

Tutorial 7: Real data identification of an actual radiator-room system aimed to virtual sensor design

Stefano Malan¹ and Cosimo Greco¹

¹ Dip. di Automatica e Informatica, Politecnico di Torino,
Corso Duca degli Abruzzi 24, 10129, Torino, Italy
E-mail: stefano.malan@polito.it
E-mail: cosimo.greco@polito.it

Abstract. The tutorial presents both theoretical issues and practical hints on data analysis and processing, in order to identify, by means of techniques implemented by Matlab/Scilab tools, linear and affine mathematical models, which can be used to design a Virtual Sensor. Actual acquired thermal and hydraulic data are used to work out the proposed identification problem and to test the algorithms. The plant, a radiator-room system, is described and its inputs (commands/disturbances) and outputs (measured/estimated) are defined to state the thermal model identification problem. Different model typologies, such as transfer matrix or state space equations and linear or affine models, are described. Some practical suggestions on data pre-processing, as detrend and scaling operations, are also pointed out. To conclude, the obtained models are compared to evaluate their suitability in describing the plant behaviour. Eventually, some notes on Virtual Sensor design are given.

1. Introduction

In the last few years, energy management arose a great interest not only from an academic point of view, but also for practical reasons due to increasing restriction on energy consumption in civil and public building. Energy saving is nowadays necessary to reduce pollution and operating costs, while management of HVAC systems is aimed to maximize user comfort and fairly rearrange cost distribution on actual consumption. In this context, the possibility to reduce cost and complexity of monitoring and right accounting systems can be achieved by the use of a virtual or soft sensor. In particular, a temperature actual sensor, necessary for the heating system power computation, can be substituted by a virtual sensor, based on a identified model of the room-radiator complex. The use of a virtual sensor is motivated by cost reduction due to the high number of radiators to be monitored in apartment or office buildings and by the necessity to avoid the installation of actual devices easily accessible and that can be even tampered. The first step to design a virtual sensor is to identify a suitable mathematical model of the system able to provide, as an output, the signal to be measured.

Many papers can be find in the literature, dealing with buildings thermal dynamics identification.

For example [1] presents a procedure to identify SISO systems with a specific application to a HVAC plant of an office building, while [2] focuses on practical issues and conditions necessary to obtain a stable model. In [3] a dynamic thermal model is developed, using neural networks techniques, for one room in order to predict the energy consumption.

From a software point of view the use of Matlab Ident tool is presented in [4] to estimate thermal properties of a building.

A state space equation model is used in [5] to describe heat transfer in a multi-chamber system; while in [6] a simplified thermal model, a trade-off between physical and black-box models, is estimated,

using genetic algorithms, to predict the transient behaviour in building heating and cooling. Similar to a state space model is the multizone one described in [7] and used to simulate large scale buildings. Nevertheless, very few examples of thermal models, developed to design a virtual sensor, can be found in the literature and among them even a smaller number deals with building description. An example is given in [8], where a temperature virtual sensor is designed.

In this tutorial, different model structures, as well as different identification techniques, available from commonly used software as Matlab or Scilab, are compared to evaluate their suitability in describing the plant behaviour. Moreover, some practical suggestions on data pre-processing, such as detrend or scaling actions, are also pointed out.

2. Plant description

The plant under study is composed of a room, a university office, of about 12 square meters, with one side exposed to the external environment, partly through a quite wide window. The other three sides, as for the floor and the ceiling, are surrounded by other independently heated rooms (see Figure 1).

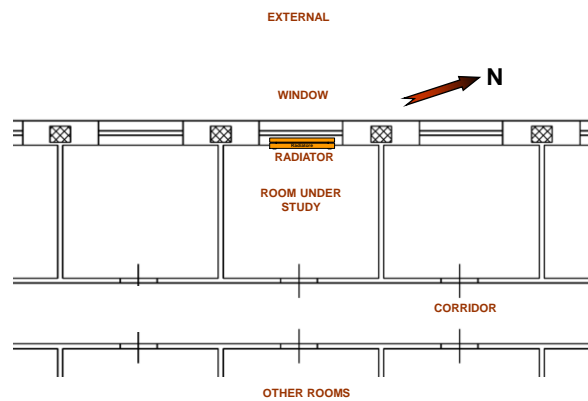


Figure 1. Plant sketch.

The considered room is heated by a radiator connected to the central heating unit. This means that it is not possible to adjust the input water temperature, while it is possible to regulate, by means of a valve, the water flow through the radiator.

The aim of the whole research program was to set up a methodology for micro-accounting the heating energy consumption, possibly for each room or even for each radiator, reducing at the same time the need of physical sensors, to be substituted by virtual or soft ones.

It is well known that the heating power is given by

$$P = Q \rho c (T_m - T_r) \quad (1)$$

where P is the power given by the heating water to the room by means of the radiator, Q is the flow in volume of heating water, ρ is the water density, c is the water specific heat, T_m is the water radiator input temperature and T_r is the water radiator output temperature. The heating energy is, of course, obtained by time integration.

Following equation (1), the measurement of three different signals is needed for micro-accounting: water flow and input and output temperatures. The aim of the research program was to substitute the

output actual temperature sensor with a virtual one, in order to cut the installation and management costs and to reduce the measurement set up complexity.

This way, the main objective of the present tutorial is to identify a model of the overall radiator-room system, suitable for the design of the virtual sensor.

In order to define the overall system, see Figure 2 for its continuous time (CT) representation, firstly we need to define its inputs and outputs. As output, the radiator output water temperature T_r is the main one, but also the room temperature T_a is chosen because it is needed to design the closed loop virtual sensor and also because, in general, from a mathematical model of a room-radiator system the knowledge of this second temperature is normally expected. As inputs, at least two or three signals must be selected: the radiator input water temperature T_m and the heating water flow Q are the mandatory ones, while the external environment temperature T_e can be neglected if its variations are quite small during the acquisition period or if the investigated room does not border the external environment. Nevertheless, if this last temperature measurement is available, it can be used to improve the accuracy of the obtained models.

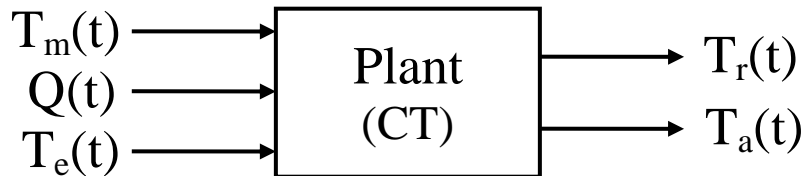


Figure 2. Block diagram of the actual plant.

Other variables act as inputs to the considered room, such as the temperatures of the surrounding rooms. These last are not considered in the model because they are supposed to be very close to the room temperature T_a ; therefore the heat exchanges between rooms are assumed negligible with respect to the heat exchange between the studied room and the external environment and between the heating water and the room, through the radiator. Furthermore, minimizing the number of inputs is a way to maintain the model as simple as possible, that is one of the usual identification requirements.

Besides inputs and outputs, also the model states can be defined. If we refer to the physical states, these correspond to the temperatures of all the homogeneous bodies that compose the model, i.e. radiator, “room” and walls temperatures. These components of the whole system can be spatially discretized originating a great number of states. The state equations are consequently defined by the heat exchanges equilibriums, involving physical parameters as thermal capacities and conductivities.

It is well known that the number of states defines the system order and that adding or subtracting a state requires to re-write the state equations. Authors’ aim was not to fix a-priori the system order, but to let it as an identification parameter to be obtained as result of an optimization process. Moreover, the physical parameters can be independently estimated with difficulty, in the sense that usually only some combinations of them can be obtained from parametric estimation.

Due to these motivations it was decided not to fix the system order and not to obtain physical parameters values, but to carry out a so called “black-box” identification, where the internal variables and parameters of the obtained model do not correspond to any physical variable or parameter.

3. Models choice

The radiator/room system is a continuous-time (CT) dynamic process (Figure 2). However, because of the need to sample the input and output signals, the same system must be represented by a discrete-time (DT) dynamic model. The technological actual configuration is in fact that depicted in Figure 3, where:

1. the ADC blocks represent multi-input *Analog to Digital Converters*, characterized by the same sampling interval T_s and number of bits N ;
2. the generic sampling instant t_i is given by $t_i = iT_s$, where “ i ” is an integer-valued variable;
3. “ i ” will represent from now on the DT independent variable.

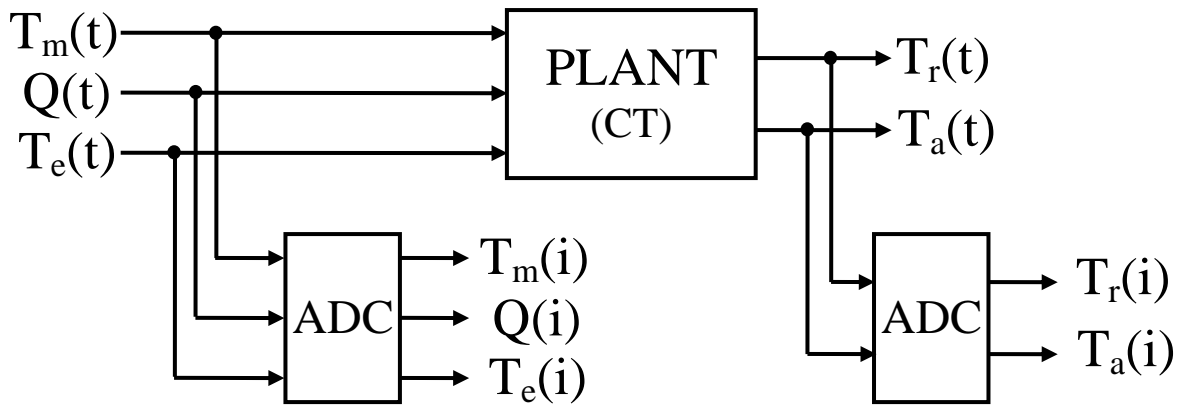


Figure 3. Plant and data acquisition system.

It is evident that the system viewed between the inputs $[T_m(i) \ Q(i) \ T_e(i)]^T$ and the outputs $[T_r(i) \ T_a(i)]^T$ has to be considered as a DT dynamic one. It is then natural to represent the original CT plant, see Figure 2, by a dynamic DT model as in Figure 4.

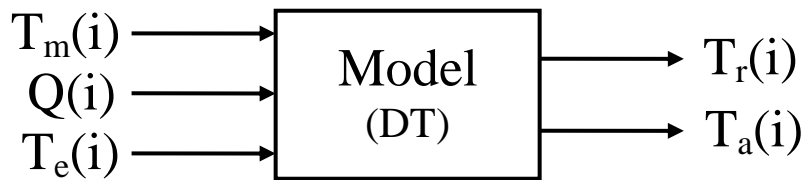


Figure 4. DT model of the CT system.

Moreover, although the room-radiator system is non-linear (weakly) and distributed-parameters, linear and lumped-parameters models have been adopted. This choice is mainly due to two reasons:

1. such models are simple and easy to analyze and manipulate;
2. literature offers in this field a great amount of results and tools.

In any case, the "goodness" of the identified models will be always verified a-posteriori. The dynamic, DT, linear and lumped-parameters model belong to two distinct classes:

- state space variable (SSV) models;
- input-output (I/O) models.

3.1 SSV models

These models are described as

$$\begin{cases} x(i+1) = Ax(i) + Bu(i) \\ y(i) = Cx(i) + Du(i) \\ \text{given the initial condition } x(0) \end{cases} \xrightarrow{\text{Z transform}} \begin{cases} zx(z) = Ax(z) + Bu(z) + zx(0) \\ y(z) = Cx(z) + Du(z) \end{cases} \quad (2)$$

where:

- x is the state column vector of dimension n (the model order);
- u is the input column vector of dimension n_u ;
- y is the output column vector, of dimension n_y ;
- A ($n \times n$), B ($n \times n_u$), C ($n_y \times n$) and D ($n_y \times n_u$) are the state space matrices.

Remember that, for a model built “from the inside”, according to the “transparent box” approach, the state vector x is made by the n “physical” state variables. In case of model built according to “black box” approaches, the n state variables have no physical meaning. This last was the authors’ choice.

Note. For an asymptotically stable system, the SSV model in stationary conditions, that is to say after the transient due to initial conditions ended, is the following:

$$\begin{cases} x(i+1) = Ax(i) + Bu(i) \\ y(i) = Cx(i) + Du(i) \end{cases} \xrightarrow{\text{Z transform}} \begin{cases} zx(z) = Ax(z) + Bu(z) \\ y(z) = Cx(z) + Du(z) \end{cases} \quad (3)$$

■

3.2 I/O model

In the I/O model the variables involved are the inputs u and the outputs y only. The model equation in the z domain is the following:

$$y(z) = G(z)u(z) + y_0(z) \quad (4)$$

where:

- y is the output column vector of dimension n_y ;
- u is the input column vector of dimension n_u ;
- G is the so called *transfer matrix* (tm) of dimensions $n_y \times n_u$;
- $G(z)u(z)$ represents the so called “forced response” (zero-state response)
- $y_0(z)$ is a column vector, of dimension n_y , which represents the so called “natural response” or zero-input response, which in turn depends on the initial conditions.

The elements of the tm $G(z)$ are scalar transfer functions (tf), rational proper, of order n , all having the same denominator of degree n (the characteristic polynomial $D_c(z)$); the numerators of such tf's, $N_{j,k}(z)$, are in general different from each other and of degree $m_{j,k} \leq n$ for all j, k , $1 \leq j \leq n_y$, $1 \leq k \leq n_u$.

The elements of y_0 are scalar functions, rational proper, of order n , all having the same denominator of degree n (the characteristic polynomial $D_c(z)$); the numerators of such functions, $N_{0,j}(z)$, are in general different from each other and of degree $m_{0,j} \leq n$ for all j , $1 \leq j \leq n_y$.

If the system is completely controllable and observable, the following result holds

$$D_c(z) = \det(zI - A) = \text{polchar}(A) \quad (5)$$

and, in general,

$$G(z) = C(zI - A)^{-1}B + D \quad (6)$$

The explicit form of the I/O model in the z domain is the following:

$$\begin{bmatrix} y_1(z) \\ y_2(z) \\ \vdots \\ y_{ny}(z) \end{bmatrix} = \frac{1}{D_c(z)} \begin{bmatrix} N_{1,1}(z) & N_{1,2}(z) & \cdots & N_{1,nu}(z) \\ N_{2,1}(z) & N_{2,2}(z) & \cdots & N_{2,nu}(z) \\ \vdots & \vdots & \cdots & \vdots \\ N_{ny,1}(z) & N_{ny,2}(z) & \cdots & N_{ny,nu}(z) \end{bmatrix} \begin{bmatrix} u_1(z) \\ u_2(z) \\ \vdots \\ u_{nu}(z) \end{bmatrix} + \frac{1}{D_c(z)} \begin{bmatrix} N_{01}(z) \\ N_{02}(z) \\ \vdots \\ N_{0ny}(z) \end{bmatrix} \quad (7)$$

Note. For an asymptotically stable system, the I/O model in stationary conditions is simply:

$$y(z) = G(z)u(z) \quad (8)$$

■

To explain the I/O model in the time domain it is suitable to refer to a single output y_j depending on a single input u_k :

$$y_j(z) = G_{jk}(z)u_k(z) = \frac{N_{jk}(z)}{D_c(z)}u_k(z) = \frac{b_{jk,n}z^n + b_{jk,n-1}z^{n-1} + \dots + b_{jk,1}z + b_{jk,0}}{z^n - a_{n-1}z^{n-1} - \dots - a_1z - a_0}u_k(z) \quad (9)$$

from which the difference equation is obtained

$$\begin{aligned} y_j(i) = & a_{n-1}y_j(i-1) + a_{n-2}y_j(i-2) + \dots + a_0y_j(i-n) + \\ & + b_{jk,n}u_k(i) + b_{jk,n-1}u_k(i-1) + \dots + b_{jk,0}u_k(i-n) \end{aligned} \quad (10)$$

With reference to the model (10), it should be noted that the $a_{(\cdot)}$ coefficients do not depend on j and k ; on the contrary for the $b_{(\cdot)}$ coefficients. In general, therefore, the complete model for any output y_j and all inputs u is as follows:

$$\begin{aligned} y_j(i) = & a_{n-1}y_j(i-1) + a_{n-2}y_j(i-2) + \dots + a_0y_j(i-n) + \\ & + b_{j1,n}u_1(i) + b_{j1,n-1}u_1(i-1) + \dots + b_{j1,0}u_1(i-n) + \\ & + b_{j2,n}u_2(i) + b_{j2,n-1}u_2(i-1) + \dots + b_{j2,0}u_2(i-n) + \\ & + b_{j3,n}u_3(i) + b_{j3,n-1}u_3(i-1) + \dots + b_{j3,0}u_3(i-n) + \\ & \vdots \\ & + b_{jn_u,n}u_{n_u}(i) + b_{jn_u,n-1}u_{n_u}(i-1) + \dots + b_{jn_u,0}u_{n_u}(i-n) \end{aligned} \quad (11)$$

Taking into account more than one output brings a similar expression where the $a_{(\cdot)}$ coefficients do not change, while the $b_{(\cdot)}$ are different according to indexes j and k .

4. Experimental setup and data

The experimental setup is shown in Figure 5.



Figure 5. Experimental setup.

The temperature sensors are the National Semiconductor LM35, with a scale of 10 mV/°C in the range $-40\text{ }^{\circ}\text{C}$ to $+110\text{ }^{\circ}\text{C}$ and an accuracy of $\pm 0.2\text{ }^{\circ}\text{C}$. They were equipped with a signal conditioner to adapt the signal range to the acquisition board input range and they were used not only to measure the heating water temperatures, but also the room and the external environment temperatures. Notice that the temperature sensors are not plunged in the heating water, but they are applied on the copper water pipe. This way they actually measure the pipe surface temperatures. This is not a problem because the error between water and pipe surface temperatures was estimated less than $0.07\text{ }^{\circ}\text{C}$, a value negligible with respect to the sensor accuracy. Moreover, as shown in equation (1), the heating power is proportional to the temperature difference, that cancel the described temperature error, supposed almost constant at radiator input and output pipes.

As water flow transducer, the ultrasonic, not invasive Dynasonic TFX Ultra was used. It outputs a current in the range $4\div 20\text{ mA}$, corresponding to a water velocity from 0 m/s to a settable maximum speed. The flow meter signal is consequently transformed in a voltage, by a $100\text{ }\Omega$ resistor, acquired, shifted to eliminate the offset and multiplied by the pipe section area, 49 mm^2 , to obtain the value of Q in equation (1).

At last, the acquisition board is a National Instruments USB-6211, $\pm 10\text{ V}$ of input range, 16 bit conversion accuracy, interfaced to a PC by LabView suitable programs.

The temperatures and water flow data were acquired for a period lasting few days in order to put in evidence the slow room thermal dynamics, with respect to the only available command, that is to say the heating water flow, regulated by means of a manual valve. The radiator input temperature time behaviour is imposed by the central heating system and cannot be tuned at will, acting this way as a disturbance. Of course, also the external environment temperature acts as a second disturbance. This last was used in some way as a command by opening and closing windows, even if this action changes the insulation structural characteristic of the room and is equivalent, from a model point of view, to modify the thermal conductivity between the room and the exterior. Anyway this was considered

acceptable because the aim was to obtain a model able to take into account the usual actions a person does inside a room, such as a window or door opening for a short time period.

The data were acquired with a sampling interval of 1 or 5 seconds. This sampling time is very short for a thermal system, but this choice is due to the necessity to put in evidence also the faster water flow dynamics, caused by the opening and closing of the valve. Moreover, in order to optimize the accuracy of the obtained models, the samples were subsequently decimated.

Two different acquisitions, taken in the last decade of March in Torino, Italy, are presented in Figure 6 and in Figure 7, where the sampling period, after decimation, was set to 15 s.

In Figure 7, about at samples 4500÷4800 and after sample 5500, it can be easily seen how the disturbance given by the opening of the window lower the room and the radiator output temperatures T_a and T_r .

Note also that the radiator input temperature T_m is measured downstream the valve (see Figure 5) so that when this last is closed, this temperature reduces even if the central system water temperature still remains at high values.

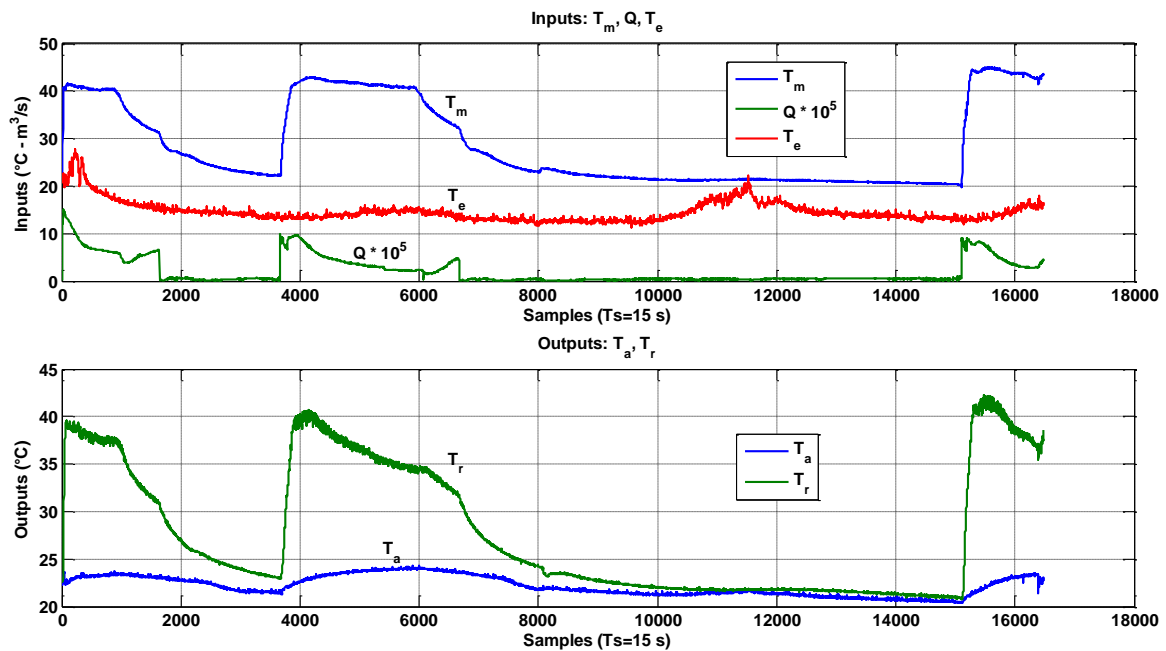


Figure 6. Three days acquired data.

5. The estimation method

To simplify the description of the parametric estimation method, it is suitable to develop the SISO case only. Under the assumptions that the system is strictly proper and of order n , the corresponding model results to be the following:

$$y(i) = a_{n-1}y(i-1) + a_{n-2}y(i-2) + \dots + a_0y(i-n) + b_{n-1}u(i-1) + b_{n-2}u(i-2) + \dots + b_0u(i-n) \quad (12)$$

or, in other form,

$$y(i) = \phi'_{i-1}\theta \quad (13)$$

where ϕ_{i-1} and θ are $2n$ -dimensional column vectors defined as follows:

$$\phi'_{i-1} \doteq [y(i-1) \ y(i-2) \ \dots \ y(i-n) \ u(i-1) \ u(i-2) \ \dots \ u(i-n)] \quad (14)$$

$$\theta' \doteq [a_{n-1} \ a_{n-2} \ \dots \ a_0 \ b_{n-1} \ b_{n-2} \ \dots \ b_0] \quad (15)$$

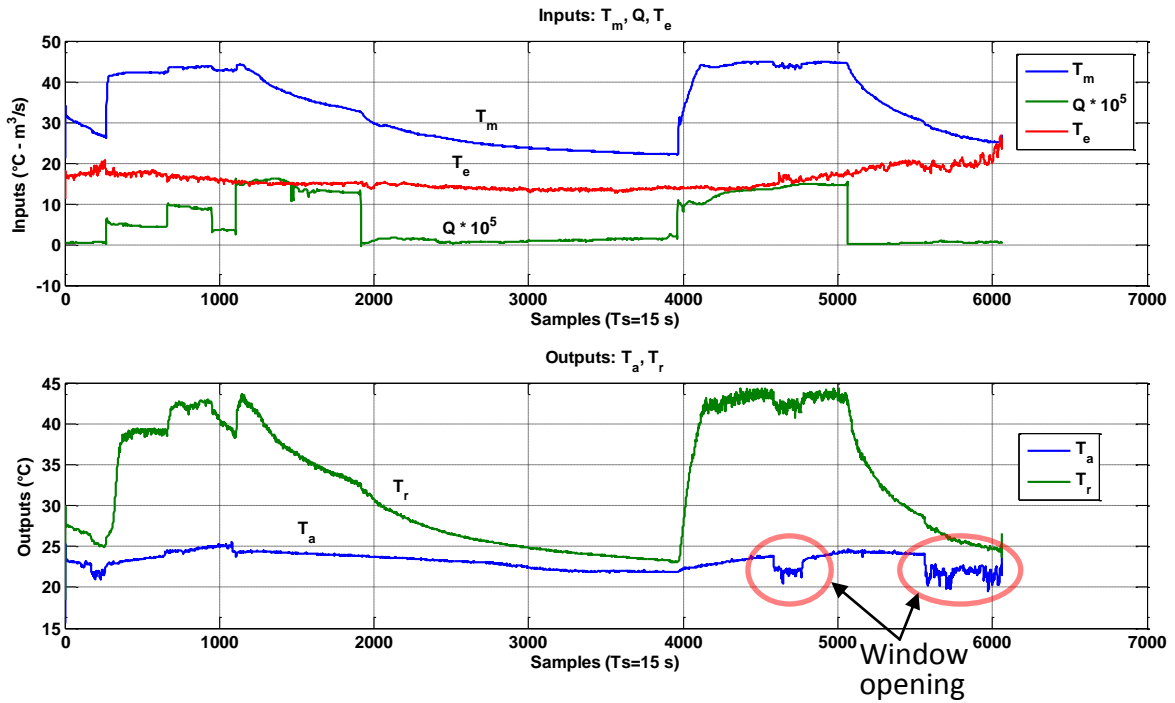


Figure 7. One day acquired data.

The problem is to estimate the numerical value of the model coefficients a and b – i.e. of θ – from samples of the I/O variables u and y ; the number of unknowns is then

$$m \doteq 2n \quad (16)$$

Under the hypothesis that the system parameters are time-invariant, the equation (13) can be also written for $y(i-1)$, for $y(i-2)$, \dots , for $y(i-N+1)$ giving:

$$\begin{bmatrix} \phi'_{i-N} \\ \vdots \\ \phi'_{i-2} \\ \phi'_{i-1} \end{bmatrix} \cdot \theta = \begin{bmatrix} y(i-N+1) \\ \vdots \\ y(i-1) \\ y(i) \end{bmatrix} \quad (17)$$

from which, defining $M_{i-1} \doteq \begin{bmatrix} \phi'_{i-N} \\ \vdots \\ \phi'_{i-2} \\ \phi'_{i-1} \end{bmatrix}$ and $y_i \doteq \begin{bmatrix} y(i-N+1) \\ \vdots \\ y(i-1) \\ y(i) \end{bmatrix}$, $N \times m$ matrix and $N \times 1$ vector respectively, the following synthetic form of (17) can be derived

$$M_{i-1}\theta = y_i \quad (18)$$

M_{i-1} and y_i are known terms at the instant i as they are formed by actual I/O samples from i backward to $i - N - n + 1$.

Equation (18) is nothing but a system of N equations in m unknowns.

The necessary condition for (18) is solvable, and then θ is estimable, is as follows:

$$N \geq m \quad (19)$$

It is in general convenient to choose

$$N \gg m \quad (20)$$

The chosen approach to solve the system (18) is the well-known *Least squares* (LS) one:

$$\min_{\theta} \left\{ \|y_i - M_{i-1}\theta\|^2 \right\} \longrightarrow \hat{\theta} = (M'_{i-1}M_{i-1})^{-1}M'_{i-1}y_i \quad (21)$$

Note. Because the solution $\hat{\theta}$ exists it is essential to ensure the following condition:

$$\text{rank}\{M_{i-1}\} = m \quad (22)$$

■

This condition implies special constraints on the columns of M_{i-1} then, ultimately, on the experimental conditions under which the I/O samples were acquired (see for example [9] and [10]).

5.1 Operational aspects

a. Multi-Input Multi-Output (MIMO) systems

In the case of MIMO processes, the equations system (18) always has the same form. With respect to the SISO case we have the following changes:

1. the number m of the unknown parameters, then the problem dimensions;
2. the way to build (and the form of) M_{i-1} ;
3. the way to build (and the form of) y_i .

b. SSV models

It is possible to set the least-squares estimation problem for SSV models also. More details can be found in the literature.

c. Offset in the I/O variables

It very often happens that the I/O variables of a real system should develop its dynamics in the neighbourhood of a constant value (offset). In such cases, to improve the numerical conditioning of the LS problem, it is suitable to "detrend" the variables values (for example subtracting from all

the samples the mean value of the variable itself). If detrend is not performed, or if the particular realization does not permit it – as in the case of real time estimation – or if the offset values are to be estimated, then it is appropriate to adopt an affine model, that is to say a linear model plus an offset. In the case of I/O models the following expression, derived from (10), can be used

$$y_j(i) = a_{n-1}y_j(i-1) + a_{n-2}y_j(i-2) + \dots + a_0y_j(i-n) + b_{jk,n}u_k(i) + b_{jk,n-1}u_k(i-1) + \dots + b_{jk,0}u_k(i-n) + \bar{Y}_{jk} \quad (23)$$

that, in the particular case of (12), becomes

$$y(i) = a_{n-1}y(i-1) + a_{n-2}y(i-2) + \dots + a_0y(i-n) + b_{n-1}u(i-1) + b_{n-2}u(i-2) + \dots + b_0u(i-n) + \bar{Y} \quad (24)$$

where \bar{Y} is a constant, to be estimated, and where the I/O variables $u(\cdot)$ and $y(\cdot)$ consist of not detrended values (the proof is given in [11]). In this case the equation (13) is yet valid,

$$y(i) = \varphi'_{i-1}\theta \quad (25)$$

but the two rhs factors are redefined as:

$$\varphi'_{i-1} = [y(i-1) \ y(i-2) \ \dots \ y(i-n) \ u(i-1) \ u(i-2) \ \dots \ u(i-n) \ 1] \quad (26)$$

$$\theta' = [a_{n-1} \ a_{n-2} \ \dots \ a_0 \ b_{n-1} \ b_{n-2} \ \dots \ b_0 \ \bar{Y}] \quad (27)$$

Similarly, in the case of SSV models the following expression, derived from (3), is used

$$\begin{cases} x(i+1) = Ax(i) + Bu(i) + \bar{X} \\ y(i) = Cx(i) + Du(i) + \bar{Y} \end{cases} \quad (28)$$

where \bar{X} and \bar{Y} are constants, to be estimated, and where the I/O variables $u(\cdot)$ and $y(\cdot)$ consist of not detrended values (the proof is given in [11]). In this case it is still possible to set up and solve the identification problem, using standard techniques, considering the “augmented” system

$$\begin{cases} x(i+1) = Ax(i) + [B \ \bar{X}] \begin{bmatrix} u(i) \\ 1 \end{bmatrix} \\ y(i) = Cx(i) + [D \ \bar{Y}] \begin{bmatrix} u(i) \\ 1 \end{bmatrix} \end{cases} \quad (29)$$

and identifying all the unknown $A, B, C, D, \bar{X}, \bar{Y}$.

Note. A constant value input (equal to 1) as in expressions (26) and (29) surely makes the identification problem numerically ill conditioned. This way, from a practical point of view, it is necessary to “perturb” with some random noise this input; of course, the amplitude of the noise

has to be chosen large enough to avoid the algorithm ill conditioning, but small enough in order not to modify the structure of the identification problem.

■

d. *Acquired data absolute values and numerical ill conditioning*

A very important practical hint to be considered, when using numerical algorithms, is that all the absolute values of the signals (plant inputs and outputs) given as input to the identification algorithm must be about of the same magnitude, in order to prevent numerical ill conditioning. If this condition is not satisfied the algorithms can fail to give a correct result because of numerical lack regarding, for example, the invertibility of a matrix or other algorithm requirements. Note that these last often are not satisfied not because of a requirement lack structurally related to the considered problem, but only because of a numerical ill conditioning that can be overcome by a suitable numerical values scaling. This is why, for example, in Figure 6 and in Figure 7 the values of the water flow Q is multiplied by a factor 10^5 . It was done not only to make it readable in the figures, but, more importantly, the same scaled values were used as inputs to the identification algorithms to prevent the described numerical faults.

5.2 Software tools

The solution of the LS problem is provided by a large number of algorithms and software tools, both for SISO and MIMO systems, both for off-line and real-time realizations. The results shown in this paper were obtained in *Matlab* environment, making use of the Ident Toolbox, but they have also been reiterated in *Scilab* environment and in C language.

6. Identification and validation results

In this section the results obtained using authors' scripts, developed in Matlab environment, together with Matlab Identification Toolbox, will be presented. Note that very similar results have been achieved using as well Scilab environment or C language scripts.

The data set shown in Figure 6 was used to identify the different models, while the one shown in Figure 7 was used to validate the obtained results. As a second step, here not reported for the sake of brevity, the roles of the two data sets were swapped to better check the identification procedure.

The following combinations of models and data pre-elaboration typologies were considered and the results will be marked using different indexes:

1. I/O models, without offset, identified from detrendized data (index *IOd*);
2. I/O models, with offset, identified from not detrendized data (index *IOo*);
3. SSV models, without offset, identified from detrendized data (index *SSd*);
4. SSV models, with offset identified, from not detrendized data (index *SSo*).

Of course other combinations of model and data pre-elaboration can be considered. For example, models without offset and data not detrendized can be as well identified in order to compare all the possible results. Nevertheless it does not make almost any sense, from a practical point of view, to consider other combinations, such as models with offset and detrendized data.

It is important to notice that one of the model parameters to be identified is the model order. It is well known that many identification techniques are able to find an "optimal", with respect to some cost criterion, model order and that these techniques are available in Matlab. In this work it was preferred to find a suitable model order by trial and error, starting from a first order and then increasing the order until a good correspondence between the acquired data and the simulated ones, assessed by direct inspection of the graphical plots and by numerical evaluation of the standard deviation, was obtained. Given the fact that, for practical applications, it is always better to have a low model order, the process was stopped when no significant gain in the above described criteria was obtained from further order increase. This way a second order was found as the suitable one and it is the one relating to the results presented in the next sections. Note that a second order is also the minimal one that can

be deduced from physical considerations, because at least two homogeneous bodies should be taken into account: the radiator and the “room”, meant as the complex of air, walls, etc. The temperatures of these two bodies constitute the physical state variables, which number is equal to the model order. Moreover, also the number of transmission zeroes is a model parameter and it can be set less than or equal to the model order (not greater because of physical feasibility constraints). It was set equal to the model order such to let the identification algorithms a greater number of degrees of freedom to find an optimal solution.

Note. To obtain the I/O model, without offset, identified from detrendized data, the Matlab ARX routine was not used because of the requirement to have the same denominator for all the transfer functions. In facts, this cannot be obtained by ARX routine, when systems with multiple outputs are considered: ARX routine identify a different denominator for each output.

■

6.1 Identification results

To check the suitability of the identified models, the two outputs T_r and T_a were considered. In Figure 8 it can be seen how all the four identified models fit the T_r acquired data. Almost no differences can be found between I/O models and SSV models.

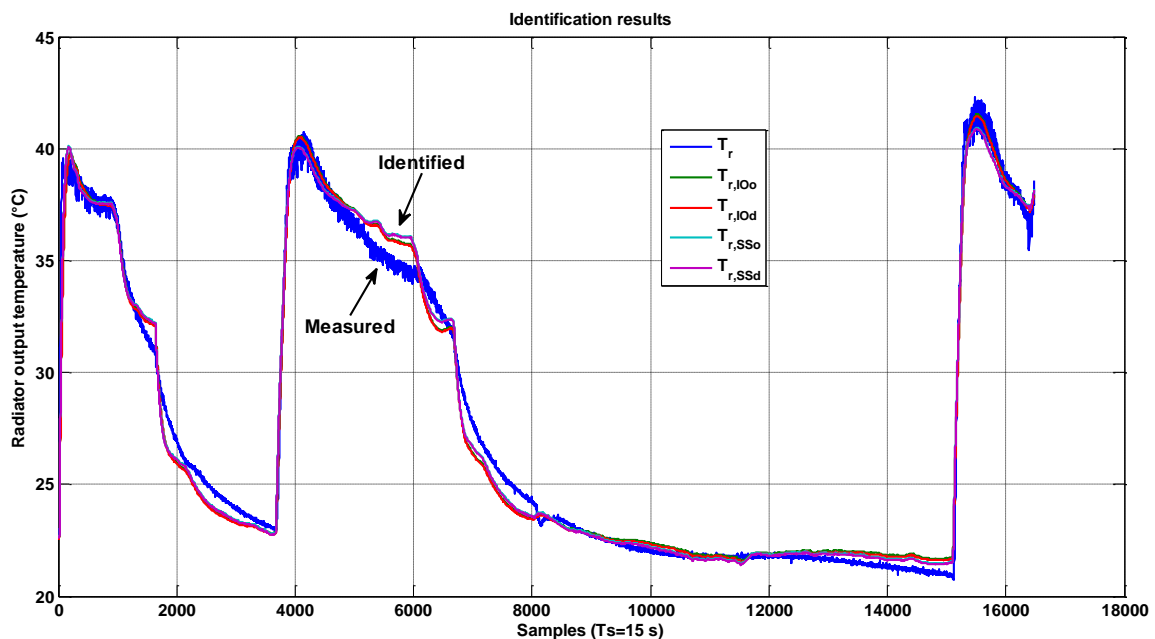


Figure 8. Identification results for radiator output temperature T_r .

On the contrary in Figure 9 the result for temperature T_a shows how the SSV models have a better fitting time behaviour with respect to I/O models, that show some not correct dynamics around samples 1800 and 6700. It must be anyway noticed that this not correct behaviour is also present in Figure 8, in which the smaller vertical scale hides it.

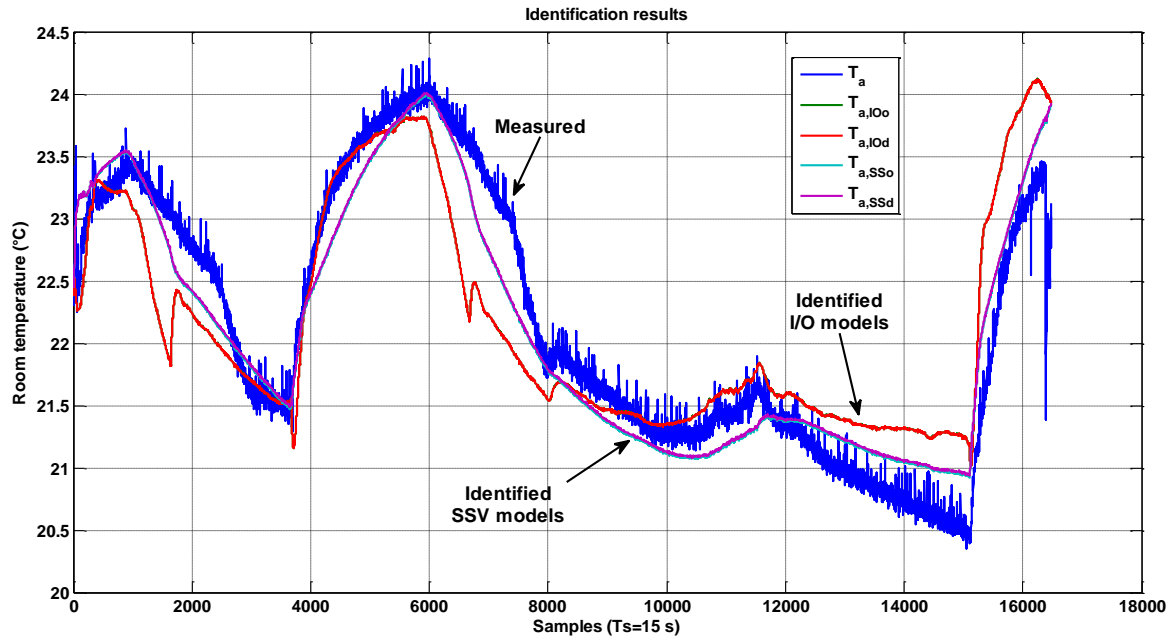


Figure 9. Identification results for room temperature T_a .

The obtained results can be also numerically evaluated by computing the standard deviation between acquired and simulated data, defined as

$$\sigma = \sqrt{\frac{\sum_{i=1}^l [y(i) - y_s(i)]^2}{l}} \quad (30)$$

where $y(\cdot)$ is the simulated output, $y_s(\cdot)$ is the acquired output and l is the samples number.

The standard deviation values are shown in Table 1. It can be noticed that the standard deviations for the SSV models are less than the ones of I/O models, for both outputs, thus confirming the better behaviour of SSV models. Note also that it does not make any sense to compare the standard deviation values of the two output temperatures, that is to say to compare the two values in each table column, because they are not normalized with respect, for example, the respective temperature mean value.

| Model | IOo | IOd | SSo | SSd |
|----------------------|--------|--------|--------|--------|
| σ_r for T_r | 0.7308 | 0.7324 | 0.6644 | 0.6667 |
| σ_a for T_a | 0.5341 | 0.5341 | 0.3035 | 0.3016 |

Table 1. Standard deviations for identified models.

To conclude, the identified DT transfer matrices are as follows.

$$\begin{bmatrix} T_r(z) \\ T_a(z) \end{bmatrix} = \begin{bmatrix} G_{11}(z) & G_{12}(z) & G_{13}(z) \\ G_{21}(z) & G_{22}(z) & G_{23}(z) \end{bmatrix} \begin{bmatrix} T_m(z) \\ 10^5 \cdot Q(z) \\ T_e(z) \end{bmatrix} \quad (31)$$

For model IOo

$$\begin{aligned} G_{11}(z) &= \frac{-0.010176 (z - 1.729) (z + 0.5303)}{(z - 0.9845) (z + 0.126)}; & G_{21}(z) &= \frac{-0.00018483 (z - 103.2) (z - 0.8742)}{(z - 0.9845) (z + 0.126)} \\ G_{12}(z) &= \frac{0.0027284 (z^2 + 1.566z + 1.162)}{(z - 0.9845) (z + 0.126)}; & G_{22}(z) &= \frac{0.0024779 (z - 1.793) (z + 0.6046)}{(z - 0.9845) (z + 0.126)} \\ G_{13}(z) &= \frac{-0.013828 (z - 0.8036) (z - 0.3412)}{(z - 0.9845) (z + 0.126)}; & G_{23}(z) &= \frac{-0.012715 (z - 1.125) (z - 0.2814)}{(z - 0.9845) (z + 0.126)} \end{aligned} \quad (32)$$

For model IOd

$$\begin{aligned} G_{11}(z) &= \frac{-0.0099935 (z - 1.745) (z + 0.5253)}{(z - 0.9845) (z + 0.1259)}; & G_{21}(z) &= \frac{-0.00018402 (z - 103.7) (z - 0.8742)}{(z - 0.9845) (z + 0.1259)} \\ G_{12}(z) &= \frac{0.0027247 (z^2 + 1.558z + 1.173)}{(z - 0.9845) (z + 0.1259)}; & G_{22}(z) &= \frac{0.002478 (z - 1.793) (z + 0.6044)}{(z - 0.9845) (z + 0.1259)} \\ G_{13}(z) &= \frac{-0.013841 (z - 0.8043) (z - 0.3396)}{(z - 0.9845) (z + 0.1259)}; & G_{23}(z) &= \frac{-0.012715 (z - 1.125) (z - 0.2815)}{(z - 0.9845) (z + 0.1259)} \end{aligned} \quad (33)$$

For model SSo

$$\begin{aligned} G_{11}(z) &= \frac{-0.030646 (z - 1.411) (z - 0.9985)}{(z - 0.9986) (z - 0.9798)}; & G_{21}(z) &= \frac{0.017656 (z - 0.9942) (z - 0.9521)}{(z - 0.9986) (z - 0.9798)} \\ G_{12}(z) &= \frac{0.0034834 (z + 2.341) (z - 0.9989)}{(z - 0.9986) (z - 0.9798)}; & G_{22}(z) &= \frac{0.0020483 (z - 1.033) (z - 0.9221)}{(z - 0.9986) (z - 0.9798)} \\ G_{13}(z) &= \frac{-0.015802 (z - 0.9992) (z - 0.8513)}{(z - 0.9986) (z - 0.9798)}; & G_{23}(z) &= \frac{-0.0092313 (z - 1.011) (z - 0.9759)}{(z - 0.9986) (z - 0.9798)} \end{aligned} \quad (34)$$

For model SSd

$$\begin{aligned} G_{11}(z) &= \frac{-0.030404 (z - 1.414) (z - 0.9985)}{(z - 0.9986) (z - 0.9798)}; & G_{21}(z) &= \frac{0.017674 (z - 0.9942) (z - 0.9522)}{(z - 0.9986) (z - 0.9798)} \\ G_{12}(z) &= \frac{0.0034558 (z + 2.366) (z - 0.9989)}{(z - 0.9986) (z - 0.9798)}; & G_{22}(z) &= \frac{0.0020333 (z - 1.033) (z - 0.922)}{(z - 0.9986) (z - 0.9798)} \\ G_{13}(z) &= \frac{-0.015828 (z - 0.9992) (z - 0.8516)}{(z - 0.9986) (z - 0.9798)}; & G_{23}(z) &= \frac{-0.0092488 (z - 1.011) (z - 0.9759)}{(z - 0.9986) (z - 0.9798)} \end{aligned} \quad (35)$$

As imposed, the denominators of each group of six transfer functions are the same, but in general differ from one group to another; the number of zeroes is equal to the number of poles and equal to the model order, that is to say 2, as previously explained. Moreover the two I/O models are very similar considering the numerical values of poles and zeroes, and the same happens for the two SSV models. On the contrary I/O models and SSV models are numerically quite different as expected from the inspection of Figure 8 and Figure 9. At last, note that the SSV models are presented as tf just to make possible a direct comparison among all the models.

6.2 Validation results

Validation results are presented in Figure 10 for temperature T_r and in Figure 11 for temperature T_a . For the first one the acquired data are fitted quite well by all the models and no particular differences can be noticed between I/O and SSV models. On the contrary, for temperature T_a , the situation is somehow worse and none of the models can be judged in general better than another. Anyway, as explained in the next section, the identification aim was to obtain a quite good model for the temperature T_r , for which a Virtual Sensor has to be designed, and this objective was reached by all the obtained models.

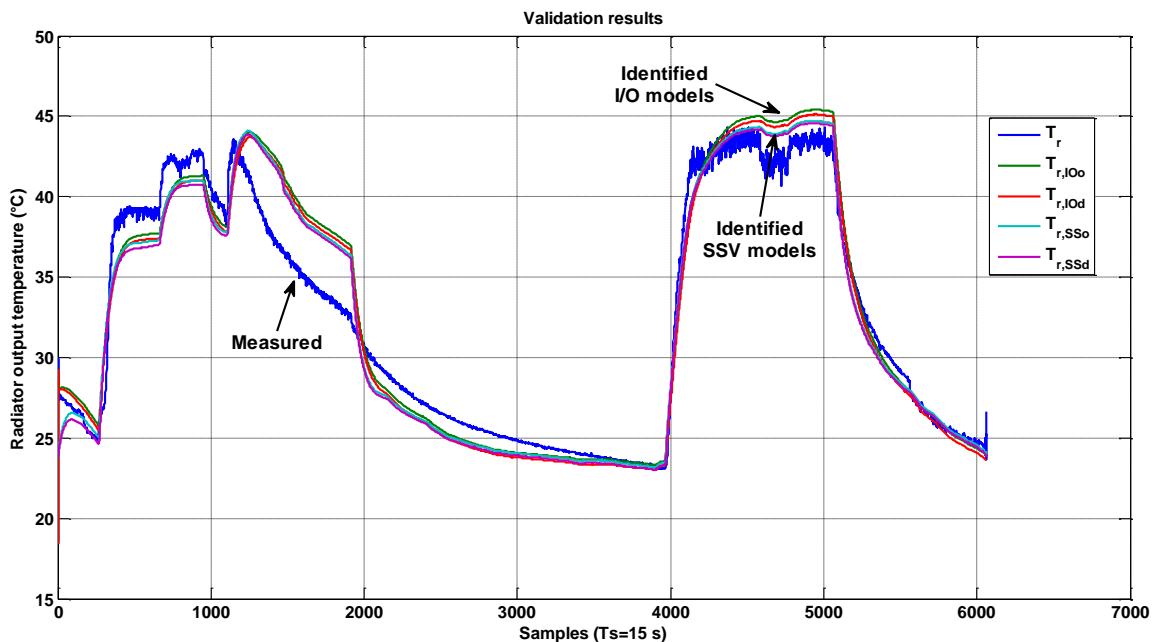


Figure 10. Validation results for radiator output temperature T_r .

Note. Some of the differences, let's say in the first 500 samples, are also given by a not perfect initial state computation that generates an initial transient producing simulated data different from the acquired ones.

■

7. Virtual sensors

A Virtual Sensor (VS) is a mathematical model able to give, in real time, the numerical value of a physical signal, that, for many reasons such as cost or accessibility, is not directly measured by an actual sensor. Considering the problem under investigation, the identified model, one of those presented in the previous section, is able to give the value of the radiator output temperature T_r on the basis of the measured values of the system inputs. In this case an *open-loop VS* is obtained.

The VS can be enhanced by designing some *closed-loop* correction able to reduce the unavoidable differences between the real plant and the identified model, thus to obtain, at least in some working conditions, the converge between the virtual measure and the (unknown) actual one. The closed-loop correction can be obtained by comparing the estimated and the measured values of a second output,

the temperature T_a in the considered plant, and feedback their suitably filtered difference (see, for example, [12]).

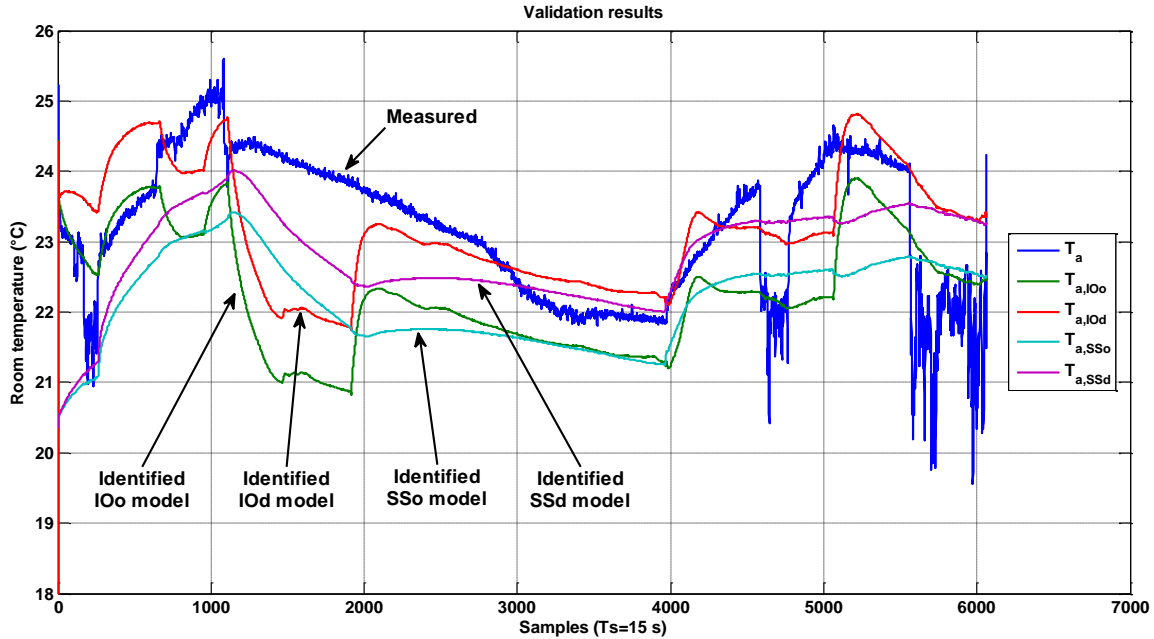


Figure 11. Validation results for room temperature T_a .

For the sake of simplicity we refer from now on to a generalized dual-input/dual-output (DIDO) model, for which the two inputs u_1 and u_2 are the generalization of the two actual plant inputs T_m and Q . Anyway, it is in general possible to extend the following presented results to a generic number of inputs, without changing the VS architecture. On the other hand it is also possible to consider the third input, the external temperature T_e , constant or “slowly” time-varying with respect to the time constants of the whole process. Note that these two inputs can be considered at the same time the plant and the model inputs.

About the two generic actual *plant* outputs y_{s1} and y_{s2} , the first one is the output to be estimated, the one that is not directly measured, and corresponds to the temperature T_r . On the contrary, the second output is directly measured and corresponds to the temperature T_a . Besides these two, also the *model* outputs y_1 and y_2 are considered hereafter.

In summary, the following general *working assumptions* are considered:

1. u_1, u_2 are measurable and are, at the same time, plant and model inputs;
2. y_{s1} is not measurable, it is to be reconstructed/estimated, and is one of the plant output;
3. y_{s2} is measurable and is the other plant output;
4. y_1, y_2 are the model outputs.

7.1 State space solution.

The first possibility to design the closed-loop VS is to consider a classical Luenberger estimator (see, for example, [13] and [14]), based on the state space DT model (2), as

$$\begin{cases} \hat{x}(i+1) = A\hat{x}(i) + B \begin{bmatrix} u_1(i) \\ u_2(i) \end{bmatrix} + L[y_{s2}(i) - C_2\hat{x}(i)] \\ \hat{y}_1(i) = C_1\hat{x}(i) \end{cases} \quad (36)$$

where \hat{x} is the estimated state, \hat{y}_1 is the estimated output and the gain L values are selected in order to obtain a closed loop asymptotically stable system with suitable time constants. In other words, the closed loop correction is based on the estimation error

$$e_2 \doteq y_{s2} - \hat{y}_2 = y_{s2} - C_2\hat{x} \quad (37)$$

and the gain L are chosen in order to ensure $\hat{y}_2 \rightarrow y_{s2}$ and to guarantee also $\hat{y}_1 \rightarrow y_{s1}$, so that the VS output \hat{y}_1 can be used as an indirect measure of the actual output y_{s1} , [11].

Equation (36) can be rewritten as follows

$$\begin{cases} \hat{x}(i+1) = (A - LC_2)\hat{x}(i) + \begin{bmatrix} B & L \end{bmatrix} \begin{bmatrix} u_1(i) \\ u_2(i) \\ y_{s2}(i) \end{bmatrix} \\ \hat{y}_1(i) = C_1\hat{x}(i) \end{cases} \quad (38)$$

where the closed loop matrix $(A - LC_2)$ is put in evidence. Note that y_{s2} (the room temperature T_a), that is one of the plant outputs, the actual measured one, becomes one of the inputs for the Virtual Sensor.

7.2 Input output solution.

A second possibility is to consider the I/O model (4), drawn in Figure 12:

$$Y(z) = G(z)U(z) \quad \text{where} \quad Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}; \quad U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}; \quad G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \quad (39)$$

and G has been computed by identification from sampled I/O data.

Let us define the actual system output Y_s as $Y_s = [y_{s1} \quad y_{s2}]^T$ and the following working assumptions:

1. the tf's G_{ij} have the same denominator but different numerators;

$$2. \quad G_{ij} = \frac{N_{ij}}{D}, \quad \forall i, j \rightarrow G = \frac{1}{D} \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} = \frac{N_G}{D}.$$

From equation (39) and Figure 12 y_2 can be computed as follows:

$$\begin{cases} y_1 = G_{11}u_1 + G_{12}u_2 \rightarrow u_2 = \frac{1}{G_{12}} y_1 - \frac{G_{11}}{G_{12}} u_1 \\ \downarrow \\ y_2 = G_{21}u_1 + G_{22}u_2 \rightarrow y_2 = \frac{G_{22}}{G_{12}} y_1 - \frac{G_{11}G_{22} - G_{12}G_{21}}{G_{12}} u_1 = \frac{G_{22}}{G_{12}} y_1 - \frac{\det(G)}{G_{12}} u_1 \end{cases} \quad (40)$$

or

$$y_2 = \frac{N_{22}}{N_{12}} y_1 - \frac{\det(N_G)}{N_{12}D} u_1 \quad (41)$$

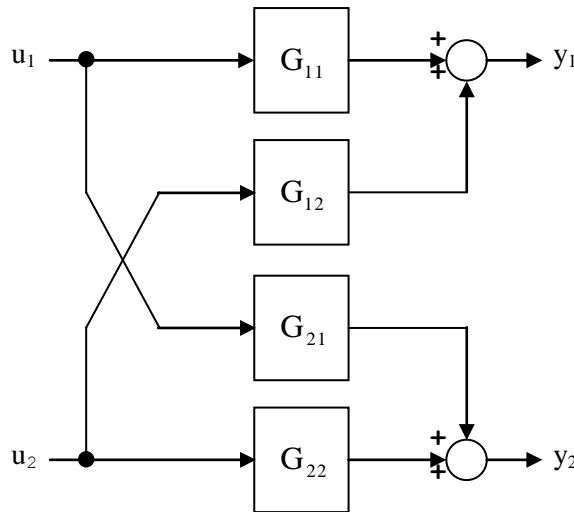


Figure 12. I/O model block diagram.

In conclusion, see also Figure 13:

$$y_2 = \frac{N_{22}}{N_{12}} y_1 - \frac{N_x}{N_{12}D} u_1 \quad (42)$$

where

$$\begin{aligned} N_x &= \det(N_G) \\ y_1 &= G_{11}u_1 + G_{12}u_2 \end{aligned} \quad (43)$$

Equation (42) puts in evidence the dependency of y_2 from y_1 and u_1 , that is of T_a from T_r and T_m .

In the block diagram of Figure 13 is already present a VS: y_l is an “open loop” estimate/reconstruction of y_{sl} , but in order to improve the reconstruction of y_l , it is reasonable to adopt again a “closed loop” configuration based on the estimation error $e_2 \doteq y_{s2} - y_2$, following the next steps and the block scheme in Figure 14:

- let the variable e_{F2} be defined as $e_{F2}(z) = \frac{N_F(z)}{D_F(z)} e_2(z)$ where the filter $\text{tf } \frac{N_F}{D_F}$ is to be designed;
- let y_l be “corrected” by the “correction signal” e_{F2} such that the VS output is finally defined as $\hat{y}_1 \doteq y_1 + e_{F2}$;

- \hat{y}_2 is defined as $\hat{y}_2 = Ky_2$ where the gain K , a further degree of freedom, is used to decouple the steady state gain of the y_{s2} path from that of u_1 and u_2 ;
- finally, the estimation error e_2 and the variable e_{F2} are redefined as $\hat{e}_2 = y_{s2} - \hat{y}_2$ and
- $$e_{F2}(z) = \frac{N_F(z)}{D_F(z)} \hat{e}_2(z)$$

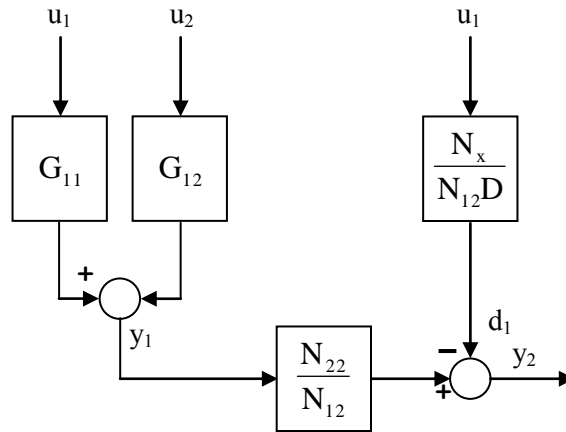


Figure 13. y_2 as a function of inputs.

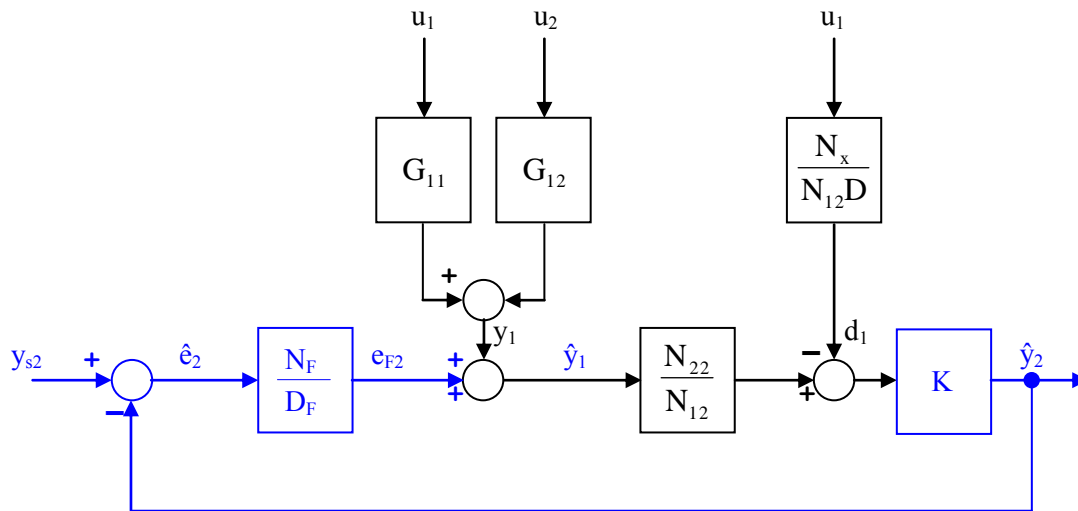


Figure 14. Closed loop VS of y_{s1} block diagram.

The idea, in other words, is to ensure $\hat{y}_2 \rightarrow y_{s2}$ despite the "disturbances" induced by y_1 and d_1 . At the same time $\hat{y}_1 \rightarrow y_{s1}$ will be guaranteed.

Some notes:

1. the tf's $\frac{N_x}{N_{12}D}$ and $\frac{N_{22}}{N_{12}}$ are proper or strictly proper;
2. it is convenient to have a "type 1" open loop by imposing $D_F(z) = (z-1)D'_F(z)$; this requirement guarantees the steady state condition $\hat{y}_2 = y_{s2}$ for y_{s2} constant or slowly time variant;

3. the item 2. also implies that for constant (or slowly time varying) values of y_I and d_I , the steady state condition $\hat{y}_2 = y_{s2}$ is still valid;
4. an appropriate design of $F(z) \doteq \frac{N_F}{D_F}$ and K can guarantee good closed loop stability margins and bandwidth (the proof is given in [11]).

For the sake of brevity only the previous few notes and no simulation results are reported here.

8. Conclusions

In this tutorial both theoretical issues and practical hints on data analysis and processing were presented. The achieved aim was to identify, by means of techniques implemented by standard software tools, linear and affine mathematical models, which can be used to design a Virtual Sensor. The considered plant was a radiator-room system, and different model typologies, such as transfer matrix or state space equations and linear or affine models, were used to describe its dynamic behaviour. Actual acquired thermal and hydraulic data were used to work out the proposed identification problem and to test the algorithms that showed their effectiveness to solve the problem. Some practical suggestions on data pre-processing, as detrend and scaling operations, were also pointed out. In conclusion, the obtained models were compared to evaluate their suitability in describing the plant behaviour. Eventually, some brief notes on Virtual Sensor design were also given.

9. References

- [1] Salsbury T I 2007 *Control Engineering Practice* **15** 109
- [2] Chen T Y and Athienitis 2003 *Building and Environment* **38** 1027
- [3] Kalogirou S A and Bojic M 2000 *Energy* **25** 479
- [4] Jimenez M J, Madsen H and Andersen K K 2008 *Building and Environment* **43** 170
- [5] Okuyama H 1990 *Building and Environment* **25** 349
- [6] Wang S and Xu X 2006 *Energy Conversion & Management* **47** 1927
- [7] Hong T and Jiang Y 1997 *Building and Environment* **32** 123
- [8] Zhang H, Krogh B, Moura J M F and Zhang W 2004 *Proc. IEEE Conf. on Decision and Control (Bahamas)* p 3792
- [9] Ljung L 1999 *System Identification - Theory for the User* (Upper Saddle River: Prentice-Hall)
- [10] Hanson R J and Lawson C L 1974 *Solving least squares problems* (Upper Saddle River: Prentice-Hall)
- [11] Greco C and Malan S 2010 *Internal Report*
- [12] Ergon R 1999 *IEEE Transactions on Automatic Control* **44** 821
- [13] Dorf R C and Bishop R H 2008 *Modern control systems* (Upper Saddle River: Pearson Prentice Hall) chapter 11 pp 769-772
- [14] Franklin G F, Powell J D and Emami-Naeini A 2006 *Feedback control of dynamic systems* (Upper Saddle River: Pearson Prentice Hall) chapter 7 pp 497-511

Acknowledgments

The presented research was partially funded by *Regione Piemonte WiFi4Energy Project* and by *VNTech Consortium research contract "Gestione automatizzata ed ottimizzata di impianti centralizzati di climatizzazione"*.