International Conference on Control, Engineering & Information Technology (CEIT'14)
Proceedings - Copyright IPCO-2014, pp.244-252
ISSN 2356-5608

Implementation of a fixed low order controller on STM32 microcontroller

Maher Ben Hariz
Université de Tunis El Manar, Facuté des Sciences de Tunis, Ecole Nationale d'Ingénieurs de Tunis
Laboratoire Analyse, Conception et Commande des Systèmes, LR11ES20
Tunis, Tunisia
maherbanhariz@gmail.com

Faouzi Bouani, Mekki Ksouri
Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis
Laboratoire Analyse, Conception et Commande des Systèmes, LR11ES20
Tunis, Tunisia
bouani.faouzi@yahoo.fr

Abstract— **The basic idea of this paper is to implement a fixed low order controller on a real electronic system by using the STM32 microcontroller. The principal aim of this controller is to guarantee some step response specifications such as the settling time and the overshoot. The controller parameters are obtained by the minimization of a non convex optimization problem. The resolution of this sort of problems may lead to a local solution. Hence, we are interested to use a global optimization method such as the GGP (Generalized Geometric Programming) method. The practical results show the effectiveness of the proposed algorithm.**

Keywords — Microcontroller; fixed low order controller; step response specifications; global optimization.

I. Introduction

Nowadays the use of microcontroller miniature devices is becoming remarkable. In fact, the devices are employed in several industrial applications such as medicine, automotive systems and transportation, aerospace, etc. Indeed, the progression of microcontrollers and the features that they combined with their speed, allow them to be more suitable for a wide variety of control applications. There is some works that have used fast devices to implement control algorithms [1], [2], [3] and [4]. It's in this context that is made our work which consists on the implementation of a linear system control algorithm on a STM32 microcontroller.

In works given in [5] and [6], authors have presented the synthesis of a fixed low-order controller for linear time invariant, Single Input Single Output (SISO) systems with some step response specifications such as the settling time and the overshoot.

The controller design is expressed as an optimization problem which takes in account the desired closed-loop performances. In [7], authors presented the methodology of fixing the desired closed loop characteristic equation by the user. Since, the controller parameters are obtained by minimizing a non-convex optimization problem. The use of global optimization method is suggested. In our work, we will apply the GGP method to resolve this optimization problem. The key idea of this method is to transform a non-convex optimization problem to a convex one by means of variable transformations. The main contribution of this paper is the implementation of the proposed algorithm on a real electronic system by the use of STM32 microcontroller.

The proposed control law was developed using Keil development tool specially-engineered for ARM processor based microcontroller devices.

The paper is organized as follows: In section II, the problem statement is exposed. The design of controllers is developed in section III. In section IV, the GGP method is introduced. Section V describes the hardware and the software development tools employed in this application. To bring out the effectiveness of the proposed controller, a practical implementation on a real system and a comparison with the PI controller is presented in section VI. The last section is devoted to conclude this paper.

II. Problem statement

A SISO system can be represented by a transfer function *G(s)*. The closed system is delineated in Fig. 1.

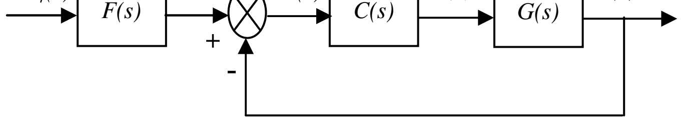

Fig. 1. A feedback control system.

$$G(s)=\frac{N(s)}{D(s)}=\frac{n_m s^m+n_{m-1}{}^{m-1}s^{m-1}+\cdots+n_0}{d_l s^l+d_{l-1}s^{l-1}+\cdots+d_0} \tag{1}$$

The controller is given by:

$$C(s)=\frac{B(s)}{A(s)} \tag{2}$$

where

$$\begin{aligned} A(s)&=s^t+a_{t-1}s^{t-1}+\cdots+a_1s+a_0\\ B(s)&=b_rs^r+b_{r-1}s^{r-1}+\cdots+b_1s+b_0 \end{aligned} \tag{3}$$

In the case of low-order controller, we have: $r\le t<l-1$.

In order to obtain a closed loop transfer function with a unit static gain, we introduce a polynomial *F(s)*:

$$F(s)=f_qs^q+f_{q-1}s^{q-1}+\cdots+f_1s+f_0 \tag{4}$$

By using Eq. 1 and Eq. 2 the closed-loop transfer function will be given by:

$$H(s)=\frac{F(s)B(s)N(s)}{A(s)D(s)+B(s)N(s)} \tag{5}$$

The closed-loop characteristic equation is given by the following relation:

$$\begin{aligned} \delta(s)&=A(s)D(s)+B(s)N(s)\\ &=\delta_ns^n+\delta_{n-1}s^{n-1}+\cdots+\delta_1s+\delta_0 \end{aligned} \tag{6}$$

where $n=l+t$.

The target, now, is to design a controller which will be able to ensure some step response performances such as the overshoot and the settling time.

Emphasize that the indented closed-loop characteristic equation must be specified by using the polynomial characteristics ratios which was expanded in [5], [6] and [7].

III. Design of controllers

We define the controller parameters vector by:

$$x=\begin{bmatrix}b_0 & \cdots & b_r & a_0 & \cdots & a_{t-1}\end{bmatrix}^T \tag{7}$$

The coefficients vectors of the closed-loop characteristic polynomial δ and the closed-loop desired polynomial $\overline{\delta}$ are respectively given by:

$$\delta=\begin{bmatrix}\delta_0 & \delta_1 & \cdots & \delta_{n-1} & \delta_n\end{bmatrix}^T \tag{8}$$

$$\overline{\delta}=\begin{bmatrix}\overline{\delta_0} & \overline{\delta_1} & \cdots & \overline{\delta_{n-1}} & \overline{\delta_n}\end{bmatrix}^T \tag{9}$$

The coefficient vector of the closed-loop characteristic polynomial δ can be expressed as a function of *x* by:

$$\delta=Px+q \tag{10}$$

where

$$P=\begin{bmatrix} n_0 & 0 & \cdots & 0 & d_0 & 0 & \cdots & 0\\ n_1 & n_0 & \cdots & 0 & d_1 & d_0 & \cdots & 0\\ \vdots & \vdots & & \vdots & d_2 & d_1 & \cdots & 0\\ n_m & n_{m-1} & \cdots & n_{m-r} & \vdots & \vdots & & \vdots\\ \vdots & \vdots & & \vdots & d_l & d_{l-1} & \cdots & d_{l-t+1}\\ 0 & 0 & \cdots & n_m & \vdots & \vdots & & \vdots\\ \vdots & \vdots & & \vdots & 0 & 0 & \cdots & d_l\\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$q=\begin{bmatrix}0 & \cdots & 0 & d_0 & \cdots & d_{l-t} & \cdots & d_l\end{bmatrix}^T$$

$$P\in R^{(n+1)\times(r+t+1)},\ x\in R^{r+t+1},\ q\in R^{n+1}$$

The controller parameters are determined so that the difference between δ and $\overline{\delta}$ is minimal. This can be obtained by the following weighted cost function:

$$\begin{aligned} f(x)&=\left[\delta-\overline{\delta}\right]^TW\left[\delta-\overline{\delta}\right]\\ &=x^T\left[P^TWP\right]x+2\left[\left(q-\overline{\delta}\right)^TWP\right]x\\ &+\left[\left(q-\overline{\delta}\right)^TW\left(q-\overline{\delta}\right)\right] \end{aligned} \tag{11}$$

where W is a weighting matrix.

It was clarified in [7] and [8] that the coefficients of lower powers of *s* in the transfer function are the most related to the step response. Then, the weighting matrix may be selected such that the weights for the low powers of *s* have greater values than those for higher powers.

The controller parameters can be obtained by the resolution of the following problem:

$$\min_x f(x) \tag{12}$$

The present optimization problem is non-convex. Thus, its resolution by a local approach may lead to a local solution. Therefore, to deal with this problem, the use of global optimization method is suggested.

IV. Generalized Geometric Programming method

A. Mathematical formulation of the GGP method

GGP problems are appear commonly in engineering design, management and chemical process industry (see e.g. [9], [10], [11], [12], [13] and [14]).The distinctive peculiarity of this method that it is dedicated to solve a class of non-convex non-linear programming problems with the objective function and the constraints are in polynomial forms. The mathematical formulation of a GGP problem with free variables is expressed as follows [15]:

$$\min_X Z(X) = \sum_{p=1}^{T_0} c_p z_p \tag{13}$$

where

$$\begin{aligned}
&z_p = x_1^{\alpha_{p1}} x_2^{\alpha_{p2}} \ldots x_n^{\alpha_{pn}},\ p = 1,\ldots,T_0\ ,\\
&X = (x_1, x_2, \ldots, x_m, x_{m+1}, \ldots, x_n),\ \underline{x_i} \le x_i \le \overline{x_i},\\
&x_i > 0,\ \text{for } 1 \le i \le m \text{ and } x_i \le 0,\ \text{for } m+1 \le i \le n,\\
&c_p \in \Re\ \alpha_{pi} \in \Re \text{ for } 1 \le i \le m,
\end{aligned} \tag{14}$$

α_{pi} is integer for $m+1 \le i \le n$ and $\underline{x_i}$ and $\overline{x_i}$ are, respectively, lower and upper bounds of continuous variables x_i. It should be noted that convexification strategy that will be introduced later is only applied for positive variables x_i because of the logarithmic/exponential transformation. Accordingly, this transformation necessitates substituting x_i by $\exp(y_i)$. Therefore, x_i must be strictly positive. Nevertheless, this is not a handicap because we can use simple translations of the variables to accomplish specifications for variables originally having negative values [16].

The GGP method has been suggested to find the global optimum based on variable transformations. This transformation technique allows the convexification of the objective function and the constraints. To begin with, some definitions are necessitated before presenting the convexification propositions and property.

Definition 1: A "*monomial*" function is a product of power terms and it can be given by:

$$f(X) = c\prod_{i=1}^{n} x_i^{p_i} \tag{15}$$

where c is a real constant and p_i can be negative or positive power for $1 \le i \le n$.

Definition 2: A "*signomial*" function is constituted of a sum with products of power terms, where each product with power terms is multiplied by a real constant [17]:

$$f(X) = \sum_{j=1}^{T} c_j \prod_{i=1}^{n} x_i^{p_{i,j}} \tag{16}$$

The constants c_j and powers $p_{i,j}$ for $1 \le i \le n$ and $1 \le j \le T$ can be positive or negative.

Definition 3: The function $f(X)$ is called a "*posynomial*", when all constants c_j, for $1 \le j \le T$, in a signomial function of Eq. 16 are positive.

Optimization problems that possess only signomial terms are called GGP problems.

We will introduce some convex analysis results before exposing the convexification rules in next subsection:

Proposition 1 [15]: A twice-differential function $f(X) = c\prod_{i=1}^{n} x_i^{p_i}$ is convex in $\Re_+^n$ for $c \ge 0$ if $p_i \le 0$.

Proposition 2 [15]: A twice-differential function $f(X) = c\prod_{i=1}^{n} x_i^{p_i}$ is convex in $\Re_+^n$ for $c \le 0$ if $p_i \ge 0$ and $(1 - \sum_{i=1}^{n} p_i) \ge 0$.

Property [12]: The function $c\exp\left(\sum_{i=1}^{n} p_i x_i\right)$ is convex in $\Re_+^n$ if $c \ge 0$ and $p_i \in \Re$.

B. Convexification strategy of the GGP method

The fundamental concept in the convexification strategy is to perform variable transformations to problem given by Eq. 13 that permits to convexify each monomial of the signomial depending on their signs. Because the objective function of the controller possesses only variables with positive powers, we will be interested only to this case in the transformation rules.

1) Positively signed term (c>0):

Let consider the function $f(X) = c\prod_{i=1}^{n} x_i^{p_i}$, where $p_i > 0$, new variable y_i are presented according to $x_i = \exp(y_i)$, $i=1,2,\ldots,n$. We can then establish the following equivalence:

$$f(X) = c\prod_{i=1}^{n} x_i^{p_i} = c\exp\left(\sum_{i=1}^{n} p_i y_i\right) \tag{17}$$

According to the Property previously presented, the signomial in the right hand is convex relatively to y_i. This transformation is called the exponential transformation.

2) Negatively signed term (c<0):

Let consider the function $f(X)=c\prod_{i=1}^{n} x_i^{p_i}$, where $p_i>0$ and $\left(1-\sum_{i=1}^{n} p_i\right)<0$, new variable z_i are presented according to $x_i = z_i^{\frac{1}{\beta}}$, *i=1,2,…,n,* where $\beta = \sum_{i-1}^{n} p_i$.

We can then establish the resulting equality:

$$f(X)=c\prod_{i=1}^{n} x_i^{p_i} = c\prod_{i=1}^{n} z_i^{\frac{p_i}{\beta}} \tag{18}$$

According to Proposition 2, the signomial in the right hand is now convex with respect to z_i, since all exponents are positive and their sum is equal to 1. Another solution to convexify $f(X)$ is to simply choose $\beta > \sum_{i=1}^{n} p_i$. This transformation is referred to as power transformation.

V. STM32 MICROCONTROLLER AND KEIL DEVELOPMENT TOOLS

In this section, we present the hardware and the software development tools used in this work.

A. STM32F100RB microcontroller

The choice to use the STM32 is based on the compromise between price (the STM32 discovery costs few dollars), low power consumption and performance. In fact, the STM32F100RB value line family includes the high-performance ARM Cortex™-M3 32-bit RISC core working at a 24 MHz frequency, high-speed embedded memories (Flash memory of 128 Kbytes size and SRAM of 8 Kbytes size), and an extensive range of enhanced peripherals and I/Os connected to two APB buses. All devices offer two 12-bit DACs, one 12-bit ADC, up to six general purpose 16-bit timers plus one PWM timer, as well as standard and advanced communication interfaces: two Serial Peripheral Interface SPIs and I2Cs, three Universal Synchronous Asynchronous Receiver Transmitter USARTs, and a USB [18]. In order to load the program into the STM32 device, the ST-LINK is used. The STM32F100 architecture is presented in Fig. 2.

B. Keil development tool

Keil which is a software development tool includes C/C++ compilers, debuggers, integrated environments, middleware, Real-Time Operating System (RTOS), simulation models, and evaluation boards for ARM, Cortex-M, Cortex-R4, 8051, C166, and 251 processor families. The used version, in this work, is the µVision 4. The µVision 4 screen supplies a menu bar for command entry, windows for source files, and a toolbar where we can choose command buttons dialog boxes, and information displays. The µVision 4 can open and view multiple source files simultaneously.

This version has two operating modes: Build Mode and Debug Mode.

- *Build Mode*: It's the standard working mode. It lets us to convert all the application files and to generate executable programs.
- *Debug Mode*: It supplies an effective debugger for testing applications.

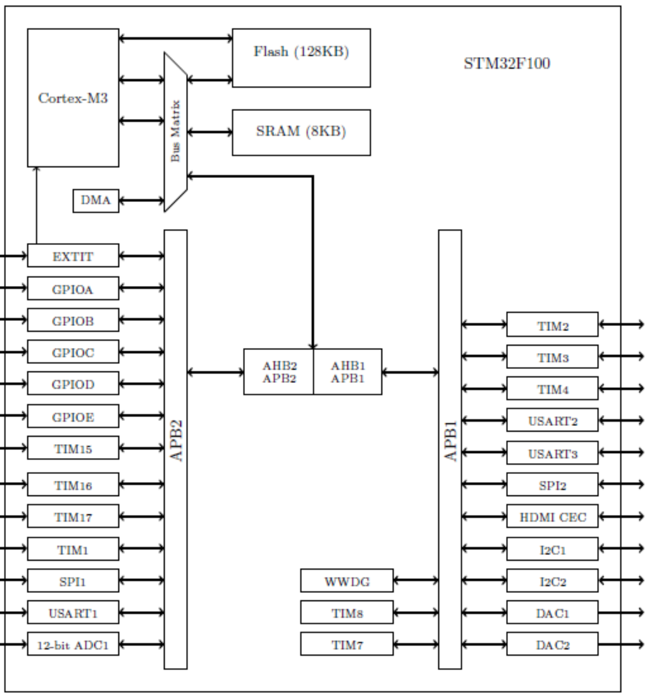

Fig. 2. STM32F100 architecture.

VI. PRACTICAL RESULTS

In order to show the performances of the presented approach, we make a practical implementation of the proposed controller on an electronic system by using an STM32 microcontroller.

A. System presentation

The practical system used in this work is a 3rd order low-pass filter as shown in Fig.3.

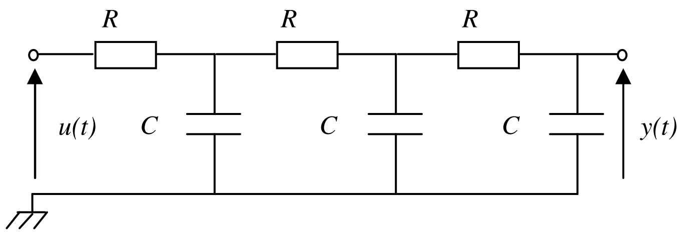

Fig. 3. 3rd order low-pass filter.

In our work we choose $R = 47\ K\Omega$ and $C = 50\ \mu F$. So, by adopting these values, the system transfer function will be given by:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{0.07705}{s^3 + 2.128s^2 + 1.086s + 0.07705} \quad (19)$$

B. Controllers design

To compare the closed-loop performances, we are interested in the design of two different types of digital controllers which are:

- Proportional Integral (PI) controller.
- Fixed low order controller.

1) Control law algorithm

The control law algorithm is explained by the following flowchart.

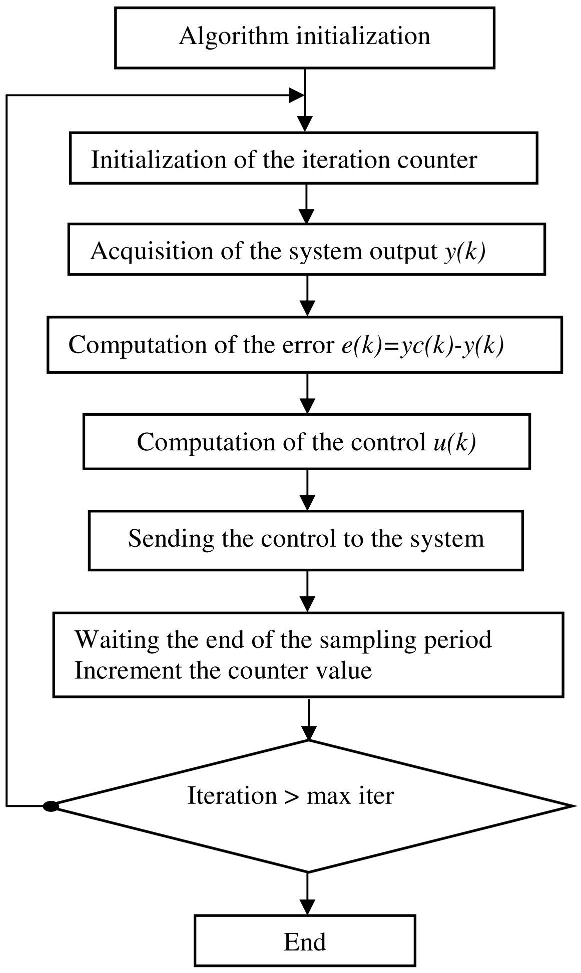
Algorithm initialization
Initialization of the iteration counter
Acquisition of the system output *y(k)*
Computation of the error *e(k)=yc(k)-y(k)*
Computation of the control *u(k)*
Sending the control to the system
Waiting the end of the sampling period
Increment the counter value
Iteration > max iter
End

Fig. 4. Control law algorithm.

By determining the system poles, we note that the dominant pole of the studied system is $p_1 = -0.084$. Thus, we can deduce the time constant *t* of this system which is given by:

$$t = -\frac{1}{p_1} = 12s \quad (20)$$

For the implementation of these two different types of controllers, the sampling period T_s will be chosen such that it is much lower than *t*.

2) PI controller:

With an aim of determining the PI controller parameters, we have used the first method of Ziegler-Nichols which takes into account the step response of the open loop system [19] as shown in Fig. 5.

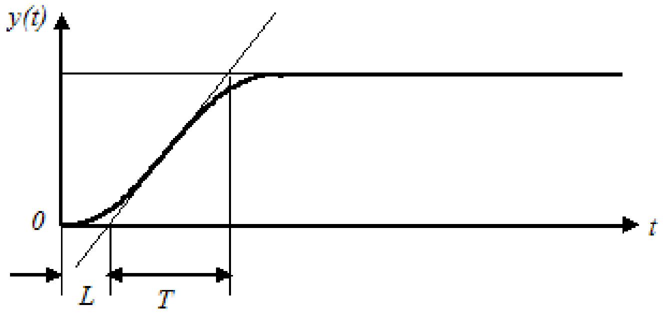

Fig. 5. Response curve for Ziegler-Nichols first method.

The controller parameters *Kp* and *Ki* are computed using the open loop step response. Parameters *L* and *T* and are determined by using the following expressions: $Kp = 0.9\frac{T}{L}$ and $Ki = \frac{L}{0.3}$.

The PI controller is expressed by:

$$C(s) = K_p + K_i \frac{1}{s} \quad (21)$$

By using this method, the PI controller will be given by:

$$C_1(s) = \frac{8.738s + 1.568}{s} \quad (22)$$

By considering this continuous PI controller and by applying the first order hold (foh) discretization method, we obtained the following digital PI controller:

$$C_1(z) = \frac{8.816z - 8.66}{z - 1} \quad (23)$$

Then, the control law obtained by this controller is given by:

$$u(k) = u(k-1) + 8.816e(k) - 8.66e(k-1) \qquad (24)$$

The evolutions of the set point, the output signal and the control signal, obtained with the PI controller are depicted in Figs. 6 and 7.

From these figures, we observe that the system output fluctuates around the set point and the control signal presents many variations. We remark, also, that the control signal is clipped because of the saturation conditions imposed to protect the microcontroller.

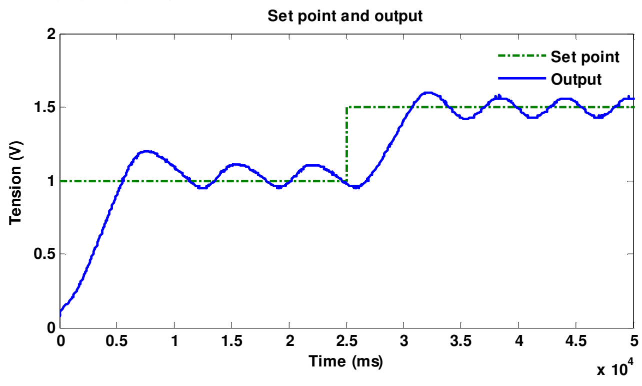

Fig. 6. Set point and output obtained with the PI controller.

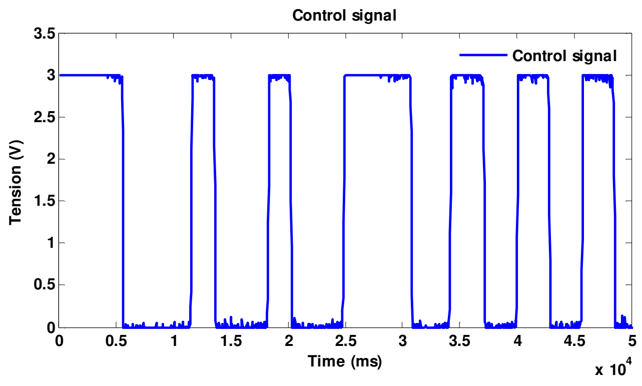

Fig. 7. Control signal obtained with the PI controller.

3) Fixed low order controller:

Let consider the following fixed second order controller:

$$\begin{cases} A(s) = s^2 + a_1 s + a_0 \\ B(s) = b_0 \end{cases} \qquad (25)$$

Assume that the objective is to synthesize a controller for the real system with the following specifications:

i) Overshoot $\leq 2\%$

ii) 2% settling time ≤ 20 s.

These performances can be obtained by choosing the parameters $\alpha_1 = 3.3$ and $\tau = 7$ which satisfy the design specifications. The resolution of the optimization problem by using the GGP method leads to the following controller:

$$C_2(s) = \frac{22.18}{s^2 + 6.874s + 18.815} \qquad (26)$$

This continuous fixed low order controller is discretize by using the 'foh' method. So, the digital fixed low order controller will be given by:

$$C_2(z) = \frac{0.0311z^2 + 0.105z + 0.022}{z^2 - 1.369\,z + 0.503} \qquad (27)$$

The recursive equation of the proposed controller is given by:

$$\begin{aligned} u(k) &= 1.369u(k-1) - 0.503u(k-2) \\ &+ 0.031e(k) + 0.105e(k-1) + 0.022e(k-2) \end{aligned} \qquad (28)$$

The evolutions of the output, the set point and the control signal, obtained by applying this controller to the real system, are shown in Figs. 8 and 9 respectively.

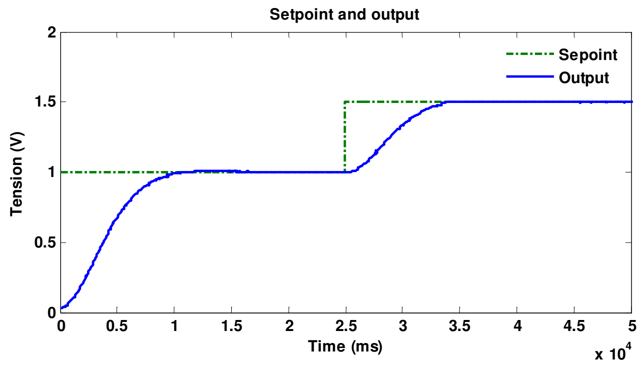

Fig. 8. Set point and output signals obtained with the fixed low order controller.

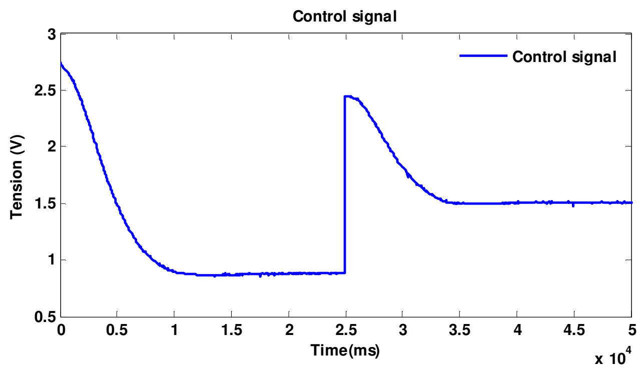

Fig. 9. Control signal obtained with the fixed low order controller.

It is remarkable that, despite the variation of the set point, the designed controller meets the specified performances which are a response with an overshoot about 2% and a settling time less than 20 s.

We note also that the control signal obtained with this last controller provides smoother variations compared to the control signals obtained with a PI controller.

Let the purpose be the design of a controller which will be able to guarantee the following specifications:

i) Overshoot $\leq 10\%$

ii) 2% settling time ≤ 15 s.

With an aim of achieving these specifications, we choose $\alpha_1 = 2.88$ and $\tau = 4.9$. The resolution of the optimization problem by applying the GGP method allows us to obtain the following controller:

$$C_3(s) = \frac{21.9}{s^2 + 0.365s + 6.16} \tag{29}$$

The digital fixed low order controller obtained by using the 'foh' method will be expressed by:

$$C_3(z) = \frac{0.809z^2 + 2.85\,z + 0.736}{z^2 + 0.597\,z + 0.833} \tag{30}$$

Then, the control law obtained by using this controller is given by:

$$\begin{aligned} u(k) &= 0.597u(k-1) - 0.833u(k-2) \\ &+ 0.809e(k) + 2.85e(k-1) + 0.736e(k-2) \end{aligned} \tag{31}$$

By using this controller with the real system, we obtain the output signal and the control signal plotted, respectively, in Figs. 10 and 11.

From Fig. 10, we note that the system output meets the desired requirements, (an overshoot less than or equals to 5% and a settling time of 15 s).

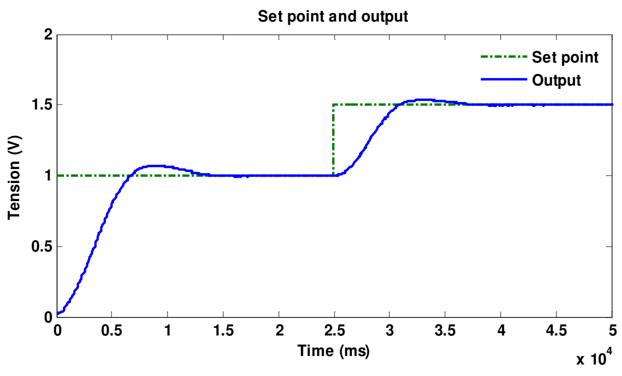

Fig. 10. Set point and output signals obtained with the second fixed low order controller.

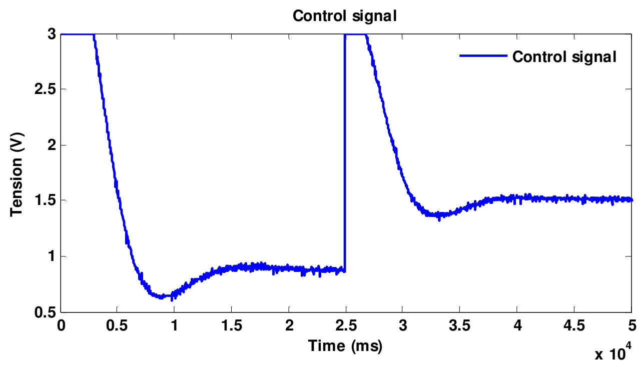

Fig. 11. Control signal obtained with the second fixed low order controller.

VII. Conclusion

The implementation of a fixed low order controller by the use of the STM32 microcontroller has been the object of this work. This controller is designed so as to ensure certain closed-loop performances. The controller parameters were obtained by solving a non-convex optimization problem using a global optimization method.

Moreover, the PI controller has been tested with the same microcontroller and compared with the proposed controller. Practical results show the effectiveness of the latter.

References

[1] K.V. Ling, S.P. Yue and J.M. Maciejowski "A fpga implementation of model predictive control," in Proceedings of the 2006 American Control Conference, 2006.

[2] Y. Jayaraman and U. Ravindran, "Fpga implementation of predictive control strategy for power factor correction," in Proceedings of World Academy of Science, Engineering and Technology, vol. 15, pp. 199–204, 2008.

[3] K. Ling, B. Wu, and J. Maciejowski, "Embedded model predictive control (mpc) using a fpga," in Proceedings of the 17th World Congress: The International Federation of Automatic Control, Seoul, Korea, 2008.

[4] A. Kheriji, F. Bouani, and M. Ksouri, "A Microcontroller Implementation of Constrained Model Predictive Control," International Journal of Electrical and Electronics Engineering vol. 5, pp. 272-279, 2011.

[5] L. Jin, Y. C. Kim "Fixed, low-order controller design with time response specifications using non-convex optimization," ISA Transactions, vol. 47, pp. 429–438, 2008.

[6] M. Ben Hariz, F. Bouani, M. Ksouri, "Robust controller for uncertain parameters systems," ISA transactions, vol. 51, pp. 632-640, 2012.

[7] Y. C. Kim, L. H. Keel, S. P. Bhattacharyya "Transient response control via characteristic ratio assignment," IEEE Transactions on Automatic Control, vol. 48, pp. 2238–2244, 2003.

[8] Y. Kim, K. Kim, S. Manabe "Sensitivity of time response to characteristic ratios". Proceeding of the 2004 American Control Conference Boston, Massachusetts," pp. 2723-2228, 2004.

[9] K. Nand, "Geometric programming based robot control design," Computers and industrial engineering, vol. 29, pp. 631-635, 1995.

[10] C. Chul and L. Dennis, "Effectiveness of a geometric programming algorithm for optimization of machining economics models," Computers and Operations Research, vol. 23, pp. 957-961, 1996.

[11] C. Maranas and C. Floudas, "Global optimization in generalized geometric programming," Computers and Chemical Engineering, vol. 21, pp. 351-369, 1997.

[12] R. Porn, K. Bjork, and T. Westerlund, "Global solution of optimization problems with signomial parts," Discrete optimization, vol. 5, pp. 108-120, 2007.

[13] J. Tsai, "Treating free variables in generalized geometric programming problems," Computers and Chemical Engineering, vol. 33, pp. 239-243, 2009.

[14] A. Kheriji, F. Bouani and M. Ksouri "A GGP approach to solve non convex min-max predictive controller for a class of constrained MIMO systems described by state-space models," International Journal of Control, Automation, and Systems, vol. 9, pp. 452-460, 2011.

[15] L. Liberti and N. Maculan, "Global Optimization From Theory to Implementation," Springer US, 2006.

[16] J. Tsai, M. Lin, and Y. Hu, "On generalized geometric programming problems with non positive variables," European Journal of Operational Research, vol. 178, pp. 10-19, 2007.

[17] K. Bjork, P. Lindberg, and T. Westerlund, "Some convexifications in global optimization of problems containing signomial terms," Computers and Chemical Engineering, vol. 27, pp. 669-679, 2003.

[18] STMicroelectronics "Datasheet stm32f100x4, stm32f100x6, stm32f100x8, stm32f100xb," Doc ID 16455 Rev 7, 2012.

[19] D. Lequesne "Régulation PID Analogique-numérique-floue. Hermès, Lavoisier, 2006.