# COMPUTED TORQUE CONTROL

Andreja Rojko

# OUTLINE

- Introduction
- Inner Feedforward Loop Design
- PD Outer-Loop Design
- PID Outer-Loop Design
- Computed Torque Like Controllers
  - PD + gravity Control
  - Classical joint control
    - PD Control
    - PID Control
- Cartesian computed torque

# Introduction

- Computed torque (CT) controllers are a class of model based controllers. They work well when robot model is fairly well known.
- CT can be considered as a special application of feedback linearization of nonlinear systems.
- CT-like controllers can be found in robust, adaptive, intelligent control…
- One way to classify robot controllers is to divide them in ‘CT-like’ and ‘non CT-like’.

# Introduction

- CT controller schemes are based on decomposition to:
  - o Inner feedforward control component
  - o Outer feedback linear control loop
- By simplification of CT controllers an independent PID joint controller can be obtained (bridge between complex non-linear control and linear control).

# Introduction

Assumptions made in this chapter:

- Robot moves free in space without a contact to the environment.
- The robot is well know rigid system (know dynamic model).
- Desired joint space motion trajectory is known.
- Positions and velocities of all robot joints are measured.
- Drives’ control is implemented and it is efficient enough that drives’ dynamics can be neglected.

# Inner Feedforward Loop design

Robot's dynamics:

$$\tau(t) = M(q)\ddot{q} + V(q,\dot{q}) + F_v\dot{q} + F_d(\dot{q}) + G(q) + \tau_D =$$

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

$$= M(q)\ddot{q} + N(q,\dot{q}) + \tau_D$$

$q(t) \in \Re^n$ joint variables

$\tau(t)$ control torque, $\Re^n$

$\tau_D(t)$ disturbance torque (unknown dynamics)

$M(q)$ inertia matrix, $\Re^{nxn}$

$V(q,\dot{q})$ Coriolis, centripetal torque vector, $\Re^n$

$G(q)$ gravity torque vector, $\Re^n$

$F_v\dot{q} + F_d(\dot{q})$ friction torque vector, $\Re^n$

# Inner Feedforward Loop design

Desired - reference trajectory is selected as:

$$q_D(t), \dot{q}_D(t), \ddot{q}_D(t)$$

Tracking errors are defined as:

$$e(t) = q_D(t) - q(t),$$
$$\dot{e}(t) = \dot{q}_D(t) - \dot{q}(t),$$
$$\ddot{e}(t) = \ddot{q}_D(t) - \ddot{q}(t)$$

Desired joint space motion trajectory needs to be known for CT!

By taking last equation and substituting $\ddot{q}(t)$ from robot dynamics ($\tau(t) = M(q)\ddot{q} + N(q,\dot{q}) + \tau_D$) following is obtained:

$$\ddot{e} = \ddot{q}_D + M^{-1}(N + \tau_D - \tau)$$

# Inner Feedforward Loop design

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

For defined control input: $u = \ddot{q}_D + M^{-1}(N - \tau)$ *

disturbance function: $w = M^{-1}\tau_D$

And system state $x(t) \in R^{2n}$ is defined as: $x(t) = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$

the tracking error dynamics is:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} w$$

This is a linear error system which consists of pairs of double integrators driven by control input $u(t)$ and disturbance $\mathrm{w}(t)$.

Feedback linearization * could be inverted as:

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

**COMPUTED TORQUE CONTROL LAW**

# Inner Feedforward Loop design

**IMPORTANT**

- For selected control $u(t)$ that stabilizes error dynamics, so that $e(t) \to 0$ the nonlinear control input $\tau_c$

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

will cause that the robot will follow desired trajectory.

# Inner Feedforward Loop design

The control $\tau_c = M(\ddot{q}_D - u) + N$ consist of :

- Inner nonlinear feedforward component for compensation of nonlinearities.
- Outer feedback control signal $u(t)$, which will depend on $q(t)$ and $\dot{q}(t)$. This is linear control law for stabilizing the trajectory error to zero.

- CT control is based on inversion of dynamics.

**Different choices for $u(t)$ and variations of CT control will be described.**

# Inner Feedforward Loop design

Draw CT control scheme $\tau_c = M(q)(\ddot{q}_D - u) + N(q,\dot{q})$, showing inner nonlinear feedforward component and outer feedback control loop.

# Inner Feedforward Loop design

**Issues:**

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

- Changing payload causes a change of robot dynamic coefficients. Then we need robust, adaptive control ..
- Numeric values of the robot dynamic coefficients are needed but robot manufacturers don't provide them (parametric uncertainty)
  - o Estimates can be found with CAD tools, Catia, Solidworks…
- Friction coefficients are (slowly) varying over time and cannot be calculated. Further it is hard to model friction (structure uncertainty).

Identification experiments are usually needed.

# OUTLINE

- Introduction
- Inner Feedforward Loop Design
- **PD Outer-Loop Design**
- PID Outer-Loop Design
- Computed Torque Like Controllers
  - PD + gravity Control
  - Classical joint control
    - PD Control
    - PID Control
  - Cartesian Computed Torque Control

# PD Outer Loop Design

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

With the choice of control $u(t) = -K_v\dot{e} - K_p e$, a proportional + derivative (PD) feedback gives:

$$\tau_c(t) = M(q)\big(\ddot{q}_D + K_v\dot{e} + K_p e\big) + N(q,\dot{q})$$

Closed loop error dynamics is:

$$\ddot{e} + K_v\dot{e} + K_p e = w$$

Or in state-space:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} w$$

Closed loop characteristic polynomial is:

$$\Delta_c(s) = \left|s^2 + K_v s + K_p\right|$$

# PD Outer Loop Design

$\ddot{q}_{calc}$ is also called calculated acceleration

$$\tau_c = M(q)\left(\ddot{q}_D + K_v\dot{e} + K_p e\right) + N(q,\dot{q}) =$$
$$= M(q)\ddot{q}_{calc} + N(q,\dot{q})$$

$K_v$ and $K_p$ are *nxn* matrices:

$K_v = diag\left(k_{v,i}\right)$, $K_p = diag\left(k_{p,i}\right)$

And closed loop characteristic polynomial:

$$\Delta_c(s) = \prod_{i=1}^{n}\left|s^2 + K_{v,i}s + K_{p,i}\right|$$

- As long as $K_{v,i}$ and $K_{p,i}$ are positive the error system is asymptotically stable $\lim_{t\to\infty} e(t) = 0$.
- $Reality$: $e(t)$ is bounded as long as disturbance $w(t)$ is bounded.

# PD Outer Loop Design

Closed loop characteristic polynomial:

$$\Delta_c(s) = \prod_{i=1}^{n} \left| s^2 + K_{v,i} s + K_{p,i} \right|$$

Standard second order characteristic polynomial:

$$p(s) = s^2 + 2\xi\omega_n s + {\omega_n}^2$$

Desired closed loop dynamics for *i*-th robot joint can be achieved by selecting gains as:

$$K_{p,i} = {\omega_{n,i}}^2, \; K_{v,i} = 2\xi\omega_{n,i}$$

Usually critical damping is chosen (why?) which gives

$K_{v,i} = 2\sqrt{K_{p,i}}$ and $K_{p,i} = {K_{v,i}}^2/4$

Gains should be chosen carefully.

# PD Outer Loop Design

- $\omega_n$ should be large for a fast response.
- However joint flexibility sets upper limit. For a joint with moment of inertia $J$ and joint stiffness $k_r$ is resonant mode $\omega_r = \sqrt{k_r/_J}$ . Then it should be $\omega_n < \omega_r/_2$.
- Another upper bound of $\omega_n$ is set by actuator $\tau_{max}$.
- For critical damping it can be shown that position error decreases with $K_{p,i}$ and velocity error with $K_{v,i}$

(Frank L. Lewis, Darren M. Dawson, Chaouki T.Abdallah, Robot manipulator control).

**REPORT REQUIRED**

# Exercise: CT-PD controller

A 2 DOF planar elbow robot has link masses *1 kg* and lengths *1 m*. The desired trajectory is given with:

$$\theta_{1,d} = k_1 sin(2\pi t/T)$$
$$\theta_{2,d} = k_2 cos(2\pi t/T)$$

Where $k_{1,2} = 0.1\, rad$, $T = 2s$.

- Design computed torque controller with PD outer loop so that time constant $(1/\omega_n)$ of closed loop system is *0.1 s*.
- Draw the control scheme.
- By simulation in MATLAB/Simulink check the controller's performance (tracking error) and torque inputs).
- Repeat the simulation with over-damping and under-damping.
- Comment the results.

**REPORT REQUIRED**

# Exercise: CT-PD controller

- Add the disturbance *1Nm* to each robot joint. By simulation check the controller's performance. Comment the results.
- Additional: Calculate controller with sample time $T_s = 1\ ms$ and model with variable step integration method. In this way more realistic simulation results are obtained (why?). Explore also the effect of higher sampling time (5 ms, 20 ms. 0.1s) to the controller's performance.

# OUTLINE

- Introduction
- Inner Feedforward Loop Design
- PD Outer-Loop Design
- **PID Outer-Loop Design**
- Computed Torque Like Controllers
  - PD + gravity Control
  - Classical joint control
    - PD Control
    - PID Control
- Cartesian Computed Torque Control

# PID Outer Loop Design

PD controller is very effective if the model is known and there is no disturbance $\tau_d$. When $\tau_d \neq 0$ PD controller results in steady-state error. Therefore an integrator is added in feedback loop -> PID computed-torque controller.

$$\dot{\varepsilon} = e$$

$$u = -K_v \dot{e} - K_p e - K_i \varepsilon$$

The CT PID control law is:

$$\tau_c = M(q)\left(\ddot{q}_D + K_v \dot{e} + K_p e + K_i \varepsilon\right) + N(q, \dot{q})$$

# PID Outer Loop Design

The states here are:

$$x = \left[\varepsilon^T, e^T, \dot{e}^T\right]^T \epsilon R^{3n}$$

Error dynamics:

$$\frac{d}{dt}\begin{bmatrix}\varepsilon \\ e \\ \dot{e}\end{bmatrix} = \begin{bmatrix}0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\varepsilon \\ e \\ \dot{e}\end{bmatrix} + \begin{bmatrix}0 \\ 0 \\ I\end{bmatrix}u + \begin{bmatrix}0 \\ 0 \\ I\end{bmatrix}w$$

Closed-loop error dynamic:

$$\frac{d}{dt}\begin{bmatrix}\varepsilon \\ e \\ \dot{e}\end{bmatrix} = \begin{bmatrix}0 & I & 0 \\ 0 & 0 & I \\ -K_i & -K_p & -K_v\end{bmatrix}\begin{bmatrix}\varepsilon \\ e \\ \dot{e}\end{bmatrix} + \begin{bmatrix}0 \\ 0 \\ I\end{bmatrix}w$$

Closed loop characteristic polynomial is:

$$\Delta_c(s) = \left|s^3 I + K_v s^2 + K_p s + K_i\right|$$

# PID Outer Loop Design

Control gains are selected diagonal:

$$K_v = diag\{k_{v_i}\}$$
$$K_p = diag\{k_{p_i}\}$$
$$K_i = diag\{k_{i_i}\}$$

Therefore closed loop characteristic polynomial is:

$$\Delta_c(s) = \prod_{1}^{n}\left(s^3 I + k_{v_i}s^2 + k_{p_i}s + k_{i_i}\right)$$

Closed loop stability conditions can be calculated by Routh-Hurwitz stability criterion.

REPORT REQUIRED

# PID Outer Loop Design

Derive stability condition for CT controller with PID outer loop.

# PID Outer Loop Design

Actuator saturation. This problem is not so common at CT PD controller, however it is almost guaranteed with CT PID controller due to integrator windup problem.

- This is the case when the integrator's input stays positive for a long time and continues to integrate upwards. Then the product $k_i \varepsilon(t)$ can increase so that $k_i \varepsilon(t) \gg \tau_{max}$.
- Consequently it can take very long until $k_i \varepsilon(t)$ decreases below $\tau_{max}$. Until then the control input is (incorrect) $\tau_{max}$.

**REPORT REQUIRED**

# Exercise: CT PID controller

Add integral part to the controller ($k_i = 500$) and repeat test with unknown dynamic disturbance of *1 Nm* in both joints.

- By simulation check the performance of the controller.
- Compare the results to the results of CT PD controller.
- Check also control torque for both links.
- Comment the results.
- Repeat the test with unknown payload of *0.5 kg* and comment the results.

# OUTLINE

- Introduction
- Inner Feedforward Loop Design
- PD Outer-Loop Design
- PID Outer-Loop Design
- **Computed Torque Like Controllers**
  - PD + gravity control
  - Classical joint control
    - PD Control
    - PID Control
- Cartesian Computed Torque Control

# CT like controllers

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

CT like controllers can be obtained by modification:

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

Carets ^ denotes design choice.

If $M(q)$and $N(q, \dot{q})$ are not known then their estimates $\widehat{M}$ and $\widehat{N}$ can be used instead -> approximate CT.

Even if $\widehat{M} \neq M(q)$ and $\widehat{N} \neq N(q, \dot{q})$ the performance of CT like controllers can be quite good (robust) if sufficiently large outer-loop gains are used (or high gear ratio).

# CT like controllers

Derivation of error dynamics with approximate control law:

$$\tau = \tau_C$$

$$M(q)\ddot{q} + N(q,\dot{q}) + \tau_D = \widehat{M}(\ddot{q_D} - u) + \widehat{N}$$

With some equation manipulation following is obtained:

$$\ddot{e} = u - \Delta u + d$$

where

$$\Delta = M^{-1}\left(M - \widehat{M}\right) = I - M^{-1}\widehat{M};$$

$$d(t) = M^{-1}\tau_d + \Delta\ddot{q}_d(t) + M^{-1}\left(N - \widehat{N}\right)$$

For exact CT: $\Delta = 0, \delta = 0$.

Otherwise the error system depends on reference acceleration and nonlinear modelling error $\delta$ and never goes to zero.

# CT like controllers

For **PD outer loop** $u(t) = -K_v\dot{e} - K_p e$ following is obtained for error dynamics $\ddot{e} = u - \Delta u + d$:

$$\ddot{e} + K_v\dot{e} + K_p e = \Delta\left(K_v\dot{e} + K_p e\right) + d(t)$$

It is not clear how such system will behave, even if the gains are selected so that left side is stable.

Problems are:

- $d = M^{-1}\tau_d + \Delta\ddot{q}_d(t) + M^{-1}\left(N - \widehat{N}\right)$
- $\Delta\left(K_v\dot{e} + K_p e\right) = M^{-1}\left(M - \widehat{M}\right)\left(K_v\dot{e} + K_p e\right)$

# OUTLINE

- Introduction
- Inner Feedforward Loop Design
- PD Outer-Loop Design
- PID Outer-Loop Design
- Computed Torque Like Controllers
  - **PD + gravity control**
  - Classical joint control
    - PD Control
    - PID Control
- Cartesian Computed Torque Control

# PD + gravity controller

In $\tau_C(t) = \widehat{M}(\ddot{q}_D - u) + \widehat{N}$: $\widehat{M} = I, \widehat{N} = G(q) - \ddot{q}_d$ in and PD feedback $u(t) = -K_v\dot{e} - K_p e$ gives:
$\tau_c = K_v\dot{e} + K_p e + G(q)$

- o Good performance at set point when $\dot{q}_d = 0$ (why?).
- o For $\dot{q}_d \neq 0$ bounded tracking errors are guaranteed.
- o With higher PD gains performance is improved.

Much simpler implementation than conventional CT

- Those statement can be proven by using Lyapunov function $V = \frac{1}{2}\left(\dot{q}^T M\dot{q} + e^T K_p e\right)$ and skew symetry of $\dot{M}(q) - 2V_m(q, \dot{q})$.
- Asymptotic stability can be proven by using Barbalat's lemma and LaSalle's extension.

# PD + gravity controller

Draw a control scheme.

REPORT REQUIRED

# Exercise: CT PD + gravity controller

- By simulation check the performance of CT PD-gravity controller.
- Assume identical PD gains for both links and critical damping.
- Observe tracking error and control torque for different $\omega_n$.
- Comment the results.

# Classical joint control

For CT like controllers class $\tau_C(t) = \widehat{M}(\ddot{q}_D - u) + \widehat{N}$ it is selected: $\widehat{M} = I$, $\widehat{N} = -\ddot{q}_d$

Following control is obtained:

$$\tau_C = -u$$

- If $u$ depends only on joint variables, this control yields *n* decoupled individual joint controllers, that is **independent joint control=classical joint control**.
- Implementation and computation is simple.
- Quite often (high reduction) suitable also for trajectory tracking not just for set-point control.

# Classical joint control, PD control

For $\tau_C = -u$ and $u = -k_v\dot{e} - k_p e$

$e(t)$ is tracking error of the motor angle:

$e_i(t) = \theta_{d_i} - \theta_i$; $q_{Mi} = n_i q_i$, $q_M = Nq$,

Linear time-inv. model of robot joint with actuators

$$\left(M_M + \frac{1}{N^2}M\right)\ddot{q} + \frac{1}{N}V\left(\frac{q_M}{N}, \frac{\dot{q}_M}{N}\right) + \left(F_M(\dot{q}_M) + \frac{1}{N}F\left(\frac{q_M}{N}\right)\right) + \frac{1}{N}G\left(\frac{q_M}{N}\right) = \tau_M$$

is simplified to model of robot joint: $J\ddot{\theta} + B\dot{\theta} = u - \frac{1}{N}\tau_D$

Closed loop characteristic polynomial for set point can be then calculated as:

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

- Gains can be selected to obtain desired natural frequency and damping ration.

# Classical joint control, PD control

Steady state error for set point:

- Only contribution to steady state error is gravitation.
- It can be concluded $|\tau_D| < g_b$
  - o $g_b$ is max. value of gravity vector for given robot arm
- Final value of set-point error for robot joint is:

$$e_{set-point} < g_b / N k_p$$

  - o High reduction and large position gain are beneficial.

# Classical joint control, PID control

For $\tau_C = -u$

$$u = k_v\dot{e} + k_p e + k_i\varepsilon;\ \dot{\varepsilon} = e$$

Closed loop characteristic polynomial is :

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

- Steady state error for set point is zero-
- By using Routh-Hurwitz stability criterion the stability condition can be obtained (verify!):

$$k_i < (B + k_v)\frac{k_p}{J}$$

# Classical joint control, PID control

Scheme of PD & PID independent joint control

REPORT REQUIRED

# Exercise: Classical joint control

By simulation check the performance of classical joint control for two link planar robot.

- Design PD joint controller. Choose PD gains for critical damping.
- Test performance for high and low gains ($\omega_n = 10\frac{rad}{s}, \omega_n = 25\frac{rad}{s}, \omega_n = 50\frac{rad}{s}$).

REPORT REQUIRED

# Exercise: Classical joint control

- Design PID joint controller.
  - By using Routh-Hurwitz stability criterion verify stability conditions for parameters.
  - Test performance for high and low gains ($\omega_n = 50\frac{rad}{s}, k_i = 1000$).
  - Suppose torque limits $\pm 35\ Nm$. Comment the results.

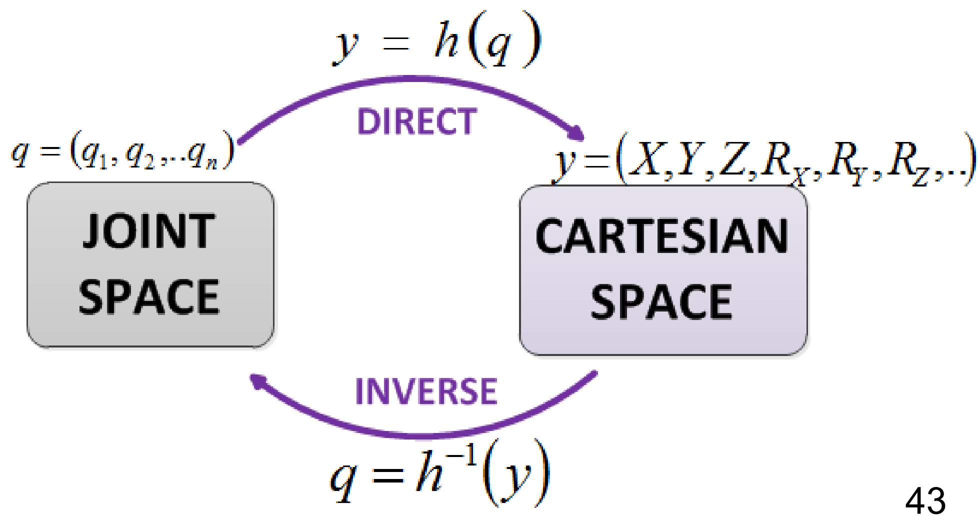
# OUTLINE

- Introduction
- Inner Feedforward Loop Design
- PD Outer-Loop Design
- PID Outer-Loop Design
- Computed Torque Like Controllers
  - PD + gravity Control
  - Classical joint control
    - PD Control
    - PID Control
- **Cartesian Computed Torque Control**

# Cartesian CT control

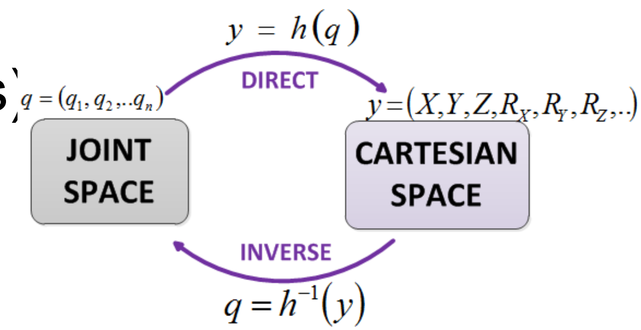
Usually robot control tasks are given in Cartesian coordinates. For Cartesian control there are two possibilities:

- To convert Cartesian desired trajectory $y_d(t)$ to a joint space trajectory $q_d(t)$ by using inverse kinematic model. Then joint space controllers can be used.
- To use Cartesian control.



# Cartesian CT control

- Cartesian error: $e_y(t) = y_d(t) - y(t) = \begin{bmatrix} e_p \\ e_o \end{bmatrix}$
  - o $y_d(t)$ desired Cart. trajectory
  - o $y(t)$ actual Cart. trajectory (end-effector's)
  - o $e_p$ position error, $e_o$ orientation error




- $y = h(q)$ and Jacobian $J = \frac{\partial h}{\partial q}$ are known
- Original CT: $\tau_c = M(\ddot{q}_d - u) + N$
- Cartesian CT: $\tau_c = M\left(J^{-1}\ddot{y}_d - J^{-1}\dot{J}\dot{q}_D - u\right) + N$
  - o Error system $\ddot{e}_y = u + w$
  - o Disturbance $w = JM^{-1}\tau_d$
  - o $J^{-1}$, problem of existence
  - o Outer loop can be selected by using any of joint-space approaches.

# Summary

- The power of the CT control law(s) is that it converts a nonlinear dynamical system into a linear one, allowing the use of any of a number of linear control synthesis tools. This is an example of a more general technique known as feedback linearization, where a nonlinear system is rendered linear via full-state nonlinear feedback.
- CT control schemes perform well when the robot dynamic equations are known but it is also otherwise quite robust.
- CT controllers can be combined with soft computing methods. Artificial neural networks or fuzzy logic systems can replace unknown part of the dynamic model ('intelligent controllers' are obtained in this way).