ADAPTIVE CONTROL

Andreja Rojko

OUTLINE

- Introduction
- Adaptive computed torque controller
  - o Example: Design of controller for 2 DOF robot
- Adaptive controller, inertia-related approach
  - o Example: Design of controller for 2 DOF robot
  - o Exercise: Adaptive controller, inertia related
- Soft computing adaptive controllers
  - o Example: ANFIS, adaptive fuzzy logic system

Introduction

- Even in well-structured environment the industrial robots have in many cases some **parametric uncertainties** in dynamic equations:
  - not exactly known inertias and masses due to the changing robot load,
  - unknown and varying friction parameters.
- Performance of already addressed control methods is very sensitive to parameter uncertainties. This limits their applicability to high precision motion control at manipulation of high weight payload.
- In adaptive controllers some parameters change (adapt) in order to compensate for initial parametric uncertainties.

Introduction

- Adaptive controllers can address parametric uncertainties thus improving the performance of the controllers (asymptotic trajectory tracking).
- Adaptation mechanism is driven by the tracking error.
- Global convergence of tracking error for adaptive controllers can be formally proven.
- Control algorithms that explicitly incorporate parameter estimation in the control law will be addressed here.

Introduction

- Adaptive controllers will be developed by separating unknown parameters (constants) from known functions of robot dynamics.
- Structural properties of robot dynamics will be used and formulation $\boldsymbol{\tau(t) = W(q, \dot{q}, \ddot{q})\varphi}$.

$$\boldsymbol{\tau(t) = W(q, \dot{q}, \ddot{q})\varphi}$$

$$friction1 = v_1 \dot{\Theta}_1 + k_1 sgn(\dot{\Theta}_1)$$

For 2-DOF planar robot.

$$\varphi = [m_1 \quad m_2 \quad k_1 \quad v_1 \quad k_2 \quad v_2]$$

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & 0 & 0 \\ 0 & w_{22} & 0 & 0 & w_{25} & w_{26} \end{bmatrix}$$

$$w_{11} = a_1^2 \ddot{\Theta}_1 + g a_1 c\Theta_1$$

$$w_{12} = [a_1^2 + a_2^2 + 2a_1 a_2 c\Theta_2]\ddot{\Theta}_1 + [a_2^2 + a_1 a_2 c\Theta_2]\ddot{\Theta}_2 - a_1 a_2 (2\dot{\Theta}_1 \dot{\Theta}_2 + \dot{\Theta}_2^2) s\Theta_2 + g a_1 c\Theta_1 + g a_2 c(\Theta_1 + \Theta_2)$$

$$w_{13} = sgn(\dot{\Theta}_1)$$

$$w_{14} = \dot{\Theta}_1$$

$$w_{22} = [a_2^2 + a_1 a_2 c\Theta_2]\ddot{\Theta}_1 + a_2^2 \ddot{\Theta}_2 + a_1 a_2 \dot{\Theta}_1^2 s\Theta_2 + g a_2 c(\Theta_1 + \Theta_2)$$

$$w_{25} = sgn(\dot{\Theta}_2)$$

$$w_{26} = \dot{\Theta}_2$$

Introduction

Important:

- Advantage of adaptive controllers over robust is that in case of changing/unknown parameters (changing load) the performance of adaptive controller improves over time.

Approximate CT controller

Centripetal, Coriolis term
$V(q,\dot{q}) = V_m(q,\dot{q})\dot{q}$

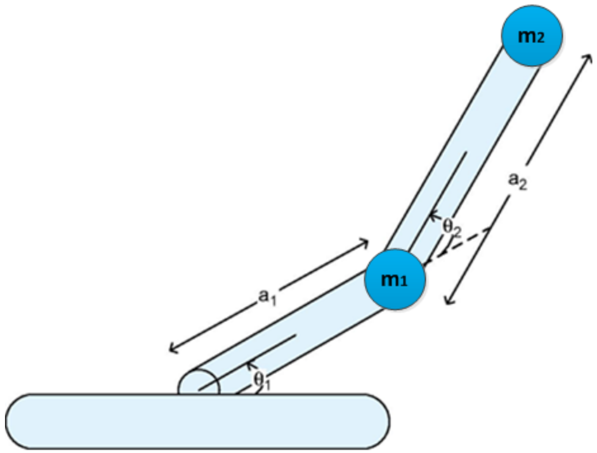
Robot dynamics:

$$\tau(t) = M(q)\ddot{q} + N(q,\dot{q})$$
$$= M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) + F(\dot{q})$$

Two common parametric uncertainties, load mass and friction parameters. Possible solution is to use CT controller, where those parameters are replaced by their estimated values → **approximate CT controller with PD feedback control loop:**

$$\tau(t) = \hat{M}(q)\left(\ddot{q}_D + K_v\dot{e} + K_p e\right) + \hat{V}_m(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q})$$
$$e(t) = q_D(t) - q(t)$$

Example: Approximate CT controller



Assumptions:

- Friction is negligible.
- Link lengths are exactly known.
- Masses are only estimated (parametric uncertainty):

$m_1 = 0.8 \pm 0.5kg,$

$m_2 = 2.3 \pm 0.1\ kg.$

Following will be investigated:

- Efficiency of conventional CT if estimated masses (different from the real ones) are used for calculation of non-linear feedforward control part.
- Error dynamics, respectively how to set parameters of PD feedback control law in such case..

Example: Approximate CT controller

Approximate CT

$\theta = q$

$$\tau_1 = \left((\hat{m}_1 + \hat{m}_1)a_1{}^2 + \hat{m}_2 a_2{}^2 + 2\hat{m}_2 a_1 a_2 c(q_2)\right)\left(\ddot{q}_{d1} + k_{v1}\dot{e}_1 + k_{p1}e_1\right) + \left(\hat{m}_2 a_1 a_2 c(q_2) + \hat{m}_2 l_2{}^2\right)\left(\ddot{q}_{d2} + k_{v2}\dot{e}_2 + k_{p2}e_2\right) - \hat{m}_2 a_1 a_2\left(2\dot{q}_1\dot{q}_2 + \dot{q}_2{}^2\right)s(q_2) + (\hat{m}_1 + \hat{m}_2) g a_1 c(q_2) + \hat{m}_2 g a_2\, c(q_1 + q_2)$$

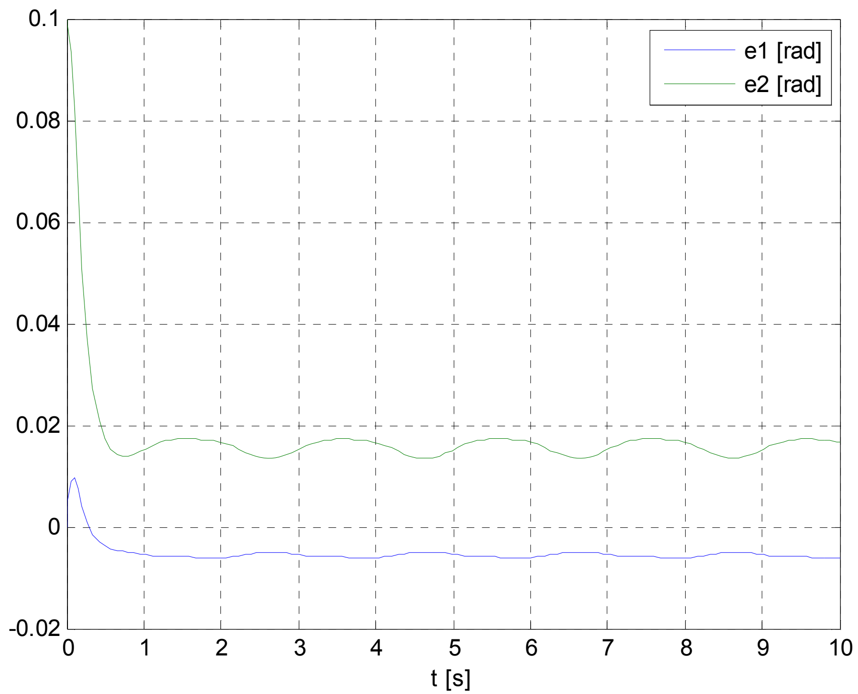
$\tau_2 = ..$

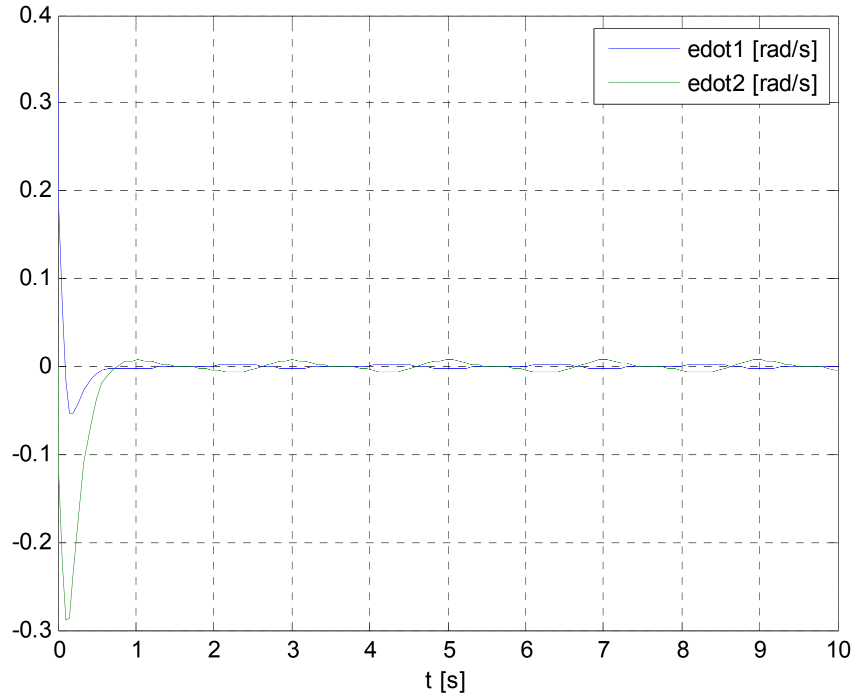
Parameters

- $a_1 = a_2 = 1\,m$, $K_{pi} = 100, K_{vi} = 20$
- Real masses (unknown):
  - o $m_1 = 0.8\,kg, m_2 = 2.3\,kg$
- Estimated masses:
  - o $\hat{m}_1 = 0.85\,kg$, $\hat{m}_2 = 2.2\,kg$

Example: Approximate CT controller

$\theta_{i,d} = sin(t)$





From simulation results following can be concluded:
Tracking errors remain bounded but don’t converge to zero.

Example: Approximate CT controller

Error dynamics for original CT: $\tau_c = M(\ddot{q}_D - u) + N$

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} w;\ w = M^{-1}\tau_D$$

For PD linear feedback control this gives:

$$\ddot{e} = u + w$$
$$\ddot{e} + K_v\dot{e} + K_p e = M^{-1}\tau_D$$

We know only estimated masses thereforeonly estimated value of moments of inertia. The real error dynamics is:

$$\ddot{e} + K_v\dot{e} + K_p e = \widehat{M}^{-1}\tau_D$$

- $\widehat{M}(q)$ is estimated inertia matrix, calculated with estimated masses
- $e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix};\ K_v = diag(k_{v,i});\ K_p = diag(k_{p,i})$

Adaptive CT controller

- Now we have seen the **issues with using CT controller in the presence of parametric uncertainties** (CT-like controller). Those are:
  - o Tracking errors remain bounded, but don't converge to zero.
  - o Linear control methods cannot be used to set the parameters.
- Next **adaptive CT-like controller will be derived:**
  - o Adaptive CT has the same structure as CT like controller with PD feedback, but selected (unknown or varying) parameters are adaptive.
  - o It guarantees asymptotic stability of trajectory tracking error in the presence of parametric uncertainties (for adaptive parameters).

Adaptive CT-like controller

$\underline{e} = [e, \dot{e}\,]^T$

where

$$e = [e_1, .., e_n\,]^T$$
$$\dot{e} = [\dot{e}_1, .., \dot{e}_n\,]^T$$

$$K_p = \begin{bmatrix} k_{p1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & k_{pn} \end{bmatrix}$$
$$K_v = \begin{bmatrix} k_{v1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & k_{vn} \end{bmatrix}$$

both diagonal, positive, nxn

$W(q, \dot{q}, \ddot{q})$ *nxr matrix*
$\varphi(q, \dot{q}, \ddot{q})$ *rx1 vector*
I_n, *nxn unit matrix*
0_n, *nxn zeros matrix*

[Craig 1985]

Robot adaptive CT controller is given by:

$$\tau = \hat{M}(q)(\ddot{q}_D + K_v\dot{e} + K_p e) + \hat{V}_m(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q})$$

By substituting $\ddot{e} = \ddot{q}_D - \ddot{q}$ we obtain:

$$\tau = \hat{M}(q)(\ddot{e} + k_v\dot{e} + k_p e) + \hat{M}(q)\ddot{q} + \hat{V}_m(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q})$$

By using robot dynamics properti $\tau = W(q,\dot{q},\ddot{q})\,\varphi = M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + G + F$

$$\tau = \hat{M}(q)(\ddot{e} + k_v\dot{e} + k_p e) + W(q,\dot{q},\ddot{q})\hat{\varphi}$$

It is also valid:

$$\tau = \hat{M}(q)(\ddot{e} + k_v\dot{e} + k_p e) + W(q,\dot{q},\ddot{q})\hat{\varphi} = W(q,\dot{q},\ddot{q})\,\varphi$$

Vector of estimated parameters

Vector of actual parameters

So the tracking error system is

$$\ddot{e} + k_v\dot{e} + k_p e = \hat{M}^{-1}(q)\,W(q,\dot{q},\ddot{q}) * [\varphi - \hat{\varphi}]$$
$$= \hat{M}^{-1}(q)\,W(q,\dot{q},\ddot{q}) * \tilde{\varphi}$$

$\tilde{\varphi}$ is parameter error, $\tilde{\varphi} = \varphi - \hat{\varphi}$

Tracking error system can be written in state-space form:

$$\dot{\underline{e}} = A\underline{e} + B\hat{M}^{-1}(q)\,W(q,\dot{q},\ddot{q})\,\tilde{\varphi}$$

$$\underline{e} = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, \quad B = \begin{bmatrix} 0_n \\ I_n \end{bmatrix}, \quad A = \begin{bmatrix} 0_n & I_n \\ -K_p & -K_v \end{bmatrix}$$

Adaptive CT controller: Derivation of adaptation law

$$\Gamma = \begin{bmatrix} \gamma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \gamma_r \end{bmatrix}$$

- Γ diagnal matrix, rxr
- P $2nx2n$ constant, positive, symetic matrix

Adaptive update law is derived by using Lyapunov stability analysis in such way to guarantee asymptotic stability of tracking error vector.

V is selected as positive definite function:

$$\boxed{V = \underline{e}^T P \underline{e} + \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi}}$$

$$\dot{V} = \underline{e}^T P \underline{\dot{e}} + \underline{\dot{e}}^T P \underline{e} + 2\tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \leftarrow [\dot{\tilde{\varphi}}^T \Gamma^{-1} \tilde{\varphi}]^T = \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \text{ since } \Gamma = \Gamma^T$$

Now $\underline{\dot{e}} = A\underline{e} + B\hat{M}^{-1}(q)W(q,\dot{q},\ddot{q})\tilde{\varphi}$ is substituted:

$$\dot{V} = \underline{e}^T P\left(Ae + B\hat{M}^{-1}(q)W(q,\dot{q},\ddot{q})\tilde{\varphi}\right) + \left(Ae + B\hat{M}^{-1}(q)W(q,\dot{q},\ddot{q})\tilde{\varphi}\right)^T P\underline{e} + 2\tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}}$$

Q is positive, symmetric matrix, which satisfies Lyapunov equation:

$$\dot{V} = -\underline{e}^T Q \underline{e} + 2\tilde{\varphi}^T\left(\Gamma^{-1}\dot{\tilde{\varphi}} + W^T(q,\dot{q},\ddot{q})\hat{M}^{-1}(q)B^T P\underline{e}\right)$$

where $A^T P + PA = -Q$

Adaptive CT controller

$$\dot V = -\underline{e}^T Q \underline{e} + 2\tilde\varphi^T\left(\Gamma^{-1}\dot{\tilde\varphi} + W^T(\underline{e},\dot{\underline{e}},\ddot{\underline{e}})\hat M^{-1}(\underline{e}) B^T P \underline{e}\right)$$

where $A^T P + P A = -Q$

For stability $\dot V$ shold be at least negative semidefinite, therefore the adaptation law is:

$$\dot{\tilde\varphi} = -\Gamma W^T(\underline{e},\dot{\underline{e}},\ddot{\underline{e}})\hat M^{-1}(\underline{e}) B^T P \underline{e}$$

and Ly. function derivative is

$$\dot V = -\underline{e}^T Q \underline{e}$$

But we need adaptive law for $\hat\varphi$. Since $\tilde\varphi = \varphi - \hat\varphi$ and $\dot\varphi = \emptyset$ (constant) we obtain adaptive update rule:

$$\boxed{\dot{\hat\varphi} = \Gamma W^T(\underline{e},\dot{\underline{e}},\ddot{\underline{e}})\hat M^{-1}(\underline{e}) B^T P \underline{e}}$$

Asimptotic stability of $\underline{e}$ (error tracking vector) can be proven by furter stability analysis (by using Barbalat's lemma & Rayleigh-Ritz theorem).

Adaptive CT controller

Summary of adaptive CT controller

Torque controller:

$$\tau = \hat{M}(q)(\ddot{q}_D + k_v\dot{e} + k_p e) + \hat{V}_m(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q})$$

Update rule:

$$\dot{\hat{\varphi}} = \Gamma W^T(q,\dot{q},\ddot{q})\hat{M}^{-1}(q)B^T P \underline{e}$$

$$\underline{e} = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, \quad B = \begin{bmatrix} \emptyset_n \\ I_n \end{bmatrix}, \quad A = \begin{bmatrix} \emptyset_n & I_n \\ -k_p & -k_v \end{bmatrix}$$

$$W(q,\dot{q},\ddot{q})\hat{\varphi} = \hat{M}(q)\ddot{q} + \hat{V}_m(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q})$$

$$A^T P + PA = -Q$$

P, Q, positive definite, symmetric matrices

stability

Tracking error vector $\underline{e}$ is asymptotically stable.

Other

$\ddot{q}$ needs to be measured. Parameter resetting needed.

Note:
Adaptation parameters should stay in specified region, if they go outside they are resetted.

Example: Adaptive CT controller

- Adaptive CT controller will be derived for 2 DOF planar robot.
- Assumptions:
  - o Friction is negligible.
  - o Link lengths are exactly known.
  - o Masses are only estimated (so we have parametric uncertainty).
- We will use CT control, calculated with estimated masses, which will be adaptive parameter.
- Update rules (adaptation law) for $\hat{m}_1$ and $\hat{m}_2$ need to be derived.

Example: Adaptive CT controller

Craig 1985

- For 2-DOF robot:

$\tilde{\tau}(t) = W(q,\dot{q},\ddot{q})\tilde{\varphi},$ this is already known property of robot dynamics

$$W(q,\dot{q},\ddot{q}) = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}, \tilde{\varphi} = \begin{bmatrix} \tilde{\varphi}_1 \\ \tilde{\varphi}_2 \end{bmatrix} = \begin{bmatrix} m_1 - \widehat{m}_1 \\ m_2 - \widehat{m}_2 \end{bmatrix}$$

$W_{11} = a_1{}^2\ddot{q}_1 + a_1 g c(q_1)$

$W_{12} = a_2{}^2(\ddot{q}_1 + \ddot{q}_2) + a_1 a_2 c(q_2)(2\ddot{q}_1 + \ddot{q}_2) + a_1{}^2\ddot{q}_1 - a_1 a_2 s(q_2)\dot{q}_2{}^2 -$
$2a_1 a_2 s(q_2)\dot{q}_1\dot{q}_2 + a_2 g c(q_1 + q_2) + a_1 g c(q_1)$

$W_{21} = 0$

$W_{22} = a_1 a_2 c(q_2)\ddot{q}_1 + a_1 a_2 s(q_2)\dot{q}_1{}^2 + a_2 g c(q_1 + q_2) + a_2{}^2(\ddot{q}_1 + \ddot{q}_2)$

Example: Adaptive CT controller

- We select following parameters/matrices:

  $K_v = k_v I_n$; $K_p = k_p I_n$; k_v, k_p are positive scalars

  $$P = \begin{bmatrix} P_1 I_n & P_2 I_n \\ P_2 I_n & P_3 I_n \end{bmatrix} = 0.5 \begin{bmatrix} (K_p + 0.5k_v) I_n & 0.5 I_n \\ 0.5 I_n & I_n \end{bmatrix}$$

  P is symmetric and with $k_v > 1$ positive definite.

- Check to see if this gives positive definite Q:

  $$-Q = A^T P + PA$$

Example: Adaptive CT controller

- It can be calculated: $Q = \begin{bmatrix} 0.5k_p I_n & 0_n \\ 0_n & (K_v + 0.5)I_n \end{bmatrix}$

- Finally it can be concluded that for $k_{v,i} > 1$ Q is positive definite and symmetric, so chosen parameters are suitable.
- Finding suitable P and Q is (usually) not an easy task.

Example: Adaptive CT controller

- $\dot{\hat{\varphi}} = \Gamma W^T(q,\dot{q},\ddot{q})\widehat{M}^{-1}(q)B^T P e$ and:

$$\hat{\varphi} = [\hat{m}_1, \hat{m}_2]^T$$

$$\dot{\hat{m}}_1 = \gamma_1\Big((W_{11} MI_{11} + W_{21} MI_{21})(P_2 e_1 + P_3 \dot{e}_1) + (W_{11} MI_{21} + W_{21} MI_{22})(P_2 e_2 + P_3 \dot{e}_2) \Big)$$

$$\dot{\hat{m}}_2 = \gamma_2\Big((W_{12} MI_{11} + W_{22} MI_{21})(P_2 e_1 + P_3 \dot{e}_1) + (W_{12} MI_{21} + W_{22} MI_{22})(P_2 e_2 + P_3 \dot{e}_2) \Big)$$

$$\widehat{M}^{-1}(e,\dot{e},\ddot{e}) = \begin{bmatrix} MI_{11} & MI_{21} \\ MI_{21} & MI_{22} \end{bmatrix}$$

$$\Delta = (2\hat{m}_2 l_1 l_2 c\Theta_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2) l_1^2)(\hat{m}_2 l_2^2) - (\hat{m}_2 l_2^2 + \hat{m}_2 l_1 l_2 c\Theta_2)^2$$

$$MI_{11} = \frac{1}{\Delta}(\hat{m}_2 l_2^2)$$

$$MI_{21} = -\frac{1}{\Delta}(\hat{m}_2 l_1 l_2 c\Theta_2 + \hat{m}_2 l_2^2)$$

$$MI_{22} = \frac{1}{\Delta}(2\hat{m}_2 l_1 l_2 c\Theta_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2) l_1^2)$$

OUTLINE

- Introduction
- Adaptive computed torque controller
  - o Example: Design of controller for 2 DOF robot
- Adaptive controller, inertia-related approach
  - o Example: Design of controller for 2 DOF robot
  - o Exercise: Adaptive controller, inertia related
- Soft computing adaptive controllers
  - o Example: ANFIS, adaptive fuzzy logic system

Adaptive controller, inertia related

- Issues with just derived adaptive CT controller:
  - o $M^{-1}(q)$ should exist. This is very hard to assure with large, unknown payloads.
  - o $\ddot{q}$ needs to be measured.
- Therefore another approach was developed [Slotine & Li, 1985]. In this case Lyapunov function is chosen and both control law and adaptation law are derived via Lyapunov stability analysis.
- Chosen Lyapunov function should be function of tracking error and parameter error.

Adaptive controller, inertia related

Lyapunov function should be function of tracking error & parameter error.

$$\boxed{V = \frac{1}{2} r^T M(e) r + \frac{1}{2} \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi}}$$

here:

$r = \Lambda e + \dot{e}$ tracking error $\quad \Gamma = \mathrm{diag}(\gamma_1 .. \gamma_r)$, pos. def., const.

$\tilde{\varphi} = \varphi - \hat{\varphi}$ parameter error $\quad \Lambda = \mathrm{diag}(\lambda_1 .. \lambda_n)$, pos. def., const.

Note dimensions:

n .. number of joints

r ... number of unknown, adaptive parameters

$\tilde{\varphi} \in \mathbb{R}^r$; $r, e, \dot{e} \in \mathbb{R}^n$

How was r called in the robust control chapter?

$$V = \frac{1}{2} r^T M(q) r + \frac{1}{2} \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} \qquad r = \Lambda e + \dot{e}$$

$$\dot{V} = r^T M(q) \dot{r} + \frac{1}{2} r^T \dot{M}(q) r + \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}}$$

→ $\dot{r}$ needs to be substituted. It turns out that for robot dynamics $\tau = M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) + F(\dot{q})$ following can be written:

$$M(q)\dot{r} = Y(\cdot)\varphi - \tau - V_m(q,\dot{q}) r \qquad (1)$$

$$\tilde{\varphi} = \varphi - \hat{\varphi}$$

$$Y(\cdot)\varphi = M(q)(\ddot{q}_d + \Lambda \dot{e}) + V_m(q,\dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(\dot{q}) \qquad (2)$$

Substituting first eq. into second gives:

$$\dot{V} = r^T(Y(\cdot)\varphi - \tau) + r^T \underbrace{\left(\frac{1}{2}\dot{M}(q) - V_m(q,\dot{q})\right)}_{\text{skew sym. property of dy.}} r + \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}}$$

$S(q,\dot{q}) \equiv \dot{M}(q) - 2V_m(q,\dot{q})$,
$S(q,\dot{q})$ is skew symmetric and,
accordingly, $x^T S(q,\dot{q}) x = 0, \forall x$.

$$\Downarrow$$

$$\dot{V} = r^T(Y(\cdot)\varphi - \tau) + \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}}$$

WE SELECT CONTROL TORQUE:

$$\tau = Y(\cdot)\hat{\varphi} + k_v r$$

$K_v = \mathrm{diag}(K_{v_1} \ldots K_{v_n})$
pos. def.

Adaptive controller, inertia related

and following is obtained:

$$\dot{V} = -r^T K_v r + \tilde{\varphi}^T(\Gamma^{-1}\dot{\tilde{\varphi}} + Y^T(\cdot)r)$$

Then we SELECT UPDATE (ADAPTIVE) RULE:

$$\boxed{\dot{\hat{\varphi}} = -\dot{\tilde{\varphi}} = \Gamma Y^T(\cdot)r}$$

to obtain

$$\dot{V} = -r^T K_v r$$

Furter analysis shows that tracking error is asymtotically stable.

Adaptive controller, inertia related

Summary

- Controller torque:

$$\tau = Y(.)\hat{\varphi} + K_v r$$
$$r = \Lambda e + \dot{e}$$

- Update (adaptation) rule:

$$\dot{\hat{\varphi}} = \Gamma Y^T(.)r$$

$$Y(.)\hat{\varphi} = \hat{M}(q)(\ddot{q}_D + \Lambda\dot{e}) + \hat{V}_m(q,\dot{q})(\dot{q}_D + \Lambda e) + \hat{G}(q) + \hat{F}(\dot{q})$$

and

$$Y(.)\varphi = M(q)(\ddot{q}_D + \Lambda\dot{e}) + V_m(q,\dot{q})(\dot{q}_D + \Lambda e) + G(q) + F(\dot{q})$$

Adaptive controller, inertia related

Persistency of excitation

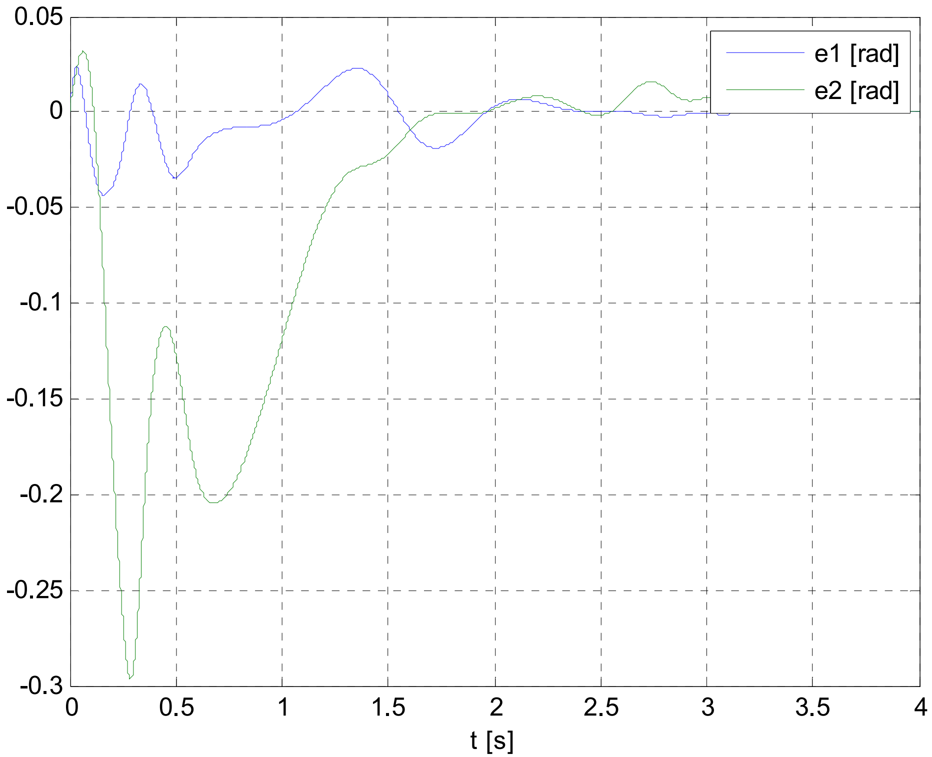
In order that adaptive control law leads to identification of real parameters $\tilde{\varphi} = \varphi - \hat{\varphi} = 0$ additional condition on regression matrix needs to be fulfilled.

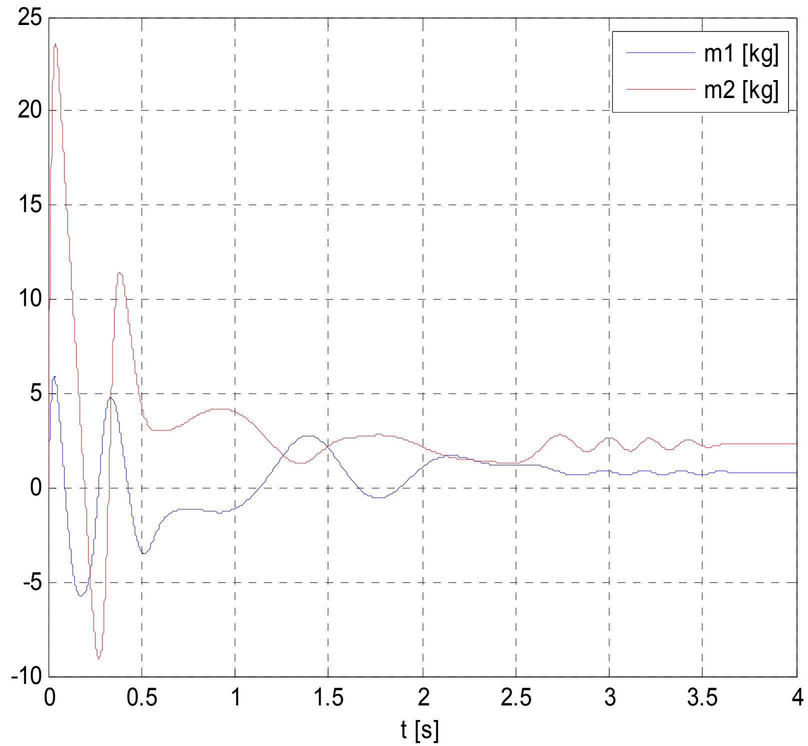
- o $\hat{\varphi} \in R^n$, time-varying estimate if unknown constant vector
- o $\varphi \in R^n$, constant value of real constant vector

Adaptation rule: $\dot{\hat{\varphi}} = -\dot{\tilde{\varphi}} = \Gamma Y^T(.)r = \Gamma Y^T(.)(\Lambda e + \dot{e})$

Condition: $Y^T(.)$ needs to vary in such way that complete parameter space is spanned. With other worlds, the desired trajectory have to be such that all unknown parameters can be identified. Only in this case the parameter error $\tilde{\varphi}$ will go to zero; otherwise it will only remain bounded. Reference matters!

Exercise: Adaptive controller, inertia related





Example of simulation results of well designed controller.

REPORT REQUIRED

Exercise: Adaptive controller, inertia related

- Design and simulate adaptive inertia-related adaptive controller for 2 DOF planar robot for $m_1 = 0.8\ kg$, $m_2 = 2.3\ kg$, $\hat{m}_1(0) = 0\ kg$, $\hat{m}_2(0) = 0\ kg$. $q_{d1} = q_{d2} = sin(t)$. Friction is negligible.
- Steps:
  - o Summarize the control law.
  - o Derive $Y(.)$.
  - o Build simulation scheme and find suitable parameters. Observe position and velocity error and adaptation of parameter vector.

OUTLINE

- Introduction
- Adaptive computed torque controller
  - o Example: Design of controller for 2 DOF robot
- Adaptive controller, inertia-related approach
  - o Example: Design of controller for 2 DOF robot
  - o Exercise: Adaptive controller, inertia related
- Soft computing adaptive controllers
  - o Example: ANFIS, adaptive fuzzy logic system

Soft computing adaptive controllers

- Soft computing adaptive controllers address parametric and structural uncertainties.
- This property is still used
  $\tau(t) = M(q)\ddot{q} + V(q,\dot{q}) + F_v\dot{q} + F_d(\dot{q}) + G(q) = W(q,\dot{q},\ddot{q})\varphi$
- Parameter vector includes unknown adaptive parameters, while regression matrix is approximates by artificial neural network or fuzzy logic system.
- Example of adaptive FLS (ANFIS): parameter vector is for example vector of position of output membership functions, which compensates for changing parameters and not sufficient expert knowledge (used to build rule base).

Soft computing adaptive controllers

- Adaptive neuro-fuzzy systems (ANFIS) are fuzzy logic systems with adaptive parameters.
- Parameters of ANFIS are adapted by learning algorithms known from ANNs.
- ANFIS are mostly used for system identification.
- For example ANFIS which is used for estimation of robot dynamic model can adapt to the changes in the robot dynamics caused by the changing robot payload or by the contact with the environment (in hybrid position-force control controllers).
- Note! Soft computing/Control systems special topics course is necessary to understand continuation of this chapter.

Soft computing adaptive controllers

Most of ANFIS fuzzy logic systems are Takagi-Sugeno with singleton MFs for outputs.

Takagi-Sugeno fuzzy rules:

Singleton rules

R^l : If $x^1 = X^{1,l}$ and .. and $x^n = X^{n,l}$ then $\boxed{y^l = c_0^l} + c_1^l x^1 + .. + c_n^l x^n$

Crisp output *y* of FLS with *M* rules, *n* inputs, with $\overline{y}^l$ denoted centers of output fuzzy sets, with sum-product inferencing and simplified centre-of-gravity method of defuzzification is given with:

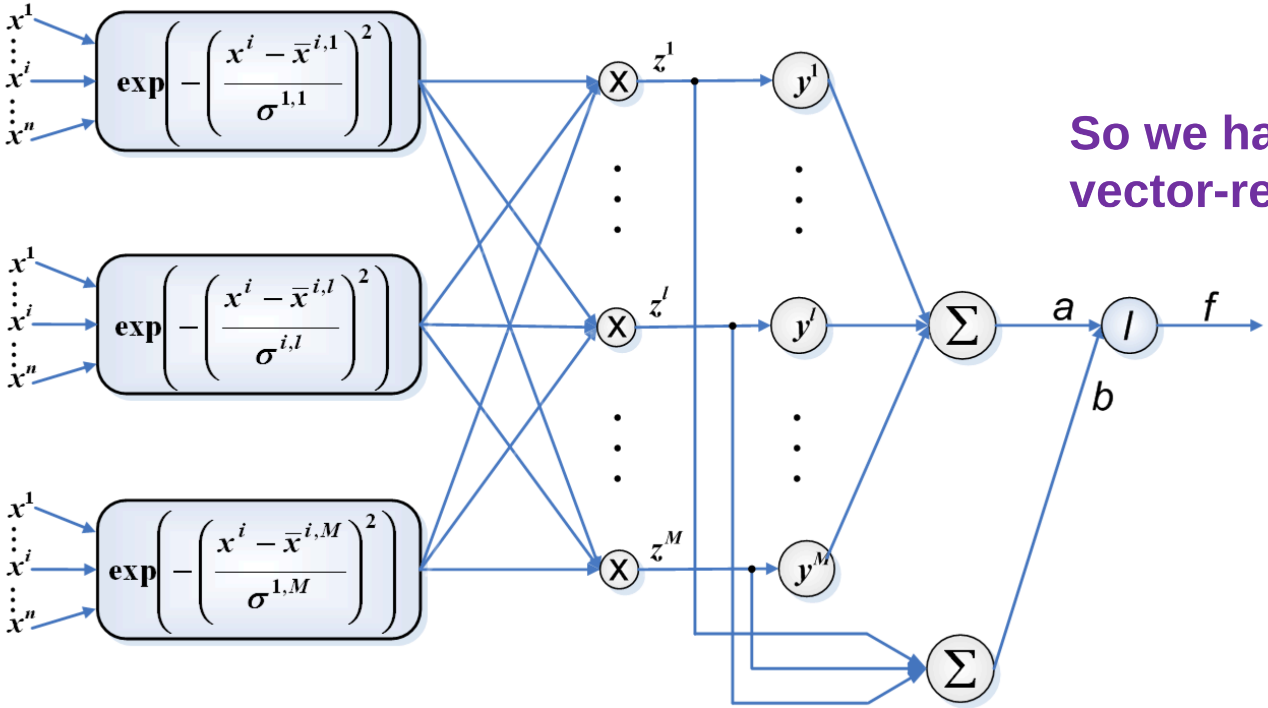
$$y = \frac{\sum_{l=1}^{M} \overline{y}^l \cdot \prod_{i=1}^{n} \mu_{X^{i,l}}\left(x^i\right)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{X^{i,l}}\left(x^i\right)}$$

Soft computing adaptive controllers

FLS with such structure and bell shaped Input MFs can be graphically presented as 3-layer ANN.

$$y=\frac{\sum_{l=1}^{M}\bar{y}^{l}\prod_{i=1}^{n}\exp\left(-\left(\frac{x^{i}-\bar{x}^{i,l}}{\sigma^{i,l}}\right)^{2}\right)}{\sum_{l=1}^{M}\prod_{i=1}^{n}\exp\left(-\left(\frac{x^{i}-\bar{x}^{i,l}}{\sigma^{i,l}}\right)^{2}\right)}=\frac{\sum_{l=1}^{M}\boxed{y^{l}}\cdot z^{l}}{\sum_{l=1}^{M}z^{l}}=\boxed{\hat{\varphi}}\cdot\hat{W}(q,\dot{q},\ddot{q})$$

Parameter vector - adaptable



So we have parameter vector-regressor form

Soft computing adaptive controllers

Model based robot motion control; ANFIS system for estimation of most of the robot's dynamics

$$\boldsymbol{\tau} = M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_{\mathbf{t}}(\mathbf{q},\dot{\mathbf{q}}) + \boldsymbol{\tau}_b + \boldsymbol{\tau}_d$$

$$\tau_k = M_{kk}\ddot{q}_k + \Delta M_{kk}(\mathbf{q})\ddot{q}_k + \sum_{j=1,\, j\neq k}^{m} M_{kj}(\mathbf{q})\ddot{q}_j + \sum_{j=1}^{m}\sum_{l=1}^{m} C_{jl,k}(\mathbf{q})\dot{q}_j\dot{q}_l + G_k(\mathbf{q}) + \tau_{t,k}(\mathbf{q},\dot{\mathbf{q}}) + \tau_{b,k} + \tau_{d,k}$$

Part of the dynamics that should be estimated by ANFIS

$$w_k = \Delta M_{kk}(\mathbf{q})\ddot{q}_k + \sum_{j=1,\, j\neq k}^{m} M_{kj}(\mathbf{q})\ddot{q}_j + \sum_{j=1}^{m}\sum_{l=1}^{m} C_{jl,k}(\mathbf{q})\dot{q}_j\dot{q}_l + G_k(\mathbf{q}) + \tau_{t,k}(\mathbf{q},\dot{\mathbf{q}}) + \tau_{b,k} + \tau_{d,k}$$

FL rule for estimation robot dynamics

IF position = small positive AND velocity= big positive AND acceleration = positive THEN torque = τ

How to keep FLS transparent -> Introduction of FL subsystems

Soft computing adaptive controllers

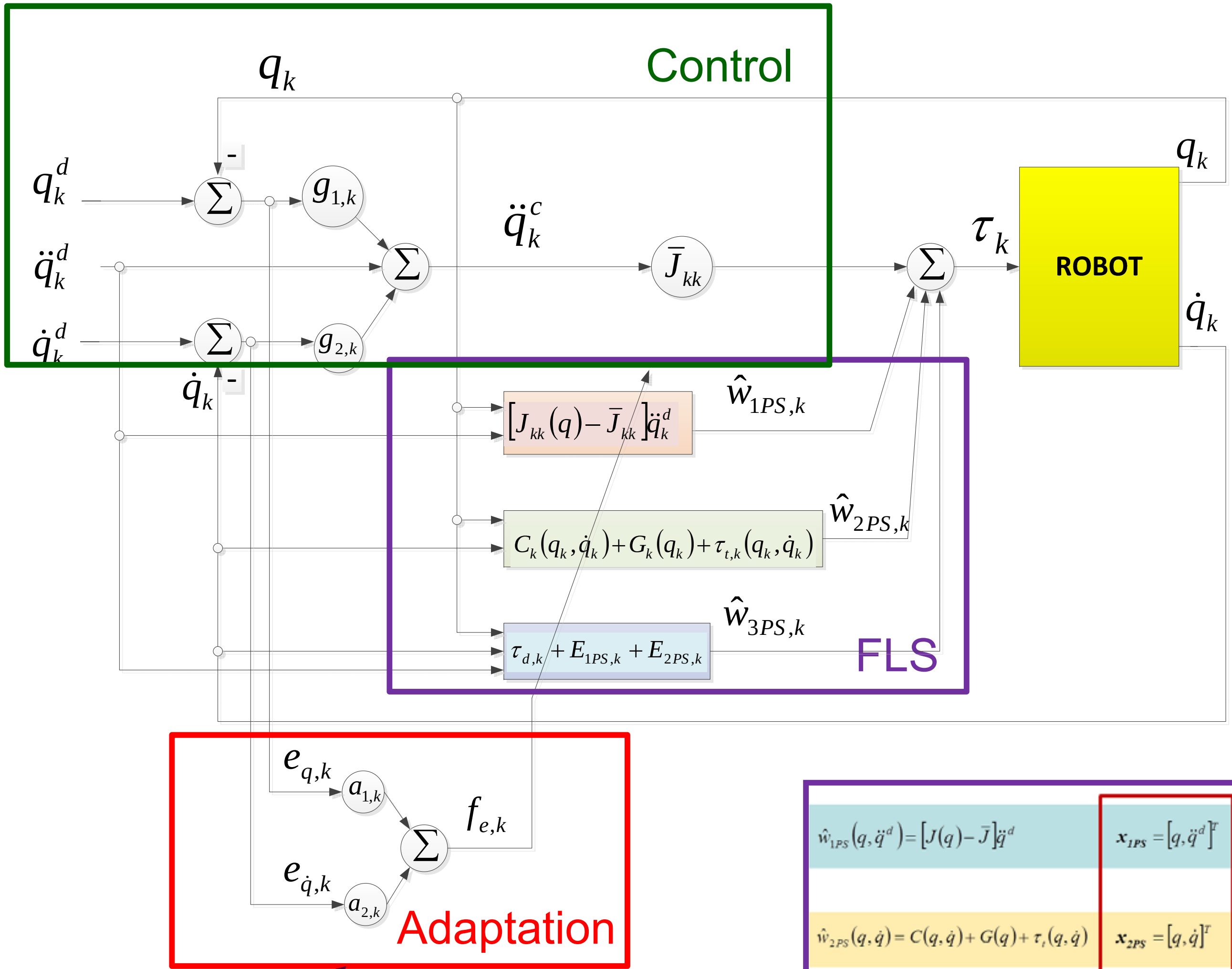
Part of the dynamics that should be estimated by ANFIS

$$w_k = \Delta M_{kk}(\mathbf{q})\ddot{q}_k + \sum_{j=1,\, j\neq k}^{m} M_{kj}(\mathbf{q})\ddot{q}_j + \sum_{j=1}^{m}\sum_{l=1}^{m} C_{jl,k}(\mathbf{q})\dot{q}_j\dot{q}_l + G_k(\mathbf{q}) + \tau_{t,k}(\mathbf{q},\dot{\mathbf{q}}) + \tau_{b,k} + \tau_{d,k}$$

To keep transparency, three FL subsystems $\hat{w} = \hat{w}_{1PS} + \hat{w}_{2PS} + \hat{w}_{3PS}$

		Inputs	Estimation error	
1.	$\hat{w}_{1PS}(q,\ddot{q}^d) = [M(q) - \overline{M}]\ddot{q}^d$	$\boldsymbol{x}_{1PS} = [q,\ddot{q}^d]^T$	$E_{1PS} = \hat{w}_{1PS}(q,\ddot{q},t,\hat{\boldsymbol{\theta}}_{1PS}) - w_{1PS}(q,\ddot{q},t,\boldsymbol{\theta}_{1PS})$	Inertia
2.	$\hat{w}_{2PS}(q,\dot{q}) = C(q,\dot{q}) + G(q) + \tau_t(q,\dot{q})$	$\boldsymbol{x}_{2PS} = [q,\dot{q}]^T$	$E_{2PS} = \hat{w}_{2PS}(q,\ddot{q},t,\hat{\boldsymbol{\theta}}_{2PS}) - w_{2PS}(q,\ddot{q},t,\boldsymbol{\theta}_{2PS})$	Centrifugal, coriolis, friction, gravitation
3.	$\hat{w}_{3PS}(q,\dot{q},\ddot{q}) = \tau_d + E_{1PS} + E_{2PS}$	$\boldsymbol{x}_{3PS} = [q,\dot{q},\ddot{q}^d]^T$		Everything else

Soft computing adaptive controllers



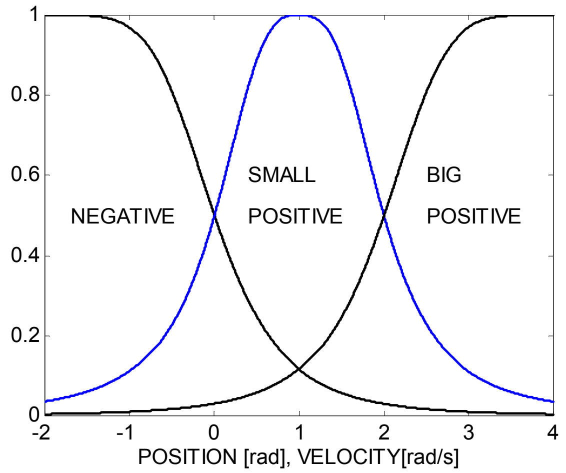
J=M

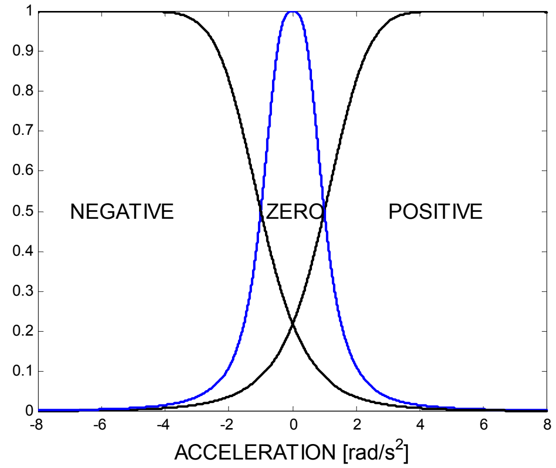
Stability of adaptation/learning law
can be proven by Lyapunov therem

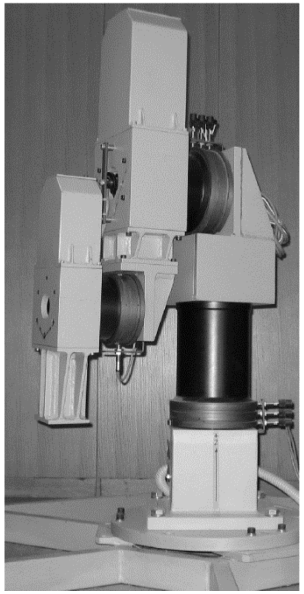
$\hat{w}_{1PS}(q,\ddot{q}^d)=[J(q)-\bar{J}]\ddot{q}^d$	$\boldsymbol{x}_{1PS}=[q,\ddot{q}^d]^T$	$E_{1PS}=\hat{w}_{1PS}(q,\ddot{q},t,\hat{\boldsymbol{\theta}}_{1PS})-w_{1PS}(q,\ddot{q},t,\boldsymbol{\theta}_{1PS})$	Inertia
$\hat{w}_{2PS}(q,\dot{q})=C(q,\dot{q})+G(q)+\tau_t(q,\dot{q})$	$\boldsymbol{x}_{2PS}=[q,\dot{q}]^T$	$E_{2PS}=\hat{w}_{2PS}(q,\ddot{q},t,\hat{\boldsymbol{\theta}}_{2PS})-w_{2PS}(q,\ddot{q},t,\boldsymbol{\theta}_{2PS})$	Centrifugal, coriolis, friction, gravitation
$\hat{w}_{3PS}(q,\dot{q},\ddot{q})=\tau_d+E_{1PS}+E_{2PS}$	$\boldsymbol{x}_{3PS}=[q,\dot{q},\ddot{q}^d]^T$		Everything else
	Inputs	Estimation error	

Soft computing adaptive controllers

Membership functions:





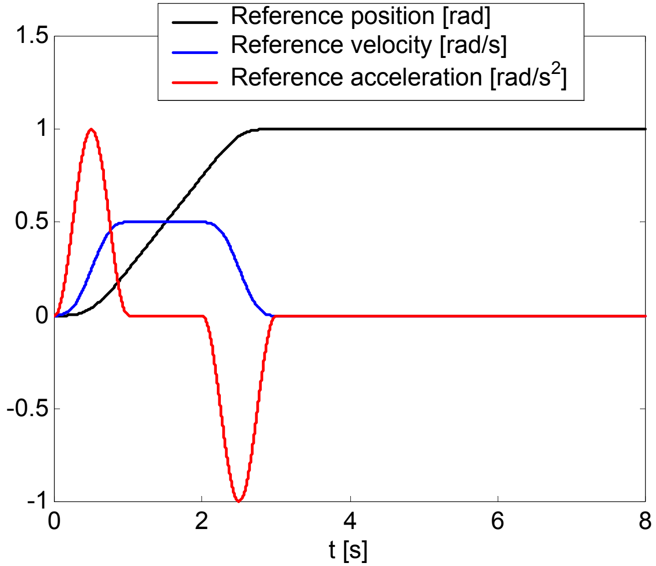


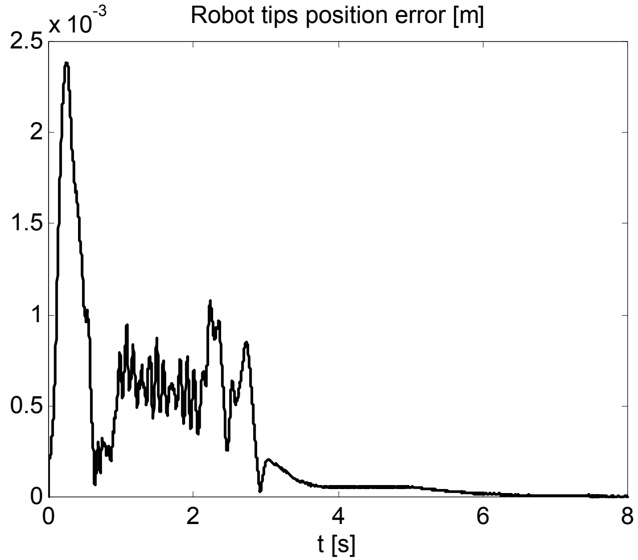
	Inputs	Estimation error	
$\hat{w}_{1PS}(q,\ddot{q}^d)=[J(q)-\bar{J}]\ddot{q}^d$	$x_{1PS}=[q,\ddot{q}^d]^T$	$E_{1PS}=\hat{w}_{1PS}(q,\ddot{q},t,\hat{\theta}_{1PS})-w_{1PS}(q,\ddot{q},t,\theta_{1PS})$	Inertia
$\hat{w}_{2PS}(q,\dot{q})=C(q,\dot{q})+G(q)+\tau_t(q,\dot{q})$	$x_{2PS}=[q,\dot{q}]^T$	$E_{2PS}=\hat{w}_{2PS}(q,\ddot{q},t,\hat{\theta}_{2PS})-w_{2PS}(q,\ddot{q},t,\theta_{2PS})$	Centrifugal, coriolis, friction, gravitation
$\hat{w}_{3PS}(q,\dot{q},\ddot{q})=\tau_d+E_{1PS}+E_{2PS}$	$x_{3PS}=[q,\dot{q},\ddot{q}^d]^T$		Everything else

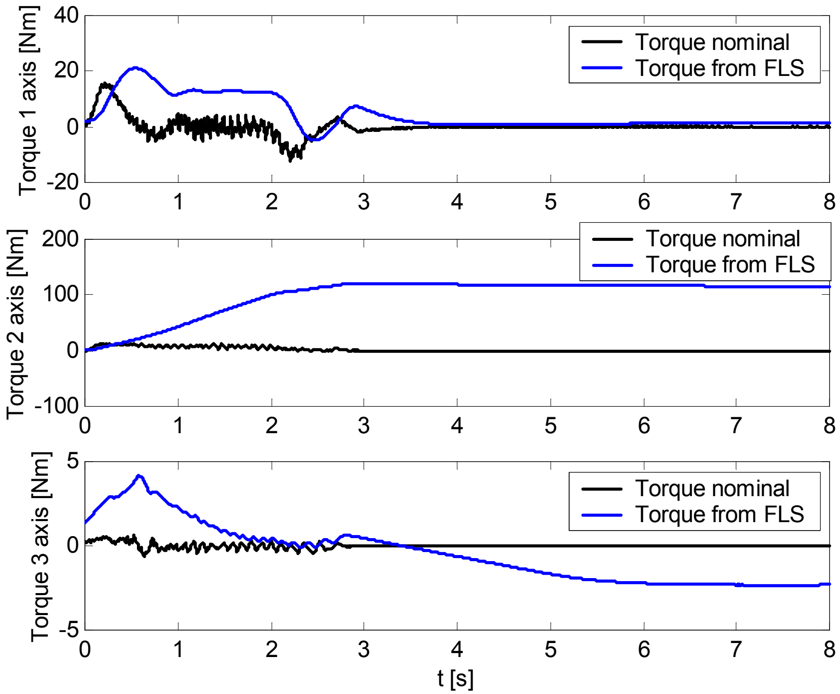
NUMBER OF RULES

1. JOINT: 3, 3, 3
2. JOINT: 3, 3, 9
3. JOINT: 3, 3, 9

Soft computing adaptive controllers







ANFIS for robot control

Sudden load change