



HAL
open science

”Intelligent” controllers on cheap and small programmable devices

Cédric Join, Frédéric Chaxel, Michel Fliess

► **To cite this version:**

Cédric Join, Frédéric Chaxel, Michel Fliess. ”Intelligent” controllers on cheap and small programmable devices. 2nd International Conference on Control and Fault-Tolerant Systems, SysTol’13, Oct 2013, Nice, France. pp.CDROM, 10.1109/SysTol.2013.6693811 . hal-00845795

HAL Id: hal-00845795

<https://hal-polytechnique.archives-ouvertes.fr/hal-00845795>

Submitted on 17 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

“Intelligent” controllers on cheap and small programmable devices

Cédric Join^{a,b,d}, Frédéric Chaxel^b and Michel Fliess^{a,c}

Abstract— It is shown that the “intelligent” controllers which are associated to the recently introduced model-free control synthesis may be easily implemented on cheap and small programmable devices. Several successful numerical experiments are presented with a special emphasis on fault tolerant control. **Keywords**— Model-free control, intelligent PID controllers, small programmable devices, estimation, identification, noise attenuation, fault tolerant control.

I. INTRODUCTION

It is well known that the overwhelming majority of industrial control applications is based on PID controllers (see, e.g., [1], [17], and the references therein). Those controllers are often manufactured by numerous companies as microcontrollers on cheap and small programmable devices, like a *Microchip*¹ PIC or a *Freescale*² DSC for instance. This communication demonstrates that the recent *model-free control* [6] and the corresponding *intelligent PID controllers* [6] may also be implemented on such devices,

- thanks to the low power computation cost which they require,
- although they obey to mathematical principles, which are quite different from those of “classic” PIDs.

When compared to a PID regulator, its intelligent counterpart contains one term more, which subsumes not only the unknown structure of the plant but also the unknown disturbances. This term, which is estimated online thanks to recent parameter identification techniques ([8], [9]), is also found in the linear *ultra-local* model which

- replaces the unknown global description of the plant,
- is continuously updated online via the same calculations,
- is of low order, which is most of the time equal to 1.

Remark 1.1: Let us emphasize moreover that the strange ubiquity of classic PIDs was mathematically explained for

^a A.L.I.E.N. (ALgèbre pour Identification et Estimation Numériques), 24-30 rue Lionnois, BP 60120, 54003 Nancy, France.

{cedric.join, michel.fliess}@alien-sas.com

^b CRAN (CNRS, UMR 7039), Université de Lorraine, BP 239, 54506 Vandœuvre-lès-Nancy, France.

{cedric.join, frederic.chaxel}@univ-lorraine.fr

^c LIX (CNRS, UMR 7161), École polytechnique, 91128 Palaiseau, France. Michel.Fliess@polytechnique.edu

^d Projet Non-A, INRIA Lille – Nord-Europe, France.

¹Let us quote Wikipedia: *Microchip Technology* is an American manufacturer of microcontroller, memory and analog semiconductors.

²Let us quote again Wikipedia: *Freescale Semiconductor* is an American company that produces and designs embedded hardware.

the first time, to the best of our knowledge, thanks to intelligent PIDs [6].

This aim is fully justified by the following facts:

- Intelligent PIDs are much easier to tune than the classic ones.
- They are robust with respect to most disturbances, including quite strong ones.
- They permit straightforward fault accommodations.
- Many successful concrete applications were already achieved in most various domains within a few years (see the numerous references in [6]).

Our paper is organized as follows. Section II summarizes some of the most important facts about model-free control and its corresponding intelligent controllers. The implementation on small programmable devices is detailed in Section III. Section IV describes some numerical experiments, with a peculiar emphasis on fault tolerant control, *i.e.*, on an important topic in control engineering (see, e.g., [3], [16], and the references therein). Several excellent simulations are provided. Some concluding remarks may be found in Section V.

II. MODEL-FREE CONTROL: BASICS³

A. The ultra-local model

The unknown global description of the plant is replaced by the *ultra-local model*

$$y^{(\nu)} = F + \alpha u \quad (1)$$

where

- the derivation order $\nu \geq 1$ is selected by the practitioner;
- $\alpha \in \mathbb{R}$ is chosen by the practitioner such that αu and $y^{(\nu)}$ are of the same magnitude.

Remark 2.1: Note that ν has no connection with the order of the unknown system, which may be with distributed parameters, *i.e.*, which might be best described by partial differential equations (see, e.g., [13] for hydroelectric power plants).

Remark 2.2: The existing examples show that ν may always be chosen quite low, *i.e.*, 1 or 2. In almost all existing concrete case-studies $\nu = 1$. The only counterexample until now where $\nu = 2$ is provided by magnetic bearings [4] where frictions are almost negligible.⁴

³See [6] for more details.

⁴See the explanation in [6].

Some comments on F are in order:

- F is estimated via the measure of u and y ;
- F subsumes not only the unknown structure of the system but also any perturbation.

B. Intelligent PIDs

Set $\nu = 2$ in Equation (1):

$$\ddot{y} = F + \alpha u \quad (2)$$

Close the loop via the *intelligent proportional-integral-derivative controller*, or *iPID*,

$$u = -\frac{F - \dot{y}^* + K_P e + K_I \int e + K_D \dot{e}}{\alpha} \quad (3)$$

where

- $e = y - y^*$ is the tracking error,
- K_P, K_I, K_D are the usual tuning gains.

Combining Equations (2) and (3) yields

$$\ddot{e} + K_D \dot{e} + K_P e + K_I \int e = 0$$

where F does not appear anymore. The tuning of K_P, K_I, K_D is therefore quite straightforward. This is a major benefit when compared to the tuning of “classic” PIDs (see, e.g., [1], [17], and the references therein).

Remark 2.3: If $K_I = 0$ we obtain the *intelligent proportional-derivative controller*, or *iPD*,

$$u = -\frac{F - \dot{y}^* + K_P e + K_D \dot{e}}{\alpha}$$

Set now $\nu = 1$ in Equation (1):

$$\dot{y} = F + \alpha u \quad (4)$$

The loop is closed by *intelligent proportional-integral controller*, or *iPI*,

$$u = -\frac{F - \dot{y}^* + K_P e + K_I \int e}{\alpha} \quad (5)$$

If $K_I = 0$, it yields an *intelligent proportional controller*, or *iP*,

$$u = -\frac{F - \dot{y}^* + K_P e}{\alpha} \quad (6)$$

Remark 2.4: Equation (6) and the corresponding iP are most common in practice. This is again a major simplification with respect to “classic” PIDs and PIs.

C. Estimation of F

F in Equation (1) is assumed to be “well” approximated by a piecewise constant function F_{est} . According to the algebraic parameter identification developed in [8], [9], rewrite, if $\nu = 1$, Equation (4) in the operational domain (see, e.g., [23])

$$sY = \frac{\Phi}{s} + \alpha U + y(0)$$

where Φ is a constant. We get rid of the initial condition $y(0)$ by multiplying both sides on the left by $\frac{d}{ds}$:

$$Y + s \frac{dY}{ds} = -\frac{\Phi}{s^2} + \alpha \frac{dU}{ds}$$

Noise attenuation is achieved by multiplying both sides on the left by s^{-2} .⁵ It yields in the time domain the realtime estimate

$$F_{\text{est}}(t) = -\frac{6}{\tau^3} \int_{t-\tau}^t [(\tau - 2\sigma)y(\sigma) + \alpha\sigma(\tau - \sigma)u(\sigma)] d\sigma$$

where $\tau > 0$ might be quite small. This integral may of course be replaced in practice by a classic digital filter.

III. IMPLEMENTATION

Let us remind that implementing controllers on small programmable devices is a well established topic in engineering (see, e.g., [11], [12], [18], [20], [21], [22], and the references therein).

A. The iP device

We first detail the implementation of the iP device, which is most of the time sufficient in practice.

1) *Device:* The device is a microchip dsPIC33FJ128GP204 characterized by:⁶

- Architecture: 16-bit.
- CPU speed (MIPS): 40.
- Memory type: Flash.
- Program memory (KB): 128.

We also utilize

- two inputs with a 12-bit, 500 KSPS analog-to-digital conversion,
- one 12-bit digital output coupled with an external digital-to-analog converter.

The values of the input and output variables, which are expressed in volts, are in the range $[0, 3.3]$. They can be connected however to an electronic amplifier stage. Let us add that our material is cheap: it costs less than 5 euros.

Figure 1 displays the corresponding architecture:

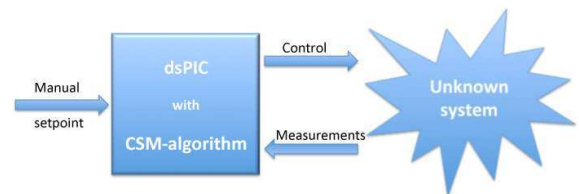


Fig. 1: Control scheme

2) *Simple calculations:* A control iteration needs:

- affectations: 19,
- conditions (*if...then...else*): 5,
- additions/subtractions: 14,
- multiplications/divisions: 16.

The computational power which is used is indeed very low.

⁵See [5] for a theoretical explanation.

⁶See <http://www.microchip.com> for more technical details, and also the prices.

B. Some extensions

Some new calculations are of course needed for iPIs, iPIDs, and iPDs.

Remark 3.1: An iPD was employed only once, for magnetic bearings [4]. Note moreover that it was not necessary until now to use iPIDs in practice!

The integrals $\int e$ appearing in iPIs and iPIDs may be dealt via the classic trapezoidal rules (see, e.g., [2]). Less than 10 basic operations are used.

The derivative $D(t) = K_D \dot{e}$ in iPIDs and iPDs may be obtained via

- a backward Euler difference scheme (see, e.g., [2])

$$D(t_k) = K_D \frac{e(t_k) - e(t_{k-1})}{t_k - t_{k-1}}$$

- a low-pass digital filter for reducing the noise (see, e.g., [10]).

As above for the integration, less than 10 basic operations are used.

Remark 3.2: With very noisy signals, more advanced tools might be necessary for the differentiation (see [7], [15], and [14]).

IV. NUMERICAL EXPERIMENTS

A. LabVIEW

As well known, LabVIEW greatly facilitates numerical simulations in engineering and in science.⁷ LabView can be programmed in order to control a real system by the use of an input-output device (analogic and logic inputs and outputs). Simple as well as advanced regulation strategies may therefore be assessed. It may also be used for emulating/simulating the behavior of a real system (see, e.g., [19]). Emulation is achieved in our experimental platform via Labview and an input-output card (see Figure 2).

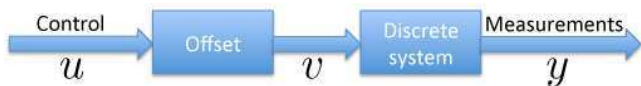


Fig. 2: LabVIEW details

Note that v becomes the true input control variable, where $-1.65 \leq v \leq 1.65$. The non-negativity condition on u (see Section III-A.1) is therefore dropped. Negative values for the control variables are now possible. Figure 3 displays a full description of our devices where the dsPIC is connected to an interface from National Instrument.⁸

B. Two experiments

In both experiments the same iP is implemented where $\alpha = 1$ and $K_P = 1$. The control variable is obtained thanks to the dsPIC with a sample time equal to 0.1ms.

⁷See <http://www.ni.com/labview/whatis> for more details.

⁸See <http://www.ni.com/white-paper/2732/en> for details.

1) *System 1:* Consider the nonlinear stable system:

$$2(3\dot{v}v^2 + v^3) = 0.5\ddot{y} + 0.5\dot{y} - y$$

where v is the saturated control after offset (see Figure 2 for an explanation). The tracking performances, which are presented in Figure 4, are quite good. They however deteriorate with important change points. This is due to the saturation of the control variable which is bounded.

Set

$$\bar{v} = \pi v$$

where

- $\pi = 1$ corresponds to the fault-free case,
- Figure 5-(c) shows the case $0 \leq \pi \leq 1$, which corresponds to a power loss of the actuator.

Figures 5-(a) and 5-(b) display an excellent fault tolerant control even with violent faults.

2) *System 2:* Consider the nonlinear unstable system:

$$9\dot{v}v^2 + v^3 = \ddot{y} + \dot{y} + y$$

Close the loop with the same iP as above. Figure 6 displays excellent tracking and fault accommodation performances.

V. CONCLUSION

This communication has demonstrated that the intelligent controllers, which are associated to model-free control, may easily be implemented on cheap and small programmable devices. Future applications will show that our academic numerical experiments may be easily extended to more realistic case-studies. It should lead to great industrial opportunities for this new setting.

Acknowledgment: The authors thank the department of *Génie Électrique et Informatique Industrielle, Institut Universitaire de Technologie Nancy-Brabois, Université de Lorraine*, for its most friendly help and its loan of some essential devices.

REFERENCES

- [1] K.J. Åström, T. Hägglund, *Advanced PID Control*, Instrument Soc. Amer., 2006.
- [2] K.E. Atkinson, *An Introduction to Numerical Analysis* (2nd ed.), Wiley, 1989.
- [3] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, Springer, 2003.
- [4] J. De Miras, S. Riachy, M. Fliess, C. Join, S. Bonnet, Vers une commande sans modèle d'un palier magnétique, *7^e Conf. Internat. Francoph. Automatique*, Grenoble, 2012. Available at <http://hal.archives-ouvertes.fr/hal-00682762/en/>
- [5] M. Fliess, Analyse non standard du bruit, *C.R. Acad. Sci. Paris Ser. I*, vol. 342, pp. 797–802, 2006.
- [6] M. Fliess, C. Join, Model-free control, *Int. J. Control*, 2013, DOI: 10.1080/00207179.2013.810345. Preprint available at <http://hal.archives-ouvertes.fr/hal-00828135/en/>
- [7] M. Fliess, C. Join, H. Sira-Ramírez, Non-linear estimation is easy, *Int. J. Model. Identif. Control*, vol. 4, pp. 12–27, 2008. Preprint available at <http://hal.archives-ouvertes.fr/inria-00158855/en/>
- [8] M. Fliess, H. Sira-Ramírez, An algebraic framework for linear identification, *ESAIM Control Optimiz. Calc. Variat.*, 9: 151–168, 2003.
- [9] M. Fliess, H. Sira-Ramírez, Closed-loop parametric identification for continuous-time linear systems via new algebraic techniques, in eds. H. Garnier and L. Wang, *Identification of Continuous-time Models from Sampled Data*, Springer, pp. 362–391, 2008.

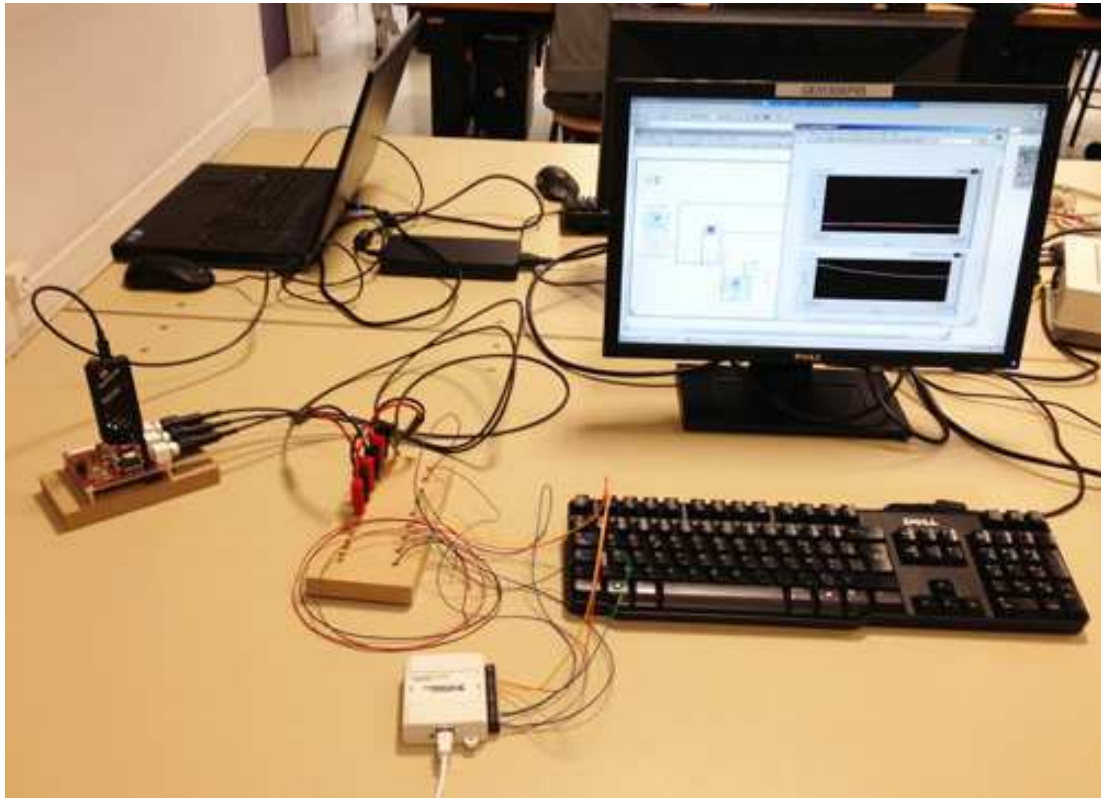
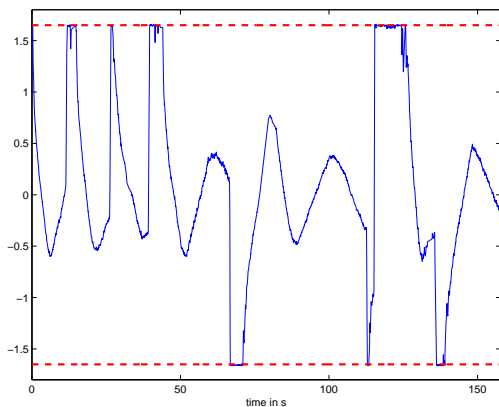
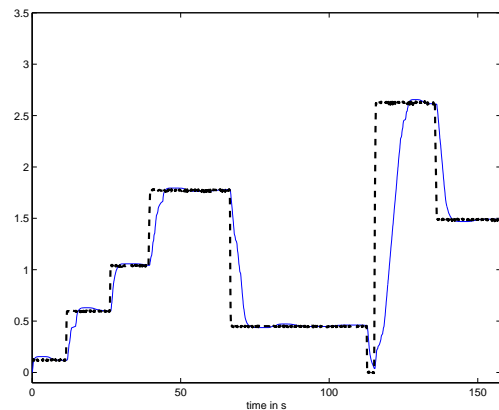


Fig. 3: Our experimental setup with a dSPIC connected to an input-output card driven by LabVIEW



(a) Control (– blue) and control limits (– – red)

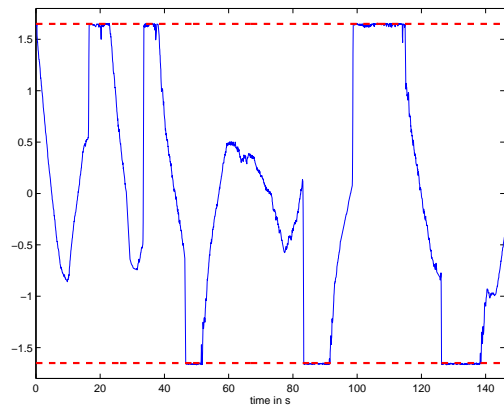


(b) Setpoint (– – black) and output (– blue)

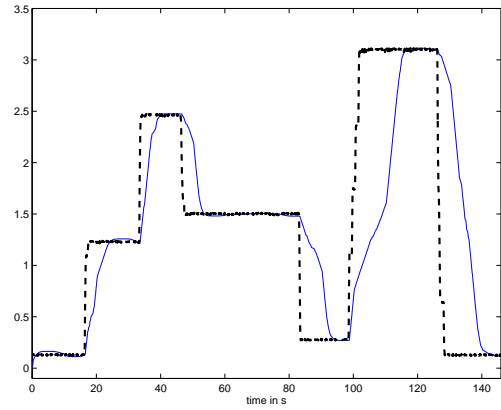
Fig. 4: Experimental results: fault-free case

- [10] G.F. Franklin, M.L. Workman, D. Powell, *Digital Control of Dynamic Systems* (3rd ed.), Addison-Wesley, 1997
- [11] A.P. Godse, D.A. Godse, *Microprocessors and Microcontrollers* (6th revised ed.), Technical Editions Pune, 2009.
- [12] D. Ibrahim, *Microcontroller Based Applied Digital Control*, Wiley, 2006.
- [13] C. Join, G. Robert, M. Fliess, Vers une commande sans modèle pour aménagements hydroélectriques en cascade, 6^e Conf. Internat. Francoph. Automat., Nancy, 2010. Available at <http://hal.archives-ouvertes.fr/inria-00460912/en/>
- [14] D.Y. Liu, O. Gíbaru, W. Perruquetti, Error analysis of Jacobi derivative

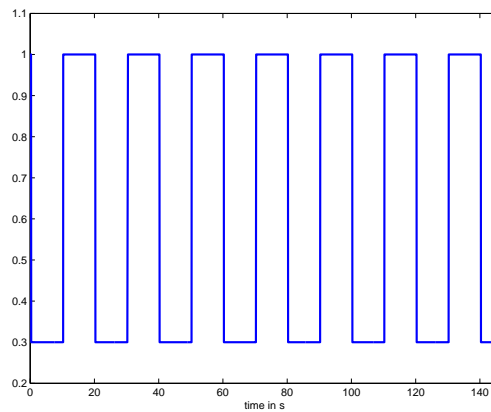
- estimators for noisy signals, *Numer. Algor.*, 58, 53–83, 2011.
- [15] M. Mboup, C. Join, M. Fliess, Numerical differentiation with annihilators in noisy environment, *Numer. Algor.*, 50, 439–467, 2009.
- [16] H. Noura, D. Theilliol, J.-C. Ponsart, A. Chamseddine, *Fault-tolerant Control Systems - Design and Practical Applications*, Springer, 2009.
- [17] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules* (3rd ed.), Imperial College Press, 2009.
- [18] J.S. Parab, V.G. Shelake, R.K. Kamat, G.M. Naik, *Practical Aspects of Embedded System Design Using Microcontrollers*, Springer, 2008
- [19] A. St. Leger, A. Deese, J. Yakaski, C. Nwankpa, Controllable analog emulator for power system analysis, *Int. J. Electrical Power Energy*



(a) Control (– blue) and control limits (– – red)

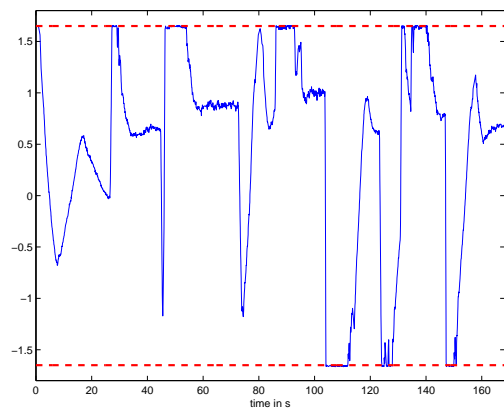


(b) Setpoint (– – black) and output (– blue)

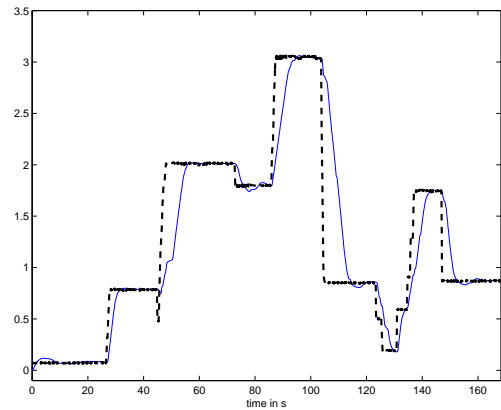


(c) Multiplicative power loss

Fig. 5: Experimental results: faulty case

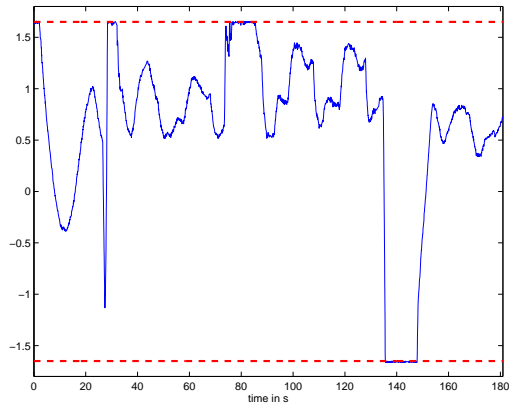


(a) Control (–,blue) and control limits (– –,red)

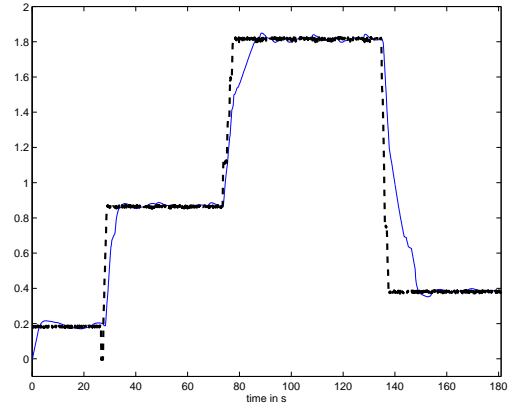


(b) Setpoint (– – black) and Output (– blue)

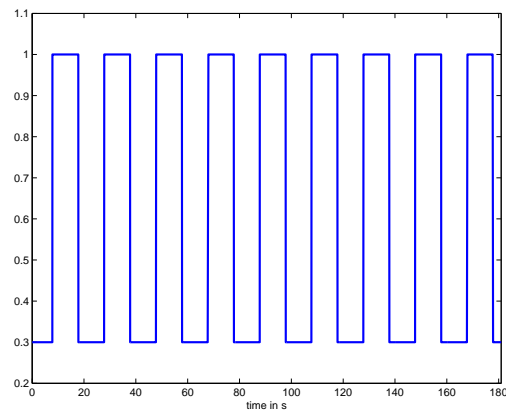
Fig. 6: Experimental results: fault free case



(a) Control (– blue) and control limits (– – red)



(b) Setpoint (– – black) and output (– blue)



(c) Multiplicative power loss

Fig. 7: Experimental results: faulty case

facings (3rd ed.), Cengage Learning, 2011.

- [22] T. Wilmshurst, *Designing Embedded Systems with PIC Microcontrollers: Principles and Applications*, Elsevier, 2007.
- [23] K. Yosida, *Operational Calculus* (translated from the Japanese), Springer, 1984.