



Advanced embedded nonlinear observer design and HIL validation using a Takagi-Sugeno approach with unmeasurable premise variables

Severus Constantin Olteanu, A Aitouche, L Belkoura

► To cite this version:

Severus Constantin Olteanu, A Aitouche, L Belkoura. Advanced embedded nonlinear observer design and HIL validation using a Takagi-Sugeno approach with unmeasurable premise variables . European Workshop on Advanced Control and Diagnosis, Nov 2014, Berlin, Germany. hal-01248954

HAL Id: hal-01248954

<https://inria.hal.science/hal-01248954>

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.



Distributed under a Creative Commons Attribution 4.0 International License

Advanced embedded nonlinear observer design and HIL validation using a Takagi-Sugeno approach with unmeasurable premise variables

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2014 J. Phys.: Conf. Ser. 570 022003

(<http://iopscience.iop.org/1742-6596/570/2/022003>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 85.171.24.183

This content was downloaded on 28/12/2015 at 15:28

Please note that [terms and conditions apply](#).

Advanced embedded nonlinear observer design and HIL validation using a Takagi-Sugeno approach with unmeasurable premise variables

S C Olteanu¹, A Aitouche², L Belkoura³

LAGIS UMR CNRS 8219

¹ University of Lille 1, France; E-mail: sc.olteanu@ed.univ-lille1.fr

² HEI Lille, France; E-mail: abdel.aitouche@hei.fr

³ University of Lille 1, France; E-mail: lotfi.belkoura@univ-lille1.fr

Abstract. The article's goals are to illustrate the feasibility of implementing a Takagi Sugeno state observer on an embedded microcontroller based platform and secondly to present a methodology for validating a physical embedded system using a Hardware In The Loop architecture, where a simulation software replaces the process. As an application, a three water tank system was chosen. For the validation part, LMS AMESim software is employed to reproduce the process behaviour. The interface to the embedded platform is assured by Simulink on a Windows operating system, chosen as it is the most commonly used operating system. The lack of real time behaviour of the operating system is compensated by a real time kernel that manages to offer deterministic response times. The Takagi-Sugeno observer in the case of this process has the complex form that considers the premise variables to be unmeasurable. The embedded system consists of two Arduino boards connected in parallel, thus offering distributed resources.

1. Introduction

A consistent increase has been seen in the last decade in the usage of small scaled embedded systems in control, diagnostics and signal manipulation, as these solution become cheaper and more powerful [1]. Also, as the requirements of more complex algorithms augment, the study of the possibilities and the feasibility of implementing complex algorithms on these types of platforms becomes more and more useful. As such, we have implemented a nonlinear state observer using a Takagi-Sugeno (T-S) [2].

Because it is preferred to test an embedded system prior to its implementation on a real world process in order to avoid unforeseen errors in modeling or communication, it is a useful practice to test the system in a hardware in the loop configuration, where the process itself is replaced by a simulation software, as the AMESim software [3], [4]. For the embedded part, based on versatility and size/price ratio, a microcontroller platform (on an Arduino board) is employed.

The T-S methodology, that can be seen as a polytopic linear differential inclusion, has seen a growing interest in the last decade as it has a simple implementation mechanism, as well as a simple structure. A common solution for obtaining a TS model from a general nonlinear one, is to apply a



sector nonlinearity transformation [5] to the state space form, thus giving an exact approximation. This model consists of a blend of linear systems that are coupled together by means of weighting functions.

As an application, a nonlinear analysis of a three tank system has been selected as it represents a basic example that brings some interesting complexity when looked upon from a nonlinear perspective. After the construction of the state space model, the T-S transformation is applied, and for the newly built representation, the observer is developed. A similar approach, also with unmeasurable premise variables can be seen in [6]. Others have used the application in nonlinear studies, even fault diagnosis like in [7].

The article begins in section 2 with a general description of the T-S representation and the construction of the observer, followed in section 3 by the embedded system description as well as the HIL validation technique. Section 4 deals with the application process, meaning the three tank system's modelling and its TS representation. Section 5 presents the implementation of the observer for the system both in simulation and on the physical process. The paper ends with a conclusions section.

2. Takagi-Sugeno Observer

2.1. Takagi-Sugeno Representation

A general TS model presents itself as a sum of linear systems as in (1), where each system contributes in a percentage given by the weighting function w_i . Here we consider B and C as non dependent on states and we consider i as the number of a fuzzy rule (this representation may also be seen as a fuzzy system).

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

It becomes obvious that this way of writing the system moves the nonlinearities from inside the system matrices into the weighting functions, adding the fact that they respect a convex sum property:

$$\sum_i w_i = 1 \quad (2)$$

To arrive from the general nonlinear form to a TS representation the most common procedure is to apply sector nonlinearity technique. But first of all, the system is rewritten in form (3):

$$\begin{cases} \dot{x} = A(z(x)) \cdot x + B \cdot u; \\ y = C \cdot x \end{cases} \quad (3)$$

The z variables, that are dependent on states, have the name of premise variables, and they contain the nonlinear information for the system. Now the sector nonlinearity method [8] can be applied by determining the minimum and maximum of each z_i . If we consider a general triangular membership function one arrives at (4) where we design MFs as membership functions:

$$\begin{cases} i = 1..n \\ MF_{min}(z_i) = \frac{z_{i,Max} - z_i}{z_{i,Max} - z_{i,min}} \\ MF_{Max}(z_i) = \frac{z_i - z_{i,min}}{z_{i,Max} - z_{i,min}} \\ w_i = MF_{Max/Min}(z_1) \cdot \dots \cdot MF_{Max/Min}(z_n) \end{cases} \quad (4)$$

2.2. Takagi-Sugeno Observer

Using the T-S representation, the observer itself can be constructed. The observer design will materialize as an optimization problem, that will be resolved by means of linear matrix inequalities (LMIs). The observer chosen here has a classical Luenberger form adapted to TS systems as in (5).

$$\begin{cases} \dot{\hat{x}} = \sum_i w_i(\hat{z})(A_i \hat{x} + B_i u + L_i(\hat{y} - y)) \\ \hat{y} = \sum_i w_i(\hat{z})C_i \hat{x} \end{cases} \quad (5)$$

As it can be seen, the weighting functions depend on estimated premise variables \hat{z} , therefore the computations are more complex. In the literature, there are many cases where premise variables are directly measurable as in [5], a supposition that indeed simplifies the stability study ($w_i(\hat{z})=w_i(z)$). This is not the case for the current work.

In order to construct the observer (6), we need to calculate the observer gains. This will be done by employing Matlab's implicit LMI solver or YALMIP toolbox to resolve the LMIs presented in (6).

$$\begin{cases} \text{For } i = 1 : 8 \\ \begin{pmatrix} A_i^T P - C^T Q_i + P A_i - Q_i C + I + 2\alpha P & P \\ P & -\lambda \end{pmatrix} < 0 \\ \text{size}(P) = \text{size}(A) \\ P > 0 \\ P = P^T \\ Q_i = P \cdot L_i \\ \text{size}(Q_i) = \text{size}(A_i \cdot C) \\ \alpha > 0 ; \text{ scalar} \end{cases} \quad (6)$$

Proof:

As the goal of any observer is to reach stability in 0 of the estimation error $e = x - \hat{x}$, the methodology adopted is Lyapunov stability analysis. This dictates to impose condition (8) upon a well chosen Lyapunov function. We have chosen a quadratic one as in (7).

$$V = e^T P e ; P = P^T > 0 \quad (7)$$

Its derivative being:

$$\dot{V} = \dot{e}^T P e + e^T P \dot{e} < 0 \quad (8)$$

By developing the expression of the estimation error's derivative as in (9), one can rearrange it to get expression (10).

$$\dot{e} = \sum_i w_i(\hat{z})(A_i x + B_i u) + \sum_i (w_i(z) - w_i(\hat{z}))(A_i x + B_i u) - \sum_i w_i(\hat{z})(A_i \hat{x} + B_i u + L_i(\hat{y} - y)); \quad (9)$$

$$\begin{cases} \dot{e} = \sum_i w_i(\hat{z})(A_i \cdot e + B_i \cdot u - L_i \cdot C \cdot e) + \Delta; \\ \Delta = \sum_i^{not} (w_i(z) - w_i(\hat{z}))(A_i \cdot x + B_i \cdot u) \end{cases} \quad (10)$$

We have separated the Δ term as it contains a difference between measured weighting function and estimated one, this expression posing a problem in the resolution of the stability requirement. A solution seen generally is to use Lipchitz constants, yet this solution is quite restrictive for systems where the constant is big. Therefore we adopted another method, which is to consider Δ as a perturbation, a value that converges to 0 when $t \rightarrow \infty$. Therefore the stability problem passes from an asymptotic one to a bounded stability, where the bound is as small as possible. Similar approach has been treated in [9].

From equation (8), making use of (10), one obtains:

$$\dot{V} = e^T \left(\sum_i w_i(\hat{z})(A_i - L_i C)^T P + \sum_i w_i(\hat{z})P(A_i - L_i C) \right) e + \Delta^T P e + e^T P \Delta \quad (11)$$

Moreover, imposing stability with rejection of the virtual perturbation Δ , (12) is reached.

$$\dot{V} + e^T R e - \xi^2 \Delta^T \Delta < 0 \quad (12)$$

Where $R > 0$, usually taken as the identity matrix and ξ is an small as possible positive coefficient. This leading to:

$$e^T \left(\sum_i w_i(\hat{z}) \left[(A_i - L_i C)^T P + P(A_i - L_i C) \right] \right) e + \Delta^T P e + e^T P \Delta + e^T R e - \xi^2 \Delta^T \Delta < 0; \quad (13)$$

These inequalities still contain variable terms multiplied with other variable terms, so it still has nonlinearities in the last part of the sum. In order to eliminate this, the following known inequality is employed, which is available for all positive λ :

$$\forall \lambda > 0, X, Y: X^T Y + Y^T X \leq \lambda X^T X + \lambda^{-1} Y^T Y \quad (14)$$

Which in our context means:

$$\Delta^T P e + e^T P \Delta + e^T R e - \xi^2 \Delta^T \Delta \leq \lambda \Delta^T \Delta + \lambda^{-1} e^T P P e + e^T R e - \xi^2 \Delta^T \Delta; \quad (15)$$

Applying this to (13) will get us to:

$$e^T \left(\sum_i w_i(\hat{z}) \left[(A_i - L_i C)^T P + P(A_i - L_i C) \right] + I \right) e + \lambda \Delta^T \Delta + \lambda^{-1} e^T P P e - \xi^2 \Delta^T \Delta < 0; \quad (16)$$

As $\lambda > 0$ can be any scalar, then one can choose $\lambda = \xi^2$, therefore considering also the convex sum property, the inequality becomes:

$$\sum_i w_i(\hat{z}) \left[(A_i - L_i C)^T P + P(A_i - L_i C) + I + \lambda^{-1} P P \right] < 0; \quad (17)$$

To finalize and to obtain a set of LMI, we use a restrictive hypothesis that each term in the sum is negative, and as w_i are all positive, then (18) is reached.

$$\begin{cases} \text{For } i = 1:8 \\ (A_i - L_i C)^T P + P(A_i - L_i C) + I + \lambda^{-1} P P < 0; \end{cases} \quad (18)$$

Using the notation $Q_i = P L_i$ and making use of the Schur transformation, we thus avoid a Bilinear Matrix problem (BMI).

Also, optionally we can improve the performance by imposing exponential stability, by replacing (12) with (19), where α is the exponential decay rate.

$$\dot{V} + e^T R e - \xi^2 \Delta^T \Delta + 2\alpha V < 0 \quad (19)$$

Of course this would also generate a BMI problem, but this can be easily resolved by choosing manually by trial, a suitable α . So from (18). Adding the exponential stability, we have proven the LMIs (6).

3. The Embedded system and HIL validation platform

The embedded platform is based on two Arduino Mega 2560, as the number of input/outputs of the ATMEGA 2560 is bigger than the standard UNO version. The two microcontrollers are connected in between themselves as in figure 1. The connection can be done both by serial transfer or parallel one (by digital pins). In the application chosen in the current article both configurations give acceptable results, yet it has to be taken into account that floating point data is transferred (stored on 16 bits), so the serial communication has a slower transfer time.

The implementation of the observer on the microcontroller and possible unpredicted errors can be tested by a hardware in the loop (HIL) architecture, where the microcontrollers based platform is connected to a computer that hosts the simulation software (AMESim). The communication protocol between the computer hosting the software and the embedded system has been chosen to be a general one, adaptable for any situation, therefore an Ethernet UDP connection fits the requirement. For this, a 5 ms sampling time can be reached at the 16Mhz of the microcontroller. An important inconvenient of

microcontroller systems is that the floating point operations are done at software level, increasing therefore the processing power.



Figure 1. Hardware implementation

The basic functional schematic at the base of the implementation of the embedded HIL validation structure is shown in Figure 2. Although the operating system is Windows, which by default does not assure real time operation, by using the Matlab Real Time Kernel, only the processing threads of interest to the task at hand are kept at a top priority level being overpassed just by core root threads like system inputs and operating system messaging interrupts.

As AMESim needs Simulink for operating a real time data exchange, the creation of a co-simulation AMESim/Simulink is required. As the co-simulation is built, the Real Time Windows Target toolbox in Simulink manages the compilation of the program that will be executed in the Real Time Windows Kernel of Matlab.

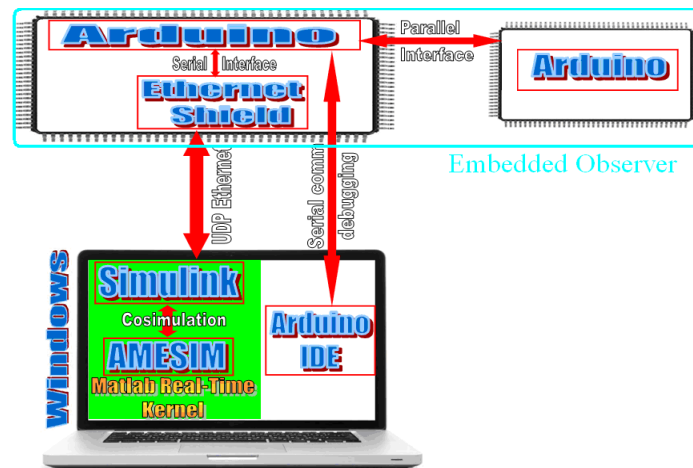


Figure 2. Functional block of the real time system

The coding done in the Arduino IDE is embedded C language, more robust for larger problems than assembly language. A variable discretization is used for the observer, meaning it is recalculated each sampling time inside each microcontroller, this helping in talking the problem of lost samples

The time lost because of the computations done in floating point (which is done at software level) and also due to the fact that TS model is represented as a sum, gave us the possibility to implement a distributed solution, where 2 processors divides the workload at each sampling time. The logical diagram is presented in Figure 3.

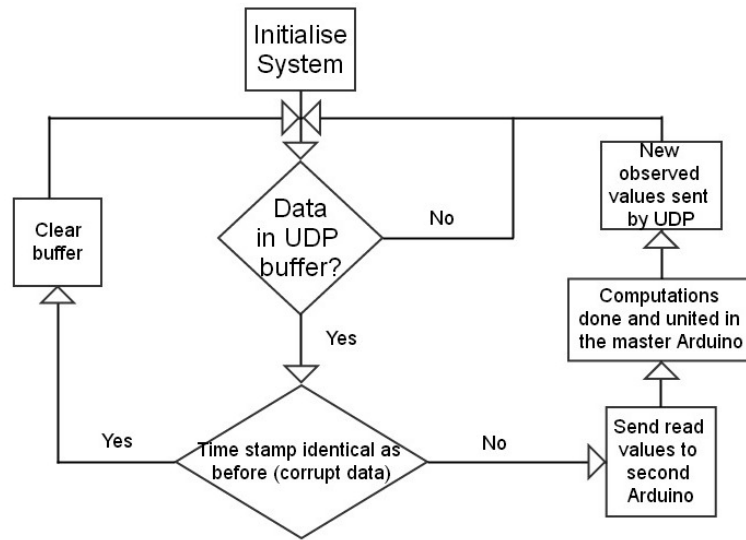


Figure 3. Logical diagram of the Arduino code

4. The application

4.1. System description and modeling

In order to show the functioning principles, an application example has been chosen, specifically a system of three interconnected water tanks. The system is analyzed from a nonlinear perspective to show the potential of using T-S techniques on an embedded platform. As it can be seen in Figure 4, the height of the first tank and the third tank has been considered measurable while the second tank's height is the estimated parameter.

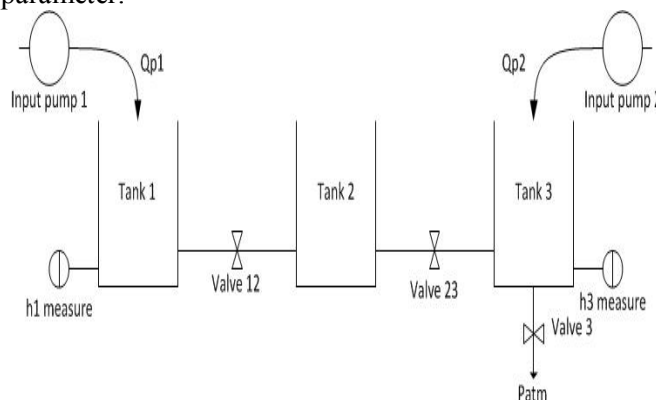


Figure 4. Three tanks system representation

In this system, we have an input pump supplying the first tank Q_{p1} as well as another pump for the third tank Q_{p2} . The last tank presents a valve evacuating water outwards, where atmospheric pressure P_{atm} is encountered. Also, a valve between it and the middle tank is present. The second tank has only the two interconnection valves while the first tank presents only one valve supplying the second tank. A supposition is made, that the flow of the third tank pump is smaller than the first one, thus the levels in tanks will always remain in a descending step, the first the highest level, then the second and the smallest water level in the third at all times.

$$\left\{ \begin{array}{l} \frac{d(V_1)}{dt} = Q_{P_1} - Q_{12} \\ \frac{d(V_2)}{dt} = Q_{12} - Q_{23} \\ \frac{d(V_3)}{dt} = Q_{P_2} + Q_{23} - Q_3 \\ V_1 = S_1 h_1; V_2 = S_2 h_2; V_3 = S_3 h_3 \\ Q_3 = C_{P_3} \sqrt{P_3} \\ Q_{12} = C_{P_{12}} \operatorname{sgn}(P_1 - P_2) \sqrt{|P_1 - P_2|} \\ Q_{23} = C_{P_{23}} \operatorname{sgn}(P_2 - P_3) \sqrt{|P_2 - P_3|} \\ P_1 = \rho g h_1; P_2 = \rho g h_2; P_3 = \rho g h_3 \end{array} \right. \quad (20)$$

We will use the notations h_i as 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, equations (20) are found. Developing equation (20), the system equations become (21).

$$\left\{ \begin{array}{l} \frac{dh_1}{dt} = \frac{1}{S_1} (Q_{P_1} - C_{P_{12}} \operatorname{sgn}(h_1 - h_2) \sqrt{\rho g |h_1 - h_2|}) \\ \frac{dh_2}{dt} = \frac{1}{S_2} (C_{P_{12}} \operatorname{sgn}(h_1 - h_2) \sqrt{\rho g |h_1 - h_2|} - C_{P_{23}} \operatorname{sgn}(h_2 - h_3) \sqrt{\rho g |h_2 - h_3|}) \\ \frac{dh_3}{dt} = \frac{1}{S_3} (Q_{P_2} + C_{P_{23}} \operatorname{sgn}(h_2 - h_3) \sqrt{\rho g |h_2 - h_3|} - C_{P_3} \sqrt{\rho g h_3}) \end{array} \right. \quad (21)$$

Now, the equivalent state-space form can be built, considering $S_1 = S_2 = S_3$ and $C_{P_{12}} = C_{P_{23}} = C_{P_3}$. As it can be seen, the inputs are the pump flows, and the states are the three heights and finally the output is the first and last tanks' height:

$$\left\{ \begin{array}{l} u = \begin{bmatrix} Q_{P_1} \\ Q_{P_2} \end{bmatrix} \\ x = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \\ y = \begin{bmatrix} h_1 \\ h_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = h(x) \end{array} \right. \quad (22)$$

$$\left\{ \begin{array}{l} \dot{x} = -\frac{C_P \sqrt{\rho g}}{S} \begin{bmatrix} \operatorname{sgn}(h_1 - h_2) \sqrt{|h_1 - h_2|} \\ -\operatorname{sgn}(h_1 - h_2) \sqrt{|h_1 - h_2|} + \operatorname{sgn}(h_2 - h_3) \sqrt{|h_2 - h_3|} \\ -\operatorname{sgn}(h_2 - h_3) \sqrt{|h_2 - h_3|} + \sqrt{h_3} \end{bmatrix} + \begin{bmatrix} \frac{1}{S} & 0 \\ 0 & 0 \\ 0 & \frac{1}{S} \end{bmatrix} u \\ y = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} x \end{array} \right. \quad (23)$$

As it can be seen, the nonlinearities are present as square roots, that have no differentiability in 0, yet for the regime in which we work, we will not reach this point.

4.2. Takagi-Sugeno transformation

In order to obtain the state observer, it is first needed to modify the state space form, so that it can be used to write the Takagi-Sugeno (TS) representation. So equation (23) is written as (24):

$$\begin{cases} \dot{x} = -\frac{C_p \sqrt{\rho g}}{S} \begin{bmatrix} z_1 & -z_1 & 0 \\ -z_1 & z_1 + z_2 & -z_2 \\ 0 & -z_2 & z_2 + z_3 \end{bmatrix} x + \frac{1}{S} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u \\ y = x_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} x \end{cases} \quad (24)$$

Where:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \frac{\text{sgn}(x_1 - x_2)}{\sqrt{|x_1 - x_2|}} \\ \frac{\text{sgn}(x_2 - x_3)}{\sqrt{|x_2 - x_3|}} \\ \frac{1}{\sqrt{x_3}} \end{bmatrix} \quad (25)$$

For the sector nonlinearity transformation we have to establish the conditions for the heights, where some additional constraints are added so that singular values are eliminated:

$$\begin{cases} 0,05 \leq h_1 \leq 0,6m \\ 0,05 \leq h_2 \leq 0,6m \\ 0,05 \leq h_3 \leq 0,6m \end{cases} \quad (26)$$

The three valves have similar characteristics, and the flow on the left pump is bigger, therefore the only possibility when the water levels would be equal is when the pump starts or when the tanks are empty, in which case the observer is not useful. Therefore, it is impossible that the difference between levels of the 3 tanks will change their sign. Even if this had happened, one could resolve this problem, by separating the system into multiple systems that switch according to the value of the difference between heights. Once all the observers are built, the one that will be active is chosen in a switching manner. Of course the main regime is the case when all the square roots are different from 0.

The parameters for the system are:

$$\begin{aligned} C_p &= 1.57 \cdot 10^{-6} \text{ m}^3 \cdot \text{s}^{-1} \cdot \text{Pa}^{-1/2} \\ \rho &= 1000 \text{ kg} \cdot \text{m}^{-3} \\ g &= 9.81 \text{ kg} \cdot \text{m} \cdot \text{s}^{-2} \end{aligned} \quad (27)$$

$$\begin{aligned} S &= 0.0164 \text{ m}^2 \\ Q_p &= 10^{-4} \text{ m}^3 \cdot \text{s}^{-1} \\ z_{1_{\min}} &= 0,5 \ ; \ z_{1_{\max}} = 2,5 \ ; \ z_{2_{\min}} = 2 \ ; \\ z_{2_{\max}} &= 10 \ ; \ z_{3_{\min}} = 1,6 \ ; \ z_{3_{\max}} = 2,6 \end{aligned} \quad (28)$$

5. Observer Implementation

5.1. The AMESim System

For the validation part, one has to first construct the AMESim model, by adding all the elements, configuring their parameters, and then the co-simulation block with Simulink needs to be added in Amesim as well as in figure 6. For the Simulink part (Figure 5), Matlab 2012a have been used. The Amesim version is R12.

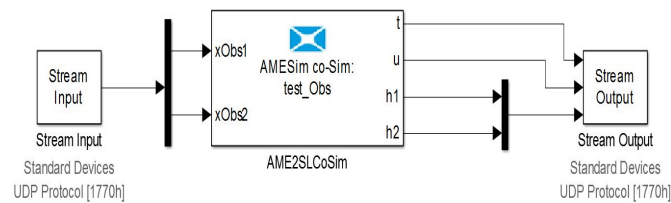


Figure 5. Simulink diagram including co-simulation block

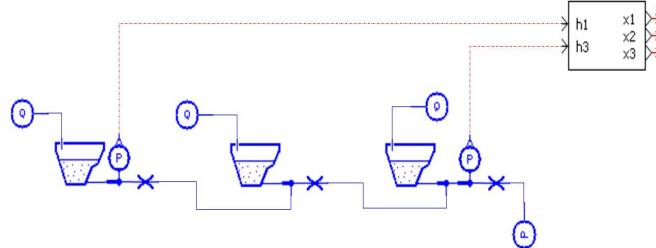


Figure 6. AMESim implementation of the three tank system

As the connection between Amesim and Simulink is done, by using Matlab real time workshop we add the input/output blocks that will connect via Ethernet to the microcontroller. The real time compilation is then done, and the whole system is run from Simulink.

5.2. The real time simulation

By means of the experimental setup described previously, a performance analysis is done in real time, and the parameters are configured for optimal results.

As it can be observed in figures 7 and 8, the estimation of the water level in the second tank stabilizes in about 10 seconds, which is an adequate time response. Also the overshoot of the estimation is in an acceptable interval.

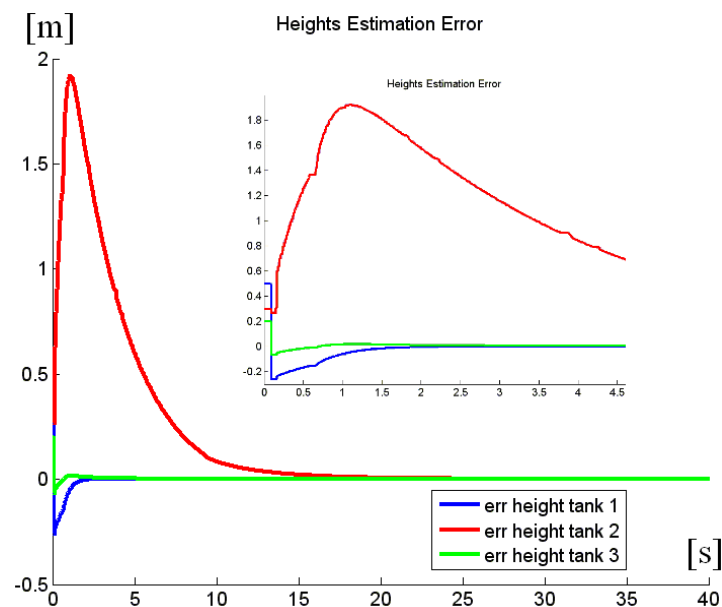


Figure 7. Estimation error dynamics of the 3 heights (tank 1, 2 and 3)

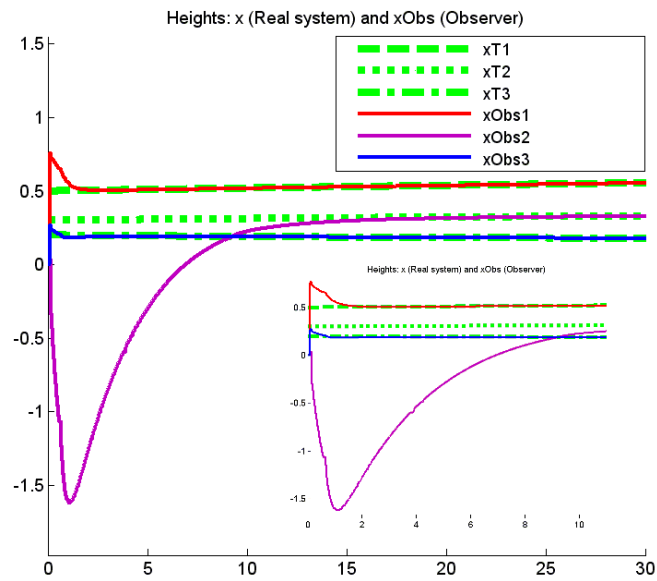


Figure 8. Real/Estimated water levels evolution

It is important to notice that perturbation added to the communication in a forcedly manner, can be seen to affect the system like in figure 7 at second 0.7 (in the zoomed pane), but performance remains good overall, thus proving its robustness to data loss.

5.3. Validation on a real system

Once the tested had been finished the embedded observer has been installed on a real 3 tank platform AMIRA DTS200 (Figure 9).



Figure 9. AMIRA DTS 200

After the implementation, also adding some filters on the inputs and outputs, the estimation error stabilizes in 8 seconds as can be seen in the following figure 10:

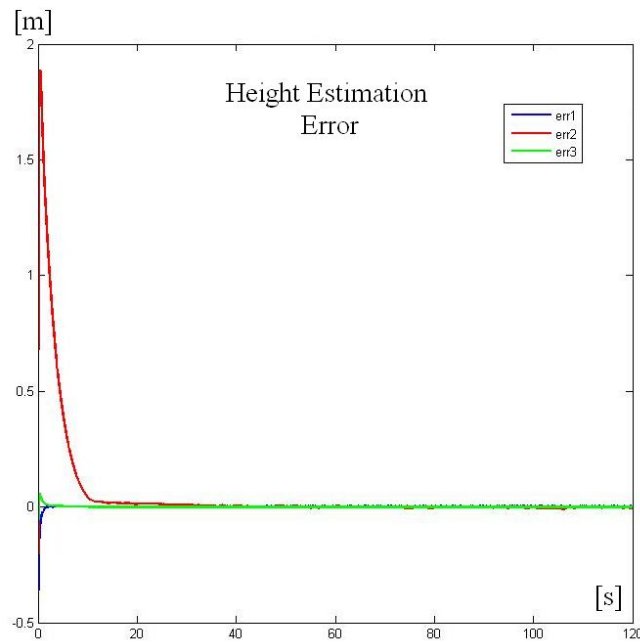


Figure 10. AMIRA DTS 200 estimation error dynamics in time

By introducing short perturbations, into the system, by means of purge valves openings, the observer still manages to regain stability (as in figure 11):

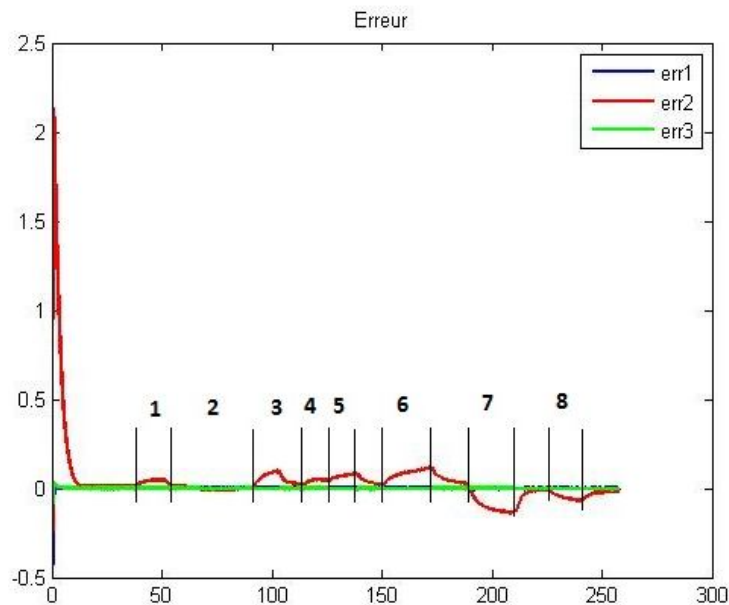


Figure 11. AMIRA DTS 200

Where the numbered areas are:

- 1) Leakage in tank 1 (fault 1)
- 2) Leakage in tank 2 (fault 2)
- 3) Leakage in tank 3 (fault 3)
- 4) Pump of tank 1 does not work (no input flow) (fault 4)
- 5) Pump of tank 3 does not work (no input flow) (fault 5)
- 6) Valve between tank 2 and tank 3 is blocked (fault 6)
- 7) Valve between tank 3 and atmosphere is blocked (fault 7)
- 8) Valve between tank 1 and tank 2 is blocked (fault 8).

6. Conclusions and Perspectives

An embedded nonlinear observer was developed using the Takagi Sugeno approach and applied to a three tank system with a prior real time HIL validation. As previously seen, the observer's performances on the HIL testing bench reproduce the behaviour of the real system thus reducing implementation time and also avoiding potential dangerous faults on the real system. The testing also takes into consideration communication losses and memory and computation timing errors. Of course the testing itself can be reproduced for any process that is supported by the simulation platform. The developed observer was applied to a real three tank system to confirm the results. Also another objective of the paper was to show the capability of a T-S system to be represented on small scaled microcontroller platforms. By doing the parallel calculus in between two microcontrollers, the complexity of the system can be augmented even further.

7. Acknowledgments

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.

8. References

- [1] Chen M, Edwards D, Boehmer E, Eller N, Slack J, Speck C, Brown S, Williams H, Wilson S, Gillum C, Lewin G, Sherriff M, Garner G, Designing a spatially aware and autonomous quadcopter, *Systems and Information Engineering Design Symposium (SIEDS)*, 2013 IEEE , pp.213,218, 26-26 April 2013
- [2] Tanaka K and Wang H, Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach, *John Wiley and Sons*, 2001.
- [3] 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, *SAE Technical Paper*, January 2007, doi:10.4271/2007-01-1300.
- [4] Feingesicht M, Olteanu S, Aitouche A and Belkoura L, HIL validation of an embedded system acting as a nonlinear Takagi-Sugeno state observer on an Arduino board , *Proceeding of the 6th European Embedded Design in Education and Research*, pp. 250-254, Milano, 11-12 September, 2014.
- [5] Lendek Z, Guerra T, Babuska R and de Schutter B, Stability Analysis and Nonlinear Observer Design Using Takagi-Sugeno Fuzzy Models, *Springer*, 2010.
- [6] Ghorbel H, El Hajjaji A, Souissi M and Chaabane M, Robust Tracking Control for Takagi-Sugeno Fuzzy Systems With Unmeasurable Premise Variables: Application to Tank System, *Journal of Dynamic Systems, Measurement and Control*, vol. 136, July, 2014.
- [7] Rincon-Pasaye J, R Martinez-Guerra and A Soria-Lopez, Fault diagnosis in nonlinear systems: An application to a three-tank system, *American Control Conference* , pp. 2136 – 2141, Seattle, June 2008.
- [8] Herrera-Lopez E, Castillo-Toledo B, Ramirez-Cordova J and Ferreira E, Takagi-Sugeno Fuzzy Observer for a Switching Bioprocess: Sector Nonlinearity Approach, *New Developments in Robotics Automation and Control*, InTech, chap. 8, pp. 155, 2008.
- [9] Yacine Z, Ichalal D, Oufroukh N, Mammar S, Vehicle nonlinear observer for state and tire-road friction estimation, *(ITSC)*, 2013 16th International IEEE Conference on Intelligent Transportation Systems , pp.2181,2186, 6-9 Oct. 2013