Dynamic Simulation of Quadruple Tank System

by

Asyraf bin Maskan

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

SEPTEMBER 2011

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Dynamic Simulation of Quadruple Tank System

by

Asyraf bin Maskan

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL ENGINEERING)

Approved by,

(DR. M. RAMASAMY)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
September 2011

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

(ASYRAF BIN MASKAN)

ABSTRACT

The problem of estimating state of dynamical system from only input and output measurement remain always an important field in the system theory. In fact, observers play a key roles during monitoring of process, a there are shown an essential component in many control application such as output regulation. Although the theories and applications for linear systems are well developed, development of observers for nonlinear system still provides an open area for research. Quadruple tank system and its mathematical model with typical parameters value will be collected from the reflected real system of quadruple tank. Dynamics simulation will be performed and through the MATLAB® enhancement. Various input changes take part. The result of the simulation will be analyzed and reported.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURE

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND OF STUDY

In the recent past, multi-variable control system design has been in great demand and need much attention in the process industry and academia. In many processes, when some or all of the manipulated variable affects more than its corresponding controlled variable, mean there are some interaction between the controlled variable, which may result in poor performance or even in instability of control process. When the interactions are not negligible, the plant must be considered as multiple inputs and multiple outputs. In this paper, a highly interactive multi-variable process has been considered i.e., quadruple tank problem. This multi-variable systems contains a transmission zeros, which can vary from left half plane (minimum phase) to right half plane (non-minimum phase) depending on the ratio of the flow to upper and lower tanks.

1.2 PROBLEM STATEMENT

1.2.1 PROBLEM IDENTIFICATION

Estimating the state of dynamical system of quadruple tank system from only input and output measurement remain always an important field in the system theory.

In fact, observers play a key roles during monitoring of process, there are shown an essential component in many control application such as output regulation. Although the theories and applications for linear systems are well developed, development of observers for nonlinear system still provides an open area for research. The main idea of this technique is to find some state transformation that make original system as a linear part, and nonlinear part depending only on measured states and inputs. The main drawback of this strategy is the difficulty to give necessary and sufficient conditions for existence of this transformation.

1.2.2 SIGNIFICANT OF THE PROJECT

1. Improve performance limitations in practice.
2. Design of quadruple tank system depends on the process parameter.

1.3 OBJECTIVES

1. To perform dynamic simulation of multivariable process for quadruple tank system.
2. To study the complexity of the mathematical model of multivariable.
3. To estimate state of dynamical system of quadruple tank system.

1.4 SCOPE OF STUDY

1.4.1 Dynamic mathematical model of quadruple tank system

1. The effect of time-varying dynamics should be considered when designing control systems
2. The sign of the steady-state gain should always be considered when designing control systems for multivariable processes
3. The cause of unexpected dynamic behaviour in control loops is often more subtle than what is first assumed
4. Under some conditions, full decoupling can lead to significantly worse performance than partial decoupling
5. Decoupling control can do more harm than good
6. Hysteresis effects should be considered when troubleshooting control problems

1.4.2 Input and output regulations of the system

The following two examples discuss various phenomena that specifically occur in MIMO feedback systems and not in SISO systems, such as interaction between loops and multivariable non-minimum phase behaviour.

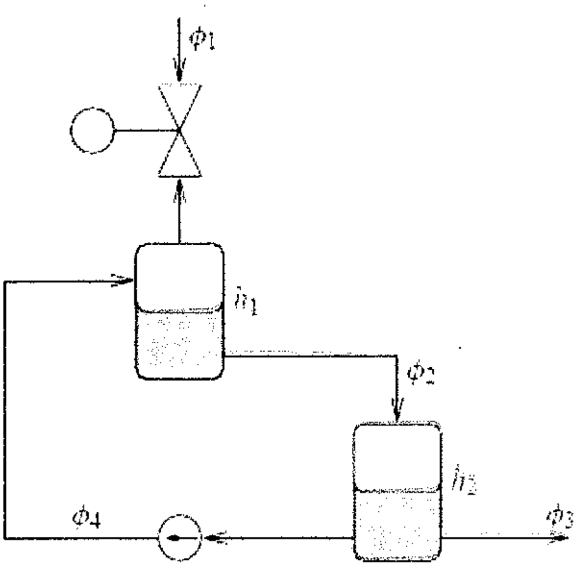

Figure 1.0: Two-tank liquid flow process with recycle

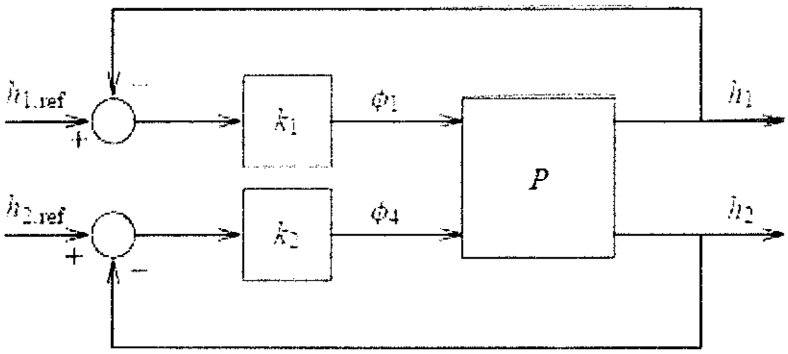

Figure 2.0: Two-loop feedback control of the two-tank liquid flow process

If the lower loop is closed with constant gain, then for high gain values (k_2→∞) the lower feedback loop is stable but has a zero in the right-half plane,

$$u_2(s) = -k_2 y_2(s), \tag{1}$$

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{s+1} & \frac{2}{s+3} \\ \frac{1}{s+1} & \frac{1}{s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}. \tag{2}$$

Thus under high gain feedback of the lower loop, the upper part of the system exhibits non-minimum phase behaviour. Conversely if the upper loop is closed under high gain feedback, $u_1 = k_1\, y_1$ with k_1 → ∞, then

$$y_2(s) = -\frac{s-1}{(s+1)(s+3)} u_2(s), \tag{3}$$

Apparently, the non-minimum phase behaviour of the system is not connected to one particular input-output relation, but shows up in both relations. One loop can be closed with high gains under stability of the loop, and the other loop is then restricted to have limited gain due to the non-minimum phase behaviour. Analysis of the transfer matrix

$$P(s) = \begin{bmatrix} \frac{1}{s+1} & \frac{2}{s+3} \\ \frac{1}{s+1} & \frac{1}{s+1} \end{bmatrix} = \frac{1}{(s+1)(s+3)} \begin{bmatrix} s+3 & 2s+2 \\ s-3 & s+3 \end{bmatrix} \tag{4}$$

shows that it loses rank at s =1. In the next section it will be shown that s = 1 is an unstable transmission zero of the multivariable system and this limits the closed loop behavior irrespective of the controller design method used. Consider the method of decoupling precompensation. (Qamar Saeed, 2010)

In the example the input-output pairing has been the natural one: output i is connected by a feedback loop to input i. This is however quite an arbitrary choice, as it is the result of our model formulation that determines which inputs and which outputs are ordered as one, two and so on. Thus the selection of the most useful input-output pairs is a non-trivial issue in multi-loop control or decentralized control, i.e. in control configurations where one has individual loops as in the example. A classical approach towards dealing with multivariable systems is to bring a multivariable system to a structure that is a collection of one input, one output control problems.

This approach of decoupling control may have some advantages in certain practical situations, and was thought to lead to a simpler design approach. However, as the above example showed, decoupling may introduce additional restrictions regarding the feedback properties of the system.

CHAPTER 2: LITERATURE REVIEW

2.1 PROCESS MODEL

The quadruple-tank process is a simple connection of two double-tank processes, which are standard processes in many control laboratories. The setup is thus simple but still the process can illustrate interesting multivariable phenomena. The process flow sheet is displayed in Figure 1. The target is to control the levels y1 and y2 in the lower two tanks with two pumps. The process inputs are υ1 and υ2 (input voltages to the pumps). The model used in the present virtual lab includes also the disturbance effect of flows in and out of the upper-level tanks.

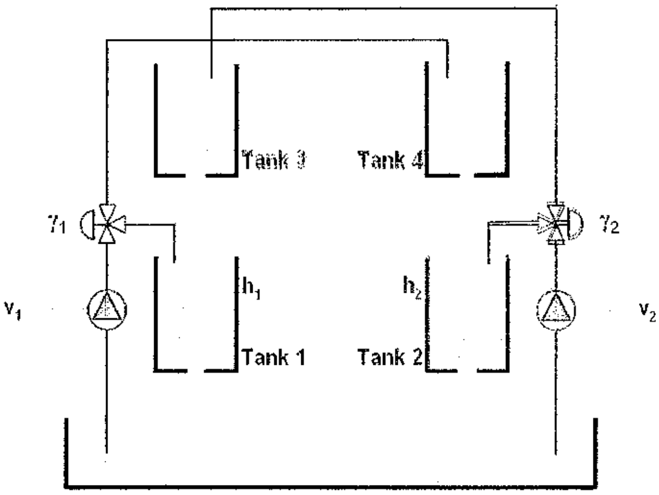

Figure 3.0: Schematic of the quadruple-tank process

Johansson (Johansson, 2000) described a laboratory quadruple-tank process which consists of four interconnected water tanks and two pumps as shown in Figure 1.0. The first principle mathematical model for this process is using mass balances and Bernoulli's law. The differential equations representing the mass balances in this quadruple-tank process are:

$$\frac{dh_1}{dt} = -\frac{a_1}{s_1(h_1)}\sqrt{2g|h_1|} - \frac{a_3}{s_1(h_1)}\sqrt{2g|h_3|} + \frac{\gamma_1 k_1 v_1}{s_1(h_1)}$$

$$\frac{dh_2}{dt} = -\frac{a_2}{s_2(h_2)}\sqrt{2g|h_2|} + \frac{a_4}{s_2(h_2)}\sqrt{2g|h_4|} + \frac{\gamma_2 k_2 v_2}{s_2(h_2)}$$

$$\frac{dh_3}{dt} = -\frac{a_3}{s_3}\sqrt{2g|h_3|} + \frac{(1-\gamma_2)k_2 v_2}{s_3} - \frac{k_{d_1} d_1}{s_3}$$

$$\frac{dh_4}{dt} = -\frac{a_4}{s_4}\sqrt{2g|h_4|} + \frac{(1-\gamma_1)k_1 v_1}{s_4} - \frac{k_{d_2} d_2}{s_4}$$

where hi is the liquid level in tank i;

ai is the outlet cross sectional area of tank i;

si(hi) is the cross-sectional area of tank i;

υj is the speed setting of pump j, with the corresponding gain kj;

γ j is the portion of the flow that goes into the upper tank from pump j;

and d1 and d2 are flow disturbances from tank 3 and tank 4 respectively, with corresponding gains kd1 and kd2.

The process manipulated inputs are υ_1 and υ_2 (speed settings to the pumps) and the measured outputs are y_1 and y_2 (voltages from level measurement devices). The measured level signals are assumed to be proportional to the true level, i.e., $y_1 = k_{m1}h_1$ and $y_2 = k_{m2}h_2$. The level sensors are calibrated so that $k_{m1} = k_{m2} = 1$. (Åström, 1992)

This process exhibits interacting multivariable dynamics because each of the pumps affects both of the outputs. The linearized model of the quadruple-tank process has a multivariable zero, which can be located in either the left or the right half-plane by simply adjusting the throttle valves γ1and γ2.

Johansson (Johansson, 2000) showed that the inverse response (non minimum phase) will occur when 0 < γ1+ γ2 < 1 and minimum phase for 1 < γ1+ γ2 ≤ 2. The valve settings will give then to the overall system entirely different behaviour from a multivariable control viewpoint. Unmeasured disturbances can be applied by pumping water out of the top tanks and into the lower reservoir. This exposes to disturbances rejection as well as reference tracking.

Parameters	Units	Values
A_1, A_3	$[cm^2]$	28
A_2, A_4	$[cm^2]$	32
a_1, a_3	$[cm^2]$	0.071
a_2, a_4	$[cm^2]$	0.057
k_c	$[V/cm]$	0.5
g	$[cm/s^2]$	981

Table 1.0: Parameter value for quadruple tank system

Operating Values	Units	P_-	P_+
(h_1^0, h_2^0)	$[cm]$	(12.4, 12.7)	(12.6, 13.0)
(h_3^0, h_4^0)	$[cm]$	(1.8, 1.4)	(4.8, 4.9)
(v_1^0, v_2^0)	$[V]$	(3.00, 3.00)	(3.15, 3.15)
(k_1, k_2)	$[cm^2/Vs]$	(3.33, 3.35)	(3.14, 3.29)
(γ_1, γ_2)	-	(0.70,0.60)	(0.43, 0.34)

Table 2.0: Operating parameter for minimun (P–) and non-minimum (P+) phases

The linearized state-space equation at operating points are $x_i = h_i - h^0_i$ and ui = $v_i - v^0_i$

.

$$\frac{dx}{dt} = \begin{pmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{pmatrix} x$$

$$+ \begin{pmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{pmatrix} u$$

$$y = \begin{pmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{pmatrix} x$$

where the time constants are $T_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^0}{g}}$.

Linearized transfer function matrix model for both minimum (P–) and non-minimum (P+) .

2.2 CURRENT PROCESS MODEL SIMULATION APPLICATION

2.2.1 Quadruple tank simulation

Most computer simulations of scientific phenomena can be described in terms of the model-control-view paradigm. This paradigm states that a simulation is composed of three parts:

1. The model, which describes the phenomenon under study in terms of
   - i. Variables, that hold the different possible states of
   - ii. The phenomenon
   - iii. Relationships among these variables (corresponding to the laws that govern the phenomenon)
   - iv. Expressed by computer algorithms

2. The control, which defines certain actions that a user can perform on the simulation

3. The view, which shows a graphical representation of the different states that the phenomenon can have. This representation can be done in a realistic or schematic form

The tool provides extensive scaffolding for creating the model but still leaves full flexibility for the analysis. This is pedagogically important, since the process of analysing good control fundamentals consists, to a great extent, in to know the basic principles to build models. In order to describe a model, the simulation needs to be able to:

1. Identify the set of variables that properly describe the system
2. Initialize, in a correct way, these variables
3. Describe how the value of these variables change in time
4. Establish how they affect each other when the user interacts with the system and modifies one or more of their values

5. Understanding control limitations due to interactions, model uncertainties, non-minimum phase behavior, and unpredictable time variations
6. Designing decentralized (often called "multiloop") controllers, and understanding their limitations
7. Implementing decouplers to reduce the effect of interactions, and understanding their limitations
8. Implementing a fully multivariable control system
9. Selecting the best control structure, based on the characteristics of the multivariable process

Past studies of (A. J. Krener, 1983) with 4-tank apparatuses implemented decentralized PI control, multivariable control, multivariable internal model control, and dynamic matrix control. The main educational focus was providing an apparatus with highly idealized and reproducible dynamics for use in illustrating multivariable interactions and multivariable transmission zeros as stated by Andersson (Andersson, 2002). In contrast, our main educational focus is to aid in understanding the advantages and disadvantages of the different control structures (e.g., decentralized, decoupling, multivariable) when applied to a multivariable process with interactions and dynamics ranging from highly ideal to highly non-ideal.

The main idea of this technique is to find some state transformation that make original system as a linear part, and nonlinear part depending only on measured states and inputs. The main drawback of this strategy is the difficulty to give necessary and sufficient conditions for existence of this transformation.

2.2.2 Developments in multivariable control simulation

Developing mathematical models of non-linear systems is a central topic in many disciplines of engineering. Models can be used for simulations, analysis of the system's behaviour, better understanding of the underlying mechanisms in the system, design of new processes and design of controllers. In a control system the plant displaying nonlinearities has to be described accurately in order to design an effective controller. In obtaining the mathematical model, the designer follows two methods.

The first one is to formulate the model from first principles using the laws governing the system. This is generally referred to as mathematical modelling.

The second approach requires the experimental data obtained by exciting the plant and measuring its response. This is called system identification and is preferred in the cases where the plant or process involves extremely complex physical phenomena or exhibits strong nonlinearities.

Obtaining a mathematical model for a complex system is complex and time consuming as it often requires some assumptions such as defining an operating point and doing linearization about that point and ignoring some system parameters.

In the present work three different models have been developed using three different soft computing techniques namely, ANN, Fuzzy and Neuro-fuzzy, for the Quadruple tank process. The rest of the paper is structured as follows: Section 2 describes the Quadruple Tank Process as stated by Rosenbrock (Rosenbrock, 1973).

2.2.3 Easy-Java simulations fundamentals

As stated by (Esquembre, 2002), Easy Java Simulations, Ejs for short, is a software tool that helps create dynamic, interactive scientific simulations in Java language.The tool is targeted for basic programming skills and is therefore very much suited to the pedagogical situation one finds in most university classrooms. Within Ejs, simulations are created by specifying a model for the simulated system and by building a view that continuously visualizes the state of this model and that readily responds to user interaction.

CHAPTER 3: METHODOLOGY/PROJECT OF WORK

3.1 METHODOLOGY

3.1.1 RESEARCH METHODOLOGY

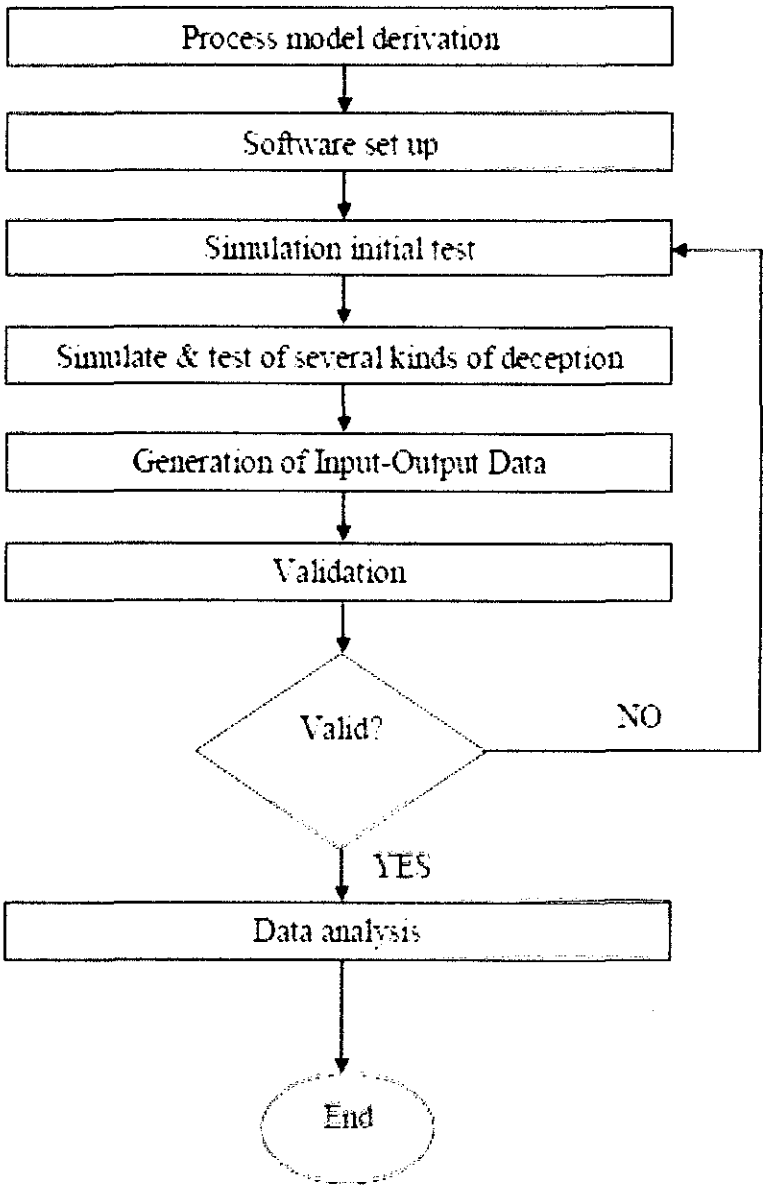

Figure 5.0: Research method flowchart

3.2 PROJECT ACTIVITIES

3.2.1 Process model derivation

The model theoretical design of quadruple tank model is referred as the main reference throughout the whole process of developing the dynamic simulation. All parameters are recognized, derived and all variables are defined.

Constant variables:

i. A1, A2, A3 and A4 are the cross-section area of each tank
ii. ai is the cross-section area of an outlet of the tank
iii. g is the gravitational constant, 981 cm/s^2

Manipulated parameters or input:

i. Voltage of the pumps
ii. Ratio of the flows

Output parameters:

i. Speed settings to the pumps
ii. Tank level

Process model:

```
dh = zeros(4,1);     % a column vector

dh(1) = -a1/A1*sqrt(2*g*h(1)) + a3/A1*sqrt(2*g*h(3)) +
g1_nmp*k1_nmp/A1*u1;
dh(2) = -a2/A2*sqrt(2*g*h(2)) + a4/A2*sqrt(2*g*h(4)) +
g2_nmp*k2_nmp/A2*u2;
dh(3) = -a3/A3*sqrt(2*g*h(3)) + (1-g2_nmp)*k2_nmp/A3*u2;
dh(4) = -a4/A4*sqrt(2*g*h(4)) + (1-g1_nmp)*k1_nmp/A4*u1;
```

3.2.2 Software set up

Apply the tools of software which are MATLAB® and Simulink. Coding and mathematical expression and functions are prepared using the tools. Whole system is then developed in order to run the whole dynamic simulation of the quadruple tank system.

The algorithms used in the ODE solvers vary according to order of accuracy and the type of systems (stiff or non-stiff) they are designed to solve. The solvers of the ODE suite can solve problems of the form $M(t, y)y' = f(t, y)$, with time- and state-dependent mass matrix M.

Solver	Problem Type	Order of Accuracy	When to Use
ode45	Nonstiff	Medium	Most of the time. This should be the first solver you try.
ode23	Nonstiff	Low	For problems with crude error tolerances or for solving moderately stiff problems.
ode113	Nonstiff	Low to high	For problems with stringent error tolerances or for solving computationally intensive problems.
ode15s	Stiff	Low to medium	If ode45 is slow because the problem is stiff.
ode23s	Stiff	Low	If using crude error tolerances to solve stiff systems and the mass matrix is constant.
ode23t	Moderately Stiff	Low	For moderately stiff problems if you need a solution without numerical damping.
ode23tb	Stiff	Low	If using crude error tolerances to solve stiff systems.

Table 3.0: ODE's Function in Matlab

3.2.3 Simulation initial test

This method is to calibrate the system developed using MATLAB® and Simulink. This is to prepare a good generation of data during the simulation.

Sample of Matlab coding for Run 1

```
function dh = height(t,h)
%Cross-section of the tank (cm2)
A1=28;
A2=32;
A3=28;
A4=32;

%Cross-section area of an outlet (cm2)
a1=0.071;
a2=0.057;
a3=0.071;
a4=0.057;

%Acceleration duo ti gravity (cm/s2)
g=981;

%Pump flow constants (cm3/sV)
k1_nmp = 0.5;
k2_nmp = 0.5;

%Ratio of the flows (lambda)
g1_nmp = 0.70;
g2_nmp= 0.30;

%Pump voltage (V)
u1=0.5;
u2=0.5;

dh = zeros(4,1);     % [illegible]

dh(1) = -a1/A1*sqrt(2*g*h(1)) + a3/A1*sqrt(2*g*h(3)) +
g1_nmp*k1_nmp/A1*u1;
dh(2) = -a2/A2*sqrt(2*g*h(2)) + a4/A2*sqrt(2*g*h(4)) +
g2_nmp*k2_nmp/A2*u2;
dh(3) = -a3/A3*sqrt(2*g*h(3)) + (1-g2_nmp)*k2_nmp/A3*u2;
dh(4) = -a4/A4*sqrt(2*g*h(4)) + (1-g1_nmp)*k1_nmp/A4*u1;
```

The function is called and the graph is plotted for data display.

```
[T,H] = ode45(@height,[0 120],[0 0 0 0]);
plot(T,H(:,1),'--',T,H(:,2),'*',T,H(:,3),'
',T,H(:,4),'.');
```

3.2.4 Generation of Input-Output Data

The data generated to train the network should contain all the relevant information about the dynamics of the Quadruple tank process. The input was given to the conventional model of the quadruple tank process, and from the conventional model, the input and output were sampled for each sampling instant and the required sampled data are obtained to train the network.

The quadruple tank system is divided with several conditions in order to define dynamic behaviour in the process by simulation.

1. Steady states

   The first simulation is to take measures in one or several steady states of the process (with several the mean value of the results is taken), knowing that if a dynamical system is in steady state, the rate of change of the state (h) is null.

$$\frac{dh}{dt} \approx 0$$

2. Valve constants

   γ_1 and γ_2 is derived. For that, all the output holes of the tanks must be covered, so that the first addends of all the expressions disappear, and consequently, second addends of the first and second equations have to be also removed. The simulation for valve constants is run.

   If that input is constant, a derivative can be expressed in non-infinitesimal time periods.

$$\frac{dx}{dt} \approx \frac{\Delta x}{\Delta t}$$

$$\frac{\Delta h_1}{\Delta t} = \frac{\gamma_1 k_1}{A_1} u_1$$

$$\frac{\Delta h_2}{\Delta t} = \frac{\gamma_2 k_2}{A_2} u_2$$

$$\frac{\Delta h_3}{\Delta t} = \frac{(1-\gamma_2) k_2}{A_3} u_2$$

$$\frac{\Delta h_4}{\Delta t} = \frac{(1-\gamma_1) k_1}{A_4} u_1$$

The parameters k_1, u_1, k_2, u_2 are constant, as A_1, A_2, A_3. A_4. Moreover, the simulation's time doesn't matter, because it can be added to the constant member k_c, considering $A_1 = A_2 = A_3 = A_4$.

3. Water tanks hole areas and pump constants

In this section, the values of the following not directly measurable parameters k_1, k_2, a_1, a_2, a_3, a_4 are derived, knowing A_1, A_2, A_3. A_4 those are easily measurable.

4. Parameters values and final model

Values ai of holes areas and k of the pumps are obtained just substituting values of bi in the first simulation to take ci and undoing the variable changes of both kinds of variables. These last parameters are:

$A_1 = A_2 = A_3 = A_4 = 15.21\ \text{cm}^2$
$a_1 = 0.2143\ \text{cm}^2$
$a_2 = 0.173\ \text{cm}^2$
$a_3 = 0.2102\ \text{cm}^2$
$a_4 = 0.1793\ \text{cm}^2$
$k_1 = 4.0356\ \text{cm}^3/\text{Vs}$
$k_2 = 3.9375\ \text{cm}^3/\text{Vs}$

Therefore, the final model with minimum phase configuration is the following:

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \\ \dot{h}_3 \\ \dot{h}_4 \end{bmatrix} = \begin{bmatrix} -0.6242 & 0 & 0.6121 & 0 \\ 0 & -0.5038 & 0 & 0.5223 \\ 0 & 0 & -0.6121 & 0 \\ 0 & 0 & 0 & -0.5223 \end{bmatrix} \begin{bmatrix} \sqrt{h_1} \\ \sqrt{h_2} \\ \sqrt{h_3} \\ \sqrt{h_4} \end{bmatrix} + \begin{bmatrix} 0.1531 & 0 \\ 0 & 0.1925 \\ 0 & 0.072 \\ 0.0924 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

This model of simulation is linearized to design the linear controllers chosen in this thesis, and the derivation of the model is repeated for the non-minimum phase process, with the same parameters.

3.2.5 Validation

The final step in developing the model is validation of the generated results of the simulation. Validation is performed by evaluating the simulation performance using trained or experimental data and test data. The input and target were presented. If the data is not valid within the range of validation, the method is re-looping start from the simulation initial test. If the data is valid, the data is analysed.

3.2.6 Data analysis

Data analysis is performed by comparing the expected results and data referred to the previous valid data of previous experiment of quadruple tank system. Several considerations are important;

1. The effect of time-varying dynamics should be considered when designing control systems
2. The sign of the steady-state gain should always be considered when designing control systems for multivariable processes
3. The cause of unexpected dynamic behaviour in control loops is often more subtle than what is first assumed
4. Under some conditions, full decoupling can lead to significantly worse performance than partial decoupling
5. Decoupling control can do more harm than good
6. Hysteresis effects should be considered when troubleshooting control problems

CHAPTER 4: RESULTS AND DISCUSSION

4.1 RESULTS

4.1.1 Steady states

Take measures in one or several steady states of the process (with several the mean value of the results is taken), knowing that if a dynamical system is in steady state, the rate of change of the state (*h*) is null.

```
function dh = height(t,h)
 %Cross-section of the tank (cm2)
A1=15.21;
A2=15.21;
A3=15.21;
A4=15.21;

%Cross-section area of an outlet (cm2)
a1=0.2143;
a2=0.1773;
a3=0.2102;
a4=0.1793;

%Acceleration due ti gravity (cm/s2)
g=981;

%Pump flow constants (cm3/sV)
k1_nmp = 4.0356;
k2_nmp = 3.9375;

%Ratio of the flows (lambda)
g1_nmp = 0.69;
g2_nmp= 0.74;

%Pump voltage (V)
u1=0.5;
u2=0.5;

dh = zeros(4,1);     % a column vector

dh(1) = -a1/A1*sqrt(2*g*h(1)) + a3/A1*sqrt(2*g*h(3)) +
g1_nmp*k1_nmp/A1*u1;
dh(2) = -a2/A2*sqrt(2*g*h(2)) + a4/A2*sqrt(2*g*h(4)) +
g2_nmp*k2_nmp/A2*u2;
dh(3) = -a3/A3*sqrt(2*g*h(3)) + (1-g2_nmp)*k2_nmp/A3*u2;
dh(4) = -a4/A4*sqrt(2*g*h(4)) + (1-g1_nmp)*k1_nmp/A4*u1;
```

```
[T,H] = ode45(@height,[0 120],[0 0 0 0]);
plot(T,H(:,1),'--',T,H(:,2),'*',T,H(:,3),'
',T,H(:,4),'.');
```

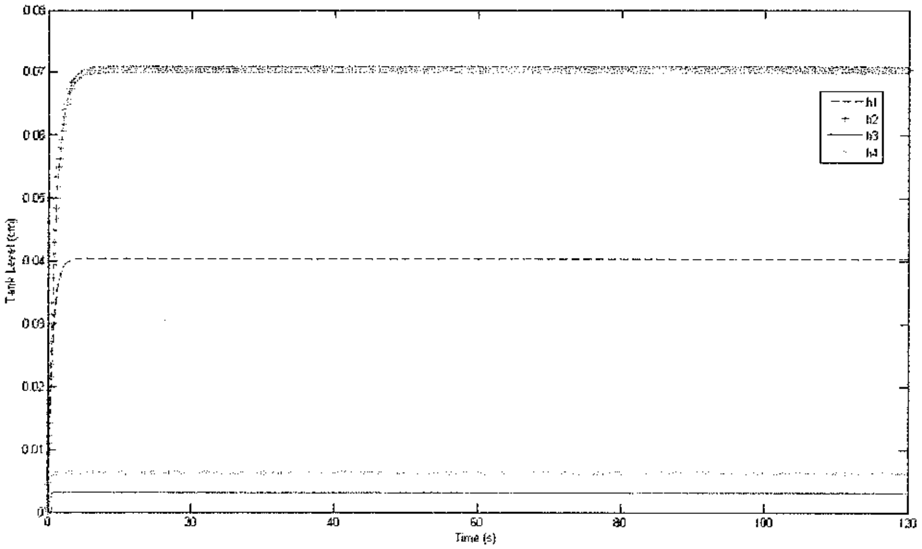

Figure 4.0: Simulation results for steady state condition

4.1.2 Valve constant

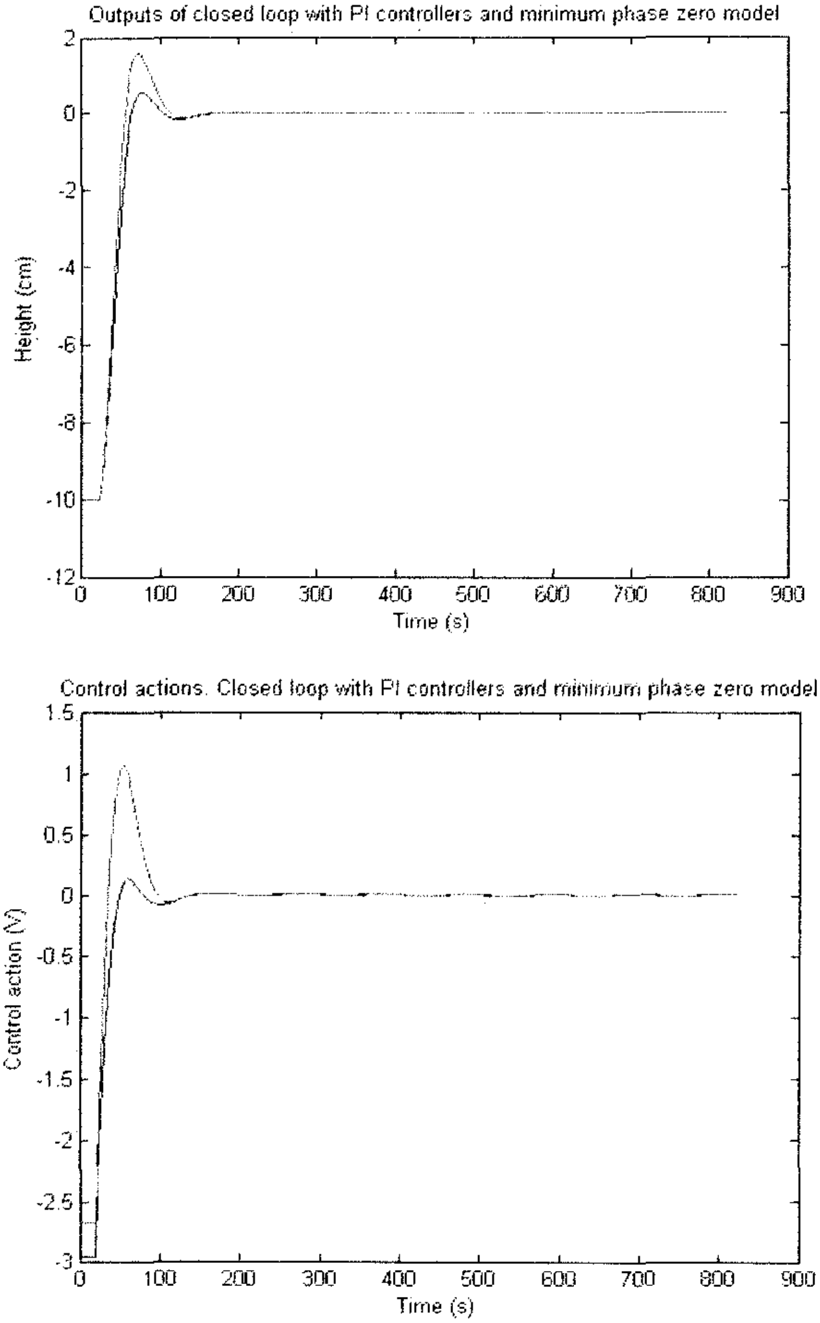

Figure 5.0: Results of minimum phase zero configurations

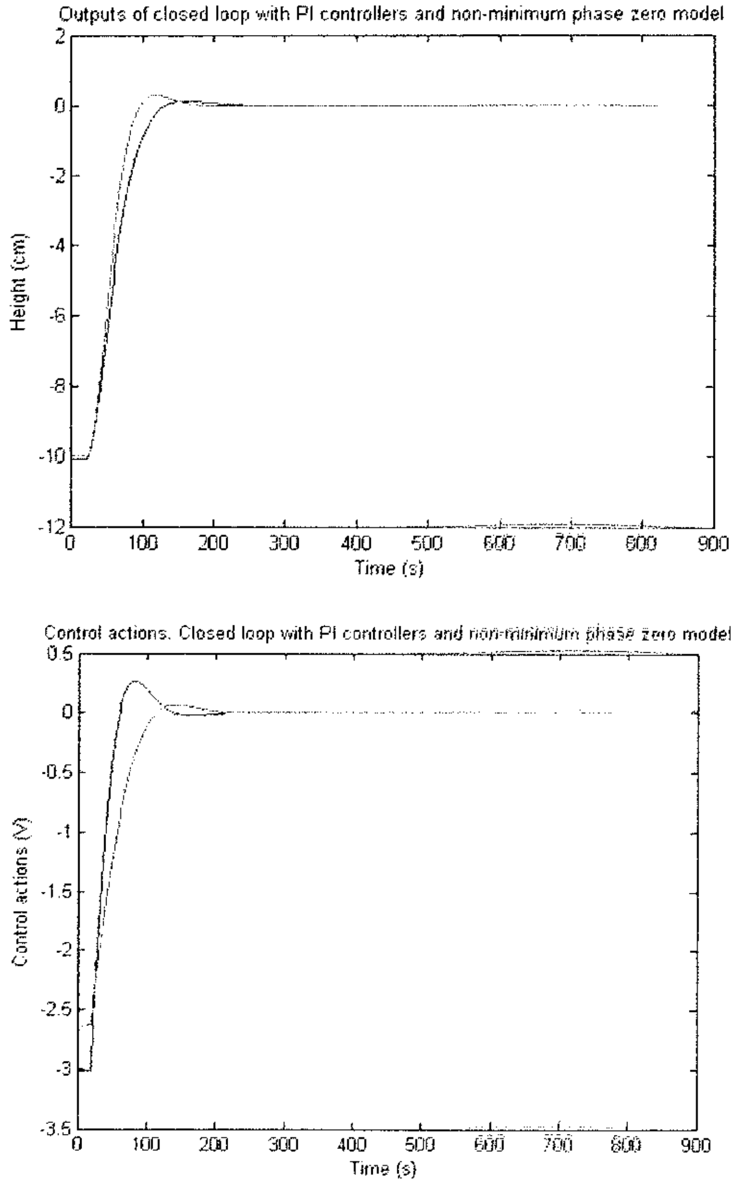

Figure 6.0: Results of non-minimum phase zero configurations

The responses have overshoot because of the influence of upper tanks, but it is not really high, even knowing that the step applied is wide, and applied from a long distance point from the equilibrium one. The response is better in non-minimum phase zero model because of the decoupling applied to obtain a stable system. The applied voltages are not much more than 1V over the linear 0V, no more than 5V adding the linearization values. Knowing that the saturation level is 12.5V, it can be said that there will be no problems with this saturation. All the states are observed during the simulation, and upper tanks don't have problems of overflow, as the lower ones. With these conclusions, and taking into account possible differences between process and model, the following step is to test the controllers in the real process.

4.1.3 Water tanks hole areas and pump constants

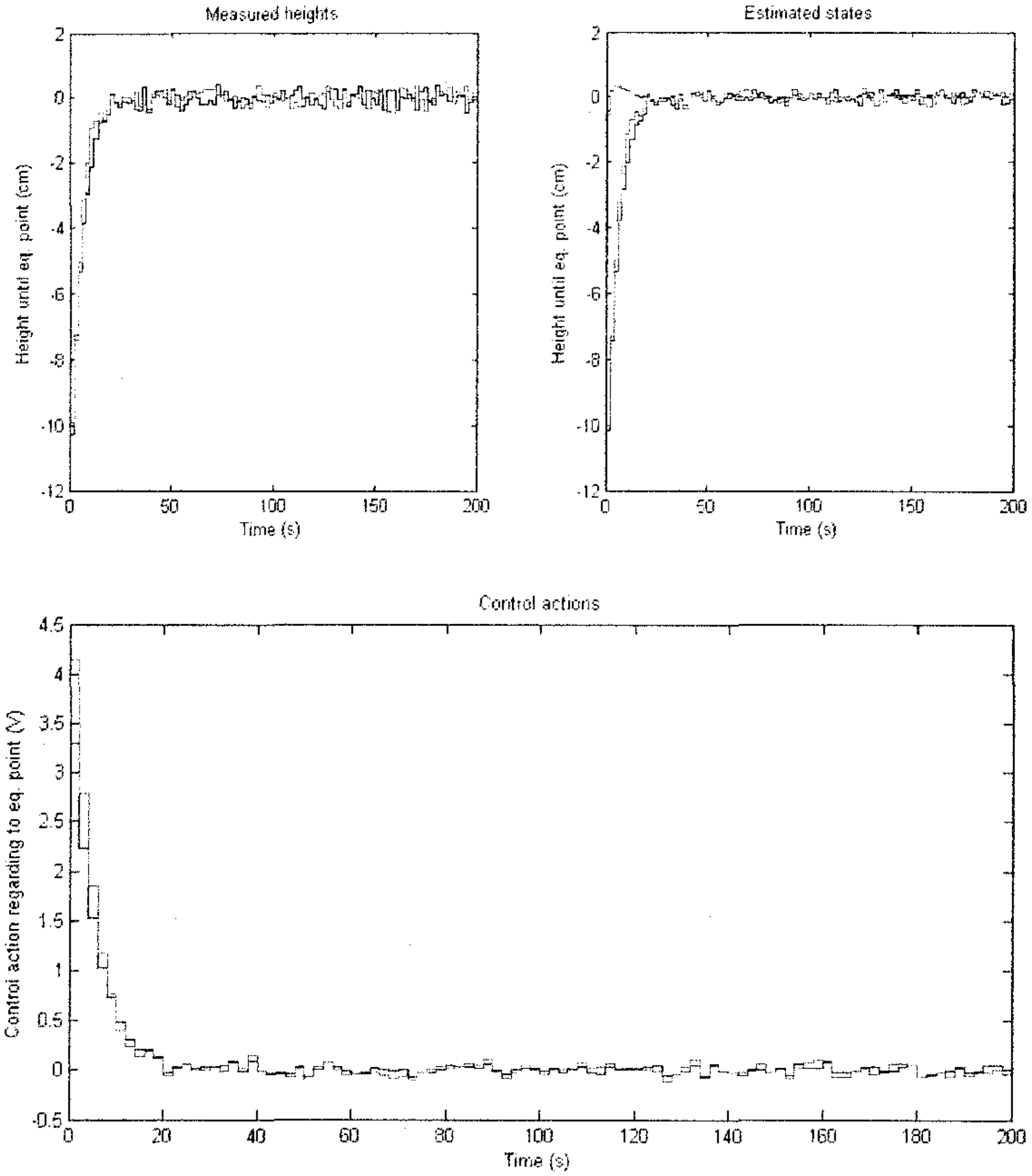

Figure 7.0: Simulation results. Process model with minimum phase zero

The response of states 1 and 2 is better than states 3 and 4, without overshoot and relatively quicker (taking into account the difference of distance to the equilibrium point). The variation of control actions is smooth too, not oscillating around the control actions in equilibrium point. Therefore, the response in simulation is what could be expected according to the design. In addition, almost the same result is obtained with the model process with non-minimum zero.

4.1.4 Deviation

To determinate the deviation, a simple simulation is done. Constant control actions of 10V are introduced. Heights data have to be captured during a relatively long time, to obtain correct mean values and therefore valid deviations. In this case, 235 samples are obtained:

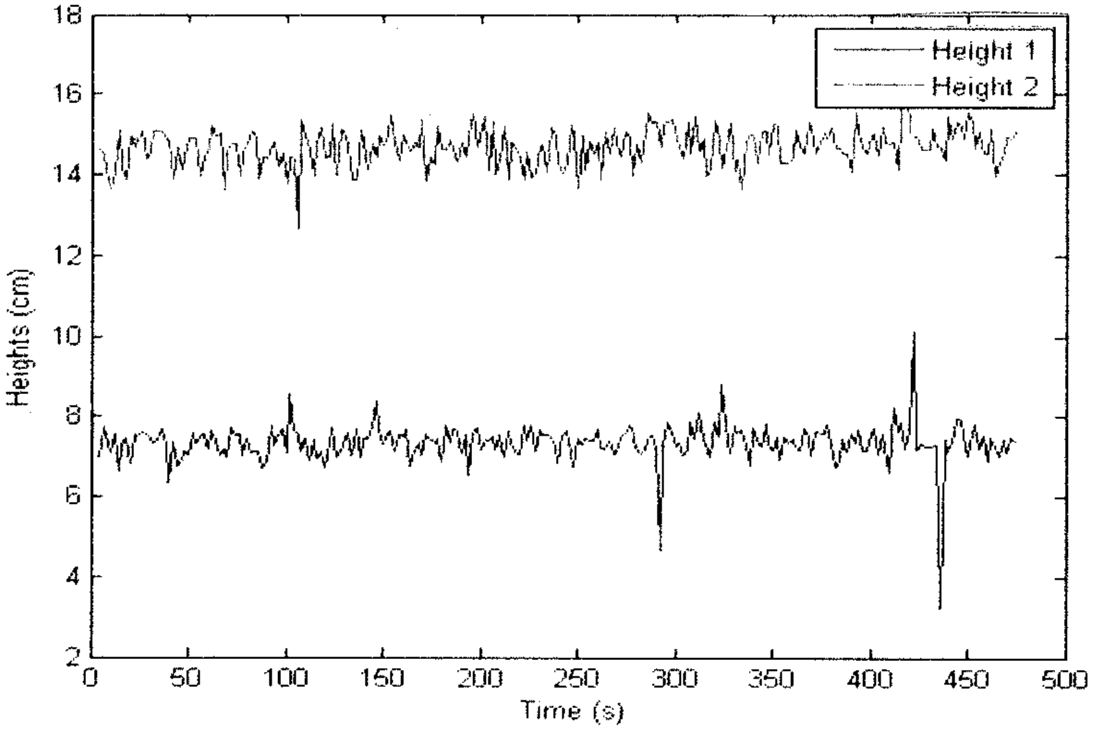

Figure 8.0: Deviation of level Tank 1 and Tank 2 for 470 seconds

4.1.5 Parameters values and final model

Some steps are introduced around the equilibrium point to test the behavior, changing dynamically the linearization point, and therefore changing the model too. Seeing some parts of the simulation, made with minimum zero process:

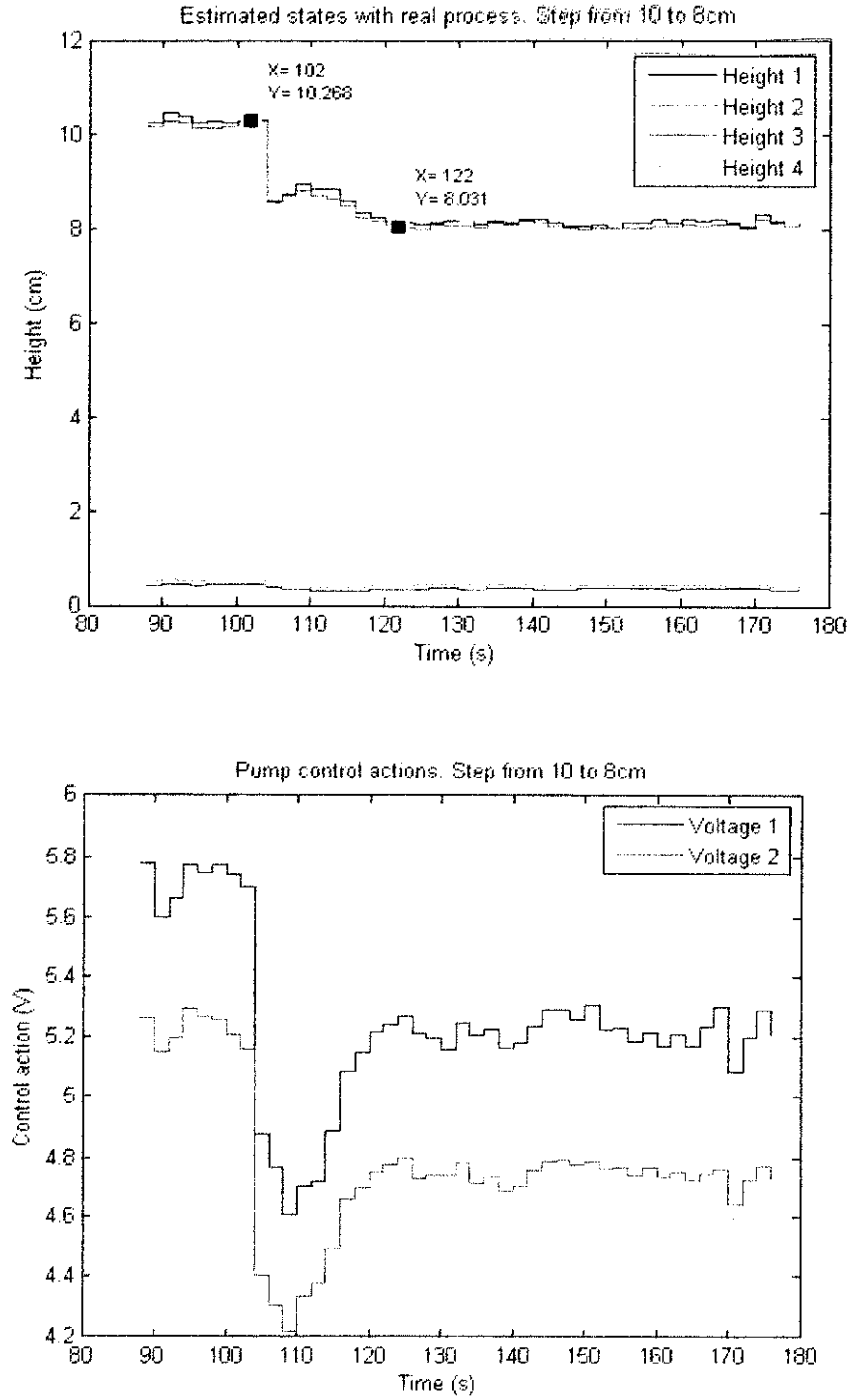

Figure 9.0: Simulation behavior with a step from 10 to 8cm. Minimum zero configuration

As it can be observed, the dynamical response is adequate with design criteria. Changes in the heights are quick (settling time of 20 seconds), with good exactitude for the 2 first states, and a good noise filtering. In addition, first samples after the change of reference have a remarkable peak. This happens because of the change of model.

Regarding to the control actions, they have a bigger overshoot, but just to bring the state of the system as fast as possible to the equilibrium point. This happens because of bigger weightings of heights 1 and 2 than the ones of both voltages. The same result can be seen in another experiment, this time with non-minimum phase zero configuration:

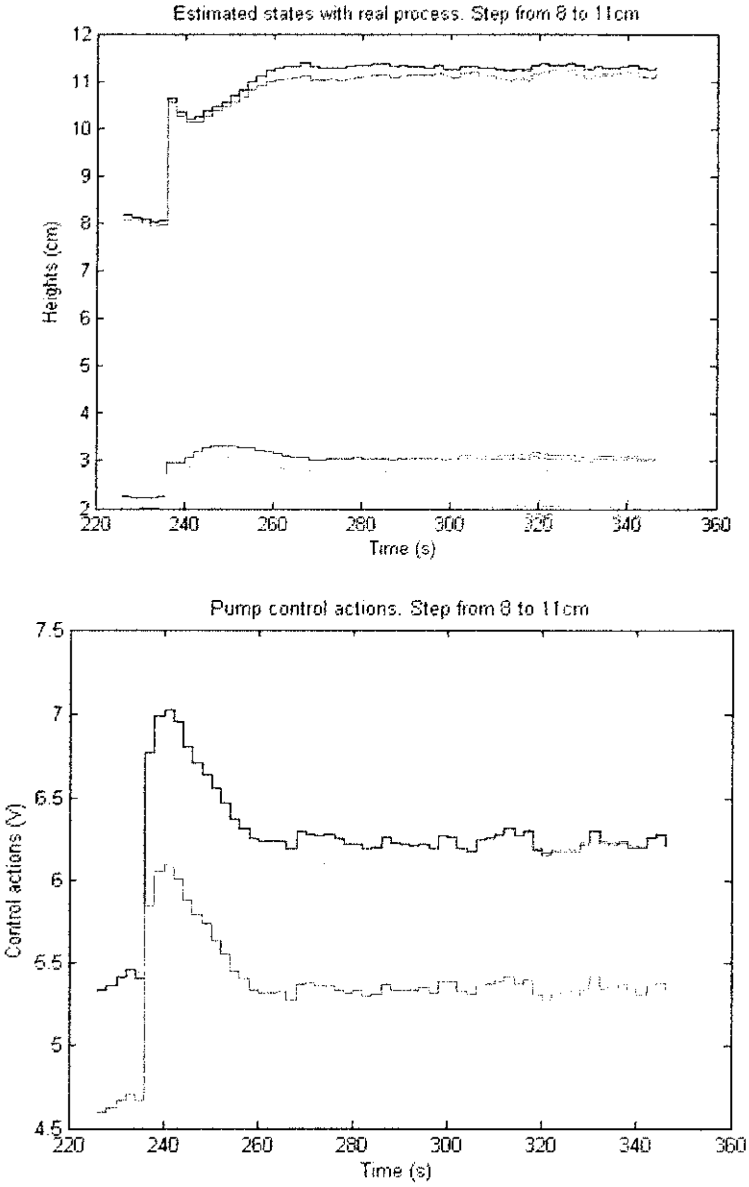

Figure 10.0: Simulation behavior with a step from 10 to 8cm. Non-minimum zero configurations

4.1.6 Validation

The dynamics are accurate, but what can be noted is that there's a delay provoked by the length of the tubes. There are 3 to 4 samples until the effect of an input change is noted in an output.

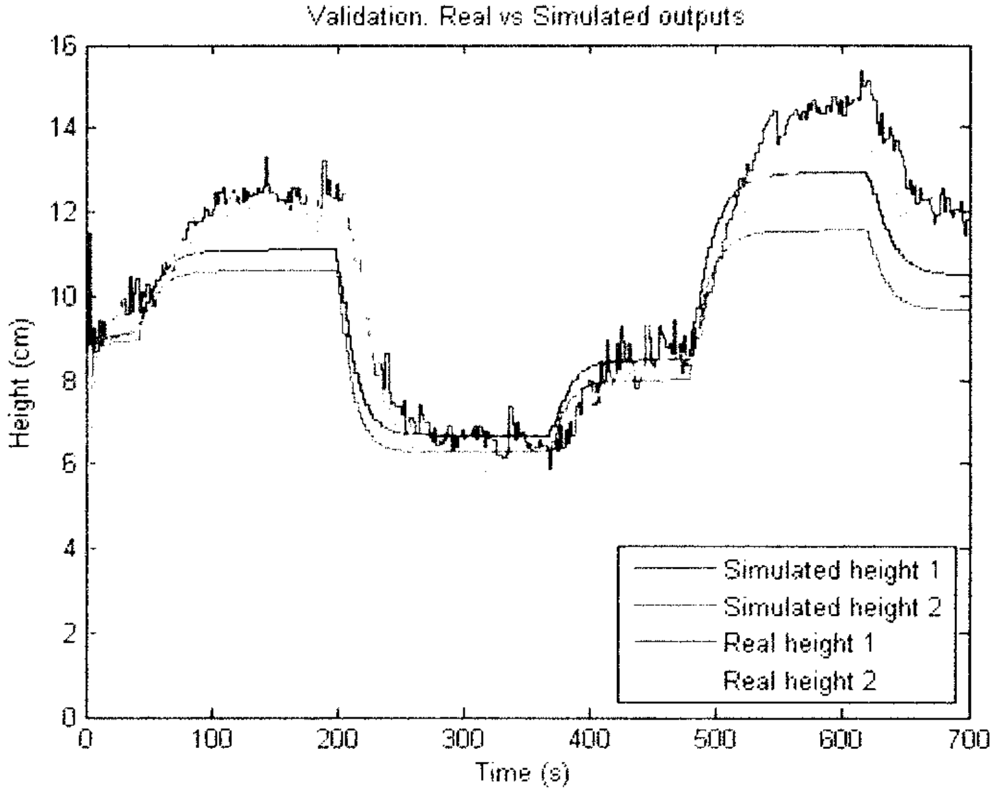

Figure 11.0: Validation

The gain model is correct around water levels of 10cm. With measures of between 12 and 14cm, the gain of the real system is higher. This happens because of the time variable behavior of the pumps and sensors. Some experiments with a separation of days between them have different results. In general, it can be said that the model is valid around 10cm, which will be the point used for most of the simulation. The dynamics are accurate, but what can be noted is that there's a delay provoked by the length of the tubes. There are 3-4 samples until the effect of an input change is noted in an output. The gain simulation is correct around water levels of 10cm. With measures of between 12 and 14cm, the gain of the real system is higher. The different percentage is 14.02% based on the average for every heights.14.02% is lesser then 15%, which is the targeted percentage difference. The data in the simulation is validated.

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

The Quadruple-Tank Process has been presented. It is a simulation process that was designed in order to illustrate various concepts in multivariable control. It has been observed that in each design technique, non-minimum phase is quite difficult to control. Multivariable system with unstable transmission zeros usually come across with internal instability problems. The sign of the steady-state gain should always be considered when designing control systems for multivariable processes. This level of understanding is needed to select the proper design of quadruple tank system and to determine whether a particular control problem can be addressed by better controller tuning, by a different control structure, by changing the process design, or by changing the operating conditions.

Dynamic simulation of multivariable process for quadruple tank system is developed using MATLAB® to study the dynamic simulation of quadruple tank system. The complexity of the dynamics of the system can be represented by the graph and data deviation of the height of each tanks controlled by the voltage of each pumps.

The state of variables which are height and voltage of the pump are validated by the real process of quadruple tank system. The feasibility design of quadruple tank system can be determined by this simulation.

5.2 RECOMMENDATIONS

The project is generally at the stage of process model derivation, input and output generation and validation to get more understanding about the quadruple tank system. The coding of the MATLAB® would be the time consuming but the project should be managed to continue to the next steps till the end by the scheduled project of work. Further research and reference is relevance in order to make sure the results of the dynamic simulation is valid and the objective of this project can be achieved.

REFFERENCES

A. J. Krener and A. Isidori, *Linearisation by output injection and nonlinear observers* , Syst. And Cont. Lett., vol. 3, 1983, pp 47-52.

Andersson, M.; Glad, T.; Norrlöf, M.; Gunnarsson, S.: (2002). A simulation and animation tool for studying multivariable control. 15th World Congress of IFAC, Barcelona 21-26 July.

Åström, K. J.; Lundh, M.: (1992) *Lund control program combines theory with hands-on experience. IEEE Control Systems Magazine* , vol 12, n° 3, pp. 22-30.

Esquembre, E.: (2002). Easy Java Simulations 3.1, http://fem.um.es/Ejs

K. H. Johansson, *The Quadruple-Tank Process: A Multivariable Laboratory Process with an Adjustable Zero* , IEEE Transactions on Control Sytems Technology, 8(3), (2000), 456–465.

Qamar Saeed, Vali Uddin and Reza Katebi, Multivar iable Pr edictive PID Control for Quadruple Tank, Wor ld Academy of Science, Engineering and Technology 67,2010.

R.Suja Mani Malar and T.Thyagarajan, *Modeling of Quadruple Tank System Using Soft Computing Techniques* , European Journal of Scientific Research, 2009.

Rosenbrock, H. H. (1973). The zeros of a system. *International Journal of Control 18* , 297-299.

Salt, J.; Albertos, P.; Dormido, S.; Cuenca, A.: (2003). *An interactive simulation tool for the study of multirate sampled data system* . 15th IFAC Symposium on Advances in Control

APPENDICES

```
function dh = height(t,h)
%Cross-section of the tank (cm2)
A1=28;
A2=32;
A3=28;
A4=32;

%Cross-section area of an outlet (cm2)
a1=0.071;
a2=0.057;
a3=0.071;
a4=0.057;

%Acceleration due ti gravity (cm/s2)
g=981;

%Pump flow constants (cm3/sV)
k1_nmp = 0.5;
k2_nmp = 0.5;

%Ratio of the flows (L/min)
g1_nmp = 0.70;
g2_nmp= 0.30;

%Pump voltage (V)
u1=0.5;
u2=0.5;

dh = zeros(4,1);    % a column vector

dh(1) = -a1/A1*sqrt(2*g*h(1)) + a3/A1*sqrt(2*g*h(3)) +
g1_nmp*k1_nmp/A1*u1;
dh(2) = -a2/A2*sqrt(2*g*h(2)) + a4/A2*sqrt(2*g*h(4)) +
g2_nmp*k2_nmp/A2*u2;
dh(3) = -a3/A3*sqrt(2*g*h(3)) + (1-g2_nmp)*k2_nmp/A3*u2;
dh(4) = -a4/A4*sqrt(2*g*h(4)) + (1-g1_nmp)*k1_nmp/A4*u1;
```

```
[T,H] = ode45(@height,[0 120],[0 0 0 0]);
plot(T,H(:,1),'--',T,H(:,2),'*',T,H(:,3),'
',T,H(:,4),'.');
```

```
package QuadTankPack


  model PRBS1

    Modelica.Blocks.Interfaces.RealOutput y;

    parameter Integer N = 10;

    parameter Real ts[N] = {  0. ,   3.3,   9.3,  15.3,
24.3,  36.3,  39.3,  42.3,  54.3,  57.3};

    parameter Real ys[N] = { 5.,  6.,  5.,  6.,  5.,
6.,  5.,  6.,  5.,  6.};

   equation

    y = noEvent(if time <= ts[2] then ys[1] else

        if time <= ts[3] then ys[2] else

        if time <= ts[4] then ys[3] else

        if time <= ts[5] then ys[4] else

        if time <= ts[6] then ys[5] else

        if time <= ts[7] then ys[6] else

        if time <= ts[8] then ys[7] else

        if time <= ts[9] then ys[8] else

        if time <= ts[10] then ys[9] else ys[10]);

  end PRBS1;


model PRBS2

    Modelica.Blocks.Interfaces.RealOutput y;

    parameter Integer N = 11;

    parameter Real ts[N] = { 0. ,   0.3,   9.3,  21.3,
24.3,  27.3,  39.3,  42.3,  48.3,

        51.3,  57.3};

    parameter Real ys[N] = {5.,  6.,  5.,  6.,  5.,  6.,
5.,  6.,  5.,  6.,  5.};
```

```
 equation
    y = noEvent(if time <= ts[2] then ys[1] else
        if time <= ts[3] then ys[2] else
        if time <= ts[4] then ys[3] else
        if time <= ts[5] then ys[4] else
        if time <= ts[6] then ys[5] else
        if time <= ts[7] then ys[6] else
        if time <= ts[8] then ys[7] else
        if time <= ts[9] then ys[8] else
        if time <= ts[10] then ys[9] else
        if time <= ts[11] then ys[10] else ys[11]);
  end PRBS2;


  model TestPRBS
    PRBS1 prbs1;
    PRBS2 prbs2;
    Real x;
  equation
    der(x) = 1;
  end TestPRBS;


  model Sim_QuadTank
    QuadTank qt;
    input Real u1 = qt.u1;
    input Real u2 = qt.u2;
initial equation
  der(qt.x1) = 0;
  der(qt.x2) = 0;
  qt.x3 = 0.024;
  qt.x4 = 0.023;

  end Sim_QuadTank;



  model QuadTank

    // Process parameters

      parameter Modelica.SIunits.Area A1=4.9e-4,
A2=4.9e-4, A3=4.9e-4, A4=4.9e-4;

      parameter Modelica.SIunits.Area a1=0.03e-4,
a2=0.03e-4, a3=0.03e-4, a4=0.03e-4;

      parameter Modelica.SIunits.Acceleration g=9.81;

      parameter Real k1_nmp(unit="m3^/s/V") = 0.56e-6,
k2_nmp(unit="m^3/s/V") = 0.56e-6;

      parameter Real g1_nmp=0.30, g2_nmp=0.30;



    // Initial tank levels

      parameter Modelica.SIunits.Length x1_0 =
0.04102638;

      parameter Modelica.SIunits.Length x2_0 =
0.06607553;

      parameter Modelica.SIunits.Length x3_0 =
0.00393984;

      parameter Modelica.SIunits.Length x4_0 =
0.00556818;



    // Tank levels

      Modelica.SIunits.Length
x1(start=x1_0,min=0.0001/*,max=0.20*/);

      Modelica.SIunits.Length
x2(start=x2_0,min=0.0001/*,max=0.20*/);

      Modelica.SIunits.Length
x3(start=x3_0,min=0.0001/*,max=0.20*/);

      Modelica.SIunits.Length
x4(start=x4_0,min=0.0001/*,max=0.20*/);
```

```
      // Inputs

      input Modelica.SIunits.Voltage u1;

      input Modelica.SIunits.Voltage u2;


  equation

      der(x1) = -a1/A1*sqrt(2*g*x1) + a3/A1*sqrt(2*g*x3)
+

                              g1_nmp*k1_nmp/A1*u1;

      der(x2) = -a2/A2*sqrt(2*g*x2) + a4/A2*sqrt(2*g*x4)
+

                              g2_nmp*k2_nmp/A2*u2;

      der(x3) = -a3/A3*sqrt(2*g*x3) + (1-
g2_nmp)*k2_nmp/A3*u2;

      der(x4) = -a4/A4*sqrt(2*g*x4) + (1-
g1_nmp)*k1_nmp/A4*u1;


end QuadTank;


model QuadTankInit

  extends QuadTank;

initial equation

  der(x1) = 0;

  der(x2) = 0;

  der(x3) = 0;

  der(x4) = 0;

end QuadTankInit;


optimization QuadTank_Opt (objective = cost(finalTime),

                     startTime = 0,

                     finalTime = 50)
```

```
    extends
QuadTank(u1(initialGuess=u1_r),u2(initialGuess=u2_r),

                     x1(initialGuess=x1_0,fixed=true),

               x2(initialGuess=x2_0,fixed=true),

                     x3(initialGuess=x3_0,fixed=true),

                     x4(initialGuess=x4_0,fixed=true));


// Reference values

    parameter Modelica.SIunits.Length x1_r = 0.06410371;

    parameter Modelica.SIunits.Length x2_r = 0.10324302;

    parameter Modelica.SIunits.Length x3_r = 0.006156;

    parameter Modelica.SIunits.Length x4_r = 0.00870028;

    parameter Modelica.SIunits.Voltage u1_r = 2.5;

    parameter Modelica.SIunits.Voltage u2_r = 2.5;


    Real cost(start=0,fixed=true);


    equation
      der(cost) = 40000*((x1_r - x1))^2 +
                        40000*((x2_r - x2))^2 +
                        40000*((x3_r - x3))^2 +
                        40000*((x4_r - x4))^2 +
                        ((u1_r - u1))^2 +
                        ((u2_r - u2))^2;


end QuadTank_Opt;
```

```
optimization QuadTank_Static(objective=(x1_meas-x1)^2 +
(x2_meas-x2)^2 +

                                        (x3_meas-x3)^2 +
(x4_meas-x4)^2, static=true)


  extends QuadTank(a1(free=true),a2(free=true));


 parameter Real x1_meas = x1_0+0.01;

 parameter Real x2_meas = x2_0-0.01;

 parameter Real x3_meas = x3_0-0.01;

 parameter Real x4_meas = x4_0+0.01;


initial equation

 der(x1) = 0;

 der(x2) = 0;

 der(x3) = 0;

 der(x4) = 0;


end QuadTank_Static;


optimization QuadTank_ParEst (objective=sum((y1_meas[i]
- qt.x1(t_meas[i]))^2 +

                                             (y2_meas[i]
- qt.x2(t_meas[i]))^2 for i in 1:N_meas),


startTime=0,finalTime=60)


    // Initial tank levels

      parameter Modelica.SIunits.Length x1_0 = 0.06255;

      parameter Modelica.SIunits.Length x2_0 = 0.06045;
```

```
       parameter Modelica.SIunits.Length x3_0 = 0.02395;

       parameter Modelica.SIunits.Length x4_0 = 0.02325;


    QuadTank qt(x1(fixed=true),x1_0=x1_0,

                    x2(fixed=true),x2_0=x2_0,

                      x3(fixed=true),x3_0=x3_0,

                      x4(fixed=true),x4_0=x4_0,

                      a1(free=true,initialGuess = 0.03e-
4,nominal=0.03e-4,min=0,max=0.1e-4),

                      a2(free=true,initialGuess = 0.03e-
4,nominal=0.03e-4,min=0,max=0.1e-4));


    parameter Integer N_meas = 61;

    parameter Real t_meas[N_meas] = 0:60.0/(N_meas-
1):60;

    parameter Real y1_meas[N_meas] = ones(N_meas);

    parameter Real y2_meas[N_meas] = ones(N_meas);

    PRBS1 prbs1;

    PRBS2 prbs2;

equation

    connect(prbs1.y,qt.u1);

    connect(prbs2.y,qt.u2);

end QuadTank_ParEst;


optimization QuadTank_ParEst2 (objective=sum((y1_meas[i]
- qt.x1(t_meas[i]))^2 +

                                              (y2_meas[i]
- qt.x2(t_meas[i]))^2 +

                                               (y3_meas[i]
- qt.x3(t_meas[i]))^2 +

                                               (y4_meas[i]
- qt.x4(t_meas[i]))^2 for i in 1:N_meas),
startTime=0,finalTime=60)


    // Initial tank levels

      parameter Modelica.SIunits.Length x1_0 = 0.06255;

      parameter Modelica.SIunits.Length x2_0 = 0.06045;

      parameter Modelica.SIunits.Length x3_0 = 0.02395;

      parameter Modelica.SIunits.Length x4_0 = 0.02325;


   QuadTank qt(x1(fixed=true),x1_0=x1_0,

                    x2(fixed=true),x2_0=x2_0,

                      x3(fixed=true),x3_0=x3_0,

                      x4(fixed=true),x4_0=x4_0,

                      a1(free=true,initialGuess = 0.03e-
4,nominal=0.03e-4,min=0,max=0.1e-4),

                      a2(free=true,initialGuess = 0.03e-
4,nominal=0.03e-4,min=0,max=0.1e-4),

                      a3(free=true,initialGuess = 0.03e-
4,nominal=0.03e-4,min=0,max=0.1e-4),

                      a4(free=true,initialGuess = 0.03e-
4,nominal=0.03e-4,min=0,max=0.1e-4));


    parameter Integer N_meas = 61;

    parameter Real t_meas[N_meas] = 0:60.0/(N_meas-
1):60;

    parameter Real y1_meas[N_meas] = ones(N_meas);

    parameter Real y2_meas[N_meas] = ones(N_meas);

    parameter Real y3_meas[N_meas] = ones(N_meas);

    parameter Real y4_meas[N_meas] = ones(N_meas);

    PRBS1 prbs1;

    PRBS2 prbs2;

equation
```

```
    connect(prbs1.y,qt.u1);

    connect(prbs2.y,qt.u2);

end QuadTank_ParEst2;



optimization QuadTank_Sens


    // Initial tank levels

      parameter Modelica.SIunits.Length x1_0 = 0.06255;

      parameter Modelica.SIunits.Length x2_0 = 0.06045;

      parameter Modelica.SIunits.Length x3_0 = 0.02395;

      parameter Modelica.SIunits.Length x4_0 = 0.02325;


   QuadTank qt(x1(fixed=true),x1_0=x1_0,

                   x2(fixed=true),x2_0=x2_0,

                     x3(fixed=true),x3_0=x3_0,

                     x4(fixed=true),x4_0=x4_0,

                     a1(free=true,initialGuess = 0.03e-
4,nominal=0.03e-4,min=0,max=0.1e-4),

                     a2(free=true,initialGuess = 0.03e-
4,nominal=0.03e-4,min=0,max=0.1e-4));


    PRBS1 prbs1;

    PRBS2 prbs2;

equation

    connect(prbs1.y,qt.u1);

    connect(prbs2.y,qt.u2);

end QuadTank_Sens;
```