Robot Control

11.12.15

Symmetric matrix

$A = A^T$

$a_{ij} = a_{ji} \quad \forall\, i,j$

$\Rightarrow A x y = (A x)^T y = x^T A^T y = x^T A y$

Skew symmetric matrix

$A^T = -A$

$-a_{ij} = a_{ji} \quad \forall\, i,j \qquad \det(A) = \det(A^T) = \det(-A)$

positive definite matrix

$x^T A x > 0 \quad \forall x \neq 0$

vector norm

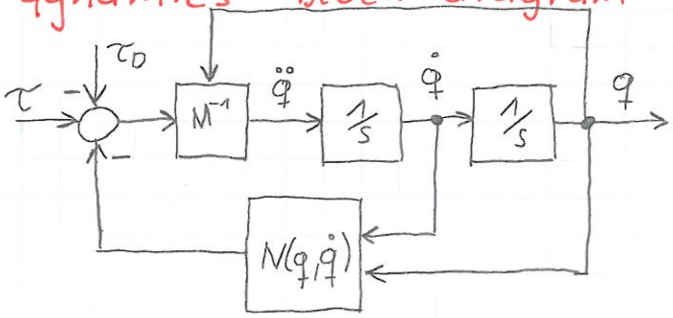
$\|x\|_1 = \sum_{i=1}^{n} |x_i| \qquad \|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2} \qquad \|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$

$\|x\| > 0 \quad \forall x \neq 0$

$\|\alpha x\| = |\alpha|\,\|x\|$

$\|x\| = 0 \iff x = 0$

Robot dynamics block diagram



$N(q,\dot{q}) = V(q,\dot{q}) + F_v \dot{q} + G(q)$

Description of robot dynamics in state space

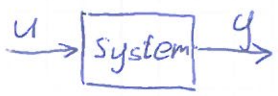
→ Recap: lin. SISO $x \in \mathbb{R}^n$ | MIMO $x \in \mathbb{R}^n \; y \in \mathbb{R}^p \; u \in \mathbb{R}^m$

$\dot{x} = A x + b u \qquad A \in \mathbb{R}^{n\times n},\; b \in \mathbb{R}^{n\times 1}$

$y = c^T x + d u \qquad c^T \in \mathbb{R}^{1\times n},\; d \in \mathbb{R}^1$

$\dot{x} = A x + B u \qquad A \in \mathbb{R}^{n\times n},\; B \in \mathbb{R}^{n\times m}$

$y = C x + D u \qquad C \in \mathbb{R}^{p\times n},\; D \in \mathbb{R}^{p\times n}$



Ex · double integrator

$$\ddot{y} = u$$

$$x_1 = y, \quad x_2 = \dot{x}_1 = \dot{y}, \quad \dot{x}_2 = \ddot{y} = u$$

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

$$y = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + 0\, u$$

MIMO, non linear system

$$\frac{d^n y(t)}{dt^n} = h\left(y, \frac{dy}{dt}, \dots, \frac{d^{n-1}y}{dt^{n-1}}, u, \frac{du}{dt}, \dots, \frac{d^n u}{dt^n}\right)$$

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 \dots & 0 \\ \vdots \\ h(\dots) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad y = x_1$$

compact form

$$\dot{x} = h(x, u)$$

$$y = C\, x \qquad C = (I, 0, 0)$$

$$u = (u, \dot{u}, \dots)$$

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + G(q) \qquad q \in \mathbb{R}^n \qquad M \in \mathbb{R}^{n \times n} \qquad \tau \in \mathbb{R}^{n \times 1}$$

input τ

output q

states $x = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} \in \mathbb{R}^{2n}$

$$\ddot{q} = M^{-1}(q)\left(-V(q, \dot{q}) - G(q) + \tau\right)$$

$$\begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} \dot{q} \\ M^{-1}(q)\left(-V(q, \dot{q}) - G(q)\right) \end{pmatrix} + \begin{pmatrix} \emptyset \\ M^{-1} \end{pmatrix} \tau$$

$$y = \begin{pmatrix} I & \emptyset \end{pmatrix} \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$$

Joint space

$q = (q_1 \dots q_n)^T$

$\dot{q} = (\dot{q}_1 \dots \dot{q}_n)^T$

Task space

$x = (x, y, z, R_x, R_y, R_z)$

$\dot{x} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$

$\ddot{x} = (a_x, a_y, a_z, \alpha_x, \alpha_y, \alpha_z)$

$\tau = J^T F$

Transformation

$\dot{x} \overset{?}{=} f(\dot{q})$

$\dot{x} = J\dot{q} \rightarrow \dot{q} = J^{-1}\dot{x}$

$\ddot{x} \overset{?}{=} f_1(\ddot{q})$

$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \rightarrow \ddot{q} = J^{-1}\ddot{x} - J^{-1}\dot{J}\dot{q} = J^{-1}\ddot{x} - J^{-1}\dot{J}J^{-1}\dot{x}$

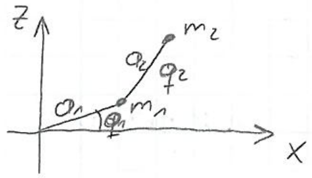
Dynamics in Joint space

$\tau = M\ddot{q} + \underbrace{N(q,\dot{q})}_{\substack{\text{centripedal}\\ \text{corriolis}\\ \text{gravitation}}} + \tau_D$ => calculate this in task space

$J^T F = MJ^{-1}\ddot{x} - MJ^{-1}\dot{J}J^{-1}\dot{x} + N(q,\dot{q}) + \tau_D$

$\Rightarrow \quad F = J^{-T}MJ^{-1}\ddot{x} - J^{-T}MJ^{-1}\dot{J}J^{-1}\dot{x} + J^{-T}N(q,\dot{q}) + J^{-T}\tau_D$

Ex



1. m_1: $\vec{r}_1 = \begin{pmatrix} a_1 c_1 \\ a_1 s_1 \end{pmatrix}$ $\quad \dot{r}_1 = \begin{pmatrix} -a_1 s_1 \dot{q}_1 \\ a_1 c_1 \dot{q}_1 \end{pmatrix}$

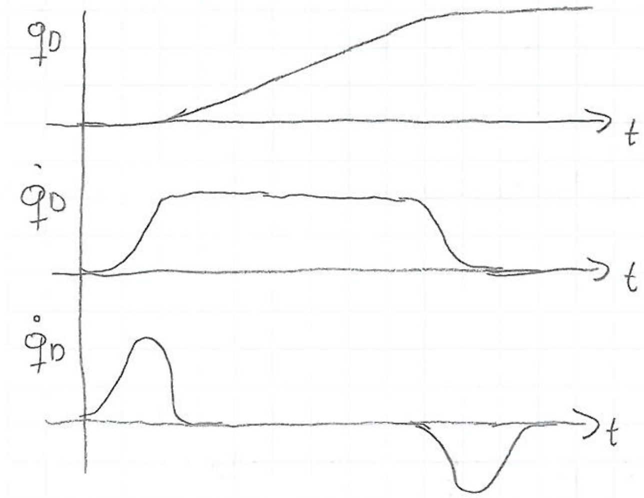
$\|\dot{r}_1\|^2 = a_1^2 \dot{q}_1^2$

2. m_2: $r_2 = \begin{pmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \end{pmatrix}$ $\quad \dot{r}_2 = \begin{pmatrix} -a_1 s_1 \dot{q}_1 - a_2 s_{12}(\dot{q}_1 + \dot{q}_2) \\ a_1 c_1 \dot{q}_1 + a_2 c_{12}(\dot{q}_1 + \dot{q}_2) \end{pmatrix}$

$\|\dot{r}_2\|^2 = a_1^2 \dot{q}_1^2 + a_2(\dot{q}_1 + \dot{q}_2)^2 + 2a_1 a_2 \dot{q}_1(\dot{q}_1 + \dot{q}_2)c_2$

$J = \begin{pmatrix} \frac{\partial r^{(1)}}{\partial q_1} & \frac{\partial r^{(1)}}{\partial q_2} \\ \frac{\partial r^{(2)}}{\partial q_1} & \frac{\partial r^{(2)}}{\partial q_2} \end{pmatrix}$

Desired joint space motion trajectory



Tracking error

$$e = q_D - q$$
$$\dot{e} = \dot{q}_D - \dot{q}$$
$$\ddot{e} = \ddot{q}_D - \ddot{q} \quad \longleftarrow \quad \ddot{q} = M^{-1}(\tau - N - \tau_D)$$
$$\ddot{e} = \underbrace{\ddot{q}_D - M^{-1}(\tau - N)}_{u \text{ control input}} + \underbrace{M^{-1}\tau_D}_{w \text{ disturbance}}$$

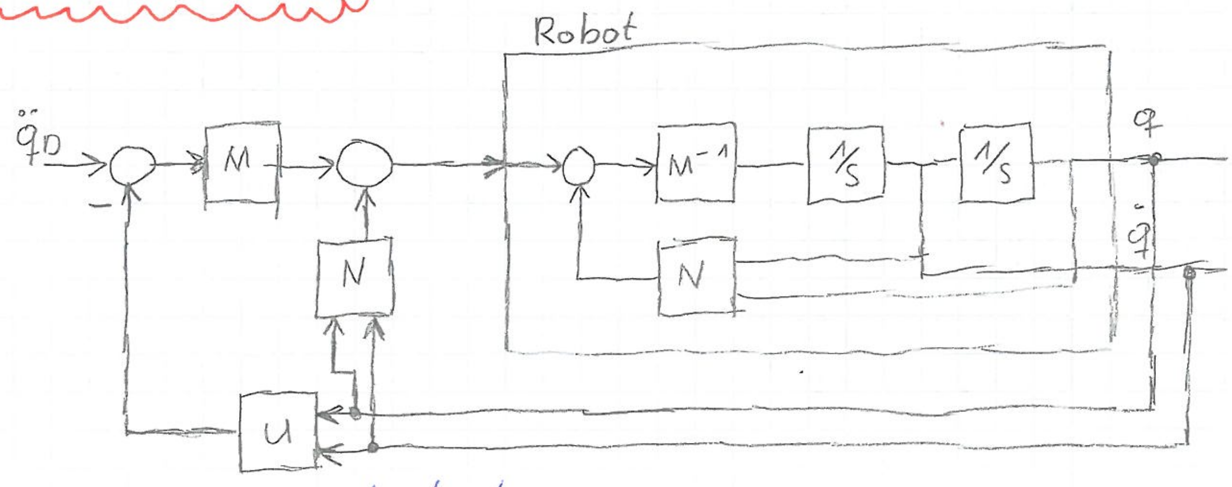
State space representation for error dynamics

$$e = \begin{pmatrix} e \\ \dot{e} \end{pmatrix} \qquad \begin{pmatrix} \dot{e} \\ \ddot{e} \end{pmatrix} = \begin{pmatrix} \emptyset & I \\ \emptyset & \emptyset \end{pmatrix} \begin{pmatrix} e \\ \dot{e} \end{pmatrix} + \begin{pmatrix} 0 \\ I \end{pmatrix} u + \begin{pmatrix} 0 \\ w \end{pmatrix}$$

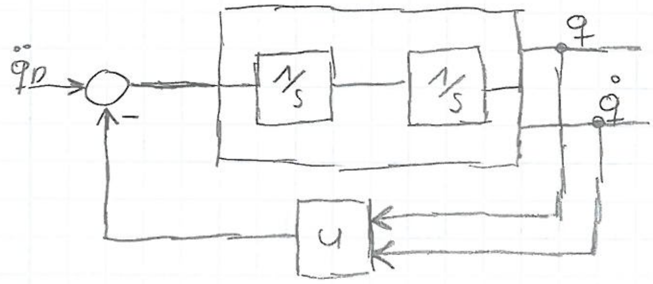
Error dynamics is linear system (double integrator) driven by the input and disturbance

From control input we derive CT - control law

$$\tau_c = M(\ddot{q}_D - u) + N$$



u - control law



Different possibilities

$u = -k_v \dot{e} - k_p e$

$\tau_c = M(\ddot{q}_D + k_v \dot{e} + k_p e) + N$

$$\begin{pmatrix} \dot{e} \\ \ddot{e} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -k_p & -k_v \end{pmatrix} \begin{pmatrix} e \\ \dot{e} \end{pmatrix} + \begin{pmatrix} 0 \\ w \end{pmatrix} \Rightarrow \ddot{e} = -k_p e - k_v \dot{e} + w$$

$$\ddot{e} + k_v \dot{e} + k_p e = w$$

$\Delta(s) = s^2 + k_v s + k_p$

$k_v, k_p > 0$

$s^2 + 2\xi\omega_n s + \omega_n^2$

12.12.15

CT (computed - torque) control law

- Group of controllers with similar structure
- The controller cancels non-linearities which reduces robot dynamics to double integrator problem
- condition: model needs to be known (quite) well

PD controller as $u = -k_v \dot{e} - k_p e$

error dynamics: $$\begin{pmatrix} \dot{e} \\ \ddot{e} \end{pmatrix} = \begin{pmatrix} 0 & I \\ -k_p & -k_v \end{pmatrix} \begin{pmatrix} e \\ \dot{e} \end{pmatrix} + \begin{pmatrix} 0 \\ w \end{pmatrix} \qquad k_p, k_v > 0$$

$$\tau_c = M\underbrace{(\ddot{q}_D + k_v \dot{e} + k_p e)}_{\text{estimate actual } \ddot{q}} + N$$

$\Delta(s) = s^2 + k_v s + k_p \overset{!}{=} s^2 + 2\xi\omega_n s + \omega_n^2$

results:

a) $1/\omega_n = 0{,}1 \quad \xi = 1$

b) $\xi > 1, \ \xi < 1$

PID outer loop design

$\dot{\varepsilon} = e \implies u = -k_v \dot{e} - k_p e - k_i \varepsilon$

integral time constant

CT - PID control law: $\tau_c = M(\ddot{q}_D + k_v \dot{e} + k_p e + k_i \varepsilon) + N$

Error dynamics:

$$\begin{pmatrix} \dot{\varepsilon} \\ \dot{e} \\ \ddot{e} \end{pmatrix} = \begin{pmatrix} \emptyset & I & \emptyset \\ \emptyset & \emptyset & I \\ -k_i & -k_p & -k_v \end{pmatrix} \begin{pmatrix} \varepsilon \\ e \\ \dot{e} \end{pmatrix} + \begin{pmatrix} \emptyset \\ \emptyset \\ W \end{pmatrix}$$

Stability of error dynamics for PID CT

closed loop char. polynom:

$\Delta(s) = Is^3 + k_i + k_p s + k_v s^2$

if we consider just one Joint (1 DoF) for ith joint

$\Delta^*(s) = s^3 + k_v s^2 + k_p s + k_i$

stability $\implies$ Routh Hurwitz

$$\begin{array}{ccc} 1 & k_p & 0 \\ k_v & k_i & 0 \\ \hline \frac{k_v k_p - k_i}{k_v} & 0 & \\ k_i & 0 & \end{array}$$

$\implies$

$k_v > 0$

$k_i > 0$

$k_v k_p > k_i$

$k_p > \frac{k_i}{k_v}$

CT like controllers

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

Here we suppose that dyn. model of system is 100% known

But that is never the case. So in reality we have

$$\tau_c = \hat{M}(\ddot{q}_D - u) + \hat{N}$$

estimated

designer choice

instead of estimating/using complete dynamic model (which is maybe even unknown) we use in CT-like controller only most influential part of dynamics

joint space	task space
$q = (q_1 \dots q_n)$	$x = (x, y, z, R_x, R_y, R_z)$
$\dot{q} = (\dot{q}_1 \dots \dot{q}_n)$	$\dot{x} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$

$$X = h(q)$$

$$\dot{x} = \frac{\partial h}{\partial q}\frac{dq}{dt} = \frac{\partial h}{\partial q}\dot{q} = J(q)\dot{q}$$

CT

$$\tau_c = M(\ddot{q}_D - u) + N$$

$$\tau_c = M(J^{-1}\ddot{x}_D - J^{-1}\dot{J}\dot{q}_D - u) + N$$

$$\tau_c = M(J^{-1}\ddot{x}_D - J^{-1}\dot{J}J^{-1}\dot{x}_D - u) + N$$

reference/desired trajectories

joint space	task space
$q_D, \dot{q}_D, \ddot{q}_D$	$x_D, \dot{x}_D, \ddot{x}_D$

$$\dot{x}_D = J \cdot \dot{q}_D \Rightarrow \dot{q}_D = J^{-1}\dot{x}_D$$

$$\ddot{x}_D = J\ddot{q}_D + \dot{J}\dot{q}_D$$

$$\hookrightarrow \ddot{q}_D = J^{-1}(\ddot{x}_D - \dot{J}\dot{q}_D)$$

Linear feedback u

PD: $u = -k_p(q_D - q) - k_v(\dot{q}_D - \dot{q})$ joint space

PD: $u = -k_p(x_D - x) - k_v(\dot{x}_D - \dot{x})$ task space

CT like controller with disturbance estimator

- decoupled joint control

$$\tau_c = M(q)(\ddot{q}_D + k_v\dot{e} + k_p e) + N = \underset{\uparrow \text{central}}{M(q)}\ \ddot{q}_{calc} + \underset{\uparrow \text{central}}{N(q,\dot{q})}$$

introduces only signals from current joint

$$M(q) = \begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1n} \\ \vdots & \ddots & & \vdots \\ M_{n1} & \cdots & & M_{nn} \end{pmatrix}$$

$$\tau_{ck} = M_{kk}(q)\ \ddot{q}_{calc} + N(q,\dot{q})$$

$\bar{M}_{kk}$ average/estimated inertia of joint

will be replaced by estimator $\hat{w}_k(\dot{q}_k, q_k)$

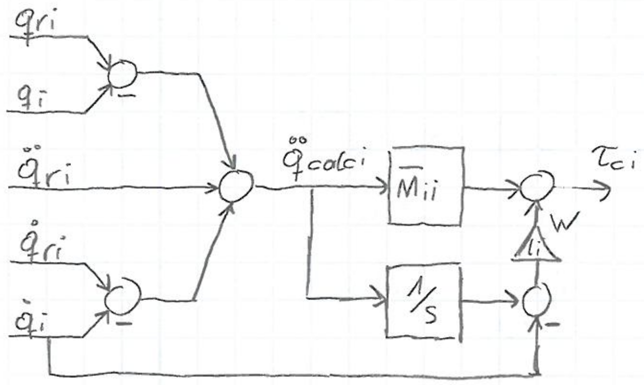
$$\tau_{ck} = \bar{M}_{kk}\ \ddot{q}_{calck} + \hat{w}_k(\dot{q}_k, q_k)$$

↳ completely decoupled

$$\hat{w}_k = l_k(\dot{q}_{calc} - \dot{q})$$

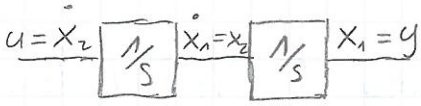
$$\dot{q}_{calc} = \int \ddot{q}_{calc}\, dt$$

16.01.12



Sliding mode control

concept for double integrator



SM is based on two conditions:

(C_1) – System motion on sliding manifold (line, surface) is described by equations of reduced order when comparing to the order of dynamic equation

2 int : 2^{nd} order
with SM we will reduce description of motion to 1 order
in SM 2 int will be described with equations of 1st order

(C_2) – System states once on sliding manifold (= in sliding mode) should move toward zero (toward origine of coordinate system)

$$\lim_{t\to\infty} x(t) = 0 \qquad x(t) \text{ states of the system}$$

In SMC design we first choose sliding manifold, which fulfills (C_1) & (C_2) for specific system and then derive necessary control law

Proposition: for double integrator suitable manifold would be

$$s = cx_1 + \dot{x}_1 \quad \text{with } s \text{ we denote sliding manifold}$$

Proof: $\dot{x} = Ax + bu$ $\qquad \dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$

$y = c^T x$ $\qquad y = (1 \quad 0)\, x$

(C_1) ✓

(C_2) $s = 0 = cx_1 + \dot{x}_1$ $\qquad \lim_{t\to\infty} x_1(t) = 0$

$\lim_{t\to\infty} x_2(t) = 0$

$0 = cx_1 + \dot{x}_1 \multimap sx_1 - x_{10} + cx_1 = 0$

$(s + c)\, x_1 = x_{10}$

$x_1 = \frac{x_{10}}{\boxed{s+c}} \multimap x_1(t) = x_{10}\, e^{-ct}$

for $c > 0$:

$\lim_{t \to \infty} x_1(t) = 0$

$x_2 = \dot{x}_1 = -c\, x_{10}\, e^{-ct}$

for $c > 0$

$\lim_{t \to \infty} x_2(t) = 0$

proposition: control law which will force the system's states (2 int) to the sliding manifold and keep them there is

$$u = \begin{cases} u^+ ; & s < 0 \\ u^- ; & s > 0 \end{cases}$$

respectively $u = -u_0 \,\mathrm{sign}(s)$

Description of nonlinear system in state space

$\dot{x} = f(x,t) + b(x,t)\, u$

sliding functions/manifolds are functions of states

$s = s(x)$

for robot states = errors we take error equation

Lyapunov design of control

$V(x) = \frac{1}{2} s^T s$

$\dot{V}(x) = \frac{1}{2} \dot{s}^T s + \frac{1}{2} s^T \dot{s} = s^T \dot{s}$

s we have

$\dot{s}$ needs to be calculated

$$\dot{s} = \frac{ds}{dt} = \frac{\partial s}{\partial x} \frac{dx}{dt} = \frac{\partial s}{\partial x} \dot{x} = \frac{\partial s}{\partial x} \left(f(x,t) + b(x,t)\, u \right)$$

$\dot{V}(x) = s^T \dot{s} < 0$

$\dot{V}(x) = s^T \frac{\partial s}{\partial x} \left(f(x,t) + b(x,t)\, u \right) < 0$

$\hookrightarrow u$

$$\begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} e + \dot{e}$$

23.01.16

Adaptive control

$\Rightarrow W(q, \dot{q}, \ddot{q})$ regression matrix, here we put everything that we know

$\Rightarrow \varphi$ parameter vector, here are all parameters that are not exactly known or are changing during the robot operation

When multiplying W and φ, we need to get original dynamic equations

$$\varphi = \begin{pmatrix} m_1 \\ m_2 \\ k_1 \\ k_2 \\ V_1 \\ V_2 \end{pmatrix} \qquad \begin{aligned} W_{22} &= (a_2^2 + a_1 a_2 \, c\theta_2)\ddot{\theta}_1 + a_2^2 \ddot{\theta}_2 + a_1 a_2 \dot{\theta}_1^2 \, s\theta_2 + g \, a_2 \, c(\theta_1 + \theta_2) \\ W_{24} &= \mathrm{sign}(\dot{\theta}_2) \\ W_{26} &= \dot{\theta}_2 \end{aligned}$$

$$W = \begin{pmatrix} \overline{\quad\quad\quad\quad} & '' & \overline{\quad\quad} \\ 0 \; W_{22} \; 0 & W_{24} & 0 \; W_{26} \end{pmatrix}$$

$$\tau_2 = W\varphi = W_{22} \cdot m_2 + W_{24} \, k_2 + V_2 \dot{\theta}_2$$
$$= \left[(a_2^2 + a_1 a_2 \, c\theta_2)\ddot{\theta}_1 + a_2^2 \ddot{\theta}_2 + a_1 a_2 \dot{\theta}_1^2 \, s\theta_2 + g \, a_2 \, c(\theta_1 + \theta_2)\right] m_2$$

CT-Like adaptive robot controller

control law

(1) $\tau_c = \hat{M}(\ddot{q}_r + k_v \dot{e} + k_p e) + \hat{V}_m \dot{q} + \hat{G} + \hat{\tau}_{fr}$

real robot dynamics

(2) $\tau = M\ddot{q} + V_m \dot{q} + G + \tau_{fr} \pm W(q, \dot{q}, \ddot{q})\,\varphi$

Derivation of error dynamics for parametrised form

$\ddot{e} = \ddot{q}_r - \ddot{q}$ acceleration error $\Rightarrow \ddot{q}_r = \ddot{e} + \ddot{q} \rightarrow (1)$

$$\tau_c = \hat{M}(\ddot{q} + \ddot{e} + k_v \dot{e} + k_p e) + \hat{V}_m \dot{q} + \hat{G} + \hat{\tau}_{fr} =$$
$$= \hat{M}(\ddot{e} + k_v \dot{e} + k_p e) + \underbrace{\hat{M}\ddot{q} + \hat{V}_m \dot{q} + \hat{G} + \hat{\tau}_{fr}}_{\text{estimated robot dynamics}} =$$

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

ideal situation τ_c = robot dynamics $= \tau$ (2)

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

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

$\tilde{\varphi}$ parameter estimation error

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

↳ state space form

$$\begin{pmatrix} \dot{e} \\ \ddot{e} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & I \\ -k_p & -k_v \end{pmatrix}}_{A} \begin{pmatrix} e \\ \dot{e} \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ I \end{pmatrix}}_{b} \hat{M}^{-1} W(q,\dot{q},\ddot{q})\tilde{\varphi}$$

$\dot{e} = Ae + b\hat{M}^{-1} W(q\dot{q}\ddot{q})\tilde{\varphi}$

$V = e^T P e + \tilde{\varphi}^T \Gamma^{-1}\tilde{\varphi}$ P p.d. & sym

$\Gamma = diag(\gamma_1, \dots \gamma_n) > 0$

$\dot{V} = \dot{e}^T P e + e^T P \dot{e} + 2\varphi^T \Gamma^{-1}\dot{\varphi} =$

$= e^T P (Ae + b\hat{M}^{-1} W\tilde{\varphi}) + (Ae + b\hat{M}^{-1} W\tilde{\varphi})^T P e + 2\tilde{\varphi}^T \Gamma^{-1}\dot{\tilde{\varphi}}$

$= \underbrace{e^T PAe + e^T A^T P e}_{PA + A^T P = -Q} + 2\tilde{\varphi}^T(\Gamma^{-1}\dot{\tilde{\varphi}} + W^T \hat{M}^{-1} bPe)$

$= -e^T Q e + \underbrace{2\tilde{\varphi}^T(\Gamma^{-1}\dot{\tilde{\varphi}} + W^T\hat{M}^{-1} bPe)}_{\text{needs to be } < 0}$

$\dot{\tilde{\varphi}} = -\Gamma W^T \hat{M}^{-1} bPe$

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

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

$\approx \emptyset$

CT-Like adaptive inertia related approach

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

new: $V = f(M)$

$$V = \frac{1}{2} r^T M r + \frac{1}{2} \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} > 0$$

$$r = \Lambda e + \dot{e}, \quad \Lambda = \mathrm{diag}(\lambda_1, \dots \lambda_n); \ \lambda_i > 0$$

We are searching: $\tau_c = ?$ control law $\dot{\hat{\varphi}} = ?$

derivative of Lyapunov function

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

$$= r^T M \dot{r} + \frac{1}{2} r^T \dot{M} r + \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \overset{!}{<} 0$$

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

$$r = \Lambda e + \dot{e} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} + \begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \end{pmatrix} \qquad \hat{\varphi} = \begin{pmatrix} \hat{m}_1 \\ \hat{m}_2 \end{pmatrix}$$

$$\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \begin{pmatrix} \hat{m}_1 \\ \hat{m}_2 \end{pmatrix} + \begin{pmatrix} k_{v1} & 0 \\ 0 & k_{v2} \end{pmatrix} \begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \end{pmatrix} + \begin{pmatrix} k_{v1} & 0 \\ 0 & K_{v2} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$$

$$= \begin{pmatrix} y_{11} \hat{m}_1 + y_{12} \hat{m}_2 \\ y_{21} \hat{m}_1 + y_{22} \hat{m}_2 \end{pmatrix} + \begin{pmatrix} k_{v1} \dot{e}_1 \\ k_{v2} \dot{e}_2 \end{pmatrix} + \begin{pmatrix} k_{v1} \lambda_1 e_1 \\ k_{v2} \lambda_2 e_2 \end{pmatrix}$$

$$\dot{\hat{\varphi}}_a = \begin{pmatrix} \dot{\hat{m}}_1 \\ \dot{\hat{m}}_2 \end{pmatrix} = \begin{pmatrix} \dots \\ \vdots \\ \dots \end{pmatrix}$$

$$\begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \begin{pmatrix} \hat{m}_1 \\ \hat{m}_2 \end{pmatrix} = \begin{pmatrix} \hat{M}_{11} & \hat{M}_{12} \\ \hat{M}_{21} & \hat{M}_{22} \end{pmatrix} \left(\begin{pmatrix} \ddot{q}_{r1} \\ \ddot{q}_{r2} \end{pmatrix} + \begin{pmatrix} \lambda_1 \dot{e}_1 \\ \lambda_2 \dot{e}_2 \end{pmatrix} \right) + \begin{pmatrix} \hat{V}_{m1} \\ \hat{V}_{m2} \end{pmatrix} \cdot \left(\begin{pmatrix} \dot{q}_{r1} \\ \dot{q}_{r2} \end{pmatrix} + \begin{pmatrix} \lambda_1 e_1 \\ \lambda_2 e_2 \end{pmatrix} \right) + \begin{pmatrix} \hat{G}_1 \\ \hat{G}_2 \end{pmatrix}$$

$$y_{11} \hat{m}_1 \ {}_{+ y_{12} \hat{m}_2} = \hat{M}_{11} (\ddot{q}_{r1} + \lambda_1 \dot{e}_1) + \hat{M}_{12} (\ddot{q}_{r2} + \lambda_2 \dot{e}_2) + \hat{V}_{m11} (\dot{q}_{r1} + \lambda_1 e_1) + \hat{V}_{m12} (\dot{q}_{r2} + \lambda_2 e_2) + \hat{G}_1$$

$$\hat{M}_{11} = a_1^2 \hat{m}_1 + a_1^2 \hat{m}_2 + a_2^2 \hat{m}_2 + 2 a_1 a_2 m_2 c q_2$$

$$\hat{M}_{12} = a_2 \hat{m}_2 (a_2 + a_1 c q_1)$$

$$\hat{V}_{m11} = -2 a_1 a_2 \hat{m}_2 \dot{q}_2 s q_2$$

$$\hat{V}_{m12} = - a_1 a_2 \hat{m}_2 \dot{q}_2 s q_2$$

$$\hat{G}_1 = g \hat{m}_2 (a_2 \cos(q_1 + q_2) + a_1 c q_1) + a_1 g \hat{m}_1 c q_1$$