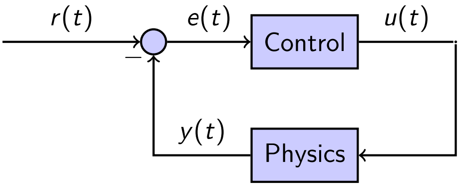
Tuning PI controllers in non-linear uncertain closed-loop systems with interval analysis

J. Alexandre dit Sandretto, **A. Chapoutot** and O. Mullier

U2IS, ENSTA ParisTech

SYNCOP
April 11, 2015

Closed-loop control systems



- **Control** is a continuous-time PI controller

$$e(t) = r(t) - y(t) \ , \qquad u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$$

- **Physics** is defined by non-linear ODEs

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), u(t)) \ , \qquad y(t) = g(\mathbf{x}(t))$$

What is tuning a PI controller?

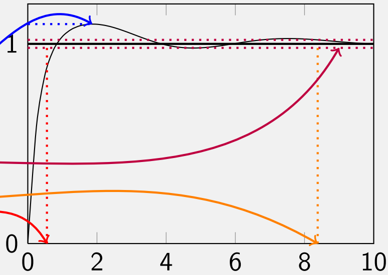
Find values for K_p and K_i such that a given specification is satisfied.

Specification of PID Controllers

PID controller: requirements based on closed-loop response

We observe the output of the plant

- Overshoot: Less than 10%
- Steady-state error: Less than 2%
- Settling time: Less than 10s
- Rise time: Less than 2s



Note: such properties come from the **asymptotic behavior** (well defined for linear case) of the system.

Classical tuning methods for PID controllers – 1

Ziegler-Nichols' closed-loop method, based on simulations:

- deactivate the I and the D part of the PID controller;
- increase K_p until a value K_u where the output of the physics oscillates with a constant amplitude and a period T_u;
- look into the Ziegler-Nichols' table to set the values of K_p, K_i and K_d. For example, $K_p = 0.6K_u$, $K_i = 2K_p/T_u$ and $K_d = K_p T_u/8$.

Taking into account uncertainties?

Models of physics are not well known during the design

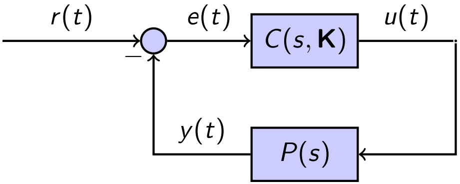
$$\dot{\mathbf{x}} = f(\mathbf{x}(t), u(t), \mathbf{p}) \quad \text{with} \quad \mathbf{p} \in \mathcal{P} \quad \text{bounded} \quad .$$

Consequence to apply Ziegler-Nichols' method, Monte-Carlo simulation should be used.

Remark: for linear systems, better PID tuning methods exists as Pole Placement.

Classical tuning methods for PID controllers – 2

Special case for **linear** **closed-loop** (with uncertainties) systems



- $C(s, \mathbf{K})$ and $P(s, \mathbf{p})$ are the transfer functions of the controller and the uncertain physics
- the closed-loop transfer function is $H(s, \mathbf{K}) = \dfrac{C(s, \mathbf{K})}{1 - C(s, \mathbf{K})P(s)}$

Tuning with interval analysis [Bondia et ali, 2004]

- With uncertain physics that is $P(s, \mathbf{p})$ with $p \in \mathcal{P}$;
- Define a specification $M(j\omega)$ and a interval $\bar{\omega}$ of frequencies;
- Find $\mathbf{K} \in \mathcal{K}$ such that $H(j\omega, \mathbf{K}, \mathbf{p}) \subseteq M(j\omega)$ for all $\omega \in \bar{\omega}$

Our approach

Hypotheses:

- **uncertain non-linear** closed-loop control system;
- **only PI controllers** are considered in this work;
- Only **specification based settling-time** is considered

$$\begin{cases} y(\mathbf{t_{end}}) \in [r - \alpha\%, r + \alpha\%], \ 100 \geqslant \alpha > 0 \\ \dot{y}(t_{end}) \in [-\epsilon, \epsilon], \ \epsilon > 0 \end{cases}$$

Our method

- Embed PI controller into the non-linear physics;
- Searching for valid parameters using interval analysis tools
  - **inclusion functions**
  - **paving**
  - **validated numerical integration**

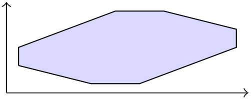
Inclusion functions based affine arithmetic

- Extension of **interval arithmetic** to reduce the dependency issue.

$$[a, b] - [c, d] = [a - d, b - c] \qquad x = [0, 1] \Rightarrow x - x = [-1, 1]$$

- Main idea: parametric variables w.r.t. a set of **noise symbols** ε_i with $\varepsilon_i \in [-1, 1]$.

$$x = x_0 + x_1\varepsilon_1 + x_3\varepsilon_3$$
$$y = y_0 + y_1\varepsilon_1 + y_2\varepsilon_2$$



- This representation encodes the linear relation between noise symbols and variables and allows for precise linear transformation.

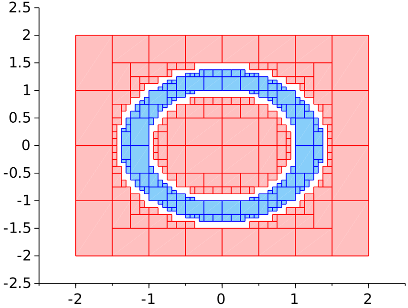
- What is a noise symbol?
  - Initial uncertainty: $[a, b] \rightarrow \frac{a+b}{2} + \frac{b-a}{2}\varepsilon$
  - Non-linear operations and round-off errors.

Paving

Methods used to represent complex sets $\mathcal{S}$ with

- inner boxes i.e. set of boxes included in $\mathcal{S}$
- outer boxes i.e. set of boxes that does not belong to $\mathcal{S}$
- the frontier i.e. set of boxes we do not know

Example, a ring $\mathcal{S} = \{(x, y) \mid x^2 + y^2 \in [1, 2]\}$ over $[-2, 2] \times [-2, 2]$



Remark: involving bisection algorithm and so complexity is exponential in the size of the state space.

Solution of uncertain non-linear ODEs

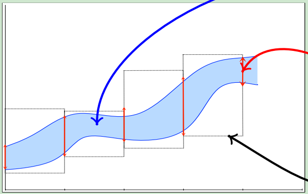
Challenge

Define method to compute reachable set of continuous dynamical systems: $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{p})$, especially when f is non-linear.

Current direction

Modified numerical integration methods to deal with sets of values and nonlinear dynamics: explicit and implicit Runge-Kutta methods.

Example

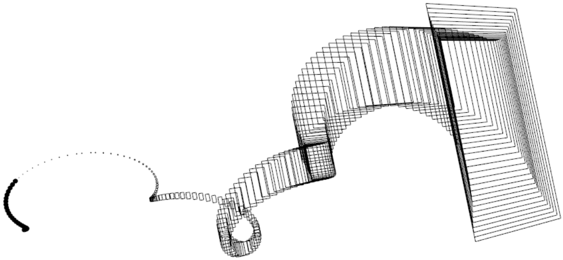


- Exact solution of $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{p})$ with $x(0) \in \mathcal{X}$.
- Safe approximation at discrete time instants.
- Safe approximation between time instants.

Example of flow-pipe with validated Runge-Kutta

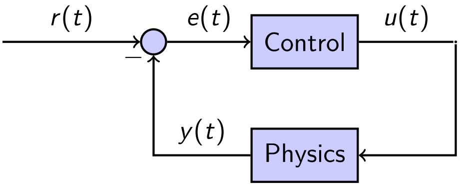
$$\begin{pmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 1 \\ x_2 \\ \frac{1}{6}x_1^3 - x_1 + 2\sin(d \cdot x_0) \end{pmatrix} \quad \text{with} \quad d \in [2.78, 2.79]$$

Simulation for 10 seconds with $x_1(0) = x_2(0) = x_3(0) = 0$



The last step is
$x(10) = ([10, 10], [-1.6338, 1.69346], [-1.55541, 1.4243])^T$

REMAINDER – closed-loop control systems



- **Control** is a continuous-time PI controller

$$e(t) = r(t) - y(t) \ , \qquad u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$$

- **Physics** is defined by **uncertain non-linear ODEs**

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), u(t), \mathbf{p}) \ , \qquad y(t) = g(\mathbf{x}(t))$$

Our goal

Find values for K_p and K_i such that a given specification is satisfied.

Case study

A cruise control system two formulations:

- uncertain linear dynamics;

$$\dot{v} = \frac{u - bv}{m}$$

- uncertain non-linear dynamics

$$\dot{v} = \frac{u - bv - 0.5\rho CdAv^2}{m}$$

with

- m the mass of the vehicle
- u the control force defined by a PI controller
- bv is the rolling resistance
- $F_{\text{drag}} = 0.5\rho CdAv^2$ is the aerodynamic drag (ρ the air density, CdA the drag coefficient depending of the vehicle area)

Case study – settings and algorithm

Embedding the PI Controller into the differential equations:

- Starting point

$$u = K_p(v_{set} - v) + K_i \int (v_{set} - v)ds,$$

  with v_{set} the desired speed

- Transforming $\text{int}_{\text{err}} = \int (v_{set} - v)ds$ into differential form

$$\frac{\text{int}_{\text{err}}}{dt} = v_{set} - v$$

$$\dot{v} = \frac{K_p(v_{set} - v) + K_i \text{int}_{\text{err}} - bv}{m}$$

Main steps of the algorithm

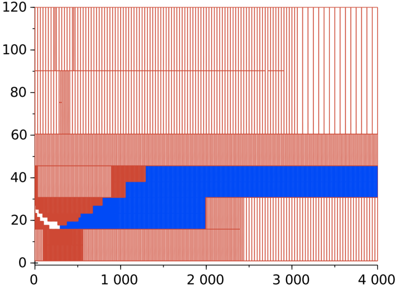
- Pick an interval values for K_p and K_i
- **Simulate** the closed-loop systems with K_p and K_i
  - if `specification is not satisfied`: **bisect** (up to minimal size) intervals and run simulation with smaller intervals
  - if `specification is satisfied` try other values of K_p and K_i

Case study – paving results

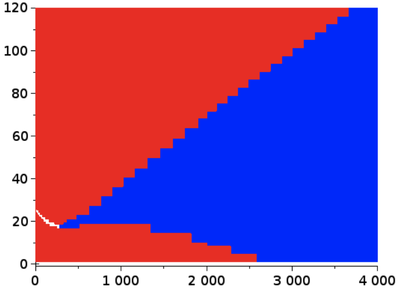
Result of paving for both cases with

- $K_p \in [1, 4000]$ and $K_i \in [1, 120]$
- $v_{\text{set}} = 10$, $t_{\text{end}} = 15$, $\alpha = 2\%$ and $\epsilon = 0.2$ and minimal size=1
  (Reminder: $y(t_{end}) \in [r - \alpha\%, r + \alpha\%]$ and $\dot{y}(t_{end}) \in [-\epsilon, \epsilon]$)

Linear case (CPU $\approx$ 10 minutes)

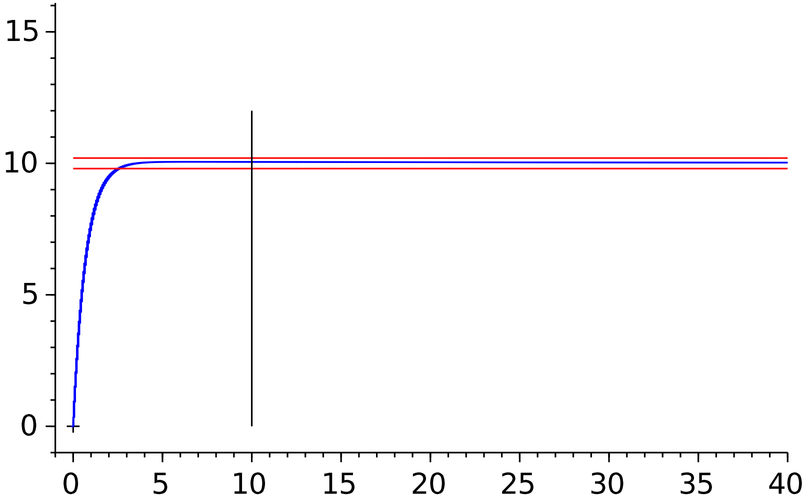


Non-linear case (CPU $\approx$ 80 minutes)



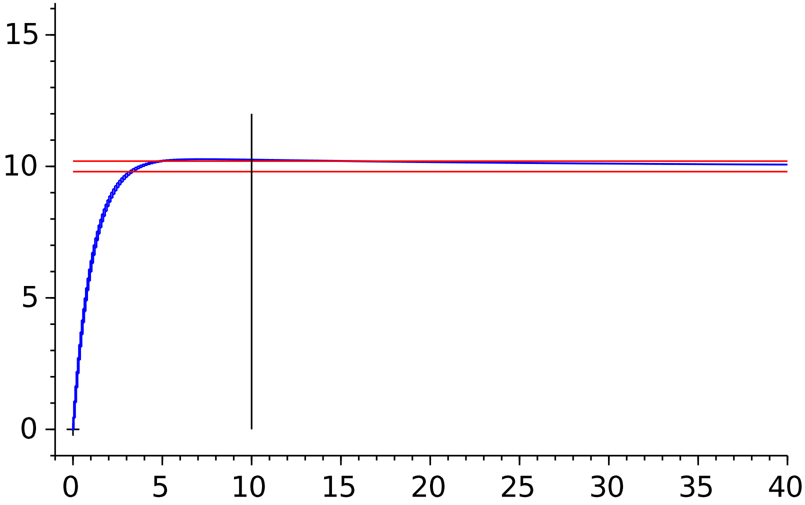
Case study – a quick verification 1 (linear case)

Taking a particular set of parameters $K_p = 1400$ and $K_i = 35$ in **validated parameter state**



Case study – a quick verification 2 (linear case)

Taking a particular set of parameters $K_p = 900$ and $K_i = 40$ in **rejected parameter state**



Conclusion

Extension of interval analysis tools

- for non-linear closed-loop control systems using **validated integration**;
- to find all the valid parameters (paving) with respect to given specification.

Future work

- Consider more examples;
- Consider PI**D** controllers;
- Consider discrete-time PID controller;
- Extended the specification considered to overshoot and rising time;
- Consider other controllers (ideas?).