

Hil validation of an embedded system acting as a nonlinear Takagi-Sugeno state observer on an Arduino board

M Feingesicht, Severus Constantin Olteanu, A Aitouche, L Belkoura

► **To cite this version:**

M Feingesicht, Severus Constantin Olteanu, A Aitouche, L Belkoura. Hil validation of an embedded system acting as a nonlinear Takagi-Sugeno state observer on an Arduino board. 6th European Embedded Design in Education and Research, Sep 2014, Milan, Italy. 10.1109/EDERC.2014.6924398 . hal-01248962

HAL Id: hal-01248962

<https://hal.inria.fr/hal-01248962>

Submitted on 29 Dec 2015

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.

HIL VALIDATION OF AN EMBEDDED SYSTEM ACTING AS A NONLINEAR TAKAGI-SUGENO STATE OBSERVER ON AN ARDUINO BOARD

*M. Feingesicht**, *S.C. Olteanu***, *A. Aitouche**, and *L. Belkoura***

LAGIS Laboratory/*HEI Lille/**University of Lille 1
Polytech Lille, bat D, 59655, Villeneuve D'Ascq, France

maxime.feingesicht@hei.fr / sc.olteanu@ed.univ-lille1.fr / abdel.aitouche@hei.fr / lotfi.belkoura@univ-lille1.fr

ABSTRACT

The article presents a method for validating a physical Embedded state observer by using a Hardware In The Loop architecture. As an application, a two water tank system was chosen. For the validation of the observer, a software has been employed, specifically LMS AMESim, interfaced by Simulink on a Windows operating system. Even though Windows is not well suited for real time operations, it is the most common operating system and it has proven to be robust enough to sustain our tests. The article also shows the capability of small scaled microcontroller based systems to handle complex algorithms. It also promotes the simplicity of implementing Takagi-Sugeno algorithms in small scaled numerical systems. Although tests were made on a Arduino UNO version, by adding the benefits of the powerful, in development, ArduinoTRE Board, the power bonus of Texas Instruments processor allows more complexity as well as signal filtering and faster and more secure communication protocols.

1. INTRODUCTION

It is often advised to test an embedded system before connecting it to the real process, this way avoiding a potential instability or lack of robustness towards uncertainties of the model or communication. Therefore a high precision simulation by use of a special software becomes an essential step in any development process. For this, AMESim comes to play [2].

As an embedded system, an Arduino platform is employed, because of its simplicity, size and price ratio. In order to compensate for the complexity of a full scale implementation of a standalone system, the Arduino TRE board [1] is envisaged. It fully responds to the demand; as the Texas Instruments Sitara processor completes the Atmel microcontroller.

Another objective of the current work being to show the ease in implementation of the Takagi-Sugeno(TS) approach to deal with nonlinear systems [5]. This methodology has grown in interest in the last decade because of its structured character. The T-S model is easily obtained considering a sector nonlinearity technique [4]. Once constructed, the model presents itself as a blending of linear systems, that by means of their interrelations give a perfect estimation of the initial system.

A nonlinear analysis of a two tank system has been chosen

as it represents a classical example, that also adds some interesting complexity when analyzed from a nonlinear perspective. This process has been approached by means of a Takagi-Sugeno representation in [3] where premise variables are unmeasurable. In the current work we have chosen a more academic approach, considering the premise variables measurable, as the restrictive suppositions made in [3] would still limit the domains of stability.

In the following, a general modeling of the implemented system is presented, followed by the Takagi-Sugeno transformation and the observer construction. Finally, the embedded system implementation is described, ending with conclusions and perspectives.

2. THE EMBEDDED SYSTEM

The hardware in the loop architecture permits the integration of a specialized software within a closed loop with a physical device in order to test the code and performance of the developed embedded system.

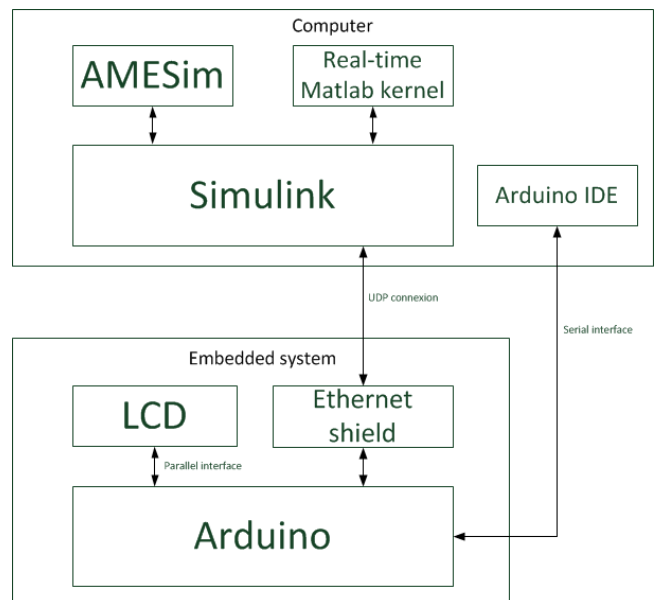


Figure 1 - Functional block of the real time system

As such the communication between the computer that hosts the software and the physical device has been chosen

as general as possible so that it's easily adapted to any potential microcontroller. Therefore an ethernet UDP connection has been established. Also a serial connection was tested and proven to be working as required, with a minimal sampling time of 5ms.

The interactions schematic behind the implementation of the embedded validation structure is shown as functional blocks in Figure 1. Although all the processes including the Matlab real time kernel, are managed by Windows, it is capable to give and maintain higher priority only to inputs and essential interrupts.

As AMESim needs Simulink for operating a real time connection, the creation of a co-simulation Amesim/Simulink is performed. Finally the Real Time Windows Target toolbox in Simulink will manage the compilation of the program that will run in the Real Time Windows Kernel of Matlab.

The Arduino code has been written directly in the Arduino's associated editor/compiler. The fact that the processor is single core does not restrict the system from obtaining a fast enough sampling time.

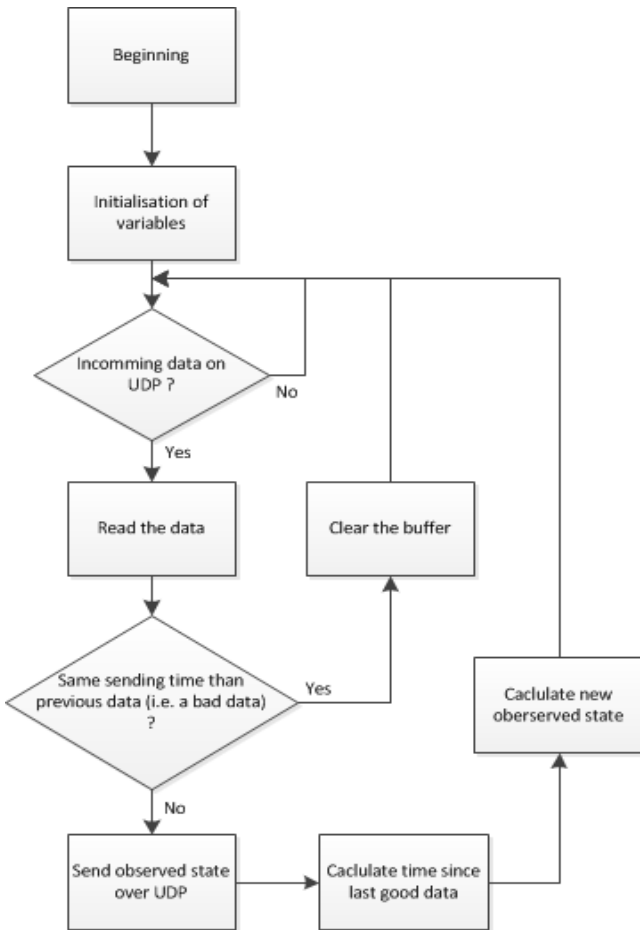


Figure 2 - Logical diagram of the Arduino code

When the observer has been programmed, a variable

discretization was employed, therefore lost samples do not affect the overall performance.

The logical diagram behind the Arduino code is presented in Figure2.

Now, in order to show the functioning principles, an application example has been chosen: System of two interconnected water tanks. The system is analyzed from a nonlinear perspective to show the potential of using T-S techniques on an embedded platform.

3. THE SYSTEM MODEL

For the two tank system(Figure 3), the height of the first tank has been considered measurable while the second tank's height is the estimated parameter.

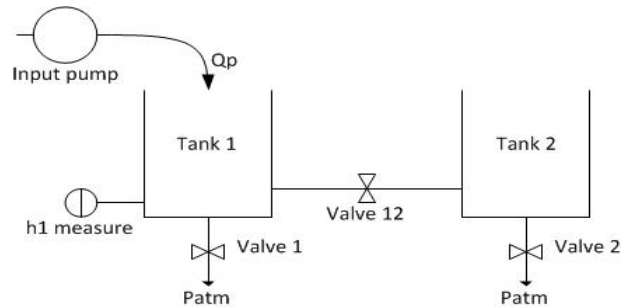


Figure 3 - Two tanks system representation

In this system, we have an input pump supplying the first tank. This tank presents a valve evacuating water outwards, where atmospheric pressure P_{atm} is encountered and another valve supplying the second tank. The second tank has only one valve, which goes towards the exterior. We will note Q_p the input flow from the pump, h_i heights of liquid in each tank and C_p the flow coefficient of each valve, all valves being considered identical. By neglecting the atmospheric pressure, the following equations are found:

$$\begin{cases} \dot{x} = -\frac{C_p \sqrt{\rho g}}{S} \begin{bmatrix} \sqrt{x_1} + \text{sgn}(x_1 - x_2) \sqrt{|x_1 - x_2|} \\ \sqrt{x_2} - \text{sgn}(x_1 - x_2) \sqrt{|x_1 - x_2|} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \frac{1}{S} u \\ y = x_1 = [1 \ 0] x \end{cases} \quad (1)$$

Where the input is the pump flow, and the states are the two heights and finally the output is the first tank's height:

$$\begin{cases} u = Q_p \\ x = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \\ y = h_1 = x_1 = h(x) \end{cases} \quad (2)$$

As it can be seen, the nonlinearities are present as square roots, that present no differentiability in 0.

4. OBSERVER DESIGN

4.1. Takagi-Sugeno Representation

The TS model is of the form (3), considering that B and C don't depend on the states and considering i the number of the fuzzy rule:

$$\begin{cases} \dot{x} = \sum_{i=1}^8 w_i(z) A_i x + B u \\ y = C x \\ z = \text{nonlinear terms (premise variables)} \end{cases} \quad (3)$$

As it can be seen, this new form represents a sum of linear coupled systems for which the nonlinearities have moved into the membership functions w_i , where:

$$\sum_i w_i = 1 \quad (4)$$

In order to advance towards the TS form, a transformation is applied to the system (1) to obtain the affine in control system (5).

$$\begin{cases} \dot{x} = -\frac{C_p \sqrt{\rho g}}{S} \begin{bmatrix} z_1 & z_3 \\ z_2 & -z_3 \end{bmatrix} x + \begin{bmatrix} 1 \\ S \\ 0 \end{bmatrix} u \\ y = x_1 = [1 \ 0] x \end{cases} \quad (5)$$

Where:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{x_1}}{x_1} \\ \frac{\sqrt{x_2}}{x_1} \\ \frac{\text{sgn}(x_1 - x_2) \sqrt{|x_1 - x_2|}}{x_2} \end{bmatrix} \quad (6)$$

The variables z are called premises variables, and they contain the nonlinear information for the system. In the following steps of the procedure, this new representation (5) is going to be used. By means of the sector nonlinearity method, the minimum and maximum of each z_i can be determined. As such, we establish the following conditions for the heights, where some additional constraints are added so that the singular values are eliminated:

$$\begin{cases} 0,05 \leq h_1 \leq 1m \\ 0,05 \leq h_2 \leq 1m \end{cases} \Rightarrow \begin{cases} 0,8 \leq z_1 \leq 1,1 \\ 0,2 \leq z_2 \leq 0,8 \\ -1,1 \leq z_3 \leq 1,2 \end{cases} \quad (7)$$

4.2. Takagi-Sugeno Observer

In this part, the focus is directed towards the observer construction, that will end with the construction of a set of conditions for their synthesis in terms of linear matrix inequalities (LMIs).

It is considered that the premise variables z are measurable, indeed an acceptable hypothesis, as the focus of the paper falls more on the embedded implementation and not on the T-S theory itself. Also the real time observer form is not affected very much by this assumption, from the point of view of the real time implementation. This means that, as in most of the cases in the literature [6], $w_i(\hat{z}) = w_i(z)$. Then, the Luenberger like observer is of the form :

$$\begin{cases} \dot{\hat{x}} = \sum_i w_i(z) (A_i \hat{x} + B_i u + L_i (\hat{y} - y)) \\ \hat{y} = \sum_i w_i(z) C_i \hat{x} \end{cases} \quad (8)$$

To obtain the observer gains, a simple numerical solution that can be resolved in Matlab, either using the incorporated optimization LMI solver, or YALMIP toolbox. This solution consists of the following LMIs:

$$\begin{cases} (\forall) i = 1:8, \left((P A_i - Q_i C) + (P A_i - Q_i C)^T \right) + 2\alpha P < 0 \\ \text{size}(P) = \text{size}(A) \\ P > 0 \\ P = P^T \\ \text{size}(Q_i) = \text{size}(A_i, C) \\ \alpha > 0 ; \text{scalar} \end{cases} \quad (9)$$

Proof:

In order to construct the stability it is required to start by demanding stability of the estimation error dynamics. Therefore the derivative of the estimation error is of the form (10) where the error is $e = x - \hat{x}$.

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{\hat{x}} = \sum_i w_i(z) (A_i x + B_i u) - \sum_i w_i(z) (A_i \hat{x} + B_i u + L_i (\hat{y} - y)) \\ \dot{e} &= \sum_i w_i(z) (A_i (x - \hat{x}) - L_i (\hat{y} - y)) = \sum_i w_i(z) (A_i - L_i C) e \end{aligned} \quad (10)$$

For the stability, based on the Lyapunov theory, the following quadratic Lyapunov function is chosen:

$$V = e^T P e ; P = P^T > 0 \quad (11)$$

Its derivative is :

$$\dot{V} = \dot{e}^T P e + e^T P \dot{e} \quad (12)$$

$$\dot{V} = \left(\sum_i w_i(z) (A_i - L_i C) e \right)^T P e + e^T P \left(\sum_i w_i(z) (A_i - L_i C) e \right) \quad (13)$$

$$\dot{V} = \sum_i w_i(z) e^T (A_i - L_i C)^T P e + \sum_i w_i(z) e^T P (A_i - L_i C) e$$

Finally :

$$\dot{V} = e^T \left(\sum_i w_i(z) \left((A_i - L_i C)^T P + P (A_i - L_i C) \right) \right) e \quad (14)$$

In order for the estimation error to decrease to 0, it is needed to have :

$$\dot{V} < 0 \quad (15)$$

Therefore the following inequality is obtained:

$$\left(\sum_i w_i(z) \left((A_i - L_i C)^T P + P (A_i - L_i C) \right) \right) < 0 \quad (16)$$

One easy way to be sure that these conditions are satisfied is to resolve the inequality (17), which even though restrictive, it is sufficient, leaving out a small interval of potential solutions:

$$(\forall) i = 1:8, \quad x_i < 0 \Rightarrow \sum_i x_i < 0 \quad (17)$$

Which applied to (16) gives:

$$(\forall) i = 1:8, \quad \left((A_i - L_i C)^T P + P (A_i - L_i C) \right) < 0 \quad (18)$$

One can easily deduce the set of LMIs, where new variables are created namely $Q_i = PL_i$. Then, using the symmetry of P :

$$(\forall) i = 1:8, \quad \left((PA_i - Q_i C) + (PA_i - Q_i C)^T \right) < 0 \quad (19)$$

Moreover, if it is wanted for the dynamic of the error of our observer to be exponentially decreasing, one can use the following equation :

$$\dot{V} + 2\alpha V < 0 \quad (20)$$

where α is the exponential decay rate. This modifies equation (20) as to obtain exactly the initial inequality (9), thus ending the proof.

In the case where it is aimed to optimize the value of α , a Bilinear Matrix Inequality (BMI) solver is needed. Here, we will only use LMIs, therefore we suppose that α is known and the LMIs in consequence, can be solved.

5. OBSERVER IMPLEMENTATION

5.1. The AMESim system

Once the model has been constructed in AMESim, by adding all the blocks, creating the connections and configuring the parameters, one has to add the co-simulation block that will create the dynamic link with Simulink. For this, AMESim R12 and Matlab 2012a have been used.

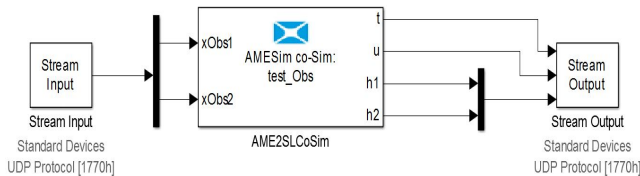


Figure 4 - Simulink diagram including cosimulation block

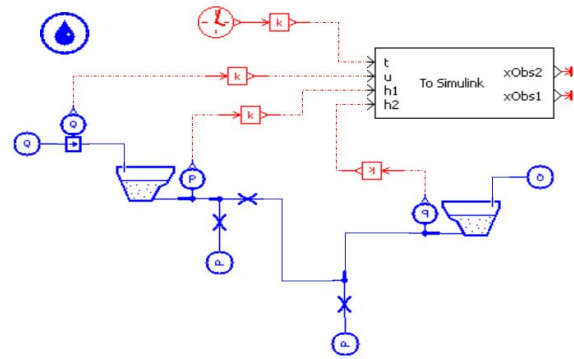


Figure 5 - Amesim implementation of the two tank system

On the other hand, Simulink has to have the corresponding co-simulation block as well, as in Figure 4.

5.2. The real time simulation

By means of the experimental setup shown in the Figure 6, a performance analysis is done.

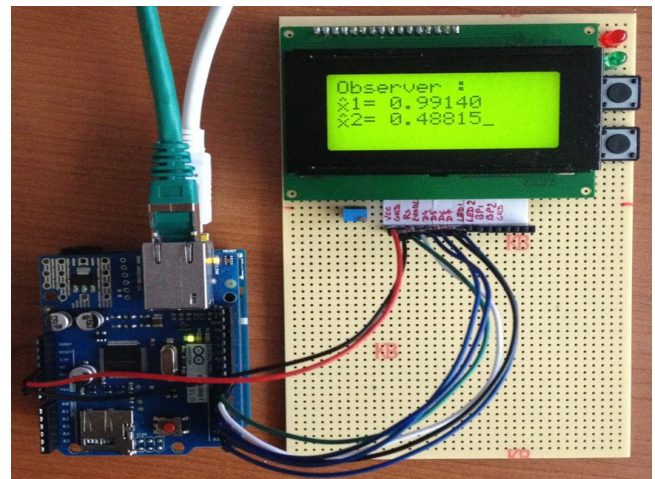


Figure 6 - Hardware implementation

Using the obtained LMIs, one gets the gains (23).

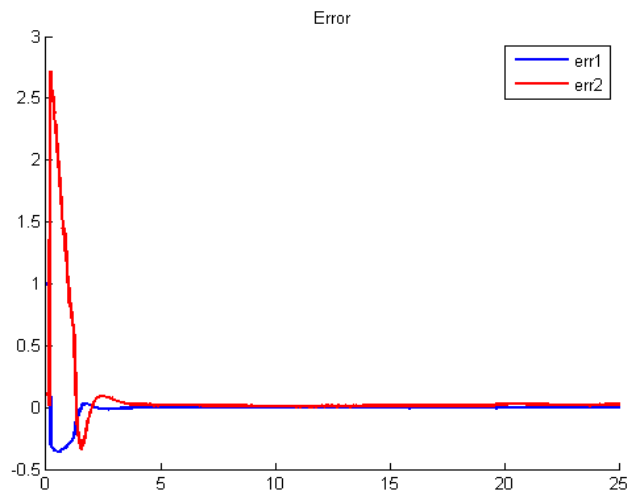


Figure 7 - Estimation error dynamics

As it can be observed in Figures 7 and 8, the estimation of the water level in the second tank stabilizes in 3 seconds, which is an adequate time response. Also the overshoot of the estimation is in an acceptable interval.

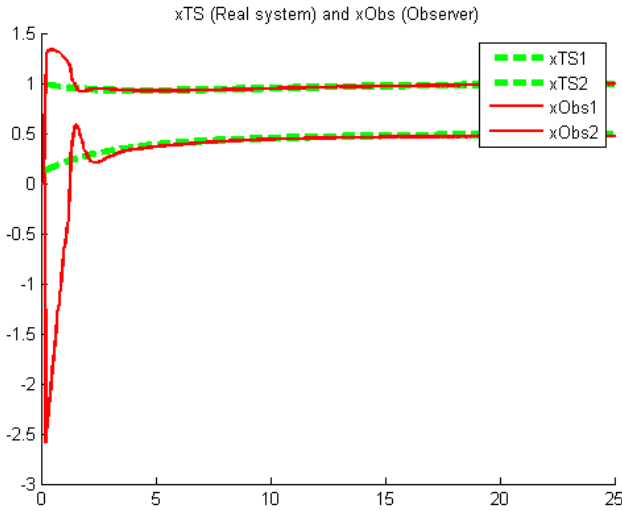


Figure 8 - Real/Estimated water levels time evolution

In the previous figures, the perturbation added to the communication in a forcedly manner, does not affect the observer's behavior.

6. CONCLUSIONS AND PERSPECTIVES

The observer's performance, despite the nonlinear character of the model, confirm the capability of both the embedded system as well as the Takagi-Sugeno algorithm. Also, the validation method can be applied to many processes, ranging from mechanical to chemical or electrical. The hardware in the loop, real time testing permits a robust validation of the observer/controller or diagnostic system, as it adds unpredictable communication and time response errors.

As perspectives, the authors envision to make use of the power of the Arduino TRE to ensure also a robust data acquisition and also better secured communication, as well as to eliminate the restrictiveness of the measurability condition of the premise variables.

7. APPENDIX

$$\begin{aligned}
 C_p &= 1.10^{-5} \text{ m}^3 \cdot \text{s}^{-1} \cdot \text{Pa}^{-1/2} \\
 \rho &= 1000 \text{ kg} \cdot \text{m}^{-3} \\
 g &= 9.81 \text{ kg} \cdot \text{m} \cdot \text{s}^{-2} \\
 S &= 0.005 \text{ m}^2 \\
 Q_p &= 0.001690811493
 \end{aligned}
 \tag{21}$$

$$\alpha = 0.1$$

$$z_{1_{\min}} = 0.8; z_{1_{\max}} = 1.1 \tag{22}$$

$$z_{2_{\min}} = 0.2; z_{2_{\max}} = 0.8$$

$$z_{3_{\min}} = 1.1; z_{3_{\max}} = 12$$

$$L_1 = \begin{bmatrix} 1.4887 \\ -6.2326 \end{bmatrix}; L_2 = \begin{bmatrix} 13.4161 \\ -60.8919 \end{bmatrix}$$

$$L_3 = \begin{bmatrix} 1.4887 \\ -6.3514 \end{bmatrix}; L_4 = \begin{bmatrix} 13.4161 \\ -61.0108 \end{bmatrix}$$

$$L_5 = \begin{bmatrix} 1.4293 \\ -6.2326 \end{bmatrix}; L_6 = \begin{bmatrix} 13.3567 \\ -60.8919 \end{bmatrix} \tag{23}$$

$$L_7 = \begin{bmatrix} 1.4293 \\ -6.3514 \end{bmatrix}; L_8 = \begin{bmatrix} 13.3567 \\ -61.0108 \end{bmatrix}$$

$$P = \begin{bmatrix} 804.2900 & 175.5070 \\ 175.5070 & 63.1366 \end{bmatrix} \tag{24}$$

8. ACKNOWLEDGEMENTS

This work was carried out in the framework of PN-7-022-BE i-MOCCA (Interregional Mobility and Competence Centres in Automation), an European territorial cooperation project part-funded by the European Regional Development Fund (ERDF) through the INTERREG IV A 2 Seas Programme and the Ministry of Education and Research, France.

REFERENCES

- [1] Arduino webpage, "ArduinoTRE documentation" in <http://arduino.cc/en/Main/ArduinoBoardTre>.
- [2] Bourdon, T., Saussol, L., and Varoquié, B., "Integration of Physical AMESim® Engine Model in Hardware in the Loop Environment, Dedicated to Engine Control Unit Testing," in *SAE Technical Paper*, January 2007, doi:10.4271/2007-01-1300.
- [3] H. Ghorbel, A. El Hajjaji, M. Souissi, and M. Chaabane, "Robust Tracking Control for Takagi-Sugeno Fuzzy Systems With Unmeasurable Premise Variables: Application to Tank System," in *Journal of Dynamic Systems, Measurement and Control*, vol. 136, July. 2014.
- [4] E. J. Herrera-Lopez, B. Castillo-Toledo, J. Ramirez-Cordova, and E. C. Ferreira, "Takagi-Sugeno Fuzzy Observer for a Switching Bioprocess: Sector Nonlinearity Approach," in *New Developments in Robotics Automation and Control*, InTech, chap. 8, pp. 155, 2008.
- [5] K. Tanaka, and H. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, John Wiley and Sons, 2001.
- [6] Z. Lendek, T. M. Guerra, R. Babuska, and B. de Schutter, "Stability Analysis and Nonlinear Observer Design Using Takagi-Sugeno Fuzzy Models", Springer, 2010.