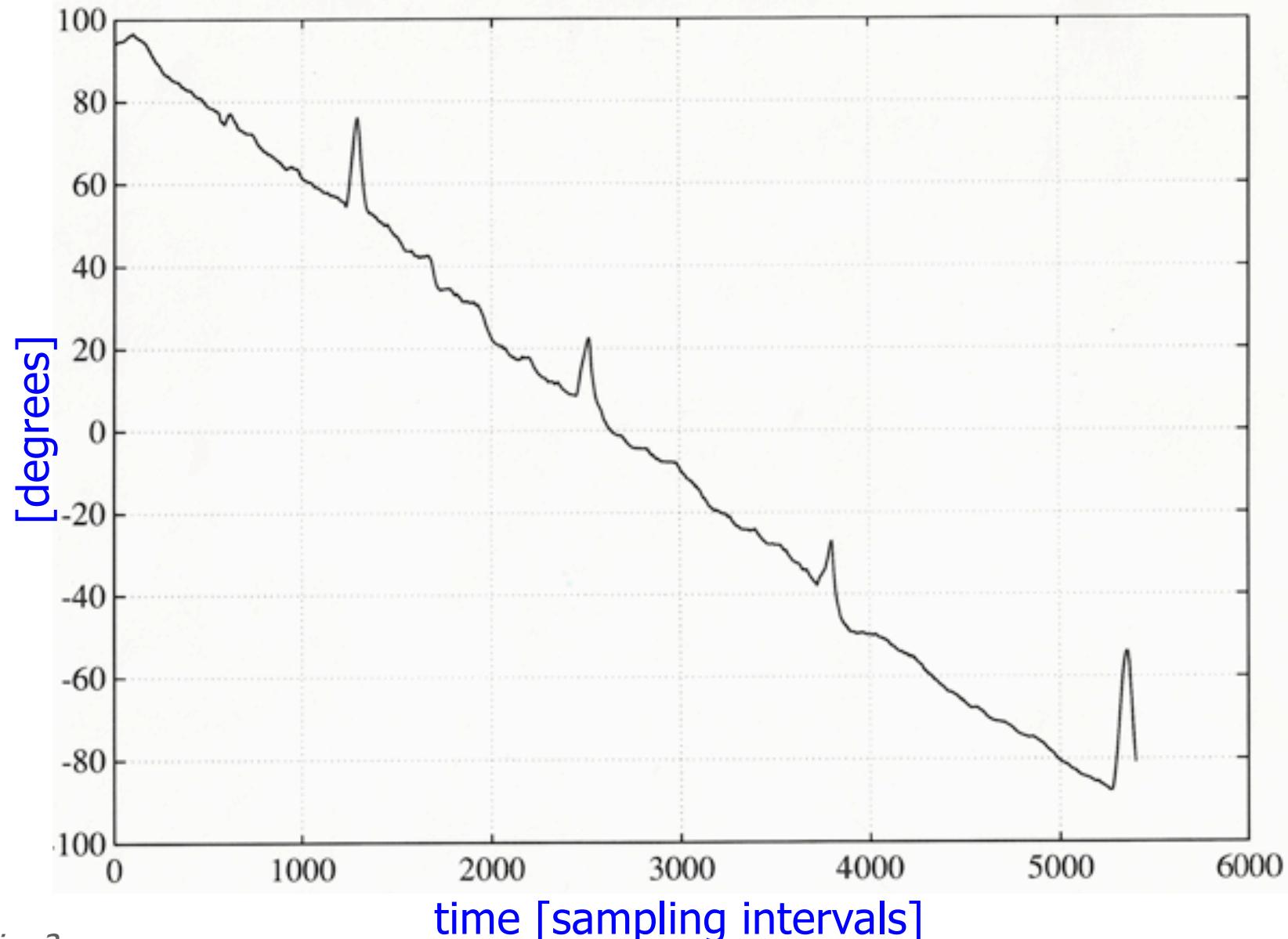
Position-based estimation of the tangent (for a circular surface traced at constant speed)





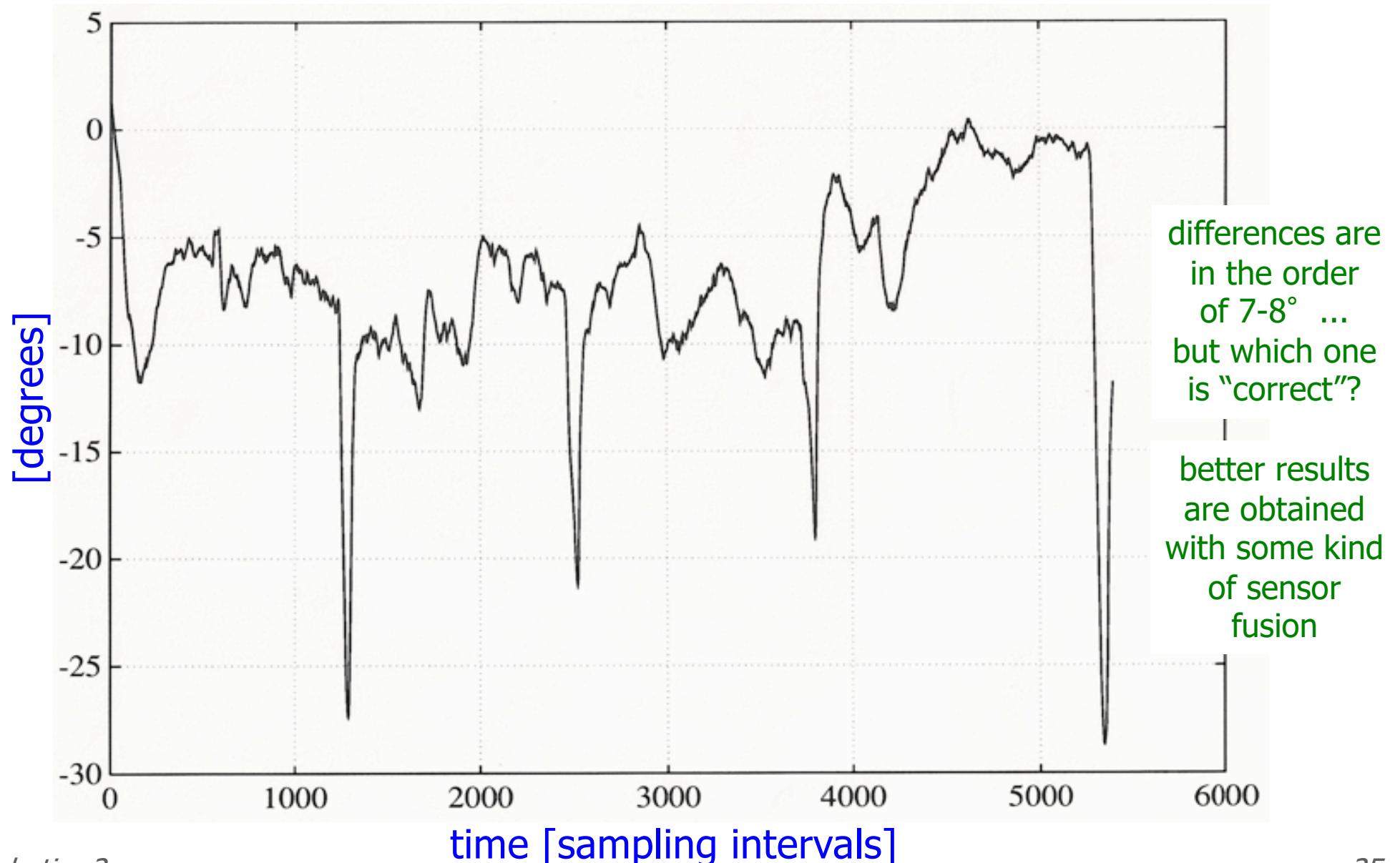
Force-based estimation of the tangent

(for the same circular surface traced at constant speed)





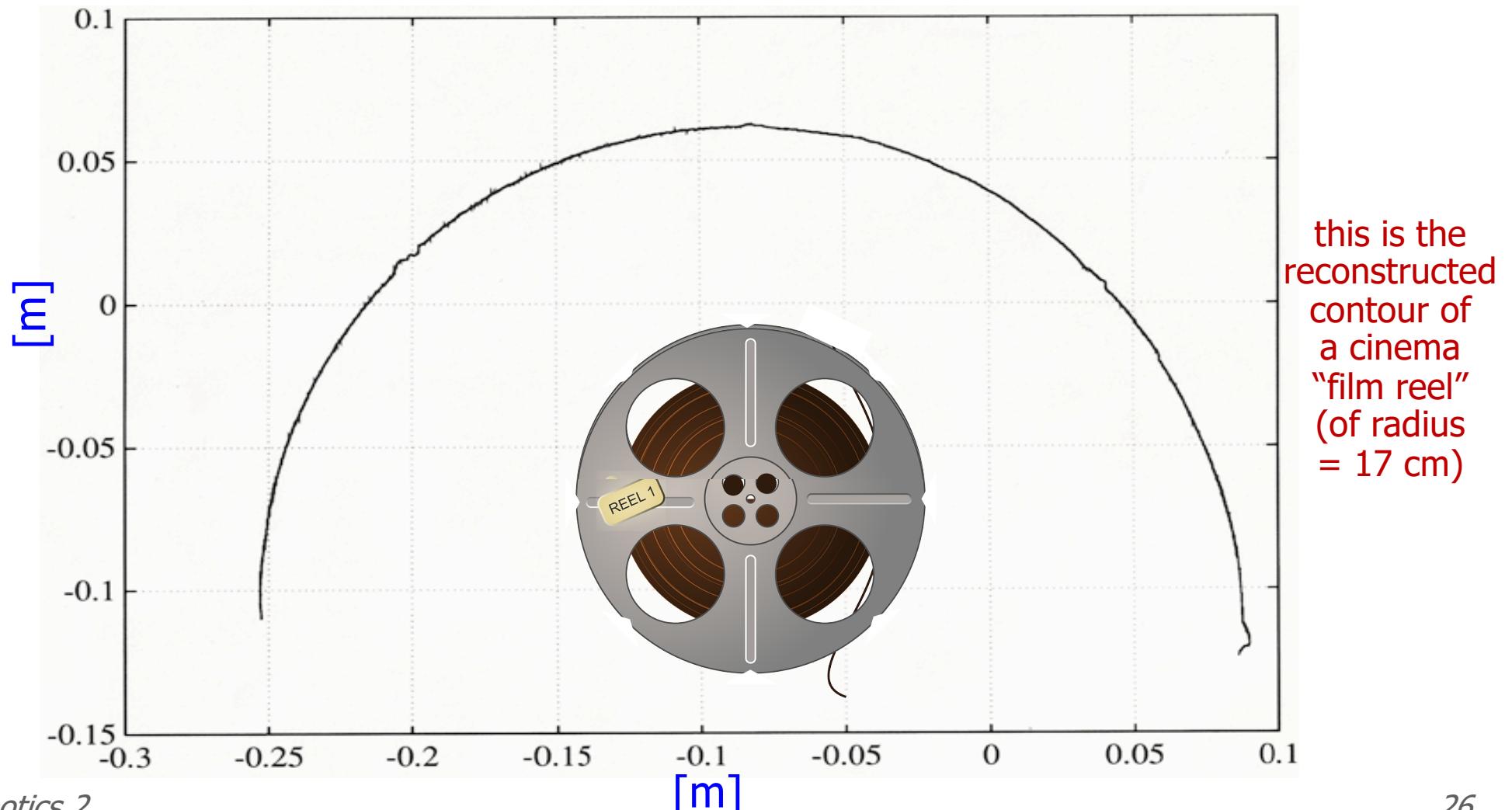
Difference between estimated tangents





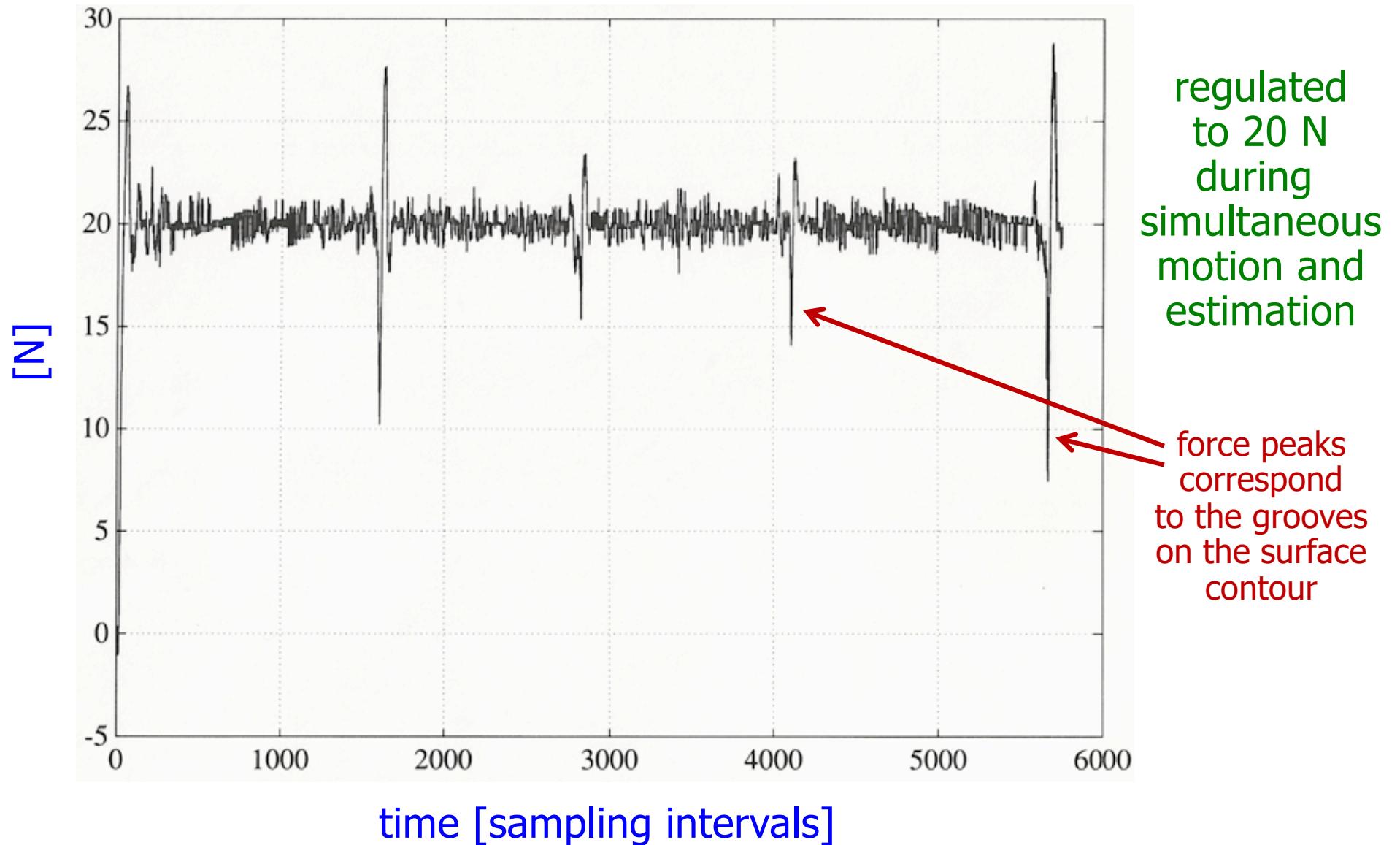
Reconstructed surface profile

estimation by a RLS (Recursive Least Squares) method: we continuously update the coefficients of two quadratic polynomials that fit locally the unknown contour, using data fusion from both force and position/velocity measurements



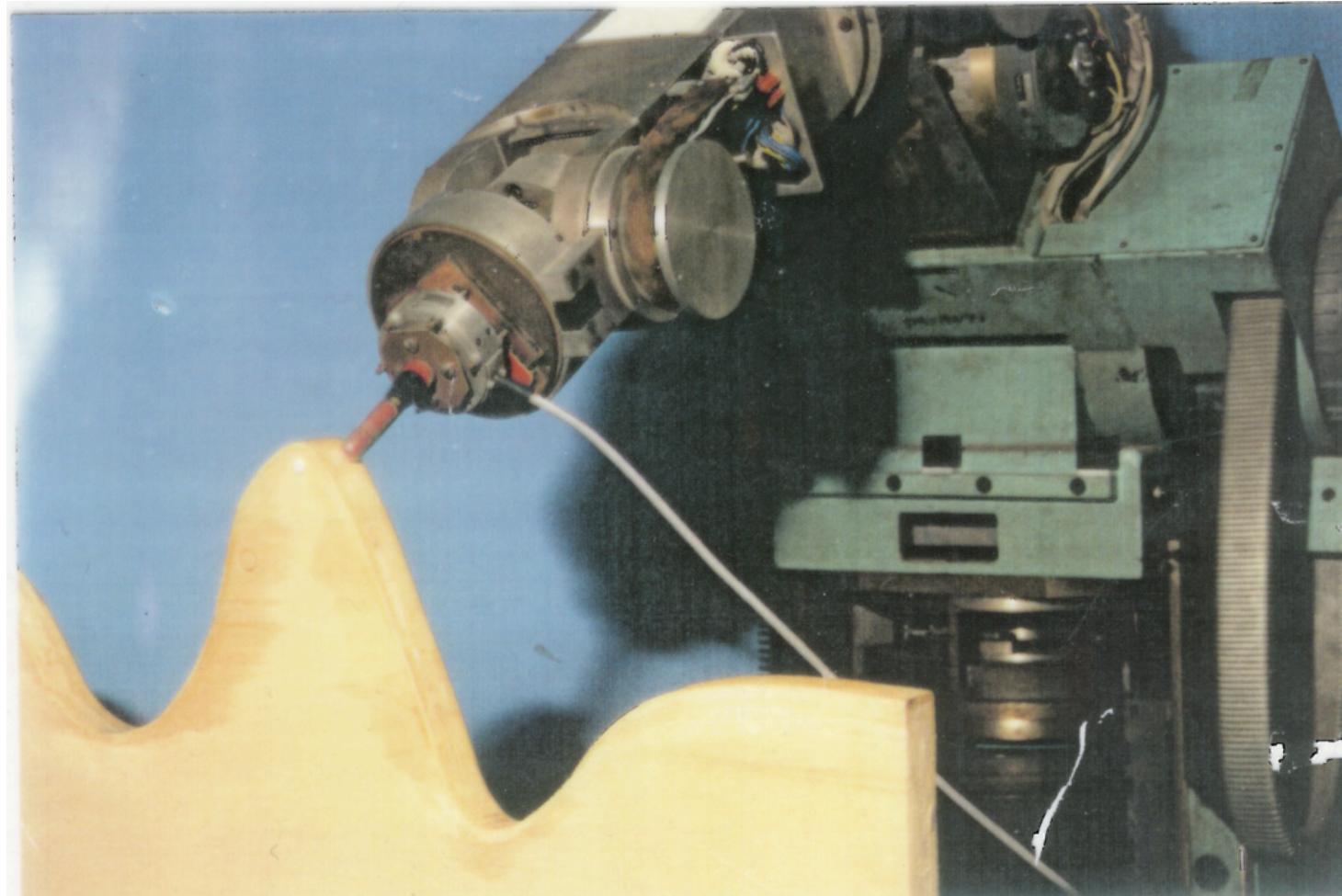


Normal force





Contour estimation and hybrid control performed simultaneously



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



Contour estimation and hybrid control

Hybrid Force/Velocity Control and Identification of Surfaces

**Università di Roma "La Sapienza"
DIS, LabRob
September 1992**



video



Robotized deburring of car windshields



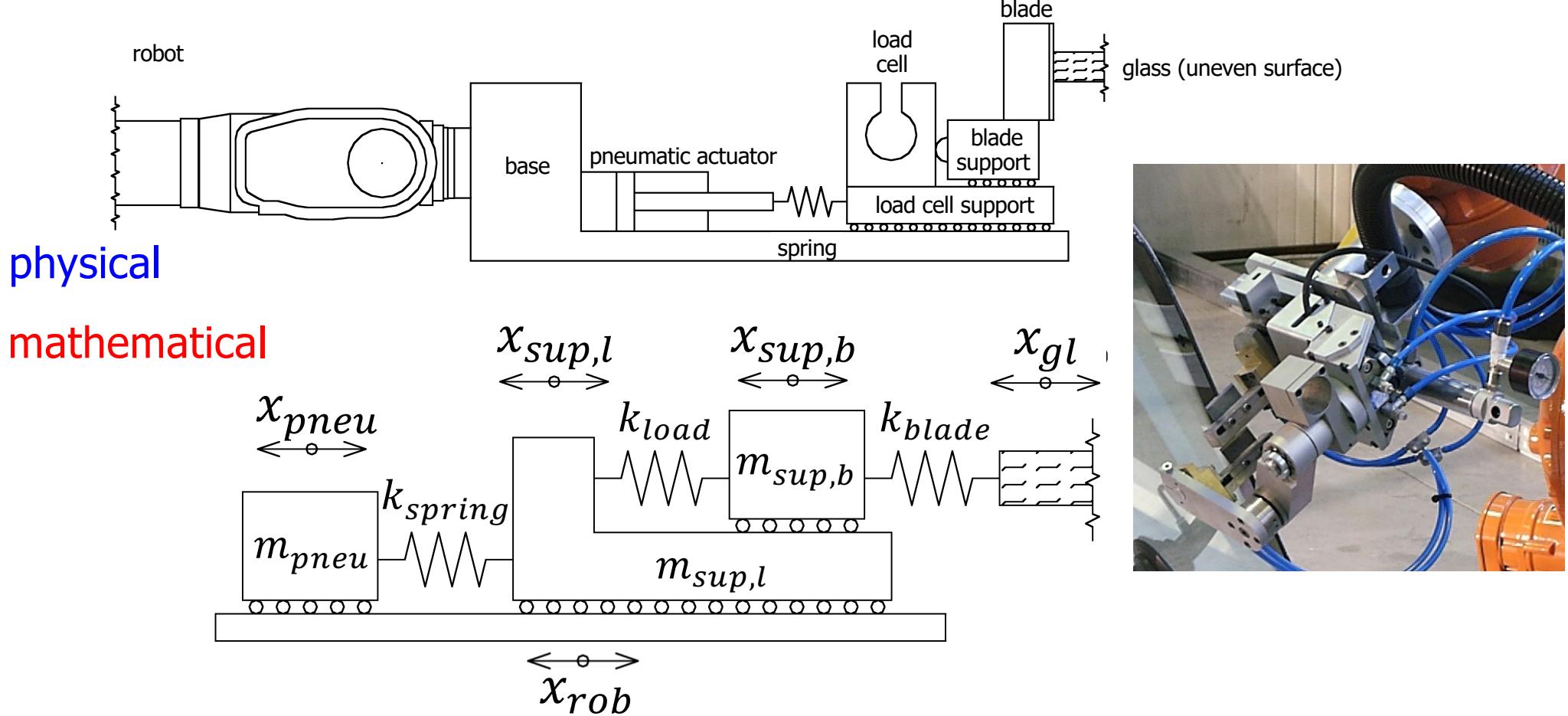
- car windshield with **sharp edges** and fabrication tolerances, with **excess of material** (PVB = Polyvinyl butyral for glueing glass layers) on the contour
- robot end-effector follows the pre-programmed path, despite the small errors w.r.t. the nominal windshield profile, thanks to the **passive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

the robotic deburring tool contains in particular

- **two blades** for cutting the exceeding plastic material (PVB), the first one actuated, the second passively pushed against the surface by a spring
- a **load cell** for measuring the 1D applied normal force at the contact
- on-board **control system**, exchanging data with the ABB robot controller



Model of the deburring work tool



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



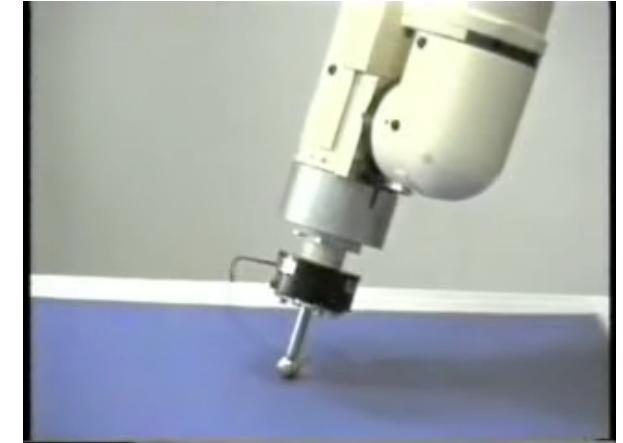
Summary through video segments



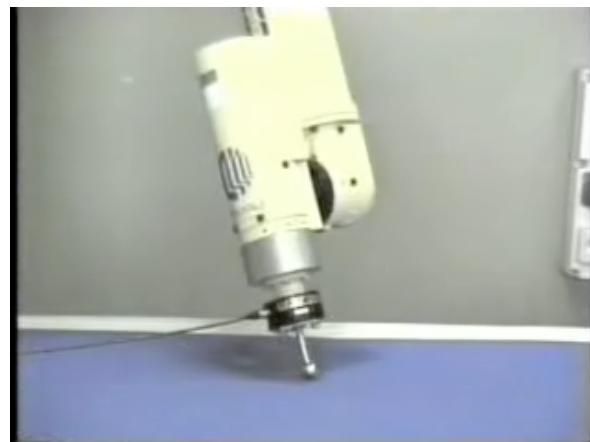
compliance control
(active Cartesian stiffness
control **without** F/T sensor)



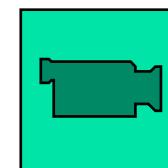
impedance control
(with F/T sensor)



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



hybrid force/position control



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



Appendix

- force control can also be realized as an external loop providing reference values to an internal motion loop (see video in slide #32)
- inner-outer (or cascaded) control scheme
 - angular position quantities (E-E orientation, errors, commands) can be expressed in different ways (Euler angles ϕ , rotation matrices R , ...)

