Numerical analysis of the para-model control with dynIBEX and FLUCTUAT - A preliminary study -

Loïc Michel

October 25, 2017

Contents

1 Model-free control approach

The model-free control methodology has been originally proposed by Fliess & Join [1], which is referred to as a self-tuning controller in [2] and which has been widely and successfully applied to many mechanical and electrical processes. This control law has been designed to "robustify" *a priori* any "unknown" dynamical system for which not only uncertainties and unexpected modifications of the model parameter(s) are considered, but also switched models and models with time-delay(s)...
The principle of this control law consists in building an ultra-local model of the controlled process from the measurements of the input and output signals, but the main disadvantage is that the derivative of the output signal is required. This ultra-local model is a part of an "auto-adjusting" or "extended" PI control and the performances are really good taking into account that no explicit model is *a priori* given - the control is only based on input & output signals.
One of the last contributions, called para-model agent (PMA) [3], removes the use of the derivatives and replaces them by an initialization function. This contribution can be considered as a *derivative-free* & *model-free control scheme*. The last application, which has been successfully experimentally validated, deals with the nonlinear control of the Epstein Frame, which is a device to characterize some physical properties of magnetic materials [4].
In this report, using dynIBEX [5] and FLUCTUAT [6], we present a preliminary analysis of the behavior in simulation of controlled linear and nonlinear systems in terms of the propagation of the numerical errors.

2 General Principle

We consider a nonlinear SISO dynamical system to control:

$$u \mapsto y, \quad \begin{cases} \dot{x} = f_{nl}(x,u) \\ y = Cx \end{cases} \tag{1}$$

where f_{nl} is a nonlinear system, the para-model agent is an application $(y^*, y) \mapsto u$ whose purpose is to control the output y of (1) following an output reference y^*. In simulation, the system (1) is controlled in its "original formulation" without any modification / linearization.

2.1 Definition of the closed-loop

Consider the control scheme depicted in Fig. 1 where $\mathcal{C}_\pi$ is the proposed PMA controller and K_{in}, K_{out} are real positive gains.

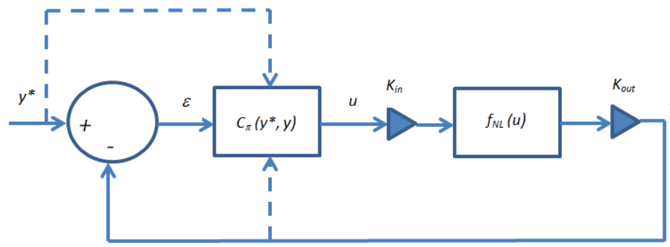


Figure 1: Proposed Para-model scheme to control a nonlinear system.

2.2 Definition of the Para-model algorithm

For any discrete moment t_k, $k \in \mathbb{N}^*$, one defines the discrete controller $\mathcal{C}_\pi$ such that symbolically:

$$\mathcal{C}_\pi : \quad \begin{array}{c} \mathbb{R}^2 \to \mathbb{R} \\ (y, y^*) \mapsto u_k = \displaystyle\int_0^t K_i \varepsilon_{k-1} d\tau \Big|_{k-1} \underbrace{\{u'_{k-1} + K_p(k_\alpha e^{-k_\beta k} - y_{k-1})\}}_{u'k} \end{array} \tag{2}$$

where: y^* is the output reference trajectory; K_p and K_I are real positive tuning gains; $\varepsilon_{k-1} = y^*_{k-1} - y_{k-1}$ is the tracking error; $k_\alpha e^{-k_\beta k}$ is an initialization function where k_α and k_β are real constants; practically, the integral part is discretized using e.g. Riemann sums. Thus, we define the set of $\mathcal{C}_\pi$-parameters of the controller as the set of coefficients $\{K_i, K_p, k_\alpha, k_\beta\}$.

3 Examples of dynamical systems

Linear case Consider the discrete first order linear dynamical system that includes an uncertain real parameter ν_1:

$$x_{k+1} = x_k + h(1000\,\nu_1\, x_k + 1000\, u_k) \tag{3}$$

with $\nu_1 \in [0.99,\ 1.02]$ and h is the time-step.
The associated discrete closed-loop, in which we introduce also an uncertain real parameter ν_2, reads:

$$\begin{cases} x_{k+1} = x_k + h(1000\,\nu_1\, x_k + 1000\, u_k) \\ u_k = \displaystyle\int_0^t K_i \varepsilon_{k-1} d\tau \Big|_{k-1} \underbrace{\{u'_{k-1} + K_p(\nu_2 k_\alpha e^{-k_\beta k} - y_{k-1})\}}_{u'k} \end{cases} \tag{4}$$

Nonlinear case Consider the discrete first order nonlinear dynamical system[1] with an uncertain parameter ν_1 :

$$x_{k+1} = x_k + \nu_1 h \frac{1000\, x_k + 1500\, u_k - 10\, x_k^2}{\sin\left(\frac{\pi}{2} + x_k\right)} \tag{5}$$

with $\nu_1 \in [0.99,\ 1.02]$ and h is the time-step.
The associated discrete closed-loop, in which we introduce also an uncertain parameter ν_2, reads:

$$\begin{cases} x_{k+1} = x_k + \nu_1 h \frac{1000\, x_k + 1500\, u_k - 10\, x_k^2}{\sin\left(\frac{\pi}{2} + x_k\right)} \\ u_k = \displaystyle\int_0^t K_i \varepsilon_{k-1} d\,\tau \Big|_{k-1} \underbrace{\left\{u'_{k-1} + K_p(\nu_2 k_\alpha e^{-k_\beta k} - y_{k-1})\right\}}_{u'k} \end{cases} \tag{6}$$

Figures 2, 3 and 4 present the "standard" (C-implementation) simulation of the controlled systems considering a non optimal para-model controller[2] and using a classical forward Euler scheme.

[1]This example has been inspired from the "Simulation of a car" example illustrated in [5].
[2]The parameters $\{K_i, K_p, k_\alpha, k_\beta\}$ in (2) have been roughly tuned i.e. not optimality tuned related a performance index.

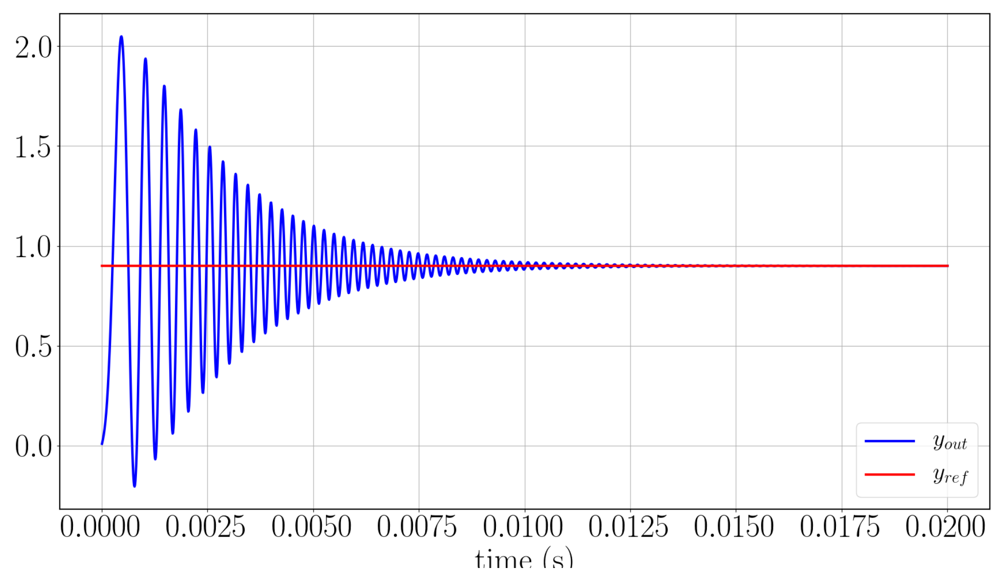
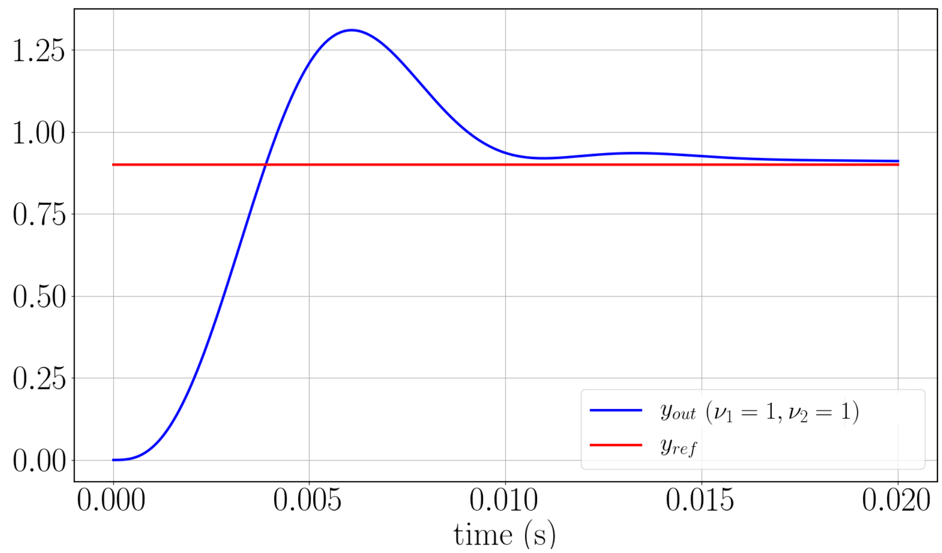
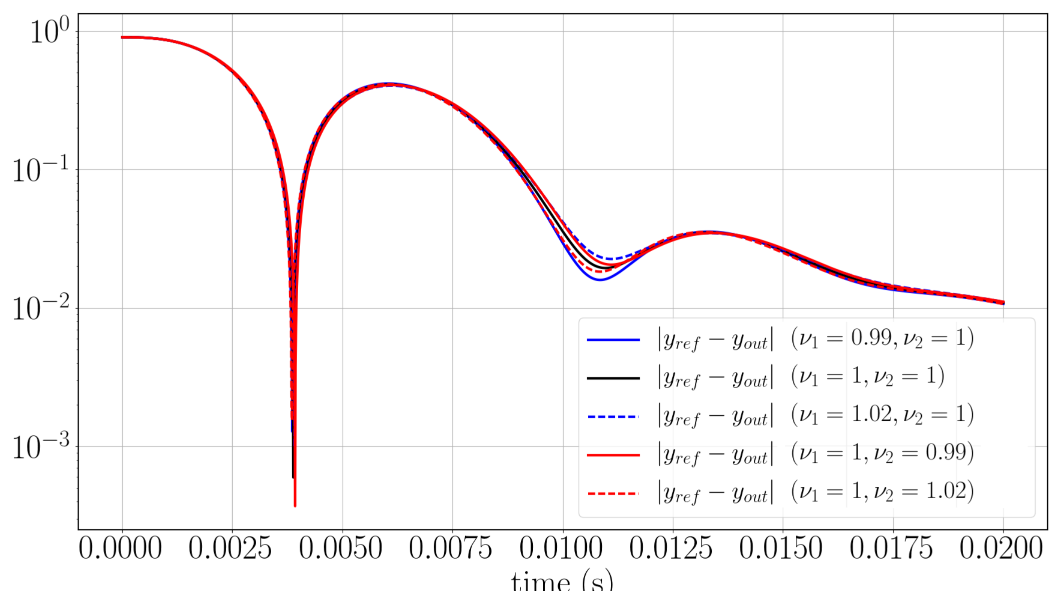


Figure 2: Example of standard simulation of the controlled linear system (4) with $\nu_1 = \nu_2 = 1$ and $h = 10\,\mu$s; the controller has been chosen to give a fast highly resonant response (it will be studied with FLUCTUAT).

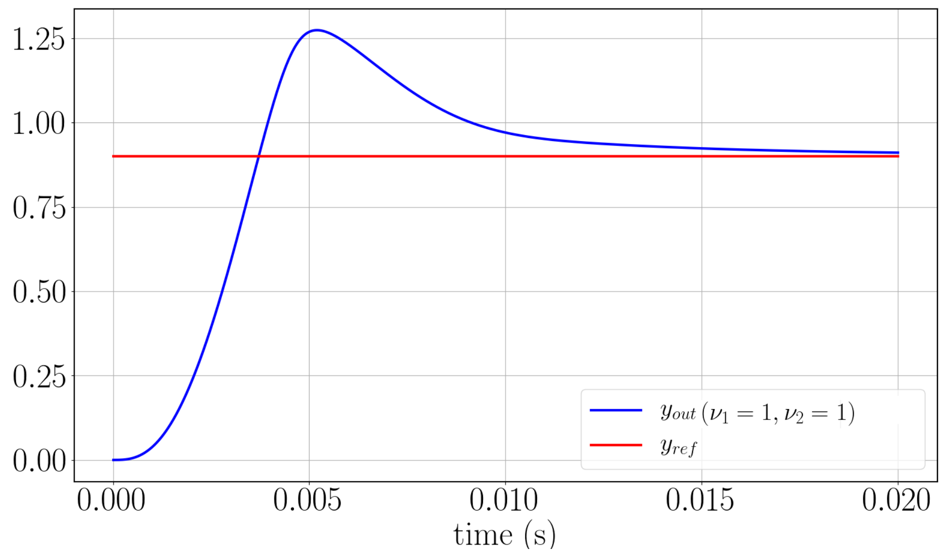


(a) Step response of the nominal system in closed-loop.

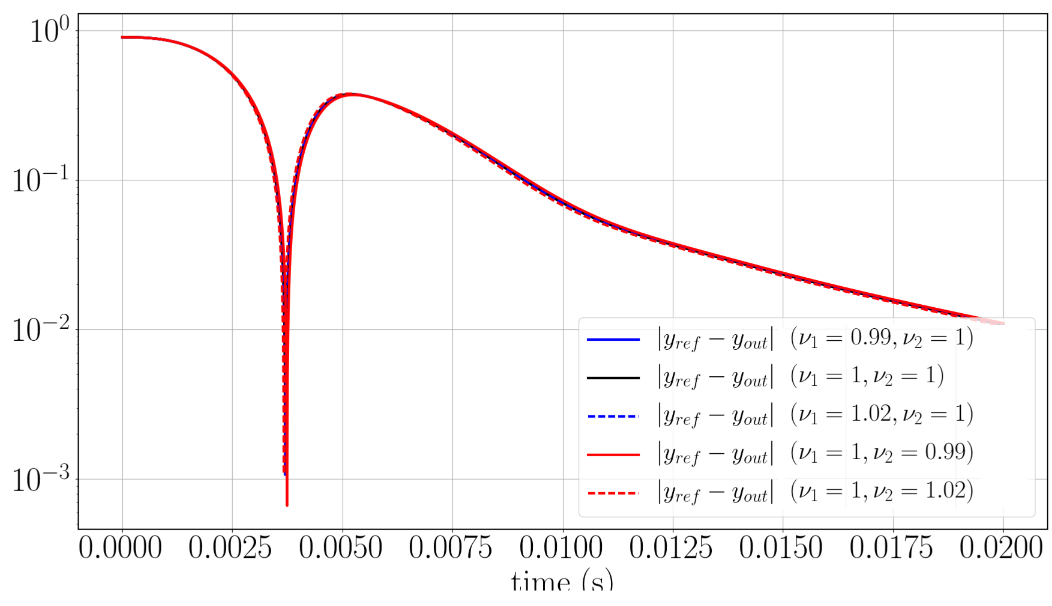


(b) Evolution of the tracking error $|y_{ref} - y_{out}|$ considering the uncertain parameters ν_1 and ν_2 in closed-loop.

Figure 3: Example of standard simulation of the step responses of the controlled linear system (4) with uncertainties (the parameters of the controller are not optimal) with $h = 10^{-5}$ s.



(a) Step response of the nominal system in closed-loop.



(b) Evolution of the tracking error $|y_{ref} - y_{out}|$ considering uncertain parameters ν_1 and ν_2 in closed-loop.

Figure 4: Example of standard simulation of the step responses of the controlled nonlinear system (6) with uncertainties (the parameters of the controller are not optimal) with $h = 10^{-5}$ s.

4 Analysis with dynIBEX

4.1 Asumptions and notations

Given a solution y_k of the "boxed"-form $[\mathbf{y_k}]$ at each discrete instant k, we denote, $\underline{y_k}$ the lower bound and, $\overline{y_k}$ the upper bound of the box $[\mathbf{y_k}]$.
Regarding the closed-loop dynamical systems under study:

- denote $< y_{out} >$, the "center" of the box $[\mathbf{y_k}]$ such as $< y_{out} >= \frac{\underline{y_k}+\overline{y_k}}{2}$;
- denote σ_{out}, the "length of the box" such as $\sigma_{out} = \overline{y_k} - \underline{y_k}$.

Denote `simLength` the (expected) simulation time index (i.e. the simulation is over when $k =$ `simLength`).

4.2 Numerical simulations

The control law still updates the input of every models at the external time-step $h_e = 10^{-5}$ s. The internal time-step h_i corresponds to the time-step "used" by the solver (in particular, if $h_e = h_i$, then $h = h_e = h_i$).

4.2.1 Internal Time-step $h_i = 10^{-5}$ s

- Figures 5 and 6 present the simulation of the controlled linear system[3] (4) resp. without uncertainties ($\nu_1 = \nu_2 = 1$) and with[4] $\nu_1 = [0.98, 1.02]$ and $\nu_2 = 1$.
- Figures 7 and 8 present the simulation of the controlled nonlinear system[5] (6) resp. without uncertainties ($\nu_1 = \nu_2 = 1$) and with[6] $\nu_1 = 1$ and $\nu_2 = [0.98, 1.02]$.

4.2.2 Internal Time-step $h_i = 10^{-10}$ s

Figure 9 presents the simulation of the controlled nonlinear system[7] (6) with $\nu_1 = \nu_2 = 1$.

Remarks : In open-loop, both systems (3) and (5) converge (i.e. $\sigma_{out} \to 0$ when $k \to$ `simLength`). In closed-loop, only the first simulation (Fig. 5) seems to remain stable; in Figs. 6, 7, 8 and 9, the simulations have been stopped due to an *a priori* divergence of σ_{out} (the simulations are stopped when $\sigma_{out} > 0.3$).

[3] Initial simulation kernel : `Function ydot(y, Return((Amat*y[0] + Bmat*y[1]), Interval(0.0)))` with `simulation simu = simulation(&vdp, 1e-5, RK4, 1e-5)`.

[4] ν_1 is defined as `Interval(0.99, 1.02)`.

[5] Initial simulation kernel : `Function ydot(y, Return((Amat*y[0] + Bmat*y[1] - 10*y[0]*y[0])/(sin(pi/2 + y[0])), Interval(0.0)))` with `simulation simu = simulation(&vdp, 1e-5, RK4, 1e-5)`.

[6] ν_2 is defined as `Interval(0.99, 1.02)`.

[7] Main change in the simulation kernel : `simulation simu = simulation(&vdp, 1e-5, RK4, 1e-10)`.

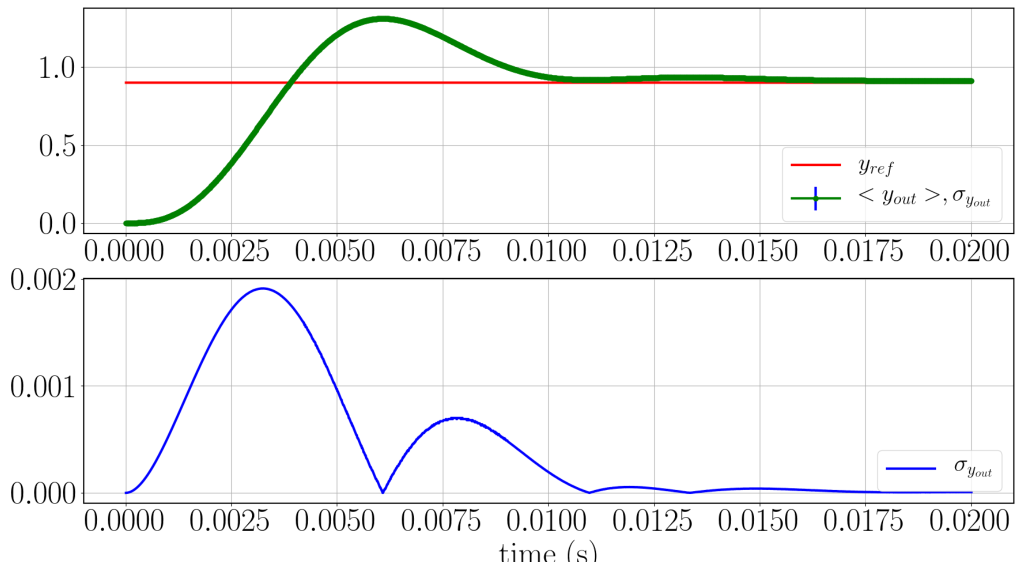


Figure 5: Simulation of the controlled linear system (4) with $\nu_1 = \nu_2 = 1$ ($h_i = 10^{-5}$ s).

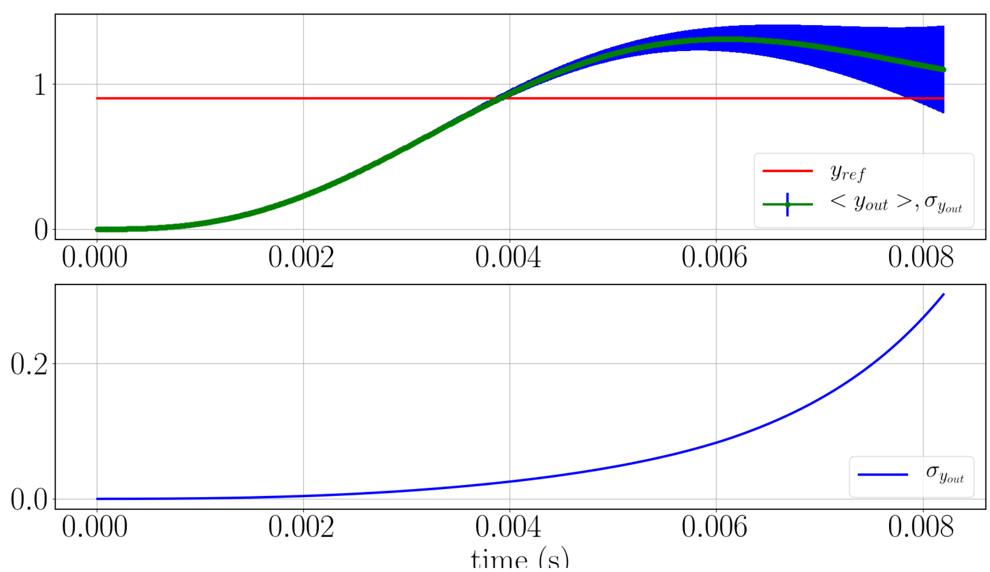


Figure 6: Simulation of the controlled linear system (4) with $\nu_1 = [0.98, 1.02]$ and $\nu_2 = 1$ ($h_i = 10^{-5}$ s).

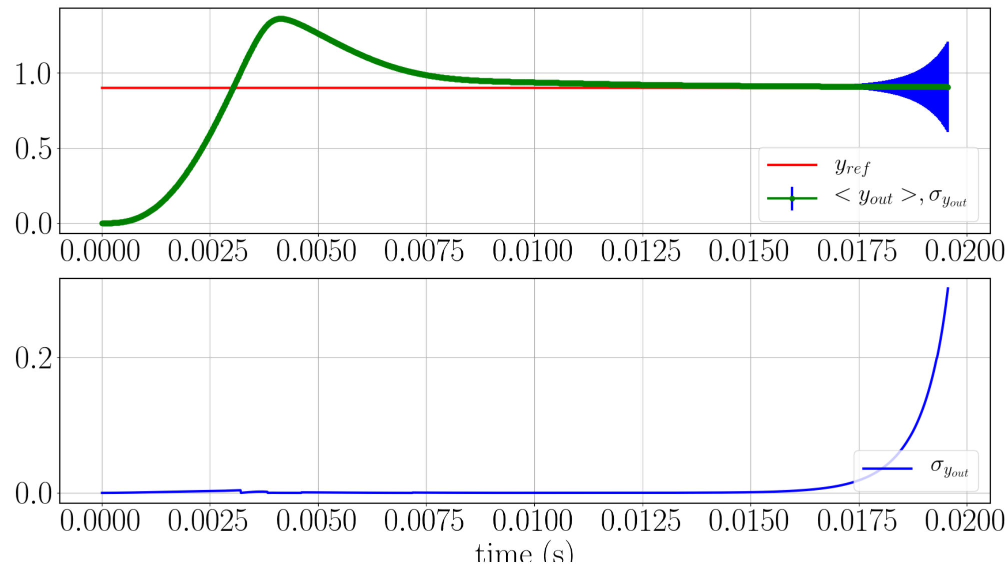


Figure 7: Simulation of the controlled nonlinear system (6) with $\nu_1 = \nu_2 = 1$ ($h_i = 10^{-5}$ s).

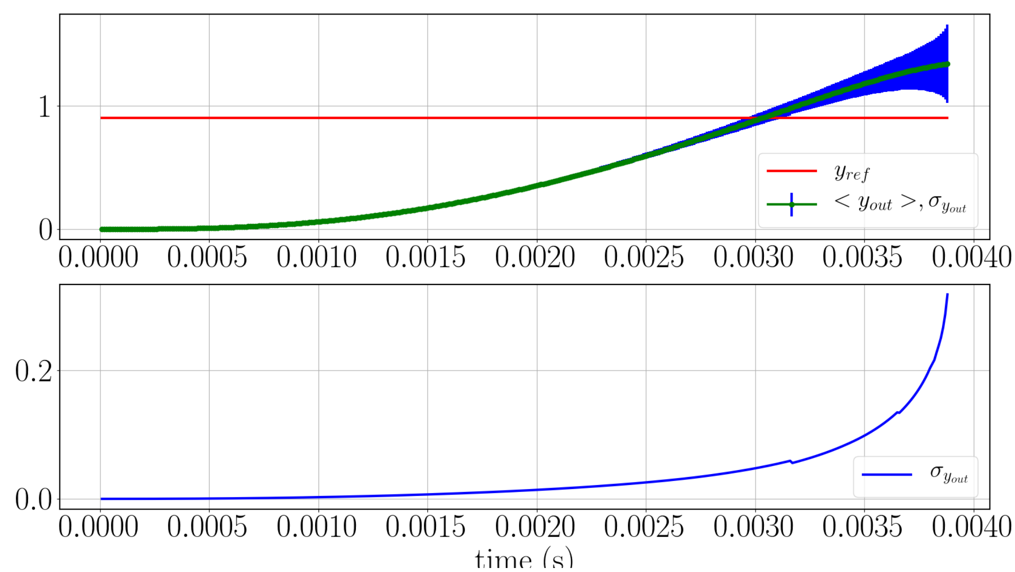


Figure 8: Simulation of the controlled nonlinear system (6) with $\nu_1 = 1$ and $\nu_2 = [0.98, 1.02]$ ($h_i = 10^{-5}$ s).

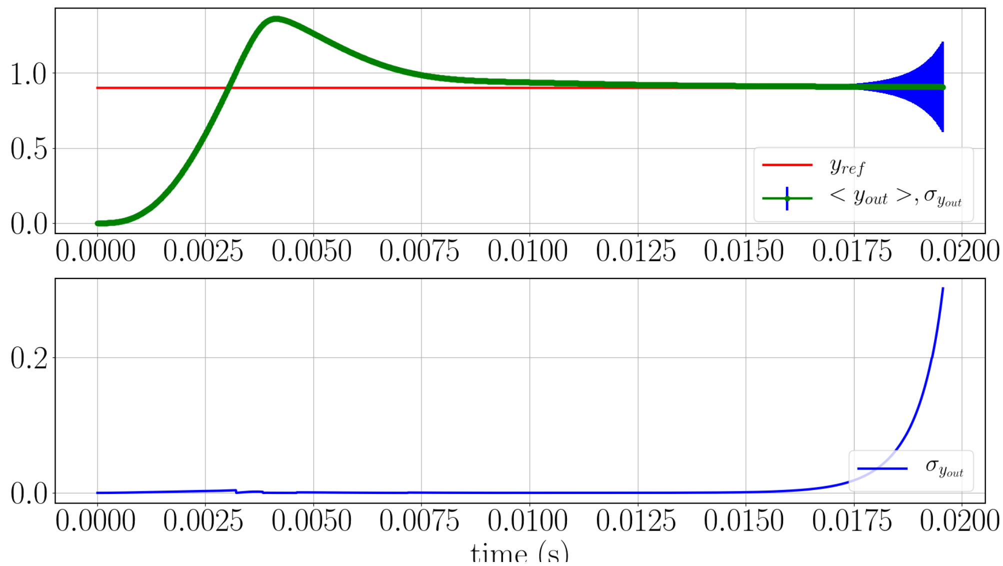


Figure 9: Simulation of the controlled nonlinear system (6) with $\nu_1 = \nu_2 = 1$ ($h_i = 10^{-10}$ s).

In Figures 6, 7, 8 and 9, the center $< y_{out} >$ seems to converge, despite the divergence of σ_{out}.

5 Analysis with FLUCTUAT

5.1 Configuration and assumptions

We set the main parameters[8] to:

MPFR bits	60
Initial unfolding **(IU)**	10
Cyclic unfolding **(CU)**	10
Depth inter-loop	-1
Widening Threshold	5
Narrowing Threshold	2

Denote `simLength` the (expected) simulation time index corresponding to $h = 10^{-5}$ s. The following "while" loop performs the simulation of the controlled linear system, for which an explicit Euler scheme is used. In order to have a reasonable simulation time, we should limit `simLength` and therefore, only the fast resonant mode (Fig. 2) is considered.

[8]Following the example given sec. 4.3, p. 51 in the manual.

```
while (ii < simLength)
{

... [controller]

y = (1 + Amat*h)*y + Bmat*h*u;

}
```

To simplify the code execution, inside the controller (2), the exponential function[9] has been replaced by a pre-calculated numerical table that gives the corresponding values at each instant k.

5.2 Numerical simulations

We evaluate first some particular configurations regarding the choice of the values of IU and CU according to the simulation time and the resulting simulated error. This comparative study is presented in the Tab. 1 and we can observe that:

- Setting (IU = 100, CU = 100) in open-loop simulation ((3) with $\nu_1 = 1$ and `simLength` = 300) shows a pretty "strange" behavior of the simulated response (Fig. 10).
- Setting (IU = 100, CU = 100) in closed-loop simulation seems to highlight a particular value of `simLength` that defines a "threshold" from which the simulation seems to diverge (`simLength` = 200).
- Setting (IU = 1000, CU = 1000) in closed-loop simulation ((4 with $\nu_1 = \nu_2 = 1$ and `simLength` = 500) seems to give a good compromise that may ensure the relative stability over a long period of simulation time, despite an increasing error (Figs. 11 and 12).

[9]The usual exp. function has been tested using the library "fluctuat-math.h" and the following warning occurs: `Rounding error of library function to be specified - level sure`.

exec. time (s) vs. config.	SimLength					
	150	200	201	250	300	500
closed-loop (IU = 1, CU = 10)		37				37
closed-loop (IU = 100, CU = 100)	57	90	3450	3589		
closed-loop (IU = 1000, CU = 1000)					350	2250
open-loop (IU = 100, CU = 100)		32			70	
open-loop (IU = 1000, CU = 1000)					37	

Table 1: Evaluation of the execution time according to the `simLength` value and the current software configuration; values in green denote a simulation that *a priori* converges and values in red denote *a priori* a divergence.

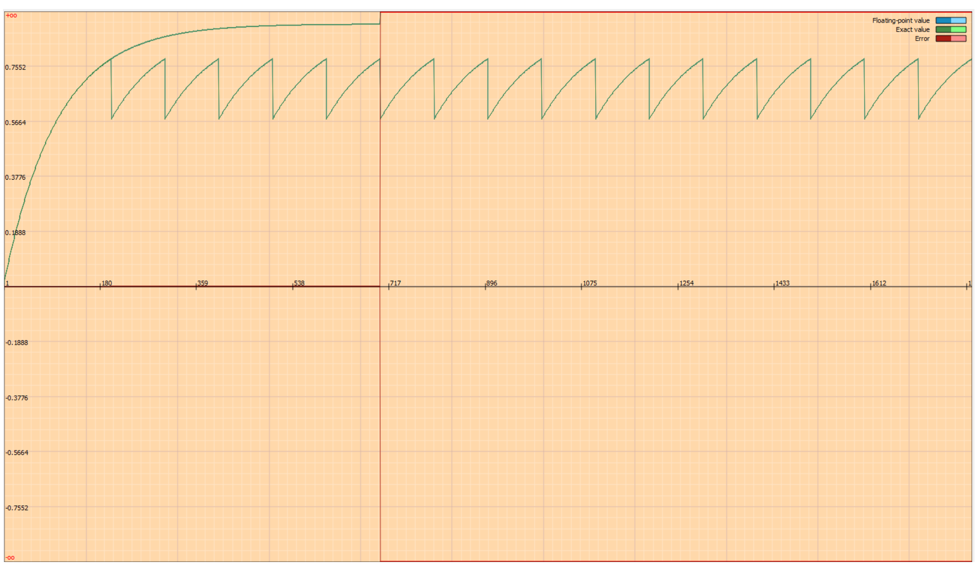
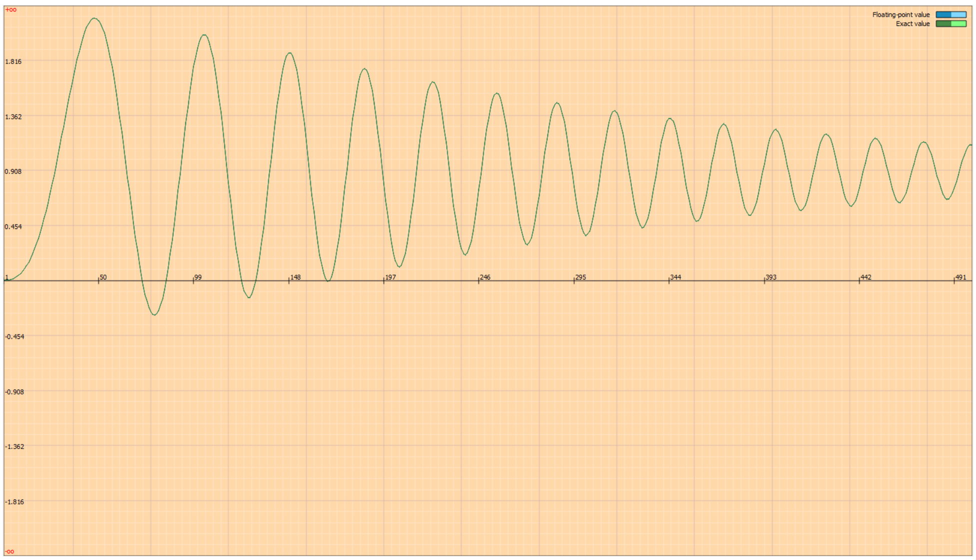
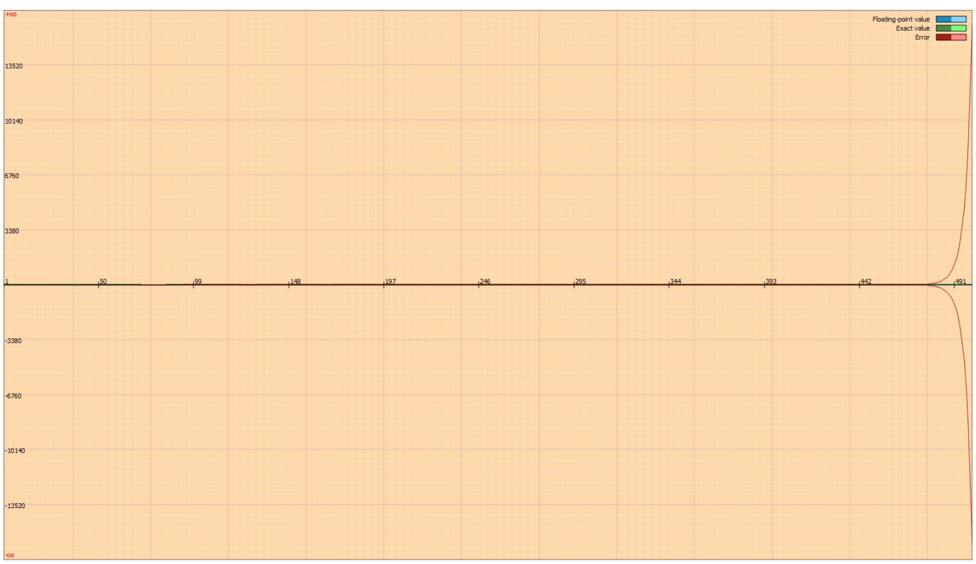


Figure 10: Open-loop simulation - Evolution of the floating point value, the exact value and the error.

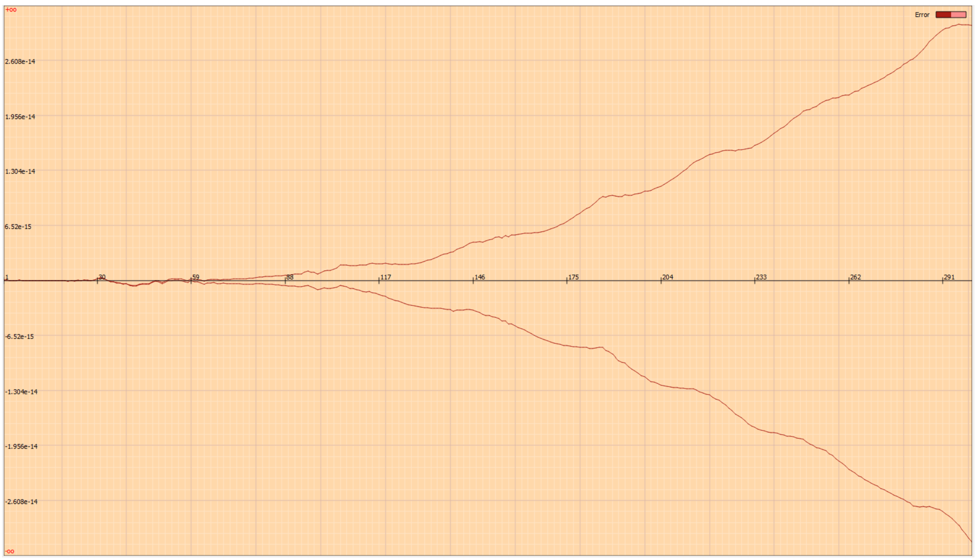


(a) Evolution of the floating point value and exact value.



(b) Evolution of the error.

Figure 11: Closed-loop simulation.



(a) Evolution of the error.

(b) Evolution of the relative error.

Figure 12: Zoom on the transient of the closed-loop simulation.

References

[1] M. Fliess and C. Join, "Model-free control", International Journal of Control, vol. 86, issue 12, pp. 2228-2252, Jul. 2013 (available at http://arxiv.org/pdf/1305.7085.pdf).

[2] K.J. Åström and P.R. Kumar, "Control: A perspective", Automatica, vol. 50, issue 1, pp. 3-43, Jan. 2014 (available at http://www.sciencedirect.com/science/article/pii/S0005109813005037).

[3] L. Michel, "A para-model agent for dynamical systems", preprint arXiv, Oct. 2016 (available at http://arxiv.org/abs/1202.4707).

[4] L. Michel *et al.*, "Model-Free Based Digital Control for Magnetic Measurements", preprint arXiv, Mar. 2017 (available at https://arxiv.org/abs/1703.05395).

[5] dynIBEX library http://perso.ensta-paristech.fr/~chapoutot/dynibex/

[6] FLUCTUAT software http://www.lix.polytechnique.fr/~putot/fluctuat.html