CONTENTS

ACKNOWLEDGMENTS

PREFACE

CHAPTER 1 MATHEMATICAL MODEL FOR THE CONTINUOUS FLOW BOILING SYSTEM *1*

1.1 MATHEMATICAL MODEL *1*

1.1.1 MODEL DESCRIPTION *1*

1.1.2 THE PHENOMENON OF BOILING SYSTEM *3*

1.1.3 PROCEDUIRE FOR ASSEMBLING THE NON-LINEAR MODEL *4*

1.1.4 BULDING OF THE NON-LINEAR MODEL *5*

1.2 LINERAZATION OF THE NON-LINEAR MODEL *8*

1.2.1 CALCULATION OF THE OPERATING POINT *8*

1.2.2 LINEARIZATION USING TAYLOR SERIES *9*

1.2.3 LINEARZATION OF THE CONTINUOUS FLOW BOILING SYSTEM *9*

1.3 SIMULATION OF THE LINEAR MODEL *11*

CHAPTER 2 CONTROL OF CONTINUOUS FLOW BOILING SYSTEM IN SINGLE INPUT SINGLE OUTPUT (SISO) APPROACH *14*

2.1 BASIC CONCEPTIONS OF CONTROL *14*

2.2 SISO SYSTEM APPROACH (PLANT, MASURMENT UNIT AND ACTUATOR) *17*

2.3 PI-CONTROLLER DESIGN USING KESSLER METHOD *19*

2.4 PI-CONTROLLER DESIGN USING CRITICAL GAIN METHOD *22*

2.5 PI-CONTROLLER DESIGN IN FREQUENCY DOMAIN *24*

2.6 OVERVIEW FOR SISO SYSTEM *26*

CHAPTER 3 CONTROL OF CONTINUOUS FLOW BOILING SYSTEM IN MULTI INPUT MULTI OUTPUT (MIMO) APPROACH *29*

3.1 MIMO SYSTEM APPROACH *29*

3.2 SYSTEM DECOUPLING AND CONTROLLER DESIGN USING THE P-CANONICAL FORM *30*

3.3 NON-LINEAR DECOUPLING ALGORITHM OF THE MIMO SYSTEM *35*

3.3.1 THEORETICAL DESCRIPTION OF THE METHOD 35

3.3.2 APPLYING THE NON-LINEAR DECOUPLING METHOD FOR THE MIMO BOILING SYSTEM *39*

3.3.3 PROVING THE CONCLUSION, USING THE RESULTS FOR THREE TANK SYSTEM (3TS) *42*

CHAPTER 4 PRACTICAL REALIZATION OF THE BOILING SYSTEM USING THE DELTA V SOFTWARE *46*

4.1 GENERAL PRESENTATION OF THE DELTA V SOFTWARE *46*

4.1.1 THE DELTA V LIBRARY *47*

4.1.2 SYSTEM CONFIGURATION *48*

4.1.3 CONTROL STRATEGIES *49*

4.1.4 PHYSICAL NETWORK *50*

4.1.5 CONTROL STUDIO APPLICATION *52*

4.1.6 RECEIPT STUDIO *54*

4.1.7 DELTA V USER MANAGER *54*

4.1.8 I/O CONFIGURATION APPLICATION *55*

4.1.9 DEVELOPING THE CONTROL STRATEGY *55*

4.1.10 SYSTEM ALARM MANAGEMENT APPLICATION 57

4.1.11 DELTA V OPERATE 57

4.1.12 DELTA V BATCH HISTORY VIEWS *59*

4.1.13 DIAGNOSTICS *59*

4.2 LABORATORY EQUIPMENT OF THE BOILING SYSTEM (NYOMAS (PRESSURE) APPLICATION) *60*

4.2.1 THE SYSTEM HARDWARE *60*

4.2.2 THE SYSTEM SOFTWARE *63*

4.3 OPTIMIZATION OF THE PID-CONTROLLER FOR THE REAL BOILING SYSTEM *65*

4.3.1 IDENTIFICATION OF THE REAL BOILING SYSTEM *65*

4.3.2 THR SIMULATION MODEL OF THE REAL BOILING SYSTEM *66*

4.3.3 REDESIGN THE PID CONTROLLER USING THE KESSLER METHOD *67*

4.3.4 IMPLEMENTING THE OPTIMIZED CONTROLLER FOR THE REAL BOILING SYSTEM *68*

CHAPTER 5 CONCLUSION *71*

REFERENCES 73

APPENDIXES 74

A1. Mathematics Program for Identification of Continuous Flow Boiling System. 75

A2. The MATLAB Program for the Output Response Curves Y_1, Y2 when Considering only one input u1 or u2 or u3. 93

A3. MATLAB Program to Compute the Transfer Function Matrix. 94

A.4 SIMULINK Models for the Linear Plant. 95

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to all how contributed their time and talent for the completion of this work, in particular to:

Professor Dr. habil. Zoltan Benyó, full professor at the Budapest University of Technology and Economics, Faculty of Electrical Engineering, Department of Control Engineering and Information Technology for his advice, patience, encouragement, support.

My scientific supervisor*, Dipl. Eng. Levente Kovács*, Ph D student at the Budapest University of Technology and Economics, Faculty of Electrical Engineering, Department of Control Engineering and Information Technology, Biomedical Engineering Laboratory for his unlimited guidance, encouragement, and help as well as his patient work in reviewing and helping prepare the final draft. His excellent activities in the field of process control, as well as his personal qualities were always exemplary for me. The time I spent under his supervision was central part of my education.

Dipl. Eng. Gergely Fördős, Ph D student at the Budapest University of Technology and Economics, Faculty of Electrical Engineering, Department of Control Engineering and Information Technology, Biomedical Engineering Laboratory for his help and guidance in the practical work with Delta V Software.

Dr. habil. Paláncz Béla, associate professor at the Budapest University of Technology and Economics, Department of Photogrammetry, Laboratory of Informatics, for his help and stimulating consultations.

My college *Eng. Rammah Mohamed Abohtyra*, graduating student at the Budapest University of Technology and Economics, Faculty of Electrical Engineering, Department of Control Engineering and Information Technology for sharing me the practical work and his activities.

Professor Dr. József Harangozó, the director of the MSc. program at the Faculty of Electrical Engineering, Budapest University of Technology and Economics for his advices and help during our studies.

Professor Dr. habil Peter Arató, the director of the Department of Control Engineering and Information Technology at the Budapest University of Technology and Economics and all his staff members for the scientific help as well as the pleasant research atmosphere.

Libyan society for making the completion of this work possible through their financial support.

PREFACE

The aim of the thesis is to study the model of a continuous flow boiling system (SISO system and MIMO system approach).
Then, the practical work refers on the control of the boiling system using the Delta V industrial control software.

The Thesis is organized into five chapters, a reference and an appendix parts. The outline of the thesis is as follows:

Chapter 1, describes the procedure for developing the non-linear mathematical model for a continuous flow boiling system employing the mass and heat balance equations and the linearization of the identified model around the operating point using a Taylor approximation. The results are presented by simulation of the linear model using MATLAB SIMULINK.

Chapter 2 discusses the control of continuous flow boiling system in single input single output (SISO system) approach. In this chapter the design of the PI controller using the Kessler, critical gain and frequency domain methods are made, then I compare the three methods from the point of quality parameters (settling time, overshoot).

In *Chapter 3* control of continuous flow boiling system in multi input multi output (MIMO system) approach is presented. This chapter deals with system decoupling, controller design using the P-canonical form and a non-linear decoupling algorithm. Finally the evaluation of the results is made using also the decoupling results of a general three thank system (3TS).

Chapter 4 describes the practical work, which refers on the control of a boiling system using the Delta V industrial control software. The boiling system is a real laboratory model, which contains already an implemented PID controller. The aim of my work was to replace the existing controller by an optimal one. Therefore, first I had to familiarize myself with the mentioned industrial software, then the identification of the plant was carried out and the design of the optimal controller was made only in the last step

Chapter 5 presents the results of my work, but also gives future plans.

CHAPTER 1

MATHEMATICAL MODEL OF THE CONTINUOUS FLOW BOILING SYSTEM

This chapter describes the procedure for developing the non-linear mathematical model for a continuous flow boiling system employing the mass and heat balance equations and the linearization of the non-linear model around the operating point using a Taylor approximation, and the simulation of the linear model.

1.1 MATHEMATICAL MODEL

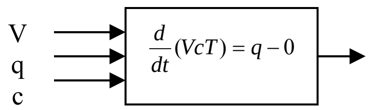
1.1.1 MODEL DESCRIPTION [3]

Suppose that a container of fluid is heated at a rate q (PCU/time). Then, the heat balance equation would state:

Rate of change of heat content = heat in - heat out

$$\frac{d}{dt}(VcT) = q - 0 \text{ (no heat loss)} \quad , \tag{1-1}$$

where: V represents the volume, and c the specific heat.
Since V, q, and c are known, this equation can be used to establish the temperature (Fig.1-1) by supplying V, q, and c.



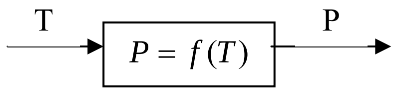


Fig.1-1. Heat Balance Model. **Fig.1-2.** Vapor Pressure /Temperature Relationship.

The vapor pressure exerted by the liquid varies with the temperature, as shown in Fig.1-2. Vaporization can be considered negligible until the temperature has reached the boiling point. At this temperature the vapor pressure P tends to exceed the actual pressure π, that is, P $> \pi$. This results in the stream of vapor issue from the boiling liquid. Because there is no resistance to departure of this vapor, a sufficient flow is emitted that (through the heat balance) automatically prevents the temperature from rising beyond the boiling point.

The vapor pressure corresponding to the boiling temperature is infinitesimally grater than the total pressure, but this minute difference is sufficient to provide the vapor flow that maintains the status quo.

The boiling heat balance is presented in Fig.1-3. The equilibrium equation that computes the vapor rate, v contains a gain factor G that is large enough to keep the (P - π) difference very small.

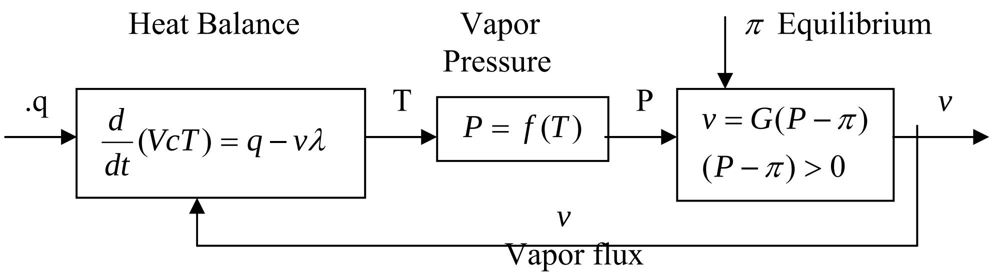


Fig.1-3. Model for Equilibrium Balance.

This model is the "natural" definition of the system. It can be regrouped as shown in Fig.1-4. The overall diagram presented in Fig.1-5 shows that the system has two inputs, π, and q, and two outputs, T, and v.

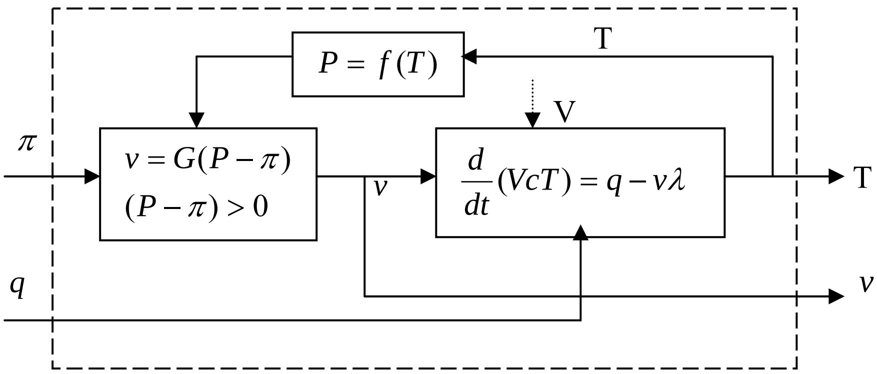


Fig.1-4. Input-Output Relationship for Microscopic Boiling Model.

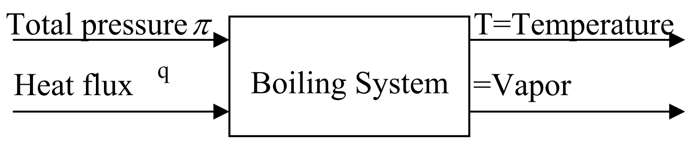


Fig.1-5. Microscopic Input-Output Relationship.

1.1.2 THE PHENOMENON OF BOILING SYSTEM [3]

The phenomenon of boiling system is such that the temperature responds only to the total pressure π and the vapor flow only to the heat flux q. This leads to the more convenient mode shown in Fig.1-6. The heat balance is used to establish the vapor flux, whereas the system pressure π indicates the temperature. In most cases the differential term $d/_{dt}(VcT)$ is very small in comperation with q and can be neglected.

In summary, the only way to change the temperature in case of boiling a single component liquid, is to change the total pressure. Changing the heating rate, this changes only the rate of evolution of vapor. The cause-and-effect relationships for single boiling fluid can be status as:

- Pressure (P) establishes the boiling temperature (T);
- Heat flux (q) establishes the vapor rate (v).

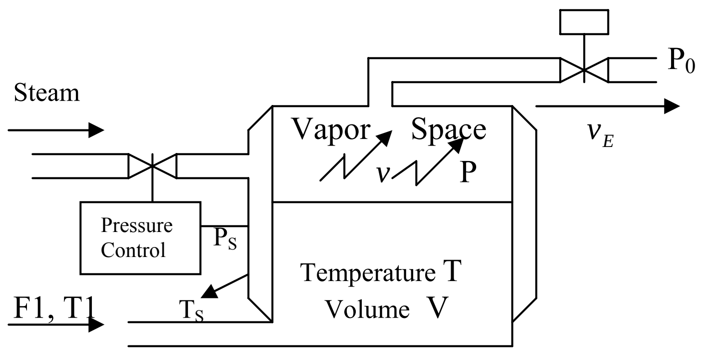


Fig.1-6. Continuous Flow Boiling system.

The complexity of boiling will be assumed for the jacket vessel. Fig.1-6.shows the feed flow supplied in liquid form and the exit flow withdrawn as vapor. The mathematical model for this boiler consists of simultaneous mass and energy balance. The mass balance equation of the liquid is:

$$\frac{dV}{dt} = F1 - v \quad , \tag{1-2}$$

where F1 is the feed rate and v is the boil up rate. The mass balance equation on the vapor is:

$$\frac{d}{dt}(m_G) = v - v_E \quad , \tag{1-3}$$

where v_E is the flow of vapor through exit valve.

Because equilibrium is assumed to exist at all times between liquid and vapor, an energy balance for the vapor is not necessary, and the vapor temperature is assumed to be the same as the liquid temperature. The energy balance in the liquid is:

$$\frac{d}{dt}(VcT) = F_1 cT_1 + q - v(cT + \lambda) \quad , \tag{1-4}$$

where $(cT + \lambda)$ is an approximation of the vapor enthalpy. The pressure in the vapor space is obtained from the gas-law relationship:

$$PV_G = mRT \qquad \text{and} \qquad V_G = V_0 - \frac{V}{\phi} \quad , \tag{1-5}$$

where ϕ represents the density (mass/unit Vol) and V_0 the total volume of vessel. The temperature is obtained from the pressure/boiling temperature relationship:

$$T = f(p) = \frac{c_2}{(\ln p - c_1)} \quad . \tag{1-6}$$

If the effluent valve is fixed, the flow rate of vapor through the valve v_E will be:

$$v_E = k\sqrt{P(P - P_0)} \quad . \tag{1-7}$$

The heat flux q is:

$$q = UA(T_S - T) \quad , \tag{1-8}$$

where T_S is the steam temperature.

1.1.3 PROCEDURE FOR ASSEMBLING THE NON-LINEAR MODEL

With the equations for each part of the system defined, the procedure for assembling the model will now be as follows:

A. Boundary values:

1. Inlet flow: F_1;
2. Inlet temperature: T_1;
3. Jacket steam pressure: P_S;
4. Exit pressure: P_0;

B. Equations:

1. Valve: $v_E = K\sqrt{P(P - P_0)} \rightarrow v_E$ (1-9)
2. Gas law: $PV = m_G RT \rightarrow P$ (1-10)

3. Vapor mass balance: $\dfrac{dm_G}{dt} = v - v_E \rightarrow m_G$ (1-11)

4. Boiling point: $T = f(P) \rightarrow T$ (1-12)

5. Jacket heat: $q = UA(T_S - T) \rightarrow q$ (1-13)

6. Heat balance:

$$\frac{d}{dt}(VcT) = F_1 cT_1 + q - (cT + \lambda)v \rightarrow v \qquad (1\text{-}14)$$

7. Mass balance on liquid: $\dfrac{dV}{dt} = F_1 - v \rightarrow V$ (1-15)

8. Gas volume: $V_G = V_0 - \dfrac{V}{\phi} \rightarrow V_G$ (1-16)

The equations described above are assembled in Fig.1-7.

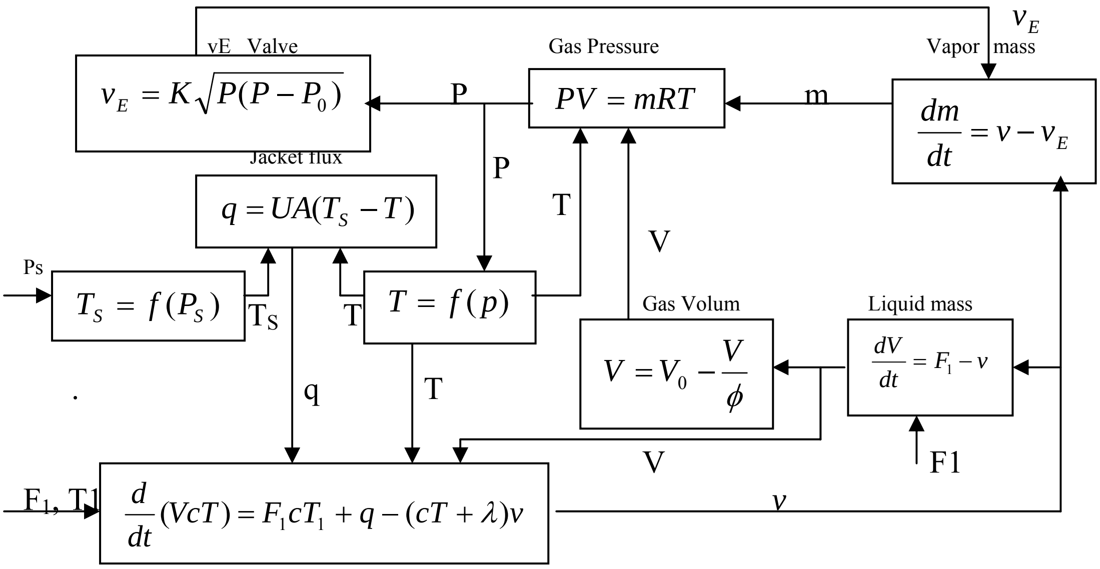


Fig.1-7. Model for continuous flow Boiling Jacketed Vessel.

1.1.4 BUILDING THE NON-LINEAR MODEL [3]

With the equations for each part of the system defined in *Section 1.1.3*, the system equations can be determined, as in the *Mathematica* software (A.1):

$$eq1 = P = \frac{m_G R(T + 273)}{V_G} \quad \text{, (Pressure from gas law)} \qquad (1\text{-}17)$$

$$eq2 = T = \frac{c_2}{\ln P - c_1} - 273 \quad , \tag{1-18}$$

(Boiling point temperature employing Antoine equation)

$$eq3 = q = UA(T_S - T) \quad , \text{(Jacket heat)} \tag{1-19}$$

$$eq4 = v = \frac{q}{T - T_1 + \lambda} \quad , \text{(Boiling rate)} \tag{1-20}$$

$$eq5 = v_E = k\sqrt{P(P - P_0)} \quad , \text{(Out vapor flow rate)} \tag{1-21}$$

$$deq1 = \frac{dm_G}{dt} = v - v_E \quad , \text{(Vapor mass balance)} \tag{1-22}$$

Hence, the mathematical model could be built up considering the following variables:

- Input variables:
  - Inlet temperature: $u_1=T_1$;
  - Jacket temperature: $u_2=T_{S;}$
  - Exit pressure: $u_3= P_0$;
- State Variables:
  - Vapor pressure: $X_1=P$;
  - Boiling temperature: $X_2=T$;
  - Vapor mass: $X_3= m_G$;
- Output variables:
  - Temperature: $Y_1 = X_2 = T$;
  - Outlet vapor flow rate: $Y_2 = v_E$;
- Constants:

R=1.98 moles/ ft^3,c1=13.96 $KJoule/(Kmol.^\circ C)$,c2=-5210.6 $KJoule/(Kmol.^\circ C)$,V_G=30000 ft^3 , λ=9717 PCU/mole, UA=1700, K=5.7 $ft^3/(Kmol.\sec.)$

While the general form of a state space description of a mathematical model is:

$$\begin{aligned} \dot{\overline{X}} &= \overline{\overline{A}}\,\overline{X} + \overline{\overline{B}}\,\overline{u} \\ \overline{Y} &= \overline{C}^T \overline{X} + \overline{D}^T \overline{u} \end{aligned} \quad , \tag{1-23}$$

where: $\overline{\overline{A}}, \overline{\overline{B}}, \overline{C}^T, \overline{D}^T$: are matrixes ($\overline{\overline{A}}$ is the state matrix, $\overline{\overline{B}}$ is the control matrix, $\overline{C}^T$ is output matrix, $\overline{D}^T$ is the direct transformation matrix), the boiling system can be written in this form:

$$\overline{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} P \\ T \\ m_G \end{bmatrix} \quad , \ \overline{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_S \\ P_0 \end{bmatrix} \text{ and } \overline{Y} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} T \\ v_E \end{bmatrix} \ . \tag{1-24}$$

To obtain the corresponding system equations for the state space description of the non-linear model, I started form (1-17)-(1-22). Taking the derivative of the equations (1-17), (1-18), (1-22):

$$\dot{P} = \frac{R}{V}(m_G.\dot{T} + (T+273).\dot{m}_G) \quad , \tag{1-25}$$

$$\dot{T} = \frac{-c_2}{P(\ln p - c_1)^2}.\dot{P} \quad , \tag{1-26}$$

$$\dot{m}_G = v - v_E \quad , \tag{1-27}$$

and substituting the equations (1-20), (1-21) on the last equation:

$$\dot{m}_G = \frac{q}{T - T_1 + \lambda} - k\sqrt{P(P-P_0)} \Rightarrow \dot{m}_G \quad , \tag{1-28}$$

The following result is obtained (substituting the equation (1-26) on the equation (1-25)):

$$\dot{P}(1 + \frac{R.m_G}{V}.\frac{-c_2}{P(\ln P - c_1)^2}) = \frac{R(T+273)}{V}.\dot{m}_G \Rightarrow \dot{P} \ . \tag{1-29}$$

Since:

$$\dot{T} = \frac{-c_2}{P(\ln p - c_1)^2}.\dot{P} \Rightarrow \dot{T} \quad , \tag{1-30}$$

with the equations (1-28), (1-29), (1-30) substituting the input, output and state variables as:

$u_1 = T_1 \qquad X_1 = P \qquad ,$

$u_2 = T_S \qquad X_2 = T \qquad ;$

$u_3 = P_0 \qquad X_3 = m_G \qquad ;$

$$Y_1 = X_2 = T \ ;$$

$$Y_2 = v_E \qquad , \tag{1-31}$$

the state space description of the non-linear model will be [1]:

$$\dot{X}_1 = \left(\frac{R(X_2 + 273).X_1.(\ln X_1 - c_1)^2}{V.X_1.(\ln X_1 - c1)^2 + c_2 R X_3} \right) \left(\frac{UA(u_2 - X_2)}{X_2 - u_1 + \lambda} - K\sqrt{X_1(X_1 - u_3)} \right)$$

$$\dot{X}_2 = \left(\frac{-c_2 R(X_2 + 273)}{V.X_1.(\ln X_1 - c1)^2 + c_2 R X_3} \right) \left(\frac{UA(u_2 - X_2)}{X_2 - u_1 + \lambda} - K\sqrt{X_1(X_1 - u_3)} \right)$$

$$\dot{X}_3 = \frac{UA(u_2 - X_2)}{X_2 - u_1 + \lambda} - K\sqrt{X_1(X_1 - u_3)}$$

$$Y_1 = X_2$$

$$Y_2 = K\sqrt{X_1(X_1 - u_3)} \quad . \tag{1-32}$$

1.2 LINERAZATION OF THE NON-LINEAR MODEL

1.2.1 CALCULATION OF THE OPERATING POINT

Considering the importance of the operating point [3], the following conditions will be applied to the system:

- The liquid level is maintained at a fixed position by a level controller. This makes V and V_G constants and also the feed flow $F_1 = v_E$;
- The liquid is initially cold and heated up to its boiling point. After boiling starts, the pressure rises to its equilibrium level, raising the temperature to a higher value.

At equilibrium point (operating point) the vapor mass will be constant, therefore:

$$\frac{d}{dt}(m_G) = v - v_E = 0 \quad , \tag{1-33}$$

under this condition I can solve the system equations (1-17)-(1-22) and compute the steady state values for the input, state variables. I will denote the steady state values of these variables by using character (o):

$$X_{1o} \equiv Steady.State.value.of.X_1$$
$$u_{1o} \equiv Steady.State.value.of.u_1$$

As a result, the following numerical results are obtained (A.1):

$u_{1o}=T_1=15$ ºC $\quad X_{1o}=P=1.68301$ atm

$u_{2o}=T_S=150$ ºC $\quad X_{2o}=T=114.71$ ºC (1-34)

$u_{3o}=P_0=1$ atm $\quad X_{3o}=m_G=65.7711$ (unit of mass).

1.2.2 LINEARIZATION USING TAYLOR SERIES [10]

Mathematically, a linear differential equation is one for which the following two properties hold:

1. If x (t) is a solution, then c·x (t) is also a solution, where c is a constant;

2. If x_1 is a solution and x_2 is also a solution, then $x_1+ x_2$ is a solution.

As a result, linearization is realized taking the nonlinear functions, expanding them in Taylor series around the steady state operating point level, and neglecting all terms after the first partial derivatives.

Let us assume I have a nonlinear function $f(x_1, x_2, t)$ of a process, where in our case x_1 is the pressure, x_2 is the temperature.

Remark: I will denote the steady state values with (o) symbol:

$$X_{1o} \equiv Steady.State.value.of.X_1$$
$$X_{2o} \equiv Steady.State.value.of.X_2$$

Now I expand the function $f(x_1, x_2, t)$ around its steady state value:

$$f(x_1,x_2,t)=f(x_{1o},x_{2o})+\left(\frac{\partial f}{\partial x_1}\right)\Bigg|_{(x1o,x2o)}(x_1-x_{1o})+\left(\frac{\partial f}{\partial x_2}\right)\Bigg|_{(x1o,x2o)}(x_2-x_{2o})$$

$$+\left(\frac{\partial^2 f}{\partial x_1{}^2}\right)\Bigg|_{(x1o.x2o)}\frac{(x_1-x_{1o})^2}{2!}+\dots \quad . \tag{1-35}$$

The linearization was carried out in our case by truncating the series after the first partial derivatives.

$$f(x_1,x_2,t)=f(x_{1o},x_{2o})+\left(\frac{\partial f}{\partial x_1}\right)\Bigg|_{(x1o,x2o)}(x_{1-}x_{1o})+\left(\frac{\partial f}{\partial x_2}\right)\Bigg|_{(x1o,x2o)}(x_{2-}x_{2o}). \tag{1-36}$$

1.2.3 LINEARIZATION OF THE CONTINUOUS FLOW BOILING SYSTEM

Using the above described procedure, I took the non-linear functions of the boiling system, (1-32), and expanding them in Taylor series expansions, around the steady state operating point, (1-34), the linearized mthematical model can be obtained:

$$\begin{aligned}
\dot{X}_1 &= f_1(X_1, X_2, X_3, u_1, u_2, u_3) \\
\dot{X}_2 &= f_2(X_1, X_2, X_3, u_1, u_2, u_3) \\
\dot{X}_3 &= f_3(X_1, X_2, X_3, u_1, u_2, u_3) \\
Y_1 &= g_1(X_1, X_2, X_3, u_1, u_2, u_3) \\
Y_2 &= g_2(X_1, X_2, X_3, u_1, u_2, u_3)
\end{aligned} \qquad (1\text{-}37)$$

The steady state operating point is:

$$\begin{aligned}
O.P. &= (X_{1o}, X_{2o}, X_{3o}, u_{1o}, u_{20}, u_{30}) \\
&= (1.68301, 114.71, 65.7711, 15, 150, 1) \quad .
\end{aligned} \qquad (1\text{-}38)$$

If now I expand the functions f1, f2, f3, f4 around the O.P and knowing that:

$$\dot{x}_i(t) = f_i(x_0, u_0) + \sum_{j=1}^{n} \left(\frac{\partial f_i(x,u)}{\partial x_j} \right)_{x_0, u_0} (x_j - x_{0j}) + \sum_{j=1}^{n} \left(\frac{\partial f_i(x,u)}{\partial u_j} \right)_{x_0, u_0} (u_j - u_{0j}) \quad , \qquad (1\text{-}39)$$

introducing:

$$\begin{aligned}
\Delta X_1 &= X_1 - X_{1o}; \Delta u_1 = u_1 - u_{1o} \\
\Delta X_2 &= X_2 - X_{2o}; \Delta u_2 - u_2 - u_{2o} \\
\Delta X3 &= X_3 - X_{3o}; \Delta u_3 = u_3 - u_{3o}
\end{aligned} \quad , \qquad (1\text{-}40)$$

and considering:

$$\dot{X}_i(t) - (f_i)_{o.p} = \Delta \dot{X}_i(t) \quad , \qquad (1\text{-}41)$$

the formulas for computing the linearized model will be:

$$\begin{aligned}
\Delta \dot{X}_i(t) &= \sum_{j=1}^{3} \left(\frac{\partial f_i(x,u)}{\partial x_j} \right)_{(xo,uo)} \Delta x_j + \sum_{j=1}^{n=3} \left(\frac{\partial f_i(x,u)}{\partial u_j} \right)_{(xo,uo)} \Delta u_j \\
\Delta Y_i &= \sum_{j=1}^{3} \left(\frac{\partial g_i}{\partial x_j} \right)_{(x_0,u_0)} \Delta x_j + \sum_{j=1}^{n=3} \left(\frac{\partial g_i}{\partial u_j} \right)_{(x_0,u_0)} \Delta u_j \quad .
\end{aligned} \qquad (1\text{-}42)$$

As result [1], the linearized model can be written in the following form:

$$\dot{\overline{X}} = \begin{pmatrix} -0.173873 & -0.00480453 & 6.00964\times10^{-8} \\ -2.98081 & -0.0823552 & 1.03012\times10^{-6} \\ -6.28936 & -0.173797 & 0 \end{pmatrix} \dot{\overline{X}} + \begin{pmatrix} 0.0000172152 & 0.00478744 & 0.123679 \\ 0.000295089 & 0.0820623 & 2.12 \\ 0.00062272 & 0.173174 & 4.47378 \end{pmatrix} \overline{u}$$

$$\overline{Y} = \begin{pmatrix} 0 & 1 & 0 \\ 6.3 & 0 & 0 \end{pmatrix} \overline{X} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -4.47 \end{pmatrix} \overline{u}$$

(1-43)

1.3 SIMULATION OF THE LINEAR MODEL

The Simulink Model for the continuous flow boiling system was realized in MATLAB (Fig.1-8) and involves three inputs and two outputs. Considering the signals u_1, u_2 and u_3 as inputs, the individual step-response of the two outputs Y_1 and Y_2 can be plotted (Fig.1-9a, Fig.1-9b) using the MATLAB Program (A.2).

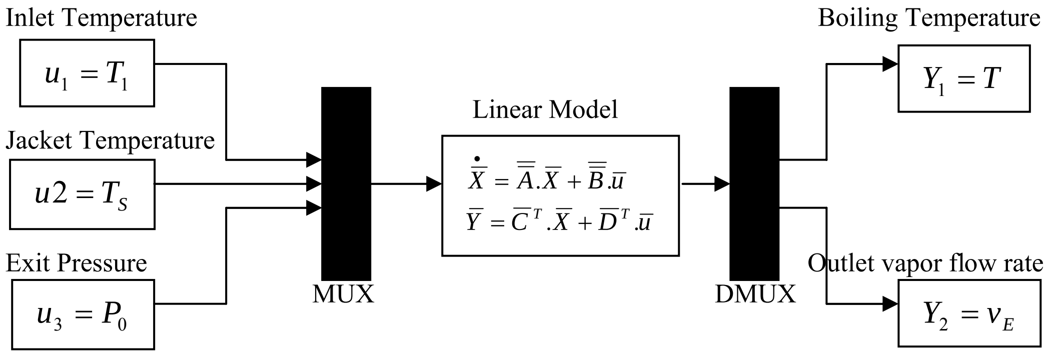


Fig. 1-8. The Simulink Model of the Linear Model.

I studied how every input affects the outputs (for the SISO system approach described in *Chapter 2*). Plotting the output response curves of the boiling system (using MATLAB (A.2)) (having three inputs (u1, u2 and u3) and two outputs (Y1 and Y2)) results are presented in Fig.1-10.

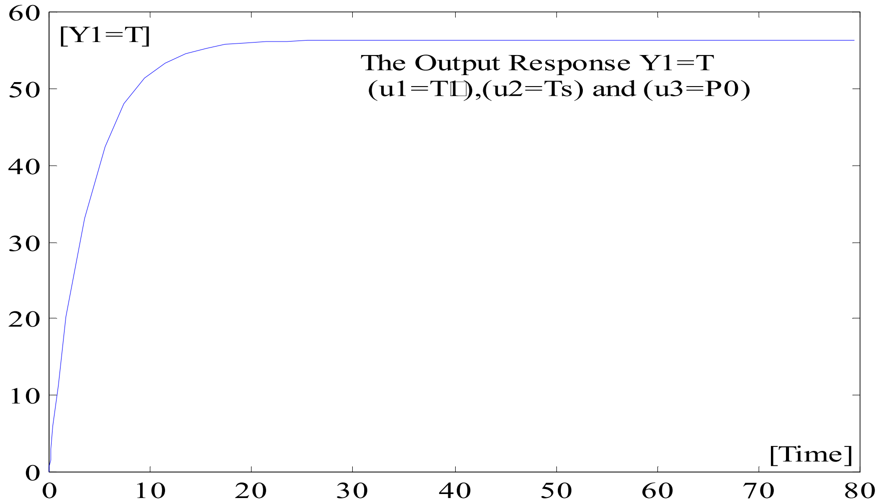


Fig. 1-9.a. The Output Step-Response Y_1=T, when the Inputs are u_1, u2 and u_3.

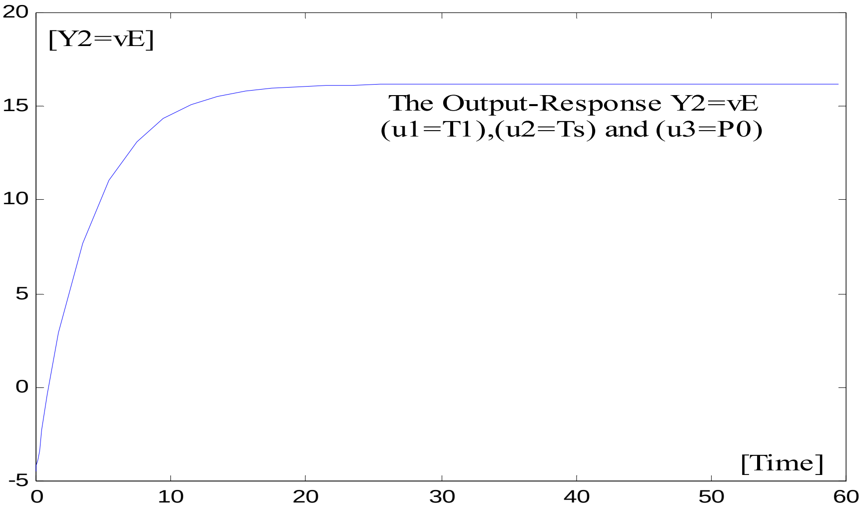


Fig. 1-9.b. The Output Step-Response $Y_2 = v_E$ when the Inputs are u_1, u2 and u_3.

Fig.1-10.a shows the output response curves Y_1, Y2 when the input is only u_1 (u_2=u_3=0), the Fig.1-10.b shows the output response curves Y_1, Y2 when the input is only u_2 (u_1=u_3=0), while Fig.1-10.c shows the output response curves Y_1, Y2 when the input is only u_3 (u_1=u_2=0). One can see that in each case the system is unstable, if only one input is considered.

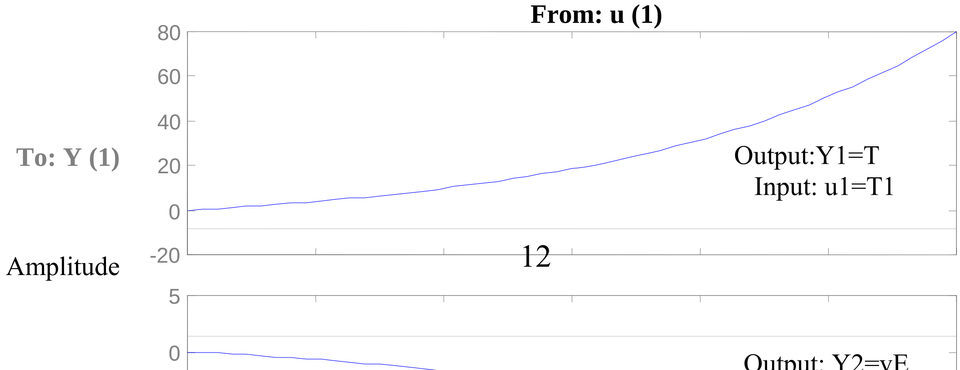


Fig 1-10.a. The Output Response Curves Y_1, Y2 when the Input is u_1 (u_2=u_3=0).

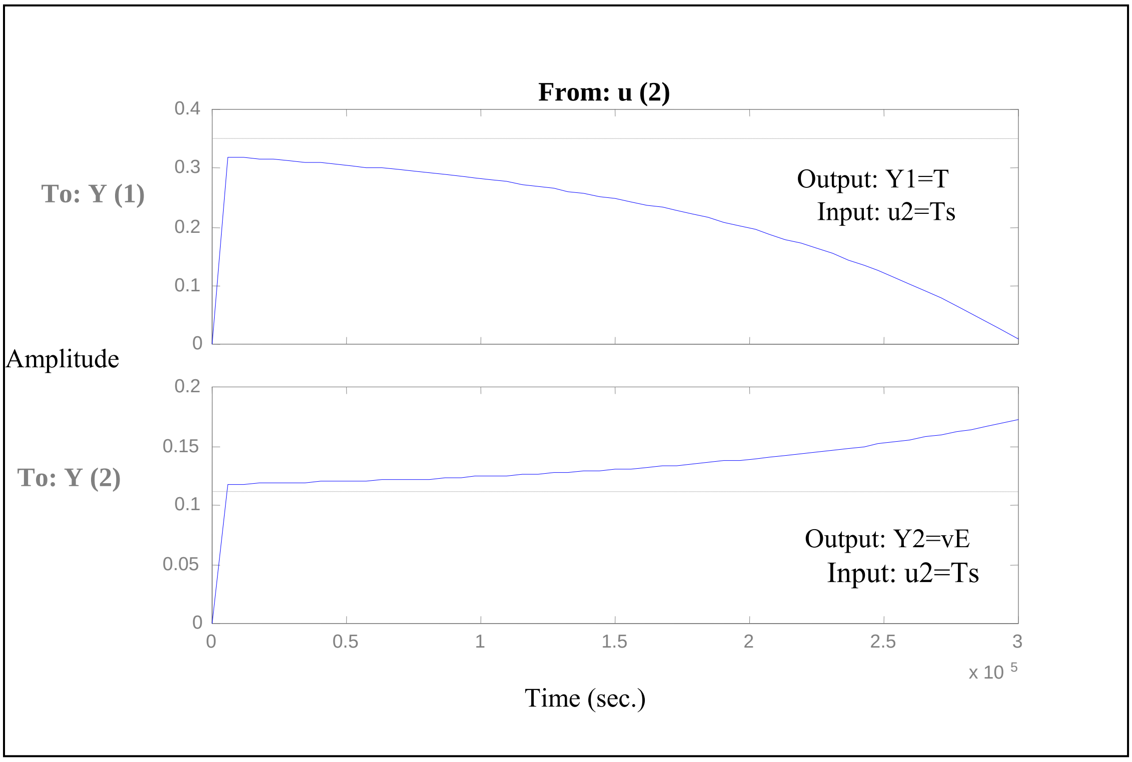


Fig 1-10.b. The Output Response Curves Y_1, Y2 when the Input is u_2 (u_1=u_3=0).

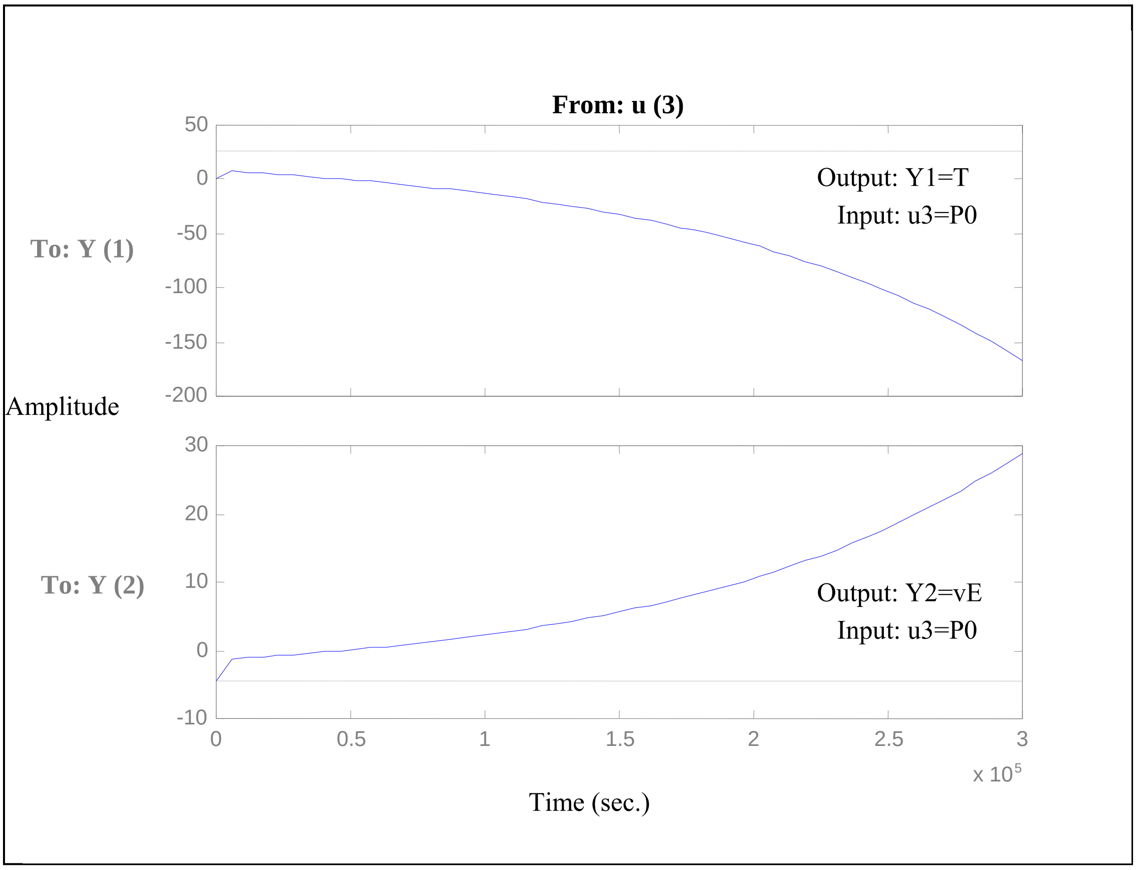


Fig 1-10.c. The Output Response Curves Y_1, Y2 when the Input is u3 (u_1=u_2=0).

CHAPTER 2

CONTROL OF A CONTINUOUS FLOW BOILING SYSTEM IN SINGLE INPUT SINGLE OUTPUT (SISO) APPROACH

2.1. BASIC CONCEPTS OF CONTROL [12]

Control means specific actions to reach prescribed, required behavior of a system, which can be a chemical process in our case. Control theory gives methods how to build up the control systems, provides methods to describe, analyze and design them. Changes in the system can be observed through signals, which in a control system can be classified as follows:

- Inner signals: state variables;
- Input signals: reference signals or disturbances;
- Output signals;
- Control signals.

Another classification can be done, e.g. according to the number of input and output signals:

- SISO: single input single output systems;
- MISO: multi input single output systems;
- MIMO: multi input multi output systems.

The general scheme of a closed-loop control system is shown by Fig.2-1.

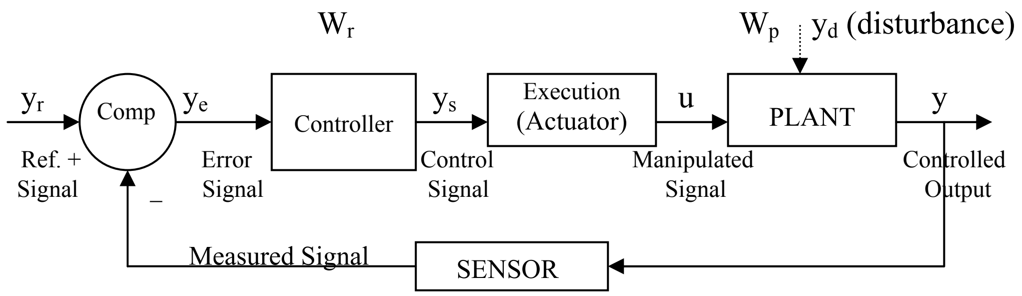


Fig. 2-1. General block diagram of a Closed-Loop System.

Arrows indicate the direction of information flow, blocks gives signal transfer properties of the individual elements executing different control tasks.
The aim of a closed-loop system is to track the reference input signal and to reject the effect of disturbances while meeting quality specific.

Specifications (quality requirements) for a closed-loop system are:

- ***Stability;***
- ***Static behavior:***

Static behavior means tracking of the reference signal with a prescribed stationary error, appropriate noise rejection.
One can analyses the static behavior of the system on the basic of the following relationships:

$$W_0(s) = W_r(s).W_p(s) \tag{2-1}$$

For a linear system superposition theorem can be applied.

$$y(s) = W(s)y_r(s) + W_d(s)y_d(s)\ , \tag{2-2}$$

where:

$$\begin{aligned} W(s) &= \langle\frac{y(s)}{y_r(s)}\rangle_{yd=0} = \frac{W_0}{1+W_0} \\ W_d(s) &= \langle\frac{y(s)}{y_d(s)}\rangle_{yr=0} = \frac{1}{1+W_0} \end{aligned} \quad . \tag{2-3}$$

As a result:

$$\begin{aligned} & y_e(s) = W_e(s)y_r(s) + W_{ed}(s)y_d(s) \\ & where: \\ & W_e(s) = \langle\frac{y_e(s)}{y_r(s)}\rangle_{yd=0} = \frac{1}{1+W_0} \\ & W_{ed}(s) = \langle\frac{y_e(s)}{y_d(s)}\rangle_{yr=0} = \frac{-1}{1+W_0} \end{aligned} \quad . \tag{2-4}$$

Let us observe, that the denominators of all transfer functions are the same, and $1+W_0(s)$ is the characteristic polynomial, whose roots determine stability conditions.

The steady state error depends on type of the system and the gain of the system, and also on the type of the input signals:

$$W_0(s) = W_r(s).W_p(s) = \frac{k}{s^i}\frac{(1+s\tau_1)(1+s\tau_2)......}{(1+sT_1)(1+sT_2)......}.e^{-S.TH} = \frac{k}{s^i}W_g(s)\ , \tag{2-5}$$

where:

$W_g(s)$: affects only transient response, $W_g\Big|_{s=0} = 1$;

K: gain;

i: type of the system, meaning the number of integrators.

Steady state error can be determined using Laplace finite value theorem.

0: type of the system, i=0 (like my case, the plant transfer function for the boiling system does not have integrator element):

$$W_e(s) = \frac{1}{1+W_0(s)} = \frac{1}{1+kW_g(s)} \quad , \tag{2-6}$$

for the step input u_r *(t)*, the steady state error will be constant: $\frac{\text{Ur}}{1 + \text{k}}$.

1: type of the system, i=1 (the plant transfer function has one integrator element):

$$W_e(s) = \frac{1}{1+W_0(s)} = \frac{1}{1+\frac{k}{s}W_g(s)} \quad , \tag{2-7}$$

for the step input u_r *(t)*, the steady state error will be zero.

2: type of the system, i=2 (the plant transfer function has two integrator elements):

$$W_e(s) = \frac{1}{1+W_0(s)} = \frac{1}{1+\frac{k}{s^2}W_g(s)} \quad , \tag{2-8}$$

for the step input u_r *(t)*, the steady state error will be zero.

The greater is the number of integrators, the better are the systems signal tracking and noise rejection properties.

- ***Dynamic behavior:***

The transient response of practical control system often exhibits damped oscillations before reaching steady state (Fig.2-2).

The transient-response characteristics of a control system to a unit step-input, is coming to specify the following:

- Delay time, td;
- Rise time, t_r;
- Peak time, t_p;
- Setting time, t_s;
- Maximum overshot, Mp;

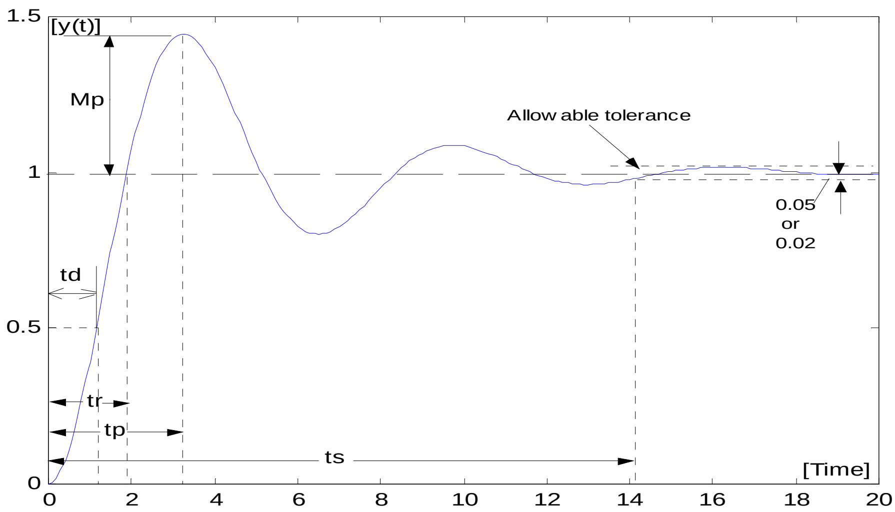


Fig. 2-2. Unit step response curve showing the quality parameters.

The maximum percent overshoot can be calculated as follows:

$$\sigma = \frac{y(t_p) - y(\infty)}{y(\infty)} \times 100\,\%, \tag{2-9}$$

The amount of the Maximum (percent) overshoot directly indicates the relative stability of the system [11].

Most control systems are time-domain systems; that means, they must exhibit acceptable time responses. (This means that the control system must be modified until the transient response is reaches a satisfactory level).

2.2. SISO SYSTEM APPROACH (PLANT, MEASUREMENT UNIT AND ACTUATOR) [1]

The MISO (multi input single output) representation for the continuous flow boiling system is shown in the Fig.2-3, the system involves three inputs and one output, meaning three transfer functions which may be defined depending in which signals are considered as input and output. Note that when considering signal u_1 as input, I assume signals u_2 and u_3 zero, and vice versa.

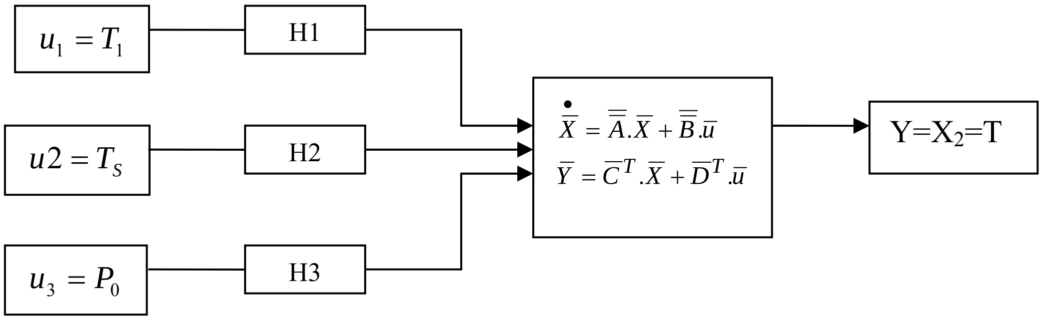


Fig. 2-3. Block diagram of the MISO Linear Model.

The calculation of the three transfer functions using MATLAB commands (A.3) gives the following results:

$$H_1(s) = \frac{Y(s)}{U_1(s)} = \frac{0.001(0.295s + 0.0164)}{s + 0.2562};$$
$$H_2(s) = \frac{Y(s)}{U_2(s)} = \frac{0.0821}{s + 0.2562}; \tag{2-10}$$
$$H_3(s) = \frac{Y(s)}{U3(s)} = \frac{2.12}{s + 0.2562};$$

For the SISO system approach, I considered as input variable the Jacket temperature ($u2 = T_S$) and as output variable the boiling temperature ($Y = X_2 = T$) (Fig. 2-4).

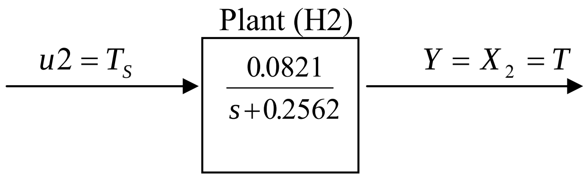


Fig. 2-4. The SISO system approach of the continuous flow boiling system and its transfer function.

The transfers function of the linear plant (SISO system):

$$H_{P1}(s) = H_2(s) = \frac{0.0821}{s + 0.2562}$$
$$H_{P1}(s) = \frac{K_{P1}}{T_{P1}s + 1} = \frac{0.3204}{3.9s + 1} \quad , \tag{2-11}$$

where $K_{P1} = 0.3204$ and $T_{P1} = 3.9$ Sec.

It can see that I get the same result with *Mathematica* program (A.1).

The problem of the measurement unit and the actuator is very important, because they make the connection and the analog-digital conversion between the plant and the controller (Fig.2-5.)

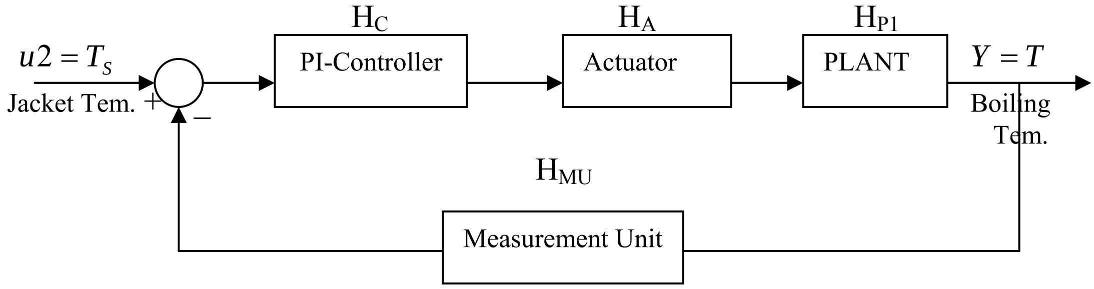


Fig. 2-5. Block Diagram of the linearized plant.

I considered for the Measurement Unit and Actuator the following transfer functions:

$$H_{MU} = \frac{1}{T_{Mu}s+1} \qquad \text{where: } T_{MU} = \frac{T_P}{10} = 0.39 \text{ Sec.} \tag{2-12}$$

$$H_A = A \quad \text{(constant(I have chosen A=1))} \tag{2-13}$$

Remark: I have chosen for the Measurement Unit's time constant a tenth smaller time constant as in case of the plant , because it is obvious that the Measurement Unit has to be faster than the process itself.

2.3. PI-CONTROLLER DESIGN USING KESSLER METHOD [6]

To assure a good response of the system (Fig. 2-6.), it is necessary to design a controller for the considered plant. I have chosen a PI controller, which will be able to reach the desired value and works without static error.

For the designing procedure I chose the Kessler-method, which is an empirical method.

In case of this method, the closed-loop transfer function must be the following form:

$$H_w(s) = \frac{H_0(s)}{1+H_0(s)} = \frac{b_0 + b_1 s}{a_0 + a_1 s + a_2 s^2 + a_3 s^3} \quad , \tag{2-14}$$

with $b_0 = a_0$ and $b_1 = a_1$ (due to the expression of $H_0(s)$).

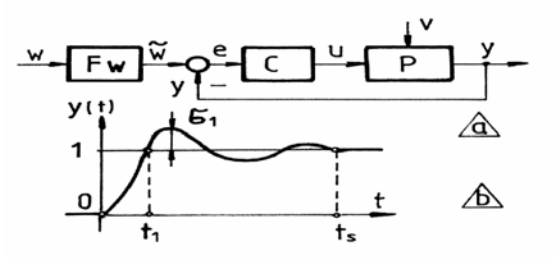


Fig. 2-6. Control system structure (a) and performance (b).

For tuning the controller's (C, Fig.2-6.a) parameters, k_C and T_C ($T_C'=T_1$ through compensation) the literature recommends the Symmetrical Optimum (SO) method introduced by Kessler (1958) that proves to be advantageous in practice because it's established tuning relations.

The optimization conditions according to the SO method are expressed as:

$$2a_0a_2 = a_1^2 \quad \text{and} \quad 2a_1a_3 = a_2^2 , \tag{2-15}$$

and lead to the following tuning relations (Table 1.a,b):

$$k_C = \frac{1}{8T_\Sigma^2 k_p} \text{ and } T_C = 4T_\Sigma \quad , \tag{2-16}$$

and to the specified performances (overshoot $\sigma_1 \approx 43\%$, settling time $T_s \approx 16.3 * T_\Sigma$, first settling time $t_1 \approx 3.7 * T_\Sigma$ (Fig.2-6.b), phase margin $\varphi_r \approx 36^o$). These performances, excepting φ_r (that cannot be modified), can be "corrected" by using a reference filter, FW (Fig.2-6.a), that does not allow to feed a step reference input w.

As a result I obtained:

$$\begin{aligned} &H_{P1} = \frac{0.3204}{3.9S+1}; H_{MU} = \frac{1}{0.39S+1}; H_A = A = 1; \\ &H_P = H_{P1}H_{MU}H_A = \frac{0.3204}{(3.9S+1)(0.39S+1)} \end{aligned} \tag{2-17}$$

The Kessler method relations are briefly presented in Table 1-a and 1-b. As it can be seen, the Modulus Criteria (CM) is used when the transfer function of the plant doesn't contain integrator components. If it contains, then the Symmetrical Criteria are used.

In my case the transfer function of the plant, including the actuator and the measurement unit has the following form:

$$H_P(s) = \frac{K_P}{(1+ST_1)(1+ST_\Sigma)}. , \tag{2-18}$$

where: K_P=0.3204; T_1=3.9 Sec. ; T_Σ =0.39 Sec.

Table (1-a) – The Modulus Criteria of Kessler method

Symb-ol	**Plant $H_P(s)$**	**C-type**	**Relations for the C (from P)**	**Quality param.**
CM 1	$\frac{K_P}{1+sT_\Sigma}$	$\frac{K_r}{s}\left(\frac{1}{sT_i}\right)$ *I-type*	$K_r = \frac{1}{2K_P T_\Sigma}$	$t_1 = 4{,}7\ T_\Sigma$ $\sigma = 4{,}3\ \%$ $t_r = 8{,}4\ T_\Sigma$
CM2	$\frac{K_P}{(1+s\cdot T_1)(1+s\cdot T_\Sigma)}$ $T_1 \rangle T_\Sigma$	$\frac{K_r}{s}(1+s\cdot T_i)$ *PI-type*	$K_r = \frac{1}{2K_P T_\Sigma}$ $T_i = T_1$	$t_1 = 4{,}7\ T_\Sigma$ $\sigma = 4{,}3\ \%$ $t_r = 8{,}4\ T_\Sigma$
CM3	$\frac{K_P}{(1+s\cdot T_1)(1+s\cdot T_2)(1+s\cdot T_\Sigma)}$ $T_1 \rangle T_2 \rangle T_\Sigma$	$\frac{K_r}{s}(1+sT_{r1})(1+sT_{r2})$ PID - type	$K_r = \frac{1}{2K_P\cdot T_\Sigma}$ $T_{r1} = T_1$ $T_{r2} = T_2$	$t_1 = 4{,}7\ T_\Sigma$ $\sigma = 4{,}3\ \%$ $t_r = 8{,}4\ T_\Sigma$

Table (1-b) – The Symmetrical Criteria of Kessler method

Symb-ol	**Plant - $H_{PC}(s)$ $T_1 > T_\Sigma$**	**C-type**	**Relations for the C (from P)**	**Quality param.**
CS1	$\frac{K_P}{s(1+sT_\Sigma)}$	$\frac{K_r}{s}(1+sT_r)$ PI - type $K_r = \frac{K_R}{T_r}$	$K_r = \frac{1}{8T_\Sigma^2 K_P}$ $T_r = 4T_\Sigma$	$\sigma_1 = 43{,}4\ \%$ $t_1 = 3{,}1\ T_\Sigma$ $t_r = 16{,}5\ T_\Sigma$
CS2	$\frac{K_P}{s(1+sT_\Sigma)(1+sT_1)}$	$\frac{K_r}{s}(1+sT_r)(1+sT_{r1})$ PID - type	$K_r = \frac{1}{8T_\Sigma^2 K_P}$ $T_{r1} = T_1,\quad T_r = 4T_\Sigma$	

From Table (1-a) (Modulus Criteria-CM2) I can compute the transfer function of the PI-Controller (PI-type):

$$Hc = \frac{K_r}{S}(1+ST_i) \quad , \tag{2-19}$$

where $T_i = T_1$, and $K_r = \frac{1}{2K_P T_\Sigma}$.

As a result I obtained: T_i = 3.9 Sec. K_r = 4.

Then, the resulted transfer function of the PI-Controller became:

$$H_C = \frac{4}{S}(1+3.9S) \quad . \tag{2-20}$$

Applying the transfer functions of the PI-Controller, Plant, Actuator and Measurement unit as in Fig.2-5. (A.4.1), the output step response for the Boiling temperature (T), when I considered a signal input the Jacket temperature (T_S) (T_S is step function from 0 to 150 C^o) is shown in the following Fig.2-7.

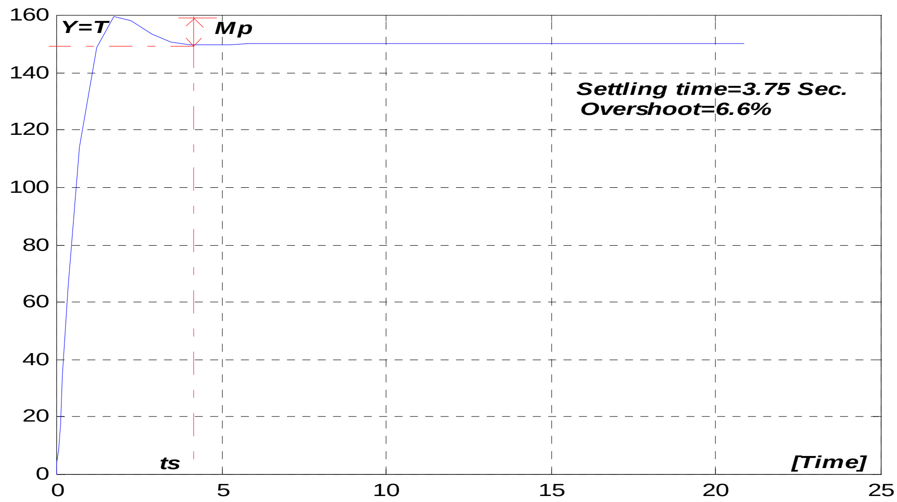


Fig. 2-7. The Output Response of the L.Plant By Kessler method.

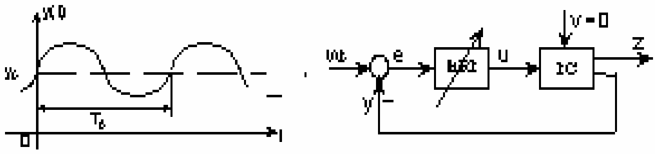
The overshoot was calculated with the relation:

$$\sigma_1 = \frac{y_{\max} - y_\infty}{y_\infty - y_0} \ (\%), \tag{2-21}$$

As it can seen the Boiling Temperature (Y=T) has reached the setpoint ($T_{S=}150\ ^oC$) at settling time 3.75 Sec. with an overshoot of 6.6%.

2.4 PI-CONTROLLER DESIGN USING THE CRITICAL GAIN METHOD [1]

This method is generally applied to systems with big time constants (like our case). In this situation the critical gain (when the system is reaching the boundary of stability (Fig. 2-8.a)) is determined for the closed loop system. At the beginning the controller was chosen as a P-controller with unit gain and with different stability methods (like Ruth-Hurwitz criteria) the critical gain can be determined (Kc) (Fig.2-8.b). At this moment the system was simulated and the period of one oscillation was read (Fig. 2-8.a). Finally with the Ziegler-Nichols tuning relations (Table 2) could be calculated obtaining the desired controller.



(a) (b)

Fig. 2-8. The system response at boundary of stability (with P-controller).

Table 2 - Ziegler -Nichols relations:

C-type	**Controller Parameters**		
	$\mathbf{K_r}$	$\mathbf{T_i}$	$\mathbf{T_d}$
P	0.5 K_{rpo}	-	-
PI	0.45 K_{rpo}	0.85 T_0	-
PID	0.6 K_{rpo}	0.5 T_0	0.125 T_0

where:

K_{rp} –gain of the P controller;

K_{rp0}- gain of the P controller when the system is reaching the boundary of stability;

T_0- one period of time at the boundary of stability.

Numerically I obtained for the closed loop system:

$$H_w(s) = \frac{0.3204K_C}{(3.9s+1)(0.39s+1)+0.3204K_C} \tag{2-22}$$

Hence the Routh–Hurwitz table for this case is:

$\mathbf{S^2}$	1.52	1+0.3204 K_c
$\mathbf{S^1}$	4.29	0
$\mathbf{S^0}$	1+0.3204 K_c	0

where the caractacteristical polynomial of the system is:

$$1.521s^2 + 4.29s + (1+0.3204K_C) = 0\,, \tag{2-23}$$

As a result the system is stable if: 1+0.9859 Kc > 0.

Therefore ***Kc* < *-1/0.3204*** means that I can not obtain a simulation for the determined stability condition: negative value of K_c, so I can not obtain a value for T_0. In conclusion, for these numerical values this method can not be used.

2.5. PI-CONTROLLER DESIGN IN FREQUENCY DOMAIN [12]

Design objectives in the frequency domain:

- ***Stability*:** In the open loop Bode amplitude diagram ω_C is on a straight line of slope (-1) meaning (-20 dB/dec) for stable system as shown in the Fig.2-9.

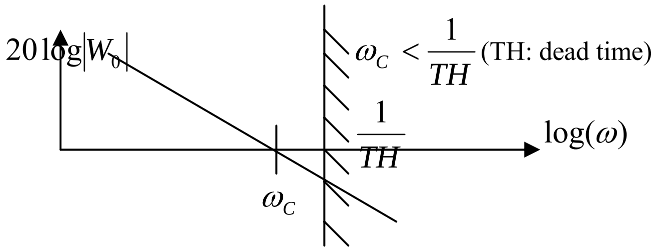


Fig 2-9. Open loop Bode amplitude diagram.

- ***Quasistationary error:*** Requirements for steady state depends on :
  - Type number of the system (number of integrators on the system);
  - The Gain (K): To reach the prescribed accuracy for the gain (K).

- ***Good transient response:*** means as quick as possible, no significant overshoot in the closed loop Bode amplitude diagram, or $\varphi_M \approx 60^\circ$, or if the closed loop substituted by a second order element, $\xi \approx 0.6 - 0.7.(\omega_C \approx \frac{1}{2TH})$., ω_C is big as possible for a fast transient.

In our case: the steps for tuning a controller for the Boiling temperature (SISO system) are:

- Determine the transfer function of the open loop system (with unit value of the controller's gain);
- Draw the Bode diagrams;
- Choose a value for the phase margin (in the [45, 60]° domain);

- For this value, read from the Bode-diagrams the amplitude and recalculate the gain of the open loop system from what the controller's gain can be calculated ($k_0=k_pk_r$);
- The time constant of the controller is chosen to compensate the plant's big time constant.

Numerically, for my case the transfer function of the open loop is:

$$H_O(s) = H_C(S).H_P(S) = \frac{0.3204K_C}{3.9S(1+0.39S)} \tag{2-24}$$

Writing a short program in MATLAB to calculate the amplitude at the desired phase margin ($\varphi_r = 60^o$), I obtained the following value for the Bode diagram (Fig.2-10.).

Num. = [1];

Den. = [0.39   1   0]

ω =log space (-2, 1, 100)

[Mag. phase] =Bode (num., den, ω);

Bode (num, den, ω)

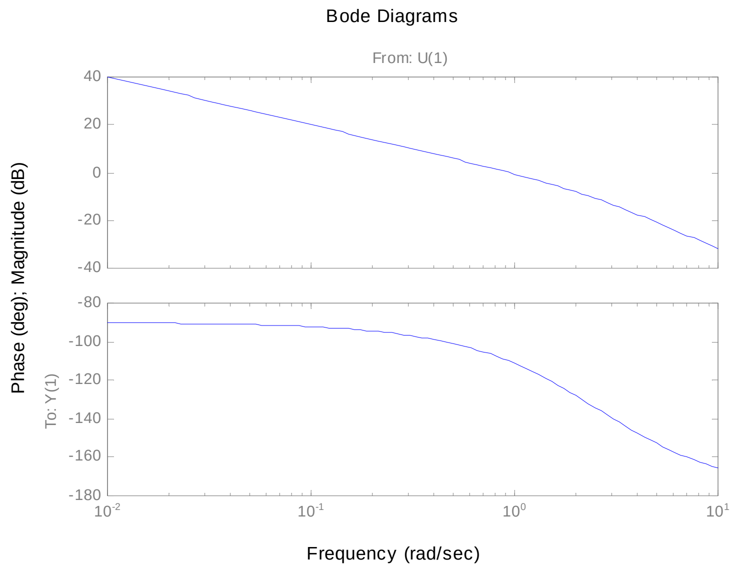


Fig. 2-10. Bode diagrams for the considered system**.**

The results are:

ω`	Amplitude	Phase
1.5557	0.5496	-121.2458

Hence, I can calculate the PI-controller:

$$K_C=11.426 \quad , \quad H_C = \frac{11.426}{3.9S}(1+3.9S) \tag{2-25}$$

Then:

$$H_C = \frac{2.929}{S}(1+3.9S) \tag{2-26}$$

Simulating the system for the tuned PI-controller, the output response is shown by Fig.2-11. (A.4.2). I could observe that the boiling temperature (Y=T) reach the setpoint 150 C^o at settling time 5.6 Sec. and with overshoot 1.4%.

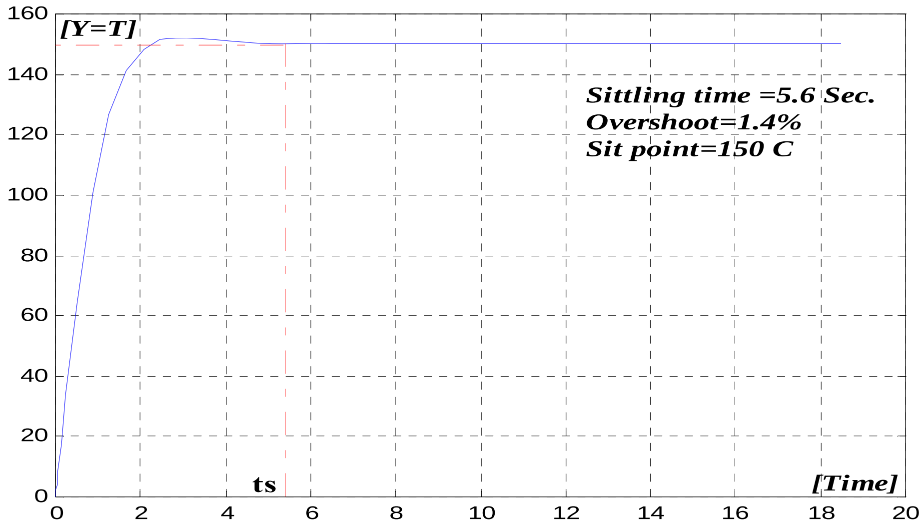


Fig. 2-11. Step-response output (Y=T) of the boiling temperature for PI-controller tuned in frequency domain.

2.6. OVERVIEW FOR SISO SYSTEM

Return back to the SIMULINK model for a continuous flow boiling system presented in Fig.2-8., and considering the signals u_1, u_3 as step disturbances at step time 50 Sec. and u_2 as step input (sit point), the step-response output [Y=T] of the boiling temperature response is presented by Fig.2-12.(A.4.3).

As it can be seen, the boiling temperature (Y=T) does not reach the setpoint ($u2=T_S=150C^o$) because the disturbances u_1, u_3 disturbed the system.

Now, applying PI-controller tuned in frequency domain for the same model (Simulink model Fig.2-8, when u_1, u_3 are considered as disturbances and u_2 as step input, the step-response output [Y=T] of the boiling temperature can be plotted as shown in the Fig.2-13. (A.4.4).

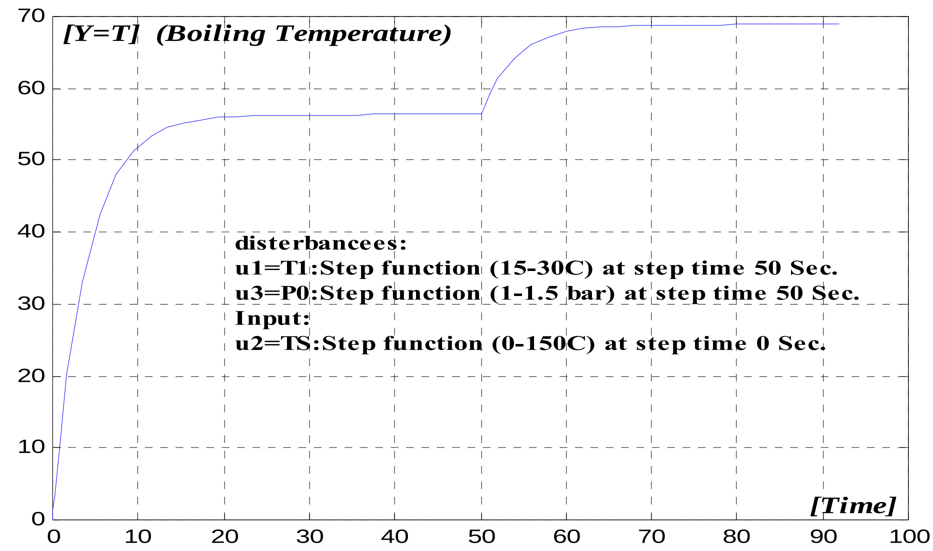


Fig. 2-12. Step-response output (Y=T) of the boiling temperature when consider u_1, u3 are disturbances and u_2 is step input.

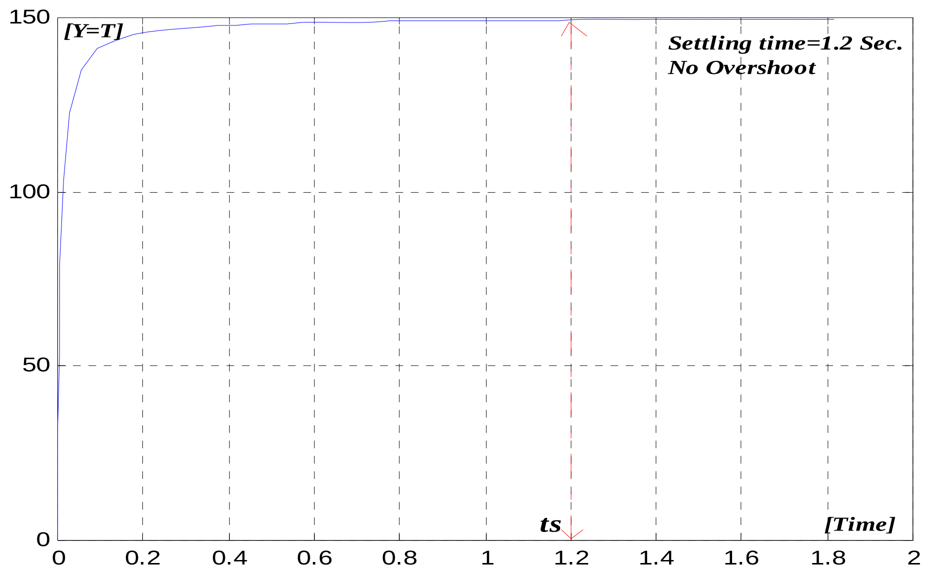


Fig. 2-13. Step-response output (Y=T) when consider u_1, u3 are disturbances and u_2 is step input and applying PI-controller tuned in frequency domain.

One can observe that the boiling temperature (Y=T) reach the setpoint 150 C^o at settling time 1.2 Sec. and without overshoot.

SUMMARY. From the SIMULINK model for a continuous flow boiling system as shown in the Fig.2-8., I have summarized the following conclusions:

- ❖ Considering the signals u_1, u_2 and u_3 as inputs, the two individual step-response outputs Y_1 and Y_2 have good quality parameters (settling time, overshoot), but does not reach the setpoint.
- ❖ Considering the signals u_1, u_3 are step disturbances and u_2 as input, the step-response output of the boiling temperature does not reach the setpoint, because the disturbances u_1, u_3 disturbed the system.
- ❖ The three used tuning methods presented demonstrated once again that the overshoot and the settling time are "in contradiction".
- ❖ The Kessler method, due to his "receipt" form is very easy, but the quality parameters of the system can not be optimized.
- ❖ The tuning in the frequency domain is more difficult, but it gives better performances and also the quality parameters can be reconsidered.
- ❖ The Ziegler-Nichols tuning method (critical gain) is also very simple, but it works only for very slow and robust systems.
- ❖ Applying PI-controller tuned in frequency domain for the same model, I could observe that the boiling temperature reach the setpoint at settling time 1.2 Sec. and without overshoot. Thus, the controller eliminates the effect of the disturbances.

CHAPTER 3

CONTROL OF CONTINUOUS FLOW BOILING SYSTEM IN MULTI INPUT MULTI OUTPUT (MIMO) APPROACH

3.1 MIMO SYSTEM APPROACH [2]

As a MIMO system approach the system is extended with other output (vapor pressure) and the mathematical model could be built up considering the following variables:

- Input variables:
  - Inlet temperature $u_1=T_1$;
  - Jacket temperature $u_2=T_S$;
  - Exit pressure $u_3=P_0$;
- State Variables:
  - Vapor pressure: $X_1=P$;
  - Boiling temperature: $X_2=T$;
  - Vapor mass: $X_3=m_G$;
- Output variables:
  - Boiling temperature: $Y_1 = X_2 = T$;
  - Outlet vapor flow rate: $Y_2 = v_E$;
  - Vapor pressure $Y_3 = X_1 = P$;
- Constants:

R=1.98 moles/ ft^3 ,c1=13.96 $KJoule/(Kmol.^\circ C)$,c2=-5210.6 $KJoule/(Kmol.^\circ C)$,V_G=30000 ft^3 , λ=9717 PCU/mole, UA=1700, K=5.7 $ft^3/(Kmol.\sec.)$

Due to the state space description of the non-linear model in the equations (1-32), the non-linear mathematical model for the MIMO Boiling system will be:

$$\dot{X}_1 = \left(\frac{R(X_2 + 273).X_1.(\ln X_1 - c_1)^2}{V.X_1.(\ln X_1 - c1)^2 + c_2 R X_3} \right) \left(\frac{UA(u_2 - X_2)}{X_2 - u_1 + \lambda} - K\sqrt{X_1(X_1 - u_3)} \right),$$

$$\dot{X}_2 = \left(\frac{-c_2 R(X_2 + 273)}{V.X_1.(\ln X_1 - c1)^2 + c_2 R X_3} \right) \left(\frac{UA(u_2 - X_2)}{X_2 - u_1 + \lambda} - K\sqrt{X_1(X_1 - u_3)} \right),$$

$$\dot{X}_3 = \frac{UA(u_2 - X_2)}{X_2 - u_1 + \lambda} - K\sqrt{X_1(X_1 - u_3)}, \qquad (3\text{-}1)$$

$$Y_1 = X_2,$$

$$Y_2 = K\sqrt{X_1(X_1 - u_3)},$$

$$Y_3 = X1.$$

Taking from (3-1) the non-linear functions and expanding them in Taylor series around the steady state operating point (as we did in the *Chapter 2*) the following linearized mathematical model can be obtained:

$$\dot{\overline{X}} = \begin{pmatrix} -0.173873 & -0.00480453 & 6.00964\times10^{-8} \\ -2.98081 & -0.0823552 & 1.03012\times10^{-6} \\ -6.28936 & -0.173797 & 0 \end{pmatrix} \overline{X} + \begin{pmatrix} 0.0000172152 & 0.00478744 & 0.123679 \\ 0.000295089 & 0.0820623 & 2.12 \\ 0.00062272 & 0.173174 & 4.47378 \end{pmatrix} \overline{u}, \qquad (3\text{-}2)$$

$$\overline{Y} = \begin{pmatrix} 0 & 1 & 0 \\ 6.29 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \overline{X} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -4.47 \\ 0 & 0 & 0 \end{pmatrix} \overline{u}$$

where:

$$\overline{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} P \\ T \\ m_G \end{bmatrix}, \quad \overline{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_S \\ P_0 \end{bmatrix}, \quad \overline{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} T \\ v_E \\ P \end{bmatrix}. \qquad (3\text{-}3)$$

It can be seen that in this case the D matrix is not zero.

3.2 SYSTEM DECOUPLING AND CONTROLLER DESIGN USING THE P-CANONICAL FORM

The P-canonical form can be applied for every multivariable system (Fig.3-1). The matrix transfer function in this case (number of the inputs=number of the outputs) is given by:

$$\overline{Y}(s) = \overline{\overline{H}}(s) \circ \overline{u}(s) \quad \text{where: } \overline{\overline{H}}(s) = [\overline{\overline{H}}_{ij}], \quad \text{where : } i,j=1,\ldots\ldots,n \tag{3-4}$$

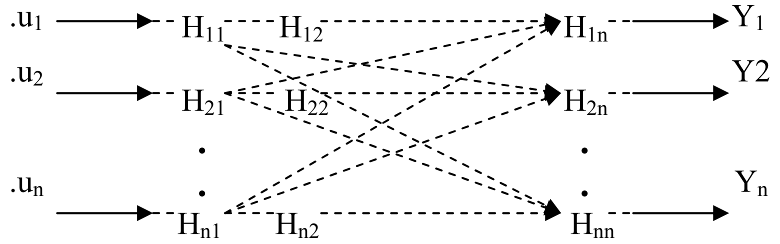


Fig. 3-1. The P-canonical form of a general multivariable system.

In my case the dependency between the outputs and the inputs of the boiling system is described by relation (3-5) and Fig. 3-2:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{3-5}$$

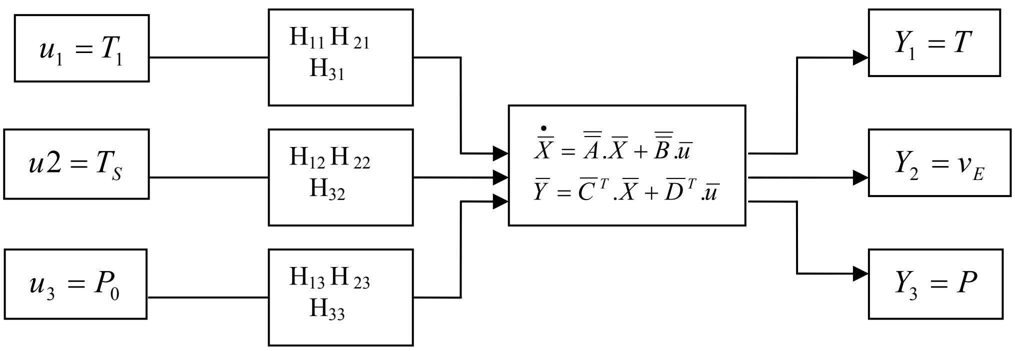


Fig. 3-2. The P-canonical form of the multivariable boiling system.

We can compute the transfer functions matrix with MATLAB (A.3):

$$\left[num, den\right] = ss2tf(A, B, C, D, 1);$$
$$H11 = \frac{y1}{u1} = \frac{0.00115}{3.9s+1}; H21 = \frac{y2}{u1} = \frac{0.000287}{3.9s+1}; H31 = \frac{y3}{u1} = \frac{0.0000456}{3.9s+1};$$
$$\left[num, den\right] = ss2tf(A, B, C, D, 2);$$
$$H12 = \frac{y1}{u2} = \frac{0.3204}{3.9s+1}; H22 = \frac{y2}{u2} = \frac{0.11748}{3.9s+1}; H32 = \frac{y3}{u2} = \frac{0.0187}{3.9s+1};$$
$$\left[num, den\right] = ss2tf(A, B, C, D, 3);$$
$$H13 = \frac{y1}{u3} = \frac{8.275}{3.9s+1}; H23 = \frac{y2}{u3} = \frac{-17.462s-1.438}{3.9s+1}; H33 = \frac{y3}{u3} = \frac{0.4828}{3.9s+1};$$

(3-6)

The desired steady state values for the input, output variables are:

$u_{1o}=T_1=15$ ºC, $u_{2o}=T_S=150$ ºC and $u_{30}=P_0=1$ atom (3-7)

$X_{1o}=P=1.68301$ atom, $X_{2o}=T=113.71$ ºC and $X_{3o}=m_G=65.7711$(unit of mass).

The following steps were carried out for the linear decoupling of the boiling system:

- I consider the diagonal form of the transfer functions matrix;
- I design the controllers for the decoupled system;
- With the so obtained controllers, I return and apply the controllers for the coupled system.

I carried out the controller for each channel using the frequency domain with the following steps:

- I determine the transfer function of the open loop system (with unit value of the controller's gain);
- I draw the Bode diagrams;
- I choose a value for the phase margin (in the [45, 60]° domain), namely 60°;
- For this value I read from the Bode-diagrams the amplitude and recalculate the gain of the open loop system from what the controller's gain can be calculated ($k_0=k_pk_r$);
- The time constant of the controller was chosen in the way to compensate the plant's big time constant.

Therefore the form of the PI-controllers (for $H_{11(}$s), H_{22}(s) and H_{33}(s)) must be:

$$H_{C1} = \frac{k_{C1}}{3.9s}(3.9s+1); H_{C2} = \frac{k_{C2}}{3.9s}(3.9s+1); H_{C3} = \frac{k_{C3}}{3.9s}(3.9s+1); \quad (3\text{-}8)$$

Numerically, for our case the transfer function of the open loop is:

$$H_{oi}(s) = H_{ci}(s).H_{ii}(s).H_{Mui}(s); where: i = 1,2,3$$
$$H_{oi}(s) = \frac{k_{pi}k_{ci}}{3.9s(1+0.39s)}; \tag{3-9}$$

I calculated in MATLAB the amplitude at the desired phase margin ($\varphi_r = 60^o$):

$$\begin{aligned} &num = [1];\\ &den = [1.521 \quad 3.9 \quad 0];\\ &w = \log space(-2,1,100);\\ &[mag, phase] = Bode(num, den, \omega);\\ &Table = [\omega', mag, phase] \end{aligned} \tag{3-10}$$

The results are:

ω'	mag.	Phase
1.5199	0.5396	-120.62

Hence, I could calculate the gain of the PI-controllers:

k_{c1}=816.2378; k_{c2}=7.990168; k_{c3}=1.93325;

Consequently, the obtained PI-controllers for the decoupled boiling system are given by:

$$\begin{aligned} H_{c1}(s) &= \frac{816.2478}{3.9s}(3.9s+1);\\ H_{c2}(s) &= \frac{7.990168}{3.9s}(3.9s+1);\\ H_{c3}(s) &= \frac{1.94425}{3.9s}(3.9s+1); \end{aligned} \tag{3-11}$$

The results for the decoupled system (A.4.5) are shown by Fig. 3-3.a, b and c. It can be seen that the system is reaching the desired steady state values.

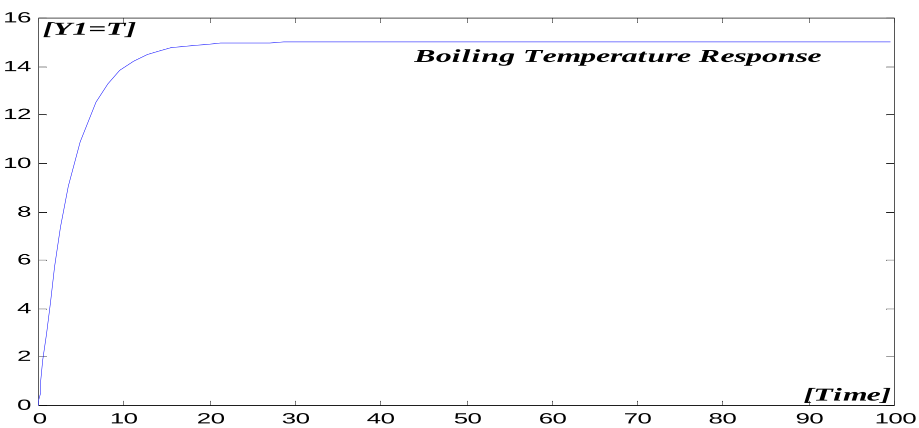


Fig.3-3.a. The first output response (Y_1=X_2=T) for the decoupled system.

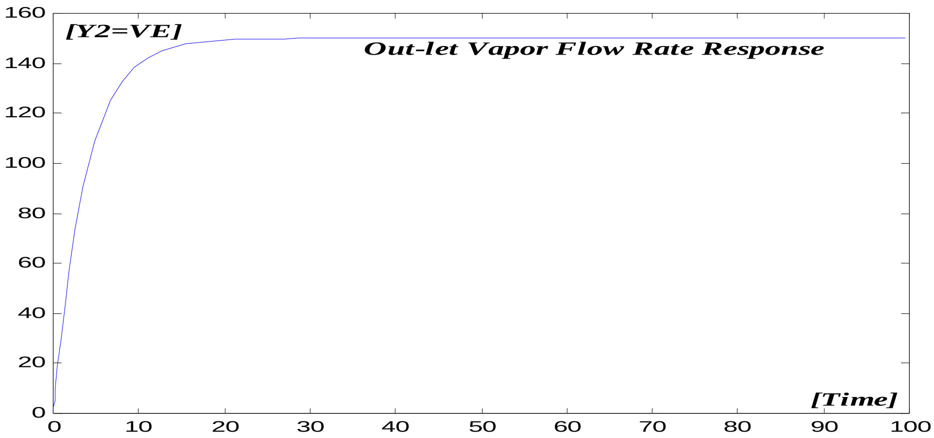


Fig.3-3.b. The second output response ($Y_2=V_E$) for the decoupled system.

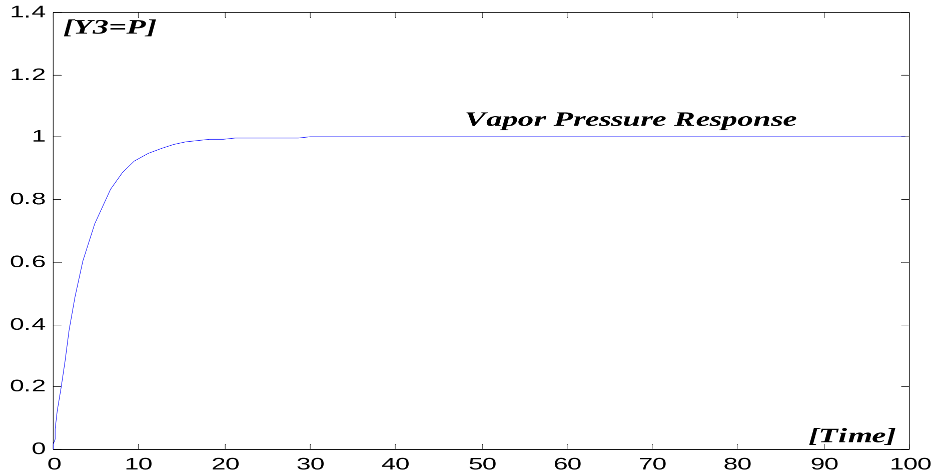


Fig.3-3.c. The third output response ($Y_3=X_1=P$) for the decoupled system.

With the so obtained controllers I turned back to the coupled boiling system and checked its behavior (Fig. 3-4.a, b, and c). The MATLAB-Simulink model of the coupled system is shown in (A.4.6).

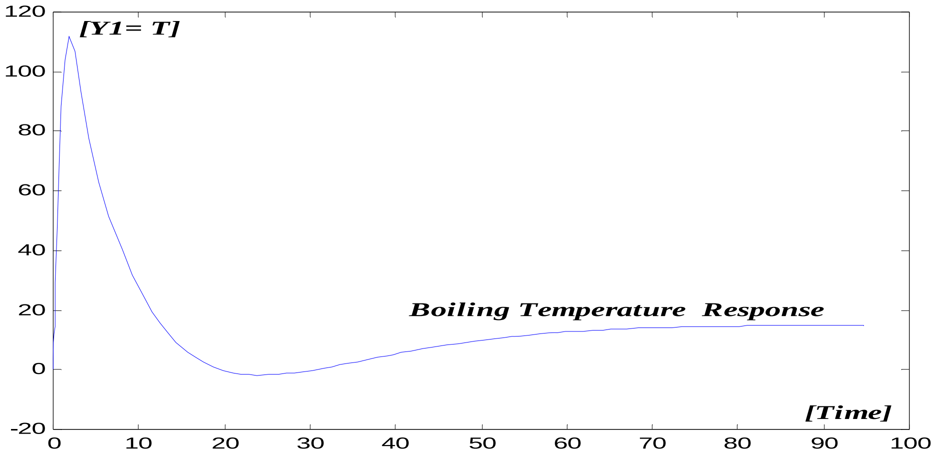


Fig.3-4.a. The first output response (Y_1=X_2=T) for the coupled system.

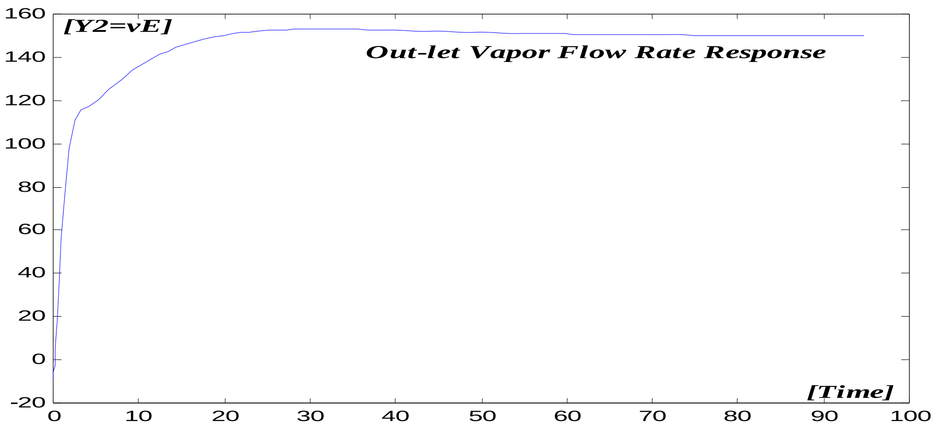


Fig.3-4.b. The second output response (Y_2=V_E) for the coupled system.

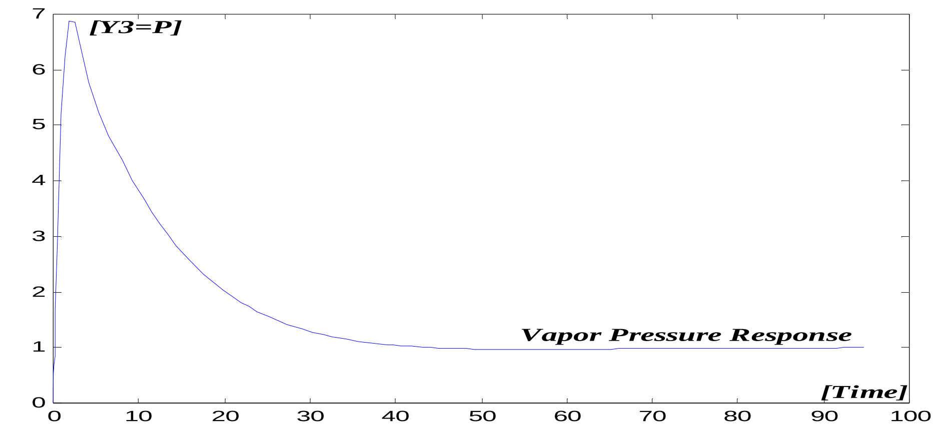


Fig.3-4.c. The third output response (Y_3=X_1=P) for the coupled system.

Comments of these results:

It can seen that in coupled way the strong dependencies between the boiling temperature T, and the vapor pressure P, gives in this way big difference in the response with those of the decoupled system (mostly in the overshoot). Therefore, a non-linear decoupling algorithm is checked for better performances.

3.3 NON-LINEAR DECOUPLING ALGORITHM OF THE MIMO SYSTEM

3.3.1 THEORETICAL DESCRIPTION OF THE METHOD [7]

Let us consider the state space representation of a non-linear system:

$$\begin{cases} \dot{\underline{x}}(t) = \underline{A}(\underline{x},t) + \underline{B}(\underline{x},t)\,\underline{u}(t) \\ \underline{y}(t) = \underline{C}(\underline{x},t) + \underline{D}(\underline{x},t)\,\underline{u}(t) \end{cases} \tag{3-12}$$

where: $\underline{x}$ – state variables-vector (*dim* $\underline{x}$ = *n*), $\underline{u}$ – inputs-vector (*dim* $\underline{u}$ = *m*), $\underline{y}$ – outputs-vector (*dim* $\underline{y}$ = *m*). The mathematical model matrixes are: {$\underline{A}(\underline{x},t)$ - (n∗1) matrix, $\underline{B}(\underline{x},t)$ - (n∗m) matrix, $\underline{C}(\underline{x},t)$ - (m∗1) matrix, $\underline{D}(\underline{x},t)$ - (m∗m) matrix}.

For decoupling the following state-feedback is introduced:

$$\underline{u}(t) = \underline{F}(\underline{x},t) + \underline{G}(\underline{x},t)\,\underline{w}(t)\ , \tag{3-13}$$

so the system will be:

$$\begin{cases} \dot{\underline{x}}(t) = \{\underline{A}(\underline{x},t) + \underline{B}(\underline{x},t)\,\underline{F}(\underline{x},t)\} + \underline{B}(\underline{x},t)\,\underline{G}(\underline{x},t)\,\underline{w}(t) \\ \underline{y}(t) = \{\underline{C}(\underline{x},t) + \underline{D}(\underline{x},t)\,\underline{F}(\underline{x},t)\} + \underline{D}(\underline{x},t)\,\underline{G}(\underline{x},t)\,\underline{w}(t) \end{cases} . \tag{3-14}$$

For decoupling, one wants that the i-th input has an effect only on the i-th output. Therefore the $\underline{F}$ and $\underline{G}$ matrixes must be determined due to **Differential Order (di)** [13]. The Differential Order (di) shows what order of the yi-th derivate (by time) depends only by the ui-th input [5]. Practically, it shows how many poles can be choosen randomly:

$$y_i(t) = C_i(\underline{x},t) + D_i(\underline{x},t)\,\underline{u}(t)\ \ , \tag{3-15}$$

where: $\underline{Ci}$ and $\underline{Di}$ are the i-th row vector of $\underline{C}$ and $\underline{D}$.

So the decoupling algorithm depends on two cases of the Differential Order (di):

- If $\underline{D}_i(\underline{x},t) \neq 0$, then $d_i = 0$ and the system can be decoupled;
- If $\underline{D}_i(\underline{x},t) = 0$, then $d_i > 0$ and the following steps have to be done:

$$y_i(t) = C_i(\underline{x},t) \ . \tag{3-16}$$

$$\dot{y}_i(t) = \frac{d}{dt}[C_i(\underline{x},t)] = \frac{\partial}{\partial t} C_i(\underline{x},t) + \frac{\partial}{\partial \underline{x}}[C_i(\underline{x},t)] \cdot \underline{\dot{x}} \ , \tag{3-17}$$

where:

$$\frac{\partial}{\partial \underline{x}}[C_i(\underline{x},t)] = \left(\frac{\partial}{\partial x_1}[C_i(\underline{x},t)], \ldots, \frac{\partial}{\partial x_n}[C_i(\underline{x},t)] \right). \tag{3-18}$$

The result will be then:

$$\dot{y}_i(t) = \frac{\partial}{\partial t} C_i(\underline{x},t) + \left\{ \frac{\partial}{\partial \underline{x}}[C_i(\underline{x},t)] \right\} \cdot \left\{ \underline{A}(\underline{x},t) + \underline{B}(\underline{x},t)\underline{u}(t) \right\} =$$

$$= \frac{\partial}{\partial t} C_i(\underline{x},t) + \left\{ \frac{\partial}{\partial \underline{x}}[C_i(\underline{x},t)] \right\} \underline{A}(\underline{x},t) + \left\{ \frac{\partial}{\partial \underline{x}}[C_i(\underline{x},t)] \right\} \underline{B}(\underline{x},t)\underline{u}(t) \tag{3-19}$$

The following notation is introduced:

$$N_A^k C_i(\underline{x},t) = \frac{\partial}{\partial t} N_A^{k-1} C_i(\underline{x},t) + \left\{ \frac{\partial}{\partial \underline{x}} N_A^{k-1} C_i(\underline{x},t) \right\} \underline{A}(\underline{x},t) \qquad , \tag{3-20}$$

where initially:

$$N_A^0 C_i(\underline{x},t) = C_i(\underline{x},t) \tag{3-21}$$

As a result one obtains:

$$\dot{y}(t) = N_A^1 C_i(\underline{x},t) + \left\{ \frac{\partial}{\partial \underline{x}} N_A^0 C_i(\underline{x},t) \right\} \underline{B}(\underline{x},t)\underline{u}(t) \tag{3-21.a}$$

Rewriting the algorithm, for this case follows:

1. If

$\left[\frac{\partial}{\partial \underline{x}} N_A^0 C_i(\underline{x},t) \right] \underline{B}(\underline{x},t) \neq \underline{0},$ then $d_i = 1$.

2. If

$\left[\frac{\partial}{\partial \underline{x}} N_A^0 C_i(\underline{x},t) \right] \underline{B}(\underline{x},t) = \underline{0},$ then d_i can be determined by the second

derivate of yi ...etc.

So the **conclusion** of the Differential Order (di) in a MIMO system decoupling is [13]:

a. If $\underline{D}_i(\underline{x},t) \neq 0$, then : $d_i = 0$; (3-22)

b. If $\underline{D}_i(\underline{x},t) = 0$, then :

$$d_i = \min\left\{ j : \left[\frac{\partial}{\partial \underline{x}} N_A^{j-1} C_i(\underline{x},t) \right] \underline{B}(\underline{x},t) \neq 0; j = 1,2,\ldots,n \right\} \tag{3-23}$$

Following the so introduced d_i one can make the decoupling of the system. From (3-15):

$$\underline{y}^*(t) = \underline{C}^*(\underline{x},t) + \underline{D}^*(\underline{x},t)\underline{u}(t) \quad , \tag{3-24}$$

where: $\underline{y}^*(t) = [y_1^{d1},\ldots,y_m^{dm}]$, $\underline{C}^*(\underline{x},t)$ is an m dimensional row matrix and $\underline{D}^*(\underline{x},t)$ is an (m*m) dimensional matrix.

For the i-th output:

$$C_i^*(\underline{x},t) = N_A^{di} C_i(\underline{x},t), \qquad i = 1, 2,..., m \quad , \tag{3-25}$$

$$\underline{D}_i^*(\underline{x},t) = \begin{cases} D_i^*(\underline{x},t) & \text{, if di} = 0 \\ \left[\frac{\partial}{\partial \underline{x}} N_A^{di-1} C_i(\underline{x},t) \right] \underline{B}(\underline{x},t) & \text{, if di} \neq 0 \end{cases} . \tag{3-26}$$

Therefore:

$$\underline{u}(t) = \underline{F}_1(t) = -\underline{D}^{*-1}(\underline{x},t)\underline{C}^*(\underline{x},t). \tag{3-27}$$

The result will be due to the steady state:

$$\underline{y}^*(t) = 0 \ . \tag{3-28}$$

So the system is decoupled, but that's only if:

$$\underline{u}(t) = \underline{F}_1(\underline{x},t) + \underline{G}(\underline{x},t)\,\underline{w}(t) \quad , \tag{3-29}$$

where: $\underline{G}(\underline{x},t) = \underline{D}^{*-1}(\underline{x},t)\underline{L}$, and $\underline{L} = \mathrm{diag}\{\lambda_i\}$ (λ_i denote the gain of the system), so as a result:

$$\underline{y}^*(t) = \underline{L}\cdot \underline{w}(t) \ . \tag{3-30}$$

If we want also to introduce dynamics for the system, not only the case of steady state, then:

$$\underline{u}(t) = \underline{F}(\underline{x},t) + \underline{G}(\underline{x},t)\,\underline{w}(t) \ , \tag{3-31}$$

where: $\underline{F}(\underline{x},t) = \underline{F}_1(\underline{x},t) + \underline{F}_2(\underline{x},t)$ and $\underline{F}_2(t) = -\underline{D}^{*\,-1}(\underline{x},t)\underline{M}^*(\underline{x},t)$, ($\underline{M}^*(\underline{x},t)$ is an m-dimensional less space vector.

As a result:

$$\underline{y}^*(t) = -\underline{M}^*(\underline{x},t) + \underline{L}\cdot\underline{w}(t) \ . \tag{3-32}$$

where:

$$\underline{M}^*_i(\underline{x},t) = \begin{cases} 0 & \text{, if } \ d_i = 0 \\ \sum_{k=0}^{d_i-1} \alpha_{ki} N_A^k C_i(\underline{x},t) & \text{, if } \ d_i \neq 0 \end{cases} \ , \tag{3-33}$$

and α_i can be freely chosen.

Consequently, the decoupling equation will be (Fig. 3-5.):

$$y_i^{(d_i)} + \alpha_{(d_i-1)i} y_i^{(d_i-1)} + \ldots + \alpha_{0i} y_i = \lambda_i w_i(t) \ , \qquad i = 1, 2 \ldots m \tag{3-34}$$

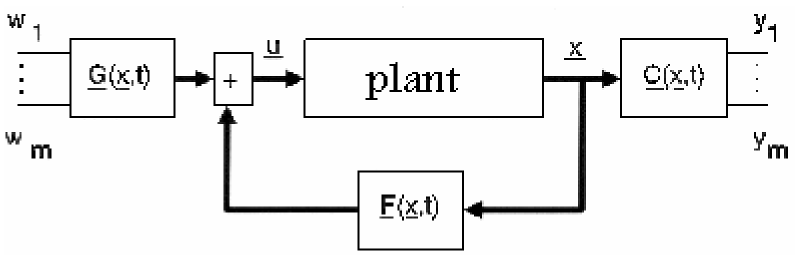


Fig.3-5. Decoupling representation for the non-linear model.

Summarizing, the final relations for the $\underline{F}$ and $\underline{G}$ matrixes are:

$$\underline{F}(\underline{x},t) = -\underline{D}^{*\,-1}(\underline{x},t)[\underline{C}^*(\underline{x},t) + \underline{M}^*(\underline{x},t)] \ , \tag{3-35}$$

$$\underline{G}(\underline{x},t) = \underline{D}^{*\,-1}(\underline{x},t)\underline{L} \ .$$

3.3.2 APPLYING THE NON-LINEAR DECOUPLING METHOD FOR THE MIMO BOILING SYSTEM

First, I write the state space representation of a non-linear system (3-1) in the following form:

$$\begin{cases} \dot{\underline{x}}(t) = \underline{A}(\underline{x},t) + \underline{B}(\underline{x},t)\underline{u}(t) \\ \underline{y}(t) = \underline{C}(\underline{x},t) + \underline{D}(\underline{x},t)\underline{u}(t) \end{cases} \tag{3-36}$$

As result I obtain the following numerical model:

$$\dot{X}1 = g1(x,t).\frac{1}{x2^3}((-1700\,x2^2 - 1700x2u1 + 1700u1u2)u1 + (1700\,x2^2 - 16.52E(6)X2$$
$$+1.6E(11) + (1700X2 - 33.03E(6))u1)\,u2 + (x2^3(5.3x1 - 1.58x1^2)\,u3 + (16.52E(6)\,x2^2 -$$
$$-1700\,x2^3 - 6.3x1\,x2^3),$$

$$\dot{X}_2 = g2(x,t).\frac{1}{x2^3}((-1700\,x2^2 - 1700x2u1 + 1700u1u2)u1 + (1700\,x2^2 - 16.52E(6)X2$$
$$+1.6E(11) + (1700X2 - 33.03E(6))u1)u2 + (x2^3(5.3x1 - 1.58x1^2))u3 + (16.52E(6)\,x2^2 -$$
$$-1700\,x2^3 - 6.3x1\,x2^3),$$

$$\dot{X}_2 = \frac{1}{x2^3}((-1700\,x2^2 - 1700x2u1 + 1700u1u2)u1 + (1700\,x2^2 - 16.52E(6)X2 +$$
$$+1.6E(11) + (1700X2 - 33.03E(6))u1)u2 + (x2^3(5.3x1 - 1.58x1^2))u3 + (16.52E(6)\,x2^2 -$$
$$-1700\,x2^3 - 6.3x1\,x2^3),$$

$$where: \qquad g1(x,t) = \frac{R(X2+273).X1.(\ln(X1)-C1)^2}{VG.X1.(\ln(X1)-c1)^2 + C2RX3}\,,$$
$$g2(x,t) = \frac{-C2R(X2+273)}{VG.X1.(\ln(X1)-c1)^2 + C2RX3}; \tag{3-37}$$

It can be seen that in this case the D matrix is not zero.

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \alpha \\ 0 & 0 & 0 \end{bmatrix}, \qquad \text{where: } \alpha = 1.579x_1^2 - 5.315x1\ . \tag{3-38}$$

Second I have to determine the Differential Order (d_i):

- $\underline{D}_i(\underline{x},t) \neq 0$ for i=2 → (**d2=0**) ; (3-39.a)
- $\underline{D}_i(\underline{x},t) = 0$ for i=1,3 → ($d1 \neq 0$, $d3 \neq 0$); (3-39.b)

Therefore the algorithm for i =1 and i =3:

a) For i=1:

- For k=0 $\rightarrow N_A^0 C_1(\underline{x},t) = C_1(\underline{x},t)$.

- For k=1 $\rightarrow N_A^1 C_1(\underline{x},t) = \dfrac{\partial}{\partial t} N_A^0 C_1(\underline{x},t) + \left\{ \dfrac{\partial}{\partial \underline{x}} N_A^0 C_1(\underline{x},t) \right\} \underline{A}(\underline{x},t)$.

  where: $\dfrac{\partial}{\partial t} N_A^0 C_1(\underline{x},t) \neq 0$, and $\dfrac{\partial}{\partial \underline{x}} N_A^0 C_1(\underline{x},t) \neq 0$

  $\Rightarrow N_A^1 C_1(\underline{x},t) \neq 0$ ==>$\underline{D}_1 \neq 0$,=>$\mathbf{d_1=1}$. (3-40)

b) For i=3:

- For k=0 $\rightarrow N_A^0 C_3(\underline{x},t) = C_3(\underline{x},t)$.

- For k=1 $\rightarrow N_A^1 C_3(\underline{x},t) = \dfrac{\partial}{\partial t} N_A^0 C_3(\underline{x},t) + \left\{ \dfrac{\partial}{\partial \underline{x}} N_A^0 C_3(\underline{x},t) \right\} \underline{A}(\underline{x},t)$.

  From the same reason as in (3-30) => D3 $\neq 0$, so $\mathbf{d_3=1}$. (3-41)

For the i-th output I use the equations (3-25) and (3-26) to calculate the $C_i^*(\underline{x},t)$, $D_i^*(x,t)$:

- For i=2:

  $C_2^*(\underline{x},t) = [6.29\ \mathrm{x1}]$; (3-42)

  $D_2^*(x,t) = [0 \quad 0 \quad \alpha\ ]$;

- For i=1:

  $C_1^*(\underline{x},t) = N_A^1 C_1(\underline{x},t)$; (3-43)

  $D_1^*(\underline{x},t) = \left\{ \dfrac{\partial}{\partial \underline{x}} N_A^1 C1(\underline{x},t) \right\} \underline{B}(\underline{x},t)$;

- For i=3:

  $C_3^*(\underline{x},t) = N_A^1 C3(\underline{x},t)$ (3-44)

  $D_3^*(\underline{x},t) = \left\{ \dfrac{\partial}{\partial \underline{x}} N_A^1 C_3(\underline{x},t) \right\} \underline{B}(\underline{x},t)$;

The derivative of the outputs due to (d_1=1, d2=0, d3=1):

$$\dot{y}_1 = g2(x,t).\frac{1}{x2^3}((-1700 x2^2 - 1700 x2 u1 + 1700 u1 u2)\, u1 + (1700 x2^2 - 16.52E(6) X2 + 1.6E(11)) +$$
$$+ (1700 X2 - -33.03E(6)) u1) u2 + (x2^3 (5.3 x1 - 1.58 x1^2)) u3 + (16.52E(6)\, x2^2 - 1700 x2^3 - 6.3 x1\, x2^3)),$$

$$y2 = (1.579 x_1^2 - 5.315 x1)\, u3 + 6.29 x1,$$

$$\dot{y}_3 = g1(x,t).\frac{1}{x2^3}((-1700 x2^2 - 1700 x2 u1 + 1700 u1 u2)\, u1 + (1700 x2^2 - 16.52E(6) X2 + 1.6E(11)) +$$
$$+ (1700 X2 - -33.03E(6)) u1)\, u2 + (x2^3 (5.3 x1 - 1.58 x1^2)) u3 + (16.52E(6)\, x2^2 - 1700 x2^3 - 6.3 x1\, x2^3)),$$

$$where: \quad g1(x,t) = \frac{R(X2+273).X1.(\ln(X1)-C1)^2}{VG.X1.(\ln(X1)-c1)^2 + C2RX3},$$
$$g2(x,t) = \frac{-C2R(X2+273)}{VG.X1.(\ln(X1)-c1)^2 + C2RX3}. \tag{3-45}$$

Therefore I obtain:

$$C^* = \begin{bmatrix} g2(x,t).\frac{1}{x2^3}\left(16.52E(6) x2^2 - 1700 x2^3 - 6.3 x1 x2^3\right) \\ 6.29\text{ x1} \\ g1(x,t).\frac{1}{x2^3}\left(16.52E(6) x2^2 - 1700 x2^3 - 6.3 x1 x2^3\right) \end{bmatrix}, \tag{3-46}$$

$$D^* = \begin{bmatrix} \frac{g2(x,t)\circ d1}{X2^3} & \frac{g2(x,t)\circ d2}{X2^3} & \frac{g2(x,t)\circ d3}{X2^3} \\ 0 & 0 & \left(1.579 x1^2 - 5.315 x1\right) \\ \frac{g1(x,t)\circ d1}{X2^3} & \frac{g1(x,t)\circ d2}{X2^3} & \frac{g1(x,t)\circ d3}{X2^3} \end{bmatrix}. \tag{3-47}$$

Because columns 1 and 2 of the D^* matrix are proportional, which means that the determinant of $\underline{D}^*=0$, so then the inverse matrix (D^{*-1}) does not exist.

Conclusion for this method: The method could not be used, only when (D^{*-1}) exist, practically when d_i are the same. As a result for this method the decoupling couldn't be made.

3.3.3 PROVING THE CONCLUSION, USING THE RESULTS FOR THREE TANK SYSTEM (3TS) [7]

The general diagram of the three-tank system (3TS) is given by Fig.3-6.

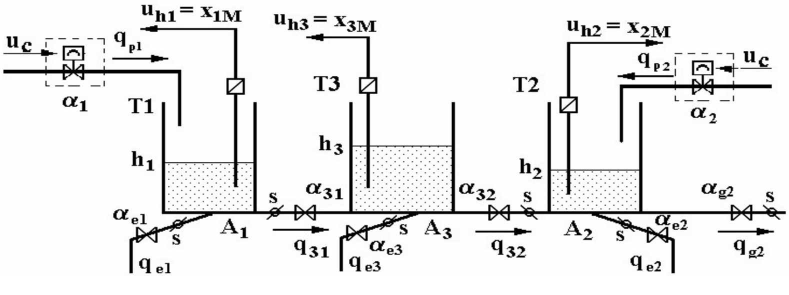


Fig. 3-6. The general diagram of the 3TS.

In the figure the variables are as follows:

- u_c – input voltage of the pumps;
- q_{p1}, q_{p2} – inflow of the pumps (the debit);
- q_{e1}, q_{e2}, q_{e3}, - outflow debit of the pumps;
- q_{13}, q_{32}, q_{g2} – valves debit;
- α_i – constants, describing the position of the valves ($\alpha_i \in [0,1]$);
- A_i – section of the tanks;

The nonlinear model of the 3TS is then, as follow [5], [7]:

T1: $$\dot{h}_1 = \frac{1}{A}\left[q_{p1} - q_{13} - q_{e1}\right], \qquad (3\text{-}48.1)$$

T2: $$\dot{h}_2 = \frac{1}{A}\left[q_{p2} + q_{32} - q_{e2} - q_{g2}\right], \qquad (3\text{-}48.2)$$

T3: $$\dot{h}_3 = \frac{1}{A}\left[q_{13} - q_{32} - q_{e3}\right], \qquad (3\text{-}48.3)$$

$$y_1 = h_1, \qquad (3\text{-}48.4)$$

$$y_2 = h_2. \qquad (3\text{-}48.5)$$

with other words[2],[7]:

$$\dot{h}_1 = -\frac{1}{A}u_{s1}\cdot \operatorname{sgn}(h_1 - h_3)\cdot\sqrt{2g|h_1 - h_3|} - \frac{1}{A}u_{e1}\sqrt{2gh_1} + \frac{1}{A}c_1u_{c1} \quad , \qquad (3\text{-}49)$$

$$\dot{h}_2 = \frac{1}{A}u_{s1}\cdot \operatorname{sgn}(h_3 - h_2)\cdot\sqrt{2g|h_3 - h_2|} - \frac{1}{A}u_{e2}\sqrt{2gh_2} + \frac{1}{A}u_{g2}\sqrt{2gh_2} + \frac{1}{A}c_2u_{c2} \quad , \qquad (3\text{-}50)$$

$$\dot{h}_3 = \frac{1}{A}u_{s1}\cdot \operatorname{sgn}(h_1 - h_3)\cdot\sqrt{2g|h_1 - h_3|} - \frac{1}{A}u_{s2}\operatorname{sgn}(h_3 - h_2)\sqrt{2g|h_3 - h_2|} - \frac{1}{A}u_{e3}\sqrt{2gh_3}, \qquad (3\text{-}51)$$

$$y_1 = h_1, \qquad (3\text{-}52)$$

$$y_2 = h_2. \qquad (3\text{-}53)$$

Now, trying to apply the non-linear decoupling algorithm, in this situation:

$$\underline{D}(\underline{x},t) = 0, \underline{u} = [u_1 \quad u_2] = [q_{p1} \quad q_{p2}], \qquad \Rightarrow \mathbf{d_i > 0}, \text{ i.e. } \mathbf{d_1 = d_2 = 1} \tag{3-54}$$

Calculating $\underline{C}^*$ and $\underline{D}^*$:

$$\begin{cases} C_i^*(\underline{h},t) = N_A^1 C_i(\underline{h},t) \\ D_i^*(\underline{h},t) = \left[\dfrac{\partial}{\partial \underline{x}} N_A^1 C_i(\underline{h},t)\right] \underline{B}(\underline{h},t) \end{cases}, \qquad i = \{1, 2\}, \tag{3-55}$$

I obtain that:

$$\begin{cases} \underline{C}^*(\underline{h},t) = \dfrac{1}{A}\begin{bmatrix} -q_{13} - q_{e1} \\ q_{32} - q_{e2} - q_{g2} \end{bmatrix} \\ \underline{D}^*(\underline{h},t) = \dfrac{1}{A}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{cases}. \tag{3-56}$$

Because D^{-1} exists the non-linear decoupling method can be used:

$$\underline{F}_1(\underline{h},t) = -A\underline{C}^*(\underline{h},t) = \begin{bmatrix} q_{13} + q_{e1} \\ -q_{32} + q_{e2} + q_{g2} \end{bmatrix}, \tag{3-57}$$

$$\underline{G}(\underline{h},t) = \underline{D}^{*-1}(\underline{h},t) \cdot L = A\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \tag{3-58}$$

But $d_1 = d_2 = 1$, so the poles of the system could freely chosen. As a result:

$$\underline{M}^*(\underline{h},t) = \begin{bmatrix} \alpha_{01} h_1 \\ \alpha_{02} h_2 \end{bmatrix}, \tag{3-59}$$

where α_{01} and α_{02} are chosen freely.Conseguently:

$$\underline{F}_2(\underline{h},t) = -\underline{D}^{*-1}(\underline{h},t)\underline{M}^*(\underline{h},t) = -\begin{bmatrix} \alpha_{01} h_1 \\ \alpha_{02} h_2 \end{bmatrix}, \tag{3-60}$$

As a result, the **decoupling equation** will be:

$$\dot{h}_i = -\alpha_{0i} h_i + \lambda_i w_i, \qquad i = 1, 2. \tag{3-61}$$

In conclusion, the transfer function for each decoupled way will be:

$$H_i(s) = \frac{h_i}{w_i} = \frac{\lambda_i}{s + \alpha_{oi}} \quad , \qquad i = 1, 2. \qquad (3\text{-}62)$$

where the gain is: $\frac{\lambda_i}{\alpha_{0i}}$, and the time constant is: $\frac{1}{\alpha_{0i}}$ [sec].

The pole is α_{0i}, and it is positive if $\lambda_i > 0$.

Remark: For steady state values h_{1o}=50 cm h_{2o}=10 cm h_{3o}=30 cm a simulation with a properly designed controllers[7] are shown by Fig. 3-7.and Fig. 3-8.for the decoupled system.

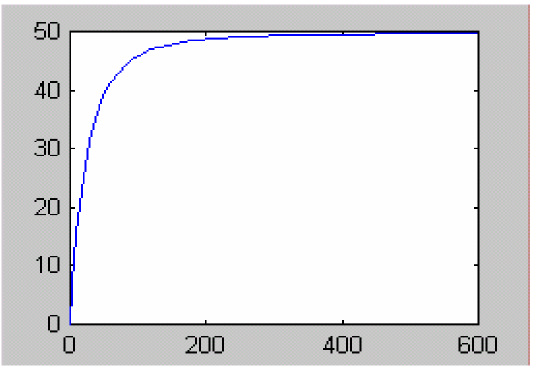


Fig. 3-7. Output response for h_{10} = 50 cm

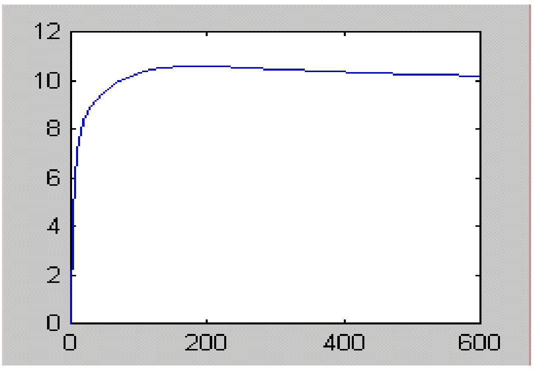


Fig. 3-8. Output response for h_{20} = 10 cm

SUMMARY:

- The system decoupling using the P-canonical form is very easy because we uses the liniarized model of the real system, and this method has shown the quality parameters (overshoot & settling time) can be reconsidered by different controllers designing methods.
- The coupled system has shown that the quality parameters are ''in contradiction '', it can not be obtained separately an optimum value. The dealing problem is the same as for SISO system.
- The non-linear decoupling method is very difficult and applying this method for the MIMO boiling system has shown that the system could not be decoupled because the inverse matrix (D^{*-1}) does not exist.

- The non-linear decoupling method could be used only when (D^{*-1}) exist, practically when the Differential Order (di) are the same, and we proved this using results of three- tank (3TS) system.
- However a non-linear decoupling method is better to use (if it is possible) because it is working directly with the non-linear mathematical model. Therefore one will obtain better results.

CHAPTER 4

PRACTICAL REALIZATION OF THE BOILING SYSTEM USING THE DELTA V SOFTWARE

The practical work is shared with my college (Rammah Mohamed Abohtyra); the practical work refers on the control of a boiling system using the Delta V industrial control software. The boiling system is a real laboratory model, which contains already an implemented PID controller. The aim was to replace the existing controller by an optimal one.

4.1 GENERAL PRESENTATION OF THE DELTA V SOFTWARE [14]

The Delta V Today's technologies deliver lower engineering, installation, and startup costs, and big ongoing operations and maintenance savings. The Delta V software, namely its graphical interface, Control Studio, Recipe Studio, Process History view, Delta V Operate, Alarm Studio, User Manger, etc., helps the engineers to control the equipments of a system distantly. This means the user can command the work of the whole process from one room, such as:

- Uses to control the equipments the Delta V controller;
- With Delta V system we can view and modify the process parameters in any time;
- The Delta V system software includes a variety of applications to help configuring, operating, documenting, and optimizing our process. The primary applications are categorized as Engineering Tools and Operator Tools, additional tools are available for Advanced Control, Installation.

4.1.1 THE DELTA V LIBRARY

Contains templates for function blocks, composites, and modules, this object may also contain device information and batch definitions, depending on a system preferences.

- **FIELDBUS DEVICES**

Contains the field bus device definitions, a device definition describes the capabilities of a field bus device for the Delta V Explorer.

- ABB Instrumentation (Device Manufacturer):

  Contains the device definitions for the devices from the ABB manufacturer;
- Device Manufacture:

  Contains the device definitions for the devices from a specified manufacturer.

- **FUNCTION BLOCK TEMPLATES**

Contains the Delta V library of function blocks, one can change the parameter values in the library function blocks. The parameter values set in the Delta V Explorer determine the default parameter values for the blocks. When one use a block in Control Studio, its initial parameter values are defined in the library block.

The Function Block Templates contains:

- Advanced control function block;
- Analog control function block;
- IO. Function block;
- Logical function block, it contains two categories: Math and Timer counter.

- **COMPOSITE TEMPLATES**

Contains composite templates that can be used throughout the system. A *composite block* is a function block that contains the functionality of more than one other blocks. One can add a new composite block with this template to any module within Control Studio by dragging the custom block from the palette to the diagram.

- **MODULE TEMPLATES**

Contains modules that one can copy and modify as follows:

- Analog Control;
- IO Redundancy;
- Monitoring;
- Motors-2State;
- Motors-3State;
- Simulation;
- Valves-Normally Closed;
- Valves-Normally Open;

4.1.2 SYSTEM CONFIGURATION

Contains the control modules in the Delta V system, the hardware defined for a system, and system-wide definitions such as alarm types and licenses.

- **Receipts**

The hierarchical list of receipts defined for this database. The *hierarchy* includes operations, procedures and unit procedures for batch control, while receipt is a set of information that uniquely identifies the ingredients, the quantities of ingredients, and the production equipment required to manufacture a product.

- **Setup**

Contains objects that have a system-wide effect, as follows:

- **Alarm Preferences:**

  Contains alarm priorities and alarm types. *Alarm priorities and types* define alarm characteristics that the system can be shared. Each alarm in the system has both a priority and type.

- **Security:**

Displays the parameter and field security icons in the right panel, one can edit the properties of these components to control user access to parameters and also fields in one’s control strategies.

- **Named Sets:**

Displays the named sets available in the system, defines a set of states that can be used as inputs to parameters in a function block. Each state has a name and a corresponding number.

- **Licenses:**

Contains the licenses that have been loaded into the system; to load a license the following steps have to be done:

  - Insert the disk containing the license file into the drive;
  - Click file→Licensing→Load License file;
  - Select the licensing file that want to be loaded;
  - Click Open;
  - Follow the instruction on the Terms and Conditions dialog;
  - The license is loaded into the License folder. The license folder is in system Configuration/Setup in the left part of the explorer.

For more information see [3].

4.1.3 CONTROL STRATEGIES

Contains various components that can be used for control, they are either not assigned to an object in the system (in the case of the Unassigned I/O References), or they can be used by multiple objects (in the case of the modules contained in areas, and in the case of the equipment units).

- **Unassigned I/O References**

Contains the device tags referenced in a control strategy that have not been associated with an I/O card channel. One can drag an unassigned device tag to an I/O channel.

We can note that one can also assign Device Tags using the *I/O Configuration* application.

- **External Phases**

Collection of phases defined in the system, which does not reside in a Delta V controller. These are typically supported by the soft phase server.

- **Area**

A logical division of a process control system. Areas typically represent plant locations or main processing functions. Modules are associated with areas based on their function within the system. One can add maximum 99 additional areas. Moreover, one can also add process cells to further organize objects for batch control.

4.1.4 PHYSICAL NETWORK

Contains the controllers, workstations and placeholders of the network. For device, the Delta V Explorer shows information related to the device. For controllers, the Delta V Explorer shows the I/O subsystem as well as any modules assigned to the controller. For workstations, the Delta V Explorer shows the assigned modules as well as the subsystems related to the operator, batch control alarms and events.

- **Decommissioned Controllers**

Contains controllers that do not have an IP address or any downloaded configuration. These controllers are physically connected to the network. One can drag them onto the control network or onto a placeholder controller. This assigns an IP address to them and enables to assign and download modules to them.

- **Control Network**

Contains the addressed controllers, workstations and placeholders on the network. A *placeholder* is a controller configured in the Delta V Explorer's Control Network that is not bound to a physical controller. After a physical controller is available, one can drag the icon for a decommissioned controller onto the placeholder (this is called *commissioning*).

- **CTLR 004006 - Controller**

This is a controller for the Delta V system. This may be a physical controller, a placeholder, or decommissioned controller. The pasts of this can be as follows:

- **Assigned Modules**: Contains the modules assigned to execute in the node. In order to execute a module in a node, one must download the module after it is assigned.
- **Module Reference:** A module that derives all its characteristics from a module in one of the areas in the Control Strategies section of the Delta V Explorer. Changes of the module under an area will affect the characteristics of this module in the database. The module in the controller will need to be downloaded if one changes either this module reference or the original module contained in the area. Visual indicators show the download and assignment status of the module.
- **I/O Subsystem:** Contains the I/O cards or placeholder cards for this controller.
- **H1 (fieldbus) Card (CO1):** An I/O card that interfaces the controller to the fieldbus devices.
- **H1 Port (PO1):** A port on the H1 card, where each fieldbus port can be connected at maximum16 fieldbus devices.

- **Professional PLUS Workstation**

This is the master node for the Delta V system. The Professional PLUS workstation is the host for the configuration database, this icon may have additional visual indicators to show its download or communication status.

4.1.5 CONTROL STUDIO APPLICATION

The Control Studio Application can be used to create, modify, and delete modules and composite templates for a control strategy. Control Studio provides full editing capabilities for control modules. Control modules contain a group of logically related system objects and have a unique tag name. In

general, control modules represent one's process control equipment, such as valves, analog control loops, pumps, agitators and so on. Modules contain the control algorithm, parameters, alarming, and more.
Together with my colleague, Rammah Mohamed Abohtyra, we use the Control Studio to create and modify modules in the Delta V system. In the following I will describe the steps of making an application, step by step. We can start Control Studio from the task bar by clicking Start → Delta V → Engineering → *Control Studio***.** Or, if we are running the Delta V Explorer, the start can be made by clicking on module in an area, then clicking on the right mouse button, and then clicking on Modify.

We can perform many tasks on a module with Control Studio. Typically, we start by defining the module, its algorithm, and then the parameters. After these were set and exist, we can use the parameters for alarming, displays, trending, journaling, and more.

Within Control Studio, one can perform the following tasks:

- Create a new module or composite block;
- Create a module from an existing module;
- Edit the algorithm for a module or composite block;
- Edit the parameters for a module;
- Define alarms for a module or composite block;
- Debug the algorithm for a module;
- Edit expressions in the algorithm;
- Assign the module to a node;
- Download a module.

The Control Studio window is divided into different sections called *views***.**
Fig.4-1. shows Control Studio and the different views.

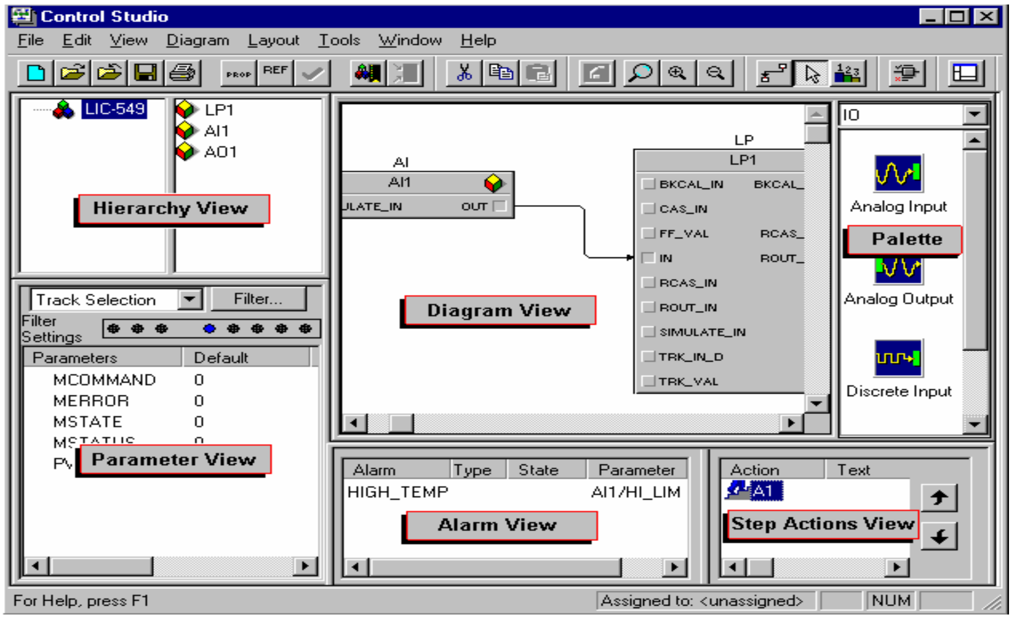


Fig. 4-1. Control Studio and the different views**.**

***Benefits*:**

- The Control Studio On-line graphically displays executing control strategies, so one can modify parameters. After one has installed a module, it can be easily viewed and manipulated to understand its execution. Simply we must "on-line" for a module and view its actual execution in the same graphical representation used to edit the module.
- The Control Studio On-line offers easy online information about operators. One can understand a loop or motor, or even determine why a sequence isn't progressing using Control Studio Online.
- Control Studio On-line can be easily accessed from the Delta V *Operator Interface,* when both are present on the same workstation. Simply select the Control Studio On-line button from the module faceplate.
- It equips one with complete troubleshooting capabilities. One can stop the running, executing a single block or step at a time, set a break point that stops execution at a particular point, or force specific values to override the actual signal all without affecting other control modules that may be running in the same controller.

- The Delta V system uses *FOUNDATION field bus* function blocks, allowing one to view and modify oner executing control strategies in the field bus device, in the Delta V system, or in a combination of both.

4.1.6 RECEIPT STUDIO

Receipt Studio gives one *powerful tools* to create, modify, and troubleshoot receipts. The graphical configuration tools in the Delta V system simplify the overall engineering effort and reduce the development and testing time required.

Even users without computer programming expertise can construct batch control strategies quickly and easily without extensive product training (Fig.4-2.)

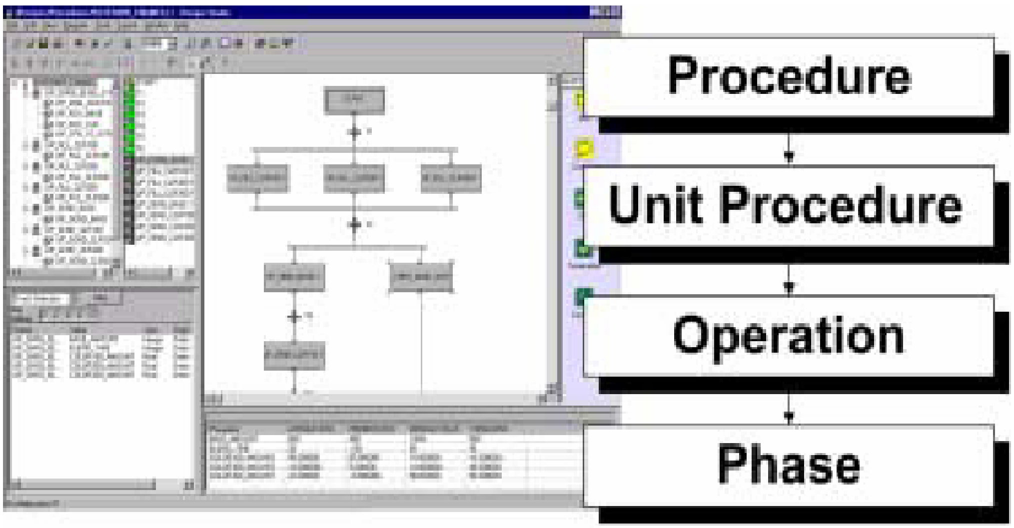


Fig. 4-2. Batch configuration with Receipt Studio.

4.1.7 DELTA V USER MANAGER

The Delta V User Manager lets one to manage the user accounts and define the types of locks that can be used in one's Delta V application. One can:

- Create and manage user accounts;
- Create and manage groups of users;
- Manage (name and describe) locks;

The overall security system involves not only the Delta V User Manager, but also the Delta V Explorer and Control Studio applications. It is in the Delta V Explorer and Control Studio that if somebody assign one's defined locks to specific module parameters and parameter fields.

4.1.8 I/O CONFIGURATION APPLICATION

The I/O Configuration application gives one the ability to view one's I/O channels, device tags, and the module parameters that reference them. The I/O configuration application also allows one to enable and disable multiple channels and edit the properties of I/O reference parameters and channels. It must just starting the I/O Configuration application from Delta V Explorer menus *(Applications → I/O Configuration)* or the Delta V Explorer tool bar by clicking.

4.1.9 DEVELOPING THE CONTROL STRATEGY

A configuration engineer uses a top-down from the (Fig. 4-3) engineering approach to develop the control strategy for a Delta V system. The Delta V system is divided into levels so that the users can choose the level of detail at which they want or need to work. Fig.4-3. shows the levels into which the Delta V system is divided.

Typically, the configuration engineer follows the following sequence:

1. Makes high-level decisions that apply to the overall system and plant and uses the Delta V Explorer to define the system characteristics. (The configuration engineer does not need to be concerned initially with lower details).
2. Moves down a level in detail and decides how to logically divide the system into areas. Areas are logical divisions of the process control system. They can be physical plant locations or main processing functions.

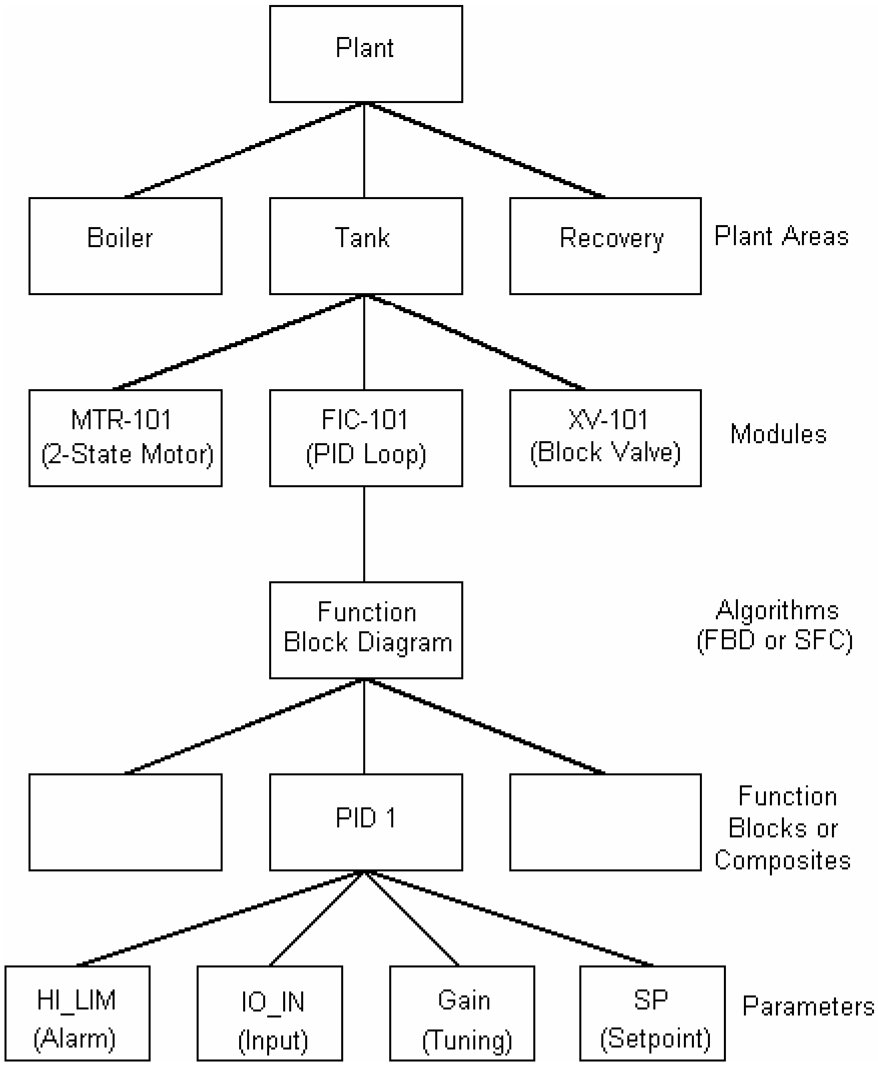


Fig. 4-3. Control Strategy Diagram.

3. Progresses another level and identifies the modules that control the field devices within those areas. The configuration engineer can use the existing modules in the library as starting points for the modules required by the control strategy.

All of the previous steps can be done in the Delta V Explorer. Using the library provided, more than three-fourths of the control strategy can be developed by duplicating existing library modules in the Delta V Explorer. Then, the control strategy for the unique modules is defined using Control Studio**.** In Control Studio, engineers can define and modify the control strategies, cut and paste a large portion of the configuration, and then fill in the details.

Engineers can also decide when to move to the next level of detail. In each level, most of the structure and characteristics for typical control strategies are already configured for the engineer, except for minor details.

4.1.10 THE SYSTEM ALARM MANAGEMENT APPLICATION

The System Alarm Management application lets one to view and work with all alarms within selected areas, units, and modules. This application provides a way to efficiently view multiple alarms, to enable and disable, them and set limits and priorities on multiple alarms.
Most operations require selecting alarms before performing an operation such as enable or disable, set limit values, and so on.

4.1.11 DELTA V OPERATE

The Delta V Operate application (Fig.4-4) contains functions in two modes as follows:

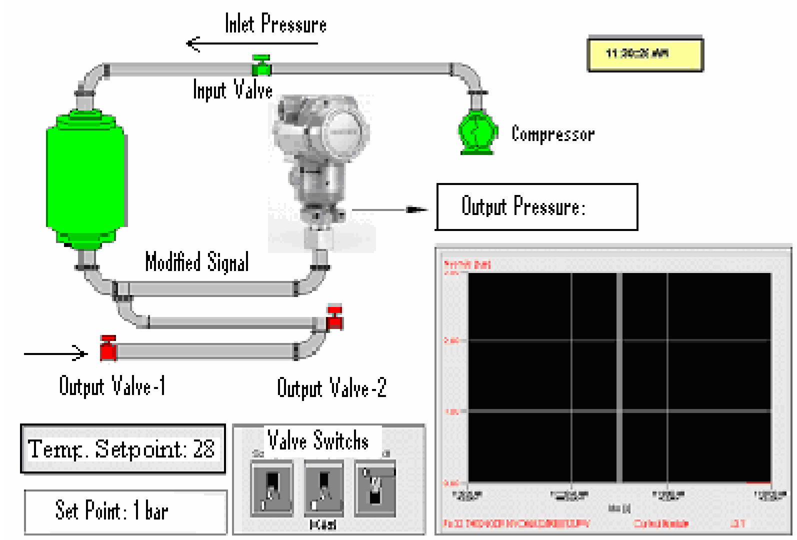


Fig. 4-4. The Delta V Operate application (Configure mode).

- Configure mode - used by engineers to create visual block diagrams (Fig 4-4, upper part). It is used to build high resolution, real-time process graphics. In configure mode, one can incorporate scanned plant images, text, graphics, animation, and sound into the process graphics a predefined desktop template simplifies the typical effort of designing operator displays. This application uses pull-down menus, toolbox buttons, drag-and-drop features, and easy-to-use drawing tools. It also provides sets of dynamos (reusable graphics with animation capabilities) for use in designing operator graphics.
- Run mode - used by operators to run visual block diagrams in the Delta V Operate application. In run mode, control system operators use these graphics in the daily monitoring and maintenance of the process. A standard operating desktop designed specifically for Delta V process control systems providing an easy-to-use, highly reliable operator environment.

The switching between Run and Configure modes either opens the current picture for justify editing (when selected from the menu), or starts Configure mode in the last known state (when switching the short cut key Ctrl-W) can be used.

- To animate the objects select (*Color Select*) tabs to configure the animation properties for the selected object. The ColorButton object is an owner drawn push button (ocx). It is associated with the color selection dialog, which pops up when the user clicks on the button. The dialog allows the user to select a color from a list of colors and display it on the button face.

  The color of the ColorButton can be associated with a color property of an object. By passing on the object's dispatch pointer and to the colorbutton, the user can let the colorbutton to update the property whenever the color is changed.

  The ColorButton object is contained by the ControlContainer object and therefore will inherit the Properties and Methods of the ControlContainer object.
- *Data soures* Provides options that allow you to configure the data source and animation settings for the selected property.

- *Data conversion* Allows you to select the type of data conversion to apply to the data from the source listed in the Data Source field. A data conversion defines how the incoming data should be processed or formatted so that your objects behave according to the properties you have assigned to them. One can apply any of the following data conversion.
    - Range - uses a range of values on which to animate an object.
      Table - attempts to match the incoming value from the data source to an entry in a lookup table. If a match occurs, the selected property is changed to match the output value.
    - Format - provides several options for formatting text data.
    - Object - processes the data received from the data source at its exact value.

4.1.12 DELTA V BATCH HISTORY VIEWS.

Delta V Batch History View is used to view the following batch related documents:

- Batch Events documents: show batch related events in a tabular format;
- Batch Overview documents: show all the batches in the Delta V Batch History Server;
- Batch Detail documents: show detailed information about a single batch.
- Batch Comparison documents: show detailed information about two batches;

4.1.13 DIAGNOSTICS

Delta V diagnostics can be used to:

- Determine if the device is commissioned;
- Check integrity on the H1 card, backup link master device, and ports;
- Check overall port statistics and communication statistics for each device;

Opening Delta V diagnostics and clicking View→Details or View →Compare one can quickly see the device state. If the device is not commissioned, one can open the Delta V Explorer and commission the device. Then, download the port and the device. If the device is commissioned, check integrity on the port and then check port and device communication statistics.

4.2 LABORATORY EQUIPMENT OF THE BOILING SYSTEM (NYOMAS (PRESSURE) APPLICATION)

4.2.1 THE SYSTEM HARDWARE

The system Hardware for a real boiling system is using Delta V software implemented as in the Control Engineering and Information Technology Department, Laboratory of Biomedical Engineering as shown in the Fig 4-5. and Fig. 4-6.

The system hardware consists from the following equipments:

- Delta V workstations (Area-A) using PC technology:
  The PC technology (computer) contains all the package of Delta V Software and the PC technology is connected with the controller package.
- A control network for communication between workstations and PID controller;
- Communications between the I/O subsystem and the control network;
- System Identifier;
- A controller package (Fig.4-6.) consisting of (from left to right):
  - System power supply *(MTL5995 Power Supply);*
  - Controller;
  - Fieldbus System Capacities.

Fig. 4-5. The system Hardware for the boiling system.

Fig. 4-6. System Power Supply (AC/DC), Controller (M3), Discrete out (24 VDC) and FOUNDATION FIELDBUS ™.

The system capacities for fieldbus are as follows:

- 64 field function blocks maximum for each H1 card;
- 25 VCRs maximum for each port on the H1 card;
- 16 devices per segment;

- I/O cards(*I/O Card Keying and Compatible I/O Terminal Blocks*):

  There are two keys on the I/O terminal block and two keys on the I/O card. The keys on the I/O card are set by the factory; you can change the keys on the I/O terminal block to match the corresponding I/O card.

  Then first we enable the channels on the cards and define the Device Tag for each channel. The Device Tags are the names that the Delta V software uses in the control modules to identify the input and output instruments and hardware devices like transmitters, valves, and so on.
- Discrete Output Function Block (*Discreet out 24 VDC*);
- One Transducer and three pneumatic valves *(one for the input and two for the outputs):*

The transducer is a Rosemount 3051S Series of Instrumentation device (Fig.4-6).

Fig.4-6 The Rosemount 3051 Pressure transmitter.

The Rosemount 3051 Pressure transmitter with FOUNDATION field bus is the newest extension to the Rosemount pressure product line. Loaded with the same superior features found in the HART-based 3051, the new fieldbus platform offers even more functionality and user benefits. The 3051 with FOUNDATION fieldbus is the ultimate transmitter for both critical and non-critical pressure applications.

4.2.2 THE SYSTEM SOFTWARE

The NYOMAS (Pressure) model is a real model implemented in the Biomedical Engineering Laboratory of the Control Engineering and Information Technology Depepartment of the Technical University Budapest (Delta V system/Control Studio) as shown in Fig. 4-7.

It contains link algorithms, conditions, alarms, displays, and other characteristics together, usually for a particular piece of equipment.

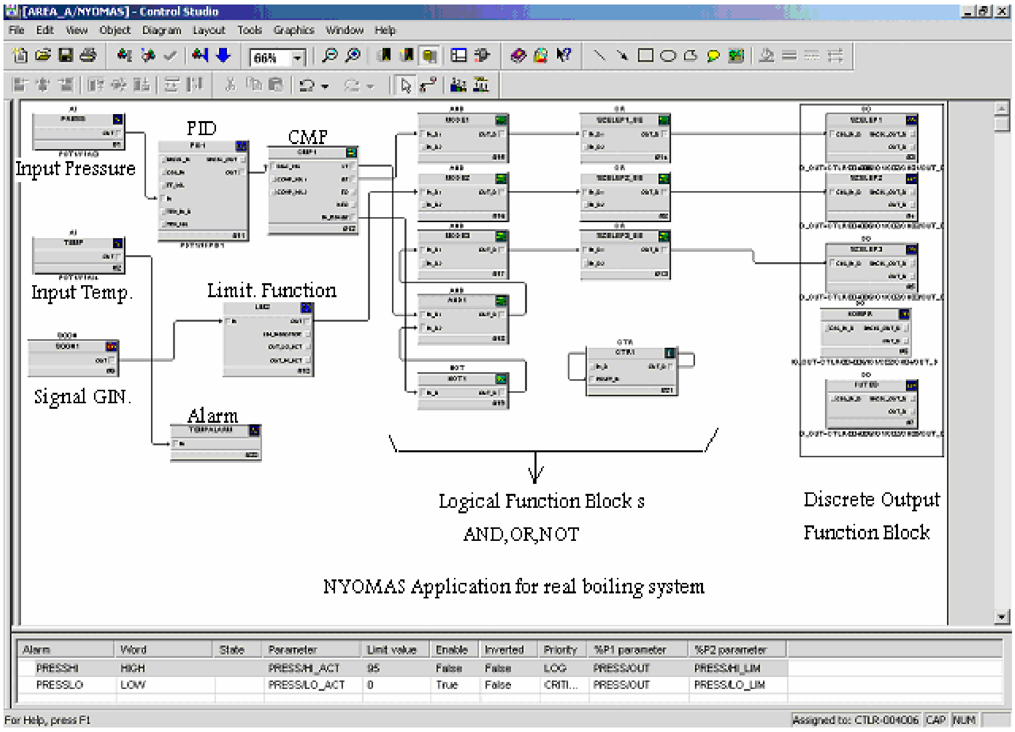


Fig. 4-7. NYOMAS (Pressure) application for the real boiling system.

The description about the function blocks are as follows:

- The Analog Input (AI) function block**:**

  Accesses a single analog measurement value and status from an I/O channel. You can configure the channel type for each I/O channel to be the transmitter’s 4 to 20 mA signal or the digitally communicated primary or non-primary variable from a HART’S transmitter.

  The AI function block supports block alarming, signal scaling, signal filtering, signal status calculation, mode control, and simulation.

- The Signal Generator (SGGN) function block**:**

  Produces an output signal used to simulate a process signal. The block uses a specified combination of a sine wave, a square wave, an input value, a bias value, and a random value to generate the output signal.

- The Limit (LIM) function block**:**

  Limits an input value between two reference values. The block supports signal status propagation. There are no modes or standard alarm detection in the Limit function block.

- The PID function block**:**

  Combines all the necessary logic to perform; analog input channel processing, proportional-integral-derivative (PID) control with the option for nonlinear control (including error-squared and notched gain), and, analog output channel processing. The PID function block supports mode control, signal scaling and limiting, and feed forward control, override tracking, alarm limit detection, and signal status propagation. To support testing, you can enable simulation.

- The Comparator (CMP) function block**:**

  That allows you to compare two values (DISC_VAL and COMP_VAL1) and set a Boolean output based on that comparison for the LT (Less Than), GT (Greater Than), EQ (Equal To), NEQ (Not Equal) outputs.

  Additionally, the Comparator function block compares the DISC_VAL against the range defined by COMP_VAL2 and COMP_VAL1 to determine the Boolean output IN_RANGE.

  There are no modes or standard alarm detection in the Comparator function block.

- The logical function blocks**:**

  Generates a discrete output value based on the logical function of two to sixteen discrete inputs.

- The Discrete Output (DO) function block**:**

  Takes a binary set point and writes it to a specified I/O channel to produce an output signal. You can confirm the physical output operation by configuring a hardware discrete input which produces a value that should match the set point.

4.3 OPTIMIZATION OF THE PID-CONTROLLER FOR THE REAL BOILING SYSTEM

The boiling system is a real laboratory model, which contains already an implemented PID controller. The aim is to replace the existing controller by an optimal one to get better performances.

First, the model of the boiling system has to be identified and compared with those from literature. Then, a simulation model has to be carried out over the identified system and finally a new controller has to be designed and implemented assuring better performances.

4.3.1 IDENTIFICATION OF THE REAL BOILING SYSTEM

The identification must be started by analyzing the inputs, the outputs and the dynamic behavior of the real boiling system with existing controller (Fig. 4.8).

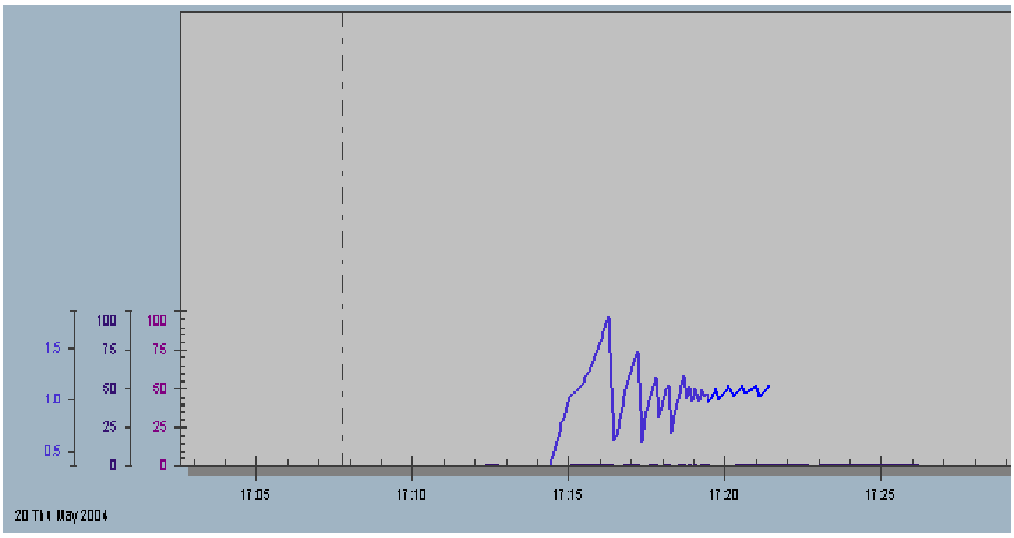


Fig. 4-8. The Boiler Pressure for the real system with Existing Controller with DeltaV.

We have seen that the dynamic behavior of the real boiling system has the following properties:

- ➢ The system response is slow;
- ➢ The output response represents undamped case ($0 < \xi < 1$);

- The overshoot ($M_P \approx 80\%$) and the peak time ($t_P \approx 15$ Sec.) are big;
- From the shape of the oscillations of the output response the order of the plant is greater than two ($n \geq 2$).

Due to these mentioned properties the order of the plant can be determined as third order (one for the integral element and two for second-order system).The dynamic behavior of the second-order system can be described in terms of two parameters (ξ and ω_n), the system is undamped ($0 < \xi < 1$) so the closed-loop poles are complex conjugates, [11]. Then, the transfer functions of the real boiling system is as follows:

$$H_P(s) = K_P \frac{\omega_n^2}{s(s^2 + 2\xi\omega_n s + \omega_n^2)} = \frac{K_P}{s(T^2 s^2 + 2\xi T s + 1)}, \tag{4-1}$$

where: $T = \dfrac{1}{\omega_n}$

Calculating the parameters (ξ and ω_n) from the following relations:

$$M_P = e^{\frac{-\Pi\xi}{\sqrt{(1-\xi^2)}}}, \qquad t_P = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n\sqrt{1-\xi^2}} \quad , \tag{4-2}$$

And substituting the M_P, t_P in the last relation i obtained:

$$\xi = 0.07, \ \omega_n = 1.38 \text{ rad.Sec} \quad \text{and } T = \frac{1}{\omega_n} = 0.724 \ \text{1/rad.Sec} \tag{4-3}$$

we can determine the gain of the plant (K_P) from the input and output of the real plant at steady state:

$$K_P = \frac{output \quad of \quad plant}{input \quad of \quad plant} = \frac{1}{20} = 0.05 \quad . \tag{4-4}$$

Substituting (4-3) and (4-4) in (4-1) the identified model of the real boiling system can be obtained as follows:

$$H_P(s) = \frac{0.05}{s(0.5247s^2 + 2.1655s + 1)} = \frac{0.05}{s(1 + 0.278s)(1 + 1.8875s)} \tag{4-5}$$

4.3.2 THE SIMULATION MODEL OF THE REAL BOILING SYSTEM

Applying the existing PID controller to the identified model (4-5), the block diagram of the closed loop system is illustrated Fig. 4-9, (A.4.7). The output pressure for the real boiling system is shown in the Fig. 4-10.

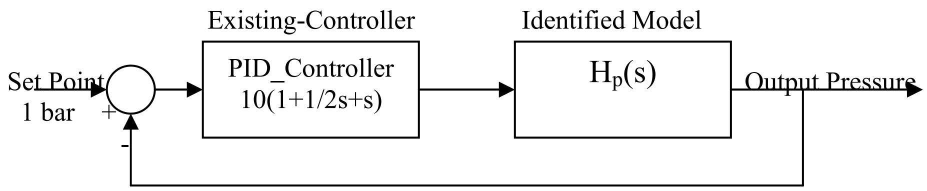


Fig. 4-9. Closed loop for the real boiling system (with the implemented controller)

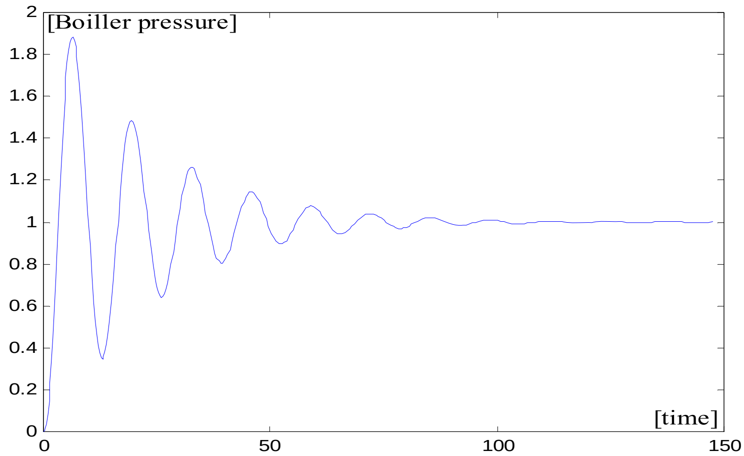


Fig. 4-10. The output pressure for the real boiling system with MATLAB Simulink

As it can seen, the overshoot ($M_P \approx 80\,\%$), peak time ($t_P \approx 15\,\text{Sec.}$) and the output pressure reached the setpoint (1 bar) at settling time ($t_S \approx 100\,\text{Sec.}$), and this is similar to the real boiling response which is implemented in the laboratory.

4.3.3 REDESIGN THE PID CONTROLLER USING THE KESSLER METHOD

As it can be seen from Fig. 4-8 and Fig 4-10., a new controller has to be designed assuring better performances. The identified plant for the real boiling system is:

$$H_P(s) = \frac{0.05}{s(1+0.278s)(1+1.8875s)} \tag{4-6}$$

Starting with the Kessler method relations presented in Table 1-a and 1-b

(*Chapter (2)*), one can see that the symmetrical criteria (CS) have to be used, while the transfer function of the plant contains an integrator component. Therefore the transfer function of the PID controller can be computed easily, as follows:

$$H_C = \frac{K_r}{s}(1+sT_r)(1+sT_{r1}) \text{, where: } K_r = \frac{1}{8T_\Sigma^2 K_P} \text{ , } T_{r1} = T_1 \text{, } T_r = 4T_\Sigma \quad (4\text{-}7)$$

As a result we obtain:

$$H_C = \frac{27.30}{s}(1+1.8875S)(1+1.1127S) \quad . \qquad (4\text{-}8)$$

Reading the obtained relation the optimal PID controller for the real boiling system will be:

$$H_C(s) = K_P(1+\frac{1}{T_i s}+T_D s) \text{ where: } K_P = 39, T_i = 3 \text{ Sec. and } T_D = 0.7 \text{ Sec.}$$

(4-9)

4.3.4 IMPLEMENTING THE OPTIMIZED CONTROLLER FOR THE REAL BOILING SYSTEM

The newly designed controller has to be built on the real boiling system controlled by Delta V software. To reach that, the obtained simulation result are corresponding also for the real system. The obtained controller first we tested in MATLAB Simulink (A.4.8): The result is presented by Fig. 4-11.

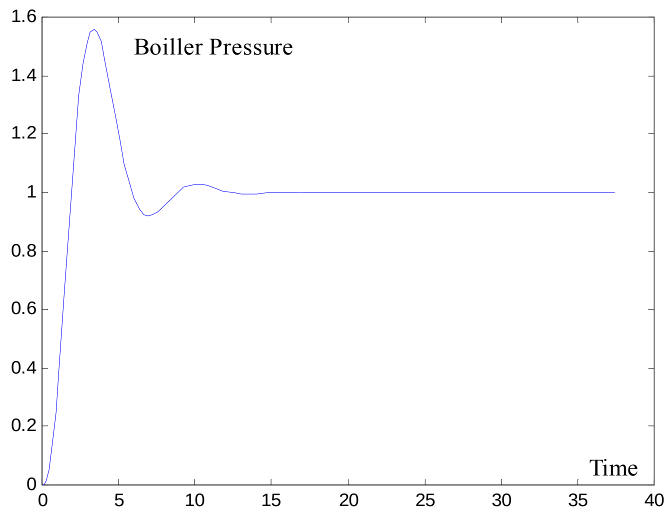


Fig. 4-11 The output pressure for the real boiling system

using the optimal PID controller with MATLAB Simulink.

Finally, the newly designed controller was implemented on the real boiling system controlled by Delta V software (replacing the existing controller by the optimal one) and using the Delta V Operate and Process History View we can show the boiler pressure as in the Fig. 4-12.

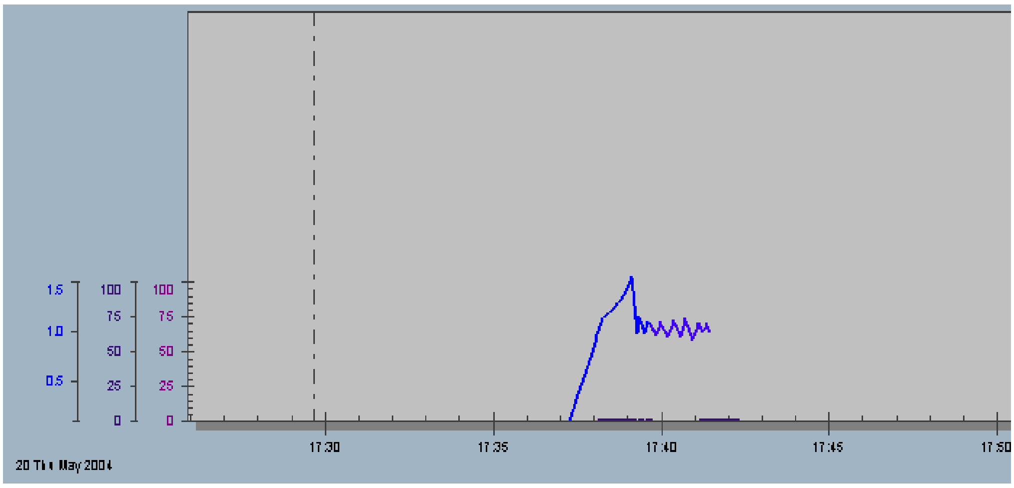


Fig. 4-12 The output pressure for the real boiling system using the optimal PID controller with DeltaV.

As it can seen the output pressure response is similar to the real boiling system and is also reaching the set point (1 bar) at settling time space 15 Sec., while the overshoot and the oscillation is reduced. As a result it is proved that this controller assumes better performances.

SUMMARY

The boiling system is a real laboratory model, which contains already an implemented PID controller. The aim was to replace the existing controller by an optimal one to get better performances.

The work was done as the follows:

- First the model of the boiling system was identified. The identification started by analyzing the inputs and the outputs and the dynamic behavior of the real boiling system:
  - The system response is slow;
  - The output response is undamped case ($0 < \xi < 1$);

- The overshoot ($M_P \approx 80\%$) and the peak time ($t_P \approx 15$ Sec.) were big;
- The output response has big oscillation then the order of the plant ($n \geq 2$).

Considering the last proprieties and using the overshoot and peak time relations and also determining the gain of the plant the transfer function of the real boiling system was obtained.

- I applied together with my colleague the existing PID-controller to the identified model. I obtained the same big overshoot ($M_P \approx 80\%$), peak time ($t_P \approx 15$ Sec.) and settling time ($t_S \approx 100$ Sec.), it's similar to the real boiling response.
- A new controller was designed using the Kessler method for assuring better performances.
- The newly designed controller was implemented built on the real boiling system controlled by Delta V software. (replacing the existing controller by the optimal one) As I replaced, the output pressure response was similar to the real boiling system and reached the set point (1 bar) at settling time 15 Sec., the overshoot and the oscillation was reduced, so better performances were obtained.

CHAPTER 5

CONCLUSIONS

My task was, first to study the model of a continuous flow boiling system (SISO system and MIMO system approach). Finally, the practical work realized a control solution on the boiling system using the Delta V industrial control software.

The non-linear mathematical model of a continuous flow boiling system has been developed employing the mass and heat balance equations. I took the non-linear functions of the boiling system, and expanding them in Taylor series expansions around the steady state operating point, the linearized mathematical model was obtained. The results were presented using MATLAB-Simulink simulations. I have demonstrated based on the presented results the necessity of a controller.

The control of the continuous flow boiling system in *SISO approach* was presented, using PI controllers designed with the Kessler method, the critical gain method (Ziegler-Nichols) and frequency domain methods. Then, I compared the three methods from the point of quality parameters (settling time, overshoot).

The control of continuous flow boiling system in *MIMO approach* was presented using PI controllers designed first for the decoupled system built in the P-canonical form, and then using a non-linear decoupling algorithm. In case of this last one, the evaluation of the results were presented also for a general three thank system (3TS). I have demonstrated that the decoupling method based on the linearized mathematical model is easier, but the non-linear decoupling method gives better results, while it is working directly with the non-linear mathematical model.

The practical work was realized using the Delta V software, where a real laboratory model, namely a boiling system was utilized. The model already contained an implemented PID controller.

The aim of the practical work was to replace the existing controller by an optimal one to get better performances. Therefore, first I identified the model together with my colleague Rammah Mohamed Abohtyra, analyzing the dynamic behavior of the real boiling system. After that, we designed a new controller and implemented the real boiling system controlled by the Delta V software. Finally, we demonstrated that the output pressure response was better then the old controller.

As future plans, it could be interesting to realize the control of the real boiling system using the Recipe Studio of the DeltaV software, and also to design the temperature control and pressure control considering the laboratory application as a MIMO boiling system.

REFERENCES

[1] FADEL M.A.L. “Control of a continuous flow boiling system (SISO system Approach)”, Project Laboratory I, BMGE, 2003.

[2] FADEL M.A.L. “Control of a continuous flow boiling system (MIMO system Approach)”, Project Laboratory II, BMGE, 2003.

[3] FRANKS R.G.E. “Modelling and Simulation in Chemical Engineering”, Wiley- Interscience New York, pp.105, 1972.

[4] WOLFRAM St. “The Mathematica Book”, Cambridge University Press 1999.

[5] Matlab and Simulink User’s Guide.

[6] PREITL St., PRECUP R.E., KOVACS L., PREITL Zs. “Control Solutions for Electrical Driving Systems. Tuning Methodologies for PI and PID Controllers”, "Kandó Conference 2002 – 60 years in Electrical Training" T.U. “Kálman Kandó”, Budapest, Hungary, Nov. 14-15 2002

[7] KOVACS L. “Multivariable Decoupling Possibilities for the Three Tank System”, RODOSZ Conference, Cluj-Napoca, Romania, Apr. 2003.

[8] FREUND E., HOYER H. “Das Prinzip nichtlinearer Systementkopplung mit der Anwendung auf Industrieroboter”, Regelungstechnik 28, pp. 80-126, 1980.

[9] COULSON J. M., RICHARDSON J. F., “Chemical Engineering”, Elsevier Science, Second edition,1993.

[10] LUYBEN W. L. “Process Modeling, Simulation, and Control for Chemical Engineers”, McGraw-Hill Book Co., New York, 1973.

[11] OGATA K. “Modern Control Engineering”, Prentice Hall; Third edition, 1996.

[12] R. Bela “Control Theory lecture notice”, BUT 2002.

[13] E.Freund, H.Hoyer: Das Prinzip nichtlinearer Systementkopplung mit der Anwendungauf Industrieroboter, Regelungstechnik 28, 1980, pp. 80-126.

[14] Delta V Book On Line, (Delta V software Help)

APPENDIXES

A1. Mathematica Program for Identification of Continuous Flow Boiling System.

A2. The MATLAB Program for the Output Response Curves Y_1, Y2 when Considering only one input u1 or u2 or u3.

```
A= [-0.173873 -0.00480453 6.0096E-8; -2.98081   –0.0823552
    1.03E-6; -6.28936 -0.173797 0];

B= [0.00001172 0.004787 0.123679; 0.000295 0.0820623
  2.12;   0.00062272 0.173174 4.47378];

C= [0 1 0; 6.28936 0 0];

D= [0 0 0; 0 0 -4.47378];

% The Output Response Curves Y1, Y2 when the Inputs are u1, u2 and u3
   step (A, B,C,D)
   grid
   title ('Output Step-Response(Y1=T,Y2=vE) Plots: Input =u1,u2,u3')

% The Output Response Curves Y1, Y2 when the Input is u1 (u2=u3=0)
   step (A, B,C,D,1)
   grid
   title ('Output Step-Response(Y1=T,Y2=vE) Plots: Input =u1,(u2=0),(u3=0)')

% The Output Response Curves Y1, Y2 when the Input is u2 (u1=u3=0)
    step (A, B,C,D,2)
   grid
   title ('Output Step-Response(Y1=T,Y2=vE) Plots: Input =u2,(u1=0),(u3=0)')

% The Output Response Curves Y1, Y2 when the Input is u3 (u1=u2=0)
   step (A, B,C,D,3)
   grid
   title ('Output Step-Response(Y1=T,Y2=vE) Plots: Input =u3,(u1=0),(u2=0)')
```

A3. MATLAB Program to Compute the Transfer Function Matrix.

$$[num, den] = ss2tf(A, B, C, D, 1);$$

$$H11 = \frac{y1}{u1} = \frac{0.00115}{3.9S+1}; H21 = \frac{y2}{u1} = \frac{0.000287}{3.9S+1}; H31 = \frac{y3}{u1} = \frac{0.0000456}{3.9S+1};$$

$$[num, den] = ss2tf(A, B, C, D, 2);$$

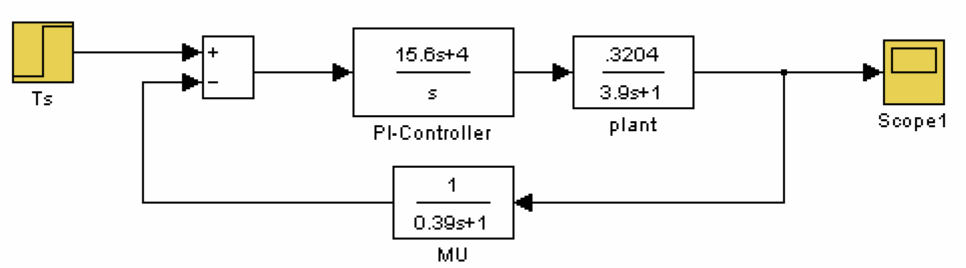
$$H12 = \frac{y1}{u2} = \frac{0.3204}{3.9S+1}; H22 = \frac{y2}{u2} = \frac{0.11748}{3.9S+1}; H32 = \frac{y3}{u2} = \frac{0.0187}{3.9S+1};$$

$$[num, den] = ss2tf(A, B, C, D, 3);$$

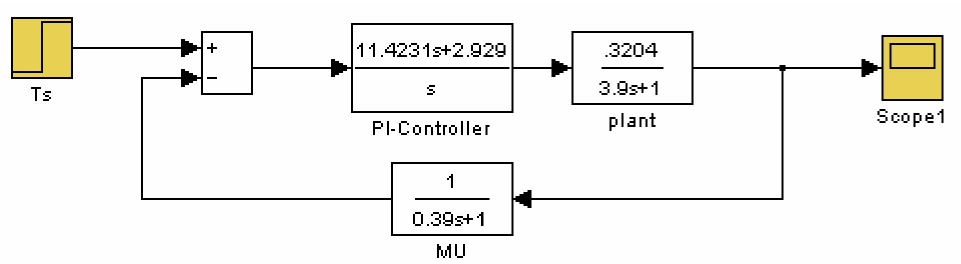
$$H13 = \frac{y1}{u3} = \frac{8.275}{3.9S+1}; H23 = \frac{y2}{u3} = \frac{-17.462S - 1.438}{3.9S+1}; H33 = \frac{y3}{u3} = \frac{0.4828}{3.9S+1};$$

A.4 SIMULINK Models for the Linear Plant

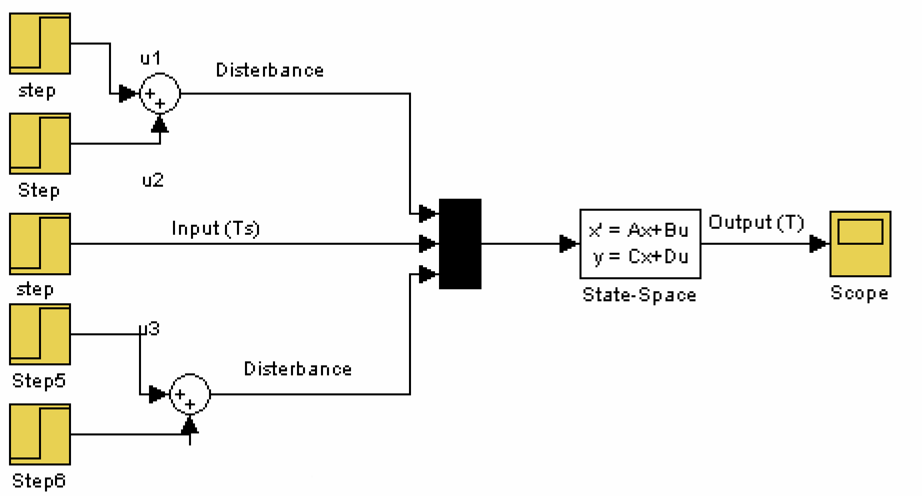
1. Kessler method



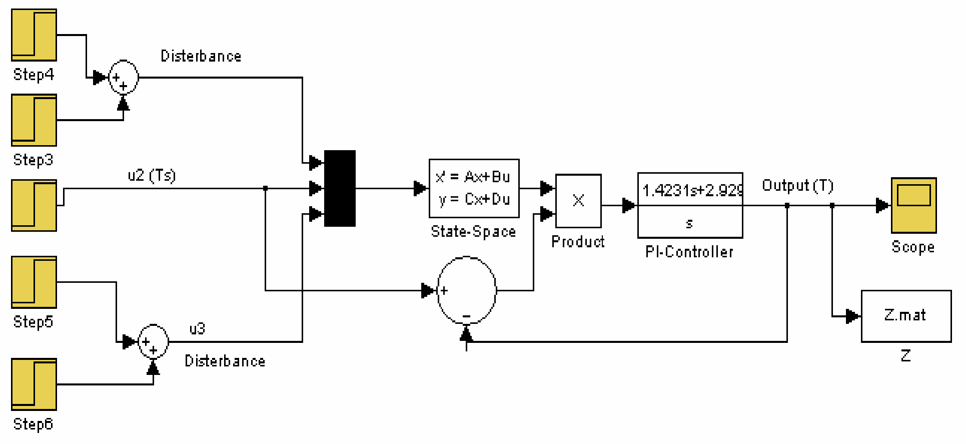
2. Frequency domain design



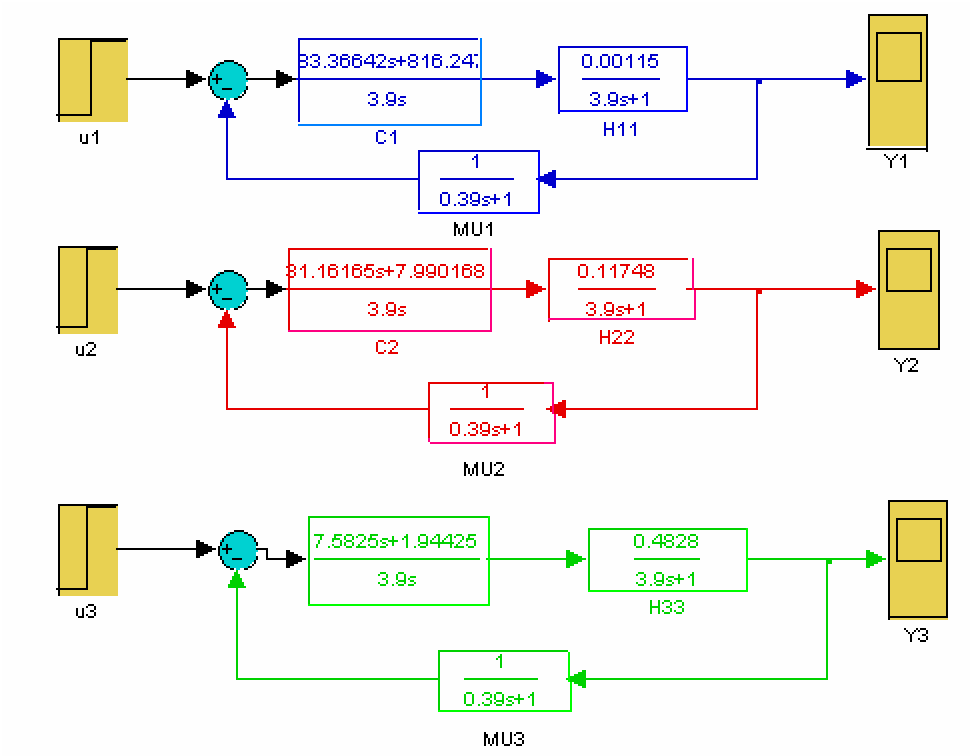
3. Simulink Model for a Continuous Flow Boiling System when Considering the Input signals u1, u_3 are Step disturbances without controller



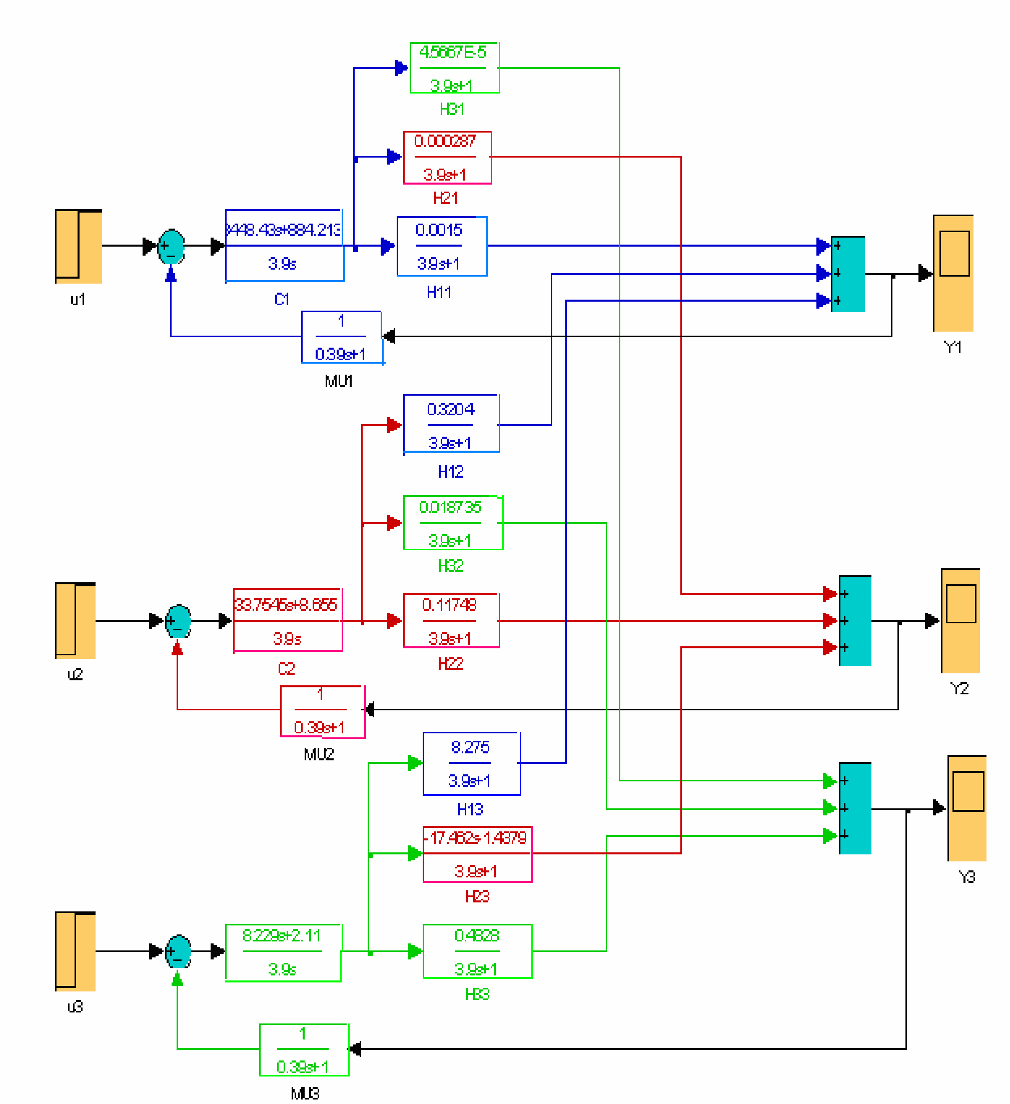
4. Simulink Model for a Continuous Flow Boiling System whenConsidering the Input signals u1, u_3 are Step disturbances with PI- controller



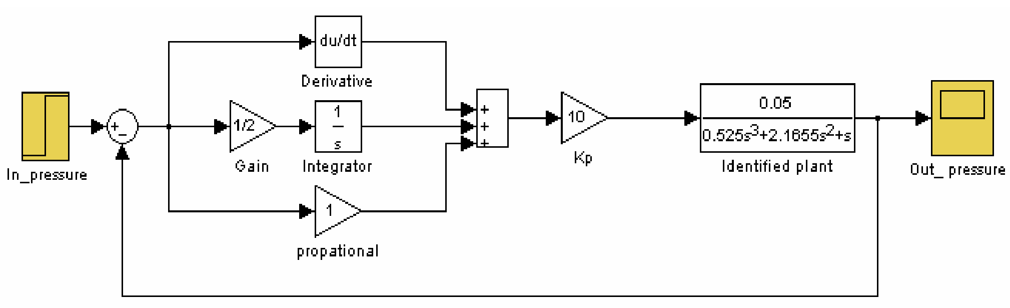
5. Simulink Model of the Decoupled MIMO Boiling System



6. MATLAB-Simulink Model of the Coupled MIMO Boiling System



7. Simulink Model of the Output Pressure for the Real Boiling System with the Existing PID-Controller



8. Simulink Model of the Output Pressure for the Real Boiling System with the Optimal PID-Controller