# State Space Realization of complex systems based on input/output data representation

1st Lotfi Hamrouni
*Department of Industrial Computing*
*University of Sousse, National School of Engineers*
Sousse, Tunisia
lotfi.hamrounii@gmail.com

2nd Kais Bouallegue
*Department of Electrical Engineering*
*University of Sousse, Higher Institute of Applied Sciences*
Sousse, Tunisia
bouallegue@yahoo.fr

3rd Youcef Soufi
*Department of Electrical Engineering*
*University Larbi Tebessi*
Tebessa, Algeria
soufi@yahoo.fr

***Abstract*—State space representation of complex systems is highly recommended in system control and automation. This theory which is usually based on physical and theoretical equations of the target system have no success when dealing with unknown systems that are represented only by input/output data measurements. in This paper, we study the possibility of extraction the state space equations of black box systems using a dataset of input/output measurements. For that, we use a subclass of input/output model which satisfies the necessary and sufficient conditions that guaranteed the realization of the state space model based on the input/output model. The resulted state-space model is identified using a feed-forward modular neural networks. This approach is applied to realize an industrial hydraulic systems.**

***Index Terms*—Modular Neural Networks, State space realization, identification and modeling**

## I. Introduction

Modelling of dynamical systems is a high topic in science an social fields that was used in many applications related to industrial processes, communication, electronics, traffics and many other things. Despite the great evolution of this theory, internal dynamical behaviours of nonlinear systems are still difficult to understand using classical modelling methods. This difficulty concerns black-box nonlinear systems which are represented only by input/output measurements that represent the relations between external inputs and outputs of the system. In spite of the success of modelling in characterising input/output relations, external input/output models alone are not suitable for many dynamical analysis and control applications. On the other hand, space models that arise naturally from the governing physical laws, and constitute the basis of stability analysis and feedback design of dynamical systems, are very difficult to achieve in nonlinear systems. On this way, many research has been done on the hope of constructing state space models based only on input/output data measurements. Few papers are published related to state space realization of nonlinear systems (See [1] [2] [3] for continuous-time systems and [4] [5] for discrete-time systems). The last results on the state space realisation are devoted to [6] [7] [5] which contribute to the establishment of the necessary and sufficient conditions that guaranteed the transformation of input/output model into an equivalent state space model. These conditions deal with the observability and controllability of the input/output model. In paper [8] these conditions are verified algebraically in continuous time domaine, the state coordinates of the model are realized first by finding the integrating factors and second by integrating certain 1-form subspaces. This procedure needs to apply the well-known Frobenius theorem and some elementary knowledge of differential 1-forms to extract the equivalent state space model. The paper [7] introduces the theoretical background of the discrete-realization theory using the differential geometry and propose explicitly the necessary and sufficient conditions to transform difference input-output map into a state space model.
In this paper, which extend the last author paper [9], we propose a general form of input/output model that satisfies the realizability conditions (controllability and observability) and we give the equivalent state space model in form of standard difference equations. The input/output model is based on a reduced NARX representation in which neural networks is used to identify the internal parameters of the model. This approach helps largely on the identification of state space coordinates based on input/output data measurements and the synthesis of output feedback controller of nonlinear systems.

The present paper is organized as follows: In section 2 we recall the essential theoretical backgrounds that formulate the necessary and sufficient conditions to transform an input/output model into a state-space model, next we propose a graphical map (based on the input/output model) that helps the design of accessible state-space models. Section 3 explains the realization of black-box systems using modular neural networks, and how to extract the corresponding state model of a non-linear system. In Section 4 we present simulation results to validate the theoretical approach. Finally, Section 5 concludes the paper.


ISSN: 2356-5608

## II. PRELIMINARIES ON STATE-SPACE REALIZATIONS

Consider a nonlinear discrete system $S$ represented by the Nonlinear AutoRegressive eXogenous (NARX) difference equation:

$$y[k] = \varphi(u[k-1],\ldots,u[k-n],y[k-1],\cdots,y[k-n]) \quad (1)$$

where $n$ is the order of the system, $\{u[k-i]\} \in \mathbb{R}^n$ and $\{y[k-i]\} \in \mathbb{R}^n$ are respectively the $n$ past values of the input and the output of the system, and $\varphi(\cdot)$ is a continuous function in $\mathbb{R}$.

**Assumption 1** *We assume that either* $\frac{\partial\varphi}{\partial u}$ *or* $\frac{\partial\varphi}{\partial y}$ *is different from zero.*

If the system (1) respects the conditions of controllability and observability, then (1) admits a state representation in the form :

$$\begin{aligned}\mathbf{x}[k+1] &= f(u[k],\mathbf{x}[k])\\ y[k] &= h(\mathbf{x}[k])\end{aligned} \quad (2)$$

where $\mathbf{x}$ is the state vector, and $f(\cdot,\cdot)$, $h(\cdot)$ are smooth ($\mathbb{C}^\infty$) functions.

Through the rest of the paper, the term "realization" means the transformation of the input/output model (1) into an equivalent state space representation such as (2).

**Problem Statement**: Given the input and output data from simple experiments, with input pulses, the goal is to extract the time-delay and construct a state-space representation such as (2).

In the rest of this section we formulate the input-output map that generate the dynamic of the system, next we extract the *minimal* state-space realization that is observable and controllable.

Indeed, the concept of observability and controllability is dual in the sense that a realization $M$ is observable if and only if the dual $M^{-1}$ is controllable and vice versa A realization $M$ is observable if the observability matrix $O(M)$ is full rank, then we can always reconstruct the initial state $x(0)$ from observing the evolution of the output. A realization is controllable if the controllability matrix $C(M)$is full rank, so for any initial state it is always possible to construct a sequence of input that conducts the system to the desired output.

Before formulating the realization structure, we first examine the input-output variables of the equation (1). We start by defining the variable blocs $\mathbf{y},\mathbf{u},\mathbf{v}$ as following

**Notation 1**

$$\begin{aligned}\mathbf{y}(t) &= (y(t),y(t+1),\cdots,y(t+m-1))\\ \mathbf{u}(t) &= (u(t),u(t+1),\cdots,u(t+m-1))\\ \mathbf{v}(t) &= (u(t+m),u(t+m+1),\cdots,u(t+2m-1))\end{aligned}$$

**Notation 2**

$$\begin{aligned}\mathbf{y}(t-m) &= (y_1,\cdots,y_m)\ ,\ for\ y_i = y(t-m+i-1)\\ \mathbf{u}(t-m) &= (u_1,\cdots,u_m)\ ,\ for\ u_i = u(t-m+i-1)\\ \mathbf{v}(t) &= (v_1,\cdots,v_m)\ ,\ for\ v_i = u(t+m+i-1)\end{aligned}$$

Using these notations, a step response realization can be constructed from model (1) by evaluating $y[t], y[t+1],\ldots,y[t+n-1]$, recursively in term of $y[t-1],\ldots,y[t-n],u[t-1],\ldots,u[t+n-1]$.
we obtain,

$$\begin{bmatrix} y(t)\\ y(t+1)\\ y(t+2)\\ \vdots\\ y(t+n-2)\\ y(t+n-1)\end{bmatrix} = \begin{bmatrix} \varphi(u_1 & \cdots & u_n & y_1 & \cdots & y_n)\\ \varphi(u_2 & \cdots & v_1 & y_2 & \cdots & y_{n+1})\\ \varphi(u_3 & \cdots & v_2 & y_3 & \cdots & y_{n+2})\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ \varphi(u_{n-1} & \cdots & v_{n-2} & y_{n-1} & \cdots & y_{2n-2})\\ \varphi(u_n & \cdots & v_{n-1} & y_n & \cdots & y_{2n-1})\end{bmatrix}$$
$$= \Phi\begin{bmatrix} u_1 & \cdots & u_n & y_1 & \cdots & y_n\\ u_2 & \cdots & v_1 & y_2 & \cdots & y_{n+1}\\ u_3 & \cdots & v_2 & y_3 & \cdots & y_{n+2}\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ u_{n-1} & \cdots & v_{n-2} & y_{n-1} & \cdots & y_{2n-2}\\ u_n & \cdots & v_{n-1} & y_n & \cdots & y_{2n-1}\end{bmatrix} \quad (3)$$

Which is equivalent to the following bloc input/output map equation:

$$\mathbf{y}(t) = \Phi(\mathbf{u}(t-m),\mathbf{y}(t-m),\mathbf{v}(t)) \quad (4)$$

where $\Phi = [\varphi_1,\varphi_2,\ldots,\varphi_n]^T$ represents the bloc input-output map of the system composed of $n$ partial function $\varphi_i(\mathbf{u},\mathbf{y},\mathbf{v})$ defined in (3) by:

$$\begin{aligned}\varphi_1(\mathbf{u},\mathbf{y},\mathbf{v}) &= \varphi(u_1,\cdots,u_n,y_1,\cdots,y_n)\\ \varphi_2(\mathbf{u},\mathbf{y},\mathbf{v}) &= \varphi(u_2,\cdots,u_n,y_2,\cdots,y_n,\varphi_1,v_1)\\ &\vdots\\ \varphi_i(\mathbf{u},\mathbf{y},\mathbf{v}) &= \varphi(u_i,\cdots,u_n,y_i,\cdots,y_n,\varphi_1,\cdots,\varphi_{i-1},\\ &\qquad v_1,\cdots,v_{i-1})\\ \varphi_n(\mathbf{u},\mathbf{y},\mathbf{v}) &= \varphi(u_n,y_n,\varphi_1,\cdots,\varphi_{n-1},v_1,\cdots,v_{n-1})\end{aligned}$$

The realization $\Phi(\mathbf{u},\mathbf{y},\mathbf{v})$ was studied largely in many paper [sadagh] to give the necessary an sufficient conditions that guaranteed the observability of the function $\Phi(\mathbf{u},\mathbf{y},\mathbf{v})$, and consequently the minimal representation of the matrix:

$$M = \begin{bmatrix} u_1 & \cdots & u_n & y_1 & \cdots & y_n\\ u_2 & \cdots & v_1 & y_2 & \cdots & y_{n+1}\\ u_3 & \cdots & v_2 & y_3 & \cdots & y_{n+2}\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ u_{n-1} & \cdots & v_{n-2} & y_{n-1} & \cdots & y_{2n-2}\\ u_n & \cdots & v_{n-1} & y_n & \cdots & y_{2n-1}\end{bmatrix} \quad (5)$$

This form of $M$ that is not necessary controllable, is the product of extended observability matrix and extended controllability matrix.

The generalization of Assumption 1 gives $\frac{\partial\Phi}{\partial\mathbf{y}}$ or $\frac{\partial\Phi}{\partial\mathbf{u}}$ is different from zero, and using the implicit theorem it can seen that $\Phi(\mathbf{u},\mathbf{y},\mathbf{v})$ is a local diffeomorphism with respect to $\mathbf{y}$. That is, there exist locally a smooth function $\Phi_y^{-1}$, such that $\mathbf{y} = \Phi_y^{-1}(\mathbf{u},\mathbf{x},\mathbf{v}) \Rightarrow \mathbf{x} = \Phi(\mathbf{u},\mathbf{y},\mathbf{v})$ for any constant $\mathbf{v}$.

Taking the partial derivative of $\mathbf{x} = \Phi(\mathbf{u},\Phi_y^{-1}(\mathbf{u},\mathbf{x},\mathbf{v}),\mathbf{v})$ with respect to $\mathbf{u}$. This results that

$$D_{\mathbf{u}}\Phi_y^{-1}(\mathbf{u},\mathbf{x},\mathbf{v}) = [D_{\mathbf{y}}(\Phi(\mathbf{u},\mathbf{y},\mathbf{v}))]^{-1}D_{\mathbf{u}}(\Phi(\mathbf{u},\mathbf{y},\mathbf{v})) \quad (6)$$

which is independent of the third variable $\mathbf{v}$.


ISSN: 2356-5608

**Definition 1** *: A realization $\Phi(\boldsymbol{u},\boldsymbol{y},\boldsymbol{v})$ is minimal if and only if it is controllable and observable.*

**Hypothesis 1** *: A realization $\Phi(\boldsymbol{u},\boldsymbol{y},\boldsymbol{v})$ is minimal if the differential $D_{\boldsymbol{x}}\Phi(\boldsymbol{u},\boldsymbol{y},\boldsymbol{v})$ with respect to the $\boldsymbol{y}$ and $\boldsymbol{u}$, are upper triangular matrices independent of the variable $\boldsymbol{v}$ having the form:*

$$D_{\boldsymbol{y}}(\Phi(\boldsymbol{u},\boldsymbol{y},\boldsymbol{v})) = \begin{bmatrix} \partial\varphi_1/\partial y_1 & \partial\varphi_1/\partial y_2 & \cdots & \partial\varphi_1/\partial y_n \\ 0 & \partial\varphi_2/\partial y_1 & \cdots & \partial\varphi_2/\partial y_{n-1} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \partial\varphi_n/\partial y_1 \end{bmatrix} \quad (7)$$

$$D_{\boldsymbol{u}}(\Phi(\boldsymbol{u},\boldsymbol{y},\boldsymbol{v})) = \begin{bmatrix} \partial\varphi_1/\partial u_1 & \partial\varphi_1/\partial u_2 & \cdots & \partial\varphi_1/\partial u_n \\ 0 & \partial\varphi_2/\partial u_1 & \cdots & \partial\varphi_2/\partial u_{n-1} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \partial\varphi_n/\partial u_1 \end{bmatrix} \quad (8)$$

*And the matrix $R = [D_{\boldsymbol{y}}(\Phi)]^{-1}D_{\boldsymbol{u}}(\Phi)$ is also independent of $\boldsymbol{v}$.*

Note that matrix $R$ play a key role on determining the realizability conditions.
$R$ has always an upper structure in the form:

$$R(\mathbf{u},\mathbf{y},\mathbf{v}) = \begin{bmatrix} \frac{\partial\varphi_1/\partial u_1}{\partial\varphi_1/\partial y_1} & r_{1,2} & \cdots & r_{1,n} \\ 0 & \frac{\partial\varphi_2/\partial u_1}{\partial\varphi_2/\partial y_1} & \ddots & r_{2,n-1} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \frac{\partial\varphi_n/\partial u_1}{\partial\varphi_n/\partial y_1} \end{bmatrix} \quad (9)$$

and the terms $r_{i,j}$ are given by

$$r_{i,j} = \frac{\frac{\partial\varphi_i}{\partial u_j} - \sum_{p=2}^{j} \frac{\partial\varphi_i}{\partial y_p} r_{(i+p-1,j-p+1)}}{\frac{\partial\varphi_i}{\partial u_1}} \quad (10)$$

In the case of a linear system $y[t] = \sum_{i=1}^{n} a_i y[t-i] + \sum_{i=1}^{n} b_i u[t-i]$, for all $i$ the matrix $R$ can be expressed in term of the coefficients of the input-output map as.

$$R = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ 0 & a_1 & \ddots & a_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & a_1 \end{bmatrix}^{-1} \begin{bmatrix} b_1 & b_2 & \cdots & b_n \\ 0 & b_1 & \ddots & b_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & b_1 \end{bmatrix} \quad (11)$$

Using the upper structure of matrices (7) (8) (9), we arrive at the following structure of the input-output bloc realization.

$$M = \begin{bmatrix} u_1 & u_2 & \cdots & u_n & y_1 & y_2 & \cdots & y_n \\ 0 & u_2 & \ddots & u_n & 0 & y_2 & \ddots & y_n \\ 0 & 0 & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & u_n & 0 & 0 & 0 & y_n \end{bmatrix} \quad (12)$$

The general form of the input-output bloc realization is given by

$$\tilde{M} = \begin{bmatrix} u_1 & \cdots & u_{1+d} & y_{1+d} & \cdots & y_{1+q} \\ u_2 & \cdots & u_{2+d} & y_{2+d} & \cdots & y_{2+q} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{n-q} & \cdots & u_{n-q+d} & y_{n-q+d} & \cdots & y_n \end{bmatrix} \quad (13)$$

for $q = 0,1,\ldots,n-1$ and $d = 0,1,\ldots,q$

And the dynamical input-output map $\Phi(\mathbf{u},\mathbf{y},\mathbf{v})$ is given by

$$\Phi(\mathbf{u},\mathbf{y},\mathbf{v}) = \begin{bmatrix} \varphi_1(u_1 & \cdots & u_{1+d} & y_{1+d} & \cdots & y_{1+(q+1)}) \\ \varphi_2(u_2 & \cdots & u_{2+d} & y_{2+d} & \cdots & y_{2+(q+1)}) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \varphi_i(u_i & \cdots & u_{i+d} & y_{i+d} & \cdots & y_{i+(q+1)}) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \varphi_{n-q}(u_{n-q} & \cdots & u_{n-q+d} & y_{n-q+d} & \cdots & y_n) \end{bmatrix} \quad (14)$$

That correspond to a realizable input-output model in the form

$$y[t] = \varphi(u_1,\ldots,u_n,y_1,\ldots,y_n) = \sum_{i=1}^{n-q} \varphi_i \quad (15)$$

The realizable analytic form of the model is given by

$$y[t] = \left[ \sum_{i=1}^{n-q-1} \varphi_i(u_i,\cdots,u_{i+d},y_{i+d},\cdots,y_{i+q+1}) \right] + \varphi_{n-q}(u_{n-q},\cdots,u_{n-q+d},y_{n-q+d},\cdots,y_n) \quad (16)$$

The state vector [10] is given by

$$\begin{aligned} x_1 &= y[k] \\ &\vdots \\ x_{q-d+1} &= y[k+q-d] \\ x_{q-d+2} &= y_1[k+q-d+1] \\ &\vdots \\ x_{n-p} &= y_{n-q-1}[k+n-d-1] \\ x_{n-p+1} &= u[k-d] \\ &\vdots \\ x_n &= u[k-1] \end{aligned} \quad (17)$$

with,

$$y_j[k+m] = \sum_{i=0}^{m-q-j-1} \varphi_i(y[k+d+i],\ldots,y[k+q+i+1], u[k+i],\ldots,u[k+d+i])$$
$$j = 1,\ldots,m-q-1$$
(18)

The state equations are obtained from the state vector as the following

$$\begin{aligned} x_1^+ &= x_2 \\ &\vdots \\ x_{q-d}^+ &= x_{q-d+1} \\ x_{q-d+1}^+ &= x_{q-d+2} + \varphi_{n-q}(u,x_1,\cdots,x_{q-d+1},x_{n-d+1},\cdots,x_n) \\ x_{q-d+2}^+ &= x_{q-d+3} + \varphi_{n-q+1}(u,x_1,\cdots,x_{q-d+1},x_{n-d+1},x_{n-d+1}^+,\cdots,x_n) \\ &\vdots \\ x_{n-d-1}^+ &= x_{n-d} + \varphi_2(u,x_1,\cdots,x_{q-d+1},x_{q-d+1}^+,x_{n-d+1},\cdots,x_n) \\ x_{n-d}^+ &= \varphi_1(u,x_1,,x_{q-d+1},x_{q-d+1}^+,x_{n-d+1},,x_n) \\ x_{n-d+1}^+ &= x_{n-d+2} \\ &\vdots \\ x_n^+ &= u \\ y &= x_1 \end{aligned}$$
(19)


ISSN: 2356-5608

State-space representation is an important property of the control system, which provides a convenient and compact way for its further modelling and analysis.

The construction of the state space model needs the identification of functions $\varphi_i$ which are separated in time and causes complexity when using standard identification toolbox like Matlab.
In this paper, we propose a modular neural networks as a solution to identify the state variables based on i/o data measurements taken from the real non-linear system.

## III. Modular Neural Networks State-Space Identification

The theory of realization presented in the last section was adopted to design the adequate Neural networks that represent the state space model based on the sub-class i/o model (16). By using a neural network with restricted connectivity structure, the structure of the NN state space model is depicted in Fig. 1 where each sub-NN represent a state variable of the system.

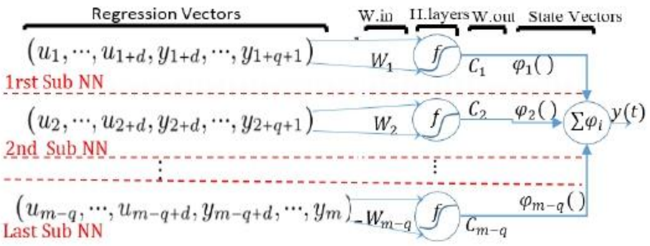


Fig. 1. Modular Neural-Network state space model

The output of the system is given by

$$y[t]=\left[\sum_{i=1}^{n-q-1} C_i f_i\Big(W[u_i,\cdots,u_{i+d},y_{i+d},\cdots,y_{i+q+1}]^T\Big)\right]$$
$$+C_{n-q}\, f_{n-q}\Big(W[u_{n-q},\cdots,u_{n-q+d},y_{n-q+d},\cdots,y_n]^T\Big)$$
$$=\sum\varphi_i$$
(20)

Where $W_i$, $C_i$ are input and output matrices of the synaptic weights of the i-th module, and $f_i(x) = 1/(1+e)^{-x}$ is an activation function of the ith sublayer neurons. Each module is chosen to have the same dimension as the whole network. The training is done in parallel using the well known Levenberg-Marquardt (LM) algorithm.
The main advantage of this structure is that state equations can be directly written down from the NN-model without any additional computations. The resulted State space equations are given by

$$x_1^+ = x_2$$
$$\vdots$$
$$x_{q-d}^+ = x_{q-d+1}$$
$$x_{q-d+1}^+ = x_{q-d+2} + C_{n-q}(W_{n-q}[u,x_1,\cdots,x_{q-d+1},x_{n-d+1},\cdots,x_n]^T)$$
$$x_{q-d+2}^+ = x_{q-d+3}+$$
$$+C_{n-q+1}(W_{n-q+1}[u,x_1,\cdots,x_{q-d+1},x_{n-d+1},x_{n-d+1}^+,\cdots,x_n]^T)$$
$$\vdots$$
$$x_{n-d-1}^+ = x_{n-d} + C_2(W_2[u,x_1,\cdots,x_{q-d+1},x_{q-d+1}^+,x_{n-d+1},\cdots,x_n]^T)$$
$$x_{n-d}^+ = C_1(W_1[u,x_1,,x_{q-d+1},x_{q-d+1}^+,x_{n-d+1},,x_n]^T)$$
$$x_{n-d+1}^+ = x_{n-d+2}$$
$$\vdots$$
$$x_n^+ = u$$
$$y = x_1$$
(21)

## IV. Realization of an Hydraulic Non-Linear System

The system considered in this paper is represented in figure2 , it is composed of an hydraulic tank with a pressure accumulator and a water flow.
The physical parameters of the system are:
$H_1$ = Height of the water in the tank
$H_2$ = Height of the water in the accumulator
$P_1$ = Pressure in the tank
$P_2$ = Pressure in the accumulator
$A_1$ = Cross sectional area of the tank
$A_2$ = Cross sectional area of the accumulator
$R_1$ = Fluid resistor between the tank and the accumulator
$R_2$ = Fluid resistor between the accumulator and the outside
$k$ = Spring constant of the spring in the accumulator
$u$ = Input volumetric flow rate of the hot water
$u_c$ = Input volumetric flow rate of the cold water
$Q_1$ = Volumetric flow rate across the first fluid resistor
$Q_2$ = Volumetric flow rate across the second fluid resistor
$T_h$ = Temperature of the input hot water
$T_c$ = Temperature of the input cold water
$T$ = Temperature of the water in the tank
$\rho$ = Density of the water
$g$=Acceleration of gravity

The system is assumed to be three dimension where the state space variables $x_1$,$x_2$ and $x_3$ represent the internal parameters $T$, $P_1$ and $P_2$ respectively; And the output is the temperature $T$.
The identification dataset is extracted from the following differential state equations given in [11].

$$\dot{x}_1 = \frac{\rho g}{A_1 x_2}\Big(uT_h + u_c T_c - x_1(u+u_c)\Big)$$
$$\dot{x}_2 = \frac{\rho g}{A_1}\Big(u + u_c - \frac{x_2 - x_3}{R_1}\Big) \qquad (22)$$
$$\dot{x}_3 = \Big(\frac{\rho g}{A_2} + \frac{k}{A^2{}_2}\Big)\Big(\frac{x_2 - x_3}{R_1} - \frac{x_3}{R^2}\Big)$$
$$y = x_1$$


ISSN: 2356-5608

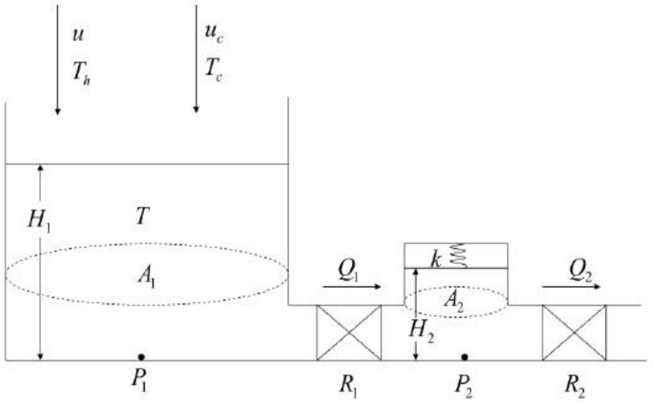


Fig. 2. Hydraulic non linear system

The input was chosen to be composed of the sum of four sine waves of different frequencies, 3, 4, 6, and 9. Each of the sine waves had a bias of 1 (i.e. mean value was 1) and the sum of the sine waves was multiplied by 0.005 to result in the mean value of the input to be $0.02m^3/s$. Plots of the input and the corresponding output are depicted respectively in figures (3, 4 , 5 and 10) .

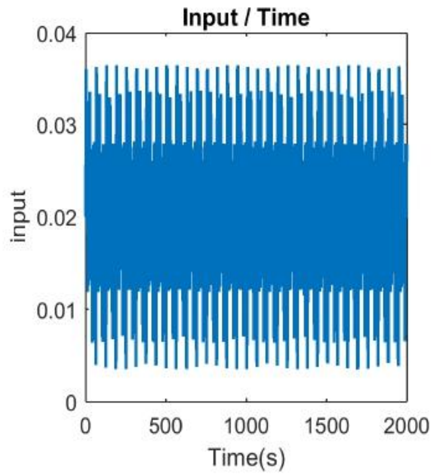


Fig. 3. Input of the system used for identification

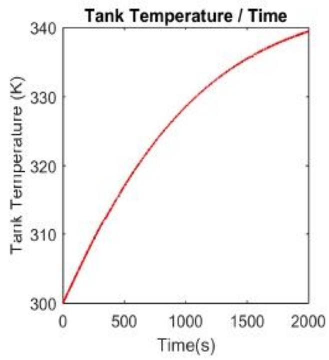


Fig. 4. Output Temperature of the Tank used for identification

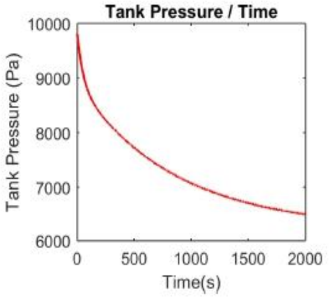


Fig. 5. Tank Pressure used for identification

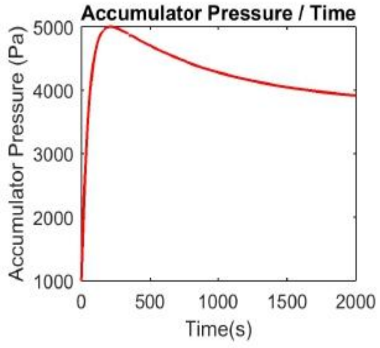


Fig. 6. Accumulator Pressure used for identification

### A. *Identification of state state space model*

To identify the state space variables of the system, we apply the theory of realization presented in section 3.
For that, we start by constructing the structure of the canonical state space model based on the model (16), we get the following state model taking $p = q = 0$.

$$\begin{bmatrix} \varphi_1\begin{pmatrix} u_1 & u_2 & u_3 \\ y_1 & y_2 & y_3 \end{pmatrix} \\ \varphi_2\begin{pmatrix} u_2 & u_3 \\ y_2 & y_3 \end{pmatrix} \\ \varphi_3\begin{pmatrix} u_3 \\ y_3 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} \varphi_1(u_1, y_1, y_2) \\ \varphi_2(u_2, y_2, y_3) \\ \varphi_3(u_3, y_3) \end{bmatrix} \quad (23)$$

The corresponding Modular Neural Networks is given in Figure 7, the learning data set is composed of 2000 samples divided into three groups: 50% for leaning, 25% for validation and 25% for testing. We use the Marquart-Levenburg algorithm to approximate the partial functions $\varphi_i$.

$$\varphi_1(u_1, y_1, y_2) = C_1 f_1(W_1[u_1, y_1, y_2]^T)$$
$$\varphi_2(u_2, y_2, y_3) = C_2 f_2(W_2[u_2, y_2, y_3]^T)$$
$$\varphi_3(u_3, y_3) = C_3 f_3(W_3[u_3, y_3]^T)$$


ISSN: 2356-5608

After learning the model parameters, the state space model is straightforward using (21) as the following

$$x_1 = y(t)$$
$$x_1^+ = x_2 + \varphi_3(u, x_1)$$
$$x_2^+ = x_3 + \varphi_2(u, x_1, x_1^+)$$
$$x_3^+ = \varphi_1(u, x_1, x_1^+)$$

The output of the actual system as well as the realized state-space model are shown on Fig.8, Fig.9, and Fig.10.

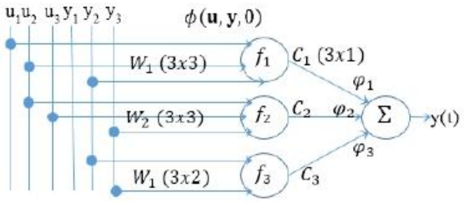


Fig. 7. Three Dimension Modular Neural-Network state space model

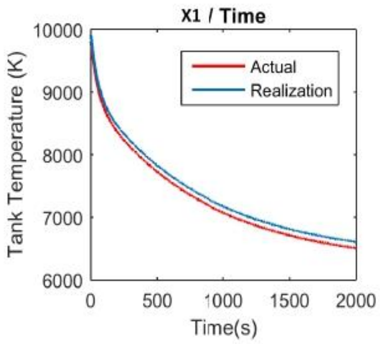


Fig. 8. Realized state space $x_1$ vs the real $x_1$

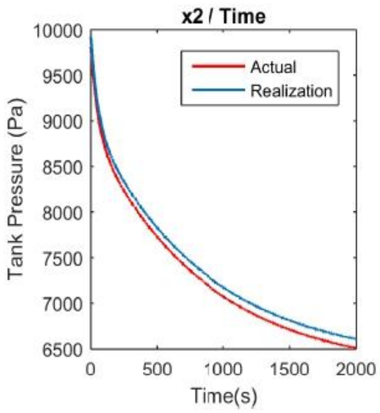


Fig. 9. Realized state space $x_2$ vs the real $x_2$

## V. Conclusion

The problem of identification and transforming of I/O models into state space equations is a challenge for many researchers. In this paper, we presented the mathematical background of this theory and formulated the necessary and sufficient conditions that, when respected, one can transform an input/output model to a state space representation when it is possible. To evaluate the performance of this approach, we used a nonlinear hydraulic system represented by a dataset of input/output to extract the equivalent state space model.

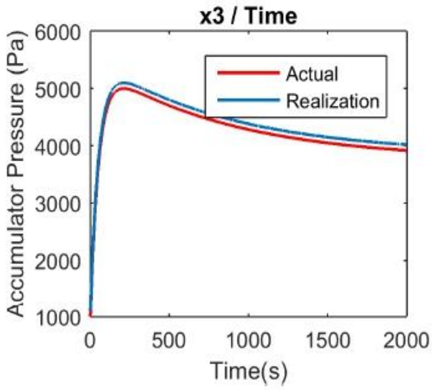


Fig. 10. Realized state space $x_3$ vs the real $x_3$

## References

[1] A. J. vanderSchaft. On realization of nonlinear systems described by higher-order differential equations. *Math. Syst. Theory*, 20(1):239–275, 1987.
[2] Belikov Juri, Kotta Ulle, and Tonso Maris. Adjoint polynomial formulas for nonlinear state-space realization. In *IEEE Transactions on Automatic Control*, volume 49, pages 256–261, 2013.
[3] Arvo Kaldme and Ulle Kotta. Realization of nonlinear input-output equations in controller canonical form. *Kybernetika*, 54(4):736–747, 2018.
[4] I. Leontaritis and S. Billings. Input/output parametric models for nonlinear systems: Deterministic nonlinear systems. *Int. Journal of Control*, 41(4):303–328, 1985.
[5] D. C. Foley and N. Sadegh. Modelling of nonlinear systems from input-output data for state space realization. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 3, pages 2980–2985, 2001.
[6] U. Kotta, S. Nomm, and F. Chowdhury. On realizability of neural networks-based input-output models in the classical state-space form. *Automatica*, 42(7):1211–1216, 2006.
[7] N. Sadegh. Minimal realization of nonlinear systems described by input-output difference equations. *IEEE Trans. Autom. Control*, 46(5):698–710, 2001.
[8] U. Kotta and M.Tonso. Realization of discrete-time nonlinear input-output equations: polynomial approach. *Automatica*, 48(2):255–262, 2012.
[9] L. Hamrouni. A state space i/o map to transform nonlinear systems into an observable form. *Int. J. Digital Signals and Smart Systems*, 1(2):163–180, 2017.
[10] U. Kotta and N. Sadegh. Two approaches for state space realization of narma models: Bridging the gap. *Mathematical and Computer Modelling of Dynamical Systems*, 8(1):21–32, 2002.
[11] G. F. Shoukry and N. Sadegh. State space realization for nonlinear systems. Master's thesis, Georgia Institute of Technology, 2008.


ISSN: 2356-5608