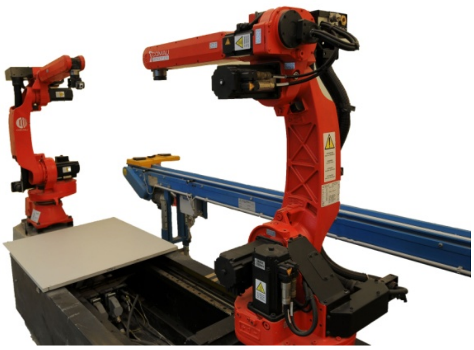
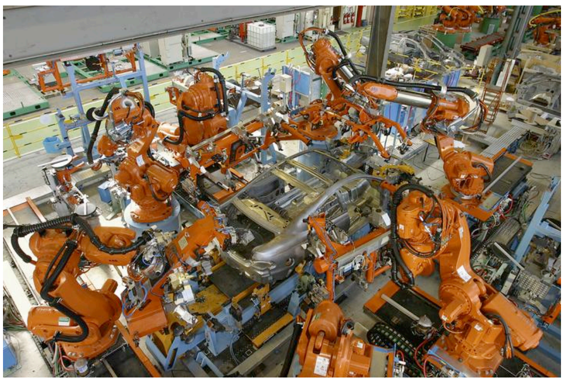
ROBOT MANIPOLATORI

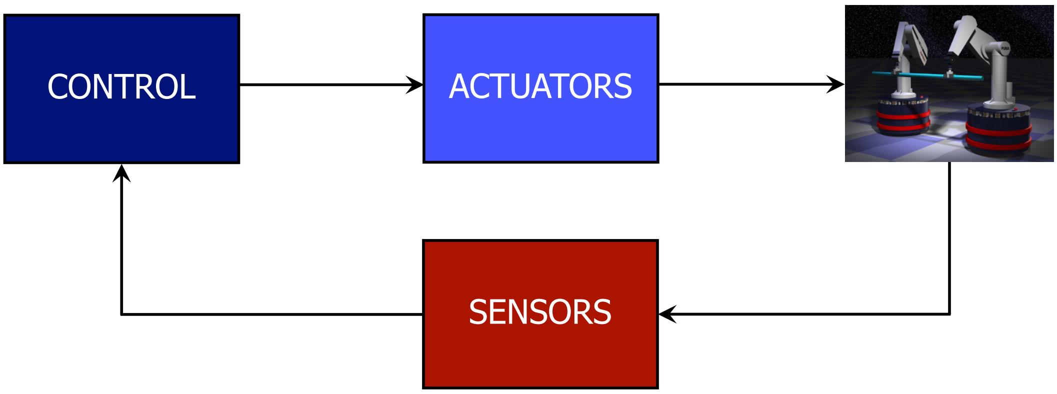
Robot Industriali





Robot di Servizio

Controllo



- Sensori
  - ★ Posizione => Controllo del moto
  - ★ Forza => Controllo di forza
  - ★ Visione => Controllo visuale

Modelling and Motion Control

Luigi VILLANI

PRISMA Lab
Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
luigi.villani@unina.it
www.prisma.unina.it

Introduction

- Modelling
  - Kinematics
  - Dynamics
- Control
  - Joint space control vs. task space control
  - Regulation
  - Tracking

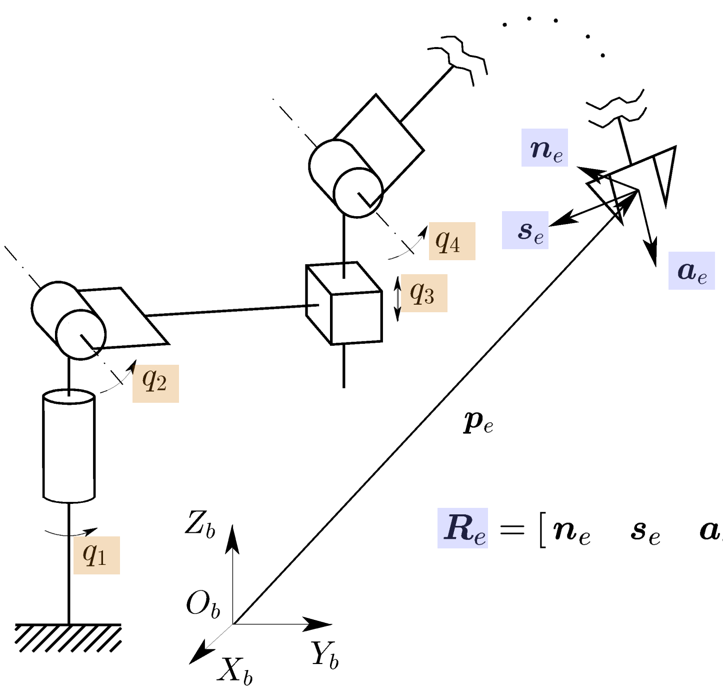
- Kinematic model

$$\boldsymbol{p}_e = \boldsymbol{p}_e(\boldsymbol{q})$$

$$\boldsymbol{R}_e = \boldsymbol{R}_e(\boldsymbol{q})$$

$$\downarrow$$

$$\boldsymbol{T}_e = \begin{bmatrix} \boldsymbol{R}_e & \boldsymbol{p}_e \\ \boldsymbol{0}^T & 1 \end{bmatrix} = \boldsymbol{T}_e(\boldsymbol{q})$$



$$\boldsymbol{R}_e = [\,\boldsymbol{n}_e \quad \boldsymbol{s}_e \quad \boldsymbol{a}_e\,]$$

- Differential kinematics

$$\begin{bmatrix} \dot{\boldsymbol{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \boldsymbol{v}_e = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}_p \\ \boldsymbol{J}_o \end{bmatrix} \dot{\boldsymbol{q}}$$

$\boldsymbol{J}$: $(6 \times n)$ matrix
geometric Jacobian

$$\dot{\boldsymbol{R}}_e = \boldsymbol{S}(\boldsymbol{\omega}_e)\boldsymbol{R}_e$$

- Kineto-static duality (principle of virtual work)

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q}) \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\mu} \end{bmatrix} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}$$

Task Space

- Task space $(m \leq 6)$ $\quad n > m$: kinematic redundacy

$$\boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{p}_e \\ \boldsymbol{\phi}_e \end{bmatrix} = \boldsymbol{k}(\boldsymbol{q})$$

$\boldsymbol{\phi}_e :\ (3 \times 1)$ Euler angles extracted from $\boldsymbol{R}_e$

- Differential kinematics

$$\dot{\boldsymbol{x}}_e = \boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}$$

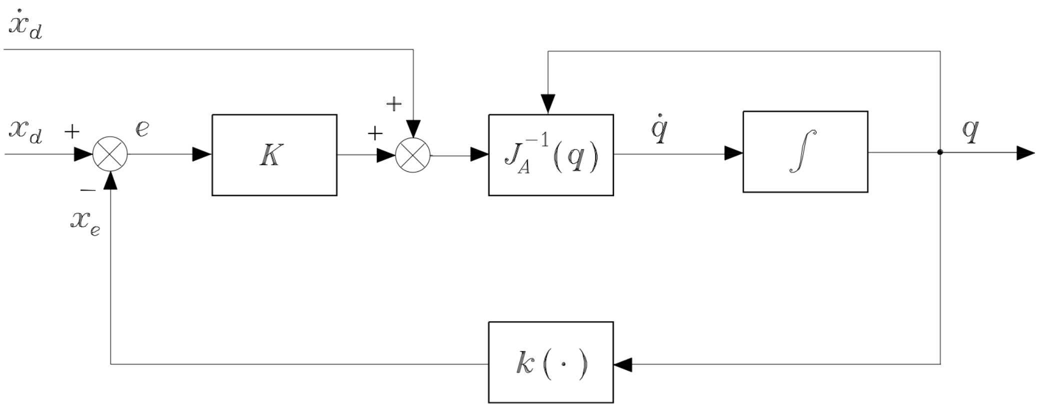
$$\boldsymbol{J}_A(\boldsymbol{q}) = \frac{\partial \boldsymbol{k}(\boldsymbol{q})}{\partial \boldsymbol{q}} :$$ $(m \times n)$ matrix

analitical Jacobian

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{T}(\boldsymbol{\phi}_e) \end{bmatrix} \boldsymbol{J}_A \qquad \boldsymbol{\omega}_e = \boldsymbol{T}(\boldsymbol{\phi}_e)\dot{\boldsymbol{\phi}}_e$$

Inverse Kinematics

- Closed Loop Inverse Kinematics (CLIK)



$$\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}_e + \boldsymbol{K}(\boldsymbol{x}_d - \boldsymbol{x}_e) = \boldsymbol{0}$$

Scuola di dottorato SIDRA
Bertinoro 2010

Bertinoro (FC)
12 Luglio 2010

- Lagrange formulation

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \qquad \xi_i = \tau_i - \tau_{fi} - \tau_{ei}$$

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \qquad \mathcal{T} = \frac{1}{2}\dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} \qquad g_i(\boldsymbol{q}) = \frac{\partial \mathcal{U}}{\partial q_i}$$

- Dynamic model

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{h} \qquad \boldsymbol{h} = \begin{bmatrix}\boldsymbol{f}\\ \boldsymbol{\mu}\end{bmatrix}$$

- Skew-symmetry of $\dot{\boldsymbol{B}} - 2\boldsymbol{C}$

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \dot{\boldsymbol{B}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}(\dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}})\right)^{\mathrm{T}}$$

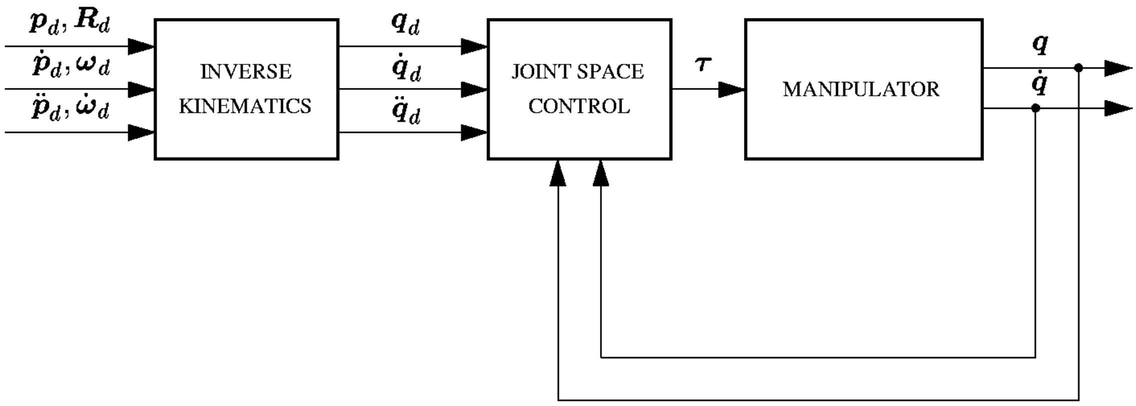
$$C_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_j\dot{q}_k \qquad c_{ijk} = \frac{1}{2}\left(\frac{\partial B_{ij}}{\partial q_k} + \frac{\partial B_{ik}}{\partial q_j} - \frac{\partial B_{jk}}{\partial q_i}\right)$$

  - Hamilton principle $\Longrightarrow$ $\dot{\boldsymbol{q}}^{\mathrm{T}}\left(\dot{\boldsymbol{B}}(\boldsymbol{q},\dot{\boldsymbol{q}}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right)\dot{\boldsymbol{q}} = \boldsymbol{0} \qquad \forall \boldsymbol{C}$

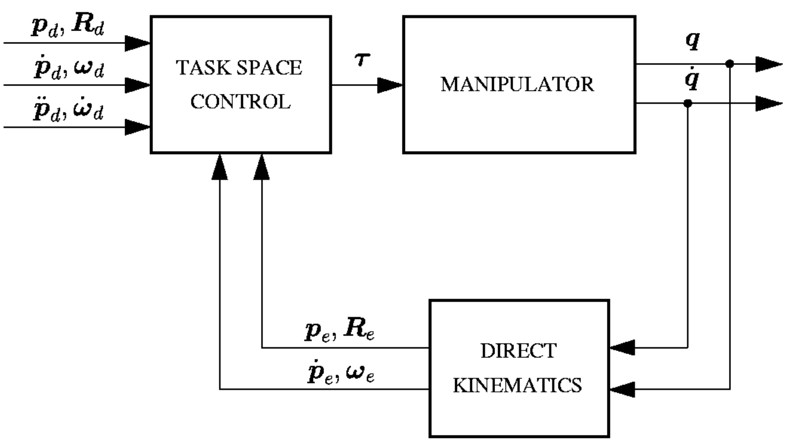
- Linearity in the dynamic parameters

$$\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\pi} = \boldsymbol{\tau} - \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{h}$$

- Joint space control
  - Task references transformed into joint references
  - Redundancy resolution at kinematic level



- Task space control
  - Control directly in task (operational) space
  - Redundancy resolution at dynamic level



Task Space Control

- Regulation
  - Static model-based compensation
  - Orientation errors
- Tracking
  - Dynamic model-based compensation
  - Redundancy resolution

Static Model-Based Compensation

- PD control with gravity compensation

$$\boldsymbol{\tau} = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q}) \begin{bmatrix} \gamma_p \\ \gamma_o \end{bmatrix} - \boldsymbol{K}_D \dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

  - Position control

$$\gamma_p = \boldsymbol{K}_{Pp} \Delta \boldsymbol{p}_{de}$$

  - Orientation control

$$\gamma_o \Longrightarrow$$

Euler angles
Angle/axis
Quaternion

Orientation Errors

- Euler angles

$$\gamma_o = \boldsymbol{T}^{-\mathrm{T}}(\boldsymbol{\varphi}_e)\boldsymbol{K}_{Po}\Delta\boldsymbol{\varphi}_{de} \qquad \Delta\boldsymbol{\varphi}_{de} = \boldsymbol{\varphi}_d - \boldsymbol{\varphi}_e$$

- Alternative Euler angles

$$\gamma_o = \boldsymbol{T}_e^{-\mathrm{T}}(\boldsymbol{\varphi}_{de})\boldsymbol{K}_{Po}\boldsymbol{\varphi}_{de} \qquad \boldsymbol{R}_d^e = \boldsymbol{R}_e^T\boldsymbol{R}_d \Longrightarrow \boldsymbol{\varphi}_{de}$$

- Angle/axis

$$\gamma_o = \boldsymbol{K}_{Po}\boldsymbol{o}'_{de}$$

- Quaternion

$$\gamma_o = \boldsymbol{K}_{P_o}\boldsymbol{R}_e\boldsymbol{\epsilon}_{de}^e$$

  - For all ... stability via Lyapunov arguments $\mathcal{V} = \frac{1}{2}\dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \mathcal{U}_p + \mathcal{U}_o$

Angle/Axis

- Orientation error: $\boldsymbol{R}_d^e = \boldsymbol{R}_e^T\boldsymbol{R}_d \Longrightarrow \boldsymbol{o}_{de}^e = f(\vartheta_{de})\boldsymbol{r}_{de}^e$

Representation	$f(\vartheta)$
Classical angle/axis	$\sin(\vartheta)$
Quaternion	$\sin(\vartheta/2)$
Rodrigues parameters	$\tan(\vartheta/2)$
Simple rotation	ϑ

angle axis

- Inverse dynamics

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{\alpha} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

$$\dot{\boldsymbol{v}}_e = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$$

$$\boldsymbol{\alpha} = \boldsymbol{J}^{-1}(\boldsymbol{q})\left(\begin{bmatrix}\boldsymbol{a}_p\\ \boldsymbol{a}_o\end{bmatrix} - \dot{\boldsymbol{J}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right)$$

  - Position control $\Delta\boldsymbol{p}_{de} = \boldsymbol{p}_d - \boldsymbol{p}_e$

$$\boldsymbol{a}_p = \ddot{\boldsymbol{p}}_d + \boldsymbol{K}_{Dp}\Delta\dot{\boldsymbol{p}}_{de} + \boldsymbol{K}_{Pp}\Delta\boldsymbol{p}_{de} \implies \Delta\ddot{\boldsymbol{p}}_{de} + \boldsymbol{K}_{Dp}\Delta\dot{\boldsymbol{p}}_{de} + \boldsymbol{K}_{Pp}\Delta\boldsymbol{p}_{de} = \boldsymbol{0}$$

  - Orientation control

$\boldsymbol{a}_o$ $\Rightarrow$ Euler angles
Angle/axis
Quaternion

- Inverse dynamics

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{\alpha} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

$$\dot{\boldsymbol{v}}_e = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$$

$$\boldsymbol{\alpha} = \boldsymbol{J}^{-1}(\boldsymbol{q})\left(\begin{bmatrix}\boldsymbol{a}_p\\ \boldsymbol{a}_o\end{bmatrix} - \dot{\boldsymbol{J}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right)$$

  - Position control $\Delta\boldsymbol{p}_{de} = \boldsymbol{p}_d - \boldsymbol{p}_e$

$$\boldsymbol{a}_p = \ddot{\boldsymbol{p}}_d + \boldsymbol{K}_{Dp}\Delta\dot{\boldsymbol{p}}_{de} + \boldsymbol{K}_{Pp}\Delta\boldsymbol{p}_{de} \implies \Delta\ddot{\boldsymbol{p}}_{de} + \boldsymbol{K}_{Dp}\Delta\dot{\boldsymbol{p}}_{de} + \boldsymbol{K}_{Pp}\Delta\boldsymbol{p}_{de} = \boldsymbol{0}$$

  - Orientation control

$\boldsymbol{a}_o$ $\Rightarrow$ Euler angles
Angle/axis
Quaternion

Scuola di dottorato SIDRA
Bertinoro 2010

Bertinoro (FC)
12 Luglio 2010

Euler Angles

- Orientation error: $\Delta\boldsymbol{\varphi}_{de} = \boldsymbol{\varphi}_d - \boldsymbol{\varphi}_e$
  - Resolved angular acceleration

$$\boldsymbol{a}_o = \boldsymbol{T}(\boldsymbol{\varphi}_e)(\ddot{\boldsymbol{\varphi}}_d + \boldsymbol{K}_{Do}\Delta\dot{\boldsymbol{\varphi}}_{de} + \boldsymbol{K}_{Po}\Delta\boldsymbol{\varphi}_{de}) + \dot{\boldsymbol{T}}(\boldsymbol{\varphi}_e, \dot{\boldsymbol{\varphi}}_e)\dot{\boldsymbol{\varphi}}_e$$

representation singularities (!) $\boldsymbol{\omega}_e = \boldsymbol{T}(\boldsymbol{\varphi}_e)\dot{\boldsymbol{\varphi}}_e$

  - Error dynamics

$$\dot{\boldsymbol{\omega}}_e = \boldsymbol{T}(\boldsymbol{\varphi}_e)\ddot{\boldsymbol{\varphi}}_e + \dot{\boldsymbol{T}}(\boldsymbol{\varphi}_e, \dot{\boldsymbol{\varphi}}_e)\dot{\boldsymbol{\varphi}}_e$$

$$\Delta\ddot{\boldsymbol{\varphi}}_{de} + \boldsymbol{K}_{Do}\Delta\dot{\boldsymbol{\varphi}}_{de} + \boldsymbol{K}_{Po}\Delta\boldsymbol{\varphi}_{de} = \boldsymbol{0}$$

$$\dot{\boldsymbol{\varphi}}_d = \boldsymbol{T}^{-1}(\boldsymbol{\varphi}_d)\boldsymbol{\omega}_d$$

$$\ddot{\boldsymbol{\varphi}}_d = \boldsymbol{T}^{-1}(\boldsymbol{\varphi}_d)\left(\dot{\boldsymbol{\omega}}_d - \dot{\boldsymbol{T}}(\boldsymbol{\varphi}_d, \dot{\boldsymbol{\varphi}}_d)\dot{\boldsymbol{\varphi}}_d\right)$$

Alternative Euler Angles

- Orientation error: $\boldsymbol{R}_d^e = \boldsymbol{R}_e^T\boldsymbol{R}_d \Longrightarrow \boldsymbol{\varphi}_{de}$
  - Resolved angular acceleration

$$\boldsymbol{a}_o = \dot{\boldsymbol{\omega}}_d + \boldsymbol{T}_e(\boldsymbol{\varphi}_{de})(\boldsymbol{K}_{Do}\dot{\boldsymbol{\varphi}}_{de} + \boldsymbol{K}_{Po}\boldsymbol{\varphi}_{de}) - \dot{\boldsymbol{T}}_e(\boldsymbol{\varphi}_{de}, \dot{\boldsymbol{\varphi}}_{de})\dot{\boldsymbol{\varphi}}_{de}$$

$$\dot{\boldsymbol{\omega}}_e = \dot{\boldsymbol{\omega}}_d - \boldsymbol{T}_e(\boldsymbol{\varphi}_{de})\ddot{\boldsymbol{\varphi}}_{de} - \dot{\boldsymbol{T}}_e(\boldsymbol{\varphi}_{de}, \dot{\boldsymbol{\varphi}}_{de})\dot{\boldsymbol{\varphi}}_{de}$$

$$\boldsymbol{T}_e(\boldsymbol{\varphi}_{de}) = \boldsymbol{R}_e\boldsymbol{T}(\boldsymbol{\varphi}_{de})$$

choose $\boldsymbol{\varphi}_{de}$ so that $\boldsymbol{T}(\boldsymbol{0})$ is nonsingular (!)

  - Error dynamics

$$\ddot{\boldsymbol{\varphi}}_{de} + \boldsymbol{K}_{Do}\dot{\boldsymbol{\varphi}}_{de} + \boldsymbol{K}_{Po}\boldsymbol{\varphi}_{de} = \boldsymbol{0}$$