The use of interactive applications in control education

S. Dormido
sdormido@dia.uned.es
Dpto de Informática y Automática, UNED

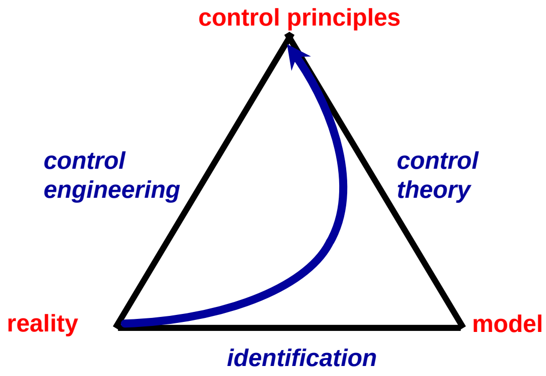
Czech Technical University in Prague, Prague 13th march 2007

Contents

Contents

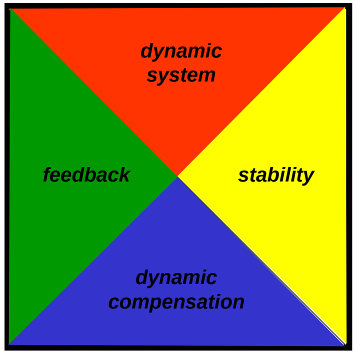
1. Introduction (1/8)

The gap theory–practice in control education

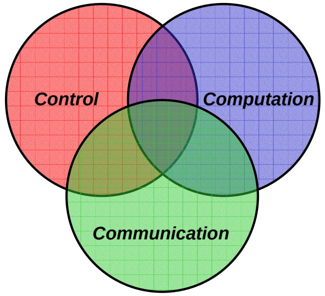


1. Introducción (3/8)

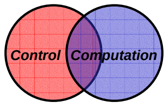
Major concepts in control education



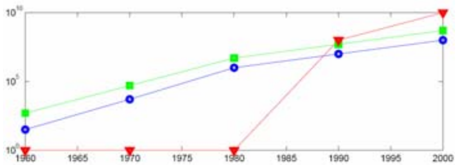
1. Introducción (4/8)



1. Introduction (5/8)



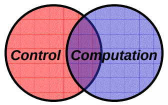
Growth of computers



- *General purpose computers and PC's*
- *Process control computers*
- *Embedded computers*

Source: K. J. Åström *"Challenges in Control Education"*, ACE'06

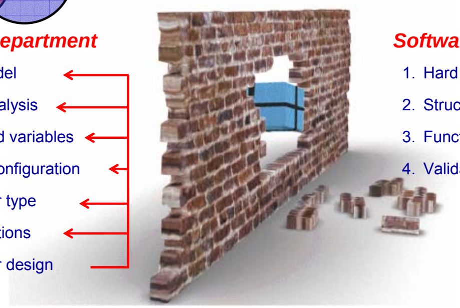
1. Introduction (6/8)



Control systems implementation

Control Department

1. Plant model
2. Model analysis
3. Controlled variables
4. Control configuration
5. Controller type
6. Specifications
7. Controller design

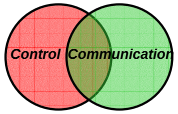


Software Department

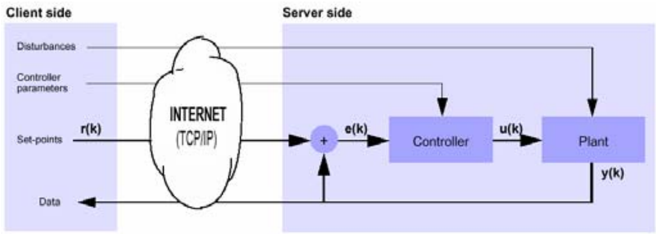
1. Hard + Soft design
2. Structural test
3. Functional test
4. Validation

The Brick Wall Metaphore

1. Introduction (7/8)



Network Control Systems

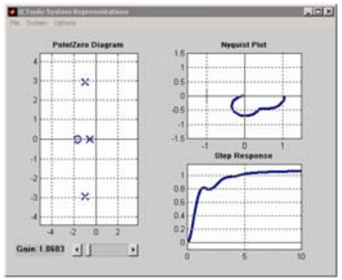


Closing the control loop at the server side

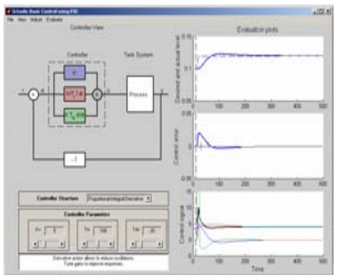
Contents

2. Interactivity in control education (1/8)

- ***The concept of dynamic picture***
- ***"Think small and simple"***
- ***Examples in IcTools***



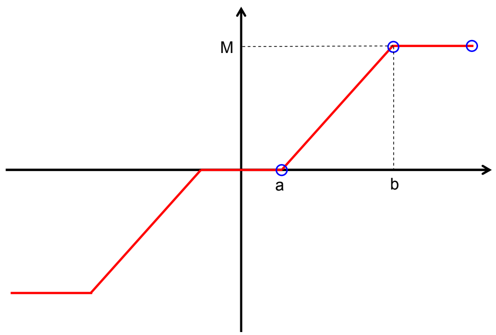
Analysis of Linear Systems



Basic Control using PID

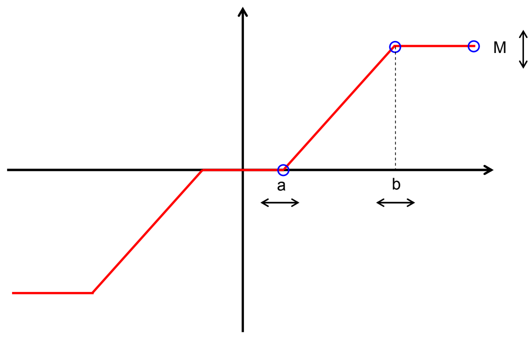
2. Interactivity in control education (2/8)

Non linearity: saturation + dead zone (static)



2. Interactivity in control education (3/8)

Non linearity: saturation + dead zone (dynamic)

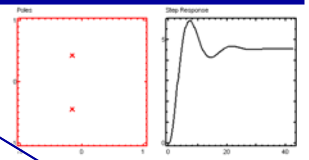


S. Dormido | Interactive Applications in Control

2. Interactivity in control education (4/8)

A simple control interactive application

```
variable den
init den = initFun
Figure "Poles"
  draw drawPoles(den)
  mousedrag den = dragPoles(den, _z0, _z1)
Figure "Step"
  draw drawStep(den)
functions
{@
function den = initFun
  den = poly([-0.5+0.3j, -0.5-0.3j]);
  subplots('drawPoles\tdrawStep');
function drawPoles(den)
  scale('equal', [-1,1,-1,1]);
  plotroots(den, 'r', 1);
function den = dragPoles(den, _z0, _z1)
  den = movezero(den, _z0, _z1);
function drawStep(den)
  step(1, den);
@}
```



Declarative part

Operacional part (Matlab code)

S. Dormido | Interactive Applications in Control

2. Interactivity in control education (5/8)

1. Study the plant to be controlled and obtain initial information about the control objectives.
2. Model the plant and simplify the model, if necessary.
3. Analyze the resulting model; determine its properties.
4. Decide which variables are to be controlled (controlled outputs).
5. Select the control configuration.
6. Decide on the type of controller to be used.
7. Decide on performance specifications, based on the overall control objectives.
8. Design a controller.
9. Analyze and if the specifications are not satisfied modify it or the type of controller.
10. Simulate the resulting controlled system, either on a computer or pilot plant.
11. Repeat from step 2, if necessary.
12. Choose hardware and software and implement the controller.
13. Test and validate the control system, and tune the controller on line, if necessary.

Steps in control system design

S. Dormido | Interactive Applications in Control

2. Interactivity in control education (5/8)

1. Study the plant to be controlled and obtain initial information about the control objectives.
2. Model the plant and simplify the model, if necessary.
3. Analyze the resulting model; determine its properties.
4. Decide which variables are to be controlled (controlled outputs).
5. Select the control configuration.
6. Decide on the type of controller to be used.
7. Decide on performance specifications, based on the overall control objectives.
8. Design a controller.
9. Analyze and if the specifications are not satisfied modify it or the type of controller.
10. Simulate the resulting controlled system, either on a computer or pilot plant.
11. Repeat from step 2, if necessary.
12. Choose hardware and software and implement the controller.
13. Test and validate the control system, and tune the controller on line, if necessary.

Steps in control system design

S. Dormido | Interactive Applications in Control

2. Interactivity in control education (5/8)

1. Study the plant to be controlled and obtain initial information about the control objectives.
2. Model the plant and simplify the model, if necessary.
3. Analyze the resulting model; determine its properties.
4. Decide which variables are to be controlled (controlled outputs).
5. Select the control configuration.
6. Decide on the type of controller to be used.
7. Decide on performance specifications, based on the overall control objectives.
8. Design a controller.
9. Analyze and if the specifications are not satisfied modify it or the type of controller.
10. Simulate the resulting controlled system, either on a computer or pilot plant.
11. Repeat from step 2, if necessary.
12. Choose hardware and software and implement the controller.
13. Test and validate the control system, and tune the controller on line, if necessary.

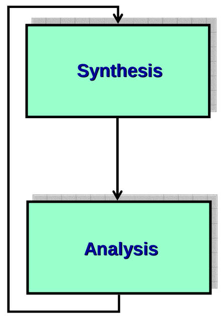
Steps in control system design

2. Interactivity in control education (6/8)

Non-interactive approach

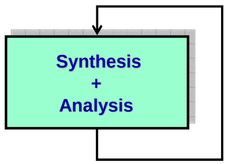
1st step: To calculate the unknown parameters according to a set of variables of design.

2th step: Evaluation of the performance and comparison with the original specifications.



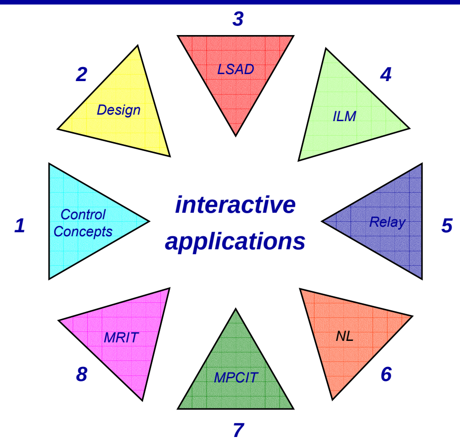
2. Interactivity in control education (7/8)

Interactive approach



- *The modification of the parameters produces an update of the graphical*
- *The design process is completely dynamic.*
- *Students notice the level of the change of the performance in relation with the elements that they are handling.*
- *This interactivity allows to identify more easily the compromises that it is possible to reach.*

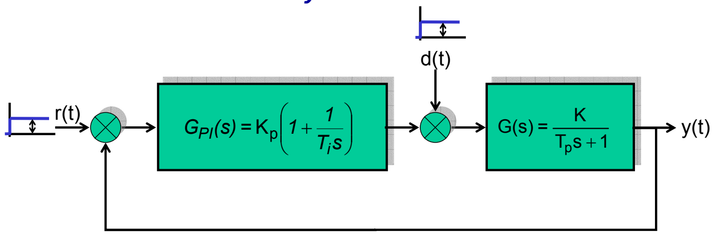
2. Interactivity in control education (8/8)



Contents

3. An introductory example

PI control of a first order system



Characteristic equation

$$s^2 + \frac{1+K_pK}{T_p}s + \frac{K_pK}{T_pT_i} = 0$$

Design characteristic equation

$$s^2 + 2\zeta_0\omega_0 s + \omega_0^2 = 0$$

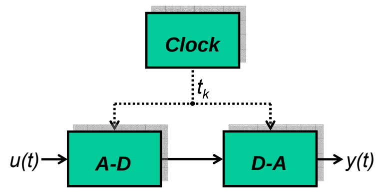
$$\Rightarrow \quad K = \frac{2\zeta_0\omega_0T_p - 1}{K_p}, \qquad T_i = \frac{2\zeta_0\omega_0T_p - 1}{\omega_0^2T_p}$$

Contents

4. Control concepts (1/10)

a) Aliasing phenomenon

A striking observation: *Sampling creates signals with new frequencies. This is clearly an evidence that we must understand in order to deal with computer-controlled systems.*



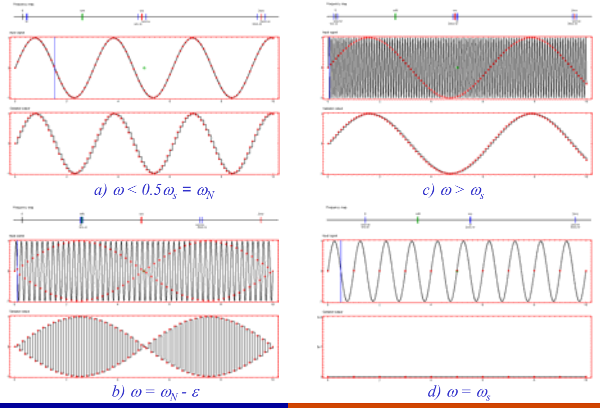
Sampling of a signal with frequency ω creates signal components with frequencies

$$\omega_{sampled} = n\omega_s \pm \omega$$

$\omega_s = 2\pi/h$ is the sampling frequency — *n is an arbitrary integer*

4. Control concepts (2/10)

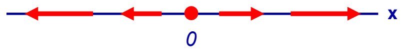
a) Aliasing phenomenon



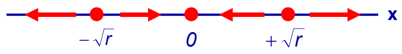
4. Control concepts (3/10)

b) Bifurcations: Subcritical bifurcation $\dot{x} = rx + x^3$

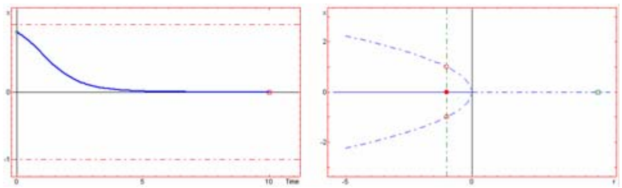
Case 1: r > 0



Case 2: r < 0



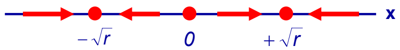
Bifurcation diagram / ***Output***



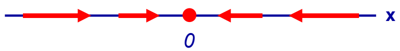
4. Control concepts (4/10)

b) Bifurcations: Supercritical bifurcation $\dot{x} = rx - x^3$

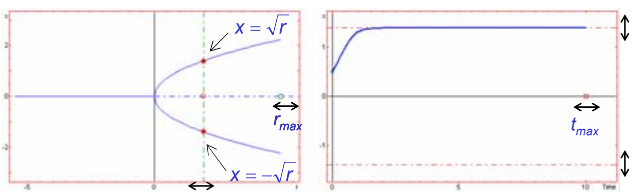
Case 1: r > 0



Cas2 2: r < 0



Bifurcation diagram / ***Output***

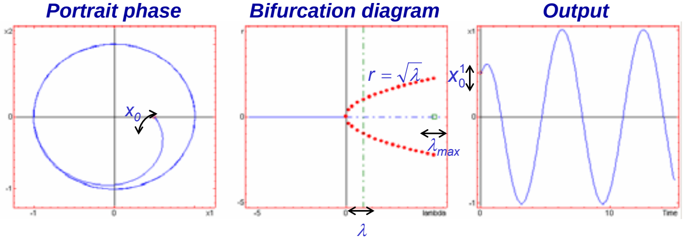


4. Control concepts (5/10)

c) Bifurcations: Poincaré-Andronov-Hopf bifurcation

$$\left.\begin{aligned}\dot{x}_1 &= x_2 + x_1\left(\lambda - x_1^2 - x_2^2\right)\\ \dot{x}_2 &= -x_1 + x_2\left(\lambda - x_1^2 - x_2^2\right)\end{aligned}\right\} \Rightarrow \begin{aligned}\dot{r} &= r\left(\lambda - r^2\right)\\ \dot{\theta} &= -1\end{aligned}$$

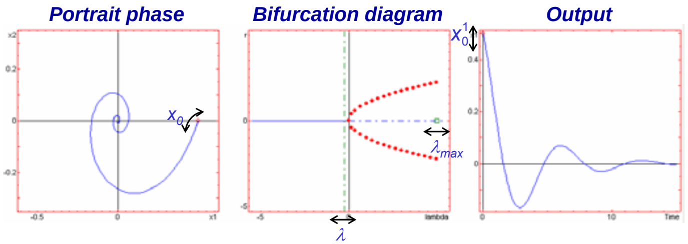
Portrait phase / ***Bifurcation diagram*** / ***Output***



4. Control concepts (6/10)

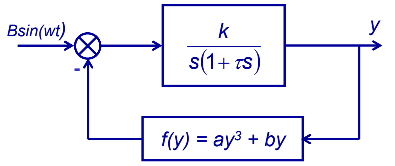
c) Bifurcations: Poincaré-Andronov-Hopf bifurcation

$$\left.\begin{aligned}\dot{x}_1 &= x_2 + x_1\left(\lambda - x_1^2 - x_2^2\right)\\ \dot{x}_2 &= -x_1 + x_2\left(\lambda - x_1^2 - x_2^2\right)\end{aligned}\right\} \Rightarrow \begin{aligned}\dot{r} &= r\left(\lambda - r^2\right)\\ \dot{\theta} &= -1\end{aligned}$$



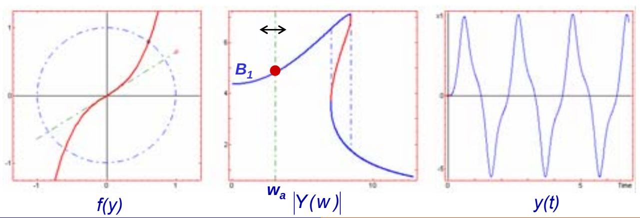
4. Control concepts (7/10)

c) Jump resonance



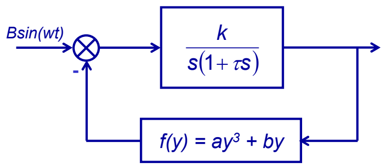
Duffing's equation

$$\ddot{y} + 2\alpha\dot{y} + \omega_0^2\left(1+\lambda y^2\right)y = B\sin(\omega t)$$



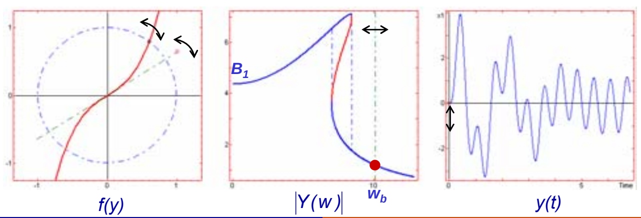
4. Control concepts (8/10)

c) Jump resonance



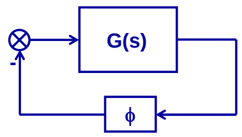
Duffing's equation

$$\ddot{y} + 2\alpha\dot{y} + \omega_0^2\left(1+\lambda y^2\right)y = B\sin(\omega t)$$



4. Control concepts (9/10)

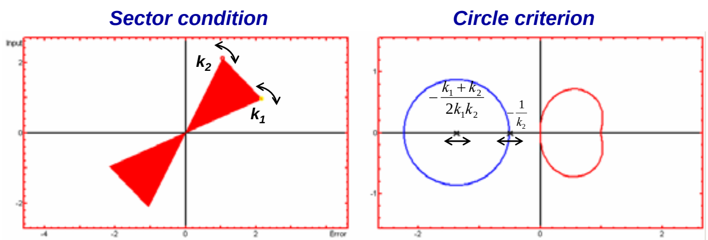
d) Absolute stability



1) G(s) has no poles in the right half plane
2) The nonlinearity f satisfies:

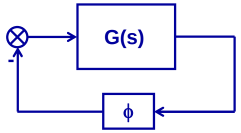
$$\phi(0) = 0, k_1 \le \frac{\phi(y)}{y} \le k_2 \text{ for } y \ne 0$$

A sufficient condition for stability is that G(iω) does not enter or enclose the circle



4. Control concepts (10/10)

d) Absolute stability



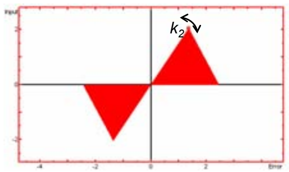
1) *G(s) has no poles in the right half plane*
2) *The nonlinearity f satisfies:*

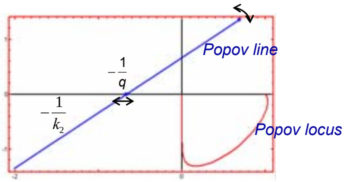
$$\phi(0) = 0, 0 \le \frac{\phi(y)}{y} \le k_2 \text{ for } y \neq 0$$

A sufficient condition for stability is:

$$Re[(1 + jwq)G(jw)] + k_2^{-1} \ge \delta > 0$$

Sector condition ***Circle criterion***



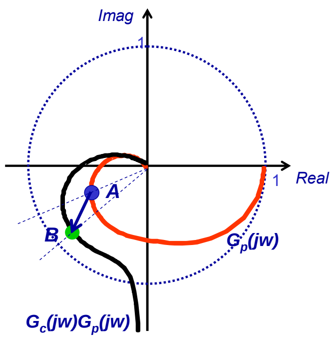


Contents

5. Controller design (1/4)

a) Design of PID controller

Loop shaping: Tuning in the frequency domain



$$A = (r_A, \phi_A) \longrightarrow B = (r_B, \phi_B)$$

$$G_C = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

$$K_p = \frac{r_B \cos(\phi_B - \phi_A)}{r_A}$$

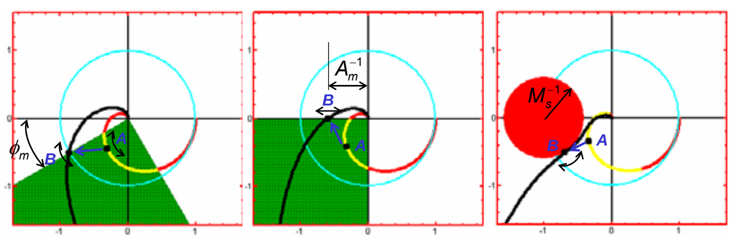
$$T_i = \frac{1}{2\alpha\omega_A}\left(tg(\phi_B - \phi_A) + \sqrt{4\alpha + tg^2(\phi_B - \phi_A)}\right)$$

$$T_d = \alpha T_i$$

5. Controller design (2/4)

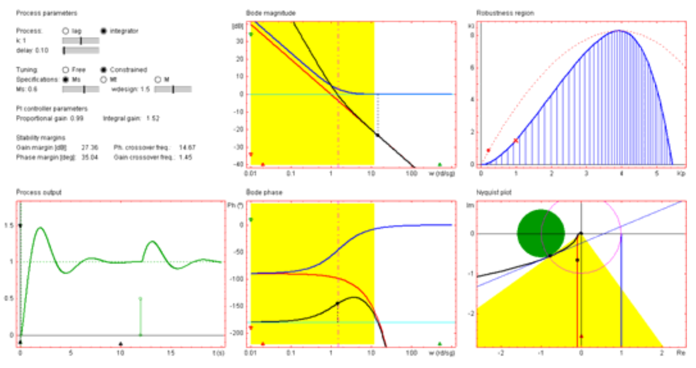
a) Design of PID controller

Phase margin design	*Gain margin design*	*Sensisitivity margin design*
$r_B = 1$, $\phi_B = \phi_m$	$r_B = A_m^{-1}$, $\phi_B = 0°$	$M_s = \max_{0 \le \omega \le \infty} \lvert S(j\omega) \rvert$, $A_m > \frac{M_s}{M_s - 1}$, $\phi_m > 2\arcsin(0.5 M_s^{-1})$



5. Controller design (3/4)

a) Design of PID controller

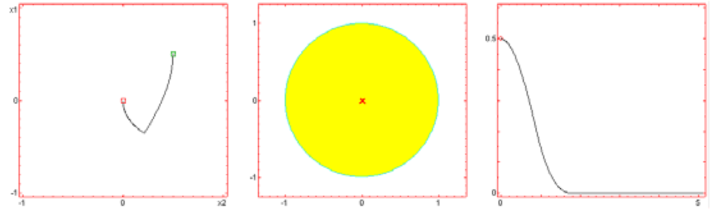


5. Controller design (4/4)

b) Pole placement design

Interactive elements in the application

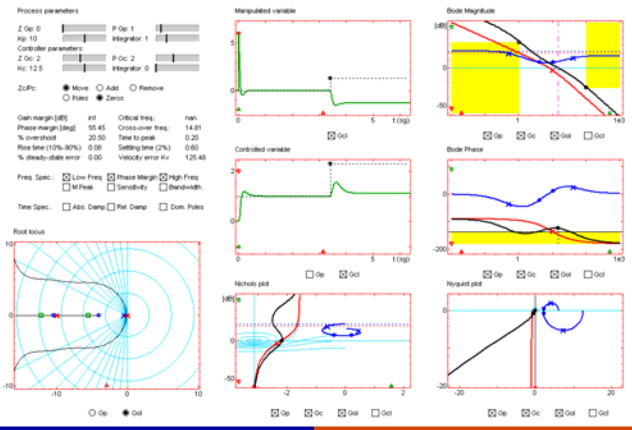
- *Plant transfer function*
- *Sampling period*
- *Initial condition*
- *Ackermann formula*
- *Desired closed loop poles*
- *Different canonical forms*



Portrait phase *CL specification* *System response*

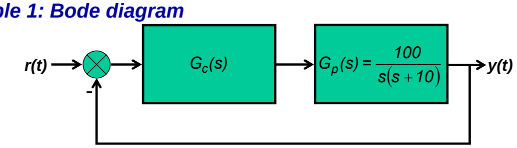
Contents

6. The LSAD tool (1/13)



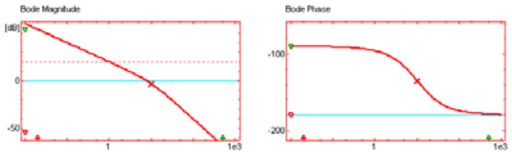
6. The LSAD tool (2/13)

Example 1: Bode diagram



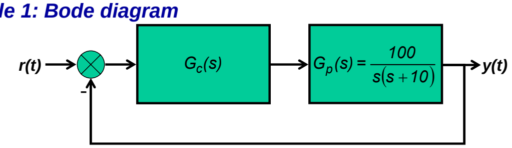
Specifications 1

a: Velocity error constant K_v = 100 sec^{-1}



6. The LSAD tool (3/13)

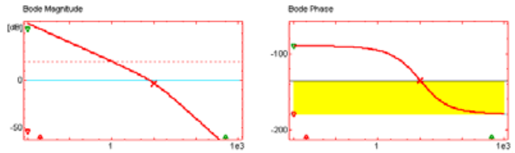
Example 1: Bode diagram



Specifications 2

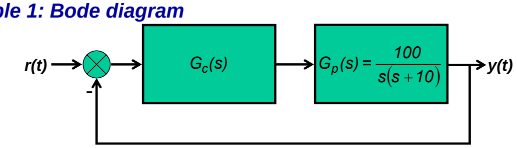
a: Velocity error constant K_v = 100 sec^{-1}

b: Phase margin ϕ_m = 45º



6. The LSAD tool (4/13)

Example 1: Bode diagram

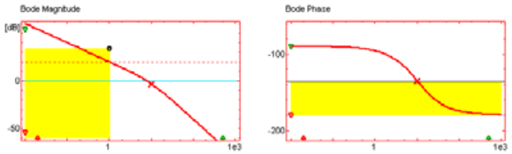


Specifications 3

a: Velocity error constant K_v = 100 sec^{-1}

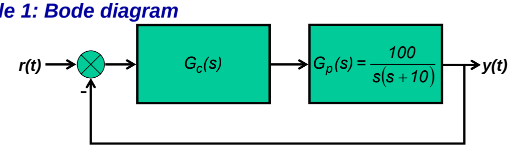
b: Phase margin ϕ_m = 45º

c: Sinusoidal inputs of up 1 rad/sec should be reproduced with $\leq$ ***2 percent error***



6. The LSAD tool (5/13)

Example 1: Bode diagram



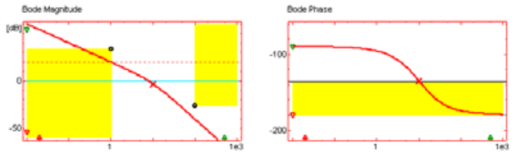
Specifications 4

a: Velocity error constant K_v = 100 sec^{-1}

b: Phase margin ϕ_m = 45º

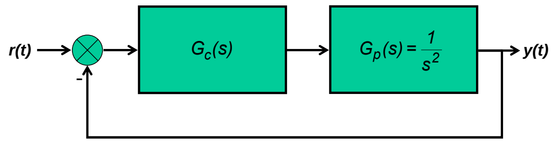
c: Sinusoidal inputs of up 1 rad/sec should be reproduced with $\leq$ ***2 percent error***

d: Sinusoidal inputs $\geq$ 100 rad/sec attenuated at the output to ***5 % of their value at the input***



6. The LSAD tool (6/13)

Example 2: Root locus



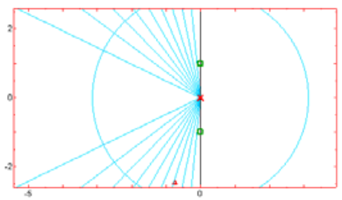
Specifications

a: Settling time (2 % criterion), $T_s \le 4$ sec

$$PO = 100 e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \le 35 \quad \Longrightarrow \zeta \ge 0.32$$

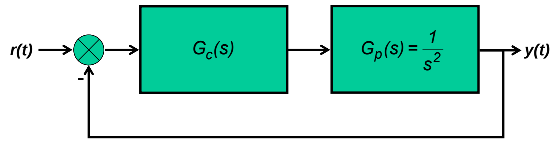
$$T_s = \frac{4}{\zeta\omega_n} = 4 \quad \Longrightarrow \zeta\omega_n = 1$$

b: Overshoot PO for a step input ≤ 35 %



6. The LSAD tool (7/13)

Example 2: Root locus



Specifications

a: Settling time (2% criterion), $T_s \le 4$ sec

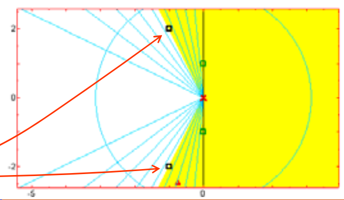
$$PO = 100 e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \le 35 \quad \Longrightarrow \zeta \ge 0.32$$

$$T_s = \frac{4}{\zeta\omega_n} = 4 \quad \Longrightarrow \zeta\omega_n = 1$$

Thus we will choose a desired dominant root location as:

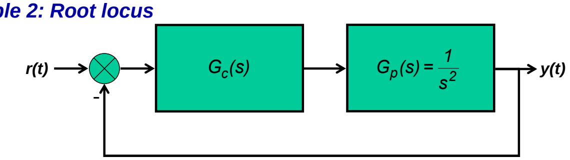
$$r_1, \hat{r}_1 = -1 \pm j2$$

b: Overshoot PO for a step input ≤ 35 %



6. The LSAD tool (8/13)

Example 2: Root locus

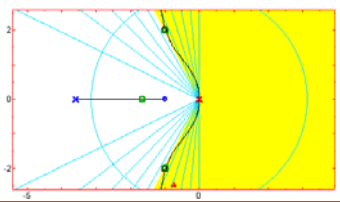


Specifications

a: Settling time (2% criterion), $T_s \le 4$ sec

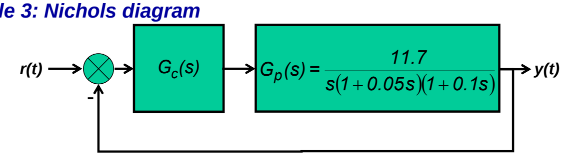
The advantage of the root locus method is the ability of the designer to specify the location of the dominant roots and therefore the dominant transient response. The disadvantage of the method is that one cannot directly specify an error constant as in the Bode diagram approach.

b: Overshoot PO for a step input ≤ 35 %



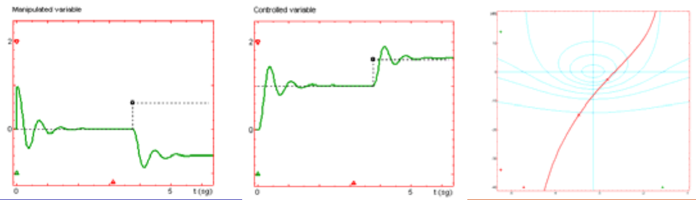
6. The LSAD tool (9/13)

Example 3: Nichols diagram



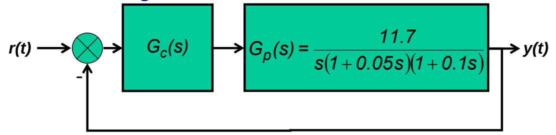
Specifications 1

a: M_p = 1.4 (max. of frequency response)



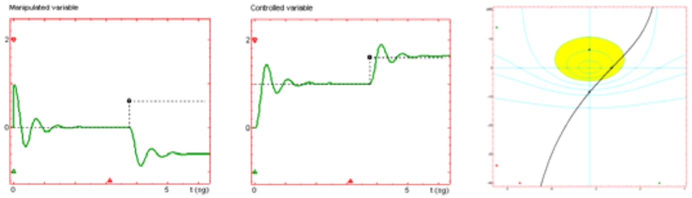
6. The LSAD tool (10/13)

Example 3: Nichols diagram



Specifications 1

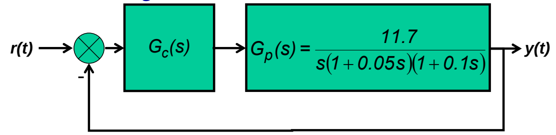
a: M_p = 1.4 (max. of frequency response)



S. Dormido | Interactive Applications in Control

6. The LSAD tool (11/13)

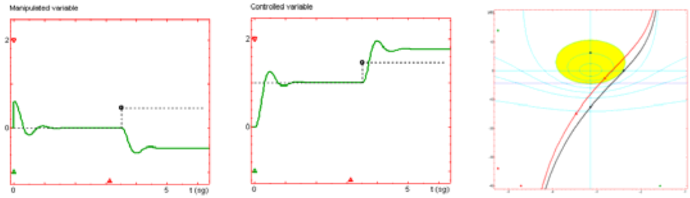
Example 3: Nichols diagram



Specifications 2

a: M_p = 1.4 (max. of frequency response)

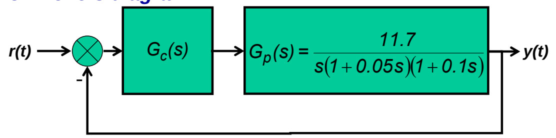
b: Velocity error constant $K_v \geq 11.5$



S. Dormido | Interactive Applications in Control

6. The LSAD tool (12/13)

Example 3: Nichols diagram

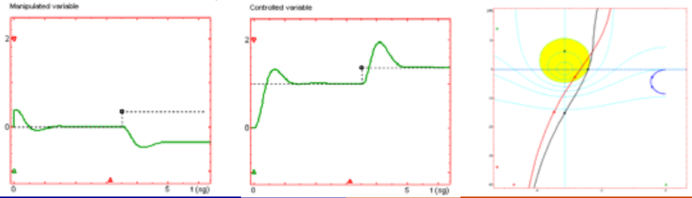


Specifications

a: M_p = 1.4 (max. of frequency response)

b: Velocity error constant $K_v \geq 11.5$

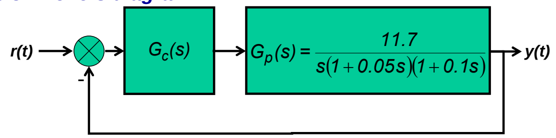
c: Settling time (2% criterion), $T_s \leq 0.9$ sec



S. Dormido | Interactive Applications in Control

6. The LSAD tool (13/13)

Example 3: Nichols diagram

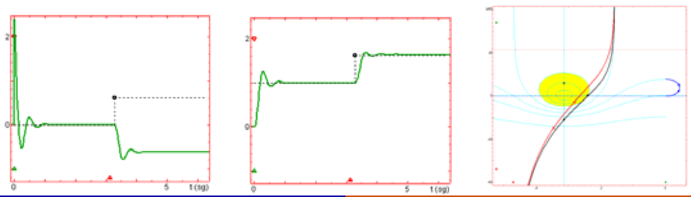


Specifications

a: M_p = 1.4 (max. of frequency response)

b: Velocity error constant $K_v \geq 11.5$

c: Settling time (2% criterion), $T_s \leq 0.9$ sec



S. Dormido | Interactive Applications in Control

Contents

7. The ILM project (1/3)

ILM = Interactive Learning Modules

- *Project participants:*

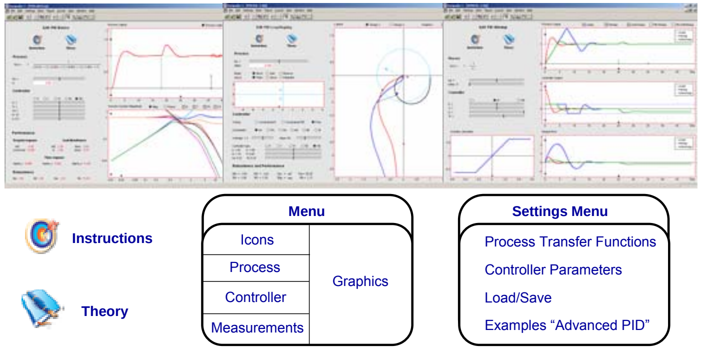
  UNED, University of Almería, University of Lund, EPFL
- *Interactive learning modules for "PID Control" based on the spread sheet metaphor.*
- *The modules are designed to speed-up learning and to enhance understanding of the behavior of loops with PID controllers.*
- *The modules are implemented in Sysquake*
- *Executable versions of the modules for PC, Mac and Unix are available on the web.*

J. L. Guzmán *"Interactive Control System Design"*, Doctoral Thesis, 2006

7. The ILM project (2/3)

Three modules ended

PID Basics *PID LoopShaping* *PID Windup*



7. The ILM project (3/3)

Modules in development phase

- *Modeling and identification*
- *PID controller design*
- *Smith predictor*
- *Feedforward control fundamentals*
- *2 × 2 PID controller*
- *Adaptive control fundamentals*

J.L. Guzmán, K. Åström, S. Dormido, T. Hägglund, Y. Piget *"Interactive learning modules for PID control"*, ACE'06

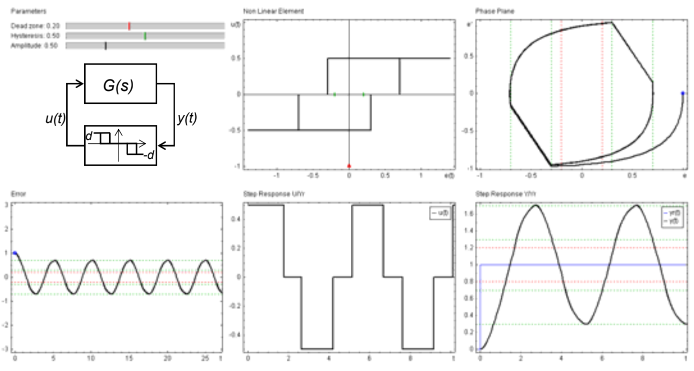
Contents

8. Relay systems (1/5)

- ***The simplest hybrid system***
  - *Two or three discrete states*
  - *Linear continuous behavior*
- ***Relays systems are still widely used in industry***
- ***Relay systems have been studied for a long time***
- ***Relay systems have rich behavior***
- ***Important problems remain to be solved***
  - *Example: Find all transfer functions such that there is a stable limit cycle.*

8. Relay systems (2/5)

A simple example



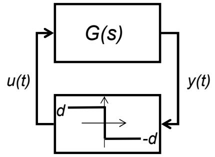
8. Relay systems (3/5)

Applications of relay feedback systems

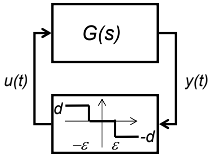
- *On–off control, temperature control*
- *Amplifiers*
- *Self-oscillating adaptive systems*
- *Relay auto-tuning*
- *DC/DC converters*
- *Δ-$\sum$ modulators*
- *Coulomb friction*
- *Variable structure systems*

8. Relay systems (4/5)

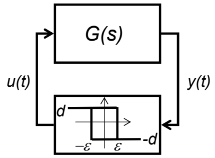
Relays systems types



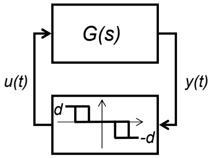
Ideal relay



Relay with dead-zone



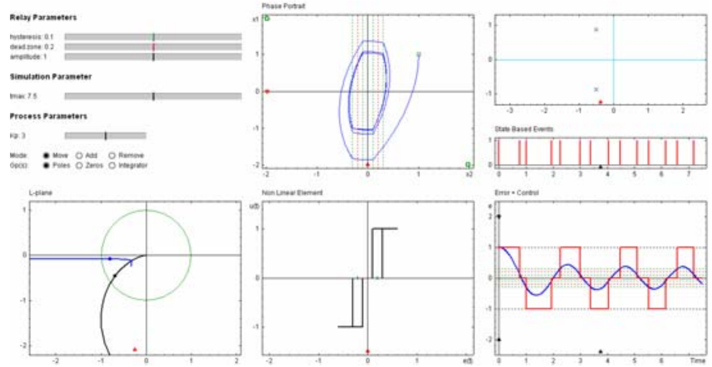
Relay with hysteresis



Relay with dead-zone and hysteresis

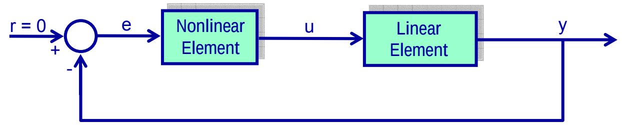
8. Relay systems (5/5)

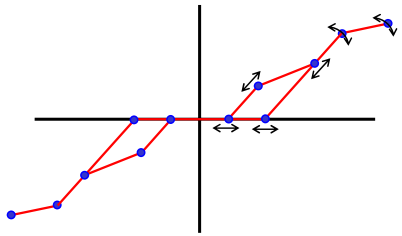
The "relaytool"



Contents

9. The nonlinear control tool (1/2)





Reconocimiento: Prof. Francisco Gordillo, Universidad de Sevilla

7. The nonlinear control tool (2/2)

Herramienta interactiva para docencia:

- *Analysis of equilibrium points*
- *Describing function determination*
- *Prediction of limit cycles and Hopf bifurcation*
- *Saddle-node bifurcation of limit cycles*
- *Sliding motion*
- *Dead zone*

S. Dormido, F. Gordillo, S. Dormido-Canto, J. Aracil:
"*An Interactive Tool for Introductory Nonlinear Control Systems Education*"
IFAC 15th World Congress b'02, july, Barcelona

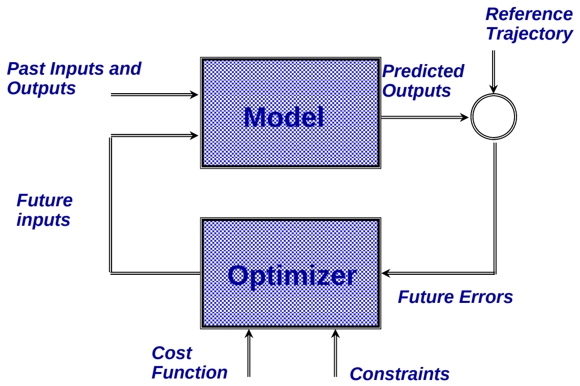
Contents

8. The MPC tool (1/13)

- *Predictive control was developed and used in industry for nearly 25 years before attracting much serious attention from the academic control community.*
- *Fortunately, the academic community has for some years now appreciated that predictive control really does offer something new for control in the presence of constraints, and has provided much analysis, and new ideas, to such an extent that it has gone beyond current industrial practice and is preparing the ground for much wider application of predictive control –potentially to almost all control engineering problems.*
- *The constant increase in computing speed and power certainly makes that a real prospect, with the interactive calculation of QP problems in order to solve constrained predictive control.*

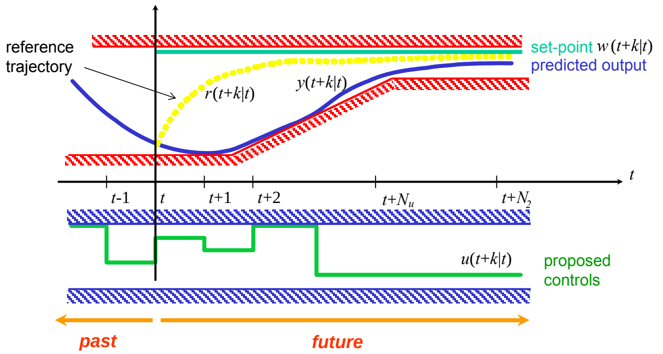
8. The MPC tool (2/13)

Basic structure of MPC



8. The MPC tool (3/13)

MPC Strategy



8. The MPC tool (4/13)

Generalized predictive control constraints

Variable	*Linear constraint*
Control signal amplitude $u_{min} \leq u(t) \leq u_{max}, \ \forall t.$	$\mathrm{l}\ u_{min} \leq T\ \Delta u + u(t-1)\mathrm{l} \leq \mathrm{l}\ u_{max}$
Control signal increment $\Delta u_{min} \leq u(t)-u(t-1) \leq \Delta u_{max}\ \forall t$	$\mathrm{l}\ \Delta u_{min} \leq \Delta u \leq \mathrm{l}\ \Delta u_{max}$
Output signal amplitude $y_{min} \leq y(t) \leq y_{max}\ \forall t$	$\mathrm{l}\ y_{min} \leq G\ \Delta u + f \leq \mathrm{l}\ y_{max}$
Output band constraints $y_{min}(t) \leq y(t) \leq y_{max}(t)\ \forall t$	$G\ \Delta u \leq y_{max} - f;$ $y_{max} = [y_{max}(t+1)\ y_{max}(t+2)\ \ldots\ y_{max}(t+N)]$ $G\ \Delta u \geq y_{min} - f;$ $y_{min} = [y_{min}(t+1)\ y_{min}(t+2)\ \ldots\ y_{min}(t+N)]$

8. The MPC tool (5/13)

Generalized predictive control constraints

Variable	*Restricción lineal*
Output overshoot $y(t+j) \leq \gamma\, w(t);\ j = N_{o1} \ldots N_{o2},$	$G\ \Delta u \leq \mathrm{l}\ \gamma\, w(t) - f\ \mathrm{l}$
Output monotonous behaviour $y(t+j) \leq y(t+j+1)$ if $y(t)<w(t)$ $y(t+j) \geq y(t+j+1)$ if $y(t)>w(t)$	$G\Delta u + f \leq \begin{bmatrix} 0^T \\ \hline G' \end{bmatrix} \Delta u + \begin{bmatrix} y(t) \\ \hline f' \end{bmatrix}$ *G', f' are the result of eliminating the last row of G and f.*
Avoid NMP behaviour $y(t+j) \leq y(t)$ if $y(t)>w(t)$ $y(t+j) \geq y(t)$ if $y(t)<w(t)$	$G\ \Delta u \geq \mathrm{l}\ y(t) - f$
Final state $y(t+N+1) \ldots y(t+N+m) = w$	$y_m = [y(t+N+1) \ldots y(t+N+m)]^T$: $y_m = G_m \Delta u + f_m \qquad G_m \Delta u = w_m - f_m$

8. The MPC tool (6/13)

SISO GPCIT tool

- *When developing a tool of this kind, the programming of the algorithms is a time consuming aspect, but one of the most important things that the teacher has to have in mind is the organization of the main windows and menus of the tool to facilitate the student the understanding of the control technique*
- *In this case, the main window is divided into several sections using colour codes to represent different objects.*
- *From our experience, it is better to try to use graphical elements more than contextual menus.*

8. The MPC tool (7/13)

SISO GPCIT tool

Features from the designer/user point of view:

- *Considerable effort to program all the algorithms.*
- *Computational cost.*
- *Different tuning knobs:*
  - ***Unconstrained case**: 21 degrees of freedom/tuning knobs in simulations*
  - ***Constrained**: + 10 design parameters*

Classical toolboxes (e.g. MPC) require many different simulation tests to be performed to see the effect of changing design/simulation parameters in closed loop performance

8. The MPC tool (8/13)

Parameters/elements that can be changed interactively:

Process parameters:

- *Model: nº zeroes, nº poles, delay, zeroes location, poles location*
- *Plant: nº zeroes, nº poles, delay, zeroes location, poles location*

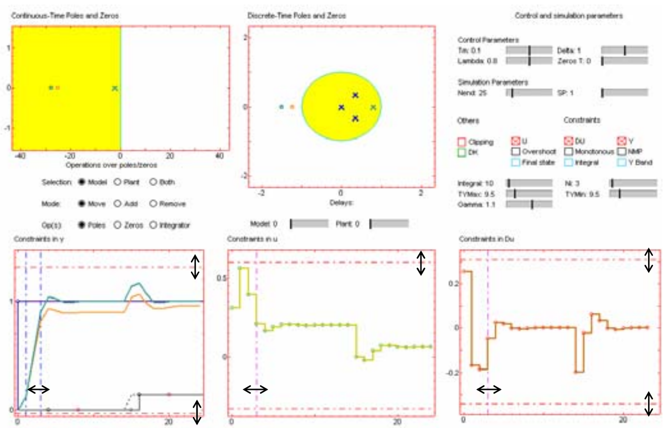
(Model and Plant: Plant-model mismatch)

- *Discretization method, sampling time*

Control and simulation parameters:

- *λ, δ, N_1, N_2, N_u, nº zeroes and location of T polynomial*
- *Constraints: u_{min}, u_{max}, Du_{min}, Du_{max}, y_{min}, y_{max}, y_{band} ($_{max}$ and $_{min}$), γ (overshoot limitation), m (final state)*
- *Future set-points*
- *Simulation time*

8. The MPC tool (9/13)



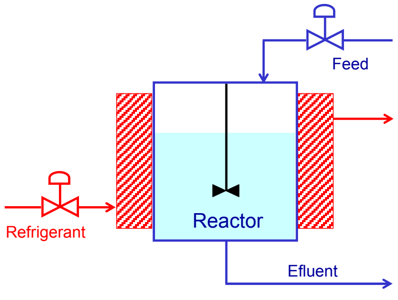
8. The MPC tool (10/13)

Ilustrative examples

- ***Simple example** (Camacho and Bordóns, 1999): effect of tuning knobs (l, d, N_1, N_2, N_u) etc.*
- ***T polynomial** (Clarke and Mohtadi, 1989): In the usual formulation of GPC, the polynomial $C(z^{-1})$ usually equals 1. When robustness is to be enhanced, this polynomial is used as a design one ("T-polynomial") that can be treated as a filter to attenuate the components of the prediction errors due to modelling uncertainties, as far as unmeasurable load disturbances.*
- ***Physical, security and performance constraints** (Berenguel, 1996): **output constraints** (amplitude, band, NMP, overshoot), **input constraints**.*
- ***Constraints and stability** (Berenguel, 1996; Clarke and Scattolini, 1994; Scokaert, 1994): GPC vs. clipping, CRHPC, GPC,*

8. The MPC tool (11/13)

Example 1: Mixed tank reactor (Camacho and Bordons)



Controled variables:

Output concentration
Reactor temperature

Manipulated variables:

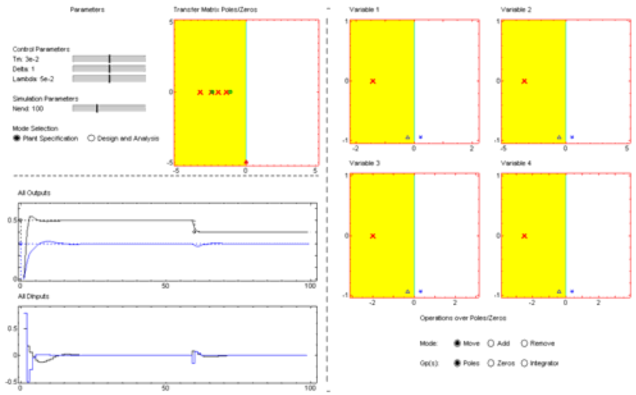
Feed flow
Refrigerant flow

Transfer matrix:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{1+0.1s} & \frac{5}{1+s} \\ \frac{1}{1+0.5s} & \frac{2}{1+0.4s} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}$$

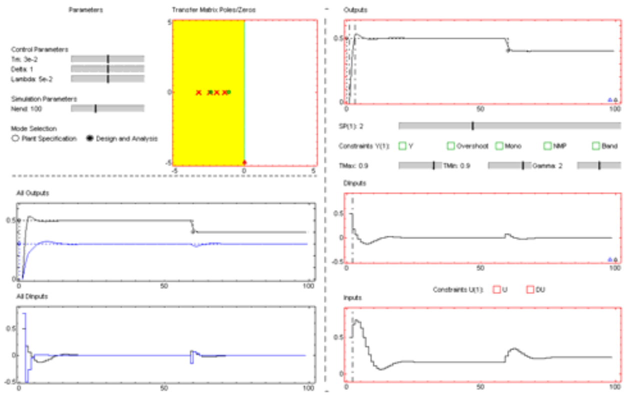
8. The MPC tool (12/13)

MIMO GPCIT tool



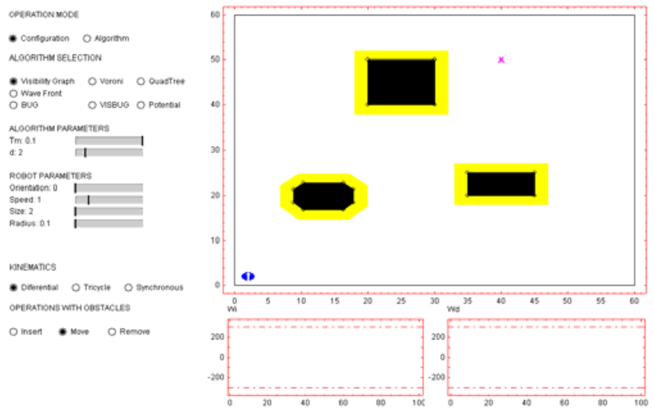
8. The MPC tool (13/13)

MIMO GPCIT tool



Contents

11. The MRIT tool



Contents

12. Conclusions (1/2)

The underlying philosophy of the interactive software is to minimize programming effort on the part of the students and allow them to concentrate on the learning of control concepts.

The use of interactive tools to highlight the essential aspects of dynamics systems and control.

- *Simulation analysis of system stability*
- *Effect of parameter variation*
- *Evaluation of feedback control system*
- *Transient and steady-state analysis*
- *Frequency response analysis and design*

From the instructor's point of view, teaching control concepts has never been more exciting now that one has the use of interactive.

12. Conclusions (2/2)

The scenario for Control Education is changing and we must react. Information technology open a whole new world of real opportunities.

and the message is...

Computers show a great potential to enhance student achievement, but only if they are used appropriately, as part of a coherent education approach.

But not just in the same way as books or labs. Computers allow us to go deeper and faster...