FEEDBACK CONTROL OF RIGID ROBOT MANIPULATORS
(open kinematic chains)

Patrick Danès
Associate Professor in Automatic Control at Université Paul Sabatier
LAAS-CNRS — Robotics & AI Dept – Robotics, Action, Perception Group
(0)561.33.78.25. – (0)685.67.76.49.
Patrick.Danes@laas.fr
<http://www.laas.fr/~danes>

INTRODUCTION

★ In manipulation robotics, the task is generally specified in the operational space

- point-to-point motion of the end effector
  - ▷ *e.g.*, pick and place
  - ▷ discrete set of (two or more) points
  - ▷ no control of the path of the end effector between them
- continuous path
  - ▷ *e.g.*, geometric straight line, welding contour, etc.
- continous path & time law along this path

but the actuators evolve in the generalized space

↪ generalized / operational reference motions must be handled

★ The problem is then how to elaborate the command signal which enables to servo the robot manipulator (in closed-loop) on this reference. A wide variety of solutions exists, depending on

- the mechanical structure of the robot
  - ▷ **P**,**R** joints
  - ▷ with/without reduction gear
- the control input signal
  - ▷ forces/torques
  - ▷ armature (rotor) voltage of a DC motor
- the considered servo problem
  - ▷ cf. the three aforementioned cases

★ These imply distinct implementations in terms of hardware and software. Always keep in mind that, as is usual in control,

Enhancement of the performances

$\Downarrow$

Increase of the complexity of the required control scheme

★ A bird's view on robot feedback control general strategies

- joint space control
- operational (cartesian) space control
- control in contact with the environment
  - ▷ impedance control
  - ▷ hybrid force/position control
  - ▷ etc.
- (exteroceptive-) sensor based control
  - ▷ vision-based control
  - ▷ laser-based control
  - ▷ etc.

↪ this course restricts to the three first topics (*in decreasing order* though, due to time reasons...)

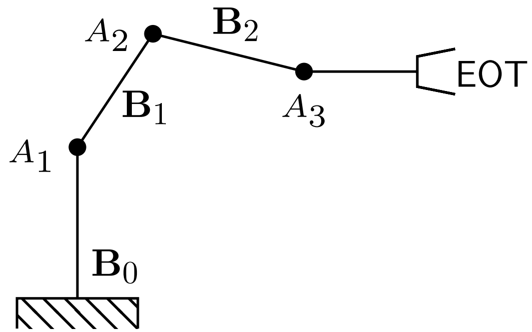
Chapter I

JOINT SPACE CONTROL

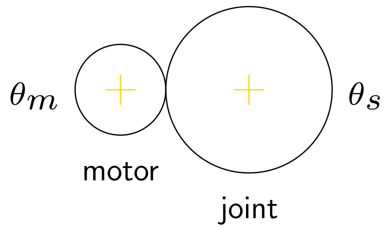
I.1 OVERVIEW

★ Features of any considered robot

- open kinematic chain
  - ▷ bodies (links)
  - ▷ articulation (joints): revolute (**R**) / prismatic (**P**)
- actuators: permanent magnet DC-motors
  - ▷ fixed stator, which produces a radial magnetic flux
  - ▷ armature, moving because of the current running through it
- gear train, connecting each DC-motor to a link
  - ▷ typical gear ratio $r \in [\frac{1}{200}; \frac{1}{20}]$
  - ⊕ decouples the system and reduces nonlinearity effects
  - ⊖ friction, elasticity, backlash
- sensors: proprioceptive/exteroceptive



EOT = end of arm tooling (end effector)



$$\theta_s = r\theta_m$$

τ_l on joint axis

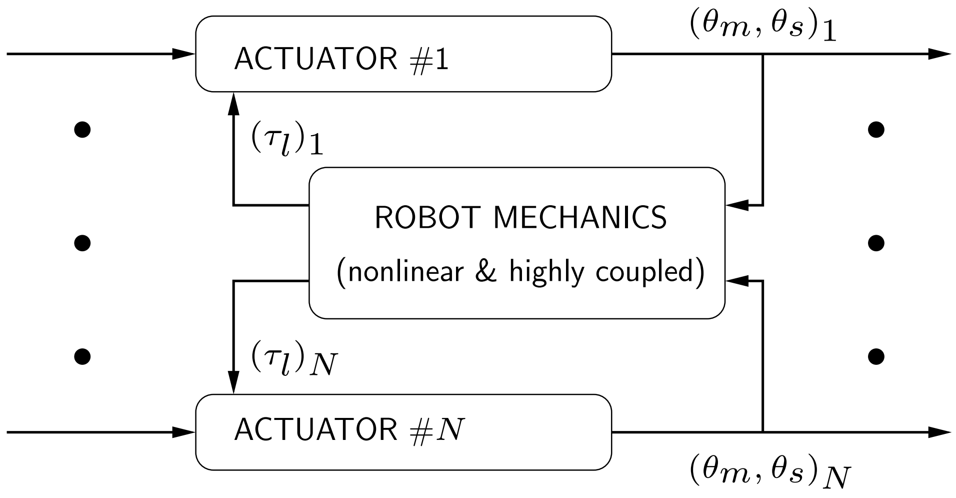
$\Downarrow$

$r\tau_l$ on actuator axis

★ Features of any considered robot (cont'd)

- one exception: direct-drive robots
  - ▷ joints driven by high-torque motors
  - ▷ no friction, elasticity, nor backlash but nonlinearities and couplings between the joints imply distinct control strategies

★ Overall schematic diagram

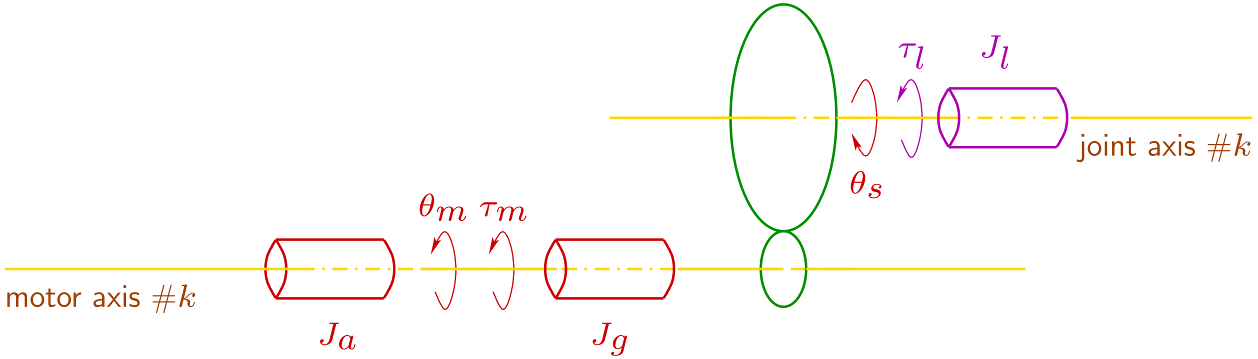


$\hookrightarrow$ Which feedback control strategies can be used for reference signal tracking and disturbances rejection?

I.2 AN ELEMENTARY APPROACH

I.2.1 Modeling of a permanent magnet DC-motor based actuation system

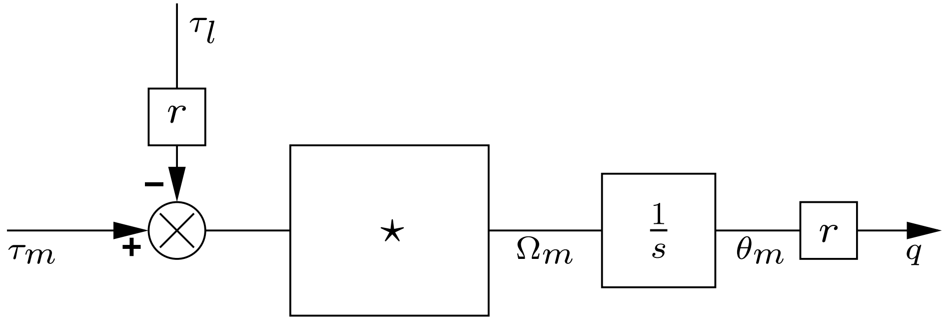
★ Schematic diagram (for each k^{th} articulation)



- ▷ J_a: actuator inertia
- ▷ J_g: gear train inertia
- ▷ J_l: load inertia
- ▷ τ_m: generalized generated effort (force/torque)
- ▷ τ_l: generalized effort on the secondary (load) axis
  - · nonlinear function of $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, etc.
  - · equiv. effort brought back on primary axis $= r\tau_l$

★ Expressing causes and effects

- τ_m causes $\theta_m \leftrightarrow (\theta_s = q_k)$
  but τ_l opposes to τ_m via the effort $r\tau_l$ brought back on the primary axis

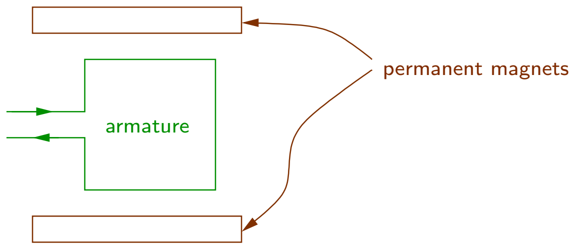


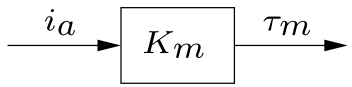
▷ fundamental law of dynamics: $J_m\ddot{\theta}_m = -B_m\dot{\theta}_m + \left(\tau_m - r\tau_l\right)$, with

- $J_m \triangleq J_a + J_g$: motor (*i.e.* actuator+gear) inertia
- B_m: coefficient of motor (*i.e.* brushes+gear) friction

$$\Rightarrow \frac{\Omega_m(s)}{\tau_m(s) - r\tau_l(s)} = \frac{1}{B_m + J_m s}$$

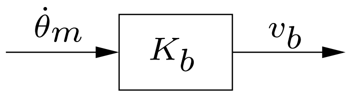
- what causes τ_m?...
  electromagnetic induction: the voltage v applied to the rotor (armature) of the motor, causes the current i_a, which in turn induces τ_m





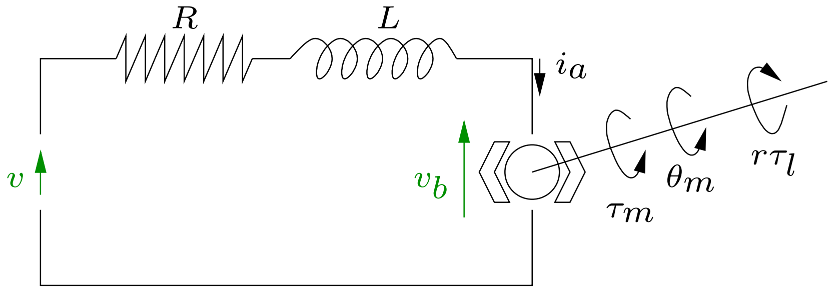
  ▷ $\tau_m = K_m i_a$, K_m torque constant

- but τ_m causes the motion θ_m, which itself gives rise to the back electromotive force (back emf) v_b which acts against v



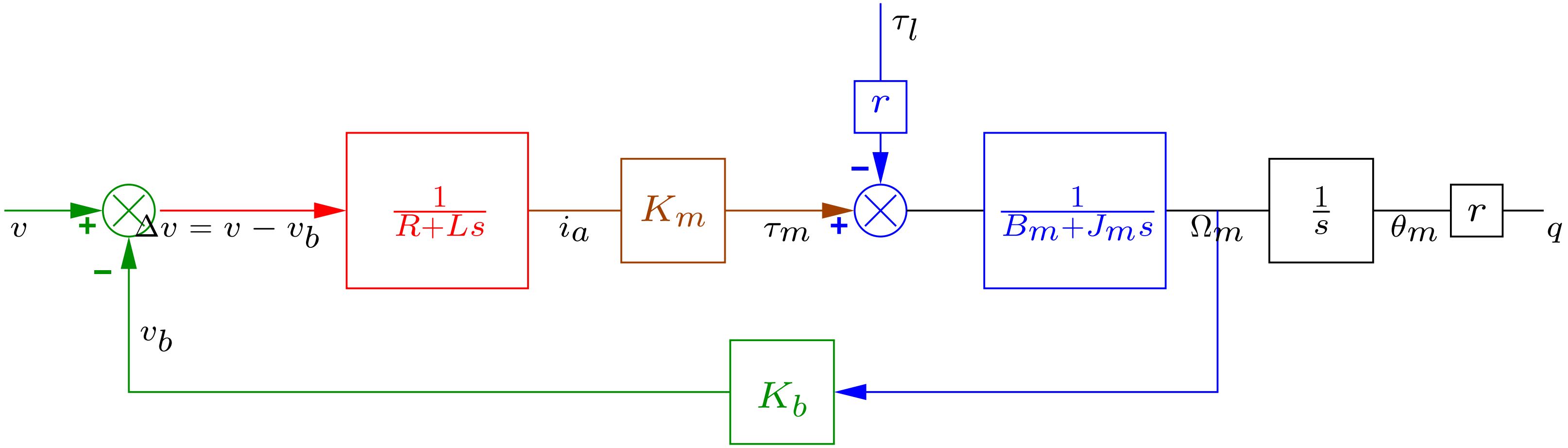
  ▷ $v_b = K_b \dot{\theta}_m$, K_b back emf constant

- last, $v - v_b$ causes i_a by (R, L being the armature's resistance and inductance)



  ▷ Kirchhoff's circuit laws: $L\frac{\mathrm{d}i_a}{\mathrm{d}t} + Ri_a = v - v_b \Rightarrow \frac{I_a(s)}{V(s) - V_b(s)} = \frac{1}{R + Ls}$

★ Finally, one gets the following model (for each k^{th} articulation)



and

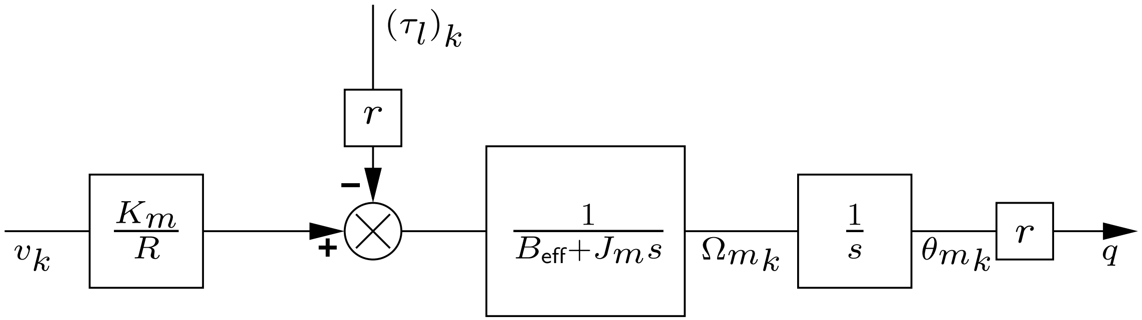
$$\frac{\Theta_m(s)}{V(s)} = \frac{K_m}{s(K_b K_m + (R + Ls)(B_m + J_m s))}$$

$$\frac{\Theta_m(s)}{\tau_l(s)} = \frac{-r(R + Ls)}{s(K_b K_m + (R + Ls)(B_m + J_m s))}$$

Note the "physical feedback" in the above model!

★ If $\frac{L}{R} << \frac{J_m}{B_m}$, then the above simplifies into (for each k^{th} articulation)

$$\Theta_m(s) = \frac{K_m}{R}\frac{1}{s(B_{\text{eff}} + J_m s)}V(s) - \frac{1}{s(B_{\text{eff}} + J_m s)}r\tau_l(s)$$



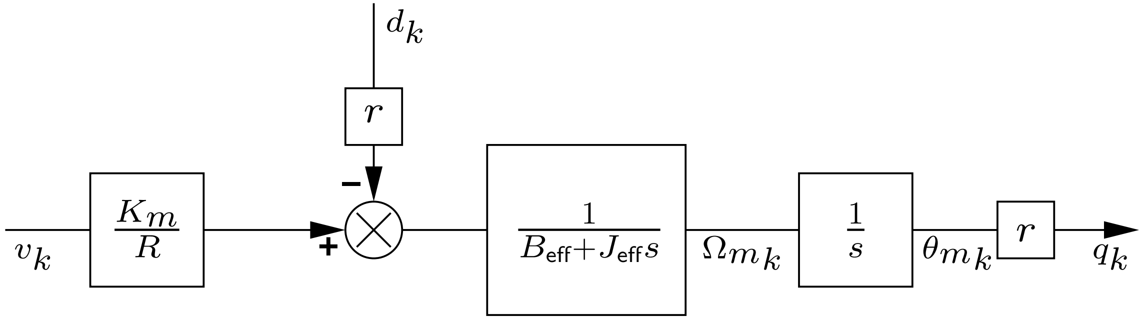
with

▷ $B_{\text{eff}} = B_m + \frac{K_b K_m}{R}$ effective friction coefficient

★ So, what are the difficulties at this stage, and what can be done in order to synthesize a "simple" feedback control law?

- as aforementioned, the generalized efforts $\left(r\tau_l\right)_k$ brought back on each k^{th} joint are coupled nonlinear functions of the robot configuration/velocities/accelerations $\mathbf{q}/\dot{\mathbf{q}}/\ddot{\mathbf{q}}$ (more on this later...)
- of course, $\left(r\tau_l\right)_k$ cannot be considered as mere perturbations
- one way around is, for each joint index k, to write $\left(\tau_l\right)_k$ as the sum of two contributions, one of which can (in first approximation) be considered as a true exogeneous signal, *e.g.* $\left(\tau_l\right)_k$ can be splitted into
  - ▷ the inertial efforts due to the body which immediately follows the articulation $\#k$: $\left(J_m\right)_k \mapsto$ "effective inertia" $\left(J_{\text{eff}}\right)_k$
  - ▷ the inertial efforts due to the other links, plus the other efforts (centrifugal, gravitational, Coriolis, etc.): $\left(\tau_l\right)_k \mapsto \left(d\right)_k$
- other ways to get a simplified open-loop model could be envisaged

★ Consequent schematic diagram:



★ More mathematical details will be given later - The important trick is that the control law can be designed on the basis of a simple schematic diagram, which enables the use of elementary control techniques for independent joint control

★ Important issues

- This model is valid when slow point-to-point motions are considered, for robots equipped with important gear reduction, without flexibility in the joints

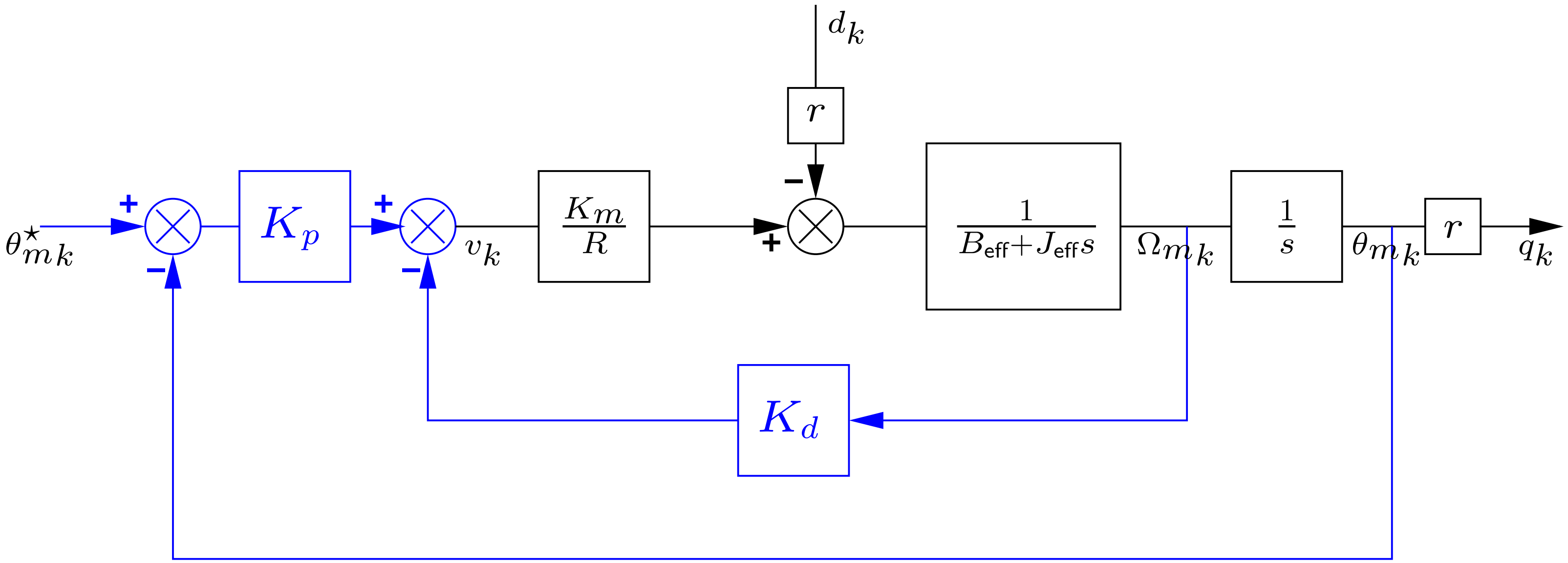
$\hookrightarrow$ but mind potential nonlinear effects! friction, backlash, etc.

- Strictly speaking, $(J_{\text{eff}})_k$ would depend on $\mathbf{q}$
  - ▷ to get a constant, $(J_{\text{eff}})_k$ is either set to an average value, or to a (pessimistic) worst-case value
- Of course, $(d)_k$ depends on $\mathbf{q}$
  - ▷ $(rd)_k$ is handled as a perturbation, known to be constant when the robot is at rest as it just amounts to gravitational effects

I.2.2 Proportional Derivative control

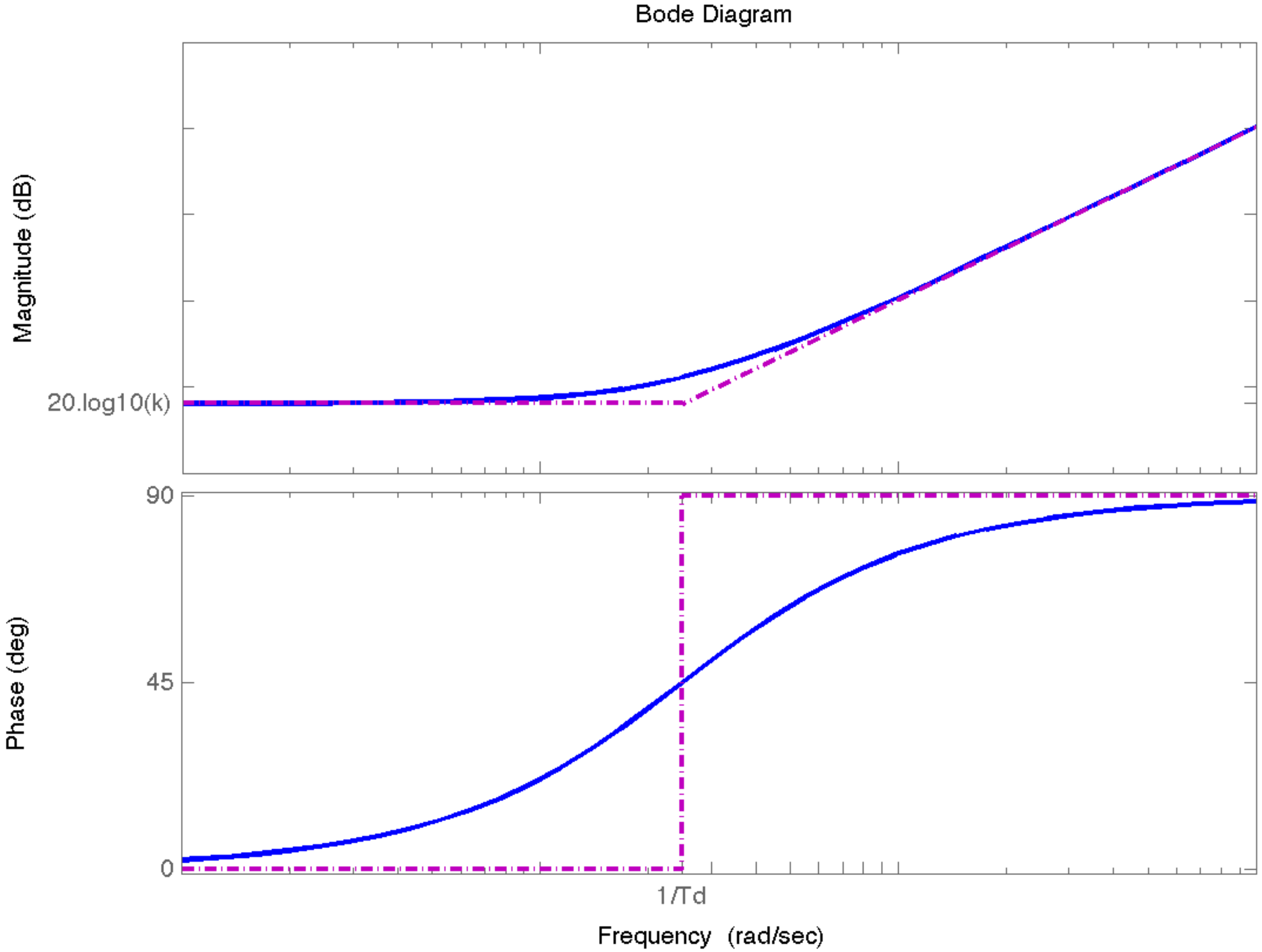
★ Control law for each k^{th} joint: $V(s) = K_p\big(\Theta^\star(s) - \Theta(s)\big) - K_d s\Theta(s)$

$$\Leftrightarrow v(t) = K_p\big(\theta^\star(t) - \theta(t)\big) - K_d\dot{\theta}(t)$$

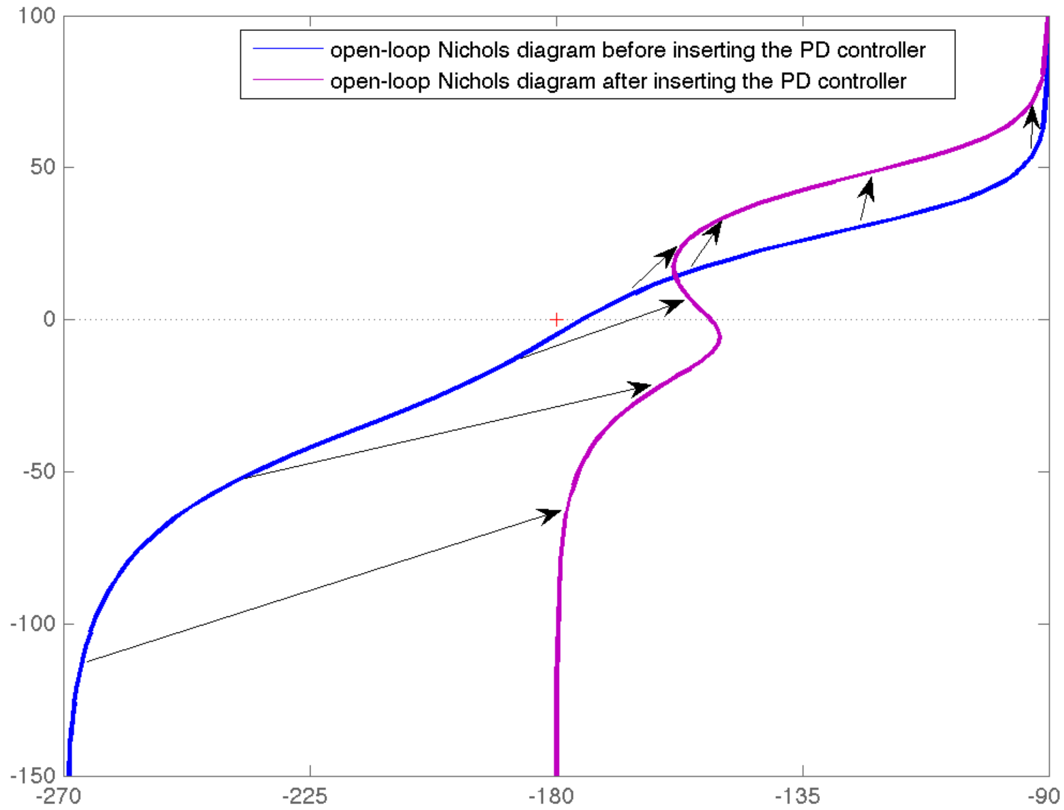


★ A short reminder on PD control and tachometer (velocity) feedback

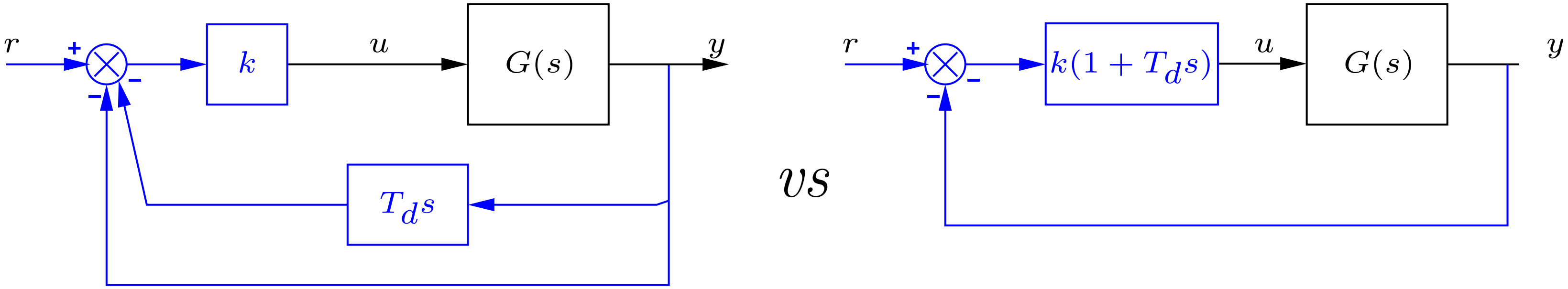
- Bode diagram of $D_{\text{PD}}(s) = k(1 + T_d s)$



- General guidelines to the selection of k, T_d
  - ▷ increase the stability margins of the feedback system, by adding a positive phase to the frequency response of the open-loop transfer function in the vicinity of the critical point



- Potential problems w.r.t. noise
  - ▷ PD $\mapsto$ phase-lead $D_{\text{PhLead}}(s) = \frac{D_{\text{PD}}(s)}{1+\tau s}$
- Why is tachometer (velocity) feedback prefered to PD control?



▷ a practical argument

- whatever the control scheme, the control signal u entails the derivative of the output y, with y a "smooth" signal
- in PD control, the derivative of the reference r (where r can be a step, for instance) is also involved in u, while this is not the case for tachometer feedback

▷ a theoretical explanation

- the two closed-loop transfer functions $F_{\text{TACHO}}(s)$ and $F_{\text{PD}}(s)$ are such that

$$F_{\text{PD}}(s) = F_{\text{TACHO}}(s)(1 + T_d s)$$

…and the adverse effects in the transient of y originate from the zero at $-\frac{1}{T_d}$ in the transfer $F_{\text{PD}}(s)$

★ Application to the robotics problem

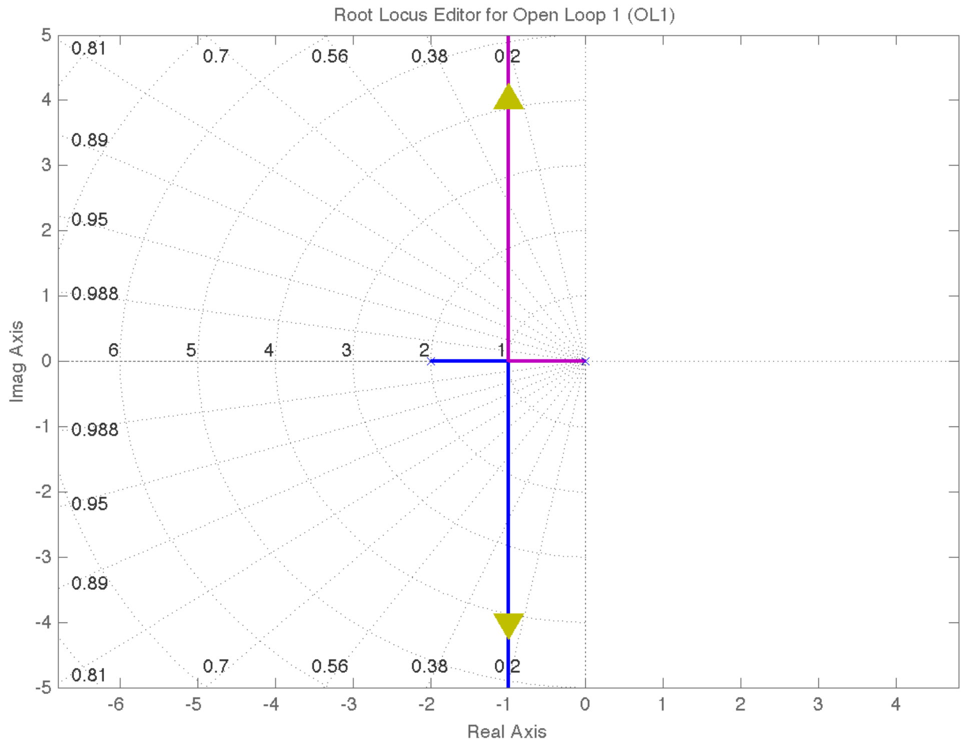
$$\Theta(s) = \frac{K_p \frac{K_m}{R}\,\Theta^\star(s)\ -\ r\,d(s)}{J_{\text{eff}} s^2 + (B_{\text{eff}} + K_d \frac{K_m}{R})s + K_p \frac{K_m}{R}}$$

- by the Routh criterion, the feedback system is stable if and only if the coefficients of the denominator of the above expression are all positive (the well-known necessary condition of stability is also sufficient here)
  - ▷ notice that this denominator is common to the transfers from any external input ($\theta^\star$ or d) to the controlled variable θ

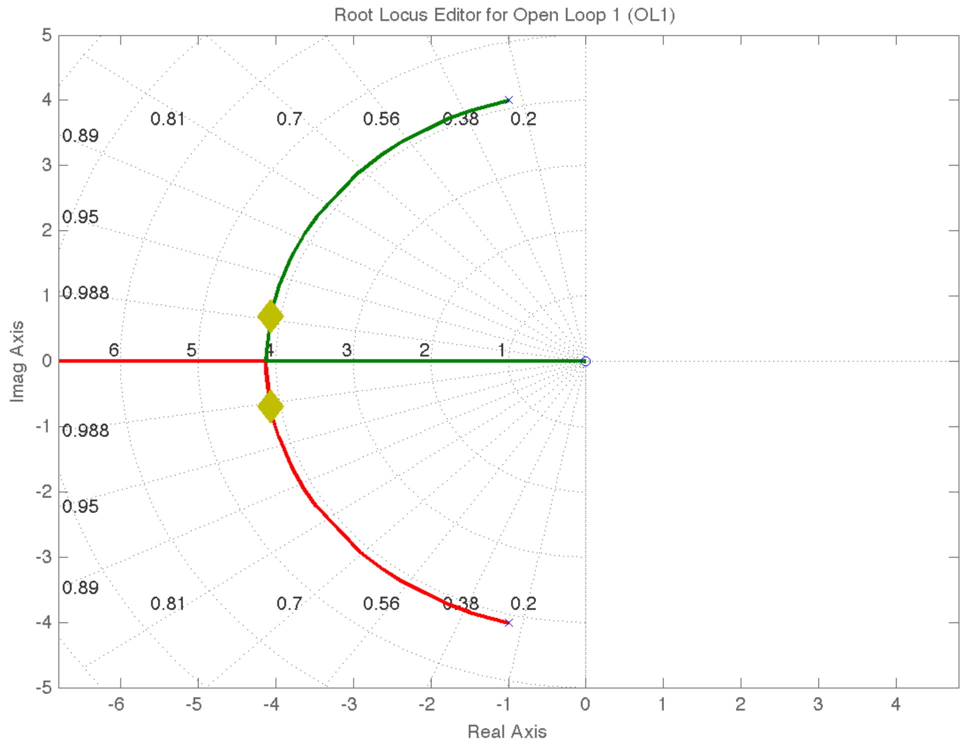
★ Tuning of the controller parameters K_p, K_d

- 2^{nd}-order characteristic polynomial – K_p, K_d are selected so that the feedback system is stable, with a unit damping ratio ($\zeta_{\text{closed-loop}} = 1$)
  - ▷ minimum settling time without overshoot

- qualitative effect of K_p, K_d on the closed-loop poles locus



effect of K_p for $K_d = 0$



effect of K_d for K_p fixed

★ Accuracy of the feedback system

- it can be easily shown that if d_k is a constant perturbation, then the position error—*i.e.* $\lim_{t \longrightarrow +\infty} \varepsilon(t)$ for a constant reference $\theta^\star(t) = \theta_0^\star$, with $\varepsilon(t) \triangleq \theta^\star(t) - \theta(t)$—reaches a constant steady state value ε_{pos} , which is all the smaller as K_p grows
  - ▷ prove that, for $d_k = d_0\Gamma(t)$,

$$\varepsilon_{\text{pos}} = \frac{r d_0}{K_p \frac{K_m}{R}}$$

  - ▷ in practice, though d_k is not a constant, the above indeed holds, for d_k amounts to gravitational effects in steady state
- similarly, the velocity error ε_{vel} for $\theta^\star(t) = \dot{\theta}_1 t \Gamma(t)$ and $d_k = d_0\Gamma(t)$ reads as

$$\varepsilon_{\text{vel}} = \frac{(K_d \frac{K_m}{R} + B_{\text{eff}})\dot{\theta}_1 + r d_0}{K_p \frac{K_m}{R}}$$

  - ▷ left as an exercise ☺

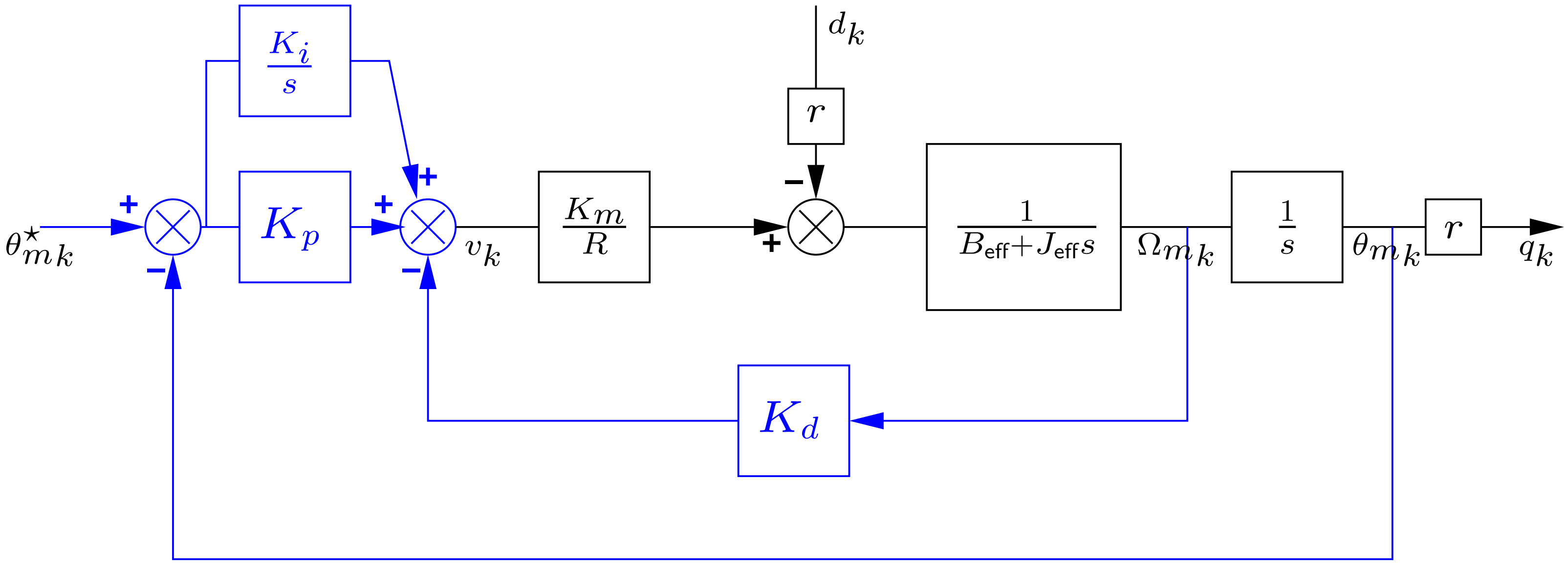
⋆ Limits

- actuators' saturations (mind the overshoot value!)
- this method can/must be extended in order to cope with flexibility in the drive train, if any

I.2.3 Proportional Integral Derivative control

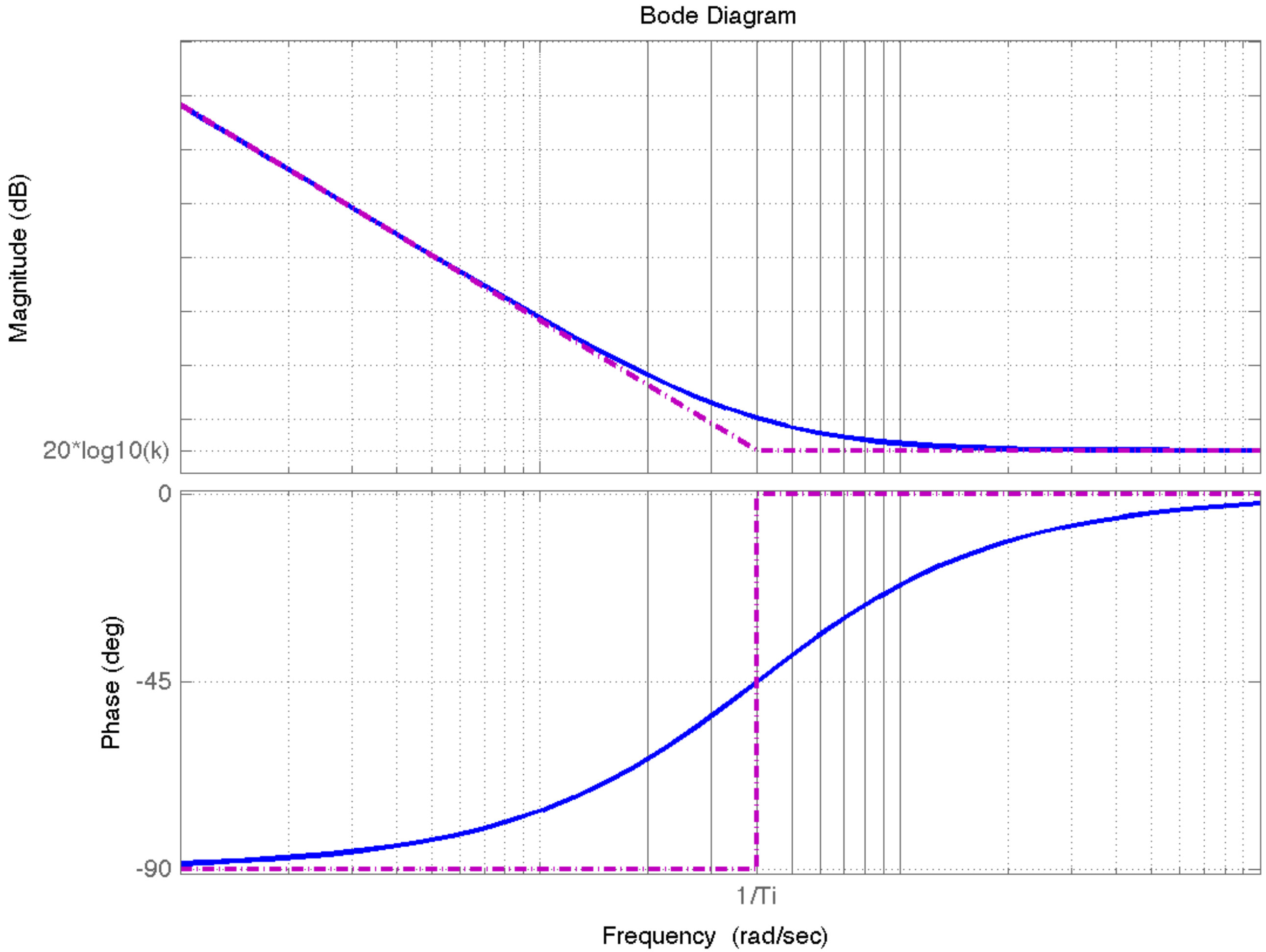
★ Control law for each k^{th} joint: $V(s) = K_p\big(\Theta^\star(s) - \Theta(s)\big) + \frac{K_i}{s}\big(\Theta^\star(s) - \Theta(s)\big) - K_d s\Theta(s)$

$$\Leftrightarrow v(t) = K_p\big(\theta^\star(t) - \theta(t)\big) + K_i\int_{-\infty}^{t}\big(\theta^\star(\tau) - \theta(\tau)\big)\mathrm{d}\tau - K_d\dot{\theta}(t)$$



★ A short reminder on Proportional Integral (PI) control

- Bode diagram of $D(s) = \frac{k}{T_i s}(1 + T_i s)$



- Effect and tuning of PI control
  - ▷ suppresses the "highest-type" nonzero finite steady-state error
  - ▷ $\frac{1}{T_i}$ must be sufficiently low so that the open-loop frequency response is not modified in the vicinity of the critical point
  - ▷ a famous tuning method: the symmetric optimum rule
  - ▷ mind the actuators' saturations! note that there exists anti-windup systems to prevent the induced instability

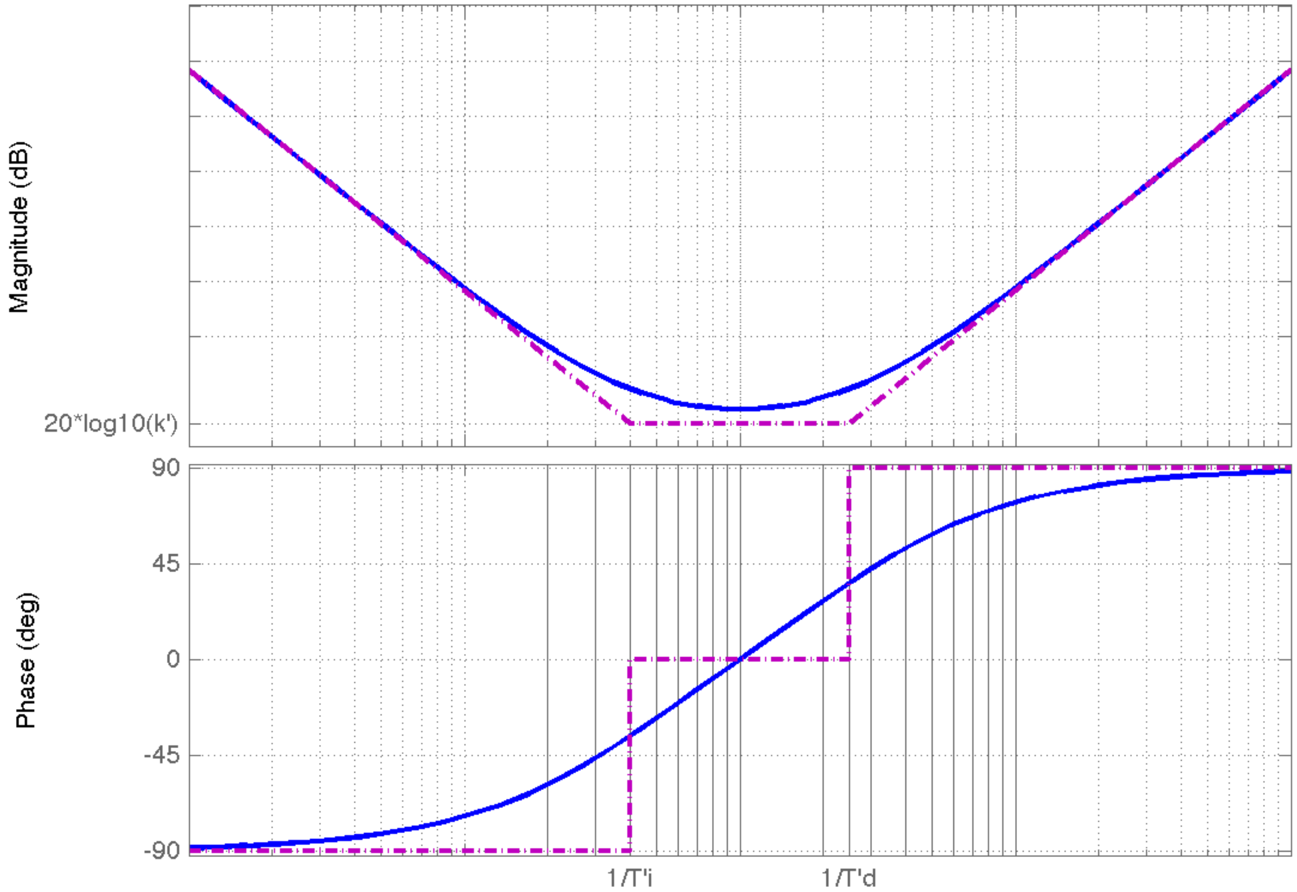
★ Proportional Integral Derivative (PID) control

- Most industrial regulators, due to relative simplicity, expert knowledge, obtained robustness
  - ▷ including in robotics servo units
- Three basic PID configurations
  - ▷ standard: $D(s) = k\big(1 + \frac{1}{T_i s} + T_d s\big)$
  - ▷ parallel: $D(s) = K_p + \frac{K_i}{s} + K_d s$
  - ▷ series: $D(s) = \frac{k'}{T_i' s}(1 + T_i' s)(1 + T_d' s)$

$\hookrightarrow$ Remind they are unequivalent, as the series form assumes real zeros

- Bode diagram of a series form

. . . / . . .



▷ mind potential errors in manual plots of the Bode diagram of the standard and parallel PID configurations!

★ Tuning methods

- analytical, frequency-based, empirical
- benefits and drawbacks of PD & PI (cf. anti-windup systems. . .)
- see also *PID controllers: Theory, Design and Tuning* - K.J. Åström, T. Hägglund Instrument Society of America, 1995

★ Application to independent joint space control in robotics

- as soon as the feedback system is stable, the steady-state position error is eliminated
- note that the PD action is generally implemented as a velocity feedback

I.2.4 Control using state-space techniques

★ State feedback

★ Output feedback, *e.g.* feedback on an observer-based estimation of the state vector

★ Optimal control, *e.g.* quadratic (energetic) criterion

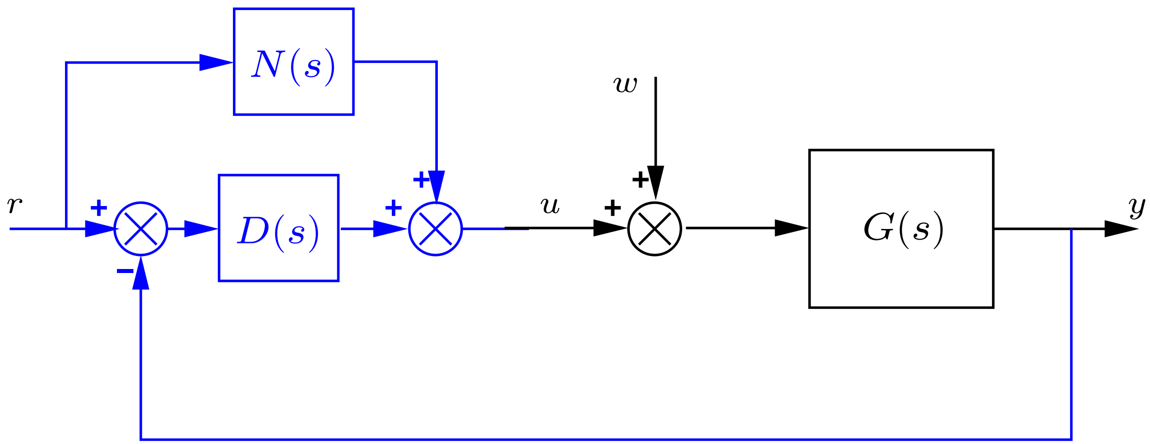
I.3 FEEDFORWARD CONTROL

I.3.1 Basics

★ Aims

- error-free tracking of time-varying reference inputs
- rejection of time-varying perturbations

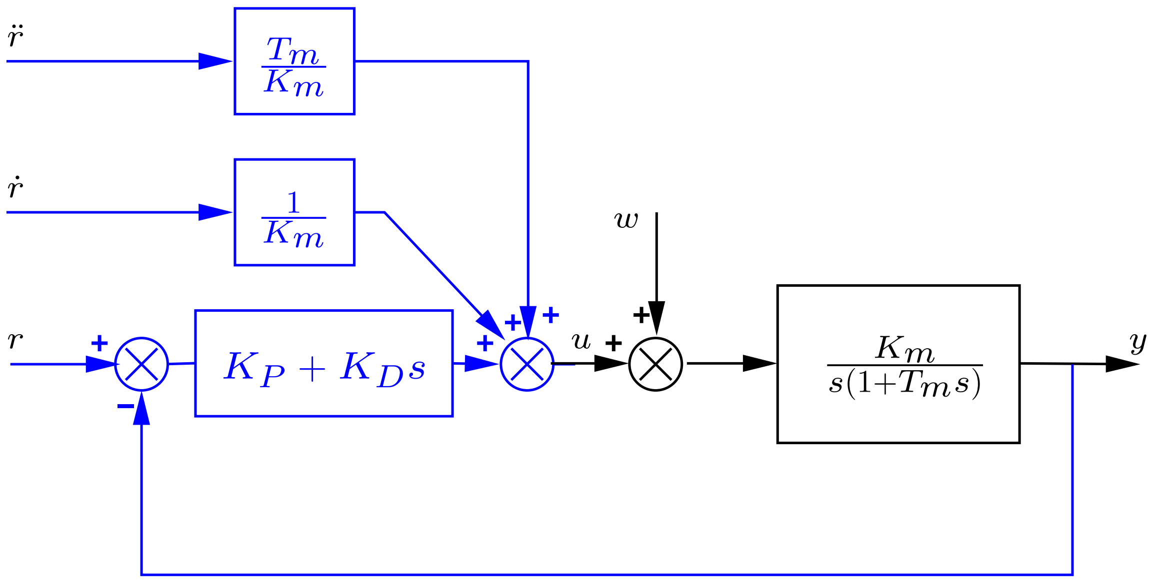
★ An introductory example



- $Y(s) = \dfrac{\big(N(s)G(s)+D(s)G(s)\big)\,R(s)+G(s)W(s)}{1+D(s)G(s)}$

★ When $w = 0$, if $N(s)G(s) = 1$, then $Y(s) = R(s)$, and the tracking is perfect !

- The controller adds in open-loop—via $N(s)$—what "was missing" in order to perform a perfect tracking
- Constraints
  - ▷ $N(s)$ must be stable ⇒ $G(s)$ must be minimum-phase as $N(s) = G(s)^{-1}$
  - ▷ How can $N(s)$ be made causal?
    - · by deriving the reference input $r(t)$ as many times as necessary!
- Example: PD control with feedforward



I.3.2 The Computed Torque Method: perturbation compensation by feedforward

★ Elements of Dynamic Modeling of rigid robots manipulators

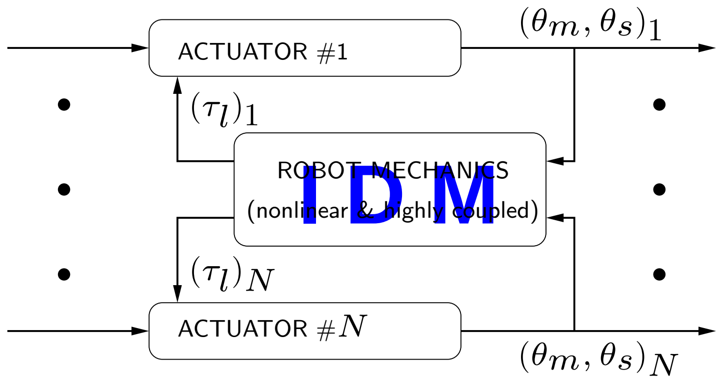
$$\underline{\mathbf{D}}(\mathbf{q})\ddot{\mathbf{q}} + \underline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau$$

with

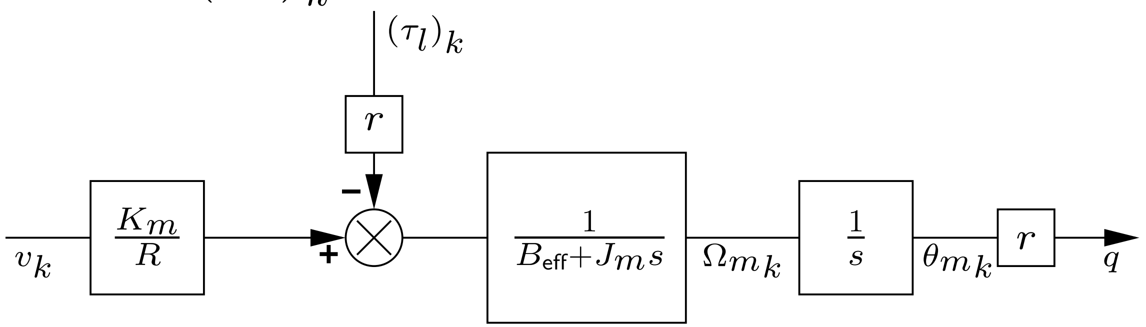
- ▷ $\underline{\mathbf{D}}(\mathbf{q})$ the inertia matrix of the robot
- ▷ $\underline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$ the matrix of centrifugal and Coriolis efforts
- ▷ $\mathbf{g}(\mathbf{q})$ the vector related to gravitation
- ▷ τ the vector of generalized efforts applied on the robot' joints

- Direct and Inverse forms of the Dynamic Model
  - ▷ DDM: from the knowledge of $\tau(.)$, infer the time evolution of $\mathbf{q}(.)$
  - ▷ IDM: from the knowledge of $\mathbf{q}(.), \dot{\mathbf{q}}(.), \ddot{\mathbf{q}}(.)$, infer τ
- Note: possibly hard to determine experimentally, for a robot with non-polyhedral or non-ellipsoidal links. . .

★ Flashback: a while ago, this model was introduced. . .

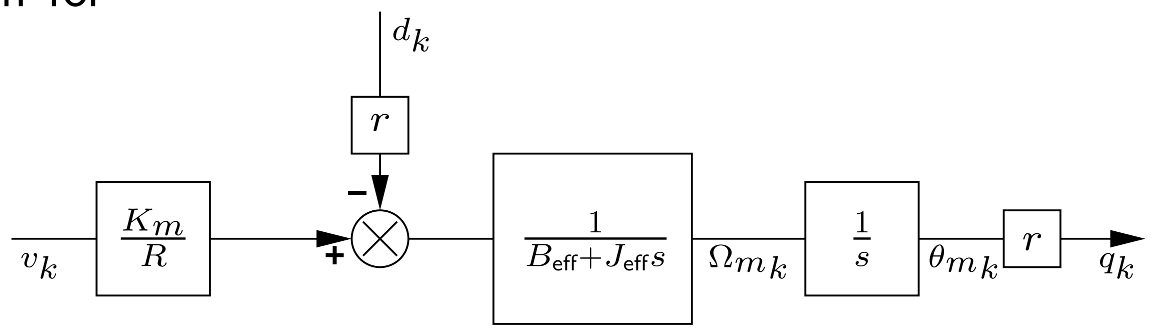


- For each k^{th} articulation, the following schematic diagram—where $r\tau_k$ stands for the generalized effort $\left(r\tau_l\right)_k$ brought back on the considered joint—



$(*)$

was traded off for



$(**)$

- The data of the Inverse Dynamic Model of the robot enables us to express J_{eff} and d_k for each k^{th} axis, by combining the two following equations:

$$\tau_k = d_{kk}(\mathbf{q})\ddot{q}_k + \underbrace{\sum_{j\neq k} d_{kj}(\mathbf{q})\ddot{q}_j + \sum_l c_{kl}(\mathbf{q},\dot{\mathbf{q}})\dot{q}_l + g_k(\mathbf{q})}_{\text{henceforth denoted by } d_k(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})}$$

$$J_m\ddot{\theta}_{m_k} + B_{\text{eff}}\dot{\theta}_{m_k} = \frac{K_m}{R}v_k - r\tau_k$$

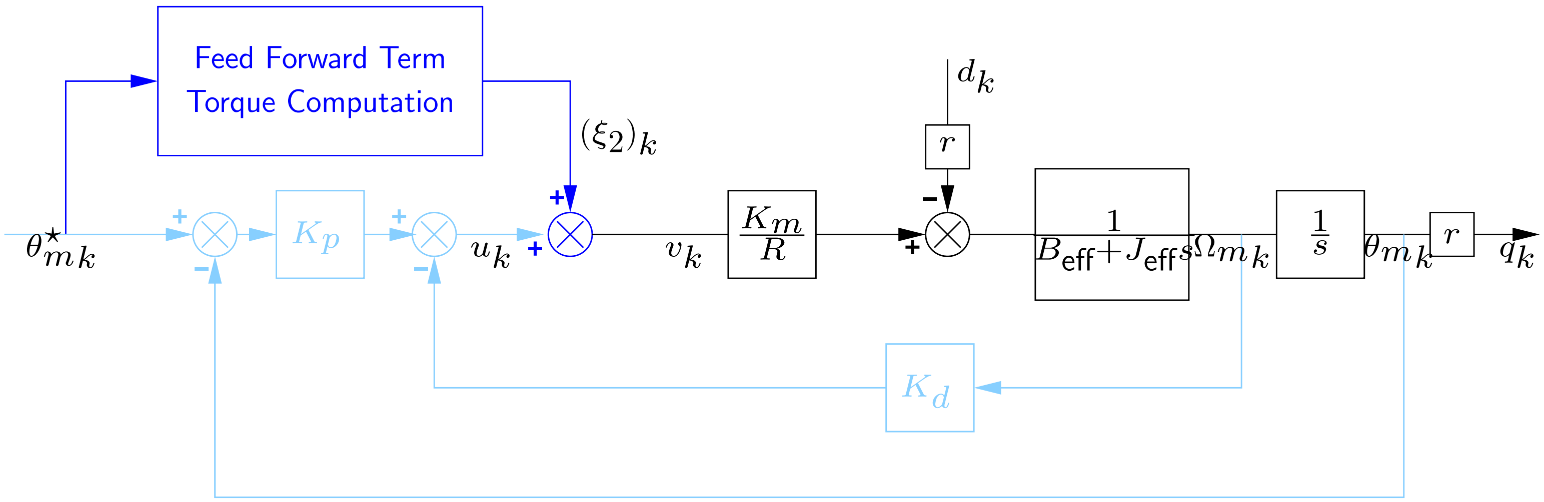
- After some computations, one gets

$$\big(J_m + r^2 d_{kk}(\mathbf{q})\big)\ddot{\theta}_{m_k} + B_{\text{eff}}\dot{\theta}_{m_k} = \frac{K_m}{R}v_k - r d_k(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$$

so that $(*)$ can be turned into $(**)$ as soon as J_{eff} is set to a mean or worst-case value of $J_m + r^2 d_{kk}(\mathbf{q})$

★ The Computed Torque Method

- The basic idea is to superimpose on the output u_k from a classical feedback controller—*e.g.* PD, PID,...—dedicated to each k^{th} axis, a signal $\xi_k = (\xi_1)_k + (\xi_2)_k$ computed in open-loop where
  - ▷ $(\xi_1)_k$ would contribute to a perfect tracking if d_k were set to 0, following the guidelines in §I.3.1
  - ▷ $(\xi_2)_k$ nearly cancels the effect of $rd_k(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$



- Let us focus on $(\xi_2)_k$ (because $(\xi_1)_k$ can be defined very easily). It can be set to

$$(\xi_2)_k = \frac{1}{\frac{K_m}{R}} r\, d_k(\mathbf{q}^\star, \dot{\mathbf{q}}^\star, \ddot{\mathbf{q}}^\star)$$

$$= \frac{1}{\frac{K_m}{R}} r \left\{ \sum_{j \neq k} d_{kj}(\mathbf{q}^\star)\ddot{q}_j^\star + \sum_l c_{kl}(\mathbf{q}^\star, \dot{\mathbf{q}}^\star)\dot{q}_l^\star + g_k(\mathbf{q}^\star) \right\}$$

  ▷ the reason of this definition of $(\xi_2)_k$ comes from the fact that $q_k = r\theta_{mk}$ is assumed to be close to $q_k^\star = r\theta_{mk}^\star$ during the task...

★ Pros, Cons, and Induced requirements

⊕ the stability of the whole feedback system stays unmodified

⊖ of course, $rd_k(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \neq \frac{K_m}{R}\xi_{2k}\ldots$, yet this leads to nice results

- Mind real-time constraints!
  ▷ an important research effort used to focus on the recursive computation of the dynamic model (Newton-Euler)
  ▷ $\mathbf{q}^\star, \dot{\mathbf{q}}^\star, \ddot{\mathbf{q}}^\star$ must be computed offline, and saved into a look-up-table

I.4 MULTIVARIABLE APPROACHES

★ Introduction

- Use of modern Automatic Control theories/techniques
  - ▷ powerful conclusions
  - ▷ sound theoretical bases
- No restrictive assumption, *e.g.* concerning the use of gear train

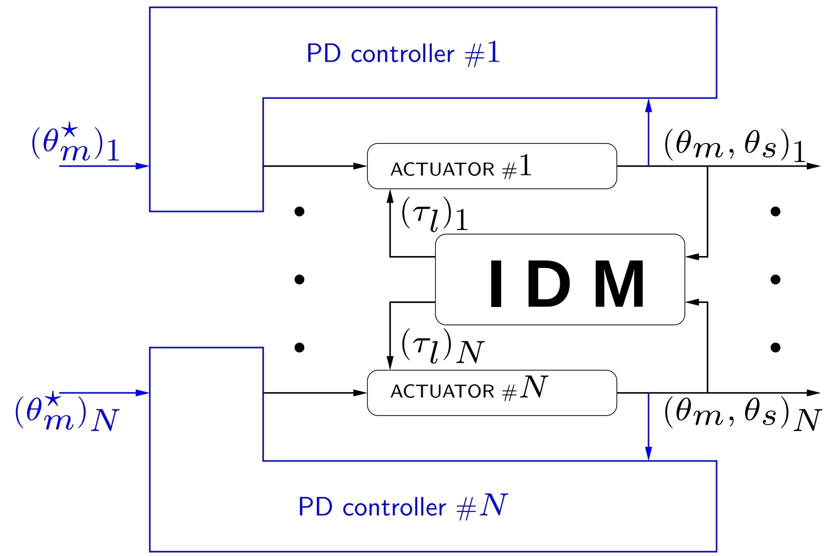
$\hookrightarrow$ possibility to consider mechanical structures enabling very high performances

I.4.1 PD control

★ A while ago, the analysis and synthesis of a PD control were studied, assuming that the robot could be handled through a very simplified model, in which some efforts brought back on the joint axes could be captured via "efficient inertias" $(J_{\text{eff}})_k$. Nevertheless, the following question remains:

.../...

- What happens if a PD controller is connected to the genuine nonlinear model of the whole robot with its actuators?



★ It can be shown that

- the PD controller does stabilize the nonlinear feedback model, but $\varepsilon(t) \triangleq \theta^\star(t) - \theta(t)$ reaches a constant steady state value ε_{pos}, which is all the smaller as K_p grows
- ε_{pos} can be made lower by inserting a feedback term function of $\mathbf{q}$ which compensates for gravitational effects

I.4.2 Feedback Linearization

★ Torque control of a robot manipulator is considered

- Robot modeling by the Direct Dynamic Model

$$\underline{\mathbf{D}}(\mathbf{q})\ddot{\mathbf{q}} + \underline{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau$$

- Strategy sometimes called "Computed Torque Method"

★ Feedback controller based on two nested loops

- Inner loop: nonlinear controller which decouples and linearizes the robot dynamics
  - ▷ $\tau = \widehat{\underline{\mathbf{D}}(\mathbf{q})}v + \widehat{\underline{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})} + \widehat{\mathbf{g}(\mathbf{q})}$
  - ▷ the inner closed-loop system becomes a set of N parallel independent double integrators, described by $\ddot{q}_k = v_k$
- Outer loop: a set of non-interactive classical linear controllers, whose role is to stabilize the above double integrators
  - ▷ these controllers are theoretically fully independent of the robot dynamics

★ Consequent schematic diagram:

★ Pros and Cons

- ⊕ Very attractive strategy!
- ⊕ Decoupling and linearization
- ⊖ Requires an excellent model of the robot
- ⊖ Requires a powerful hardware (because of many nonlinear computations)

★ Pros and Cons (cont'd)

- ⊖ The robot's DDM need to be simplified in order to synthesize the inner controller $\Rightarrow$ some important properties of the feedback system may be lost, *e.g.* stability, decoupling, etc.
  - ▷ a solution may consist in using linear robust control techniques to design the outer controller, the difference between the actual and ideal inner feedback system being embedded into an uncertainty which worst-case realization is considered

I.4.3 Variable Structure Nonlinear Control

★ Nonlinear control which

(1) drives the state vector of the feedback system onto a manifold of the state space (sliding manifold/surface)

(1) once this manifold is reached, the closed-loop system dynamics are governed by state space equations issued from the definition of this manifold

★ Pros and Cons

⊕ High robustness to model parameter uncertainties

⊕ Possibly high performances

⊖ But...

- ▷ how can the sliding surface be defined?
- ▷ how to ensure that the sliding surface is reached?
- ▷ high-frequency switching control (chattering)

I.4.4 Adaptive Control

⋆ Online modification of the control law so as to cope with time variations of the system parameters, *e.g.*

- Gain scheduling
- Adjustment of the control parameters so that the closed loop performance obeys a reference model
- Adjustment of the control parameters based on an identification of the system while it is running
- etc.

Chapter II

CARTESIAN-SPACE CONTROL

II.1 OVERVIEW & POTENTIAL PROBLEMS

⋆ The reference signal is now expressed by a time vector function $\mathbf{x}^{\star}(.)$.

⋆ Some important potential problems relative to the definition of reference (position,attitude) motions (cartesian trajectories)

- Even if $\mathbf{x}^{\star}(\text{BEGIN})$ and $\mathbf{x}^{\star}(\text{END})$ are located within the robot's workspace, a "simple" cartesian trajectory between them may not be achievable
  - ▷ example. . .

★ Some important potential problems (cont'd)

- If, when the robot is at configuration $\mathbf{q}$, its Jacobian $J(\mathbf{q})$ is singular, then for given $\dot{\mathbf{x}}$, there may be no $\dot{\mathbf{q}}$ solution to $\dot{\mathbf{x}} = J(\mathbf{q})\dot{\mathbf{q}}$
  - ▷ Crossing differential singularities may require very high velocities at the joint level. Due to its actuators' saturations, the robot may deviate from this reference behavior.

- $\mathbf{x}^{\star}$(BEGIN) and $\mathbf{x}^{\star}$(END) cannot be reached by the same solution of the robot's Inverse Geometrical Model, and there is no possibility of reconfiguration between adjacent aspects.

II.2 SOME PROMINENT CARTESIAN CONTROL SCHEMES

II.2.1 An Elementary Controller

⋆ Problem: in open-loop w.r.t. $\mathbf{x}$!

- Fair robustness to uncertainties in the robot's geometric/differential models
- Lack of genericity in the computation of the robot's Inverse Geometric Model

II.2.2 Elementary Feedback Cartesian Control Schemes

★ Inverse Jacobian Cartesian Scheme

★ Transpose Jacobian Cartesian Scheme

⋆ Comments

- Performances analysis may be tricky
- Cost of the Jacobian computation in the IJCS
- The dynamics still depends on the configuration of the robot!

II.2.3 Feedback linearization in Cartesian space

★ Inverse Dynamic Model on $\mathbf{x}$

$$\underline{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{x}} + \underline{\mathbf{V}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathcal{F}$$

with $\mathcal{F}$ fictitious effort, such that $\tau = J^T(\mathbf{q})\mathcal{F}$

★ Two nested loops feedback controller

- Implementation at several rates
  - ▷ low rate for the computation of $\widehat{\underline{\mathbf{M}}(\mathbf{q})}$, $\widehat{\underline{\mathbf{V}}(\mathbf{q}, \dot{\mathbf{q}})}$, $\widehat{\mathbf{G}(\mathbf{q})}$
  - ▷ higher rate for the computation of the controller output