# MODELLING AND CONTROL OF MULTI PROCESS SYSTEM USING BOND GRAPH AND DECENTRALIZED MODEL PREDICTIVE CONTROLTECHNIQUE

**BY**

**VALENTINE CHIBUEZEIKPO**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**FACULTY OF ENGINEERING,**

**AHMADU BELLO UNIVERSITY, ZARIA**

**NIGERIA.**

**MAY, 2015**

# MODELLING AND CONTROL OF A MULTI PROCESS SYSTEM USING BOND GRAPH AND DECENTRALIZED MODEL PREDICTIVE CONTROLTECHNIQUE.

**BY**

**Valentine Chibueze IKPO**

**B.ENG (FUT.YOLA)2004,**

**MSc/Eng/23415/2012-2013**

*ezeval7@yahoo.com*

**A THESIS SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, ZARIA**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE (M.Sc) DEGREE IN CONTROL ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**FACULTY OF ENGINEERING,**

**AHMADU BELLO UNIVERSITY, ZARIA**

**NIGERIA.**

**MAY, 2015**

# DECLARATION

I IKPO Valentine Chibueze, hereby declare that this Thesis titled "**Modelling and Control of a Multi Process System, Using Bond Graph and Decentralised Model Predictive Control Technique**" has been carried out by me in the Department of Electrical and Computer Engineering of Ahmadu Bello University, Zaria. The information derived from the literature has been duly acknowledged in the text and a list of references provided. No part of this Thesis was previously presented for another degree or diploma at this or any other Institution.

IKPO, Valentine Chibueze ____________________ ____________________

Signature Date

## CERTIFICATION

This Thesis titled "Modelling and Control of Multi Process System,Using Bond Graph and Decentralized Model Predictive Control Techniques" meets the requirement for the award of degree of Masters of Science (M.Sc) in Control Engineering by Ahmadu Bello University, Zaria and it's approved for its contributions to knowledge and literature.

Professor M.B Mu'azu ________________ _________________
(Chairman Supervisory Committee)Signature Date

Dr. T. H. Sikiru ________________ _________________
(Member Supervisory Committee) Signature Date

Professor M.B. Mu'azu ________________ __________________
(Head of Department)Signature Date

Professor A. Z. Hassan ________________ _________________
(Dean Post Graduate School, A.B.U)Signature Date

# DEDICATION

This research is dedicated to all my teachers from the kindergarten to the future and to all hearts that love unconditionally.

**ACKNOWLEDGEMENTS**

I earnestly express profound gratitude to my supervisor and chairman, Professor Muhammed B. Mu'azu whose contribution to this work and impact to my life is immeasurable. His attentiveness and candid interest reformed my approach to things. I am grateful. My warm regards also go to Dr. Sikiru H. Tajudeen, who is a member of the supervisory team of this work, for his attention, prompt and perfect contributions. Thanks to the entire team of highly intellectual and excellence driven academic giants whose contributions shaped this work amongst whom are Professor B. G. Bajoga, Dr. B. Jimoh , Dr. J. Y. Oricha, Dr. S. M. Sani, Dr. S. Garba, Dr. A.M.S. Tekanyi, Dr. A.E. Adedokun, Dr. I. Umoh, Dr. A.D Usman, Dr Y. Jibril, and Dr. K.A Abubilal. I appreciate the contributions of my bosses; Engr. Festus Daudu, Director of Spectrum Management Department, Mr. Kabir Director of Human Resources, Dr. Engr. Malambe Mustafa, Assistant Director of Frequency Coordination and Liaison Services, all from Ministry of Communications Technology, Abuja for their support and contributions. I am highly indebted to all my lecturers Engr. Mu’azu, Engr. A. I. Abdullahi, Engr. Josiah, Engr. Taiwo, Engr. S. Musa, Engr. E. Okafor and Engr. Sae’ed. Often, I had to make use of the Robotics and Embedded System Control System Laboratory, hence the need to acknowledging the patience and cooperation of my friend and colleague Mal. Tukur Lawal, the resident technician.

Also, my appreciations goes to the monks on the control group of this University: Engr. Mustafa, Engr. Husseini, and Ovie, all Ph.D. Students whose support I had benefited from immensely. My appreciations to my colleagues; O. A.

Abdulrahman, Kashim, T. Alonge, Busayo, Sir Husseini, Sir Adamu, B. O. Sadiq, Abubakar Umar, Ndubuisi N., Aliu Daniel, I. Yusuf, Y. Abdulmumuni, Y. A. Basira, I. L. Maryam, and I. Bukola. Their continuous support and contributions toward the success of this research will never be forgotten. May God reward them and strengthen our friendship.

I also sincerely acknowledge the support of Mrs Victoria Godson Ikpo, Mr and Mrs G.O. Ikpo, Mr. and Mrs. Ndahi, Mr and Mrs. Ekea, Mr Victor Ikpo and Family, and Engr. and DR. Onwuamaenyi Amaziro for their supports. I specially dedicate this work in memory of Pa Godson E. Ikpo. Thank you Hagios Pneuma.

## ABSTRACT

This research proposes a simplified approach to the modelling of quadruple tank system via the use of bond graph technique. The novelty of this approach is that quadruple tank system equations are easier to obtain. In addition, the process of obtaining the system equations is less prone to error because of the use of the pictorial advantage of bond graph. This is an advantage for complex systems such as Multi Input Multi Output (MIMO) systems. The 20-SIM software (bond graph technique software) was used in this work. In order to address the inherent multivariable constraint issues of the quadruple tank system such as looping and coupling effects, while minimizing the cost of control, the quadruple tank system was partitioned into Single Input Single Output (SISO) systems. This established the basis for the implementation of the Decentralized Model Predictive Control (DMPC) for the quadruple tank system. Due to the numerical ill-conditioning of the system state equations, a Quadratic Programming (QP) based scheme with inequality constraints was adopted. This was implemented in the Control toolbox of MATLAB. The classical predictive control scheme, achieved control horizon of a numerical value of one, and prediction horizon has values between 1 and fifty with an average settling time of 0.8439 seconds. However, the QPDMPC schemes' control horizon ($M \geq 10$) and prediction horizon ($M \Rightarrow \infty$) accommodating increasing values (infinite horizon) with time while the settling time of 0.04 seconds was attained. Validation of the controllers was carried out relative to classical PID controller.

TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

| Acronyms | Meanings |
|---|---|
| MIMO | Multi Input Multi Output. |
| SISO | Single Input Single Output. |
| SIMO | Single Input Multi Output. |
| MISO | Multi Input Single Output. |
| MPC | Model Predictive Control. |
| DMPC | Decentralized Model Predictive Control. |
| QP-DMPC | Quadratic Programming based Decentralized Model Predictive Control. |
| DMC | Dynamic Matrix Control. |
| LQG | Linear Quadratic Gaussian. |
| LQT | Linear Quadratic Tracking. |
| PID | Proportional Integral and Deferential. |
| TITO | Two Input Two Output. |
| FOPDT | First Order Plus Dead Time. |
| RHP | Right Half Plane. |
| Sf | Source Flow. |
| e | Effort. |
| f | Flow. |
| MSF | Modulated Source Flow. |
| p.e | Point of common Effort. |
| p.f | Point of common Flow. |
| TF | Transformer. |
| RGA | Relative Gain Array. |
| s.t | Such that. |

# CHAPTER ONE

# INTRODUCTION

## 1.1 BACKGROUND

The production of qualitative, desired finished products in the process industries involves a multi processes which includes transformation of raw materials from their chemical or physical form to usable forms. The activity involves energy streams interacting with raw materials to transform the raw materials to the desired finished products(Wolfgang, 2010). Many of the multi processes are predominantly carried out in biological processing units, biochemical, bio-fuels enterprises, cement industries, chemical producing plants, electrochemical industries, glass, ceramics process industries, power generation industries, water industries and tank system storage departments. Constraints like loop interaction, phase shift, instability, plant mismatch, minimum and non-minimum phase behaviours, etc. are often encountered in the industrial processes. Some of these have been studied with the use of the benchmark tank systems.

Continuous stirred tanks, two tanks, three interconnected tanks and quadruple tanks have over the years been used to study the behaviour of these industrial multi process characteristics. The quadruple tank system is used as a benchmark for applications in process control, mainly because of its strong nonlinear behaviour. It is a control of multiple inputs and multiple outputs (MIMO), in which inputs pump voltages from two inputs and outputs are liquid levels of lower tanks. This system presents not only a nonlinear feature, but also an interaction between inputs and outputs linked to the system flow directionality.

Several control techniques to improve the controllability, observability and robust performance of these industrial processes are found in literature. Among the popular ones are Proportional Integral Differential (PID) control, Linear Quadratic Gaussian (LQG), Sliding Mode Control and H-Infinity control. However, Model Predictive Control (MPC) has become a popular advanced control technology implemented in large scale industrial process plants due to its ability to handle  input and state constraints(Necoara & Clipici, 2013). Model Predictive Control is also found to be suited for real time and on-line implementation(Alipouri & Poshtan, 2013; Mahapatro *et al.*, 2014).

## 1.2 MOTIVATION

A serious challenge in the steps for the design and analysis of a mechatronics system is to generate a computer model for control analysis, diagnosis design, sensor selection/positioning, and actuator sizing(Samantaray & Bouamama, 2008). Indeed, mechatronics problems are interdisciplinary engineering systems problems, involving engineering knowledge(mechanical, electrical, pneumatic, thermal.) and various technological components such as sensors, controllers, actuators, and transducers that need to be properly designed and integrated(Paynter, 1970; Wolfgang, 2010). Hence, bond graph, which is a graphical modelling technique, is easily most suitable for unifying the various aspects of mechatronic systems (Paynter, 1970; Wang, 2009). The significance of bond graphing over system equation derivation techniques like variational method, network graphic technique is the ease with which the approach enables hierarchy and well-structured modelling based on a unified language(Paynter, 1970) . The modelling, analysis and control of a multi process system which involves MIMO state variables(Wang, 2009), Tank system requires the use of a robust modelling technique that guarantee accurate modelling of the nonlinear physical phenomenon of the system, as well as an adaptive control mechanism. Efficient control mechanism for quadruple tank system requires an adaptive controller. Model predictive control scheme is well adapted for such problems some of which are in the forms of loop interactions, non-minimum phase zero, minimum phase zero, non-linearity etc.

The overall advantages of an adaptive control scheme to nonlinear control systems are as stated in (Wang, 2009);

(i) The technique must use simple concepts.

(ii) The controller tuning must be physically realisable and yet robust.

(iii) The technique should be usable in either supervisory or primary control modes.

(iv) Automatic constraint setup and handling should be feasible for real time applications(Rosinova & KozĂˇkovĂˇ, 2012) .

The quadruple tank is a benchmark multiprocess setup (besides single tank, two tank, and three tank), which has been found in literatures to exhibit most of the challenges inherent in industrial process production lines(Garg & Tangirala, 2014). Productivity and safety therefore relies on effective handling of the challenges; uncertainty as regards to model mis-match due to errors in modelling, loop interaction, disturbance, noise and instability. Bond graph has been selected for the purpose of modelling the system. This is because the technique is based on the energy characteristics of each constituent component that contribute to the entire system built (Wellstead, 2000). Adding to this is the need to control the system behaviours based on the model. The adaptive controllers such as those based on Model Predictive Control techniques have been used for this purpose. The advantage includes the fact that it is effected based on the very system behavioural equations; hence it tracks the system trajectory naturally.

## 1.3 AIM AND OBJECTIVES

The aim of this research is the modelling of a multi-process system (quadruple tank system), using bond graph technique and its control based on Decentralised Model Predictive Control approach.

The following are the objectives of this work:

1. Development of quadruple tank system model using Bond Graph Method.
2. Extraction/Validation of the system parameter equations from the decoupled causal model using 20-sim
3. Determination of the linear model of the discrete state space format using MATLAB/Simulink control tool box software
4. Design a Decentralised Model Predictive Control based on Robust Quadratic Programming approach.

## 1.4 PROBLEM STATEMENT

A physical system can be described by the use of mathematical description of its dynamic behaviour. Hence, a suitable unifying concept which captures most physical variables and dynamics is energy. Accuracy in the modelling of a nonlinear multi process system that is characterised by multi-stage energy transformation without over-simplification is the most important challenge faced by researchers and hence impacts the system performance subsequently. Perhaps, thinking of a physical system as operating upon a pair of variables whose product is proportional to power makes modelling as close to reality as possible. Bond graph method which incorporates these advantages is used for modelling of the quadruple plant in this work. Furthermore, the control of the multi-process system requires a robust controller algorithm with efficient predictive characteristics and high accuracy.

## 1.5 METHODOLOGY

This section captures the methodological approach adopted in this research.

1. Identify the various elements and energy exchange, ports and storage of the quadruple tank.
2. Map out the bonds and apply causality on them.
3. Validation of the nonlinear model using 20-sim.
4. Model the plant in Simulink
5. Write a code to extract the discrete state space model.
6. Test for controllability and observability of the system.
7. Design a suitable decoupling scheme for the plant.
8. Decouple the system into two Single Input Single Output (SISO) systems.
9. Use Relative Gain Array, to choose the input/output pair with less interaction.
10. Design the DMPC controller for specific systems obtained in (9) and incorporate it in the controlled system derived from (1-9) above.
11. Result analysis and validation using Classical controller PID.

## 1.6 SIGNIFICANT CONTRIBUTIONS

The following are the significant contributions of this research:

1) Development of a Bond Graph model of the quadruple tank system, which has overcome the mathematical complexities of conventional method and which has been validated using 20-sim Simulator.
2) Design of a robust Quadratic Programming Decentralized Model Predictive Controller (QPDMPC) which stabilized and efficiently tracked the system even in the presence of numerical-ill conditions.

## 1.7 THESIS OUTLINE

The general idea for this work is shown in Chapter One. In Chapter Two, the bases and foundation for achieving the aims and objectives of this work was established through the review of fundamentals and related works pertaining to modelling and control of systems, in particular to multi-process systems. The control theory and principles of operation are also reviewed. Furthermore, Chapter Three presents the modelling of the quadruple tank MIMO system via bond graph technique with derivation of system equations. The use of 20-sim to validate the modelled system was also carried out herein. The system decoupler and decoupling design implementation was shown and the SISO systems obtained. While chapter four, discusses the results obtained during simulations and their effects were discussed. Finally, in Chapter Five, the conclusion based on the entire work is presented.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 INTRODUCTION

The review of literature relevant to the research is presented in this chapter. The chapter is divided into two sections: The first dealing with review of fundamental concepts and the other dealing with review of similar previous works.

## 2.2 OVERVIEW OF FUNDAMENTAL CONCEPTS

This section contains the review of concepts fundamental to the research. This covers areas such as Dynamic Modelling, Bond Graph Technique, Model Predictive Control and Decentralized Model Predictive Control.

### 2.2.1 Dynamic Modelling

The dynamic behaviour of a physical system is the outward manifestation of the energy transaction within the system. The possibility of handling a wide range of systems and subsystems in a common framework is done with little or less difficulty. However, derivation of the equations that describe the characteristics of the model of a system physically had been realisable via different techniques. Network modelling technique, which embodies the implementation of Kirchhoff's law of current and voltage had been used in this regards (Bhat & Osting, 2011). The problem formulated involves technical development of interconnected constraints systems (Kim *et al.*, 2011;Sano *et al.*, 2011; Seetharaman *et al.*, 2012). Furthermore, the use of variational method which is based on Hamilton and Lagrange equations derived from first principles had also been in use by researchers. This can be seen in literatures like (Cassolato *et al.*, 2011; Czaban & Lis, 2012; Russer & Russer, 2012; Tian *et al.*, 2013). Hence, the designer needs tools that will simplify and abstract such systems efficiently and reduce errors to the barest minimum. Graphical tools especially bond graph, had in recent past been engaged in modelling of such systems. In the year 1959, Professor H. Paynter at Massachusetts Institute of Technology (MIT) devised the graphical tool bond graph, which is a natural two-dimensional technique based on flow and effort (Energy exchange and rule of conservation) for modelling of physical systems(Borutzky & Cellier, 1996).

### 2.2.2 Multi-Process Systems

Energy can neither be created nor destroyed, but it may be transformed from one form to another which may result in the creation of new products of energy transformation (Mach, 2014). The synergistic utilization and implementation of the above fundamental principle is evident in every process, whether in, electrical, hydraulic and mechanical systems (Heier, 2014). Therefore, industrial processes are best modelled as energy ports with capabilities for power transfer between them (Nacusse & Junco, 2014).

Some specific multi-processes include:

a) Storage systems or Tank Systems.
b) Wireless and fixed cable networks.
c) Fractional distillation and crude oil refining systems.
d) Beverage production plants.
e) Vehicle Manufacturing Systems.
f) Traffic Control Management Systems.

### 2.2.3 Quadruple Tank System

The target is to control the level of non-compressible fluid in the lower two tanks; Tank 1 and Tank 2 and their inlet flow rates as shown in Figure 2.1. The output of each pump is split into two by using a three- way valve. Thus, each pump's output goes to two tanks, one lower for example Tank 1 and another upper, Tank 4, which is diagonally opposite. The ratio of the split up is controlled by the position of the valve. With the change in position of the two valves, the system can be appropriately placed either in the minimum phase or in the non-minimum phase (Garg & Tangirala, 2014). Some challenges arise in this setup, which impact on the normal expected outcome of the flow. These are seen in the form of nonlinearities, possibly overflow of fluid, excessive temperature, pollution (noise, sound, smell, radiation, heat, etc.), instability, pipe worn-out, tearing just to mention but a few. There is the need for a control technology that adequately reduce these effects or possibly completely contend with these uncertainties.

In the design of modelling and control of systems, different combinations of input and output configurations do exist. The input-output configurations may be classified as Single Input Single Output (SISO) systems, Single Input Multi Output (SIMO) systems, Multi Input Single Output (MISO) systems, and Multi Input Multi Output (MIMO) systems (Rosinova & KozĂˇkovĂˇ, 2012). The quadruple tank might be classified as a MIMO system. Figure 2.1, illustrates a typical demonstration of the interconnectivity of the MIMO quadruple tank system. MIMO systems exhibit inherent looping challenges, non-linearity and control difficulties. It is important therefore, to decentralize the MIMO systems into subsystems that their collective control objective is cost effective. The system, therefore can be split into a SISO, SIMO, or MISO subsystems. These subsystems are easily managed and analysed, though the SISO subsystem is the most cost-effective, because the input and output pair is more simplified (Yusuf *et al.*, 2014).

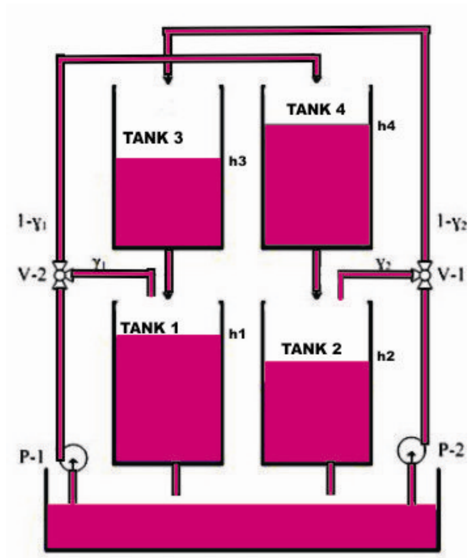



Figure 2.1: The Quadruple Tank System

The system may be described as follows:

(i) Number of inputs: 2, illustrated by Pump 1(P-1) and Pump 2(P-2) .The valve [v-1, v-2] which is a three-way configuration transforms the fluid speed and is represented by [γ1, γ2] ratio characteristics.

(ii) Number of outputs: 2, indicated by the heights, for example, h1 for Tank 1 and h2 for tank 2.

### 2.2.4 Fluid Dynamics Theory

Generally, tank systems hold fluids either as reservoirs or process plants and obey fluid dynamics of Bernoulli's equation (Truesdell, 1984). This equation governs the conservation of mass principles in fluid mechanics. The principle of conservation of mass relation for closed system undergoing a change is expressed as (Saeed *et al*, 2013):

$$\frac{dm_{sys}}{dt} = 0 \tag{2.1}$$

Where in equation(2.1), $dm_{sys}$ denotes change in mass and $dt$ is rate of change in time.

For a control volume (cv) of an open system, the mass balance is expressed in rate form as

conservation of mass:

$$\dot{m}_{in} - \dot{m}_{out} = \frac{dm_{cv}}{dt} \tag{2.2}$$

Where in equation(2.2), $\dot{m}_{in}$ represents total mass flowing in and $\dot{m}_{out}$ is total mass flowing out respectively. However on the right hand side of equation(2.2); $dm_{cv}$ denotes change in mass and $dt$ is rate of change in time of the left part of the same equation.

Hence, for an incompressible fluid, the following assumptions were made during steady state flow (Saeed *et al.*, 2013).

1. Total mass remains the same.
2. Temperature remains constant.

Therefore:

$$\sum \dot{m}_{in} = \sum \dot{m}_{out} \quad (kg/s) \tag{2.3}$$

i.e. for steady state flow (steady flow)

$$\dot{m}_1 = \dot{m}_2 \rightarrow \rho_1 \nu_1 A_1 = \rho_2 \nu_2 A_2 \quad (2.4)$$

Where:

$\dot{m}_i$; is the mass change,

$\rho_i$; is the density of the fluid,

$\nu_i$; is the fluid flow speed,

$A_i$; is the cross-sectional area.

The energy flow of the mechanical system from which the Bernoulli's equation is derived at is stated therefore as(Teixeira *et al.*, 2012):

$$\Delta e_{mech} = \frac{P_2 - P_1}{\rho} + \frac{\nu_2^2 - \nu_1^2}{2} + g(z_2 - z_1) \qquad \text{(Kj/ Kg)} \quad (2.5)$$

Where;

$P_i$; is pressure quantity.

$z_i$; is the elevation of the point above the reference plane point upwards,

g; is the acceleration due to gravity and $i \in \Re^n$.

However the assumption is that the flow is steady;

i. System is not affected by mechanical work.

ii. Friction is zero.

therefore: steady, incompressible flow (along streamline) gives (Teixeira *et al.*, 2012)

$$\frac{P}{\rho} + \frac{\nu^2}{2} + gz = k \quad (2.6)$$

Where $k$ is constant.

### 2.2.5 Controllers

Control system design deals with the problem of making a concrete physical system behave according to certain desired specifications. The controller could be hardware or software based(Maciejowski, 2002). Its primary design aim and application are to achieve an internally stable closed system, make the influence of certain exogenous disturbance inputs on given to-be-controlled output variables as small as possible. Some controllers are designed to impact in feed-forward mode called feed forward controllers e.g. Neuro-fuzzy controllers(Deepa & Arulselvi, 2014; Kumar, 2013)and most heuristic and meta-heuristic intelligent controllers. Also some others are applied in feed-back mode called feedback controllers e.g. Proportional, Integral and Differential Controllers (PID), Linear Quadratic Regulators/Tracking (LQR/LQT), Linear Quadratic Gaussian (LQG) (Halvarsson *et al.*, 2011), $H_2$ and H-infinity ($H_\infty$) optimal controllers and Model Predictive/Adaptive Controllers (Maciejowski, 2002). Control is approached and applied with respect to system specifications. Central control is more efficient when the number of subsystems in an entire system group are few(Morilla *et al.*, 2008). However, the application of decentralized/distributed control strategy is often used when the number of system networks becomes complex and sparse.

### 2.2.6 Decentralized Control

The design of decentralized control, is achieved by cascading control systems consisting of independent feedback controllers; interconnect a subset of the outputs measurements/commands with subset of the manipulated inputs in such a way that the subsets should not be used by any other controller (Raimondo *et al.*, 2007). The controllers do not share information. Some of the advantages that prompted the choice of decentralized control includes(Christofides *et al.*, 2013).

i. The cost associated with obtaining good plant models which are prerequisites for applying multivariable control.
ii. Each controllers is usually tuned one at a time with minimum of modelling effort, and sometimes on-line as the case may require by selecting a few parameters for example gain, and integral constant of a PI-controller.
iii. Modelling effort is compensated to save time.
iv. Operators find it easy to follow and understand.

To be able to tune the controllers independently, it requires that the loop interactions within the system are limited and reduced minimally. Hence, decoupling of the system is carried out.

### 2.2.7 System Decoupling

Decomposition is achieved by successfully decoupling the loops without compromising the controllability of the entire system (Garelli *et al.*, 2006). Literature reveals several attempts to decouple MIMO systems for effective control. One of such methodologies is the application of inverted decoupling which is one of the techniques proposed in (Garelli *et al.*, 2006) and (Husek, 2014). Inverted decoupling, however, exhibits such challenges which necessitates that one of the controllers ought to be manually manipulated. Another of such methodology is ideal decoupling for controller design which was presented by (Jevtović & Mataušek, 2010). The work of (Nordfeldt & Hägglund, 2006) was the decoupling of a MIMO system into subsystems using the ideal decoupling scheme. A decentralised PI/PID controller for TITO (Two Input Two Output) systems, was also researched upon in (Tavakoli *et al.*, 2006), who applied ideal decoupling with time delay incorporated to the First Order Plus Dead Time (FOPDT) subsystems. Subsequently, detuning method was used also in the decoupling of MIMO systems, while incorporating relay auto-tuning as shown in the research presented by (Maghade & Patre, 2012).

Furthermore, (He *et al.*, 2015) presented, a modified adjoint transfer matrix based decoupler was presented. It was carried out by dividing the module suspension control system of a magnetic levitation (maglev) train into two independent SISO control systems.

Hence, the main goal for the use of SISO controller structures for multi-loop control is its simplicity, easy tuning, and ability to meet control objectives (Kirubakaran *et al.*, 2014). This work therefore, attempts at the implementation of an ideal decoupler matrix with decentralised Model Predictive Control (DMPC) for quadruple tank multivariable system control.

### 2.2.8 Right Half Plane Zeros

A zero is defined as a value for which a system, G(s) loses rank and for square system that may be computed as the solution such that:

$$det G(s) = 0 \tag{2.7}$$

Hence, a right half plane zero (RHP zero) transmission zero of G(s) limits the achievable bandwidth of the system (Holt & Morari, 1985). RHP transmission zeros impairs decentralized systems so that it becomes necessary to avoid it or preferably make to lie in the plants direction as a high frequency (Laughlin *et al.*, 1987) as its effect on the system performance in this configuration is meant to be handled as disturbance rejection. If G(s) has a transmission zero at $s = z_k$, while there exist a vector $u_k$ and a scalar $z_k$ such that $u(t) = u_k e^{z_k t}$ results in the output y (t) not containing $e^{z_k t}$ terms. Then, vector $u_k$ is called the zero direction of $z_k$ (Misawa *et al.*, 2014).

Given that:

$$G(s) = M(s)G(s)\,\mathrm{N(s)} \tag{2.8}$$

Such that:

$$G(s) = \begin{pmatrix} \frac{\varepsilon_1(s)}{\Psi_1(s)} & 0 & 0 & 0 \\ 0 & \frac{\varepsilon_2(s)}{\Psi_2(s)} & 0 & 0 \\ 0 & 0 & \frac{\varepsilon_3(s)}{\Psi_3(s)} & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \frac{\varepsilon_i(s)}{\Psi_i(s)} \end{pmatrix} \tag{2.9}$$

Where M(s) and N(s) are unimodular matrices and for each i polynomials $\varepsilon_i(s)$ and $\Psi_i(s)$ are relatively prime (i.e., no common roots) with the additional property (Rodríguez *et al.*, 2014; Russo & Bequette, 2014) that:

$$\varepsilon_1(s) \mid \varepsilon_2(s) \mid \varepsilon_3(s) \mid \cdots \mid \varepsilon_p(s) \tag{2.10}$$

$$\Psi_p(s) \mid \Psi_{p-1}(s) \mid \Psi_{p-2}(s) \mid \cdots \mid \Psi_1(s) \tag{2.11}$$

The transmission zeros of G(s) as $i \geq 1$ are therefore obtained from

$$G(s) = \prod_1^p \varepsilon_i(s) \tag{2.12}$$

### 2.2.9 The Bond Graph Approach

Modelling and simulation of multi-domain systems, including interactions of physical effects from various energy domains, pose formidable new challenges which demand new strategies and techniques (Paynter, 1970). Today's engineering domain is faced with increasing complex systems with evolving challenges whose solution depends on approaches that incorporate and considers all contributing subsystems to the task and cost function (fahime, 2010). For this reason, engineers need modelling and simulation tools which enable a system to be analysed with respect to capabilities, capacities and behaviour without really constructing the system. Some applications and fundamentals of bond graph are found in (Karnopp *et al.*, 2012) and (Borutzky, 2013). So also, Tiernego and Bos, (1985) had used multibond graph for the modelling of mechanical systems. In their work, they showed that the programme can be used to find generalised coordinates experimentally. Research work by (Kayani & Malik, 2008) showed that bond graph technique is a tool that can be used to model systems with different energy domain characteristics while applying genetic algorithm in their work. Also, for concurrent modelling of mechatronic systems, Borusky demonstrated the usefulness of bond graph (Borutzky, 2013). The bond graph approach, which is a graphical modelling tool that takes into account multi-energy exchange systems, domain independent approach (Mechatronics, economics, science, Human behaviour, etc.) and less time consuming, is being applied in the work for modelling of a multi-process (MIMO) via a benchmark Quadruple Tank System,. The successful usage of bond graph in the modelling and simulation of single tank benchmark system was demonstrated by (Aravind *et al.*, 2013). A single conical tank which describes a nonlinear model for level control and real time was designed and simulated in Simulink. Two tank systems have been modelled and simulated by (Deepa & Arulselvi, 2014). Likewise, for three tanks systems, the modelling and control have been cited in literature of recent amongst which are the works of (Triki *et al.*, 2014).

Bond graph methodology is a graphical notation of energy port-based description for modelling dynamic systems (Kayani & Malik, 2008); it is a technique that takes into consideration power and energy exchange on multi-process combination of systems (mechanical, electrical, hydraulic, biological, economics, aerodynamics amongst others). The bond graph methodology clearly and intuitively starts by considering energy flows between the ports of the (conceptual) components of an engineering system(Breedveld, 1985; Wolfgang, 2010). Bond graphs are labelled and have directions based on the direction of effort and flow of power. Each bond has a vertex and an edge; the vertices represent sub-models while the edge stands for the energy connection between power ports(Wang, 2009). This is illustrated in Figure 2.2 (Wang, 2009)

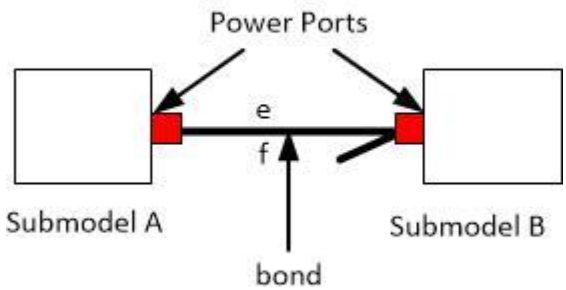



Figure 2.2: Schematic Diagram of Energy

Action and reaction are equal and opposite in time as in this case the flow and effort relates by (Borutzky, 2013)

$$Power = e \times f \tag{2.13}$$

Where in equation(2.13), e is effort and f is flow.

#### *2.2.9.1 Causality*

The output of an effort source (Se) is effort (e), while that of a corresponding flow source (Sf) is flow (f), however, the decision to make which of the component models to reflect the causality is systematic and energy directed. In other words, which of the two power variables is an outgoing signal, or an output variable in one of the constitutive relations. For storage elements so-called integral causality is preferred (Ngwompo, 2011). The power variable that is integrated with respect to time is the input variable. The conjugate power variable, related to the state of the energy store, is taken as output variable. Hence effort is assigned to for a C energy store. Consequently, the causal stroke points away from the port of the C element (Das, 2009). On the other hand, the effort of the power port is taken as an input variable for the dual I element. Hence, the causal stroke points to the port of the I element. The effort is integrated with respect to time. The flow related to the resulting state of the I energy store is its output. If, for a C energy store, the causal stroke is on the side of the power port, or if the casual stroke points away from the port of an energy store, then this means that the output is obtained by differentiation of the conjugate power variable. In this case, so-called derivative causality has been assigned to the port (Wolfgang, 2010).

The resistor has linear constitutive characteristic, causal strokes could be applied in two options. (a) On the side of a port (conductance causality), or (b) Pointing away from the port (resistance causality) as shown in Figures2.3, 2.4 and 2.5 respectively (Wolfgang, 2010).

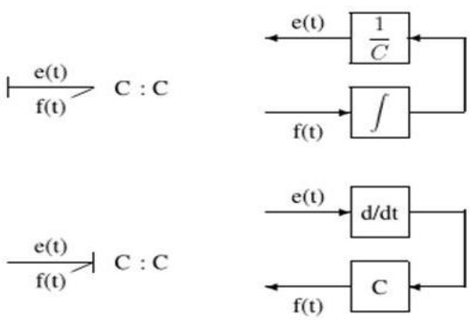



Figure 2.3: C-Capacity Element Causality Integral/Differential

The design of causality for a capacity component is shown in Figure 2.3. The first image demonstrates integral causality while the object underneath show differential causality.

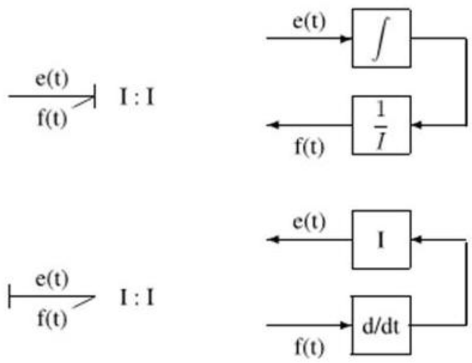



Figure 2.4: I - Inertial Causality Integral/Differential Configuration

For the design of causality of an inertia component, Figure 2.4 shows the two variants. The first image demonstrates integral causality while the object underneath show differential causality.

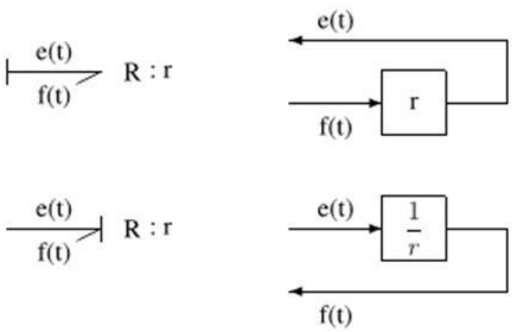


Figure 2.5: R-Resistance Integral/Differential Configuration

For the design of causality of anR-Resistance component of a system, Figure 2.5 shows the two variants. The first image demonstrates integral causality while the object underneath show differential causality

The constitutive equations of the bond graph elements are introduced via examples from the electrical and hydraulic domains. The nature of the constitutive equations lay demands on the causality of the connected bonds (Borutzky, 2013). Bond graph elements are drawn as letter combinations (mnemonic codes) indicating the type of element. The bond graph elements are the following:

(i) **C** - Storage element for a q-type variable, e.g. capacitor (stores charge), spring (stores displacement).

(ii) **I** - Storage element for a p-type variable, e.g. inductor (stores flux linkage), mass (stores momentum).

(iii) **R** - Resistor dissipating free energy, e.g. electric resistor, mechanical friction.

(iv) **Se, Sf** - Sources, e.g. electric mains (voltage source), gravity (force source), pump (flow source).

(v) **TF**- Transformer, e.g. an electric transformer, toothed wheels, lever.

(vi) **GY** - Gyrator, e.g. electro-motor, centrifugal pump.

(vii) **0** – and **1** – junctions, for ideal connection of two or more sub-models.

While considering the elements of the bond graph as lump parameters, the electrical and hydraulic components share the same characteristics (Das, 2009). This means that their modelling patterns and considerations are almost similar;

(The capacitance/Capacity exhibited by Tank/Capacitors, the Resistance/Inductiveness by the Resistor/Pipe, the transformation by Transformer/Valve. Of course, the supply by pump and voltage source (Li et al., 2014). Bond graph which was developed by Henry Paynter of Massachusetts Institute of Technology-MIT

(Borutzky & Cellier, 1996; Breedveld, 1985) established a common/unified language for modelling a multi-domain energy transaction process. It is widely used in the industry for mechatronics system modelling, fault detection and identification (FDI) and simulation(Borutzky, 2013; Tsai & Gero, 2010) and integrated system design(Nacusse & Junco, 2014; Seo *et al.*, 2005). Bond graph implementation impacts the four levels of system modelling(Das, 2009) as follows:

1. **Technological Level**: the architectural representation of systems and subsystems in the form of icon (heat exchangers, tanks, valves and resistors) are used for system models. The use of word bond graph (describing the network of components by words and their names within their blocks) is suitably used (Karnopp *et al.*, 2012).
2. **Physical Level**: modelling at this level is done using energy descriptions of physical phenomena. The basic concepts of physics such as energy dissipation, transformation, accumulation, sources and so on are considered with their relevant physical dynamics (inertia, friction, and compressibility, to mention but a few.), which are easily achieved using bond (Borutzky, 2013).
3. **Mathematical level**: The mathematical model of a system is represented by use of algebraic and differential equations, that are derived from

system behaviour abstractions. Hence in bond graph, the use of constraints and components are utilized to achieve this with much ease and accuracy (Dajun, 2013).

4. **Algorithmic Level**: At this level information flow determines how the mathematical models are formed, as bond graph assignment of causality clearly captures this procedure directly(Borutzky, 2013).

Bond graph, however, is seen as a unique language for physical domain modelling, because the use of it in modelling demonstrates clearly, cause and effect relations. This creates room for expansion, incorporation of new sub-models in an existing one, efficient structural analysis, fault detection and isolation (FDI) capabilities. Bond graph is easily incorporated with most simulation software's, like 20-sim, Simulink/ Matlab, Dymola and labVIEW (Ngwompo, 2011; Paynter, 1989)

### 2.2.10 20-sim Simulation Software

20-sim is an application software, developed for the validation and simulation of dynamics systems. It is a modelling and simulation program that runs under Microsoft Windows, Linux and Mac Operating software. With 20-sim, the behaviour of dynamic systems can be simulated, such as electrical, mechanical and hydraulic systems or any combination of these.

20-sim fully supports graphical modelling, allowing to design and analyse dynamic systems in an intuitive and user friendly way, without compromising power. 20-sim supports the use of components (Nacusse & Junco, 2014). This allows the user to enter models as in an engineering sketch: by choosing components from the library and connecting them. The engineering scheme is actually rebuilt, without entering a single line of math! It contains a bond graph package which makes it suitable for use as shown in literature (Aridhi *et al.*, 2014; Nacusse & Junco, 2014). 20-sim is built with the capability to interact across other modelling and simulation mathematically based software's like MATLAB, labVIEW, Dymola, Modelica etc. Some of the tool boxes are, Bond graph Toolbox, Mechanics Toolbox, Animation Toolbox, Frequency Domain Toolbox, Control Toolbox, Mechatronics Toolbox, Real Time Toolbox, Time Domain Toolbox (Broenink, 1999).

### 2.2.11 Decentralized Model Predictive Control

The motivation for the implementation of predictive control scheme in the control of processes is imperative (Mercangöz & Doyle, 2007) as issues pertaining to complexity in plant processes and dynamics become more pertinent (Wang, 2009). Predictive control based on model dynamic approaches have been implemented to address challenges arising from complex system configurations (Christofides *et al.*, 2013; Scattolini, 2009). In the scheme, different structures of the controller are designed to address the control objective of the process plant(Raimondo *et al.*, 2007), amongst which are,

1. **Complete Decentralised Structures**: The subsystems, which makes up the entire system block are partitioned in such a manner as to avoid coupling. The dynamics and control of one partially affect or totally does not impact the other. Control is autonomous for each subsystem.
2. **Distributed Control and Hierarchical Structures**; some information is shared between the systems and update is carried out in this scenario (Alvarado *et al.*, 2011) . Hence control is done in hierarchical manner.

The subsystems may be linked to a central control but basically, there is a share of information between one subsystem and the rest.

The components upon which the implementation of Model Predictive Control is founded are defined as follows:

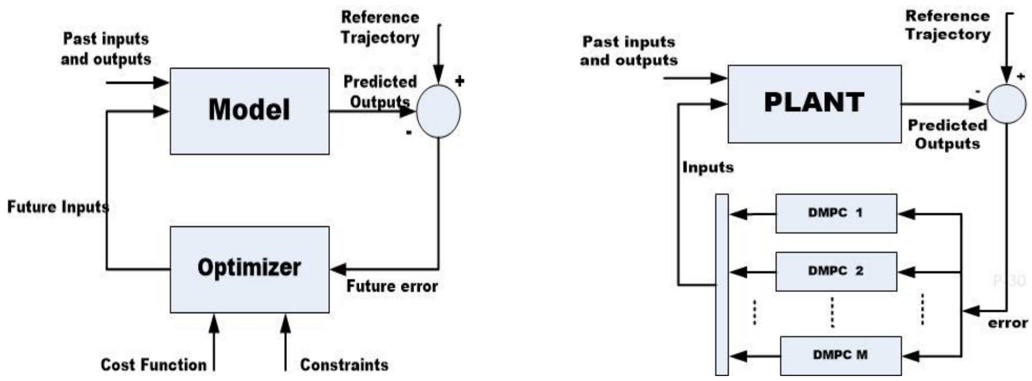
(i) **Moving Horizon Window**: the time-dependent window from an arbitrary time $t_i$ *to* $t_i + T_P$. The length of the window $T_P$ remains constant. However, $t_i$ which defines the beginning of the optimization window, increases with continuous process.

(ii) **Prediction Horizon (Np)**: This dictates the extent to which the future can be predicted for the performance of the system. This parameter equals the length of the moving horizon window, $T_P$.

(iii) **Receding Horizon Control (Nc)**: Although, the optimal trajectory of future control signal is completely described within the moving horizon window, the actual control input to the plant only takes the first sample of the control signal, while neglecting the rest of the trajectory {Zheng, 2010}.

The decentralised approach proposed in the control of the quadruple tank is perceived to be able to address and adequately fit alongside with the use of bond graph, which is aimed at modelling and control from energy-component levels of the subsystems. With the use of DMPC, the control objective is transformed into an optimization cost function such that at any sampling time instant, a sequence of future control values is computed by solving a finite horizon optimal control problem (Gupta *et al.*, 2012). Then, only the first element of the computed sequence is effectively used and the overall procedure is repeated at the next sampling time according to the receding horizon (or moving horizon) principle(Mercangöz & Doyle, 2007). Hence, the flexibility in design is made possible by the optimization nature of the formulated control problem which can enable the incorporation of several constraints reflecting different performance issues. Moreover, it allows for handling static and dynamic constraints on the plant variables(Christofides *et al.*, 2013)

A basic implementation of MPC in general can be attained. The decentralised scheme also is as represented by Figure 2.6(a and b) {Zheng, 2010}




(a) A Basic Optimization Diagram

(b) DMPC Design

Figure 2.6: MPC Schemes

Reliability and robustness of the overall control system are part of the reasons for avoiding centralized solutions and for adopting decentralized approaches (Husek, 2014). Figure 2.7 is a representation of the design scheme for a two input, two output (TITO) systems. It is observed thus, that the dynamics of each subsystem $S_1, S_2$ is influenced by the inputs and the states of each other.

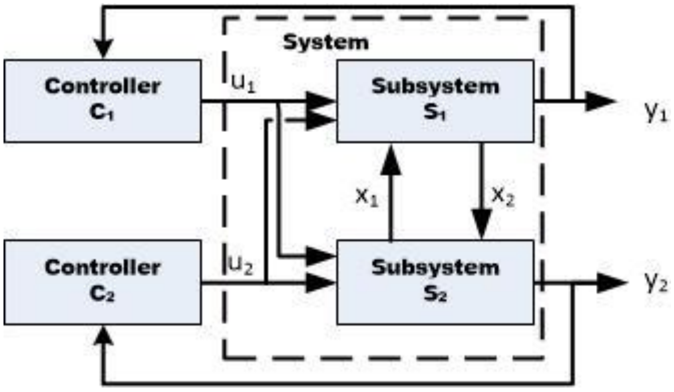



Figure 2.7: A DMPC Schematic Representation of a TITO System

The formulation is basically expressed in state space equations that present a generic and simple representation of the control strategy. The standard MPC problem based on the linear discrete-time prediction model is considered (Maciejowski, 2002).

$$x(k+1) = Ax(k) + Bu(\mathrm{k}) \tag{2.14}$$

$$y(k) = Cx(k) \tag{2.15}$$

Where $x(k) \in \mathbb{R}^n$ is the state vector and $u(k) \in \mathbb{R}^m$ is the vector of command variables at time

step k, and $y(k) \in \mathbb{R}^p$ is the vector of the measured output. A is the states matrix, B is the input matrix, C is the output matrix. Following the formalism for the predictive control, the cost function is assumed to be quadratic and the constraints are in the form of linear inequality (Wang, 2009). Thus, the following optimization problem has to be solved:

$$\begin{aligned} \min_u V(k) = \min_u x(k+H_P)^T P x(k+H_P) + \sum_{j=0}^{H_P-1} x(k+j)^T Q x(k+j) \\ +u(k+j)^T R u(k+j) \end{aligned} \tag{2.16}$$

$$x(k+1) = Ax(k) + Bu(k) \tag{2.17}$$

Subject to:

$$x(0) = x(k) \tag{2.18}$$

$$\begin{gathered} u_{\min} \le u(k) \le u_{\max}, \\ k = 0, \ldots, H_P - 1 \end{gathered} \tag{2.19}$$

Where $H_p$ is the prediction horizon, $x(0)$ is the initial condition of the state vector, $u_{\min}$ and

$u_{\max}$ are known vectors defining the saturation constraints on inputs variables (operational ranges). $Q_i \in \mathbb{R}^{mm}, P_i \in \mathbb{R}^{rr}$ *and* $R_i \in \mathbb{R}^{rr}$ are symmetric and positive semi-definite weighting matrices specified by the user.

They could be chosen in some cases as;

$Q_i = qIm, P_i = pIr$ *and* $R_i = r0Ir$

For some positive parameters q, p and r0. They are diagonal. Moreover, P is chosen as zero in order to obtain MPC with offset-free control, i.e., y = r in steady state. The weighting matrices are almost always chosen as time invariant matrices, i.e., the weighting matrices are constant over the prediction horizon M so that;

$$Q_1 = Q_2 = \cdots = Q_{H_P} \tag{2.20}$$

$$P_1 = P_2 = \cdots = P_{H_P} \tag{2.21}$$

Trial and error method is often used. Meanwhile, choosing Q, P and R as diagonal positive definite matrices and if the prediction horizon $H_P$ is large (infinite) then it can be proved some remarkable properties of the closed loop (controlled) system.

Equation(2.17) can be recast as a Quadratic Programming (QP) problem, whose solution

$$U^*(k) \square [\mathrm{u}^*(k)^T \ldots u(k+H_P-1)^T]^T \in R^{H_{Pm1}} \tag{2.22}$$

is a sequence of optimal control inputs that generates an admissible state sequence. At each sampling time k, equation(2.17) is solved for the given measured (or estimated) current state $x(k)$. Only the first optimal move $u^*(k)$ of the optimal sequence $U^*(k)$ is applied to the process,

$$u_{mpc}(k) = u^*(k) \tag{2.23}$$

The remaining optimal moves are discarded and the optimization is repeated at time $k+1$

### *2.2.11.1 Decentralized Cost Function Problem*

The minimization for the subsystems ($i$) objective functions is as shown:

$$\min_{x_{ii}u_i} \Phi_i(x_{ii}, u_i) \tag{2.24}$$

s.t

$$x_{ii}(k+1) = A_{ii}x_{ii} + B_{ii}u_i \tag{2.25}$$

And

$x_{ii}(k) = \hat{x}_{ii}(k)$;

$t = k, k+1, \ldots, k+N-1$ in which $\hat{x}_{ii}(k)$ denotes the estimated states on the DMPC.

Therefore, the quadratic cost function of the $i^{th}$ decentralized subsystem is stated thus:

$$\Phi_i(\hat{x}_i(k), \hat{u}_i(k) = \frac{1}{2}\hat{x}_i^T(\mathrm{k})\mathrm{Q}_i\,\hat{x}_i(k) + \frac{1}{2}\hat{u}_i^T(\mathrm{k}) \tag{2.26}$$

Such that

$$x_i(k+1) = A_i x_i(k) + \Gamma_i \nu_i(k) + B_i u_i(k) \tag{2.27}$$

in which;

$Q_i = diag(Q_i(1), \mathrm{Q}_i(2), \ldots, Q(N))$

$R_i = diag(R_i(0), R_i(1), \ldots, R(N-1))$

Denoting the state to input penalties in the regulator through the control horizon (N).

Subject to constraints:

$x_{\min,i} \le x_i \le x_{\max,i}, \ \nabla k$

$u_{\min,i} \le u_i \le u_{\max,i}, \ \nabla k$

The performance criterion in the application of MPC is determined by the optimization of the objective function of the system. However, the objective function is either a standard least-square or a quadratic objective function. The standard least-square function is a formulation based on the "sum of squares" of predicted errors (difference between the set-points and the model predicted outputs) and the control moves (step by step changes in control action). Figure 2.8 depicts the current stage (k) and subsequent one (k+1) stages and steps considered during prediction for present and future move while capturing the past moves, in order to arrive at an acceptable optimal predicted solution.

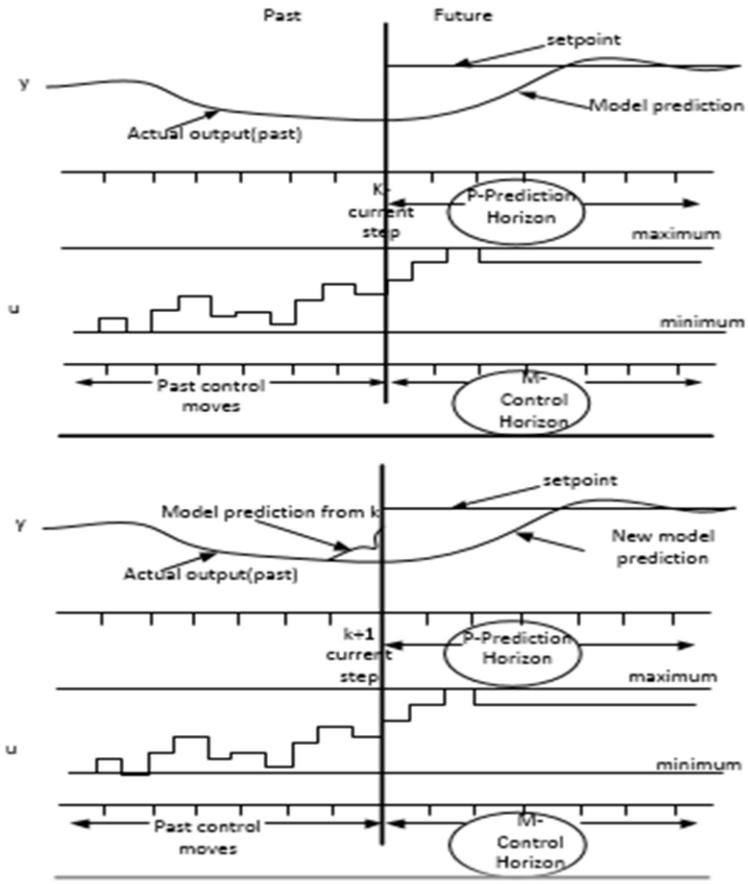



Figure 2.8: The Prediction Working States of MPC

The prediction at step k, is based on the information at k-1, hence with current state, it is shown that an estimated state can be realised as in equation(2.28);

$$\begin{aligned} x_{k+1} &= \Phi x_k + \Gamma u_k + \Gamma^d d_k \\ y_{k+1} &= C x_{k+1} = C\Phi x_k + C\Gamma u_k + C\Gamma^d d_k \end{aligned} \tag{2.28}$$

Furthermore, it can be shown, using control changes that

$$\begin{aligned} u_k &= u_{k-1} + \Delta u_k \\ y_{k+1} &= C\Phi x_k + C\Gamma u_{k-1} + C\Gamma^d d_k + C\Delta u_k \end{aligned} \tag{2.29}$$

When, disturbance is considered, an estimate of equation(2.29) can make a representation in the form of equation(2.30)

$$\hat{y}_{k+1|k} = C\Phi x_{k|k} + C\Gamma u_{k-1} + C\Gamma^d d_{k|k} + C\Delta u_k \tag{2.30}$$

With $\hat{y}_{k+1|k} \ldots$ Predicted output estimated at k+1

$C\Phi x_{k|k} \ldots$ State estimate at k and Measured through k

$C\Gamma u_{k-1} \ldots$ Previous estimate.

$C\Gamma^d d_{k|k} \ldots$ Estimated disturbance at k including measurements and

$C\Delta u_k$ Current change.

When propagated into the future, (P - steps predictions), then representation in states space can be shown as in equation(2.31);

$$\begin{pmatrix} \hat{y}_{k+1|k} \\ \hat{y}_{k+2|k} \\ \vdots \\ \hat{y}_{k+P|k} \end{pmatrix} = \begin{pmatrix} C\Phi \\ C\Phi^2 \\ \vdots \\ C\Phi^P \end{pmatrix} \hat{x}_{k|k} + \begin{pmatrix} C\Gamma \\ C\Phi\Gamma + C\Gamma \\ \vdots \\ \sum_{i=1}^{P} C\Phi^{i-1}\Gamma \end{pmatrix} u_{k-1} + \begin{pmatrix} C\Gamma^d \\ C\Phi\Gamma^d + C\Gamma^d \\ \vdots \\ \sum_{i=1}^{P} C\Phi^{i-1}\Gamma^d \end{pmatrix} \hat{d}_{k|k} \tag{2.31}$$

With a forced response given as in equation(2.32);

$$\begin{pmatrix} C\Gamma & 0 & \cdots & 0 \\ C\Phi\Gamma & C\Gamma & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i-1}^{P} C\Phi\Gamma & \sum_{i-1}^{P} C\Phi_{i-1}\Gamma & \cdots & 0 \end{pmatrix} \begin{pmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{pmatrix} \tag{2.32}$$

### 2.2.12 Parameter Estimation Application

Prediction made based on measurement and estimation made through k - 1, at k is given by

$$\hat{x}_{k|k-1} = \Phi\hat{x}_{k-1|k} + \Gamma u_{k-1} \tag{2.33}$$

At instant k, the new prediction through k is:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k (y_k - C\hat{x}_{k|k-1}) \tag{2.34}$$

Hence, the state error covariance estimate results to

$$P_k = \Phi P_{k-1} \Phi^T + \Gamma^w Q_{k-1} \Gamma^{wT} - \Phi P_{k-1} C^T (C P_{k-1} C^T + R)^{-1} C P_{k-1} \Phi^T \tag{2.35}$$

From which the Kalman steady-state gain is given as:

$$L_k = P_k C^T (C P_k C^T + R)^{-1} \tag{2.36}$$

### 2.2.13 Proportional Integral and Differential (PID) Control

The classical control era was predominantly dominated with use of PID control policy (Aravind *et al.*, 2013). It's often tuned manually by trial and error methods while experience of the operator is an advantage in such process plants (Romero *et al.*, 2014). The proportional, integral and differential compensate for system errors to ensure stability and control (Vijula & Devarajan, 2013). Amongst the popular configuration of PID's are full matrix implementation in MIMO systems, and decentralized PID with incorporation of decoupler for SISO systems (Tavakoli *et al.*, 2006). In configuration of the control policy as shown in Figure 2.9, the continuous controller/discrete controller takes any of these forms; P, I, D, PI, PD, PID, and PIDD. In this work, the PID configuration's performance is compared against that of QPDMPC designed in this work using MATLAB/Simulink 2013b's Control Toolbox (Kala *et al.*, 2014).

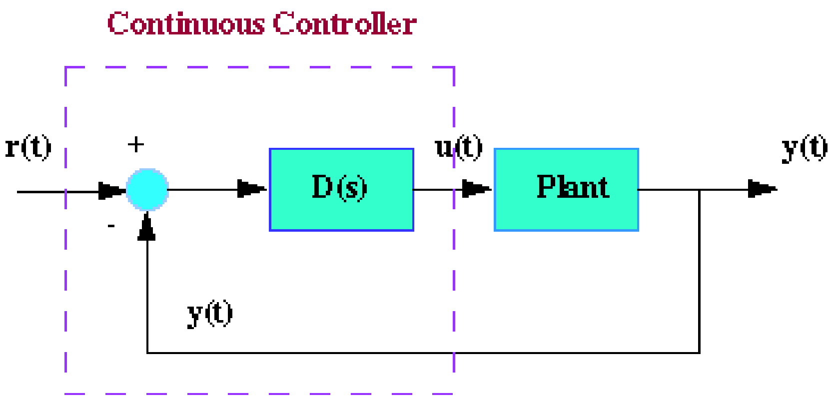



Figure 2.9: Basic Feedback Control Scheme

In Figure 2.9, the controller group is made of the decoupler and controllers while the process represents the system controlled. The output of the PID is given in terms of the three gain types as shown in equation (2.37):

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{de(t)}{dt} \tag{2.37}$$

In which:

$k_p$ ; is the proportional gain factor,

$k_i$ ; is the integral gain factor,

$k_d$ ; is the differential gain factor.

The tuning of the gain factors aims at a possible achievement of a stable and robust controlled system.

## 2.3 REVIEW OF SIMILAR WORKS

In order to effectively control and make the uttermost use of resources with improved productivity, a number works have been carried out by researchers in the area of multi-process modelling and control. These works and others from the background section formed the direction this work is built upon.

In the work of **Edward *et al*, (2000)**, model based control of a four tank-system was developed, for a practical demonstration and teaching of process control systems. Implementation and methodology were based on Internal Model Control (IMC) and Dynamic Matrix Control (DMC). However, the implementation of the DMC was partial owning to the non-availability of measuring instruments for flow calculation as the flexibility of incorporating constraints (sudden leakage, pump error, source oscillations, spilling and over flow) on real experiment could compromise safety.

In the work of **Åström*et al*,(2002)**, the design of a control scheme which was based on a single loop Proportional Integral (PI) and static decoupler control was implemented in the control of a Two-by-Two system (a multi-variable system). The use of such scheme, is as old as the process industry itself. In addition, this control scheme has been used effectively based on experience. The challenge with the authors' proposed approach is that its effective implementation relied on the expertise of the operator, since the tuning of the PI controller is done by trial and error.

Also, in the work done by **Garelli, (2006)**, a method to restrict the interaction of loops prevalent in Multi-Input Multi-Output systems was proposed. The method, which was based on Sliding Mode attributes for decentralised control scheme requires the designer to determine the extent to which the interactions with respect to a reference vector would occur. However, the multi-variable system model was developed based on first principles from differential equations and therefore requires some assumptions which makes the process susceptible to error as system complexity increases.

**Kubalčík and Bobál(2013)** developed an application for self-tuning predictive controller of a real-time control of three-tank-system, which was a Two-Input Two-Output System. The control algorithm was based on the Generalised Model Predictive Control Method (GMPC). The model of the 3-tank system was derived from first principles, and was converted to difference equations. However, this approach makes the reverse conversion of the state – space form to its original (differential equation form) quite a challenge.

**Vijula, (2013)**developed a model of quadruple tank system using mass balance Bernoulli's principle equation and differential equations. The use of first principles based on mass balance of Bernoulli equation was used for modelling the tank system. Assumptions made in the modelling from first principles could lead to substantial errors e.g. round off errors.

**Nacusse*et al*, (2014)** proposed a switchable structured bond technique. The method was implemented in modelling of a two tank system effecting coupling/decoupling of the physical systems. The efficiency of the use bond graph for modelling was demonstrated in this work, however, complexity of structural consideration was limited to system of two tank system with emphasis on fixed causality.

**Saeed *et al*, (2013)**proposed a research work on improving a niching genetic scheme while using bond graphs for Topology and parameter optimization of a Mechatronic System. The novel multi-modal evolutionary optimization algorithm is said to be able to model complex problems with the objective of realizing the best topology and parameters from multi-criteria view point and different preferences. However computational time was long.

The work of **Triki *et al*, (2014)** had presented a Simulink model for systems with mixed continuous and discrete behaviours. The model developed was based on modelling a system using bond graph methodology. The modelling was carried out using hybrid bond graphs (HBGs), that incorporates local switching functions that enable the reconfiguration of energy flow paths. However, the structure was that of three tank system connected serially with less complexity as compared to quadruple tank.

Subsequently, **Srinivasarao *et al* (2014)** developed a multi-input multi-output process system control based on Model Predictive Control with centralised strategy. The authors proposed a centralised implementation and control strategy which maybe simple in implementation and control. The modelling of the system also was based on differential equations and first principles which are tasking and time consuming and error bound. The controller is centralized and may become very costly to implement as the complexity of the system increases (expensive optimization function and difficult constraints may arise).

In the work of **Aridhi *et al*, (2014)**, a real data acquisition of temperature in an open loop was carried out on a single water tank system using LabVIEW data acquisition toolbox. A comparison was made between the data acquired against pseudo bond graph model that was realised using 20-sim software. But in carrying out the modelling in 20-sim, the temperature constraint which played a vital role in the data collection was overlooked; This can lead to a poor temperature set-point tracking causing degradation in system performance.

**Parisio *et al*, (2014)**worked on the optimal operation planning of a microgrid. Their aim was to minimize the overall microgrid operating costs to meet predicted load demand at certain periods (typically one day) while satisfying complex operational constraints, such as the energy balance using Model Predicted Control centralized approach via mixed integer linear programming. The challenge in the work is that when the system complexity increased, with more microgrids injected into the system, achieving the same aim will be more difficult and more expensive.

**Li *et al* (2014)**explored a kind of decentralized PID tuning method for a wide-spectrum process, and then applied it to the linear and nonlinear ALSTOM gasifier benchmark control problem to verify its feasibility. However, constraint handling was not addressed, which is a deficiency with the Proportional Integral and Differential (PID) controller adopted in the work.

This work applied the bond graph technique in modelling the quadruple tank system in order to minimize the mathematical complexities and time consuming constraints of the conventional differential equations based techniques. In the design of decentralized MPC, consideration is made first of a classical predictive control approach in optimizing the cost function. However, because of the possible numerical ill-conditioning of the problem (which may lead to the resultant system not being stabilized and tracked satisfactorily by classical predictive controller), a more robust cost function (quadratic programming (QP)) may be considered at the possible expense of convergence time.

# CHAPTER THREE

# QUADRUPLE TANK PROPOSED SOLUTION

## 3.1 INTRODUCTION

In this chapter, the mathematical equations which formed the model of the quadruple tank is derived from physics of the various components of the entire system. The energy exchange that is flow and effort are articulated and traced along the lump parameter components (energy dissipaters and energy stores).

## 3.2 BOND GRAPH IMPLEMENTATION

The procedures for the derivation of equations from a causal bond graph includes the following steps:

(i) Write the constitutive equations for all independent sources. Their outputs are given functions of time.

(ii) In contrast, the output of a controlled source is algebraically related to its input. If the latter is not an output of an independent source or an energy store with integral causality, then it can be expressed by means of such outputs by back propagation of causal paths in the junction structure and by eliminating intermediate variables.

(iii) The outputs of resistors depend algebraically on their inputs. By back propagation along causal paths through the junction structure, their outputs can be expressed by sources either independent or controlled ones, or outputs of energy stores. The outputs of dependent sources do not need to be eliminated, since they have already been determined in the previous step.

(iv) For storage ports, the derivative with respect to time of an output is a function of the input(s). By working back causal paths, the inputs can be expressed by outputs of other energy stores, of resistors, or sources. If there are causal paths between resistor ports, implicit, algebraic equations will result. Implicit algebraic equations also result, if there are closed causal paths in the junction structure (Benmoussa *et al.*, 2014).

If there are storage elements that must accept derivative causality, then their output variables are algebraically dependent on the outputs of the storage elements in preferred integral causality and the number of states is smaller than the number of storage elements.

Thus, in deriving the equations of the multi process tank systems of Figure 2.1, consideration is made vitally of the flow and effort directions what;

(i) Each element is contributing to the system,

(ii) The system gives back to each storage element as can be obtained from Figure 3.1.

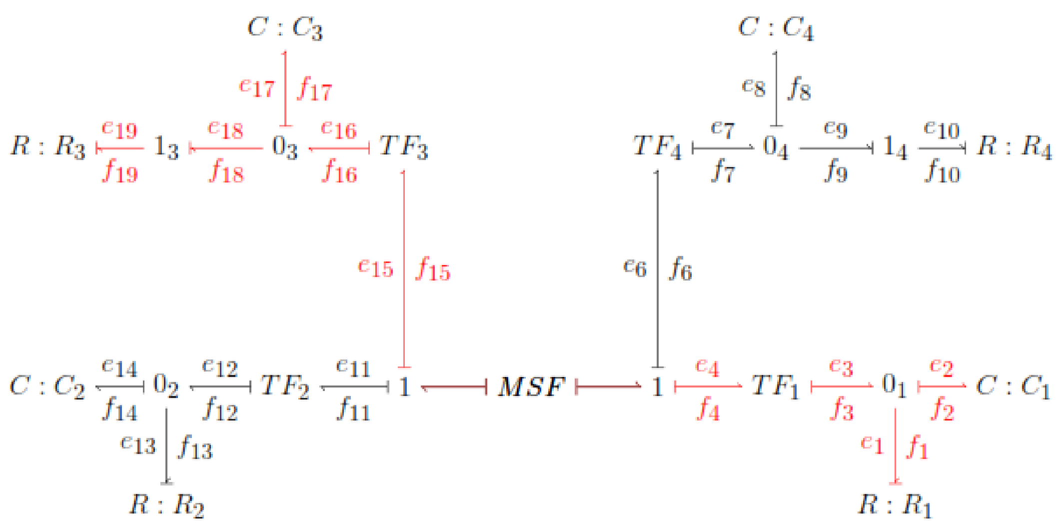



Figure 3.1: Quadruple Tank Indicating Flow and Effort Information Bonds Stage 1

The entire system model is graphically captured as shown in Figure 3.2, with all connections between subsystems.

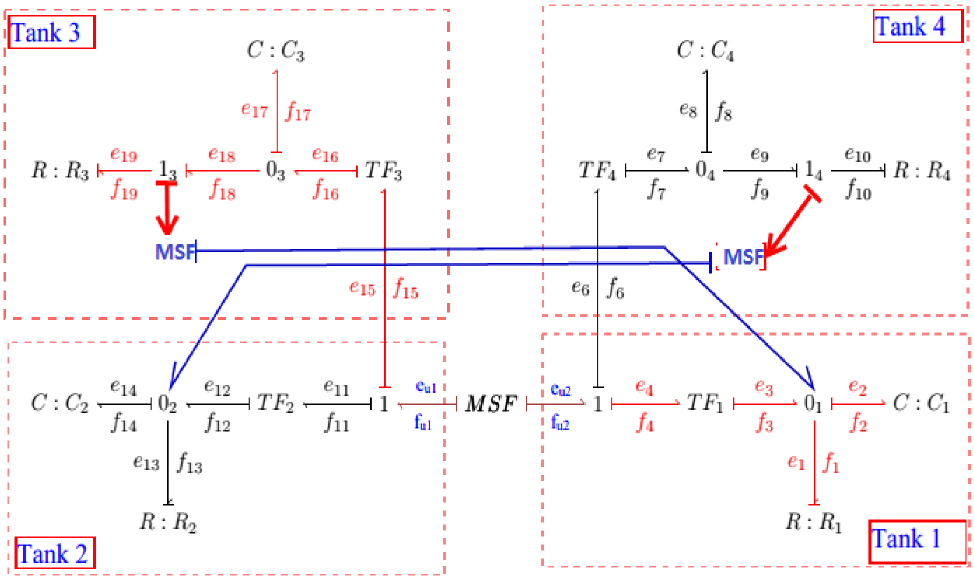
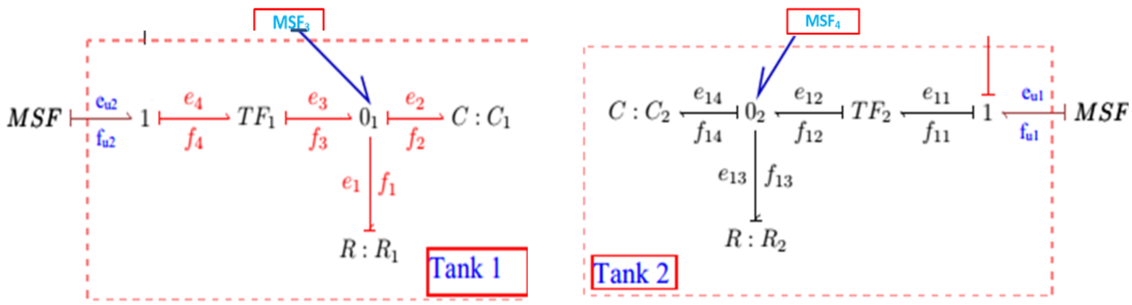



Figure 3.2: The Quadruple Tank Crossed Looped Information Scheme: Stage 2

Furthermore, the systems when segmented into sub-plants, Figures 3.3 and 3.4 are obtained as shown

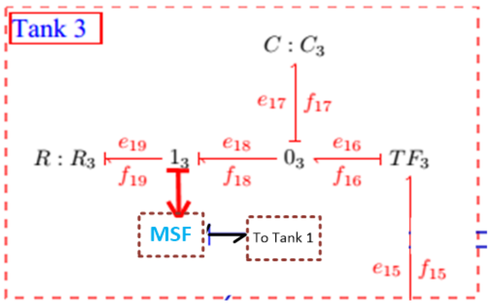


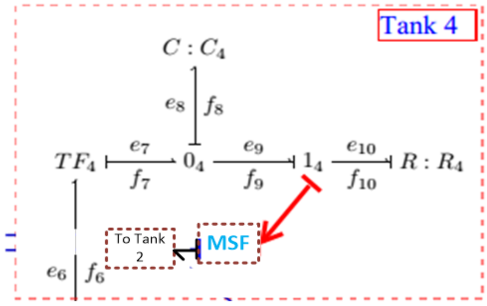

(a) Bond Information Tank 1

(a) Bond Information Tank 2

Figure 3.3: Tank –Segmentation 1

(a) Bond Information Tanks




(b) Bond Information Tank 4

Figure 3.4: Tank Segmentation 2

The physical build-up super-imposed on the bonds is visualized as illustrated by Figure 3.5:

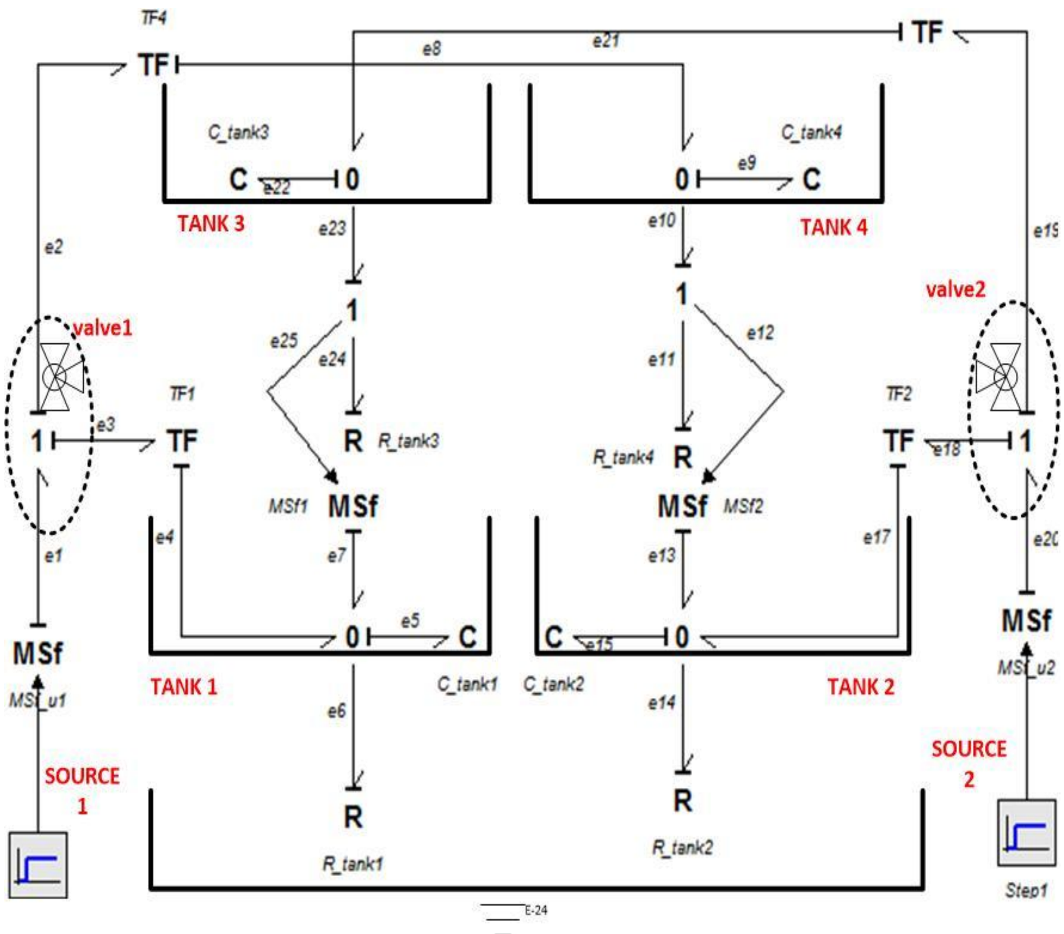



Figure 3.5: Bond Graph of a Quadruple Tank

From Figure 3.5, the sources are from Source1 and Source2 which are physical pumps through separate voltage supplies. The flow of non-compressible liquid is modulated via this sources. The valves (three-way) splits the direction to which the quantity of fluid flows at each sampling period, by proportional variations ($\gamma$ &$1-\gamma$ )$\in$ [0, 1]. The voltages (efforts) across the pipes and the tanks i.e. the resistors ($R-$ elements) and capacitors ($C-$ elements) are represented by the 1 junctions. The $R-$ elements and $C-$ elements are however attached to the 0 junctions: 0 junctions are points at which all potentials are equal. The bonds are also numbered from (effort $e_1$ to $e_{25}$ ), but each bond represents a flow and an effort as in Figure 3.5. The validation of the bond graph model of the quadruple tank system as carried out using 20-sim is as shown in Figure 3.6

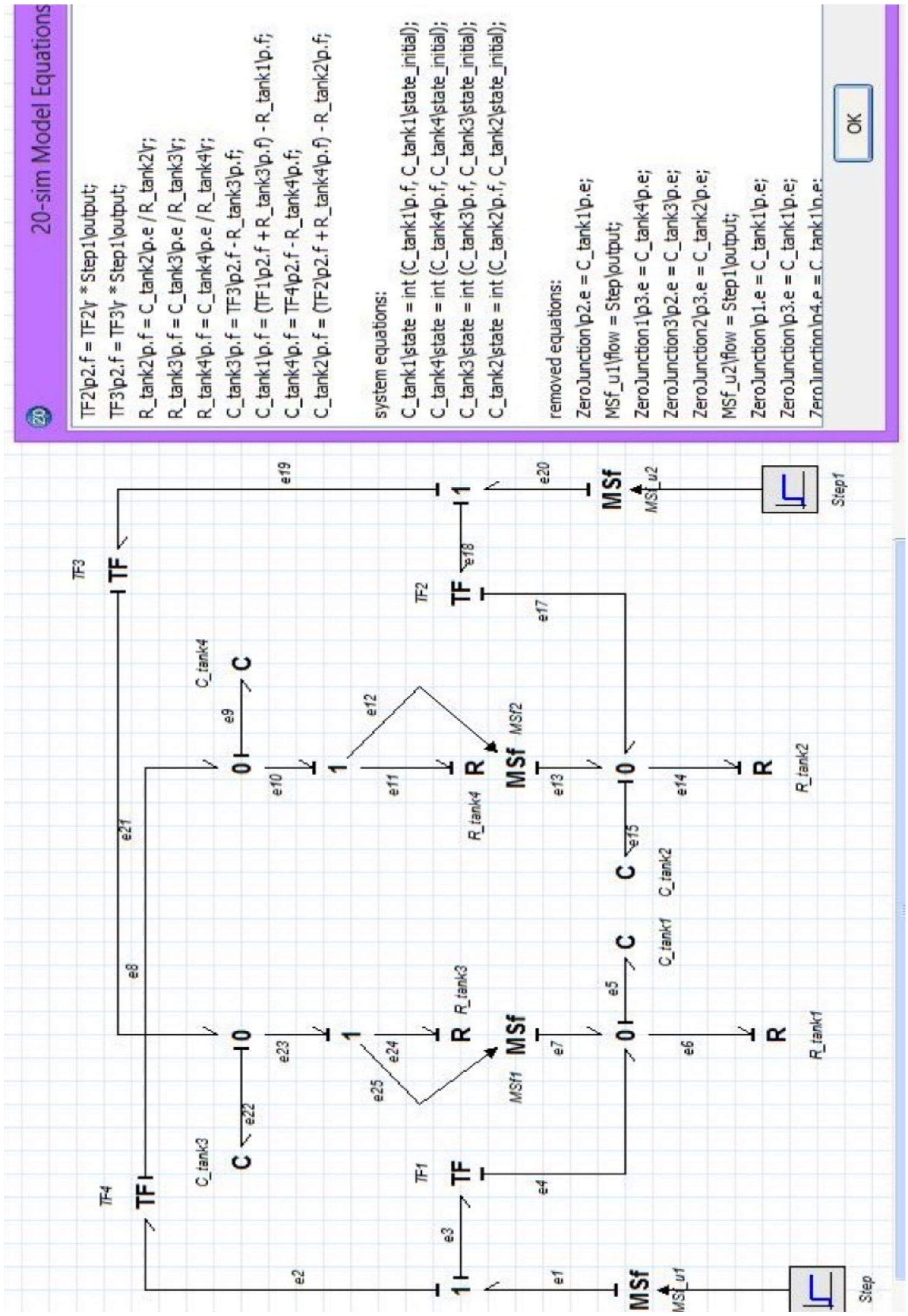


Figure 3.6: Quadruple Tank Validation in 20-simValidated Structure

### 3.2.1 Variables

The vector, H, containing all power variables is deduced from bond model of Figure 3.2

$H = [e_{14}\, f_{14} e_{13}\, f_{13} e_1\, f_1 e_2\, f_2 e_{10}\, f_{10} e_8\, f_8 e_{17}\, f_{17} e_{19}\, f_{19}]$

Which results to a total number of components, i.e., nc = 8

The set, K, of known variables is:

$K = [MSF_1, MSF_2, MSF_3, MSF_4]$

### 3.2.2 Structural Equations

From the segmented tanks of Figure 3.3 and Figure 3.4 the structural equations are derived as

a. $F_{J0}$: Junction 0 linked to R, or C elements. The equation of these structures are:

$0_1, 0_2, 0_3, 0_4$ are given by ( $e_m = e_n$ where m = 1, 2… and n = 1, 2…)

$$\begin{aligned} F1 &: f_2 = f_3 - f_1, \\ F2 &: f_{14} = f_{12} - f_{13}, \\ F3 &: f_{17} = f_{16} - f_{18}, \\ F4 &: f_8 = f_7 - f_9. \end{aligned} \tag{3.1}$$

b. Equations for the 1 junctions $1_1, 1_2, 1_3, 1_4$ .

$$\begin{aligned} F_5 &: eu_2 - e_4 - e_6 = 0 \\ F_6 &: \mathrm{eu}_1 - e_{11} - e_{15} = 0 \\ F_7 &: e_{19} = e_{18} \\ F_8 &: e_9 = e_{10}. \end{aligned} \tag{3.2}$$

c. Transformers: $TF_{(1-4)}$

$$
\begin{aligned}
&F9: e_4 = TF_1 * e_3 = \gamma * e_3 \\
&F10: e_{12} = TF_2 * e_{11} = \gamma * e_{11} \\
&F11: e_{15} = TF_3 * e_{16} = (1-\gamma) * e_{16} \\
&F12: e_6 = TF_4 * e_7 = (1-\gamma) * e_7
\end{aligned}
\tag{3.3}
$$

The model equations as validated by 20-sim are shown in Figure 3.7

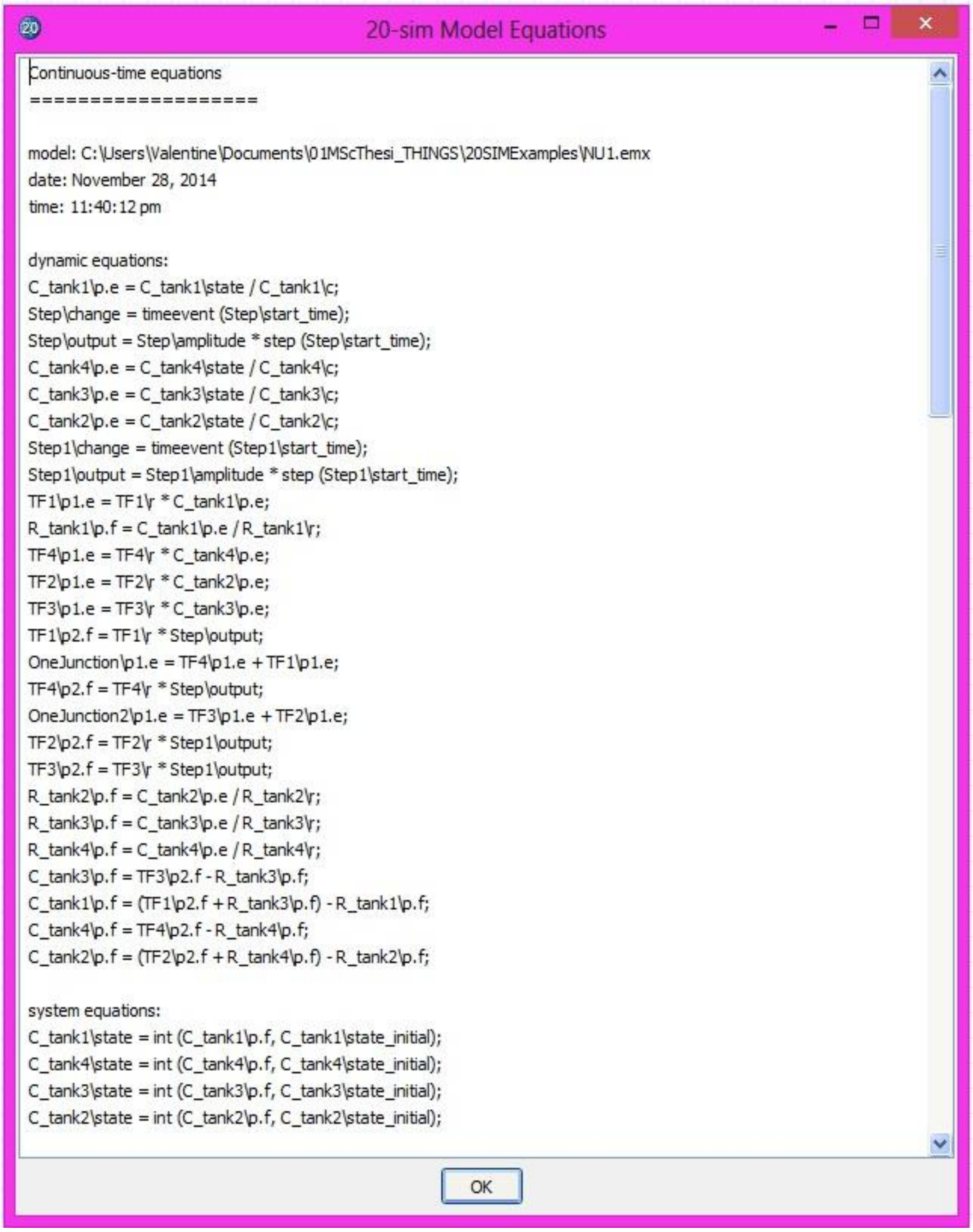



Figure 3.7: 20-sim Model Equations for Quadruple Tank

From Figure 3.7, the following; system behavioural equations, systems dynamic equations, and the nonlinear model equations of the quadruple tank derived are validated.

### 3.2.3 Behavioural Equations

Each bond graph element is associated with a physics characteristic equation. This fundamental constitutive equation is known as the behavioural equation of the element. The physical laws expressing how the energy is transformed are mathematically described by the behaviour model. For the quadruple tank, the constitutive equations obtained as follows:

$$\begin{aligned}
&F_{13}:R:R_1 \rightarrow f_1 = |a|\sqrt{|e_1|},\\
&F_{14}:R:R_2 \rightarrow f_{13} = |a|\sqrt{|e_{13}|},\\
&F_{15}:R:R_3 \rightarrow f_{19} = |a|\sqrt{|e_{19}|},\\
&F_{16}:R:R_4 \rightarrow f_{10} = |a|\sqrt{|e_{10}|},\\
&F_{17}:C:C_1 \rightarrow e_2 = \frac{1}{C_1}\int f_2 dt,\\
&F_{18}:C:C_2 \rightarrow e_{14} = \frac{1}{C_2}\int f_{14} dt\\
&F_{19}:C:C_3 \rightarrow e_{17} = \frac{1}{C_3}\int f_{17} dt\\
&F_{20}:C:C_4 \rightarrow e_8 = \frac{1}{C_4}\int f_8 dt
\end{aligned} \tag{3.4}$$

### 3.2.4 Nonlinear System Dynamics

System dynamics is an approach to understanding the behaviour of complex systems over time. It deals with internal feedback loops and time delays that affect the behaviour of the entire system. Hence, the interconnections between the elements are derived as follows:

$$
\begin{aligned}
&R_{tank1} \backslash p.f = C_{tank1} \backslash p.e / R_{tank1} \backslash r;\\
&R_{tank2} \backslash p.f = C_{tank2} \backslash p.e / R_{tank2} \backslash r;\\
&R_{tank3} \backslash p.f = C_{tank3} \backslash p.e / R_{tank3} \backslash r;\\
&R_{tank4} \backslash p.f = C_{tank4} \backslash p.e / R_{tank4} \backslash r;\\
&C_{tank1} \backslash p.f = (TF_1 \backslash p_2.f - R_{tank3} \backslash p.f) - R_{tank1} \backslash p.f;\\
&C_{tank2} \backslash p.f = (TF_2 \backslash p_2.f - R_{tank4} \backslash p.f) - R_{tank2} \backslash p.f;\\
&C_{tank3} \backslash p.f = (TF_3 \backslash p_2.f - R_{tank3} \backslash p.f);\\
&C_{tank4} \backslash p.f = (TF_4 \backslash p_2.f - R_{tank4} \backslash p.f);
\end{aligned} \tag{3.5}
$$

### 3.2.5 System Equations

The system equations describe the entire characteristics of the physical system modelled. These are derived as follows:

$$C_{tank1} \backslash state = \int (C_{tank1} \backslash p.f, C_{tank1} \backslash state\text{-}initial); \tag{3.6}$$

$$C_{tank2} \backslash state = \int (C_{tank2} \backslash p.f, C_{tank2} \backslash state\text{-}initial); \tag{3.7}$$

$$C_{tank3} \backslash state = \int (C_{tank3} \backslash p.f, C_{tank3} \backslash state\text{-}initial); \tag{3.8}$$

$$C_{tank4} \backslash state = \int (C_{tank4} \backslash p.f, C_{tank4} \backslash state\text{-}initial); \tag{3.9}$$

With $h_1, h_2, h_3\, and\, h_4$ represents the liquid level of $tank_1, tank_2, tank_3,$ and $tank_4$ respectively; $A_1, A_2, A_3, and\, A_4$ are the cross section areas of the tanks, related to the tanks hydraulic capacities by the relation $C_i = \frac{A_i}{\rho g}$ (with i = 1, 2, 3, 4); row is the constant density of the fluid, g is the gravitational acceleration while $a_1, a_2, a_3\, and\, a_4$ are the cross sectional areas of the pipe assumed to be equal. The source is also given by $k_1u_1$ and $k_2u_2$ for the two inputs. The resultant system model is then derived as

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1\kappa_1}{A_1}u_1 \tag{3.10}$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2\kappa_2}{A_2}u_2 \tag{3.11}$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{\gamma_2\kappa_2}{A_3}u_2 \tag{3.12}$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{\gamma_1\kappa_1}{A_4}u_1 \tag{3.13}$$

## 3.3 LINEARISATION OF SYSTEM DESCRIPTIVE EQUATIONS

The work of (Vadigepalli *et al.*, 2001) was an experiment-based control of quadruple tank Table 3.1:values for tank simulation} and Table 3.2 adapted from it were used for parameter estimation of the model developed in the work. The general formation of the realization of the nonlinear state space equation is given by:

$$\begin{aligned} \dot{h} &= f(h,u) \\ y &= g(x,u) \end{aligned} \tag{3.14}$$

is linearised to

$$\dot{h} = f(\bar{x},\bar{u}) \tag{3.15}$$

The output comprises of linear system matrices given by:

$$A = \left[\frac{\partial f}{\partial x}\right]; B = \left[\frac{\partial f}{\partial u}\right]; C = \left[\frac{\partial g}{\partial x}\right]; D = \left[\frac{\partial g}{\partial u}\right]; \tag{3.16}$$

Where A, B, C and D are Jacobian matrices under the conditions of state x (i.e. initial conditions). This is obtained by the implementation of equation(3.14) and application of equation(3.15) on equations(3.10) - (3.13) which formed the nonlinear system equation of the quadruple tank system.

Hence, the approximated state space model is

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{3.17}$$

Represented thus in states space as

$$\dot{x} = \begin{pmatrix} -\frac{1}{T_1} & 0 & \frac{1}{T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{1}{T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{pmatrix} x + \begin{pmatrix} \frac{\gamma_1 \kappa_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 \kappa_2}{A_2} \\ 0 & \frac{(1-\gamma_2)\kappa_2}{A_3} \\ \frac{(1-\gamma_1)\kappa_1}{A_4} & 0 \end{pmatrix} u \tag{3.18}$$

$$y = \begin{pmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{pmatrix} x \tag{3.19}$$

With $x = [h_1, h_2, h_3, h_4]^T$ , $u = [\upsilon_1 . \upsilon_2]^T$ and $y = [h_1, h_2]^T$

Where the time constant calculation is done with the use of:

$$T_i = \frac{A_i}{a_i} \sqrt{\frac{2h_i^0}{g}} \tag{3.20}$$

Where i = 1,2,3,4.

A is the cross-sectional area of tank,

a; is the cross-sectional area of pipe,

h; is the height of liquid in tank,

g; is the acceleration due to gravity.

## 3.1 Tank Parameters and Control Test

The parameter values implemented in Matlab code are based on the values from laboratory process used in (Johansson, 2000) given in Table 3.1

Table 3.1: Values for Simulation

| **Parameters** | **Values** |
|---|---|
| A1, A2 ($cm^2$) | 28 |
| A2, A3 ($cm^2$) | 32 |
| a1, a3 ($cm^2$) | 0.071 |
| a2, a4 ($cm^2$) | 0.057 |
| k(V/cm) | 1.00 |
| g ($cm/s^2$) | 981 |

The model and control of the quadruple-tank process are studied at two operating points shown in Table 3.2 points: P- at which the system will be shown to have minimum-phase characteristics and P+ at which it will be shown to have nonminimum-phase characteristics.

The chosen operating points correspond to the following parameter values,

Table 3.2: Operating Points Used

| Parameters | P- | P+ |
|---|---|---|
| $(h_1^0, h_2^0)$ (cm) | (12.4, 12.7) | (12.6, 13.0) |
| $(h_3^0, h_4^0)$ (cm) | (1.8, 1.4) | (4.8, 4.9) |
| $(\upsilon_1^0, \upsilon_0^0)$ (V) | (3.00, 3.00) | (3.15, 3.15) |
| (k1, k2)($cm^3$/Vs) | (3.33, 3.35) | (3.14, 3.29) |
| $(\gamma 1, \gamma 2)$ | (0.70, 0.60) | (0.43, 0.34) |

#### *3.3.1.1 Minimum Phase and Nonminimum Equations of System*

This subsection show minimum phase behaviours of the system in two ways;

**A.** Transfer function in minimum phase.

**B.** State Space in minimum phase.

**C.** Transfer function in nonminimum phase.

**D.** State Space in nonminimum phase.

**A. Transfer function in minimum phase:**

$G11_{num}$ = [2.6]; $G11_{den}$ = [62 1];

$G12_{num}$ = [1.5]; $G12_{den}$ = [1426 85 1]

$G21_{num}$ = [1.4]; $G21_{den}$ = [2700 120 1];

$G22_{num}$ = [2.5]; $G22_{den}$ = [90 1];

Sys1 = tf ($G11_{num}$, $G11_{den}$)

Sys2 = tf ($G12_{num}$, $G12_{den}$)

Sys3 = tf ($G21_{num}$, $G21_{den}$)

Sys4 = tf ($G22_{num}$, $G22_{den}$)

H = [Sys1 Sys2 Sys3 Sys4]; Continuous transfer function

Hd = c2d (H, 1); discrete transfer function

The respective output give:

H=

From input 1 to output…

**1:** $\dfrac{2.6}{62s+1}$

**2:** $\dfrac{1.4}{2700s^2+120s+1}$

**From input 2 to output…**

**1:** $\dfrac{1.5}{1426s^2+85s+1}$

**2:** $\dfrac{2.5}{90s+1}$

**Continuous-time function**

**Hd =**

**From input 1 to output**

**1:** $\dfrac{0.00416}{z-0.984}$

**2:** $\dfrac{0.0002555z+0.0002517}{z^2-1{,}956z+0.9565}$

**From input 2 to …**

**1:** $\dfrac{0.0005156z+0.0005055}{z^2-1.941z+0.9421}$

**2.** $\dfrac{0.022762}{z-0.989}$

Sample time: 1 seconds (using zero order hold method of sampling)

Discrete-time transfer function.

**B. State Space in minimum phase:**

$$A = \begin{pmatrix} 0.9842 & 0 & 0.0407 & 0 \\ 0 & 0.9890 & 0 & 0.0326 \\ 0 & 0 & 0.9590 & 0 \\ 0 & 0 & 0 & 0.9672 \end{pmatrix} \tag{3.21}$$

$$B = \begin{pmatrix} 0.0826 & 0.0010 \\ 0.0005 & 0.0625 \\ 0 & 0.0469 \\ 0.0307 & 0 \end{pmatrix} \tag{3.22}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{3.23}$$

$$D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{3.24}$$

The controllability test using the Matlab13b control toolbox, gives

$$Ac \;\; = \begin{pmatrix} 0.0826 & 0.0010 & 0.0813 & 0.0029 & 0.0800 & 0.0047 & 0.0787 & 0.0063 \\ 0.0005 & 0.0625 & 0.0015 & 0.0618 & 0.0025 & 0.0611 & 0.0034 & 0.0604 \\ 0 & 0.0469 & 0 & 0.0449 & 0 & 0.0431 & 0 & 0.0413 \\ 0.0307 & 0 & 0.0297 & 0 & 0.0287 & 0 & 0.0278 & 0 \end{pmatrix}$$

(3.25)

With a rank of 4, equalling the number of rows: proving that our system is controllable.

**C. Transfer function in nonminimum phase:**

$G11_{numm}$ = [1.5];

$G11_{denm}$ = [63 1];

$G12_{numm}$ = [2.5];

$G12_{denm}$ = [2457 102 1];

$G21_{numm}$ = [2.5];

$G21_{denm}$ = [5096 147 1];

$G22_{numm}$ = [1.6];

$G22_{denm}$ = [91 1];

sysm1 = tf ($G11_{numm}$, $G11_{denm}$)

sysm2 = tf ($G12_{numm}$, $G12_{denm}$)

sysm3 = tf ($G21_{numm}$, $G21_{denm}$)

sysm4 = tf ($G22_{numm}$, $G22_{denm}$)

Hm = [sysm1 sysm2; sysm3 sysm4]; continuous transfer function

Hdm = c2d (Hm, 1); discrete transfer function

Hm =

From input 1 to output...

$$1: \frac{1.5}{63s+1}$$

$$2: \frac{2.5}{5096s^2+147s+1}$$

From input 2 to output...

$$1: \frac{2.5}{2457s^2+102s+1}$$

$$2: \left(\frac{1.6}{91s+1}\right)$$

Continuous-time transfer function.

Hdm =

From input 1 to output...

$$1: \frac{0.02362}{z-0.9842}$$

$$2: \frac{0.0002429z+0.0002406}{z^2-1.971z+0.9716}$$

From input 2 to output...

$$1: \frac{0.0005018z+0.0004949}{z^2-1.959z+0.9593}$$

$$2: \frac{0.01747}{z-0.9891}$$

Sample time: 1 seconds

Discrete-time transfer function.

**C. States Space in nonminimum phase:**

$$An = \begin{pmatrix} 0.9843 & 0 & 0.0251 & 0 \\ 0 & 0.9892 & 0 & 0.0176 \\ 0 & 0 & 0.9747 & 0 \\ 0 & 0 & 0 & 0.9824 \end{pmatrix} \tag{3.26}$$

$$Bn = \begin{pmatrix} 0.0478 & 0.0010 \\ 0.0005 & 0.0348 \\ 0 & 0.0765 \\ 0.0554 & 0 \end{pmatrix} \tag{3.27}$$

$$Cn = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{3.28}$$

$$Dn = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{3.29}$$

The controllability test using the Matlab2013b control toolbox, results to

$$Anc = \begin{pmatrix} 0.0478 & 0.0010 & 0.0470 & 0.0029 & 0.0463 & 0.0047 & 0.0456 & 0.0065 \\ 0.0005 & 0.0348 & 0.0015 & 0.0344 & 0.0024 & 0.0341 & 0.0033 & 0.0337 \\ 0 & 0.0765 & 0 & 0.0746 & 0 & 0.0727 & 0 & 0.0708 \\ 0.0554 & 0 & 0.0544 & 0 & 0.0535 & 0 & 0.0525 & 0 \end{pmatrix} \quad (3.30)$$

The rank obtained is 4 and this shows that our system is controllable. The rank shows that at least 4 columns are independent of each other.

The control test was performed in Matlab Window using the code

$A_{bar}, B_{bar}, C_{bar}, T, k] = ctrbf\,(A, B, C)$

In this section a demonstration of the use of bond graph in modelling of a multiprocess system, in this case a quadruple tank system was successfully carried out. The modelling which was carried out based on energy characteristics of the lumped parameters of the system effectively captured the system dynamics in a visualized and graphical manner. Hence, the final system is nonlinear and exhibits the constraint of natural tank system with its dynamics, hence requires a robust and efficient control algorithm/technique minimize its cost objective, while meeting its target, effectively.

### 3.3.2 Input - Output Pairing

The selection of proper pairing depicts a very important step in multivariable control design (Luyben, 1970). This is often achieved via Relative Gain Array (RGA) calculation. The RGA, attempts to show the inputs and outputs that their pairing will not violate control objectives. It is employed to the minimum phase and non-minimum phase tank scenario. RGA is calculated thus (Vijula & Devarajan, 2013).

$$\Lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix} \quad (3.31)$$

Where

$$\lambda_{11} = 1 - \lambda_{12} \quad (3.32)$$

$$\lambda_{11} = \frac{G_{11}(s)G_{22}(s)}{G_{11}(s)G_{22}(s) - G_{12}(s)G_{21}(s)} \quad (3.33)$$

This essentially means that:

1. For large values of $\lambda$ , the diagonal of the transfer function will dominate i.e. output (1); $h_1$ will be most affected by input (1); ($u_1$), while output (2); ($h_2$) will be affected by input (2); ($u_2$).
2. For small values of $\lambda$ output (1); ($h_1$) will depend on input (2); ($u_2$) and output (2); ($u_2$) on input (1); ($u_1$).

Hence, from the tank parameters on minimum phase, $\lambda_{11} = 0.63$; ($0.5 < \lambda_{11} < 1$) so that the input/output pair is $h_1 \Rightarrow u_1$ and $h_2 \Rightarrow u_2$, while in non-minimum phase, $\lambda_{11} = 0.375$; ($0.375 < \lambda_{11} < 0.5$) so that the input/output pair is $h1 \Rightarrow u_2$ and $h_2 \Rightarrow u_1$

### 3.3.3 Decoupling Derived Linear Time Invariant Model

In ideal decoupler design, the assumption is that, the decoupling elements independent of the feed-forward path controllers (Potts *et al.*, 2015), hence the on-line tuning does not require redesigning of decoupler elements (Liu *et al.*, 2013; Vijula & Devarajan, 2013), while the diagonals are taken as unity. Hence the steps for designing a suitable decoupler comprises

(i) Start with K = I, then D = adj (G) (s).

(ii) Remove the largest common time delay of each decoupler column by multiplying the corresponding diagonal elements of K by inverted time delays (if there is time delay).

(iii) Remove common poles and zeros of the decoupler columns by multiplying the corresponding diagonal elements of K by the inverse of the poles and zeros possibly also multiply the diagonal elements of K by low-pass filter to avoid giving the decoupler high pass filter characteristics (Nordfeldt & Hägglund, 2006)

Therefore a decoupler matrix is given by:

$$D(s) = \begin{pmatrix} 1 & D_{11}(s) \\ D_{21}(s) & 1 \end{pmatrix} \tag{3.34}$$

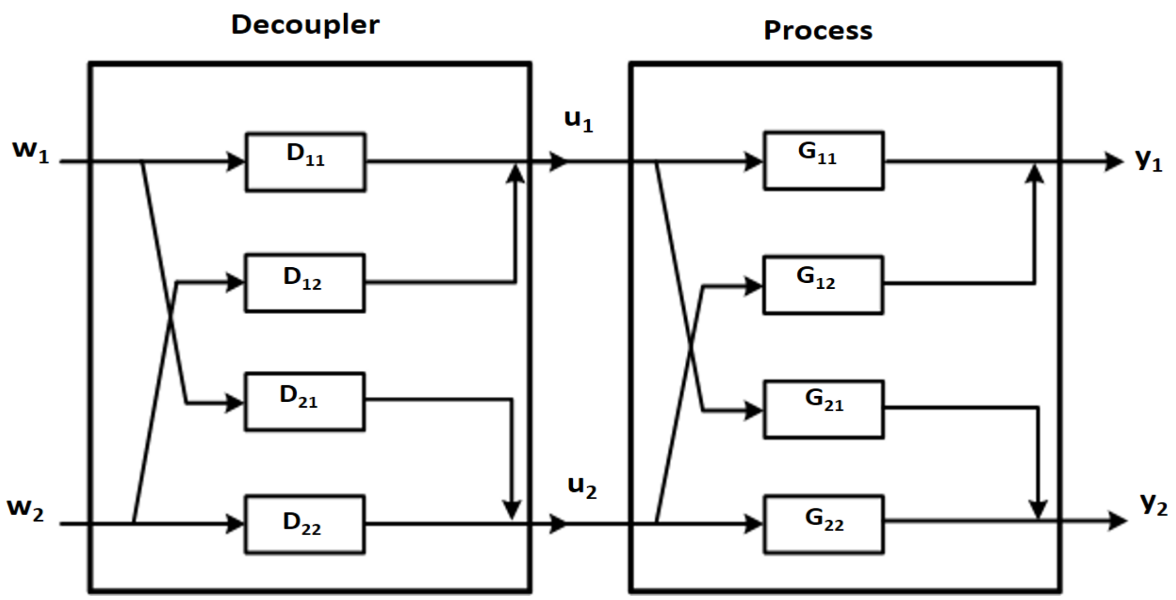



Figure 3.8: The Ideal Decoupling Structure

The structure of the decoupler and process are presented as in Figure 3.8. The $G_{ij}(s)$ - $G_{ii}$ are the plants subsystems while $D_{ij}$ - $D_{ii}$ are the decoupler subsystems. Hence, the complete design in Simulink is as show in Figure 3.9.

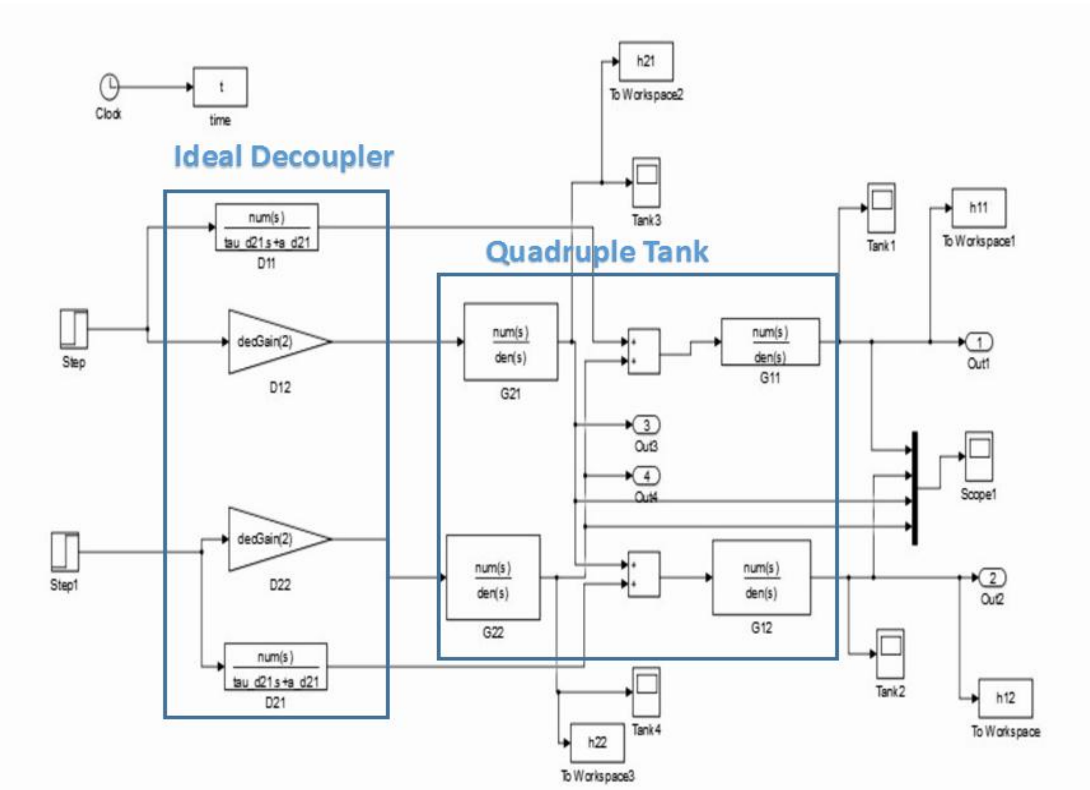



Figure 3.9: The Open-loop Decoupled System Structure with Simulink

Figure 3.9 shows the connectivity of the entire system, G(s); $G_{11}(s)$, G12(s), $G_{21}(s)$, $G_{22}(s)$ and the decoupler, D(s); $D_{11}(s)$, $D_{12}(s)$, $D_{21}(s)$, $D_{22}(s)$. The product of the System, G(s) and Decoupler, G(s) yields G(s)*D(s). Hence, the off diagonal elements are equal to zero equating the product of the G(s)*D(s) matrices to SISO system; $W_{siso}(s)$ which is the diagonal elements of the products of $D_{12}(s)$ and $D_{21}(s)$. Thus,

$$D_{11}(s) = D_{22}(s) = 1 \tag{3.35}$$

$$D12(s) = -\frac{G_{12}(s)D_{22}(s)}{G_{11}(s)},\ D_{21}(s) = -\frac{G_{21}(s)D_{22}(s)}{G_{22}(s)} \tag{3.36}$$

s.t

$$W_{siso}(s) = diag[W_1(s), W_2(s)] \tag{3.37}$$

As such, from Figure 3.9, minimum phase (min) and nonminimum phase (Nmin) matrices are derived as shown in equations (3.38) and (3.40).

Figure 3.9 is the Simulink design of Figure 3.8. The decouplers and the process are as shown in the figure. The scopes are used in visualizing the outputs of the system at designated nodes.

Hence, the decouplers' algorithm are obtained as follows:

$$D_{min}(s) = \begin{pmatrix} 1 & -\dfrac{G_{12min}(s)D_{11min}(s)}{G_{11min}(s)} \\ -\dfrac{G_{21min}(s)D_{11min}(s)}{G_{22min}(s)} & 1 \end{pmatrix} \tag{3.38}$$

$$D_{min}(s) = \begin{pmatrix} 1 & -\dfrac{93s+1.5}{3708s^2+221s+2.6} \\ -\dfrac{126s+1.4}{6750s^2+300s+2.5} & 1 \end{pmatrix} \tag{3.39}$$

While

$$D_{Nmin}(s) = \begin{pmatrix} 1 & -\dfrac{G_{12Nmin}(s)D_{11Nmin}(s)}{G_{11Nmin}(s)} \\ -\dfrac{G_{21Nmin}(s)D_{11Nmin}(s)}{G_{22Nmin}(s)} & 1 \end{pmatrix} \tag{3.40}$$

$$D_{Nmin}(s) = \begin{pmatrix} 1 & -\dfrac{157.3s+2.5}{3686s^2+153s+1.5} \\ -\dfrac{227s+1.5}{8154s^2+235.2s+1.6} & 1 \end{pmatrix} \tag{3.41}$$

# CHAPTER FOUR

# RESULTS AND DISCUSSIONS

## 4.1 INTRODUCTION

This chapter sums up the results of this work. Included are the challenges, constraints, and graphical analysis of the entire sub-systems and the solution. The results includes those obtained without controller action, those obtained during the application of classical controller based on Dynamic Matrix Control and Quadratic Programmed based Controllers.

## 4.2 OPEN-LOOP RESPONSE OF MIMO QUADRUPLE TANK SYSTEM

The open-loop response of the MIMO system which comprises of first order sub-system tanks (1-4) is obtained by simulation of the system at initial state [$h_1$ = 0cm, $h_2$ = 0cm, $h_3$ = 0cm, $h_4$ = 0cm] as shown in Figure 4.1 and agitated states as also shown in Figure 4.2

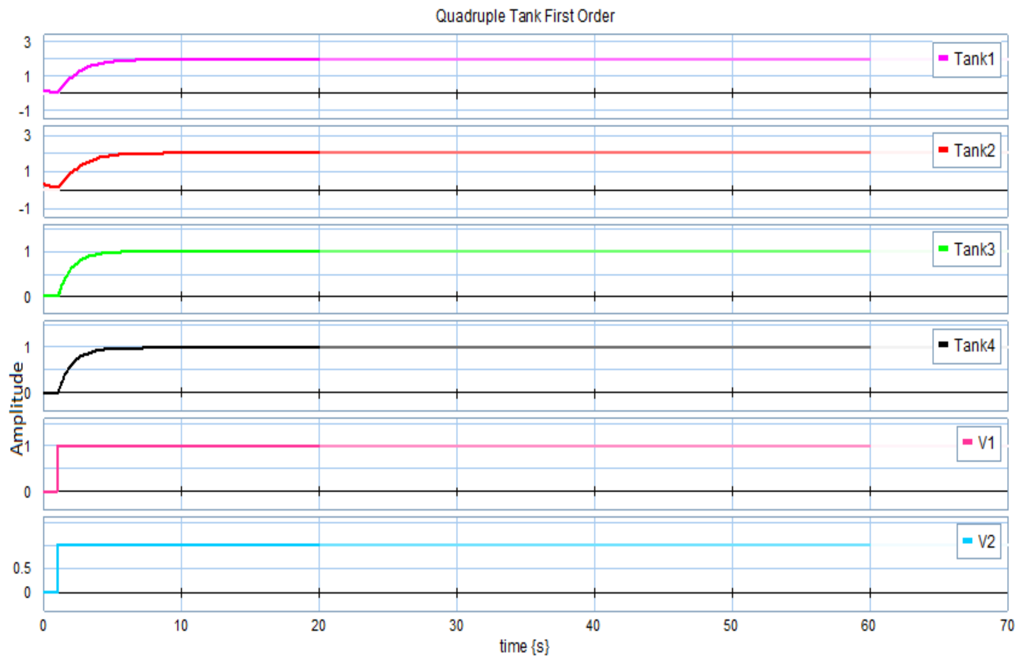



Figure 4.1: 20-sim Step Response of Open-loop System

Furthermore, the system was subjected to several perturbations of which the system behaviour depicts accuracy in plants' model at equilibrium point of (0; 0; 0; 0)

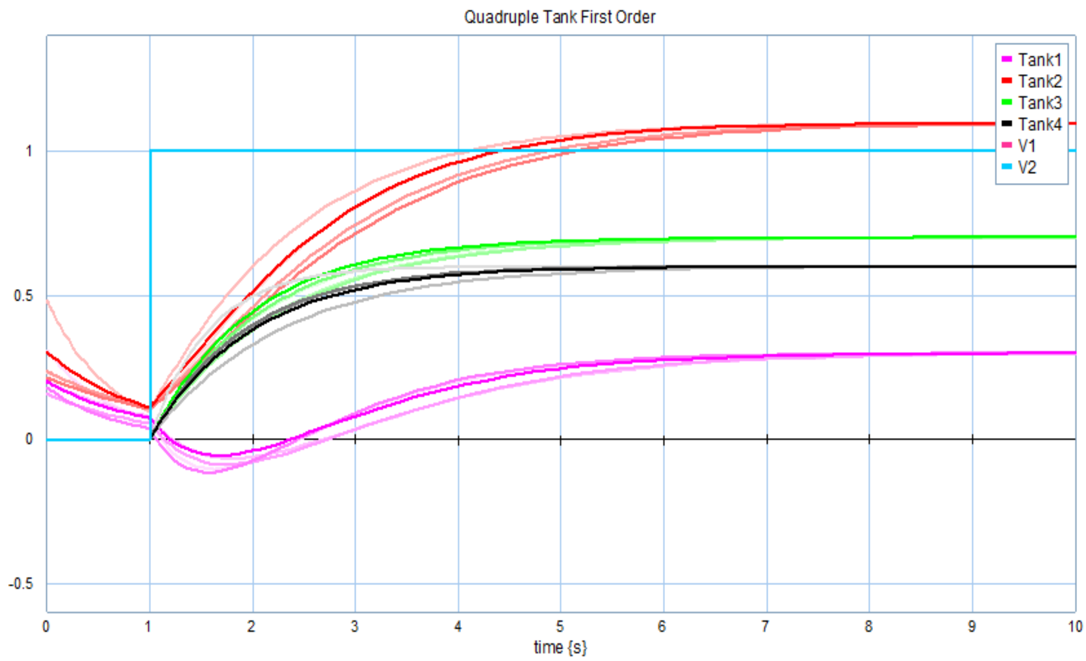



Figure 4.2: 20-sim Step Response under Multiple Excitations

Subsequently this system was linearized. The resultant MIMO tank system's response at minimum and nonminimum phase.

is given below.

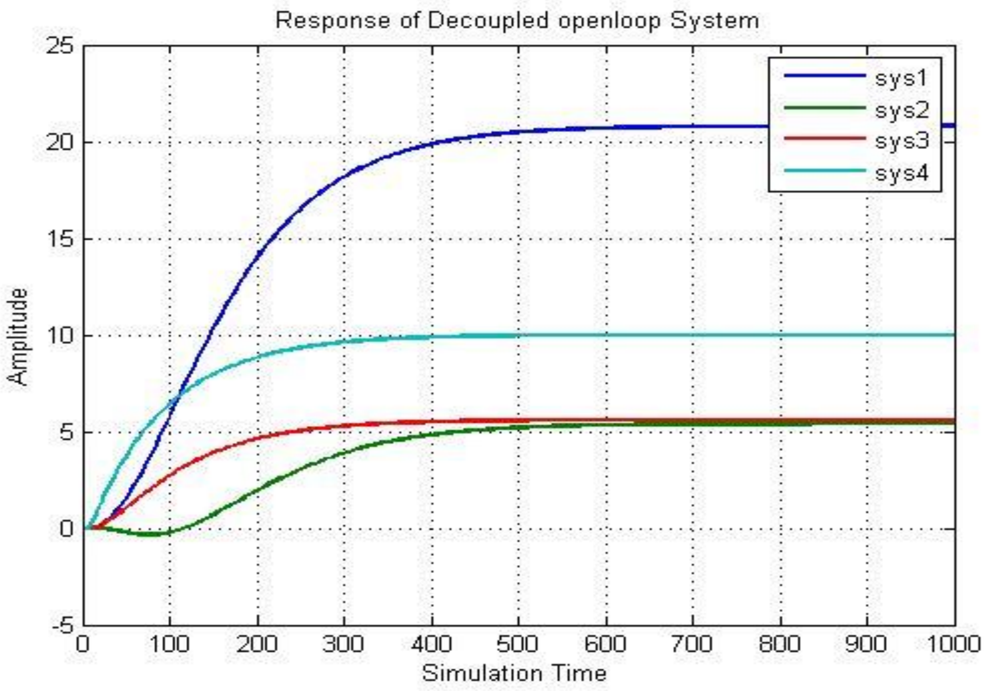



Figure 4.3: Open-Loop Minimum /Nonminimum Phase Step Response

Figure 4.3 displays the responses of tank (2) having a right plane zero, as against tank (1) in minimum phase state without a right-plane zero (transmission zero).

### 4.3 CONTROLLERS AND APPLICATION

This section presents the application the DMPC variants and the PID controllers. A comparison is drawn to validate the control strategies.

### 4.3.1 Decentralized Model Predictive Control Application on Quadruple Tank System

The simulation is carried out within Matlab/Simulink. The design and implementation representation incorporates the Decentralized Model Predictive Controller implementation and Simulation labelled "MyTank12.mdl" as carried-out with the control toolbox as shown in Figure 4.4.

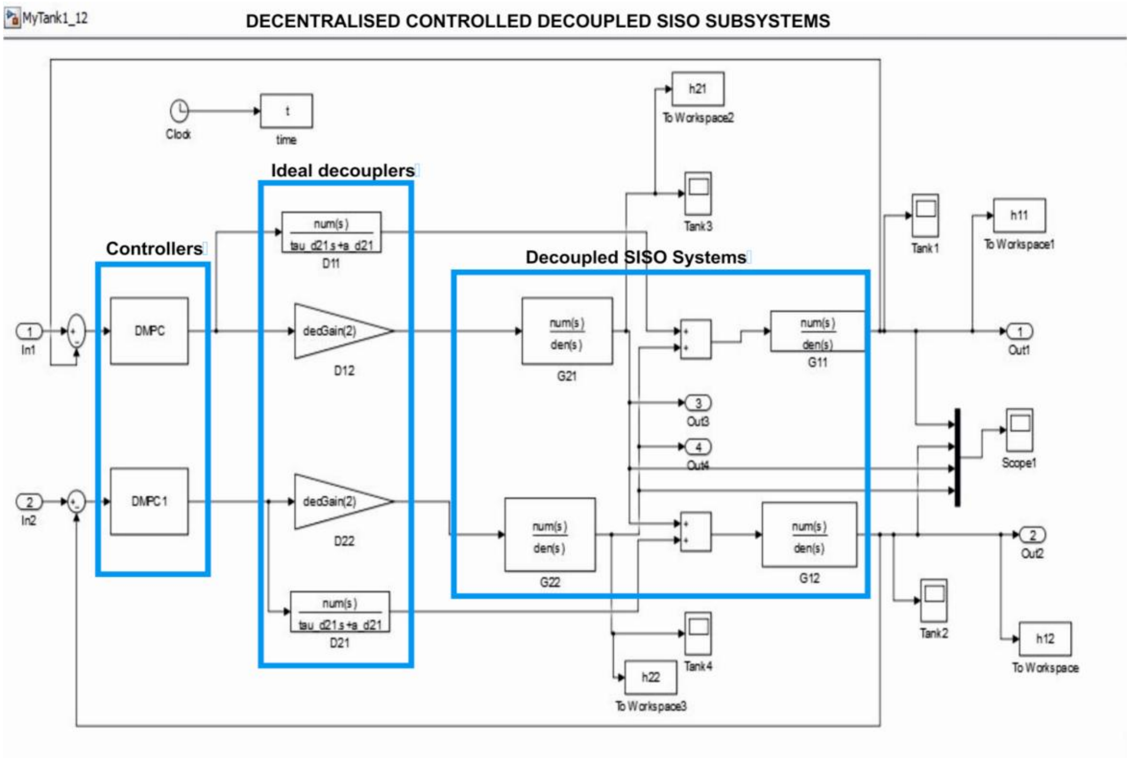



Figure 4.4: Schematic of the SISO Control Subsystems of the Quadruple Tank

### 4.3.2 Results for DMPC

This subsection, gives the brief on the results obtained during the simulations of system in Figure 4.4 produced in MatlabR2013b.

Dynamic Matrix Control, carried out without constraint for the minimum-phase (Tank1 and Tank2) and non-minimum phase are as shown in Figures 4.5 to 4.8. The system demonstrated stable behaviour up until the prediction horizon increased beyond $t \geq 5$ (in this case) i.e. at large command position.

The random noise was generated using the matlab code:

*rand ('state', 0);.*

*noise = sd*randn (kfinal, 1);*

The standard deviation *sd = 0.1*;

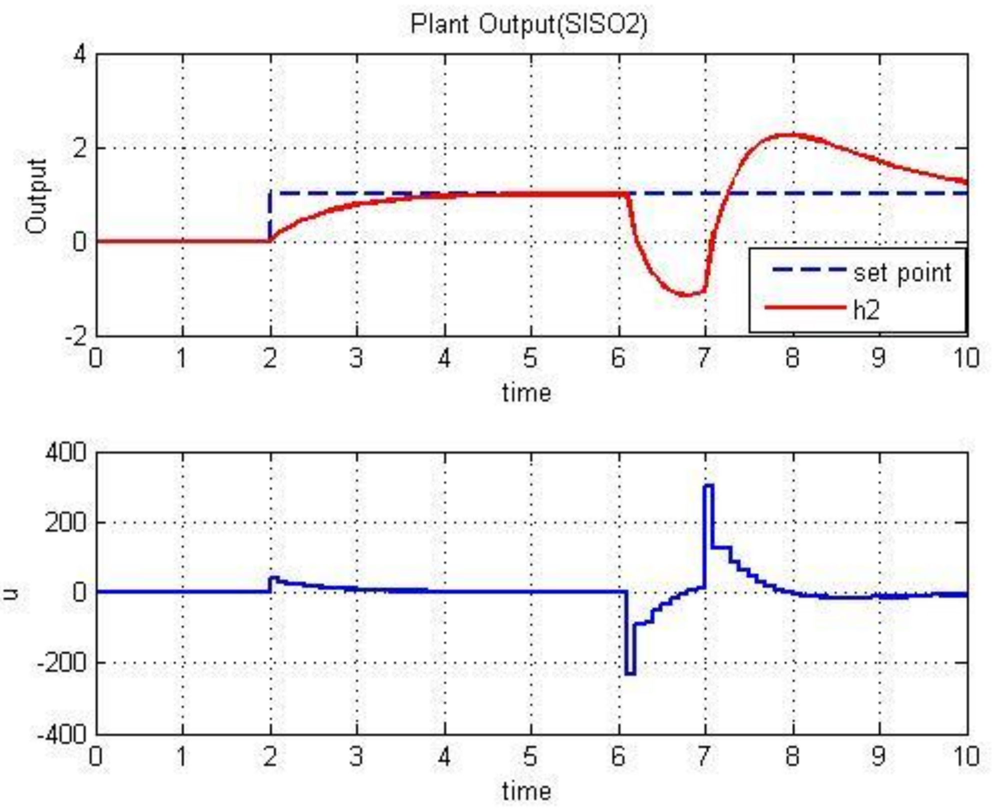



Figure 4.5: Non-Minimum Phase Plot of DMC Scheme h2

The objective function used in the optimization based on classical predictive control is least mean square error. This can lead to challenges if numerically ill-conditioned situations arise. This resulted in the system of Figure 4.5 tending towards divergent results, hence instability is observed at a receding horizon of $t \geq 6$ (control horizon; M=1, prediction horizon; N=5 and sampling time; ts = 10sec) with a spike (influence of transmission zero).

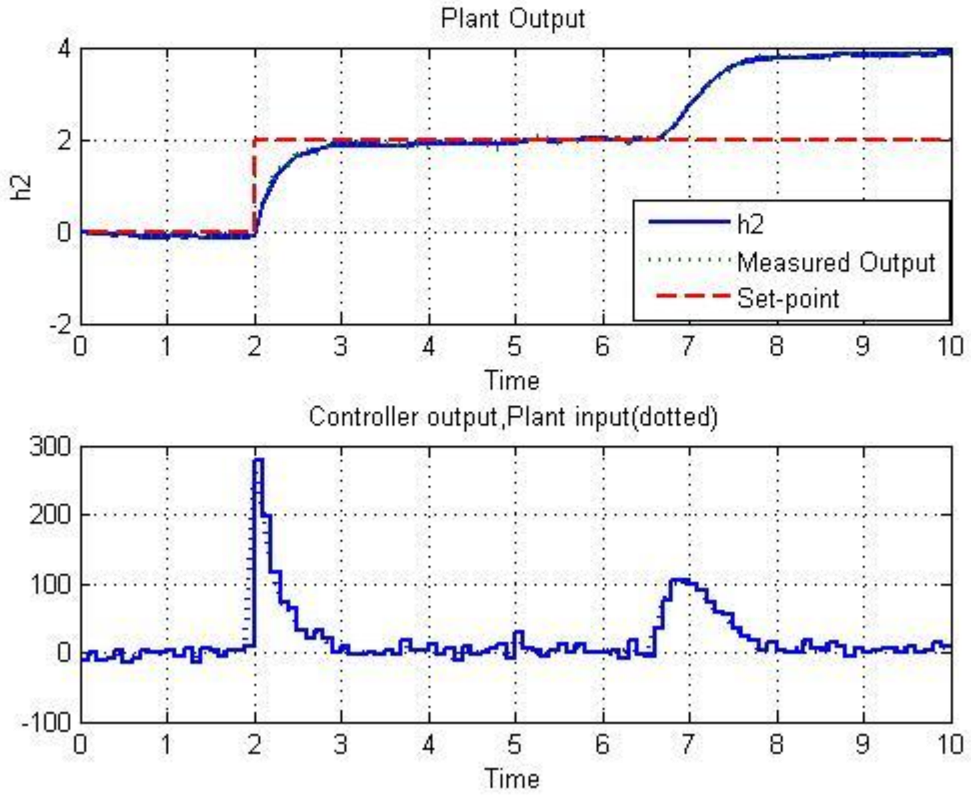



Figure 4.6: Minimum Phase Plot of DMC Scheme h2

The response of Figure 4.6 was obtained using specifications as N =1, M = 5, and time 10sec. The result showed numerically ill-conditioned situation with instability.

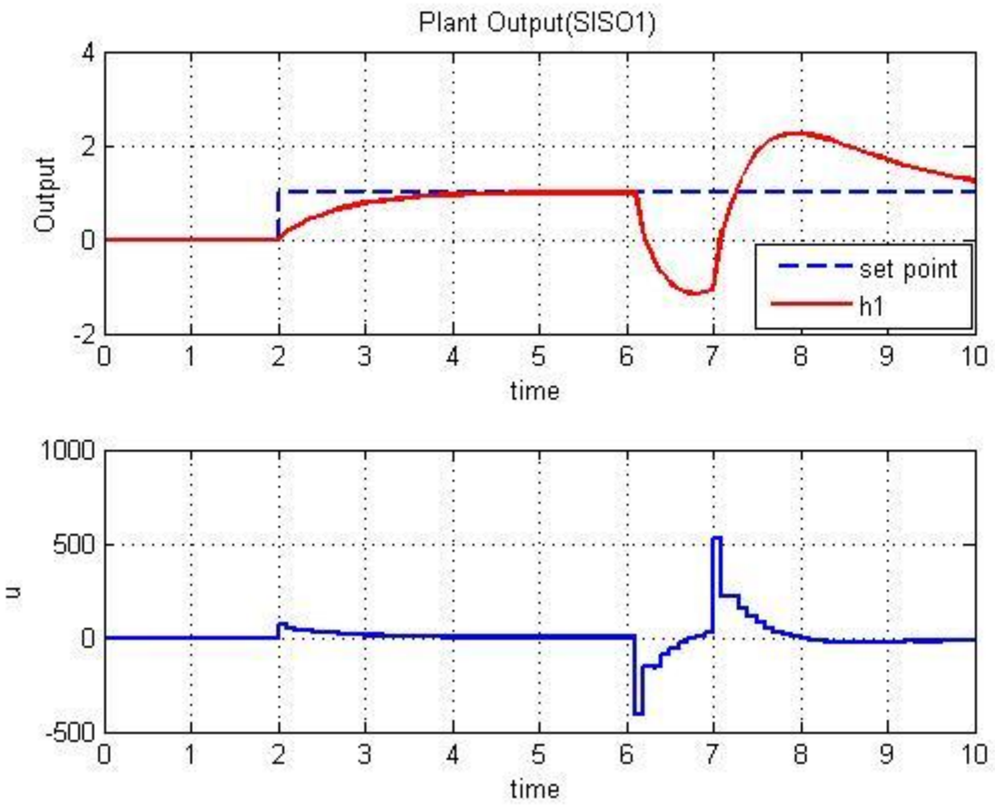



Figure 4.7: Nonminimum Phase Plot of DMC Scheme h1

Figure 4.7 is the nonminimum phase response of the subsystem tank 1. In the non-minimum phase, the controller had to further spend a little more time to optimise the solution as could be observed in the graphs. The effect of the transmission zero could be observed at $t \geq 6$.

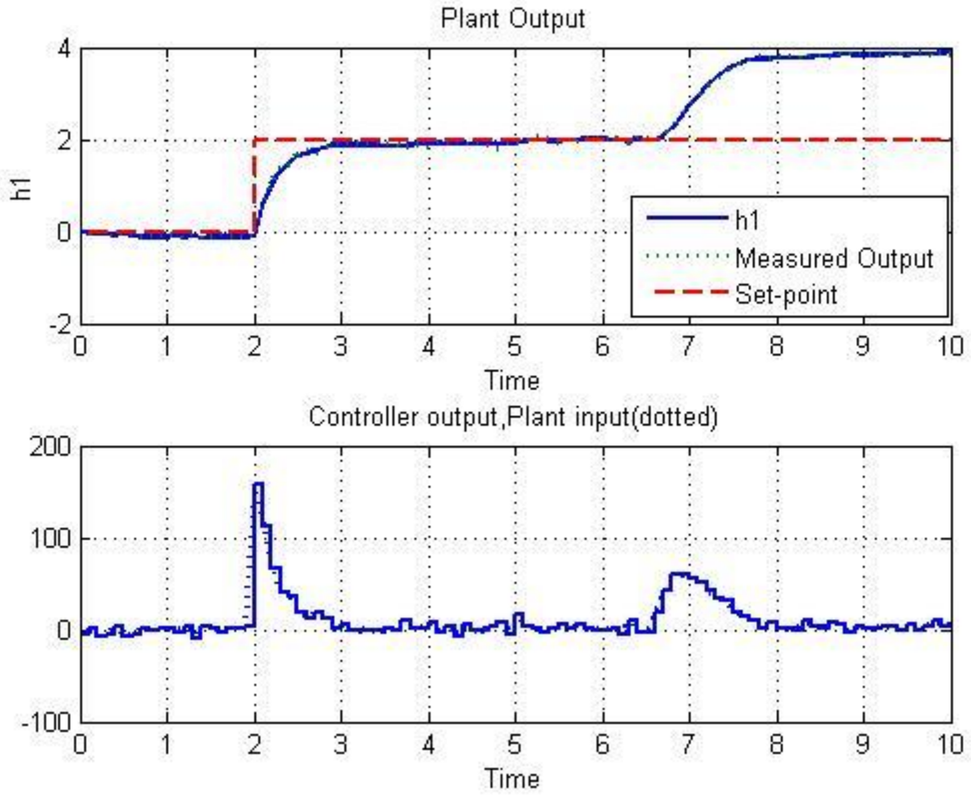



Figure 4.8: Minimum Phase Plot of DMC Scheme h1

The design of the DMPC controller had been carried out based on classical predictive scheme. However, the solution in Figure 4.8, tends to experience divergence at a future time due to numerical ill-conditioning of the co-opted optimization function (least mean square) with the possibility of instability setting in at the increase of control horizon. This subsection, presents the results obtained during simulation of the system in MatlabR2013b.

The implementation of DMPC, via quadratic programming algorithm (QP) considers inequality constraint as set in the following script:

```
lb = -0.1*ones(n_in,1);
ub = 0.1*ones(n\_in,1);
Lb = kron(ones((M+1),1),lb);
Ub = kron(ones((M+1),1),ub);
```

The reference signals were generated by

*ref = ones(1,T);*

*ref(201:600) = -ref(201:600);*

while the noise was randomly generated from the output data by

*noise = 1e-2*cumsum(randn(T,n_out)')' + 1e-2*randn(T,n_out);*

*[bfilt,afilt] = butter(4,0.9);*

*noise = filter(bfilt,afilt,noise)';*

and the constraint by

*if* $i \leq 50$*, y(:,i) = y(:,i)+0.3*ones(n_out,1);*

*else if*

$i \geq 300$*, y(:,i)= y(:,i)+0.5*ones(n_out,1); end*

With a suitable scaling weighted exponential factor, an unstable design model could be substituted with the transformed design model that is stable with all poles inside the unit circle. Hence, as a result, the prediction horizon N can be selected to be sufficiently large without numerical problems. In essence, asymptotic closed-loop stability is guaranteed (Liu *et al.*, 2013). Hence, with N = Prediction horizon, M = control horizon, q and rr as weight matrices, the results are given as follows:

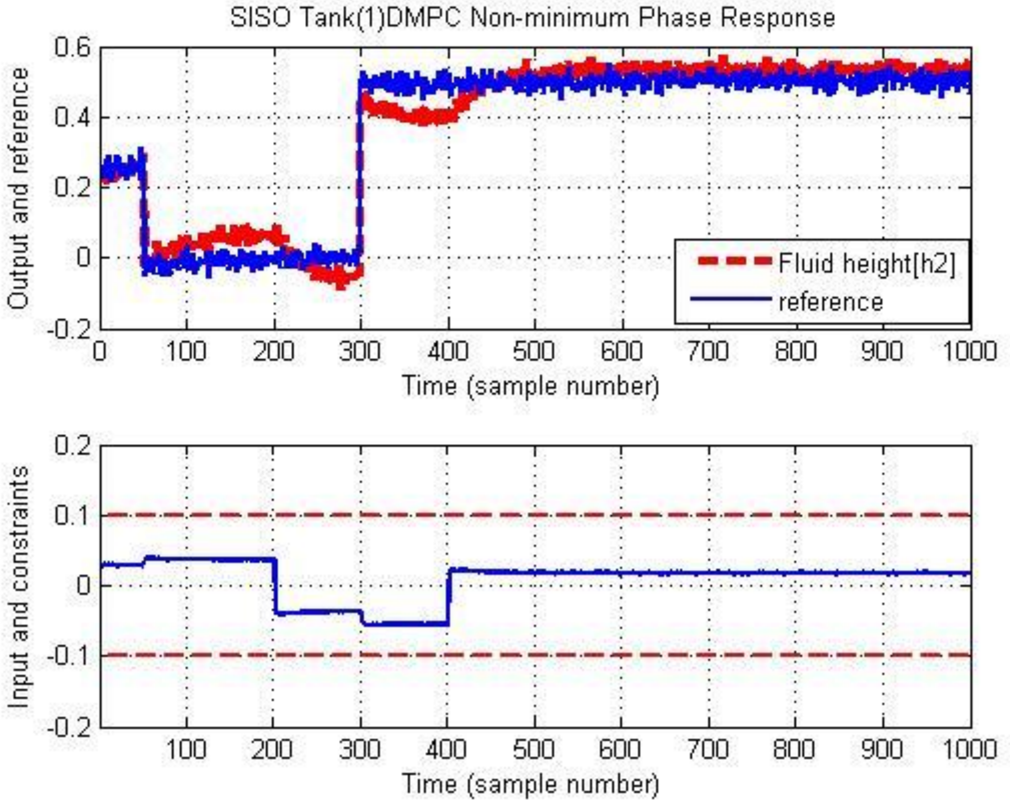



Figure 4.9: 2-Dimensional Nonminimum Phase Plot of Tank (1)-h2

With M = 1, N = 10, rr = 2. Height = h2, applied for control, the result is as shown in Figure 4.9 .The set-points realization increased the work of the controller as depicted by the deviations in the tracking error of the signal. This is as a result of the existence of transmission zero (right-hand plane zero)

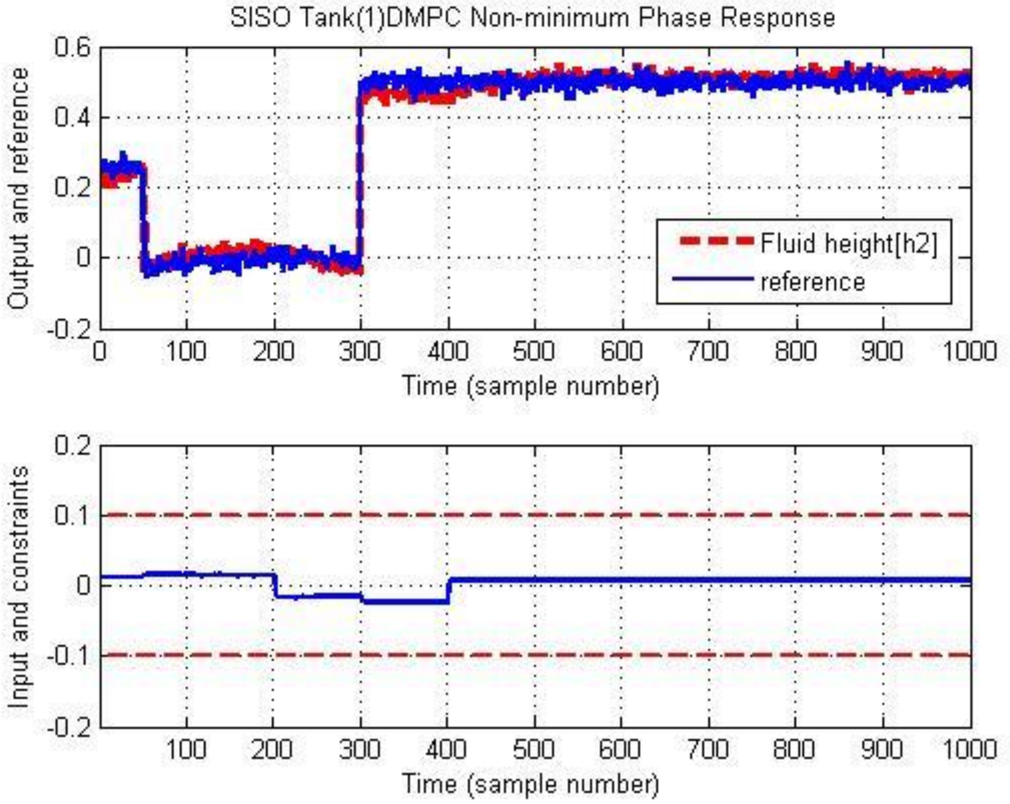



Figure 4.10: Non-Minimum Phase Plot of Tank (1)-h2

Figure 4.10 is the response obtained by application of the following parameters on the system control command; M = 3, N = 15, rr = 3 for height = h2. Set-point was reached with some penalty on time of convergence due to the existence of transmission zero (right-hand plane zero).

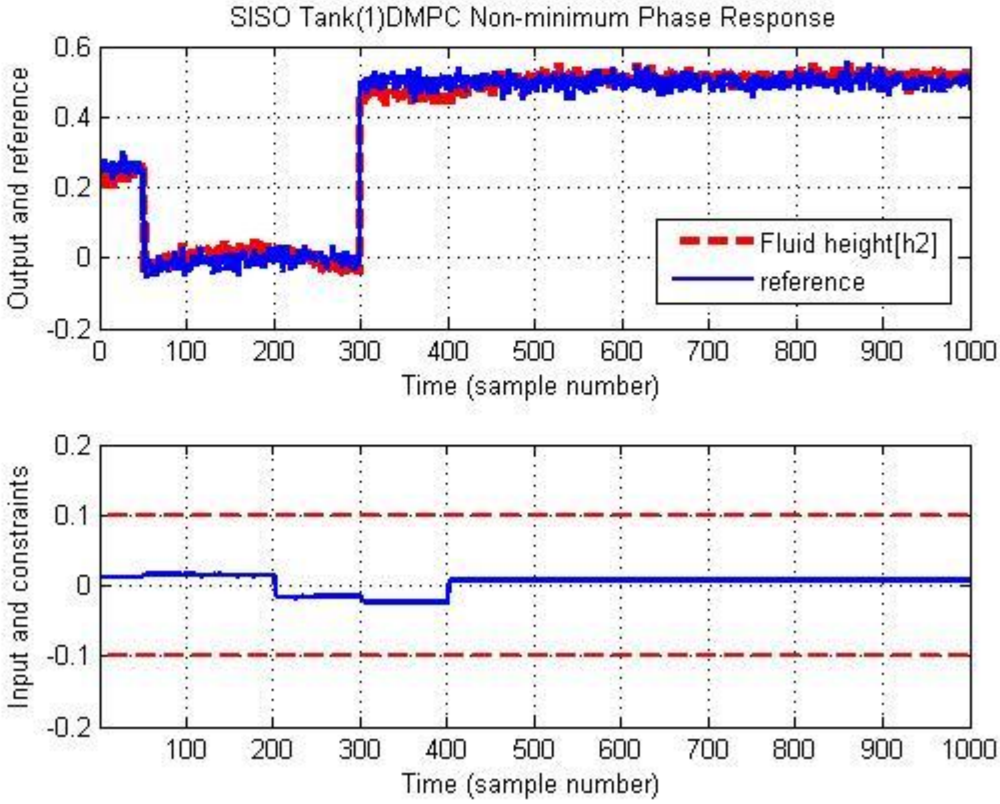



Figure 4.11: Non-Minimum Phase Plot of Tank (1)-h2

With M = 4, N = 20, rr = 3. Height = h2, responses of Figure 4.11 was obtained. Set-point was reached with some penalty on time of convergence due to the existence of transmission zero (right-hand plane zero).

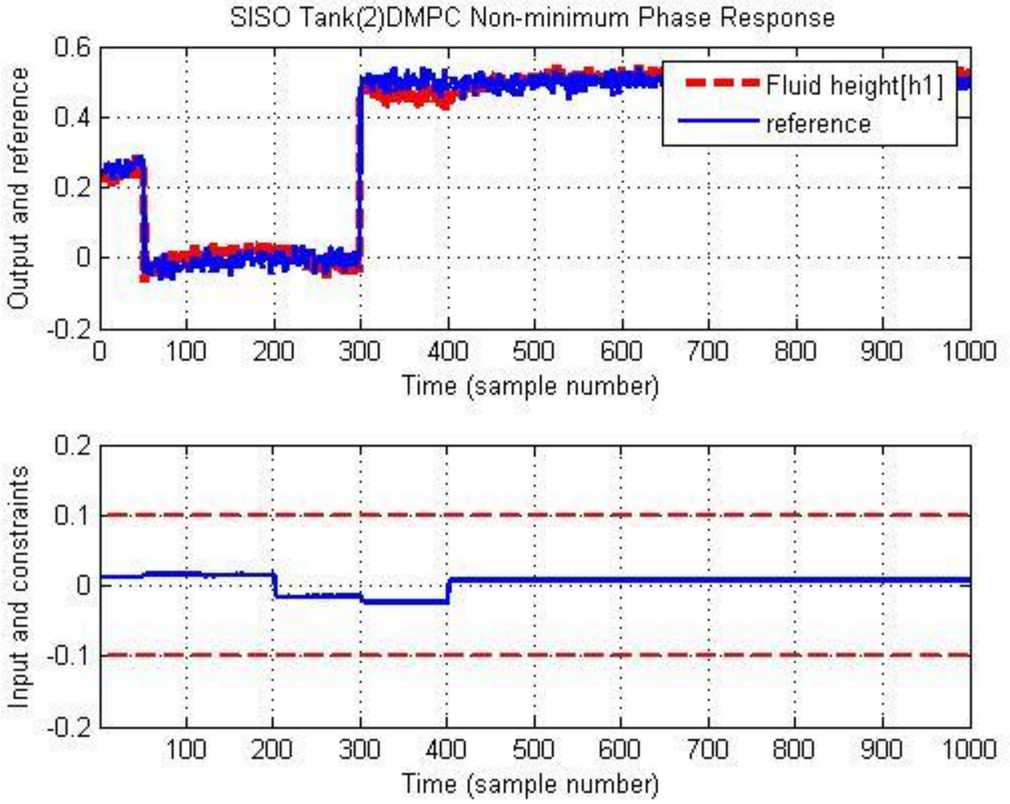



Figure 4.12 Non-Minimum Phase Plot of Tank (2)-h1

With M = 8, N = 28, rr = 4 and Height = h1 as parameters utilized in control command,

Figure 4.12 shows the system response with perfect realization and tractable solution. The set-points were achieved and the trajectory realized efficiently. The plant is stabilized as the numerical ill-conditioned problem was eliminated.

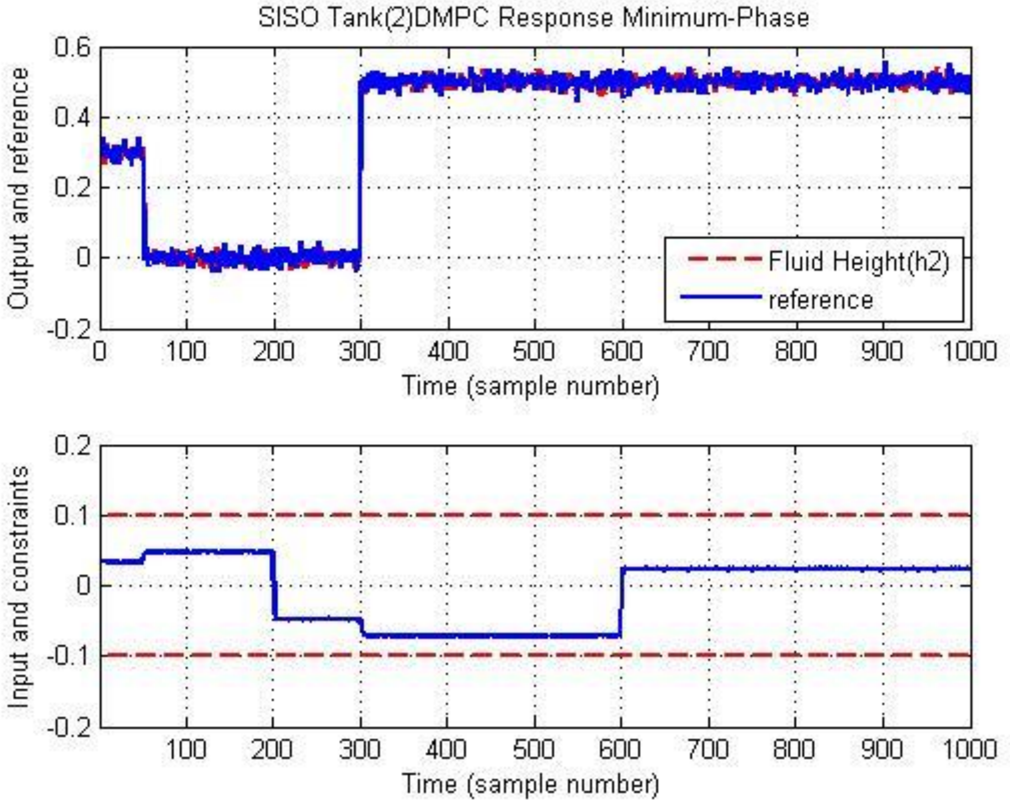



Figure 4.13: Minimum Phase Plot of Tank (2)-h2

Figure 4.13 shows the response of the system when the parameters: M = 10, N = 100, rr = 1 used to predict Height = h2. The solution was satisfactory as the trajectory was tracked effectively. The weighting matrices values were also less, hence lesser work done to achieve efficient results. The same constraint and disturbance as non-minimum case were also applied.

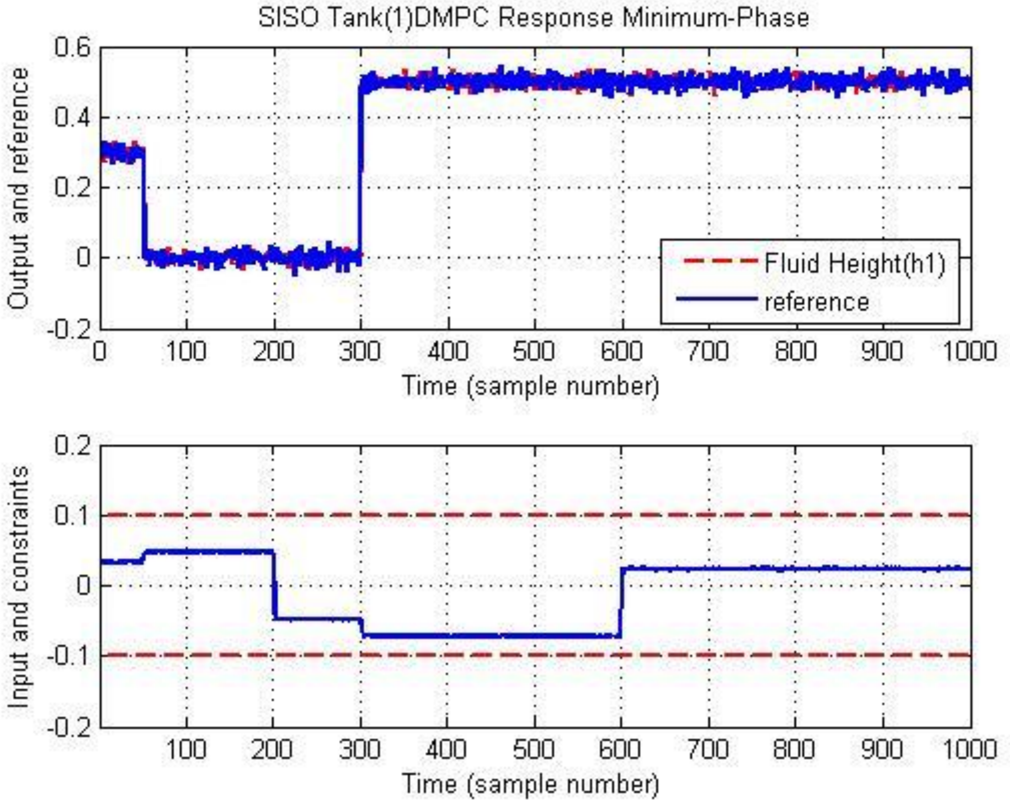



Figure 4.14: Minimum Phase Plot of Tank (1)-h1

Figure 4.14 shows the system response when the following parameters was set for the command: M = 10, N = 100, rr = 1, for a height h1. The controller predicted and exerted better control with lesser weighting matrices under the same constraint and disturbance. Generally, the Quadratic Programmed controller technique stabilized the system, even with constraints, disturbance and noise applied. The system was simulated at varied set-points which was effectively tracked as shown.

### 4.3.3 Results Proportional Integral Differential Controller Application

Use is made of the MATLAB/Simulink Control Toolbox for the design of this control application. Figure 4.15 shows the control layout used in PID application. The tuning parameters and the resultant effects were captured as shown on the right half part (data) of the figures showing the tuning parameters, the settling time, the phase and stability status of the system for minimum and nonminimum conditions. The plants were stable both in minimum phase and non-minimum phase conditions. Numerical analysis between the PID and MPC variants is captured further in the tables 4.1 and 4.2. The subsystems and decouplers retains the values as used in the QPDMPC where the G(s) represents the plants and D(s) represents the decouplers.

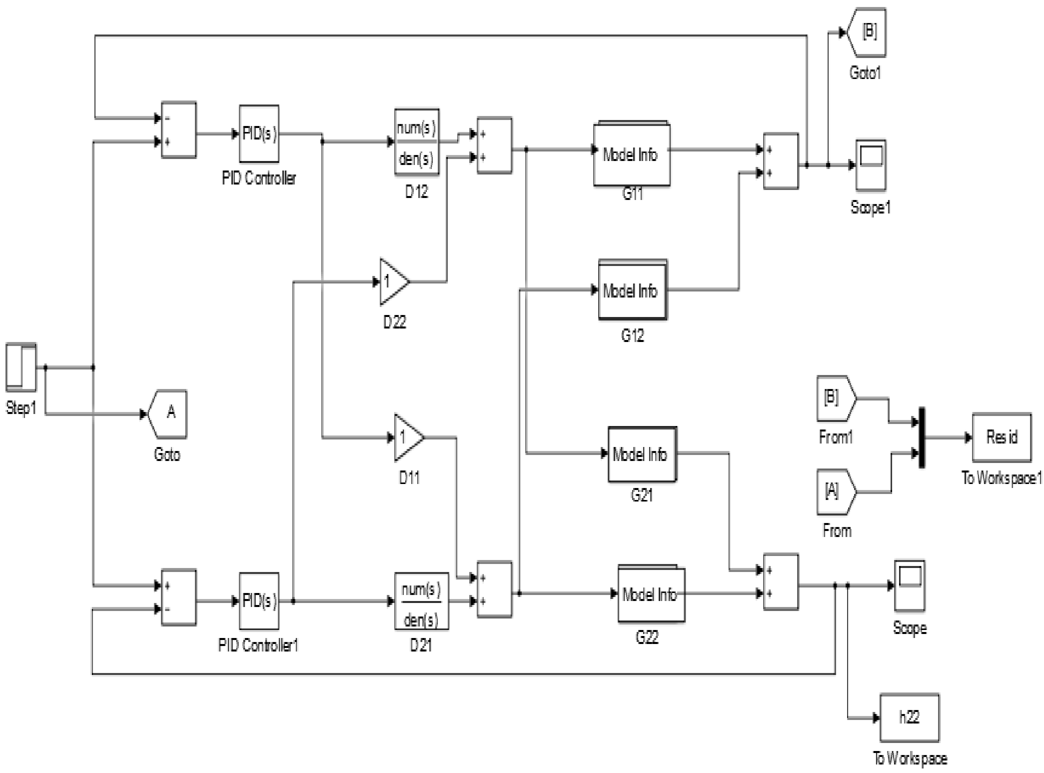



Figure 4.15: PID Control Configuration with Plant

In Figure 4.15, the controllers are the two PID's as presented with the source which is a step signal (reference value of 1).

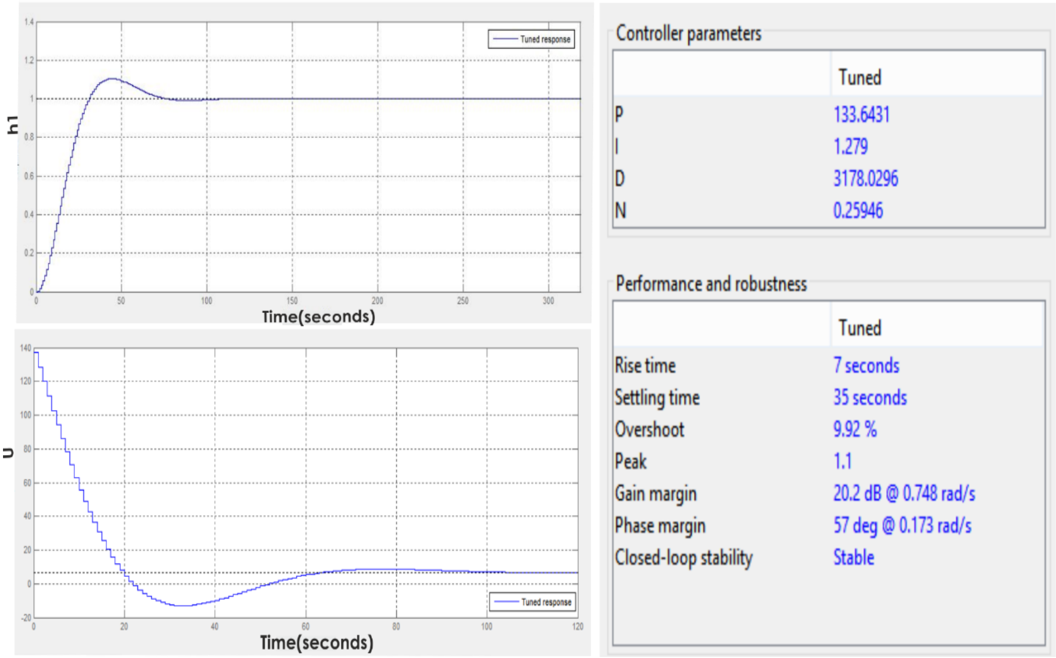



Figure 4.16: Minimum Phase (h1) Plot PID

The PID in minimum phase is tuned as represented on right half of Figure 4.16. The result as shown on the right half part of the above figure compared to Figure 4.8 and 4.14 of the DMC and QPDMPC respectively all of minimum phase performance, shows stable behaviours. Though the QPDMPC had a better performance as shown in table 4.1 for minimum phase characteristics of the controllers.

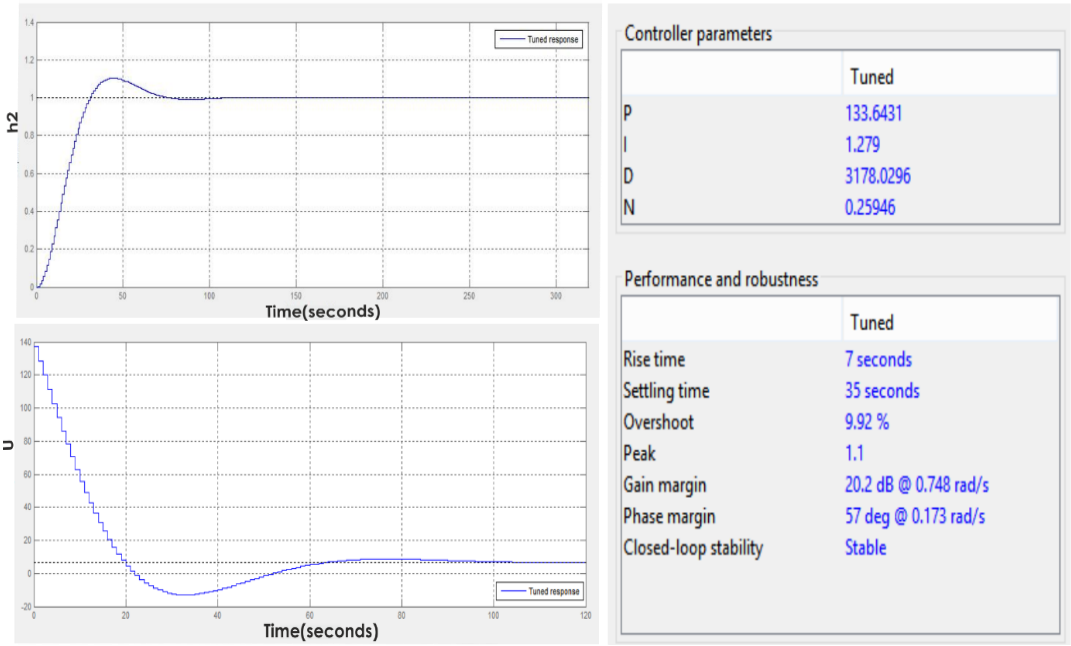



Figure 4.17: Minimum Phase (h2) Plot PID

Figure 4.17 shows the PID action, in minimum phase action for height of fluid (h2). The tuned parameters are as represented on right half of the figure. The result, when compared to Figure 4.6 and 4.13 of the DMC and QPDMPC respectively all of minimum phase performance, shows stable behaviours. Though the QPDMPC had a better performance as shown in Table 4.1 for minimum phase characteristics of the controllers.

Table 4.1 Comparing Controllers Actions under Minimum Phase

| **Parameters** | **PID** | **DMC** | **QPDMPC** |
|---|---|---|---|
| Rise Time(sec) | 7.0000 | 2.3581 | 0.0400 |
| Settling Time(sec) | 35.0000 | 1.3878 | 0.6100 |
| Settling Min | - | -0.0082 | 9 |
| Settling Max | - | 0.0052 | |
| Overshoot (%) | 9.92 | 0 | 0.5719 |
| Peak(sec) | 1.1 | 10 | 0.0052 |
| Stability | Stable | Stable | Stable |

The data sheet on the right half of the figures shows stable SISO systems obtained when the PIDs were applied and tuned.

The system yielded satisfactory results both for classical control PIDs and modern control policy DMPC. Table 4.2 represents the summary of the results obtained. These parameters depict the performance of controllers on the SISO systems.

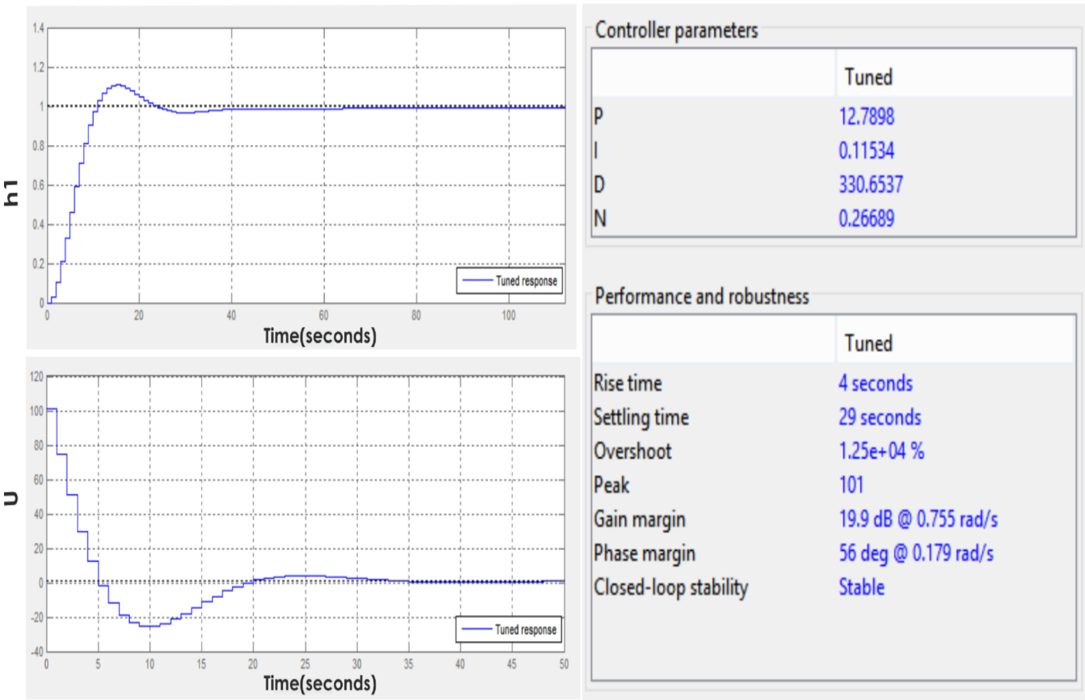



Figure 4.18: Nonminimum Phase (h1) Plot PID

The PID in nonminimum phase is tuned as represented on right half of Figure 4.18. The results compared to Figure 4.7 and 4.14 of the DMC and QPDMPC respectively all of minimum phase performance, shows stable behaviours. Though the QPDMPC had a better performance with faster rise times, settling time (better sensitivity and response) as shown in Table 4.2 for nonminimum phase characteristics of the controllers.

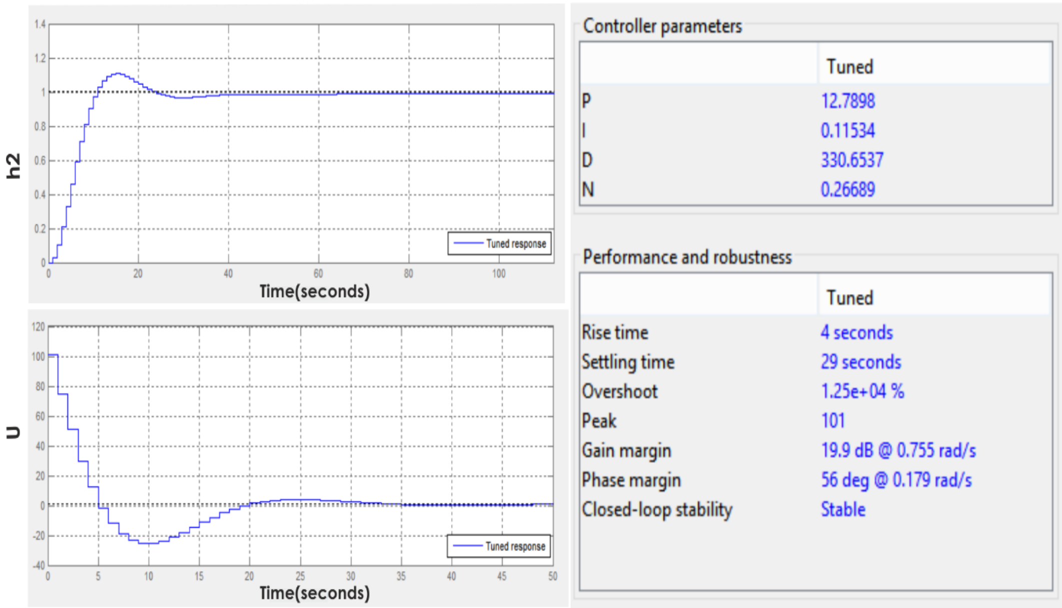



Figure 4.25: Nonminimum Phase (h2) PID

The PID in nonminimum phase is tuned as represented on right half of Figure 4.25. The result compared to Figure 4.5 and 4.11 of the DMC and QPDMPC respectively all of minimum phase performance, shows stable behaviours. But the QPDMPC had a better performance as shown in table 4.2 for nonminimum phase characteristics of the controllers.

Table 4.2: Comparing Controller Actions under Nonminimum Phase

| **Parameters** | **PID** | **DMC** | **QPDMPC** |
|---|---|---|---|
| Rise Time(sec) | 2.3581 | 0.8439 | 0.04825 |
| Settling Time(sec) | 29.0000 | 0.6576 | 0.1092 |
| Settling Min | - | 9 | -0.0058 |
| Settling Max | - | 10 | 0.0062 |
| Overshoot (%) | 1.25e+04 | 0 | 0 |
| Peak(sec) | 10 | 0.0062 | 101 |
| Stability | Stable | Stable | Stable |

The Tables 4.1 and 4.2 show the strength of the various controllers compared in this work in the control of the multiprocess system which was decentralized to meet control objectives. From both tables the QPDMPC demonstrated the fastest rise times and settling times. It also accommodated constraints (see Code is presented in Appendix B). Likewise, the DMC based controller had good settling time, though as seen in response of Figure 4.6 and 4.7, the penalty of ill-conditioning that led to plants shift from reference trajectory and possibly may lead to instability should the control horizon be increased further ($M > 1$) (Matlab script presented in Appendix A). The PID controller had performed satisfactorily though the overshoot and settling time showed slow response, but the SISO systems were stable.

# CHAPTER FIVE

# SUMMARY AND CONCLUSIONS

## 5.1 INTRODUCTION

In concluding this research work, constraints and challenges were encountered. Measures taken and the approaches used in addressing them are briefly summarized and conclusions drawn in this chapter.

## 5.2 SUMMARY

This research is about modelling of a multi-process system very close to reality as possible and controlling the system. A bench mark process, in this case a multivariable quadruple tank system was used for this research. The technique used in modelling is bond graph, which specifically, views a model from flow and effort energy relations (bonds) and interconnectivity. Furthermore, natural physical plants exhibit nonlinearities which was also observed, hence, the need for control and stability prompted the use of a controller algorithm, that had addressed the various constraints and challenges, like model-mismatch, looping, and uncertainties, and disturbances exhibited by nonlinear systems, the QPDMPC. The need for on-line incorporation prompted the use of a decoupling methodology that is on-line applicable. Moreover, validation of the controller action was carried out via comparison with a classical controller based policy.

## 5.3 CONCLUSION

In this work, we aimed at the modelling of a multi-process system (quadruple tank system), using bond graph technique and a predictive control of the system based on Decentralised Model Predictive Control for robust performance, efficient tracking, and stability. Hence, the various elements for energy interaction were identified and their bonds established. Subsequently, after validation of the quadruple tank modelled using the energy-bond graphical concept; bond graph, a state space model of the linear version was determined using Matlab/Simulink control toolbox. However, the MIMO system was decoupled and partitioned into SISO systems relative to the input/output pairing technique implemented. The control objective was to achieve a satisfactory performance, with a tractable solution amidst constraints inherent in most process control systems. This was realized, after using a QP-DMPC controller in place of DMC controller.

## 5.4 LIMITATIONS

A Physical experimental setup of the quadruple tank benched mark system was not yet in place. This would have aided a physical appreciation of the word outside the computer space.

## 5.5 RECOMMENDATIONS FOR FURTHER WORK

The following are recommended further work:

1. Application of bond graph modelling to higher complex multi-variable process systems.
2. Application of other variants of DMPC to quad-tank system.

# REFERENCES


Alipouri, Y., & Poshtan, J. (2013). Optimal controller design using discrete linear model for a four tank benchmark process. *ISA transactions, 52*(5), 644-651.

Alvarado, I., Limon, D., de la Pena, D. M., Maestre, J., Ridao, M., Scheu, H., Marquardt, W., Negenborn, R., De Schutter, B., & Valencia, F. (2011). A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *Journal of Process Control, 21*(5), 800-815.

Aravind, P., Valluvan, M., & Ranganathan, S. (2013). Modelling and Simulation of Non Linear Tank. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2*(2), 842-849.

Aridhi, E., Abbes, M., & Mami, A. (2014). A Data Acquisition with LabVIEW of Temperature of a water tank system: Comparative study with a Pseudo Bond graph Model. *Proceedings-Copyright IPCO*, 148-152.

Åström, K. J., Johansson, K. H., & Wang, Q.-G. (2002). Design of decoupled PI controllers for two-by-two systems. *IEE Proceedings-Control Theory and Applications, 149*(1), 74-81.

B. El fahime, B. R., N. Moujibi, A. Saka, A. Bouayad,M. Radouani. (2010). Modeling, Simulation and Analysis of Mechatronic Systems by Bond Graphs And Monte Carlo Simulations. *Journal of Modelling and Simulation of Systems*.

Benmoussa, S., Bouamama, B. O., & Merzouki, R. (2014). Bond graph approach for plant fault detection and isolation: Application to intelligent autonomous vehicle. *Automation Science and Engineering, IEEE Transactions on, 11*(2), 585-593.

Bhat, H. S., & Osting, B. (2011). Kirchhoff's laws as a finite volume method for the planar Maxwell equations. *Antennas and Propagation, IEEE Transactions on, 59*(10), 3772-3779.

Borutzky, W. (2013). Bond Graph Modelling and Simulation of Mechatronic Systems An Introduction into the Methodology. *European Control Conference (ECC).*

Borutzky, W., & Cellier, F. (1996). *Tearing in bond graphs with dependent storage elements.* Paper presented at the Proc. CESA.

Bos, A. M., & Tiernego, M. J. (1985). Formula manipulation in the bond graph modelling and simulation of large mechanical systems. *Journal of the Franklin Institute, 319*(1), 51-65.

Breedveld, P. C. (1985). Multibond graph elements in physical systems theory. *Journal of the Franklin Institute, 319*(1), 1-36.

Broenink, J. F. (1999). 20-sim software for hierarchical bond-graph/block-diagram models. *Simulation Practice and Theory, 7*(5), 481-492.

Cassolato, M., Love, J., & Tait, M. (2011). Modelling of a tuned liquid damper with inclined damping screens. *Structural Control and Health Monitoring, 18*(6), 674-681.

Christofides, P. D., Scattolini, R., de la Pena, D. M., & Liu, J. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering, 51*, 21-41.

Czaban, A., & Lis, M. (2012). Mathematical modelling of transient states in a drive system with a long elastic element. *Przegląd Elektrotechniczny, 88*(12b), 167-170.

Dajun Du, M. F. a. T. J. (2013). Modelling and stability analysis of MIMO networked control systems withmulti-channel random packet losses.

*Transactions of the Institute of Measurement and Control 35*. doi: 10.1177/0142331211406605

Das, S. (2009). *Mechatronic Modeling and Simulation Using Bond Graphs*: CRC Press.

Deepa, N., & Arulselvi, S. (2014). Design and Implementation of Neuro Controllers for a Two-Tank Interacting Level Process. *ME Thesis, Annamalai University*.

Emna Aridhi, M. A., Abdelkader Mami. (2014). A Data Acquisition with LabVIEW of

Temperature of a water tank system:

Comparative study with a Pseudo Bond graph

Model. *International Conference on Automation, Control, Engineering and Computer Science*.

Garelli, F., Mantz, R., & De Battista, H. (2006). Limiting interactions in decentralized control of MIMO systems. *Journal of Process Control, 16*(5), 473-483.

Garg, A., & Tangirala, A. K. (2014). *Interaction Assessment in Multivariable Control Systems through Causality Analysis.* Paper presented at the Advances in Control and Optimization of Dynamical Systems.

Gatzke, E. P., Meadows, E. S., Wang, C., & Doyle, F. J. (2000). Model based control of a four-tank system. *Computers & Chemical Engineering, 24*(2), 1503-1509.

Gupta, A., Bhartiya, S., & Nataraj, P. (2012). *Explicit-Model Predictive Control: A simulation based scalability study*. Paper presented at the IFAC Symposium on Advanced Control of Chemical Processes.

Halvarsson, B., Wik, T., & Carlsson, B. (2011). New Input-Output Pairing Strategies Based on Linear Quadratic Gaussian Control: Citeseer.

He, G., Li, J., & Cui, P. (2015). Decoupling Control Design for the Module Suspension Control System in Maglev Train. *Mathematical Problems in Engineering, 2015*.

Heier, S. (2014). *Grid Integration of Wind Energy: Onshore and Offshore Conversion Systems*: John Wiley & Sons.

Holt, B. R., & Morari, M. (1985). Design of resilient processing plants—VI. The effect of right-half-plane zeros on dynamic resilience. *Chemical Engineering Science, 40*(1), 59-74.

Husek, P. (2014). Decentralized PI controller design based on phase margin specifications. *Control Systems Technology, IEEE Transactions on, 22*(1), 346-351.

Jevtović, B. T., & Mataušek, M. R. (2010). PID controller design of TITO system based on ideal decoupler. *Journal of Process Control, 20*(7), 869-876.

Johansson, K. H. (2000). The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *Control Systems Technology, IEEE Transactions on, 8*(3), 456-465.

Kala, H., Deepakraj, D., Gopalakrishnan, P., Vengadesan, P., & Iyyar, M. K. (2014). Performance Evaluation of Fuzzy Logic and PID Controller for Liquid Level Process. *Performance Evaluation, 2*(3).

Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2012). *System dynamics: modeling, simulation, and control of mechatronic systems*: John Wiley & Sons.

Kayani, S. A., & Malik, M. A. (2008). *Bond-graphs+ genetic programming: Analysis of an automatically synthesized rotary mechanical system.* Paper presented at the Proceedings of the 10th annual conference companion on Genetic and evolutionary computation.

Kim, Y.-A., Przytycki, J. H., Wuchty, S., & Przytycka, T. M. (2011). Modeling information flow in biological networks. *Physical biology*, *8*(3), 035012.

Kirubakaran, V., Radhakrishnan, T., & Sivakumaran, N. (2014). Distributed multiparametric model predictive control design for a quadruple tank process. *Measurement, 47*, 841-854.

Kumar, A. (2013). Performance Comparison of Level Control with the Three, Five & Nine Fuzzy Rules based method. *International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), 2*(7), pp: 561-566.

Laughlin, D. L., RIVERA, D. E., & MORARI, M. (1987). Smith predictor design for robust performance. *International Journal of Control*, *46*(2), 477-504.

Li, D., Xue, Y., Wang, W., & Sun, L. (2014). *Decentralized PID Controllers Tuning Based on Desired Dynamic Equations.* Paper presented at the World Congress.

Liu, C., Pedersen, E., Æsøy, V., Hildre, H. P., & Zhang, H. (2013). *Dynamic Modelling Of The" Searazor"-An Interdisciplinary Marine Vehicle For Ship Hull Inspection And Maintenance.* Paper presented at the ECMS.

Luyben, W. L. (1970). Distillation decoupling. *AIChE Journal, 16*(2), 198-203.

Mach, E. (2014). *History and Root of the Principle of the Conservation of Energy*: Cambridge University Press.

Maciejowski, J. M. (2002). *Predictive control: with constraints*: Pearson education.

Maghade, D., & Patre, B. (2012). Decentralized PI/PID controllers based on gain and phase margin specifications for TITO processes. *ISA transactions, 51*(4), 550-558.

Mahapatro, S. R., Subudhi, B., & Ghosh, S. (2014). *An experimental evaluation of optimal control design for coupled tank system.* Paper presented at the Automation, Control, Energy and Systems (ACES), 2014 First International Conference on.

Marek Kubalčík, V. B. (2013). Adaptive Predictive Control of Three-Tank-

System. *INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES, 7*(2).

Mercangöz, M., & Doyle, F. J. (2007). Distributed model predictive control of an experimental four-tank system. *Journal of Process Control, 17*(3), 297-308.

Misawa, E., Hedrick, J., Slotine, J., & Verghese, G. (2014). *Sliding observer design for nonlinear state estimation.* Paper presented at the Nonlinear Control System Design. Selected PapersJiom the RAC Symposium.

Morilla, F., Vázquez, F., & Garrido, J. (2008). *Centralized PID control by decoupling for TITO processes.* Paper presented at the Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on.

Nacusse, M. A., & Junco, S. J. (2014). Switchable structured bond: A bond graph device for modeling power coupling/decoupling of physical systems. *Journal of Computational Science, 5*(3), 450-462.

Necoara, I., & Clipici, D. (2013). Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC. *Journal of Process Control, 23*(3), 243-253.

Ngwompo, R. (2011). Bond graph-based filtered inversion of multivariable physical systems. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 0959651811399508.

Nordfeldt, P., & Hägglund, T. (2006). Decoupler and PID controller design of TITO systems. *Journal of Process Control, 16*(9), 923-936.

Parisio, A., Rikos, E., & Glielmo, L. (2014). A model predictive control approach to microgrid operation optimization. *Control Systems Technology, IEEE Transactions on, 22*(5), 1813-1827.

Paynter, H. (1970). System graphing concepts. *INSTRUMENTS & CONTROL SYSTEMS, 43*(7), 77-&.

Paynter, H. (1989). The differential analyzer as an active mathematical instrument. *Control Systems Magazine, IEEE, 9*(7), 3-8.

Potts, A. S., Massaro, L. C., & Garcia, C. (2015). Detection of decoupled input/output pairs in multivariable systems. *ISA transactions, 55*, 195-207.

Prof. D.Angeline Vijula, A. K., Honey Mol P, Poorna Priya S. (2013). Mathematical Modelling of Quadruple Tank System. *International Journal of Emerging Technology and Advanced Engineering, 3*(12).

Raimondo, D., Magni, L., & Scattolini, R. (2007). Decentralized MPC of nonlinear systems: An input-to-state stability approach. *International Journal of Robust and Nonlinear Control, 17*(17), 1651-1667.

Rodríguez, C., Guzmán, J. L., Berenguel, M., & Hägglund, T. (2014). Optimal feedforward compensators for systems with right-half plane zeros. *Journal of Process Control, 24*(4), 368-374.

Romero, S., Vanessa, H., Tore, A., & Karl, J. (2014). *Design of measurement noise filters for PID control.* Paper presented at the World Congress.

Rosinova, D., & KozĂˇkovĂˇ, A. (2012). *Decentralized Robust Control of MIMO Systems: Quadruple Tank Case Study.* Paper presented at the Advances in Control Education.

Russer, J. A., & Russer, P. (2012). *Lagrangian and Hamiltonian formulations for classical and quantum circuits,".* Paper presented at the Proc. 7th Vienna Conference on Mathematical Modelling, MATHMOD, Vienna.

Russo, L., & Bequette, B. (2014). *CSTR PERFORMANCE LIMITATIONS DUE TO COOLING.* Paper presented at the Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes (DYCORD+'92): Selected Papers from the 3rd IFAC Symposium, Maryland, USA, 26-29 April 1992.

Saeed, Q., Uddin, V., & Katebi, R. (2013). MIMO PREDICTIVE PID CONTROL: A PRACTICAL APPROACH FOR QUADRUPLE TANK. *Journal of Circuits, Systems, and Computers, 22*(03), 1350009.

Samantaray, A. K., & Bouamama, B. O. (2008). *Model-based process supervision: a bond graph approach*: Springer Science & Business Media.

Sano, M. B., Caldwell, J. L., & Davalos, R. V. (2011). Modeling and development of a low frequency contactless dielectrophoresis (cDEP) platform to sort cancer cells from dilute whole blood samples. *Biosensors and Bioelectronics, 30*(1), 13-20.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control–a review. *Journal of Process Control, 19*(5), 723-731.

Seetharaman, K., Namazi, H., & Kulsih, V. V. (2012). Phase lagging model of brain response to external stimuli—modeling of single action potential. *Computers in biology and medicine, 42*(8), 857-862.

Seo, K., Goodman, E. D., & Rosenberg, R. C. (2005). *Design of air pump system using bond graph and genetic programming method.* Paper presented at the Proceedings of the 7th annual conference on Genetic and evolutionary computation.

Srinivasarao, P., & Subbaiah, P. (2014). Tuning and Analysis of Multiple Interactive Loop Process by Model Predictive Control. *parameters, 1*(3), 5-7.

Tavakoli, S., Griffin, I., & Fleming, P. J. (2006). Tuning of decentralised PI (PID) controllers for TITO processes. *Control engineering practice, 14*(9), 1069-1080.

Teixeira, A., Pérez, D., Sandberg, H., & Johansson, K. H. (2012). *Attack models and scenarios for networked control systems.* Paper presented at the

Proceedings of the 1st international conference on High Confidence Networked Systems.

Tian, X. K., Tang, S., & Zhu, Q. J. (2013). *Flight Dynamics Modeling and Analysis of Flexible Hypersonic Flight Vehicles.* Paper presented at the Applied Mechanics and Materials.

Triki, S., Mekki, T., & Kamoun, A. (2014). Modeling Switched Behavior with Hybrid Bond Graph: Application to a Tank System. *arXiv preprint arXiv:1402.2925*.

Tsai, J. J.-H., & Gero, J. S. (2010). A qualitative energy-based unified representation for buildings. *Automation in Construction, 19*(1), 20-42.

Vadigepalli, R., Gatzke, E. P., & Doyle, F. J. (2001). Robust control of a multivariable experimental four-tank system. *Industrial & engineering chemistry research, 40*(8), 1916-1927.

Vijula, D. A., & Devarajan, N. (2013). Design of Decentralised PI Controller using Model Reference Adaptive Control for Quadruple Tank Process. *International Journal of Engineering and Technology*, *5*(6), 1347-1356.

Wang, L. (2009). *Model predictive control system design and implementation using MATLAB*®: Springer Science & Business Media.

Wellstead, P. E. (2000). *Introduction to Physical System*

*Modelling*

Wolfgang, B. (2010). Bond graph methodology; development and analysis of multidisciplinary dynamic system models: Springer: London. Isbn.

Yusuf, A., Muhammad, A., Ahmad, K., & Jibrin, M. M. (2014). SISO Control of TITO Systems: A Comparative Study *Transactions on Engineering Technologies* (pp. 311-326): Springer.

Zheng, T. (2010). *Model Predictive Control* T. ZHENG (Ed.)

# APPENDICES

## APPENDIX A

### Dynamic Matrix Control Matlab Script



```
A =[-0.0158730158730159,0,0,0,0,0,0,0;0,-0.0415140415140415,-
0.000407000407000407,0,0,0,-0.0166666666666667;0,1,0,0,0,0,0,0;0,0,0,-
```

```
0.0288461538461538,-0.000196232339089482,0,0;0,0,0,1,0,0,0;0,0,0,0,0,-
0.0109890109890110,-0.0166666666666667;0,0,0,0,0,0,-0.0333333333333333]
B = [1;0;0;1;0;0;1] C =
[0.0238095238095238,0,0.00101750101750102,0,0,0,0;0,0,0,0,0.000490580847
723705,0.0175824175824176,0] D = [0]
```

```
 A = [-0.0161290322580645,0,0,0,0,0,0;0,-0.0596072931276297,-
0.000701262272089762,0,0,0,-0.0166666666666667;0,1,0,0,0,0,0;0,0,0,-
0.0444444444444444,-0.000370370370370370,0,0;0,0,0,1,0,0,0;0,0,0,0,0,-
0.0111111111111111,-0.0166666666666667;0,0,0,0,0,0,-0.0333333333333333]
 B = [1;0;0;1;0;0;1]
 C =
[0.0419354838709677,0,0.00105189340813464,0,0,0,0;0,0,0,0,0.000518518518
518518,0.0277777777777778,0]
 D = [0]
% Ad = [0.319148936170213,0,0,0,0,0,0;0,0.296038151137197,-
0.000366837857666911,0,0,0,-
0.00303815938074472;0,0.901320616287601,0.333455612619222,0,0,0,0.00101
271979358157;0,0,0,0.307502641775273,-
0.000176118351532230,0,0;0,0,0,0.897499119408242,0.333392039450511,0,0;0
,0,0,0,0,0.323529411764706,-
0.00300727032385988;0,0,0,0,0,0,0.303370786516854];
% Bd = [0.947823983292606;0.00107415155022500;-
0.000358050516741686;0.951941570163785;-
0.317313856721262;0.00106323061943100;0.953402401599839];
% Cd = [0.0225672376974430,-
0.000324241920940273,0.000959265170254312,0,0,0,-3.64316765101430e-
07;0,0,0,-
0.000155668100824795,0.000462513876489054,0.0166378066161541,1.869416
47372518e-05];
% Dd = [-0.00797859459882777;4.84275995266572e-05];
%%Initialisation of parameters without noise nor disturbance
P = 3; %prediction horizon
M = 1; %control horizon[1]
```

```
N = 60; %model lenght
w = 0.0;%weight[0]
ysp = 3;%output set point change[1]
timesp = 2; %time to setppoint change[1]
delt = 0.1; %sampling time interval
tfinal = 10; %final simulation time
noise = 0;
```

```
%define time
tvec = 0:delt:tfinal;
ksp = fix(timesp/delt);
kfinal = length(tvec);
```

Define set point vector

```
r = [zeros(1,ksp),ones(1,(kfinal-ksp))*ysp];
%%define Plant as SISO LTI OBJECT
% c = input('enter plant in 1.statespace 2.transferfunction 3.polezero
4.frequencyresponse');
% if c==1
%     a = input('enter A matrix');
%     b = input('enter B matrix');
%     c = input('enter C matrix');
%     d = input('enter D matrix');
%     plant = ss(a,b,c,d);
% elseif c==2
%     nump = input('enter numerator coefficients');
%     denp = input('enter denominator coefficients');
%     plant = tf(nump,denp);
% elseif c==3
%     zero =input('enter zeroes');
%     poles = input('enter poles');
%     K = input('enter gain');
%     plant = zpk(zero,pole,K);
% elseif c==4
```

```
%     resp = input('response');
%     freq = input('frequencies');
%     plant = frd(resp,freq,'Units','Hz');
% end
plant=ss(A,B,C,D);
plant = tf(plant);
%%discretize the plant
plant = c2d(plant,delt);
%%define model here
%assumption plant = model
model = plant;
%[numm,denm,tm] = tfdata(plant);
numm = get(model,'num');
numm = numm{:};%get numerator of polynomial
denm = get(model,'den');
denm = denm{:}; % get denominator of polynomial
%%define step response coefficient matrix
s = step(model,0:delt:N*delt);
%define free response i.e Sp matrix for past control moves
for i = 1:P
    for j = 1:N-2
        if (i+j<=N-1)
            Sp(i,j) = s(i+j);
        else
            Sp(1,j)=0;
        end
    end
end
%define forced response i.e Sf matrix for future and control moves
for i = 1:P
    for j =1:M
        if (i+j>0)
            Sf(i,j)=s(i+1-j);
```

```
        else
            Sf(i,j)=0;
        end
    end
end
Sf
%obtain W matrix
W = w*eye(M,M);
%obtain Kmat where Kmat = ((Sf*Sf + W)^-1*Sf'
Kmat = inv(Sf'*Sf+W)*Sf';
```

noise and disturbance

```
sd = 0.05; %Standard deviation of measurement noise[0.1]
randn('state',0); %Reset state of random number generator. Change to
                  %to get differnt random sequancies generated
noise = sd*randn(kfinal,1); %Measurement noise, normal distribution,mean =0
                            %sd = standard deviation
udist = 0.1*ones(kfinal,1);% Input disturbance(default constant 0.1)
ydist = 0.1*[ones(floor(kfinal/2),1);-ones(ceil(kfinal/2),1)];
%output disturbance(default constant +/-0.1,changing sign halfway
```

plant initial condition

```
ndenm = length(denm)-1;
nnumm = length(numm)-1;
umpast = zeros(1,nnumm);
ympast = zeros(1,ndenm);
%uu = zeros(1,kfinal);
%yy = zeros(1,kfinal);
%xinit = zeros(1,size(
%nump=[zeros(1,ndenp-nnump-1),nump]; Pad numerator withleading zeros
%numm=[zeros(1,ndenm-nnumm-1),numm];
uinit = 0;
yinit = 0;
```

```
%initialize input vector
u = ones(1,min(P,kfinal))*uinit;
u;
dist(1)=0;
y(1)=yinit;
%x(:,1)=xinit;
dup=zeros(1,N-2);
for k = 1:kfinal
    [m,p]=size(Kmat);
    for i = 1:p
        if k-N+i>0
            uold(i)=u(k-N+i);
        else
            uold(i)=0;
        end
    end

dvec = ydist(k)*ones(1,p);
rvec = r(k)*ones(p,1);
dnoise = noise(k)*ones(1,p);
y_freed = Sp*dup'+s(N)*uold'+dvec';
y_free = y_freed+dnoise';
E = rvec-y_free;
delup(k) = Kmat(1,:)*E;
if k>1
    u(k)=u(k-1)+delup(k);
else
    u(k) = delup(k)+uinit;
end
%%plant equations
umpast = [u(k),umpast(1,1:length(umpast)-1)];
y(k+1)= -denm(2:ndenm+1)*ympast'+numm(2:nnumm+1)*umpast';
ympast= [y(k+1),ympast(1:length(ympast)-1)];
%%model prediction
```

```
if k-N+1>0
    ymod(k+1)=Sf(1,1)*delup(k)+Sp(1,:)*dup'+s(N)*u(k-N+1);
else
    ymod(k+1)=Sf(1,1)*delup(k)+Sp(1,:)*dup';
end
%%disturbance compensation
dist(k+1)= y(k+1)-ymod(k+1);
%additive disturbance compensation
%put input change into vector of the past control moves
dup = [delup(k),dup(1,1:N-3)];
end
%%stairs plotting for input(zeros order hold)mand setpoint
[tt,uu] = stairs(tvec,u);
[ttr,rr] = stairs(tvec,r);
```

```
figure(1)
subplot(2,1,1)
plot(tvec,y(1:length(tvec)),'-',tvec,y(1:length(tvec))+noise',':',ttr,rr,'--
','linewidth',2);
%mesh(y(1:length(tvec)),y(1:length(tvec))+noise',ttr);
grid;
title('SISO(1)')
%plot(ttr,rr,'o',tvec,y(1:length(tvec)))
ylabel('h1');
xlabel('Time');
title('Plant Output');
legend('h1','Measured Output','Set-point');
subplot(2,1,2)
plot(tt,uu,'-','linewidth',2); hold on;
plot(tvec,u+udist',':','linewidth',2)
grid on;
title('Controller output,Plant input(dotted)')
xlabel('Time')
stepinfo(tt,uu)
```

```
ans =

        RiseTime: 2.3581
    SettlingTime: 1.3878
     SettlingMin: 9
     SettlingMax: 10
       Overshoot: 0
      Undershoot: 0
            Peak: 10
        PeakTime: 2.2653
```

# APPENDIX B
## Quadratic Programming Matlab Code



```
function y = MyTank_MPC_1(M,rr,A,B,C,D,N)
%----------------------------------------------
global T Ts sim_length
% Construct model
```

```
%-----------------------------------------------

%Specify system here in state space form with A B C D
% A = [-0.0161290322580645,0,0,0,0,0,0;0,-0.0596072931276297,-
0.000701262272089762,0,0,0,-0.0166666666666667;0,1,0,0,0,0,0;0,0,0,-
0.0444444444444444,-0.000370370370370370,0,0;0,0,0,1,0,0,0;0,0,0,0,0,-
0.0111111111111111,-0.0166666666666667;0,0,0,0,0,0,-0.0333333333333333]
% B = [1;0;0;1;0;0;1]
% C =
[0.0419354838709677,0,0.00105189340813464,0,0,0,0;0,0,0,0,0.000518518518
518518,0.0277777777777778,0]
% D = [0]
A = [0.319148936170213,0,0,0,0,0,0;0,0.296038151137197,-
0.000366837857666911,0,0,0,-
0.00303815938074472;0,0.901320616287601,0.333455612619222,0,0,0,0.00101
271979358157;0,0,0,0.307502641775273,-
0.000176118351532230,0,0;0,0,0,0.897499119408242,0.333392039450511,0,0;0
,0,0,0,0,0.323529411764706,-
0.00300727032385988;0,0,0,0,0,0,0.303370786516854];
B = [0.947823983292606;0.00107415155022500;-
0.000358050516741686;0.951941570163785;-
0.317313856721262;0.00106323061943100;0.953402401599839];
C = [0.0225672376974430,-
0.000324241920940273,0.000959265170254312,0,0,0,-3.64316765101430e-
07;0,0,0,-
0.000155668100824795,0.000462513876489054,0.0166378066161541,1.869416
47372518e-05];
D = [-0.00797859459882777;4.84275995266572e-05];
%%Initialisation
```

MPC PARAMETERS

```
Ts = 1; %Sample interval
%sim_length = 1000; %Simulation length in sec
sim_length = 20; %Simulation length in sec
T = sim_length/Ts; %number of samples in simulation
% t = 0:Ts:sim_length-Ts; %time steps

 M = 3; %Control Horizon[5]
 N = 7; %Prediction Horizon[100]
 rr = 0.1;%weight[0-1]
 %qq  = 1.0;
 %Ts = 0.1; %Sample interval
 %Ts = 1.0; %Sample interval

% A=[0.9 0;0.9 0.9]; B=[0.1;0.2]; C=[1 1]; D=0;

%Specify system dimensions

n_in = size(B,2); %# inputs

n_out = size(C,1); %# outputs

n_states = size(A,1); %# states

%-----------------------------------------------

% Specify horizons

%-----------------------------------------------

% M = M; %Control Horizon
%
% N = 8; %Prediction Horizon
```

```
%-----------------------------------------------------

% Initial Matrix Filling

%-----------------------------------------------------

Lambda = zeros((N+1)*n_out,n_states);

Phi = zeros((N+1)*n_out);

AB = zeros((N+1)*n_out,n_in);

AB(1:n_out,:) = D;

for i = 2:N+1,

AB((i-1)*n_out+1:(i)*n_out,:) = C*A^(i-2)*B

end;

for i = 1:M+1,

Phi(1+(i-1)*n_out:end,1+(i-1)*n_in:(i)*n_in) = AB(1:(N-i+2)*n_out,:)

end;

Phi = sparse(Phi);
%Lambda(1:n_out,:) = C;
Lambda(1:n_out,:) = C;

for i = 2:N+1,
```

```
Lambda(1+(i-1)*n_out:i*n_out,:) = C*A^(i-1)

end;

%-----------------------------------------------

% Quadratic cost weighting Q (error) and R (control) generation.

%-----------------------------------------------

r = diag([rr*ones(1,n_in)]); %Weights on input moves (i.e. ||u(t)-u(t-1)||)

q = diag([ones(1,n_out)]); %Weights on output deviation from setpoint

Q = sparse(kron(eye(N+1),q));

R = kron(diag([2*ones(1,2*N + 1),1]),r);

R = R + kron(diag([2*ones(1,2*N + 1),-1]),r);

R = R + kron(diag([2*ones(1,2*N + 1),1]),r);

%-----------------------------------------------

% Generate the Hessian H

%-----------------------------------------------

H = full(Phi'*Q*Phi) + R;
%H = full(Phi'*Q*Phi)
%-----------------------------------------------

% Create an observer
```

```
%------------------------------------------------

Aa = blkdiag(A,eye(n_out)) ;

Ba = [B;zeros(n_out,n_in)];

Ca = [C eye(n_out)];

Da = D;

Ga = diag([ones(1,n_states),ones(1,n_out)]);

Qa = 1*eye(n_states+n_out);

Ra = 1*eye(n_out);

L = dlqe(Aa,Ga,Ca,Qa,Ra);

L = Aa*L;

%------------------------------------------------

% Create plant constraints

%------------------------------------------------

%lb = -0.01*ones(n_in,1); ub = 0.1*ones(n_in,1);
lb = -0.1*ones(n_in,1); ub = 0.1*ones(n_in,1);
Lb = kron(ones((M+1),1),lb); Ub = kron(ones((M+1),1),ub);

%------------------------------------------------
```

```
% Set initial conditions

%------------------------------------------------

% sim_length = 1000; %Simulation length in sec
% % sim_length = SimulationTime; %Simulation length in sec
% T = sim_length/ts; %number of samples in simulation
%
t = 0:Ts:sim_length-Ts %time steps

%-----------------------------------------------

% Set system initial conditions......

%-----------------------------------------------

x = zeros(n_states,T);

x(:,1) = -2*ones(n_states,1); %Plant state

xh = zeros(n_states+n_out,T); %Estimated state

u = zeros(n_in,T); %Input record

u_old = zeros(n_in,1);

y = zeros(n_out,T); %Output record

Aobs = Aa-L*Ca; %Generate observer state gain matrix

Bobs = Ba-L*Da; %Generate observer input gain matrix

fd = full([Phi'*Q*Lambda, Phi'*kron(ones(N+1,1),q)]);
```

```
opt=optimset('display','off');
%opt=optimset('display','on');
%-----------------------------------------------

% Generate reference signal

%-----------------------------------------------

ref = ones(1,T);
%ref(201:600) = -ref(201:600);
ref(4:10) = -ref(4:10);
%-----------------------------------------------

% Generate noise signal

%-----------------------------------------------

noise = 1e-2*cumsum(randn(T,n_out)')' + 1e-2*randn(T,n_out);

[bfilt,afilt] = butter(4,0.9);

noise = filter(bfilt,afilt,noise)';

%-----------------------------------------------

% MPC Evolution

%-----------------------------------------------

for i = 1:T,

%SIMULATE REAL PLANT-------------------------------
```

```
x(:,i+1) = A*x(:,i) + B*u(:,i);

y(:,i) = C*x(:,i) + D*u(:,i) + noise(:,i);

%Add disturbance
% if i<=50, y(:,i)= y(:,i)+0.3*ones(n_out,1);
% else i>300, y(:,i)= y(:,i)+0.5*ones(n_out,1); end
%if i>300, y(:,i)=y(:,i)+0.5*ones(n_out,1); end
if i<=1, y(:,i)= y(:,i)+0.05*ones(n_out,1);
elseif i>8, y(:,i)= y(:,i)+0.05*ones(n_out,1); end
%CONTROLLER STARTS HERE----------------------------

%Observe state

xh(:,i+1) = Aobs*xh(:,i) + Bobs*u(:,i) + L*y(:,i);

%Solve dynamic optimisation problem

f = fd*[xh(1:n_states,i+1);xh(n_states+1:end,i+1)-ref(:,i)];

f(1:n_in) = f(1:n_in)-r*u_old;

U = quadprog(H,f,[],[],[],[],Lb,Ub,[],opt);
%U = quadprog(H,f,[],[],[],[],Lb,Ub,[],[]);
u(:,i+1) = U(1:n_in);
%u_old = u(:,i);
u_old = u(:,i+1);

end;

%-----------------------------------------------
```

```
% Plots

%-----------------------------------------------
figure(1);
subplot(2,1,1);

plot(y(1,:),'r--','Linewidth',2); hold on
%
plot(y(2,:),'b-','Linewidth',2); hold off;
%plot(y,ts,'r--','Lindwidth',2); hold on
%plot(y2,ts,'b-','Linewidth',2); hold off
title('SISO Tank(2)DMPC Response Minimum-Phase')
ylabel('Output and reference');
legend('Fluid Height(h2)','reference')
xlabel('Time (sample number)');
grid on;
subplot(2,1,2);

plot([1 T],[ub ub],'r--','Linewidth',2); hold on;
%plot([1 T],[Lb Ub],'r--','Linewidth',2)
plot([1 T],[lb lb],'r--','Linewidth',2)
%plot([1 T],[lb ub],'r--','Linewidth',2);

plot(u(:,2:T)','b-','Linewidth',2); hold off;

xlabel('Time (sample number)');

ylabel('Input and constraints');

axis([1 T 2*lb 2*ub]);
grid on
figure(2)
mesh(y,'linewidth',2)
```

```
stepinfo(y(1,:))
stepinfo(y(2,:))
```

```
        RiseTime: 0.6851
    SettlingTime: 18.9366
     SettlingMin: 0.0263
     SettlingMax: 0.1009
       Overshoot: 43.7951
      Undershoot: 26.8279
            Peak: 0.1009
        PeakTime: 10


ans = 

        RiseTime: 4.6318
    SettlingTime: 19.8783
     SettlingMin: 0.0324
     SettlingMax: 0.0981
       Overshoot: 59.9243
      Undershoot: 18.3687
            Peak: 0.0981
        PeakTime: 10
```

*Published with MATLAB® R2015a*