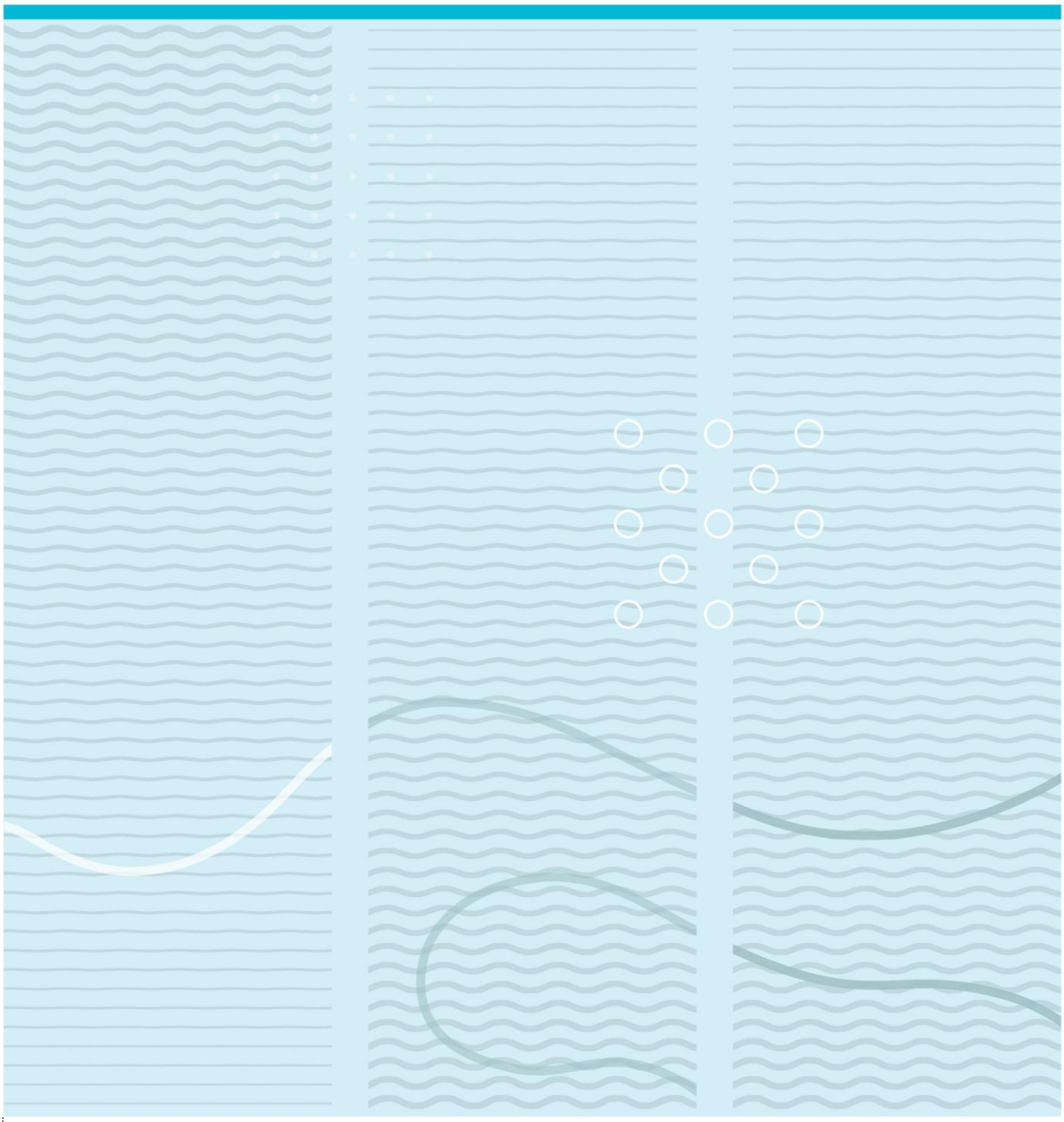
University College of Southeast Norway
Faculty of Technology

Master's Thesis
Study programme: Systems and Control Engineering(SCE)
Spring/Autumn 2016

Teghese IKOKO

Model Predictive Control using DeltaV of the Quadruple Tank Process.



University College of Southeast Norway

MASTER'S THESIS, COURSE CODE FMH606

Student:	Teghese Ikoko
Thesis title:	Model Predictive Control using DeltaV of the Quadruple Tank Process.
Signature:	. . . . .
Number of pages:	70
Keywords:	Model Predictive Control, DeltaV, System Identification, DSR, Quadruple Tank Process, MATLAB
Supervisor:	David Di Ruscio Sign: . . . . .
2nd supervisor:	Rune Andersen Sign : . . . . .
Censor:	Sign : . . . . .
External partner:	Emerson Process Management
Availability:	
Archive approval	Date : . . . . .

Abstract:

Model predictive control is widely used in process industries for multivariable systems. The Quadruple Tank Process (QTP), a 2-input, 2-output system is presented. The strong interactions between the input and output variables make this process very challenging to control. Using first principle, a non-linear model of the QTP was developed.

The major objective in this work is to investigate the use of the MPC strategy in DeltaV to control the Quadruple tank. A model of the process must be present before MPC can be implemented. DeltaV MPC creates a step response model of the process using input and output data of the process. The modelling technique in DeltaV (step response modelling) didn't give a correct representation the Quadruple tank process.

To compare this modelling technique, the subspace system identification method (DSR) was used. The DSR method produces state space model matrices from known input and output data. Validation results for the DSR showed the modelling results were more accurate when compared to the modelling technique in DeltaV. The mean square error was used as a basis of comparing these two methods. All simulations were done using input and output data logged from the real plant.

The default setting of the generated controller in DeltaV MPC didn't give good tracking because of the poor models of the process. Tuning of this controller was done to improve performance.

To achieve better control results of the QTP, the state space model identified by the DSR method was also used in the implementation of MPC using MATLAB. This resulted in a more robust controller.

DeltaV MPC was used to control a SISO system (2-tank process), and much better results were achieved when compared to the MIMO system (QTP)

The DSR system identification method was compared with the model prediction strategy in DeltaV MPC for both the MIMO and SISO systems, and the DSR showed better results especially in the MIMO systems. But overall, the DSR gave better results for both the MIMO and the SISO systems-

Table of Contents

1 Introduction 1

1.1 Background 1

1.2 Overview of the QTP 2

1.3 Objective of the thesis 2

1.4 Thesis Outline 2

2 The Quadruple Tank 4

2.1 The Quadruple Tank 4

2.2 Development of Non Linear Model 6

3 Model Predictive Control 10

3.1 Introduction to MPC 10

3.2 Defining Prediction Models 13

3.2.1 Prediction Models from State Space Models 13

3.2.2 Prediction Models from FIR and step response models 15

3.3 MPC Strategy in DeltaV 15

3.3.1 Generation of MPC Controller in DeltaV 17

3.3.2 The Penalty of moves (POM) and the Penalty of error (POE) 18

3.3.3 Output Constraints Handling 19

4 Development of Process Model/System Identification of Process Model 20

4.1 Model identification in DeltaV 20

4.1.1 DeltaV Predict Algorithm 20

4.2 System identification based model 22

4.3 DSR method of system Identification 22

4.3.1 DSR Algorithm 22

4.4 Model Performance Indices 23

4.4.1 MSE 23

4.4.2 IAE 23

5 Implementation of MPC 25

5.1 Collection of Data 25

5.2 Model Prediction in DeltaV Predict 27

5.3 Generation of Controller 30

5.4 System Identification Using DSR 30

5.4.1 Removing trends in Data 30
5.4.2 System Identification from Real data 31
5.5 MPC with Integral action 31
6 Results (Modelling, Verification and Control) 33
6.1 Implementation in DeltaV 33
6.2 DSR 37
6.2.1 Model Performance Indices 38
6.3 MPC from DSR Identified Model 38
6.4 SISO system 42
6.4.1 Model Creation in DeltaV of the 2-tank 43
6.4.2 Controller Generation for the 2-tank process 45
6.4.3 System Identification in DSR 46
6.4.4 Model Comparison 48
7 Discussion of Results 49
8 Conclusion and Recommendation 51
8.1 Conclusion 51
8.2 Future Work 51
References 53
Annexes 55

List of Figures

Figure 2-1: Schematic diagram of the quadruple tank 5
Figure 3-1: Conceptual picture of MPC-1 [18] 12
Figure 3-2: Conceptual picture of MPC-2 [18] 13
Figure 3-3: MPC blocks in DeltaV 16
Figure 5-1: Input signals 26
Figure 5-2: Output signals 26
Figure 5-3: Flow chart for implementation in DeltaV MPC using external historical data 28
Figure 5-4: Input and Output of process in DeltaV 29
Figure 6-1: Step Response models 33
Figure 6-2: Actual and predicted model for control output 1 (Level in Tank1) 35
Figure 6-3: Actual and predicted model for control output 2 (Level in Tank2). 35
Figure 6-4: Simulation for the MPC control for level in Tank 2 37
Figure 6-5: Simulation for the MPC control for level in Tank 1 37
Figure 6-6:DSR identification results for real and predicted levels in Tanks 1 and 2. 38
Figure 6-7: *MPC Output and Reference for Level in Tank 1. With matrices* $Q = 10000100$ *and* $R = 0.1000.1$*and prediction horizon of 15secs and a sampling time of 0.1secs. The input constraints are* $Umin = 0V$ *and* $Umax = 5V$ 39
Figure 6-8: MPC Output and Reference for Level in Tank 2. With matrices $Q = 10000100$ *and* $R = 0.1000.1$*and prediction horizon of 15secs and a sampling time of 0.1secs. The input constraints are* $Umin = 0V$ *and* $Umax = 5V$ 40
Figure 6-9: MPC Output and Reference for Level in Tank 1. With matrices $Q = 100000010000$ *and* $R = 0.1000.1$*and prediction horizon of 15secs and a sampling time of 0.1secs. The input constraints are* $Umin = 0V$ *and* $Umax = 5V$ 41
Figure 6-10: *MPC Output and Reference for Level in Tank 2. With matrices* $Q = 10000100$ *and* $R = 0.1000.1$*and prediction horizon of 15secs and a sampling time of 0.1secs. The input constraints are* $Umin = 0V$ *and* $Umax = 5V$ 42
Figure 6-11: 2-tank Process 43
Figure 6-12: 2-tank Control using MPC in DeltaV [20] 44
Figure 6-13: Model verification using DeltaV Predict of the 2-tank system showing actual and predicted models 45

Figure 6-14: MPC controller for the level in the 2tank process. 46
Figure 6-15: Input and output data for the 2-tank 47
Figure 6-16: Model development using DSR method of the 2-tank system 48

List of tables

Table 2-1: Parts of the QTP and their functions..... 5
Table 2-2: Parameter definitions for the Non Linear model of the QTP 7
Table 2-3: QTP parameter values 8
Table 6-1: Step Response Design..... 34
Table 6-2: MSE from DSR method 38
Table 6-4: Comparison of model performance indices for the 2-tank process 48

Nomenclature

HSN	University College of Southeast Norway
MPC	Model Predictive Control
QTP	Quadruple Tank Process
SISO	Single Input single Output
MIMO	Multiple Input Multiple output
FIR	Finite Impulse Response
ARX	Auto-Regressive with eXternal inputs
MSE	Mean Square Error
IAE	Integrated Absolute Error
DSR	Deterministic and Stochastic system and Realization
TSS	Time to steady state
POM	Penalty of Move
POE	Penalty of Error

Foreword

The thesis work is a prerequisite for the completion of the masters' programme in Systems and Control engineering in the University College of Southeast Norway (HSN).This project work was done in accordance with the master's thesis topic *"Model Predictive control using DeltaV of the Quadruple Tank Process"*. The entire work was carried out in HSN, Porsgrunn.

DeltaV Books online is an online documentation about DeltaV. It is necessary to mention that most of the contents on DeltaV MPC as presented in this work are largely based on information from DeltaV Books Online. Otherwise, all other materials and sources used in this work have been referenced accordingly.

I would like to specially thank my supervisor David Luigi Di Ruscio for his help and guidance all through the course of this work. Thank you David.
My gratitude goes to Rune Andersen at Emerson Process Management for his assistance and helpful suggestions and for all the tutorials on DeltaV MPC he put together. I would also like to thank Fredrik Hansen and Elvind Fjelddalen for their help in refurbishing the experimental Quadruple Tank Process.
Finally, I am very grateful to my son and mum, for having each other's back these past two years I was away, I love you and I can't wait to come home.

Porsgrunn, June 2016

IKOKO, Teghese

1 Introduction

This introductory chapter gives an overview of this thesis work. The Quadruple tank process is presented; the technique of Model Predictive control and the DeltaV control system are introduced. Previous work on the Quadruple tank, the task of this thesis work, and an outline of the different chapters are also shown.

1.1 Background

Different industries ranging from oil and gas, petrochemical industries, manufacturing, refineries, pharmaceuticals, telecommunication, food industries requires one or more process variable(s) to be controlled. Process control industries have evolved over the years in terms of complexities in their plants and how these plants are controlled. Most industrial processes require control of at least two variables. Each of the input signals (variable that is manipulated) may affect one or more output signals (variable that is controlled). Many real time applications of MIMO (multiple input, multiple output) systems have interactions where an input affects more than one controlled variable. This complex interaction between the input and output signals makes it more difficult to control a multivariable system when compared to SISO (single input, single output) systems. The way the pairing of the input/output pairs is done is very crucial in managing these interactions and achieving the desired control objective[1]. Cross coupling poses a major challenge in MIMO systems.

In 1996, the Quadruple Tank Process(QTP), a multivariable process with two inputs and two outputs was developed by Johansson et al [2] to teach and practice the diverse dynamics of multivariable control theory. The quadruple tank was built here in HSN for the same purpose. In the past, different controllers have been designed to control both the linear and non-linear models of the QTP. A multivariable predictive PI(D) controller was implemented to control the QTP [3]. The Model Predictive Controllers strategy with integral action was used to control the QTP [4], [5], [6]. The IMC (Internal Model Control) and DMC(Dynamic Matrix Control) strategies was employed by Gatzke et al to control the non-linear model of the QTP [7]. A non-linear MPC was implemented for the control of the QTP [8] .

1.2 Overview of the QTP

The Quadratic Tank Process consists of 4 differently inter-connected tanks with two pumps and two level sensors. The inputs to the system are the voltages to the two pumps. Pump 1 extracts water from the reservoir to tanks 1 and 4 while Pump 2 extracts water from the reservoir and discharge to Tanks 2 and 3. The outputs are the voltages from the level sensors. The objective is to control the level of Tanks 1 and 2. More details of the QTP will be discussed in later chapters.

1.3 Objective of the thesis

The objective of this thesis work is to investigate the use the Model Predictive Control strategy in DeltaV[1] to control the level of the two lower tanks in a QTP.

Building a control structure entails much more than just designing a controller. According to Skogestad and Postlethwaite [9], building a control structure involves first identifying the plant, getting a simplified model of the plant, deciding the input signals(manipulated variables) and the output signals (controlled variables), the choice and design of controller , implementation, testing, validation and tuning of the designed controller. These build up to meeting the controlling objective will be presented in this thesis.

The major tasks in this work can be subdivided into the following:

- Development of the model for the Quadruple tank; the model will be developed from first principle
- System Identification of the Quadruple tank model. This will be done both in DeltaV and from subspace based DSR method [10]. These system identification methods will be compared using a model performance index.
- Investigation of the control of the QTP using the inbuilt MPC strategy in DeltaV.

1.4 Thesis Outline

This thesis work will be broadly divided into eight (8) chapters.

The first and introductory chapter will cover a brief outline of the QTP, previous work that has been done and the significance of the research work

[1] DeltaV is a proprietary process automation system from Emerson Process Management

Chapter 2 covers more details of the quadruple tank process and its key components. The non-linear model of the QTP from first principle is also presented.

Chapter 3 describes the principle of Model Predictive Control (MPC). The fundamentals and principles of the MPC strategy in DeltaV is also introduced in this chapter.

Chapter 4 covers all aspects of developing the model of the process to be used for the implementation of MPC. The Integral Absolute Error (IAE) and Mean Square Error (MSE) performance indices will be discussed and used to compare the derived models.

Chapter 5 shows the implementation of the MPC in DeltaV, steps in programming in DeltaV MPC, the model identification and validation and the generation of the controller. The model identification using DSR method is also presented.

The results from the implementations and discussion of results will be discussed in Chapters 6 and 7 respectively.

Chapter 8 will cover the conclusion and recommendation for further work.

2 The Quadruple Tank

This chapter gives a comprehensive depiction of the process plant: the quadruple tank process and its key components. The non-linear model of the QTP from first principle is also presented.

2.1 The Quadruple Tank

The set-up of the QTP consists of four interacting tanks, two pumps, two level sensors and two valves. Tanks 1 and Tanks 2 are positioned below Tanks 3 and 4. Tanks 1 and 2 receive water from tanks 3 and 4 by gravitational action. The input to the process are the voltages to the pumps u_1 and u_2 . The voltages from the level sensors y_1 and y_2 are the output of the process. A schematic diagram of the QTP is as shown in Figure 2-1.

A reservoir placed below tanks 1 and 2 also serves to accumulate the water from these tanks. Pump 1 extracts water from the reservoir to tanks 1 and 4 while Pump 2 extracts water from the reservoir and discharge to Tanks 2 and 3. The water flow is split using two 3-way valves to feed these tanks. The valve position gives the ratio in which the flow from the pump is divided between the upper and lower tanks.

The objective is to control the water level in the lower tanks; which are the levels in Tank-1 and Tank-2. The strong coupling between the tanks creates a major difficulty in controlling the QTP.

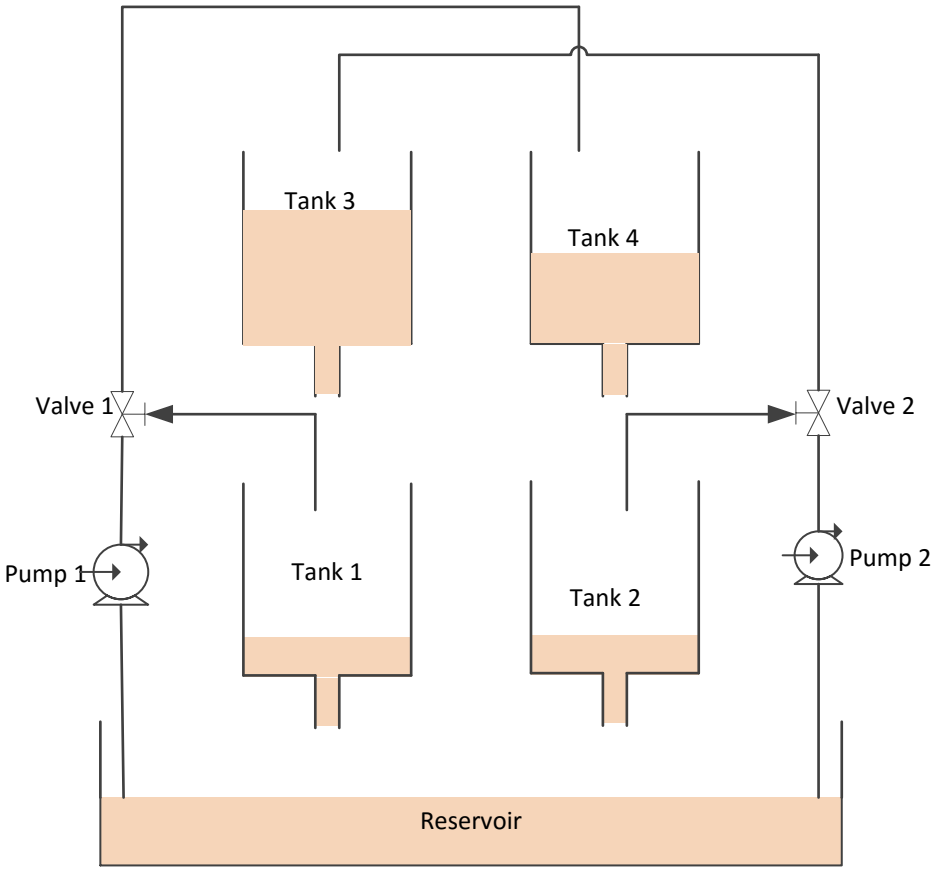


Figure 2-1: Schematic diagram of the quadruple tank

Figure 2-1 shows that the output of Tank 1 depends on Tank 3 and position of Valve 1 while the output of tank 2 depends on Tank 4 and the position of Valve 2.

The functions of the different parts of the quadruple tank process are summed up in Table 2-1

Table 2-1: Parts of the QTP and their functions

PART	FUNCTION
Pump 1	Supplies water from the reservoir to tanks 1 and 4
Pump 2	Supplies water from the reservoir to tanks 2 and 3
Valve 1	Used to manually control the water flow in tanks 1 and 4
Valve 2	Used to manually control the water flow in tanks 2 and 3
Level Sensor 1	Used to control the liquid level in tanks 1 and 4
Level Sensor 2	Used to control the liquid level in tanks 2 and 3
Reservoir	Stores the accumulated water

2.2 Development of Non Linear Model

Various models (both linear and nonlinear models) of the quadruple tank process exist and have been presented in different literatures [11], [2], [12], [13].

It is very important to understand the dynamics behavior of a system/process with time. Models show this dynamic behavior. Models can be built from first principle mechanics-based (such as physics, chemistry) or from system identification methods (that is by measured input and output data) [14].

As shown in the schematic diagram of the QTP in Figure 2-1, the following assumptions were made prior to the development of the model:

- The input to pump 1 is u_1
- The input to Pump 2 is u_2
- The two valves are set to make the process easy to control. The valves are set in a way that it defines the parameters $[\gamma_1\gamma_1] \in [0\ \ 1]$
- The flow to Tank 1 is proportional to γ_1
- The flow to Tank 4 is proportional to $1-\gamma_1$. That is if $\gamma_1 = 1$, then all flow from pump goes to Tank1, and if $\gamma_1 = 0$, then the all the flow goes to Tank 4
- The flow to Tank 2 is proportional to γ_2
- The flow to Tank 3 is proportional to $1-\gamma_2$. Similarly, at is if $\gamma_2 = 1$, then all flow from pump goes to Tank 2, and if $\gamma_2 = 0$ then the all the flow goes to Tank 3
- The flow through Tank 1 when the input voltage u_1 is applied is k_1u_1.
- The flow through Tank 2 when voltage u_2 is applied is k_2u_2.
- The flow through Tank 1 after crossing valve 1 is $\gamma_1k_1u_1$
- The flow through tank 4 after crossing valve 1 is $(1-\gamma_1)k_1u_1$
- The flow through tank 2 after crossing valve 2 is $\gamma_2k_2u_2$
- The flow through tank 3 after crossing valve 2 is $(1-\gamma_2)k_2u_2$

The mathematical modelling is developed from the mass balance equation which states that the rate of accumulation of mass in the system is given as the difference between the mass flow rate into the system and the mass flow rate out of the system. The rate of accumulation of mass in the system $\frac{dm}{dt}$,is expressed as;

$$\frac{dm}{dt} = \dot{m}_i - \dot{m}_o \qquad (3.1)$$

Where $\dot{m}_i$ the mass flow rate into the system and $\dot{m}_o$ is the mass flow rate out of the system. Expressing in terms of volumetric flow, it follows that

$$\rho \frac{dV(t)}{dt} = \rho \dot{q}_i - \rho \dot{q}_o \tag{3.2}$$

Where ρ is the density of the liquid. Since the liquid across all tanks is water, then the density is assumed constant. Hence

$$\frac{dh(t)}{dt} = \frac{1}{A} (\dot{q}_i - \dot{q}_o) \tag{3.3}$$

Where A and h is the cross sectional area and height of the tank respectively.

Using Bernoulli and the Torricelli equation, and equating the potential energy PE to the kinetic energy KE

$$PE = KE \tag{3.4}$$

$$mgh = \frac{1}{2} mv^2 \tag{3.5}$$

This follows that the velocity, v is given as

$$v = \sqrt{2gh} \tag{3.6}$$

Multiplying the velocity by the area of the outlet hole of the tank a_i, then the volumetric flow rate $q_{out\,i}$ of tank i for all $i \forall 1.2.3.4$ then becomes

$$q_{out\,i} = a_i v = a_i \sqrt{2gh_i} \tag{3.7}$$

Applying mass balance on the different tanks, the non-linear model of the QTP is given by the following state equations.

$$A_1 \frac{dh_1}{dt} = \gamma_1 k_1 v_1 + a_3 \sqrt{2gh_3} - a_1 \sqrt{2gh_1} \tag{3.8}$$

$$A_2 \frac{dh_2}{dt} = \gamma_2 k_2 v_2 + a_4 \sqrt{2gh_4} - a_2 \sqrt{2gh_2} \tag{3.9}$$

$$A_3 \frac{dh_3}{dt} = (1 - \gamma_2) k_2 v_2 + a_3 \sqrt{2gh_3} \tag{3.10}$$

$$A_4 \frac{dh_4}{dt} = (1 - \gamma_1) k_1 v_1 + a_4 \sqrt{2gh_4} \tag{3.11}$$

Where the definition of terms is as shown in Table 2-2

Table 2-2: Parameter definitions for the Non Linear model of the QTP

Parameter	Description	Unit

A_i	Cross section area of tank i	$(cm)^2$
h_i	Level of water of tank i	cm
a_i	Cross section area of outlet hole of tank i	$(cm)^2$
i	1,2,….4	$-$
v_1	Input signal to Pump 1	$Volt$
v_2	Input signal to Pump 2	$Volt$
γ_1	Valve constant of valve 1	$-$
γ_2	Valve constant of valve 2	$-$
k_1	Gain of pump 1	$cm^3V^{-1}s^{-1}$
k_2	Gain of pump 1	$cm^3V^{-1}s^{-1}$
g	Gravitational acceleration	cm/s^2

According to Johansson [12], the QTP is said to be in minimum and non-minimum phase based on the location of the multivariable zeros. The system is said to be in the minimum and non-minimum phase when Equations (*3.12*) and (*3.13)* are satisfied respectively.

$$0 < \gamma_1 + \gamma_2 < 1 \qquad (3.12)$$

$$1 < \gamma_1 + \gamma_2 \leq 2 \qquad (3.13)$$

Johansson further stated that the tank is easier to control when it is in the minimum phase. For all simulations done in this work, valve constants γ_1 and γ_2 were chosen as 0.7 and 0.6 respectively. And as such minimum phase is considered for all simulations.

The tank parameter values are summed up in Table 2-3.[11]

Table 2-3: QTP parameter values

Tank Parameter	Value	Unit
A_1, A_4	28	$(cm)^2$
A_2, A_3	32	$(cm)^2$
a_1, a_4	0.071	$(cm)^2$
a_2, a_3	0.057	$(cm)^2$
γ_1	0.7	$-$

γ_2	0.6	$-$
k_1	3.33	$cm^3V^{-1}s^{-1}$
k_1	3.35	$cm^3V^{-1}s^{-1}$
g	981	cm/s^2

3 Model Predictive Control

In this chapter, the principle of Model Predictive Control (MPC) is described and presented. The fundamentals and principles of the MPC strategy in DeltaV is also introduced.

3.1 Introduction to MPC

The history of MPC technology dates back as far as the 1960's from the work of Kalman et al [15] . The first generation of MPC technology (the Dynamic matrix Controller DMC and the Identification and Command IDCOM) were first designed in the 1970's. The second generation of MPCs, the quadratic dynamic matrix control (QDMC) were developed in the 1980's. The (IDCOM-M and Shell Multivariable Optimizing Controller SMOC, Hierarchical constraint control HIECON), the Predictive Control Technology, PCT and the Robust model predictive control, RMPC are considered to be the third generation of MPC controllers, and these came into existence in the 1990's. The fourth generation of MPC are the Robust Model Predictive control technology, RMPCT (a fusion of the RMPC and the PCT) developed in 1995 and the Dynamic Model Control Package, DMC-plus (a fusion of the SMCA and DMC) developed in 1998 [16].

MPC works by creating and solving a new optimization problem during every time step. It predicts future output values or states from the current time. The MPC uses a model of a system to calculate a predicted output signal over a prediction horizon in the future. A new optimization is created at each control interval within the prediction horizon by taking a new control input and feeding it into the system at each time step, while still considering the constraints [17]. This ability to handle constraints is integrated in the structure of MPC and this has made it more popular and more advantageous than other classical controllers such as the PID. MPC is also very effective in controlling MIMO systems because of the optimal way it handles interaction between the controller variables[17]. MPC can handle delays (both process input and output delays) in a very efficient way.

However, one major drawback of MPC is that a model of the plant or process must be present before the controller is implemented. The high computational time involved in its implementation also poses a difficulty in real time applications. Since the optimization

problem has to be solved within each of these sampling times, choosing a suitable sampling time can also pose another challenge.

An MPC algorithm consists of the following:

- Cost Function

The cost function or control objective is a criterion that measures the difference between the future outputs $y_{k+1|L}$ and a specified reference $r_{k+1|L}$ while considering that the control signal u_k is costly. The cost or objective function is used to solve the optimization problem. The objective is a measure of the behavior of the process over the prediction horizon. [17]
Different objective criterions can be implemented in MPC; ranging from economic objectives such as profit maximizing, loss or cost minimization, setpoint tracking, sum of squared error, sum of absolute errors etc.

- Constraints

A constraint is a limitation or restriction; it is the condition in an optimization problem which the solution must satisfy. The ability to handle constraints is integrated in the structure of MPC. Commonly considered constraints are the system input amplitude constraints, the system input rate of change constraints and the process output constraints [17].
The system input amplitude constraint is amplitude constraints on the input signal written mathematically as

$$u_{k|L}^{min} \leq u_{k|L} \leq u_{k|L}^{max} \quad (3.1)$$

The system input rate of change constraint are limitations on the rate of change. This is written mathematically as

$$\Delta u_{k|L}^{min} \leq u_{k|L} \leq \Delta u_{k|L}^{max} \quad (3.2)$$

Process output constraints are constraints on the process output which can be mathematically represented by

$$y^{min} \leq y_{k+1|L} \leq y^{max} \quad (3.3)$$

Where L is the prediction horizon.

- Model of the Process

A process model that predicts the future process output up to the prediction horizon must be present for MPC to be implemented. The unavoidable existence of a model is the major drawback to the implementation of MPC. Mechanistic models are developed from first principles where possible, otherwise models can be identified from Blackbox or Subspace methods such as DSR method by Di Ruscio[10]. The model is used to evaluate the predicted output $y_{k+1|L}$ and the states over a horizon in the future[17].

A conceptual picture of MPC is shown in Figure 3-1 and Figure 3-2. MPC uses the model of the process to predict future outputs based on the current input values. Then it uses the predicted information to calculate an optimal value of future inputs with respect to a defined cost function [18].

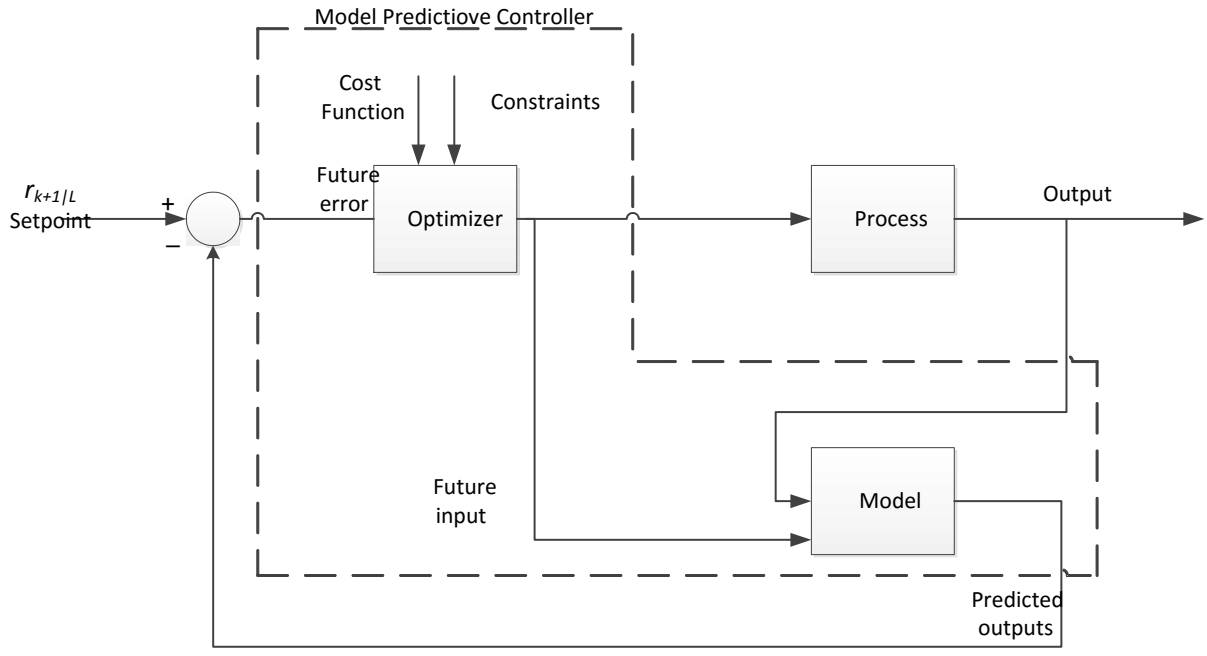


Figure 3-1: Conceptual picture of MPC-1 [18]

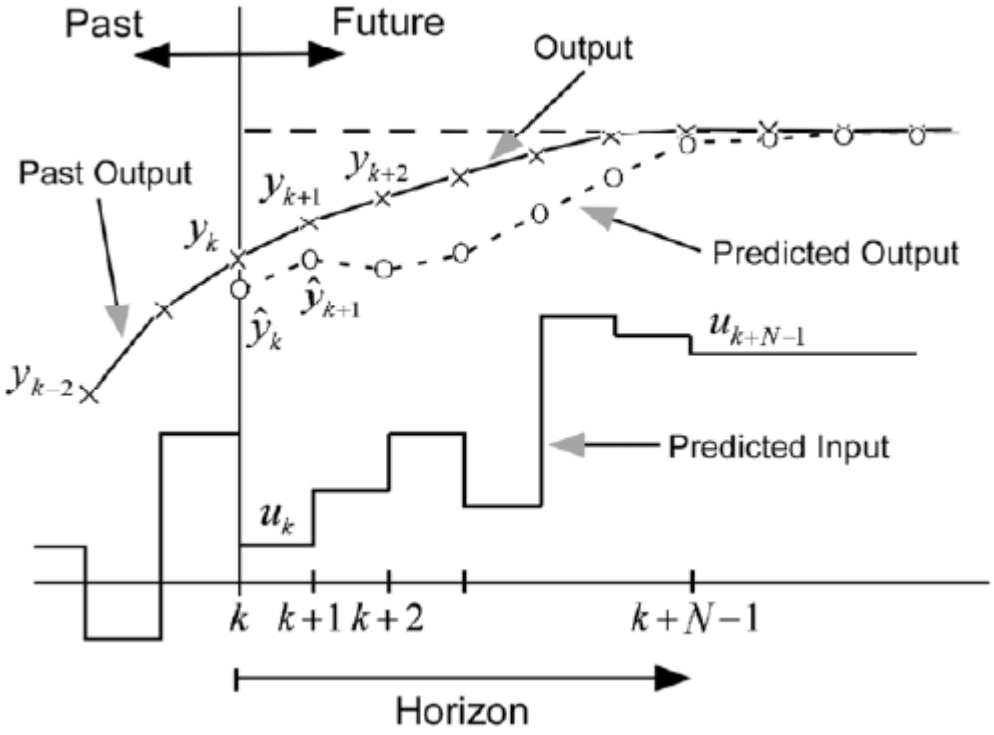


Figure 3-2: Conceptual picture of MPC-2 [18]

3.2 Defining Prediction Models

The prediction model is an integral part of the optimization problem which primarily describes the relationship between future outputs and future control inputs. MPC applications differ in the ways their prediction models are developed from the models of the process[17]. Prediction models are developed from state space models, Finite Impulse Response (FIR) and step response models, transfer function models, the ARMAX models etc.

3.2.1 Prediction Models from State Space Models

Prediction models from state space models are very popular amongst variants of MPCs because of the ease of deriving them. It is also relatively easy and common for models like transfer function models, ARX models, FIR etc. to be transformed to state space models for use in MPC.

Given the general form of a state space model described by

$$x_{k+1} = Ax_k + Bu_k \qquad (3.4)$$

$$y_k = Dx_k \tag{3.5}$$

The prediction model can be described simply by

$$y_{k+1|L} = F_L u_{k|L} + p_L \tag{3.6}$$

Where

$$F_L = [O_L B \quad H_L^d] \tag{3.7}$$

$$p_L = O_L A x_k \tag{3.8}$$

x_k is the state of the process which can be calculated simply from the known past inputs and outputs as shown by Di Ruscio [19].

Where O_L is the extended observability matrix of the pair A and D.

H_L^d is the Toeplitz matrix of impulse responses

L is the prediction horizon and $F_L \in \mathbb{R}^{Lm \times Lr}$ is a constant matrix derived from the model of the process. $p_L \in \mathbb{R}^{Lm}$ is a vector dependent on the number of inputs and outputs.

In cases where it is not possible to measure or compute the state, it can be estimated using a state observer such as a Kalman filter.

Equation (3.6) will be the basis for the MPC algorithm to compute the predicted outputs in the future.

Given an objective function defined as

$$J_k = \left(y_{k+1|L} - r_{k+1|L}\right)^T Q\left(y_{k+1|L} - r_{k+1|L}\right) + u_{k|L}^T P u_{k|L} \tag{3.9}$$

Where J_k is the objective function, $u_{k|L}$ is the input, and $y_{k|L}$ is output. Q and P are user defined symmetric and positive weighting matrices. $r_{k|L}$ is the reference.

Substituting p_L to the objective function, reduces it to a quadratic function of standard form

$$J_k = u_{k|L}^T H u_{k|L} + 2f^T u_{k|L} + J_0 \tag{3.10}$$

Where

$H = F_L^T Q F_L + P$

$f = F_L^T Q(P_L - r_{k+1|L})$

$J_0 = (P_L - r_{k+1|L})^T Q(P_L - r_{k+1|L})$

The optimization problem is thus minimizing the objective function J_k with respect to $u_{k|L}$.

Where the solution is given by $u_{k|L}^* = -H^{-1} f$

3.2.2 Prediction Models from FIR and step response models

Prediction models can be developed from the FIR and step response models. Considering the state space model defined above in Equations *(3.4)* and *(3.5),* an expression for

$$y_k = DA^i x_{k-i} + DC_i u_{k-i|i} \quad (3.11)$$

Where C_i is the extended controllability matrix.

Assuming that the system is stable, then $A^M \approx 0$ when $M = i \geq 1$ is large. Then we have that

$$y_k = DC_M u_{k-M|M} \quad (3.12)$$

Where DC_M is a matrix of impulse response matrices and M is the model horizon

$$DC_M = [H_1 \quad H_2 \quad \dots \quad H_M] = [DB \quad DAB \quad \dots \quad DA^{M-1}B] \quad (3.13)$$

The input output model defined in equation *(3.12)* is the FIR model which can be used to express y_{k+1} and subtracting y_k gives the step response model;

$$y_{k+1} = y_k + C_M \Delta u_{k+1-M|M} \quad (3.14)$$

$$\Delta u_{k+1-M|M} = u_{k+1-M|M} - u_{k-M|M} \quad (3.15)$$

A prediction model can be defined from Equations (3.12) and (3.14).

It is worthy to note here that model identification in DeltaV MPC is based on a step response being created from the FIR and the ARX models. Details will be explained in the next section. The Dynamic Matrix Controller (DMC) is an example of an algorithm of MPC that uses the step response model.

3.3 MPC Strategy in DeltaV

DeltaV is a process automated system from Emerson Process Management. DeltaV has the advantage of a very friendly common operator interface, increased reliability, it is very easy to configure, and has a user friendly testing environment.

DeltaV MPC works on the already discussed principle of MPC where the controller learns from the past to predict the future output by using the mathematical model of the process.

The DeltaV MPC function blocks are primarily used for implementing control of different interactive processes in DeltaV MPC. Three different MPC function blocks exist in DeltaV and these are the MPC, MPCPro and MPCPlus function blocks, they are as shown in Figure 3-3.

Figure 3-3: MPC blocks in DeltaV

The DeltaV Predict is used to commission the MPC function block while the DeltaV PredictPro is used to commission MPC-PRO and MPC-PLUS function blocks. For the purpose of the controlling the QTP (2 inputs and 2 outputs), the MPC function block (allows for a maximum of 8X8 input-output interaction) will be used and hence discussed further.

The DeltaV MPC function block controls interactive processes with consideration for the measurable operating constraints and disturbances. The MPC function block is launched in Control Studio in DeltaV environment.

In DeltaV Predict application, an automated test of the process is run by automatically collecting data from the process, these data is used to create a step response of the process.
The DeltaV Predict implements MPC technology in controlling small and medium sized multivariable processes within different measurable operating constraints and disturbances.
The DeltaV MPC function block is used to control an interactive process while taking into consideration the measurable operating constraints and measurable disturbances. Multivariable control is implemented effectively in the DeltaV system with the MPC function block. The MPC function block is used by simply dragging and dropping into control studio, and downloading afterwards. The input and outputs of the MPC function block is configured in DeltaV control studio, in a manner that suits your control structure.
The inputs to the MPC function block can be any of the following:

- Controlled variable (CNTRL): The controlled parameter which is maintained at a setpoint by adjusting the manipulated variable

- Disturbance(DSTRB): The disturbance on constraint and controlled parameter whose impact is reduced by adjusting the manipulated input
- Constraint(CNSTR)

The output of the MPC function block is

- Manipulated variable (MNPLT): The manipulated output from the function block. This variable is adjusted either automatically by the function block or manually by the user.

The controlled variable (CV), the manipulated variable (MV), the constraint variables, and the disturbance variables are defined by the user in the MPC function block. It is imperative to identify these variables during the programming in DeltaV.

3.3.1 Generation of MPC Controller in DeltaV

The MPC controller minimizes future control errors and control moves. Generation of the MPC controller in DeltaV is fully automated. The control calculations assume that your process response is reasonably linear over its normal operating range. If the process model has been correctly identified, the default setting of controller generated for the process gives optimal performance. However if the default controller setting do not give good tracking of setpoint, adjustments can be made as an expert user.

MPC in DeltaV has its roots in Dynamic Matrix Control (DMC) where a dynamic matrix is built from step responses such that the process outputs can be predicted from manipulated variables over the control horizon.

The MPC Controller minimizes the squared error of a controlled variable over the prediction horizon and also the squared error of the controller output over the control horizon in the following way

$$\min_{\Delta U(k)} \left(\Gamma^{y}(CV(k) - R(k))^2 + (\Gamma^{u}\Delta U(k))^2 \right) \qquad (3.16)$$

Where p is the prediction horizon

c is the control horizon

$\mathrm{CV(k)}$ is the controlled output p-step ahead prediction vector

$\mathrm{R(k)}$ is the p-step ahead setpoint vector

$\Delta U(k)$ is the c-step ahead incremental controller output moves vector

Γ^{y} is a diagonal penalty matrix on the controlled output error

Γ^{u} is a diagonal penalty matrix on the control moves

The solution for the process with dynamic matrix S^{n} satisfying Equation (3.16) is in the form

$$\Delta U(k) = (S^{nT}\Gamma^{yT}\Gamma^{y}S^{n} \quad + \quad \Gamma^{uT}\Gamma^{u})^{-1}S^{nT}\,\Gamma^{yT}\Gamma^{y}E_{P}(k) \qquad (3.17)$$

Where

S^{n} is the $p \times c$ process dynamic matrix built from the step responses of dimensions $p \times c$ for a SISO model and $pn \times cm$ for a MIMO model with m manipulated inputs and n controlled outputs.

$E_{P}(k)$ is the error vector over the prediction horizon p

3.3.2 The Penalty of moves (POM) and the Penalty of error (POE)

The POM and the POE are convenient tuning parameters used in getting the desired controller performance.

- POM

The POM parameter impacts the robustness of the controller. The POM is a basic controller tuning parameter at the phase where the controller is been generated, increasing the POM makes the controller less aggressive and decreasing it makes the control action more aggressive and results a faster control response. The POM is defined independently for each of the manipulated variable(s).

- POE

The POE is also known as a 'controlled variable tuning weight'. It allows more weight to be given to a specified controlled variable.

However, if the controller performance is satisfactory using the default settings, there is no need to alter these parameters.

3.3.3 Output Constraints Handling

The MPC configuration does not take any action on the constrained variable except the constraint is violated, in which case, the MPC controller changes the working setpoint of the selected controlled variable.

$$\Delta SP_{CV} = -rG_{CV-AV}\Delta AV \quad (3.18)$$

Where

ΔAV is the magnitude of the predicted steady state constraint violation

G_{CV-AV} is the gain relationship between the constraint variable *AV* and *CV*

r is the relaxation factor, which is always less than 1

ΔSP_{CV} is the change in working setpoint of the controlled variable

4 Development of Process Model/System Identification of Process Model

This chapter covers all aspects of developing the model of the process, such that this model will be used for the MPC.

The IAE and MSE performance indices will be discussed and used to evaluate model performance.

4.1 Model identification in DeltaV

Model identification in DeltaV is based on a step response model. The step response model describes the relationship between the process inputs and future outputs over the prediction horizon. The step response model is also validated by a support strategy contained as an integral part of the model generation. The verified model is then used to generate a controller.

4.1.1 DeltaV Predict Algorithm

Identification of the step response in DeltaV Predict is done by both the FIR and the ARX. Comparing models from these two techniques helps in confirming correct representations of predicted models. The FIR has the advantage of not requiring a preliminary knowledge about the process while the ARX uses fewer coefficients in the calculation. The model is validated by comparing the real process data to the simulated process data. Process optimization is achieved by either minimizing or maximizing a selected process input.

DeltaV Predict uses the step response modelling which makes prediction of process outputs available for display in the application. It also computes the predicted error vector, which serves as an input to the MPC controller. The MPC function block then develops future process outputs as a process state and uses modified state space for the process modelling.

In a SISO process, the predicted future process output is as shown below

$$x_{k+1} = Ax_k + B\Delta u_k + F\Delta w_k \quad (4.1)$$

$$y_0 = Cx_{k+1} \quad (4.2)$$

Where

$x_k = \left[y^0,\ y^1, y^i, \ldots\ldots, y^{p-1}\right]^T$ is a vector of the future output prediction *0,1.....i,.....p-1* steps ahead at the time k.

A is a shift operator which is defined by as $Ax_k = \left[y^0,\ y^1, y^i, \ldots\ldots, y^{p-1}\right]^T$

$B = \left[b_0, b_1, \ldots, b_i, \ldots, b_{p-1}\right]^T$ is vector of p step response coefficients

$\Delta u_k = u_k - u_{k-1}$ is change on the process input/controller output at the time instant k

Δw_k is the process output measurement-process model output, mismatch resulted from the noise, unmeasured disturbances and model inaccuracy.

F is the p dimension filter vector with unity default values

P is the dimension vector with unity default values

C is the operator that takes the first component of the x_{k+1} vector

For a multivariable process, with *m* inputs and *n* outputs, the vector x_k has dimension *n*p* and vector *B* is converted into a matrix with dimension *n*p* rows and *m* columns.

The FIR model uses a short horizon process to avoid over fitting the model as it provides an initial part of the step response, such that it is enough to evaluate the dead time using a heuristic approach. The dead time is then used in the ARX model which has fewer coefficients that the FIR.

In MIMO systems, superposition is applied from each input on every output. The FIR and the ARX model for a SISO system is defined by Equations *(4.8)* and *(4.9)* respectively.

$$\Delta y_k = \sum_{j=1}^{p} h_i \Delta u_{k-1} \qquad (4.3)$$

Where $\Delta y_k = y_k - y_{k-1}$is the change of the process output at the time instant k, p is the prediction horizon, Δu_{k-1} is the change of the process input at the time instant $k-1$.

$$y_k = \sum_{j=1}^{A} \alpha_i y_{k-1} + \sum_{j=1}^{V} \beta_i u_{k-d-1} \qquad (144)$$

Where A and V are auto regressive and moving average orders of ARX with a default value of 4, $\alpha_i$and β_i are the coefficients of the model and d is the dead time.

4.2 System identification based model

System Identification uses statistical methods to build mathematical models of dynamic systems from measured or observed data [10]. The measured inputs and outputs , measured states, measured properties, measured disturbances, feedback signals etc all serves as the input, and the model of the system then becomes the output [10] .

4.3 DSR method of system Identification

System identification method where subspace method is used to identify a system from a set of known input and output data and also the system order is known as the combined Deterministic and Stochastic system and Realization (DSR) [10]. The complexity of first principle models in system identification have made the DSR[10] method become more popularly used. The DSR system identification is a realization based approach for estimating the state space models from known input and output data. The DSR method is very easy to use and very effective for both SISO AND MIMO systems.

4.3.1 DSR Algorithm

The primary step in the DSR algorithm is to identify the system order and the extended observability matrix from the column space of known data matrix. The DSR algorithm is very simple to implement and based on the singular value decomposition.

The DSR algorithm is implemented in the D-SR toolbox in MATLAB. The DSR algorithm estimates the system order and values of the model matrices A, B, D, E, F and the initial state vector x_0.

The algorithm is shown below in the equation (4.5)

$$[A, \quad B, \quad D, \quad E, \quad F, \quad x_0] = dsr\,[Y, \quad U, \quad L, \quad g, \quad J, \quad M, \quad n] \tag{4.5}$$

Where

$$Y = \underbrace{\begin{bmatrix} {y_0}^T \\ {y_1}^T \\ \vdots \\ y_{N-1}^T \end{bmatrix}}_{Known\ data\ of\ output\ variables} \in \mathbb{R}^{N \times m} \tag{4.6}$$

$$U = \underbrace{\begin{bmatrix} {u_0}^T \\ {u_1}^T \\ \vdots \\ u_{N-1}^T \end{bmatrix}}_{Known\ data\ of\ input\ variables} \in \mathbb{R}^{N \times r} \quad (4.715)$$

g is the structure parameter used to predict E. If $g = 0$, then $E = 0$

N is the number of samples, m is the number of output variables, r is the number of input variables, L is the future horizon that is used for predicting the system order.

J is the user-defined size of the past horizon

n is the number of states

A, B, D, E are the matrices of the state space model.

4.4 Model Performance Indices

The mean square error (MSE) and the integral absolute error (IAE) method will be used to compare between the different models, both for the DSR identified models and the model identified by DeltaV Predict.

4.4.1 MSE

The mean square error (or mean squared deviation) gives a measure of the square of the errors, it gives the square of the difference between the real value and the what is estimated[20].

The MSE is expressed mathematically as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x - \hat{x})^2 \quad (4.8)$$

Where N is the number of the samples or observations. $(x - \hat{x})^2$ is the square of the errors

The smaller the value of the MSE, the closer the predicted model is to the real process.

4.4.2 IAE

The Integral absolute error gives the absolute value of the error between actual and predicted output. The IAE is expressed mathematically as

$$IAE = \int_0^N |e|\, dt = \int_0^N |r - y|\, dt \qquad (4.9)$$

Where e is the control deviation error and N is the number of the samples or observations.

5 Implementation of MPC

In this chapter, the steps for the programming in DeltaV will be shown, the model identification and validation in DeltaV MPC and the generation of the controller.

The model identification using DSR method will also be shown, and the implementation of the MPC with integral action.

5.1 Collection of Data

Communication of the QTP with DeltaV was not possible because the input to the QTP laboratory plant (pumps) require an input of 0-10Volts. This is not compatible with The DeltaV substation (4 to 20mA output signal). It was therefore not possible to connect the QTP directly to the DeltaV substation directly.

However, a set of input and output data has been collected from previous work done by Di Ruscio [6] where LabVIEW and a simple DAQ device from National Instruments was used to log the data at a sampling time of 0.1seconds. These data was used for the modelling in both DSR and DeltaV Predict. The data matrices are as defined below

$$U \in \mathbb{R}^{N \times 2} \quad (5.1)$$

$$Y \in \mathbb{R}^{N \times 2} \quad (16)$$

Where U is the input; Y is the output of the process.

N is the number of samples.

A simple plot of the input and output of the process is shown in Figure 5-1 and Figure 5-2 respectively.

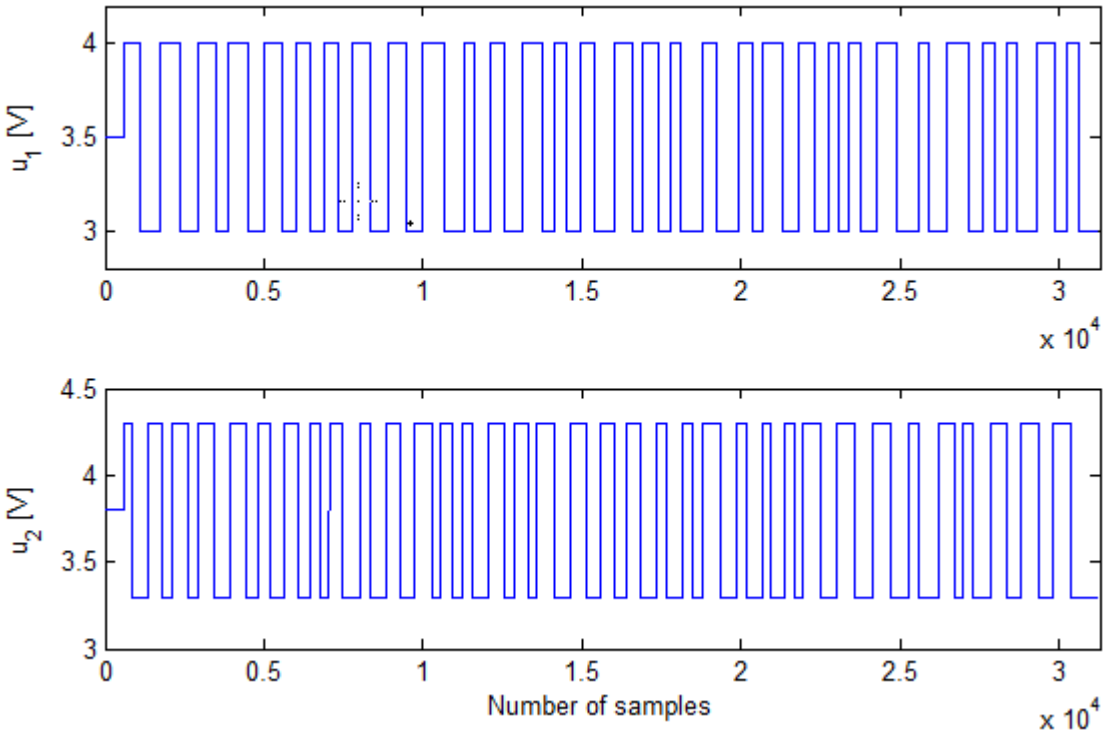


Figure 5-1: Input signals

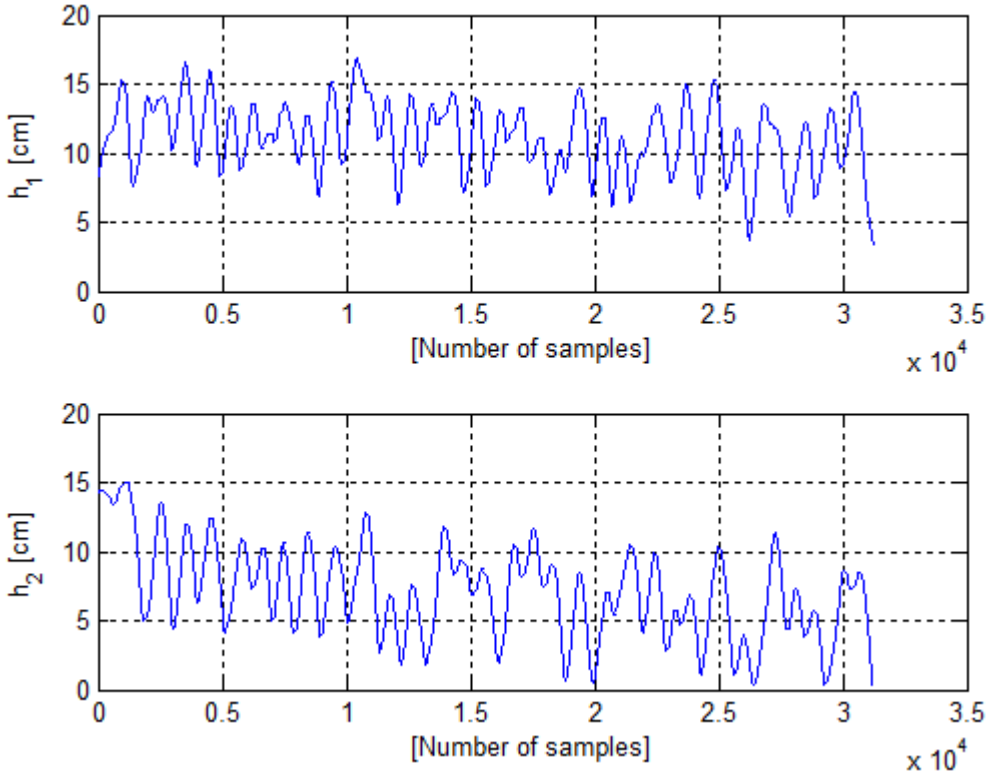


Figure 5-2: Output signals

5.2 Model Prediction in DeltaV Predict

Model predictive strategy in DeltaV is based on a dynamic model of the process. Process identification, modelling and controller generation are performed within MPC DeltaV. The identified process model and the process inputs are used to predict future variations of the process.

DeltaV Predict uses the Finite Impulse Response (FIR) and the Auto-Regressive with eXternal inputs (ARX) modelling techniques. It first creates a step response model; the step response model provides an insight into the dynamics between the process outputs and inputs.

External data from a non DeltaV environment will be used in DeltaV Predict to create a model of the process. For best results in using an external data, it is extremely important that the data should correctly reflect normal conditions over the operating range of the process.

Figure 5-3 shows a flow chart of how DeltaV MPC handles external historical data. .

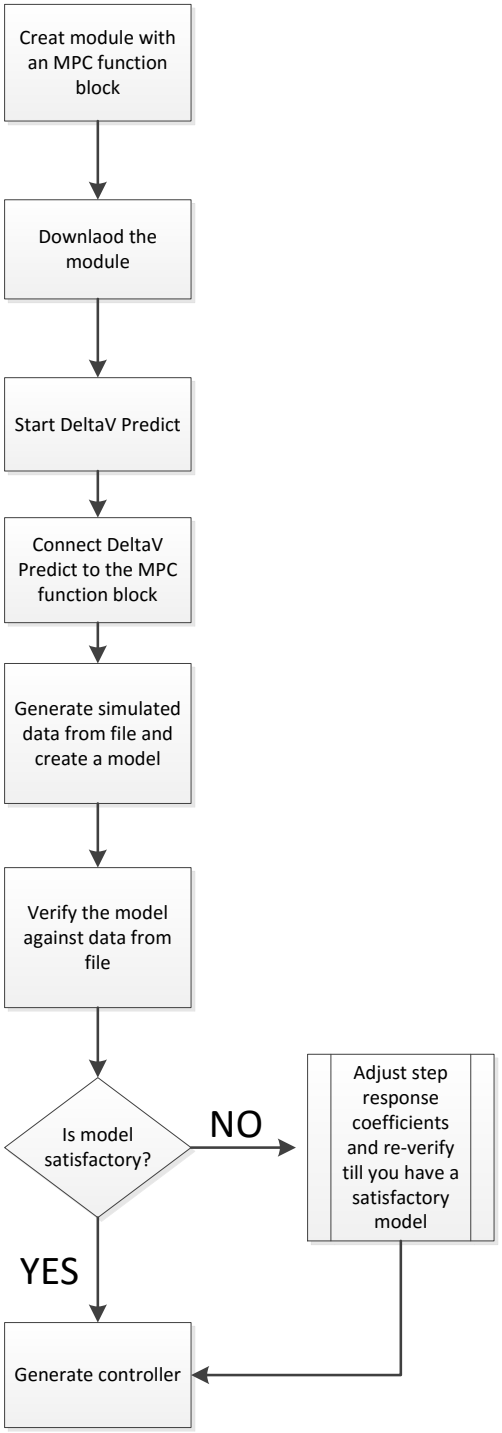


Figure 5-3: Flow chart for implementation in DeltaV MPC using external historical data

As shown in Figure 5-3, the desired MPC block that references the inputs and outputs of the process is created and downloaded. Figure 5-4 shows the MPC blocks used for the QTP with the inputs and outputs referenced accordingly.

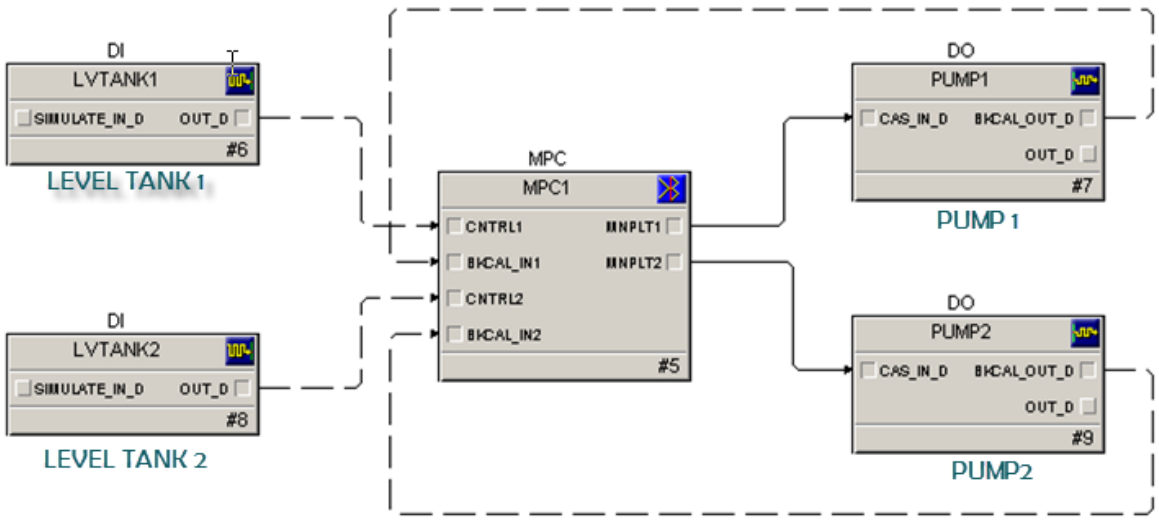


Figure 5-4: Input and Output of process in DeltaV

After the module is downloaded, then DeltaV Predict is started. The external data is loaded into DeltaV Predict, which uses these data and creates a step response model of the process. After using DeltaV Predict to identify this model, it is very important to review the model and ensure that it reflects the correct dynamics of the process. The validation errors between the calculated output and the actual output for the selected data must be within acceptable limit. If there is a relatively large validation error, then it is necessary to go back to the step response and effect necessary adjustments.

Using data that is not sufficiently excited or data that is very noisy or shows insufficient time may lead to unsatisfactory process models. But situations where it is not possible to get better process data, it is also possible to correct the model based on the prior knowledge of the process. This can be done by studying trends in the measurements or process simulations or analyzing the data outside DeltaV. Microsoft Excel or MATLAB can be used to study simulations on these data.

Another check for sub optimal models is to view the FIR and the ARX models on the same plot, if they differ from each other, then the model is poor.

After the model has been identified by DeltaV Predict, a controller is generated accordingly.

DeltaV Predict provides a simulation environment that enables testing of the MPC strategies many times faster than real-time. After getting satisfactory results, this can now be tested on the real process.

5.3 Generation of Controller

After a satisfactory model has been created in DeltaV Predict, the expert option in DeltaV enables the creation of the model and the controller generation. DeltaV uses the model to provide default settings for controlling of the process. It is possible to modify these generated default controller parameters generated if not satisfactory. However, the default settings in most cases provide good responses.

In cases where the default settings of the controller do not provide the desired robustness, tuning is required. The Penalty of move and the Penalty of Error are adjusted to meet user requirements. But caution must be exercised when asking these adjustments. A new controller generation and download is necessary after any change in these parameters.

5.4 System Identification Using DSR

The DSR is a subspace identification method. It is implemented in the DSR toolbox for MATLAB developed by David Di Ruscio.

The DSR method will be used to identify a linear state space model using the measured input and output data from the process. The input and output data is simply fed into the DSR algorithm to generate a linear state space model. The model is subsequently validated by using a different set of data different from those used in the identification.

5.4.1 Removing trends in Data

Trends in data are non-zero constants or mean values or low frequency noises. It is very important to remove trends from the raw process data in order to get a suitable model. It is also imperative for the trends to satisfy the steady state relationship of the system[10]. The sample mean is usually used as an approximate value of the steady state trends.

$$dy_t = y_t - y^o \quad (5.3)$$

$$du_t = u_t - u^o \quad (5.4)$$

Where y^o and u^o are mean values expressed as

$$y^o = \frac{1}{N}\sum_{t=1}^{N} y_t \quad (5.5)$$

$$u^o = \frac{1}{N}\sum_{t=1}^{N} u_t \quad (17.6)$$

The model can then be identified from the adjusted data.

5.4.2 System Identification from Real data

Using DSR method, the first 15000 samples was used for the identification and the other 15000 samples was used in the validation process.

5.5 MPC with Integral action

The diverse models used in implementation of MPC results in different algorithms of MPC. In [5], Di Ruscio presented an MPC controller for a MIMO system with integral action where the DSR method was used to identify the model.

Using first principle a new set of data is simulated of the QTP to be used for the implementation in MPC with integral action. DSR system identification is used to develop a model of the process. The model is then used to define the prediction model.

MPC from the DSR identified models has been compared from MPC formulations from models developed from first principle method (model free MPC), the MPC from DSR model was more robust, showed better and faster set point tracking [6]. The robustness and ease of implementation has resulted in its high relevance in recent years.

The development of the MPC from the DSR identified model can be broken down into the following steps

Given a process model of the form

$$x_{k+1} = Ax_k + Bu_k + v \quad (5.7)$$

$$y_k = Dx_k + w \quad (5.8)$$

Where $\mathbf{x_k} \in \mathbb{R}^n$ is the state vector, initial state x_0, $\mathbf{u_k} \in \mathbb{R}^r$ is the input vector, $\mathbf{y_k} \in \mathbb{R}^m$ is the output vector

A, B and D are identified subspace matrices system from DSR method based on trended input and output data.

$\mathbf{v}$ and $\mathbf{w}$ are unknown process and measurement noise vectors. $\mathbf{v}$ and $\mathbf{w}$ are not known and but can be eliminated by subtracting values of x_k and y_{k-1} from Equations (5.7) and (5.8).

$$x_k = Ax_{k-1} + Bu_{k-1} + v \quad (5.9)$$

$$y_{k-1} = Dx_{k-1} + w \quad (5.1018)$$

Further simplification leads to

$$\Delta x_{k+1} = A\Delta x_k + B\Delta u_k \quad (5.1119)$$

$$y_k = y_{k-1} + D\Delta x_k \quad (5.1220)$$

Where $\Delta x_{k+1} = x_{k+1} - x_k$ and $\Delta u_k = u_{k+1} - u_k$

The state space model can be derived by augmenting Equations (5.11) and (5.12)

$$\underbrace{\begin{bmatrix}\Delta x_{k+1}\\ y_k\end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix}A & 0_{n\times m}\\ D & I_{m\times m}\end{bmatrix}}_{\tilde{A}}\underbrace{\begin{bmatrix}\Delta x_k\\ y_{k-1}\end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix}B\\ 0_{m\times r}\end{bmatrix}}_{\tilde{B}}\Delta u_k \quad (5.13)$$

$$y_k = \underbrace{\begin{bmatrix}D & I_{m\times m}\end{bmatrix}}_{\tilde{D}}\underbrace{\begin{bmatrix}\Delta x_k\\ y_{k-1}\end{bmatrix}}_{\tilde{x}_k} \quad (21)$$

This leads to the state space model given by

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}\Delta u_k \quad (5.15)$$

$$y_k = \tilde{D}\tilde{x}_k \quad (5.16)$$

The Prediction model can then be defined as expressed as described in section 3.2.1.

6 Results (Modelling, Verification and Control)

In this chapter, all results from the implementation done in chapter 5 both MatLab with the DSR method and DeltaV (FIR and the ARX) are presented. The model identification and verification will be done in DeltaV and also using the DSR method.

6.1 Implementation in DeltaV

Table 6-1 shows parameters for design of the step responses. The step responses are shown in Figure 6-1.

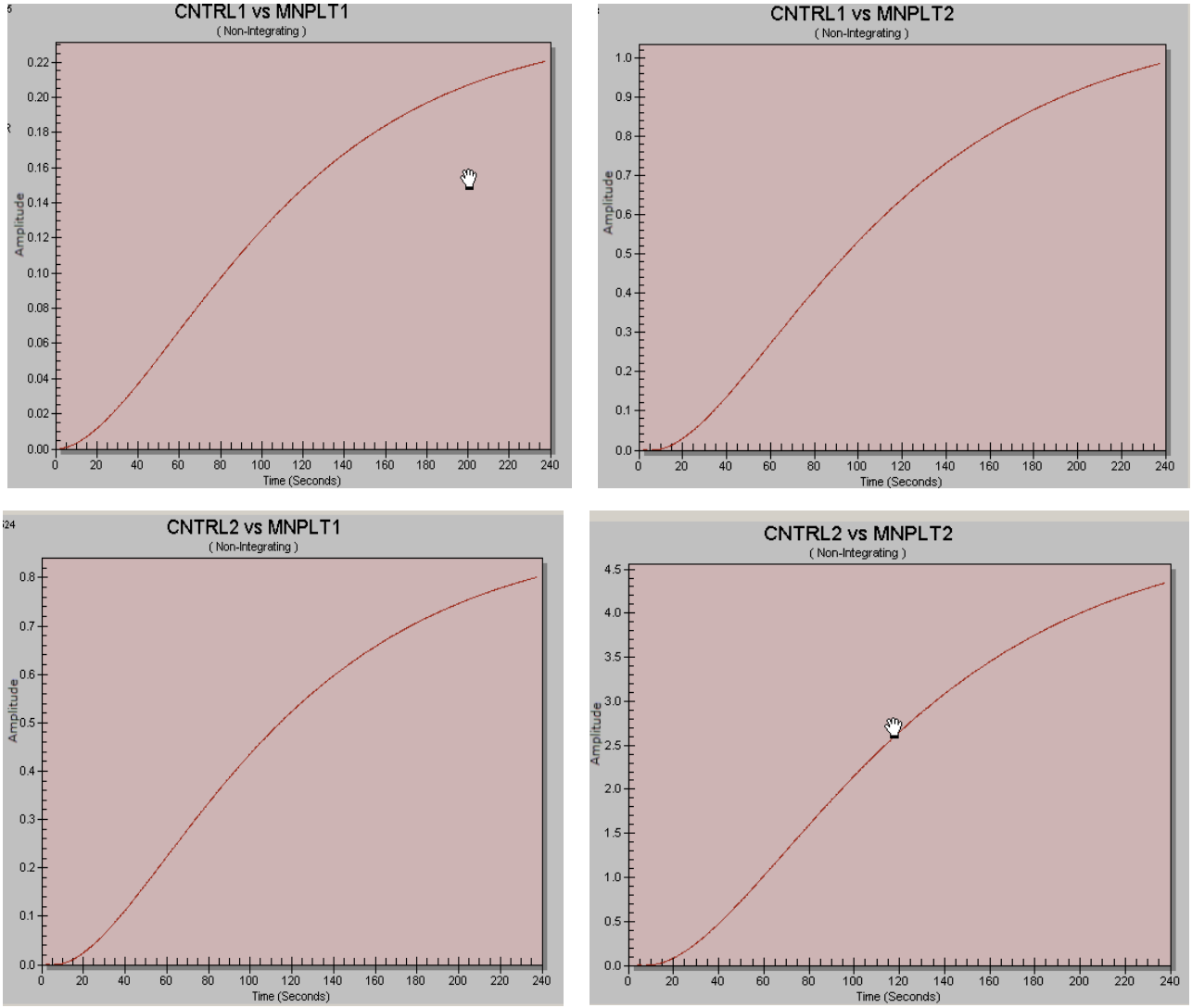


Figure 6-1: Step Response models

Table 6-1: Step Response Design

MPC Parameter	Pump 1	Pump2
Level Tank1	$K = 0.6sec$ $T_1 = 16.9231\ sec$ $T_2 = 0sec$ $\tau = 2sec$	$K = 0.944099$ $T_1 = 68.6007sec$ $T_2 = 11.3993sec$ $\tau = 4sec$
Level Tank2	$K = 1.93264$ $T_1 = 58.6844sec$ $T_2 = 21.3156sec$ $\tau = 6sec$	$K = 1.87221$ $T_1 = 55.3846sec$ $T_2 = 0sec$ $\tau = 6sec$

Where T_1 is the first time constant, T_2 is the second time constant, τ is the process dead time, and K is the process gain.

The step response model did not give a good representation of the dynamics of the process and this is evident in the verified models in Figure 6-2 and Figure 6-3 for the control output in tanks 1 and tanks 2 respectively. The squared error for the first controlled output is 3.38 and for the second controlled output 7.60 and hence the predicted models are relatively poor.

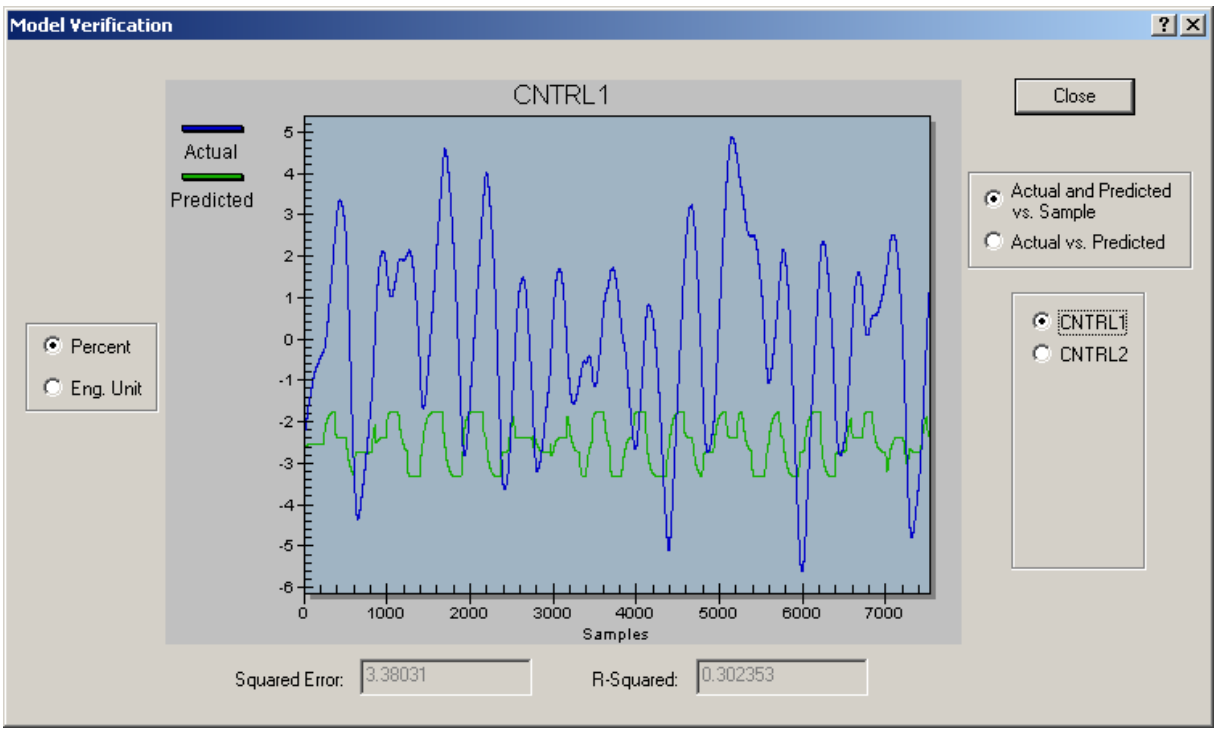


Figure 6-2: Actual and predicted model for control output 1 (Level in Tank1)

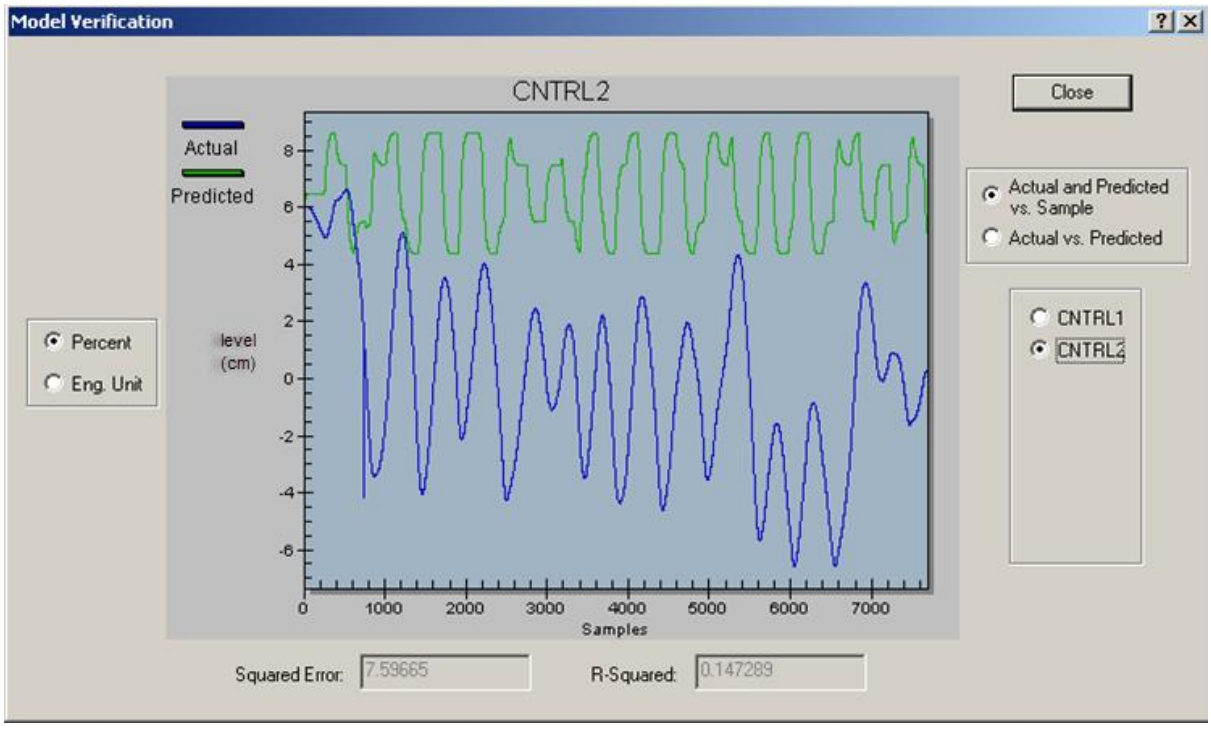


Figure 6-3: Actual and predicted model for control output 2 (Level in Tank2).

In generating the controller, the default setting did not show good response because of the poor model of the process. This was somewhat expected. To improve the controller robustness, adjustments had to be made to the default setting. The Penalty of move was adjusted for this purpose.

To achieve good setpoint tracking, the penalty of move was altered for both of the manipulated variables.

The controlled outputs (levels in tanks 1 and 2) are shown in Figure 6-4 and Figure 6-5 respectively.

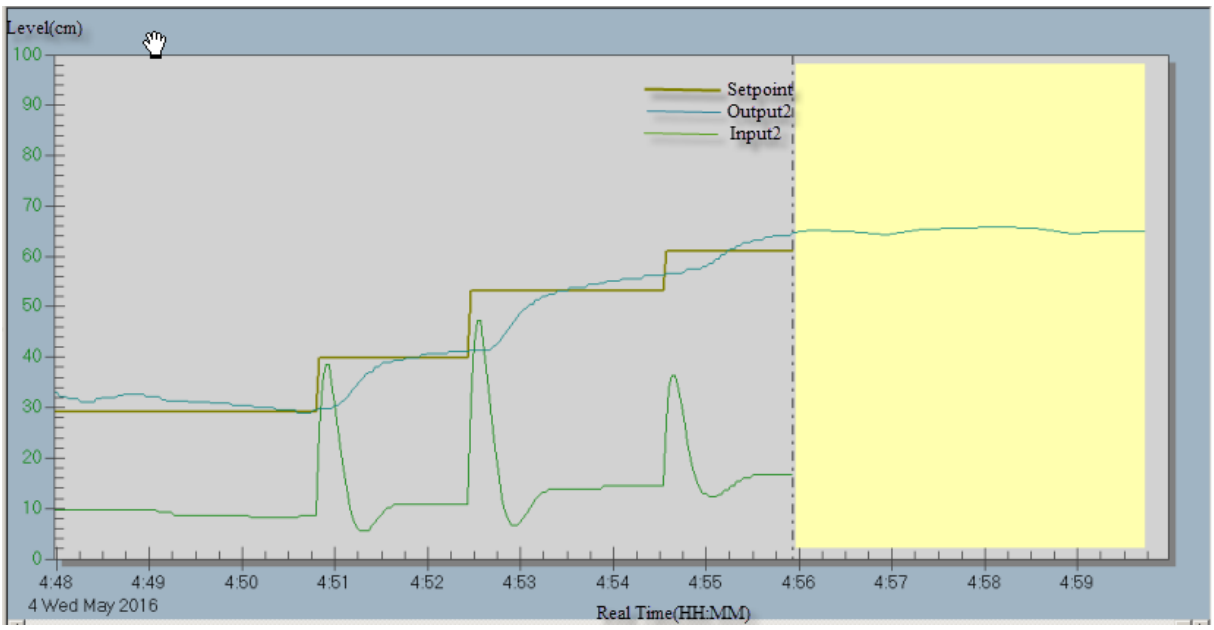


Figure 6-4: Simulation for the MPC control for level in Tank 2

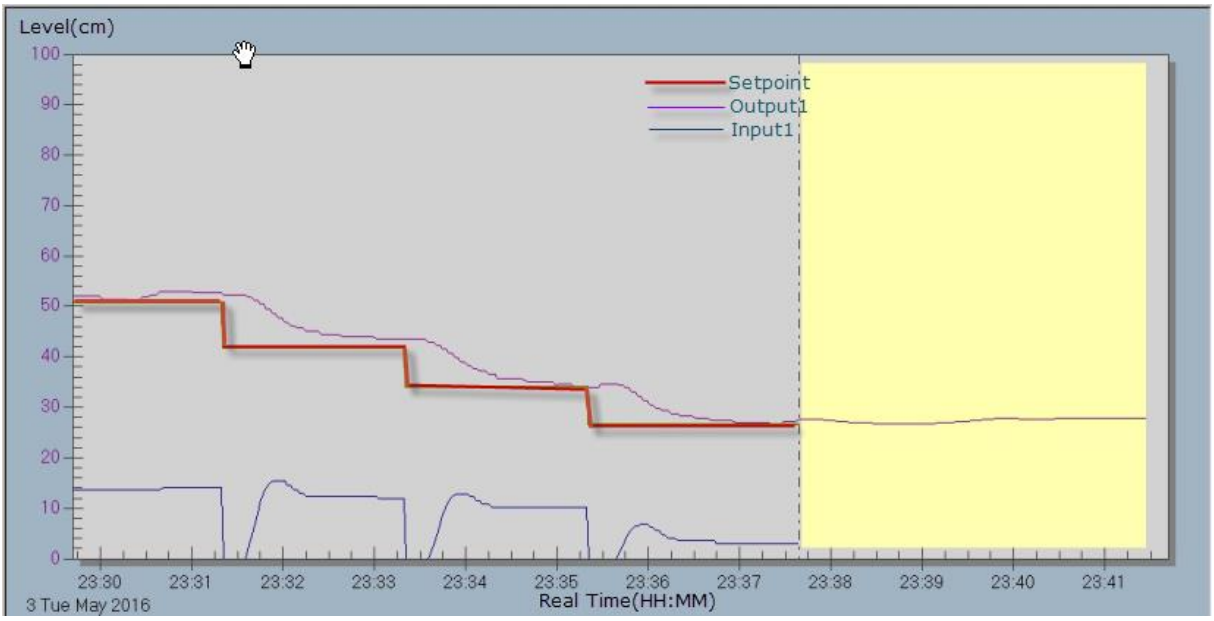


Figure 6-5: Simulation for the MPC control for level in Tank 1.

6.2 DSR

The plot of the real and predicted model of the process is as shown in Figure 6-6. It is seen that the predicted model is very correctly represented. The script for the implementation in MATLAB is shown in Annex 3.

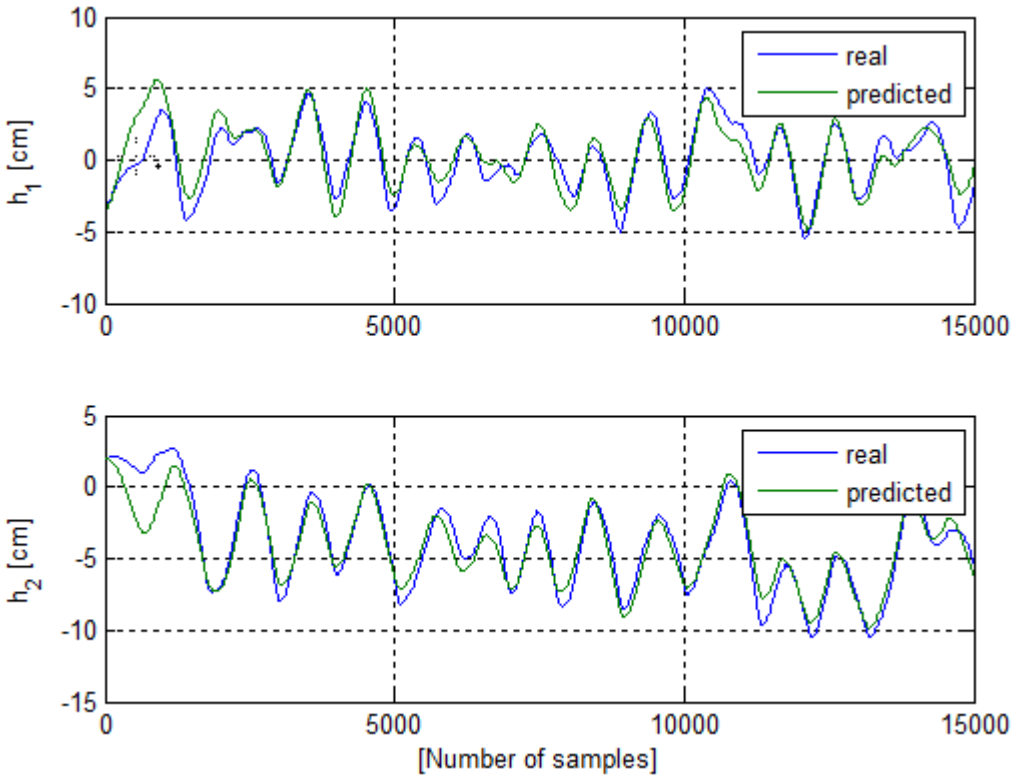


Figure 6-6:DSR identification results for real and predicted levels in Tanks 1 and 2.

6.2.1 Model Performance Indices

The model performance index between DSR and DeltaV is as shown in Table 6-2:.

Table 6-2: MSE from DSR method

Indices	Tank 1	Tank2
MSE (DSR)	0.1825	0.2195
MSE(DeltaV)	3.3833	7.5966

6.3 MPC from DSR Identified Model

Due to the poor results from the control achieved in the DeltaV MPC, an implementation of the model free MPC with integral action in MATLAB was developed to control the QTP. A new set of data was simulated from first principle, for this purpose.

The script for the implementation of the MPC in MATLAB is shown in Annex 2 and Annex 3.

The controller is as shown in Figure 6-7, Figure 6-8, *Figure 6-9: MPC Output and Reference for Level in Tank 1*Figure 6-9 , Figure 6-10. The matrices Q and R are varied to view how these weighting matrices affect the performance of the controller.

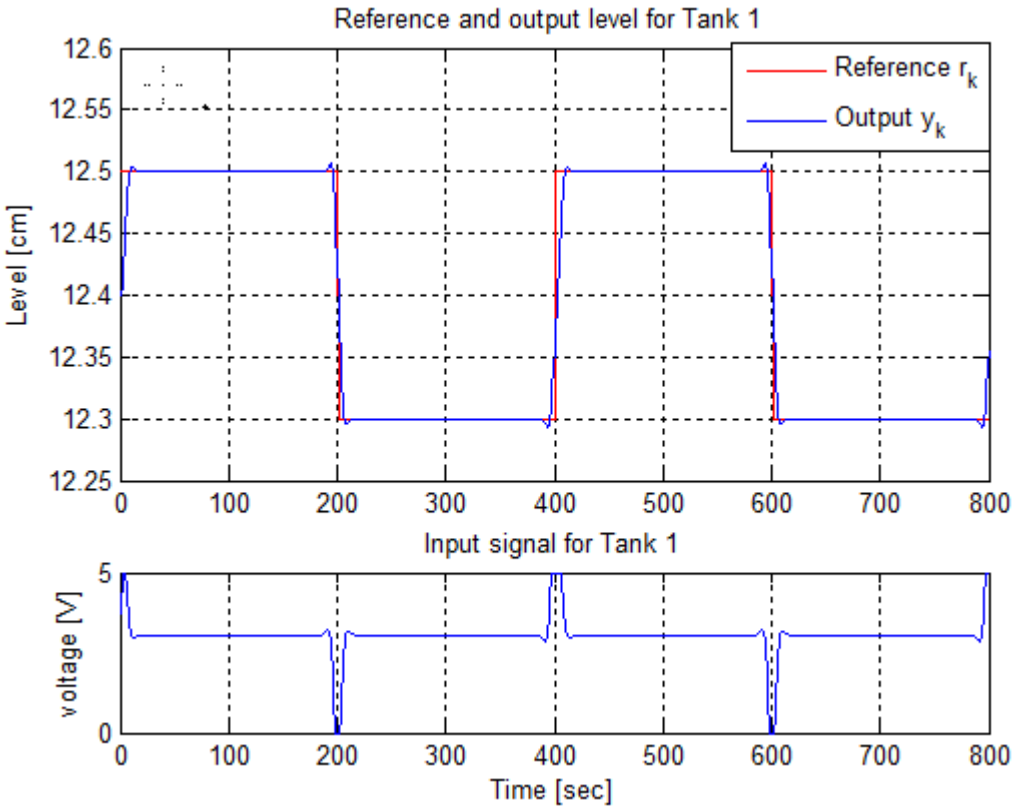


Figure 6-7: *MPC Output and Reference for Level in Tank 1. With matrices* $Q = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$ *and* $R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ *and prediction horizon of 15secs and a sampling time of 0.1secs. The input constraints are* $Umin = 0V$ *and* $Umax = 5V$

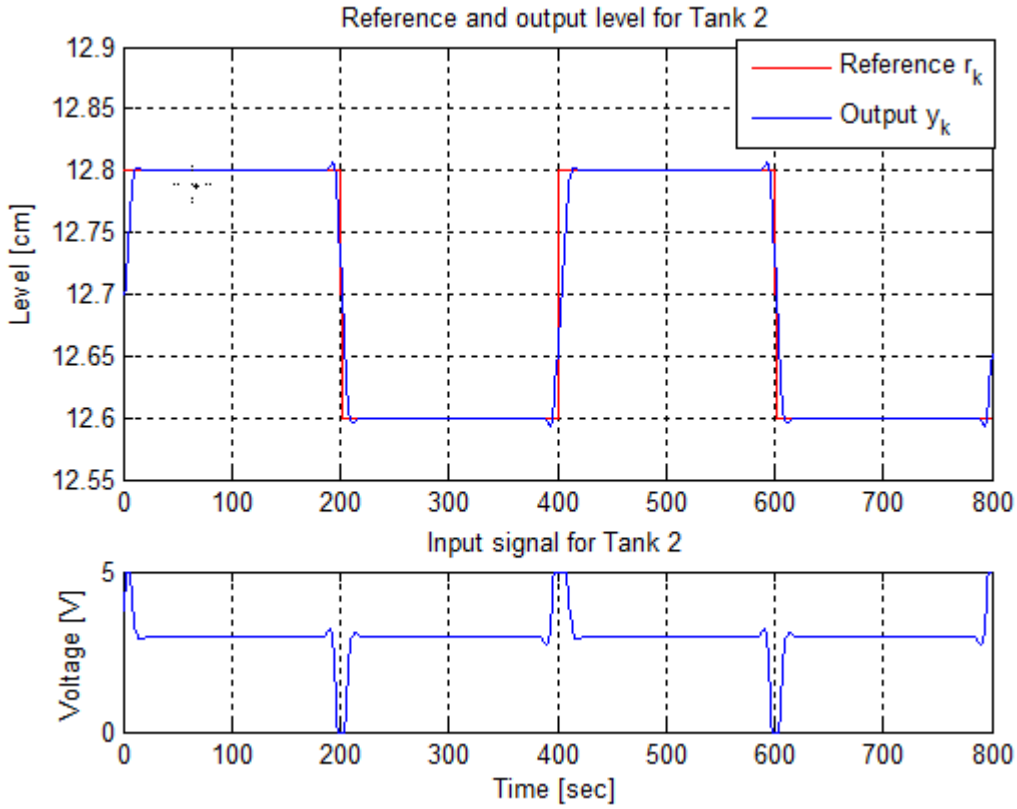


Figure 6-8: MPC Output and Reference for Level in Tank 2. With matrices $Q = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$ *and* $R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ *and prediction horizon of 15secs and a sampling time of 0.1secs. The input constraints are* $Umin = 0V$ *and* $Umax = 5V$

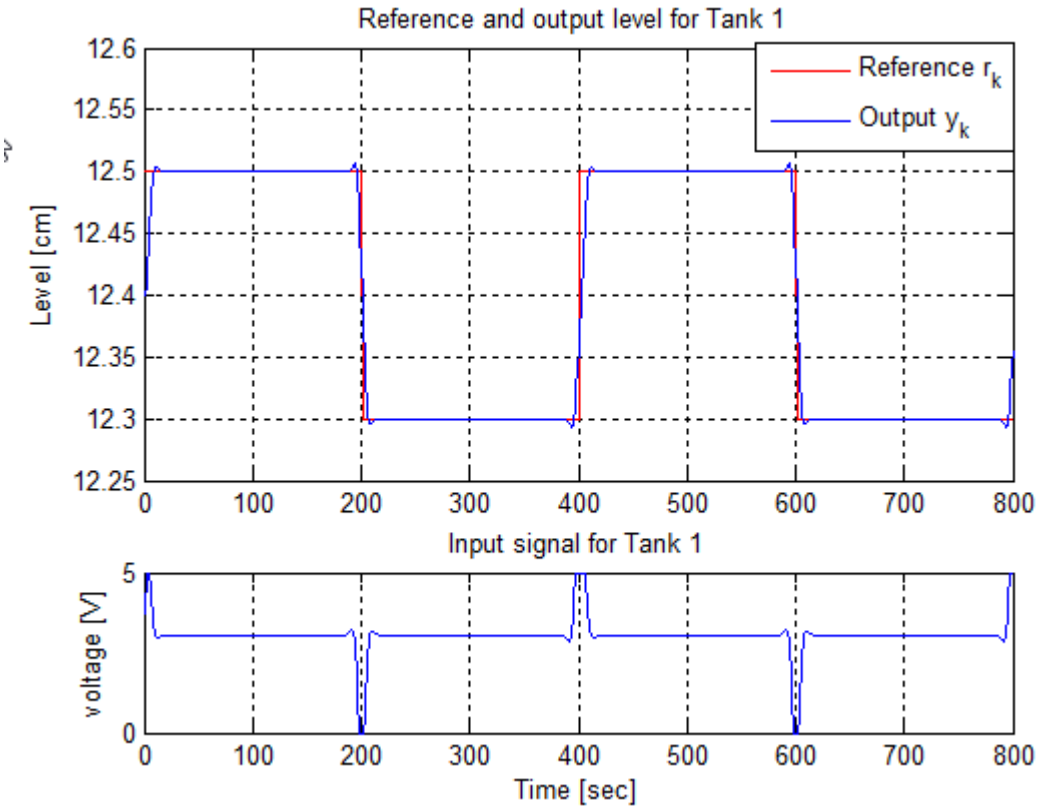


Figure 6-9: MPC Output and Reference for Level in Tank 1. With matrices $Q = \begin{bmatrix} 10000 & 0 \\ 0 & 10000 \end{bmatrix}$ *and* $R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ *and prediction horizon of 15secs and a sampling time of 0.1secs. The input constraints are* $Umin = 0V$ *and* $Umax = 5V$

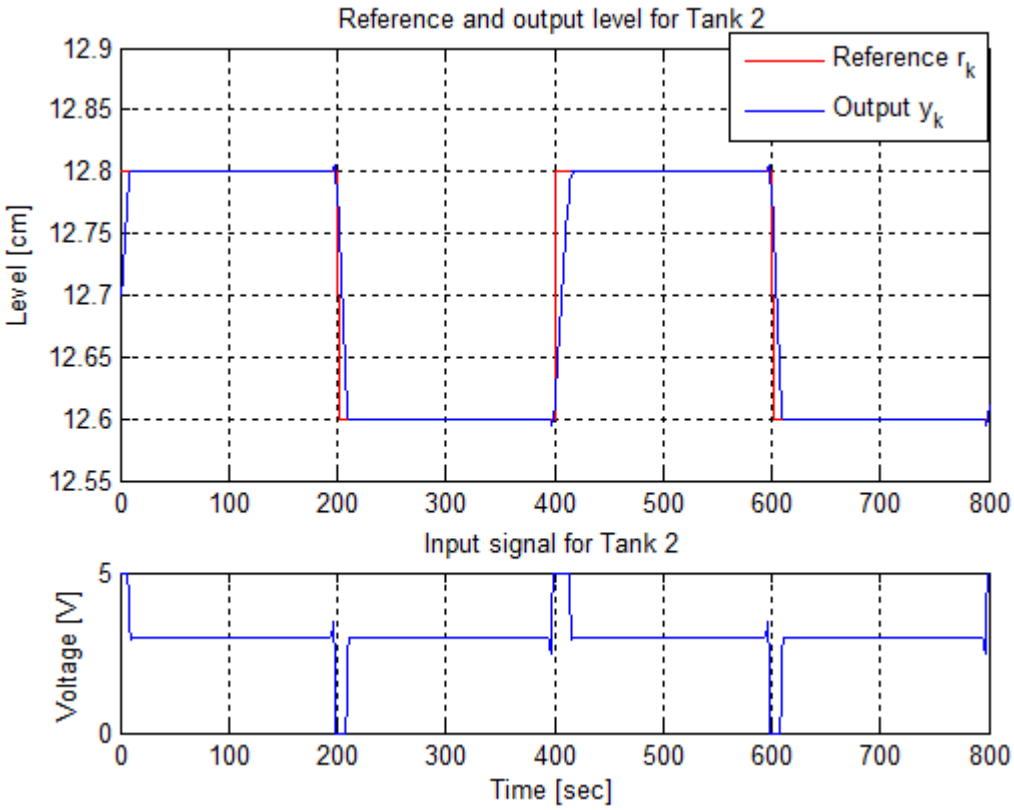


Figure 6-10: *MPC Output and Reference for Level in Tank 2. With matrices* $Q = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$ *and* $R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ *and prediction horizon of 15secs and a sampling time of 0.1secs. The input constraints are* $Umin = 0V$ *and* $Umax = 5V$

6.4 SISO system

DeltaV MPC didn't give good model representation of the QTP, there arose the need to compare the model identification of DeltaV Predict on a SISO system with a subspace system method (DSR). The 2 tank system was analyzed for this purpose.

The 2 tank is process that is made up of 2 tanks, a reservoir for water storage, a pump controlled by a PID controller for filling the tank, two level transmitters that helps to measure the level of the two tanks, two solenoid valves and four level switches. The control objective in this process is to control the level of upper tank with the use of the pump. The input to the 2-tank system is signal to the pump in volts and the output is the level in the lower tank.

The laboratory 2-tank process is shown in Figure 6-11.

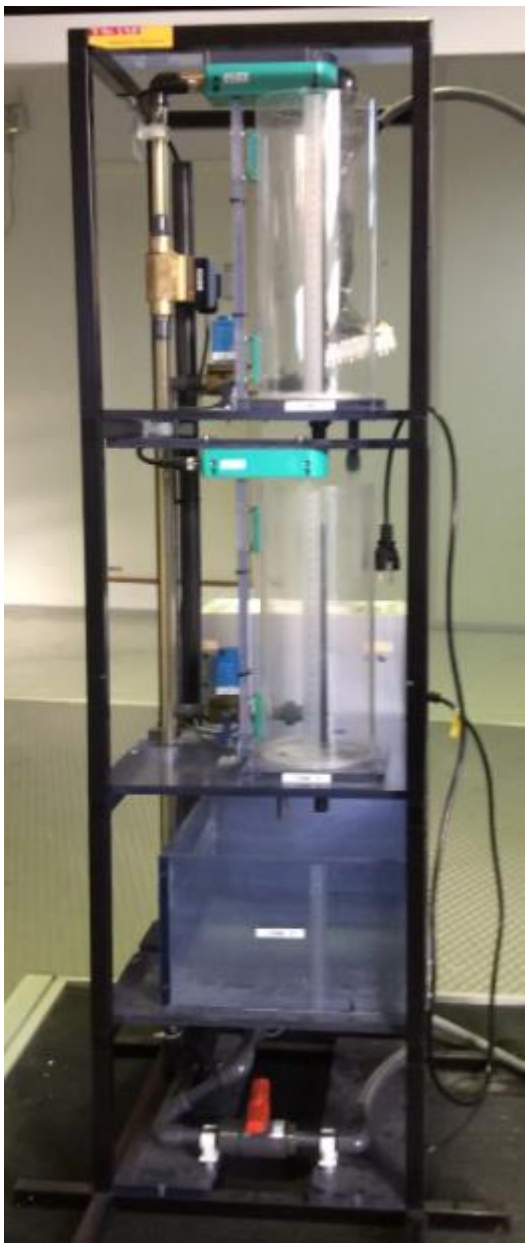


Figure 6-11: 2-tank Process

6.4.1 Model Creation in DeltaV of the 2-tank

The physical 2-tank plant is connected to DeltaV, as presented by Rune Andersen [21]. The system overview of the tank when connected to DeltaV is depicted in Figure 6-12. After connection, DeltaV logs the input and output from the 2-tank and stores in the Custodian Historian and uses this logged data to create a step response model of the 2-tank process.

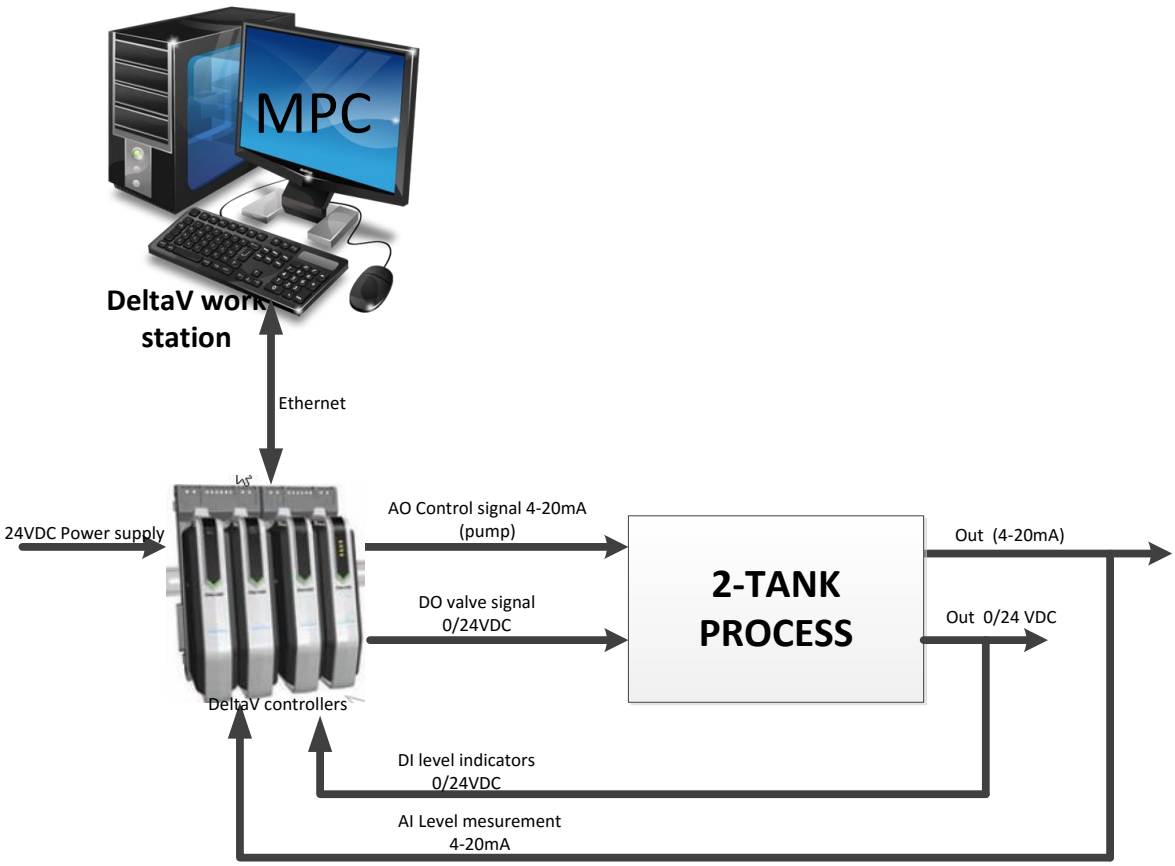


Figure 6-12: 2-tank Control using MPC in DeltaV [20]

The verification results are as shown in Figure 6-13 showing plots of the actual and predicted model.

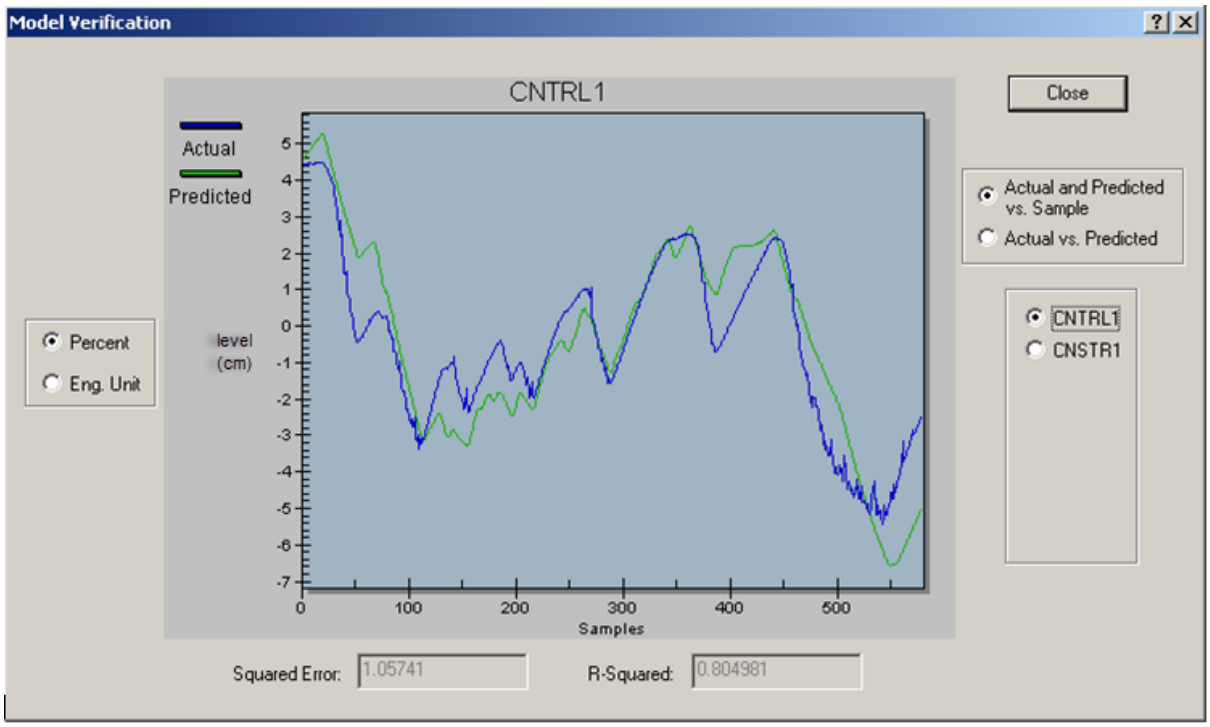


Figure 6-13: Model verification using DeltaV Predict of the 2-tank system showing actual and predicted models

6.4.2 Controller Generation for the 2-tank process

After a satisfactory model has been created, DeltaV automatically generates a controller for the process. DeltaV provides a simulation environment to test the controller first, and if it is working correctly, then it is then tested on the real plant.

Very good set point tracking is achieved as seen in Figure 6-14.

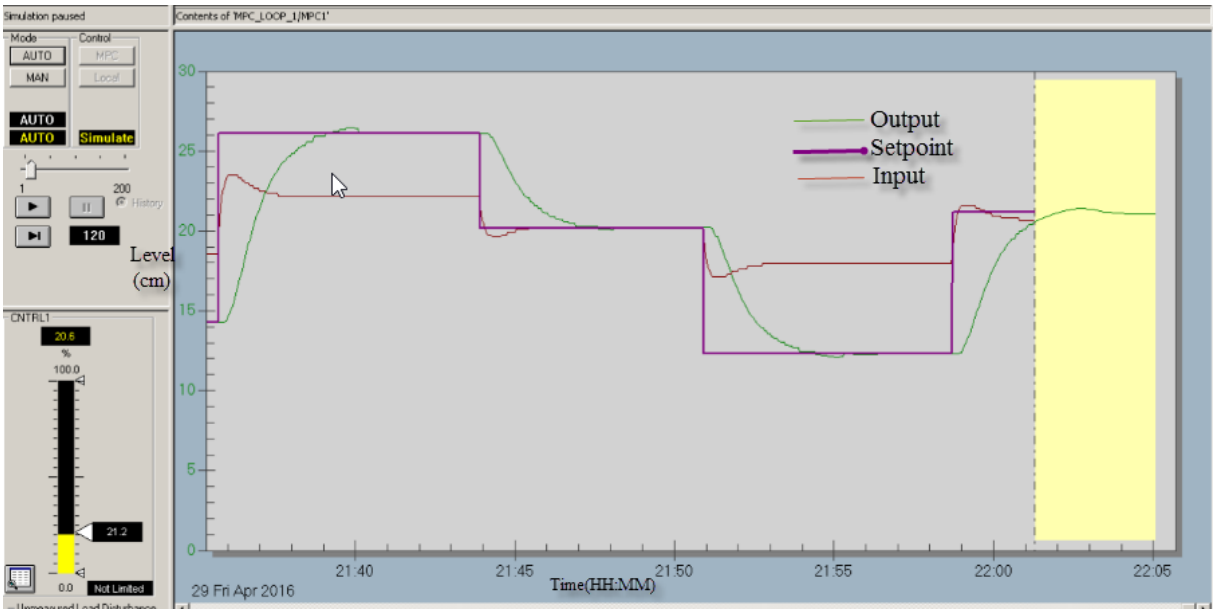


Figure 6-14: MPC controller for the level in the 2tank process.

6.4.3 System Identification in DSR

To compare the model creation in DeltaV Predict with the DSR method of system identification, the data logged in DeltaV is exported to MATLAB for use in the DSR toolbox. A simple plot of the input and output is shown in Figure 6-15. The script for the implementation in MATLAB is as shown in Annex 4.

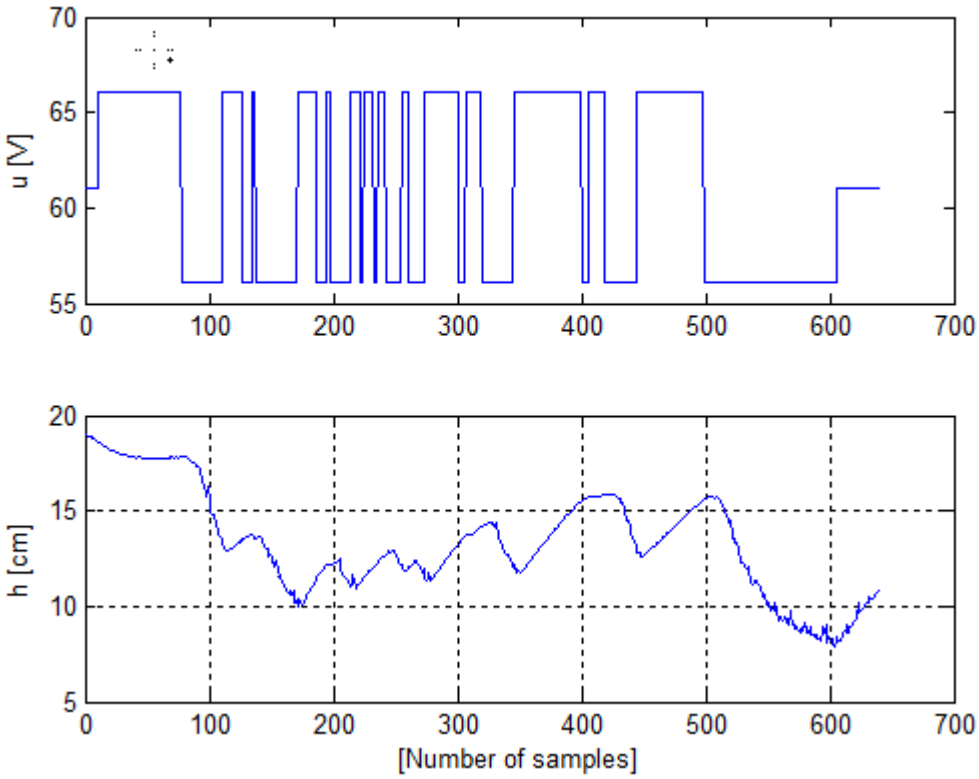


Figure 6-15: Input and output data for the 2-tank

The predicted model in DSR is shown in Figure 6-16.

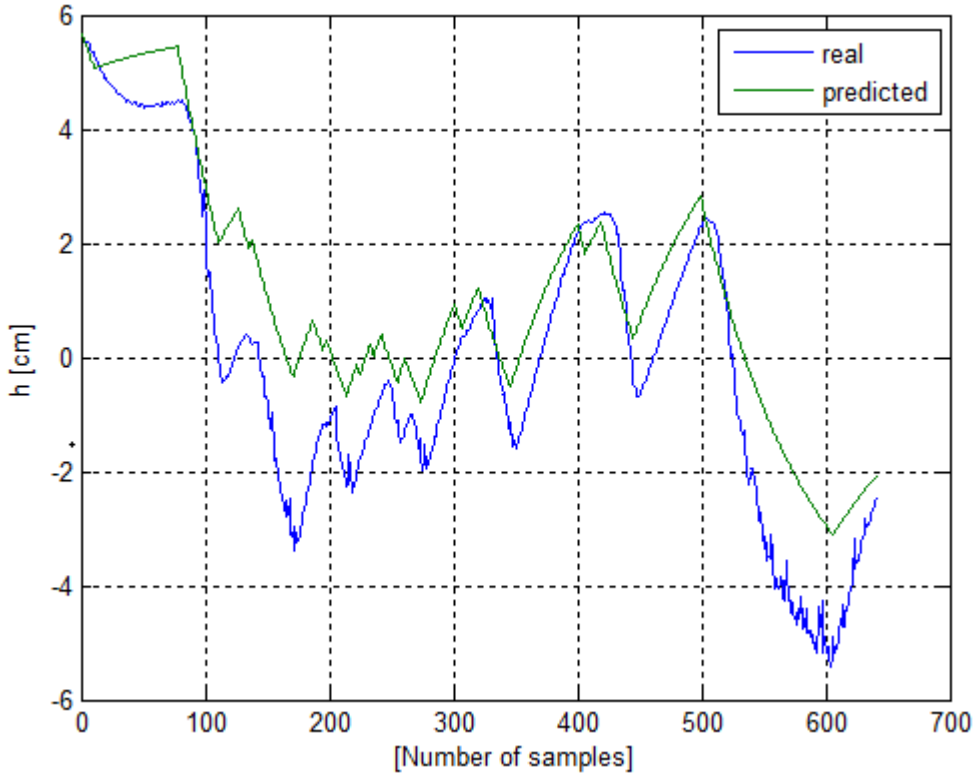


Figure 6-16: Model development using DSR method of the 2-tank system

6.4.4 Model Comparison

To compare the predicted models the MSE indices was used to compare both the DeltaV Predict and the DSR method for the 2-tank system. Table 6-3 shows the results from the comparison. DSR has a better MSE value.

Table 6-3: Comparison of model performance indices for the 2-tank process

Method	MSE
DSR	0.0064
DeltaV Predict	1.05741

7 Discussion of Results

DeltaV model prediction is based on step responses of the input-output model of the process. The magnitude of the step change is very important in step response models. If the step input change is too small, the measured output may not change enough to develop a correct model, alternatively if the step input change is too large, the change in output may be too large and non-linear effects may set in.

The step input change of the data used for the simulations in DeltaV was too small to properly capture the step response dynamics of the process and so the models representation of the QTP was not accurately captured. This problem should have been better handled if DeltaV had been connected to the real process.

To make a step response change in the input variable, the process must be brought to a consistent and steady state change operating point. When the real plant is used for testing in DeltaV, the logging of values is integrated such that the input to the process is logged to ensure that a step response is realized for each input. The pseudo random way DeltaV takes input is calculated so as to give better results. During testing on a physical plant, data that do not represent normal working condition can be excluded before the creation of the step response model creation. This ensures that an optimal model of the process is realized.

The importance of getting a good model of a process cannot be overemphasized as seen in the controller generation in DeltaV. The default settings showed very poor tracking, adjustments (tuning) had to be done to force the tracking of different reference values. This is seen from the very sharp changes in the manipulated variables in Figure 6-4 and Figure 6-5.

On the other hand the DSR method handled the modeling very accurately. The DSR method involves identifying state space matrices from known input-output data of a process. The QTP model creation using the DSR system identification method showed very good results. The MSE from the DSR method was much lower than that achieved using DeltaV. The identified matrices (state space) from the DSR method were used to implement an MPC with integral action and very good set point tracking was achieved.

To further compare the MPC strategy in DeltaV, the 2tank process (a SISO system), was analyzed for this purpose. When DeltaV was connected to real plant of the 2-tank system, much better results were achieved. The input and output log of the data from the 2tank when connected to DeltaV clearly shows that the logging done in DeltaV is random (see Figure 6-15).

DeltaV handled the SISO system better than it did on the MIMO quadruple tank process. Comparing the MSE between models in DeltaV and DSR, even if the DSR was better for both the MIMO and SISO systems, the result from the SISO system was much more comparable. But overall, the DSR in both the MIMO and SISO systems showed better model predictions.

The DSR handled both the MIMO and SISO systems very well. The model performance indices were very low when compared with the models from DeltaV MPC.

8 Conclusion and Recommendation

Conclusion on the results achieved in this work is presented here, with recommendations for further work.

8.1 Conclusion

The non-linear model of the QTP was developed using simple mass balance equations. The ability of DeltaV MPC in controlling the QTP using external historical data was investigated. Very poor step response models were achieved. The controller was tuned to successfully track the setpoint.

DSR method was also used to identify models from the same input-output data of the QTP and a well fitted model was realized. The MSE showed lower values and hence a better model.

DeltaV MPC was used to compare between prediction methods in DeltaV to that in DSR system identification methods. The DSR method produced better and more accurate representative models for both the QTP (MIMO) and the 2-tank process (SISO).

Using MPC with integral action, the quadruple tank was controlled and very robust control was achieved for the levels in the two tanks when compared to that achieved with DeltaV MPC.

In summary, step response models are limited to some MIMO processes. On the other hand, state space model representations are easier, faster, shows better results and more versatile for different processes. DeltaV handles SISO systems better than MIMO systems. DSR shows optimal results for both MIMO and SISO systems.

All the tasks as described in the task description have been successfully completed. However creating a communication interface between DeltaV and the QTP was not achieved because the QTP was undergoing refurbishment for a large part of the period of this study and as such, external data had to be used.

8.2 Future Work

A new QTP should be made available so that the data can be logged directly when connected to DeltaV. The communication should be made available from the Volts to 4-20mA so that

connection to DeltaV can be possible. The data logged in DeltaV can be imported to MATLAB for comparison with other system identification methods; the Prediction error method (PEM), N4SID methods can be analyzed for this purpose.

References

[1] B. Halvarsson, "Interaction Analysis in Multivariable Control Systems," *PhD, Uppsala University,* 2010.

[2] K. H. Johansson, A. Horch, O. Wijk, and A. Hansson, "Teaching multivariable control using the quadruple-tank process," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, 1999, pp. 807-812.

[3] Q. Saeed, V. Uddin, and R. Katebi, "Multivariable predictive PID control for quadruple tank," *World Academy of Science, Engineering and Technology,* pp. 861-866, 2010.

[4] M. Mohsin, "Model Predictive control (MPC) with integral action; Reducing the control horizon and model free MPC.," Masters', Telemark University College, 2013.

[5] D. Di Ruscio, "Model predictive control with Integral action: a simple MPC algorithm," 2013.

[6] D. G. Danushka, "Experimental Subspace Identification and Model Predictive Control of a Four Tank System," *Telemark University College,* 2012.

[7] E. P. Gatzke, E. S. Meadows, C. Wang, and F. J. Doyle, "Model based control of a four-tank system," *Computers & Chemical Engineering,* vol. 24, pp. 1503-1509, 2000.

[8] I. Drca, "Nonlinear model predictive control of the four tank process," 2007.

[9] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design* vol. 2: Wiley New York, 2007.

[10] D. D. Ruscio, "Subspace System Identification: Theory and Applications," *Lecture notes, Telemark University College, Porsgrunn, Norway,,* 2014. .

[11] K. H. Johansson, "The quadruple-tank process: a multivariable laboratory process with an adjustable zero," *Control Systems Technology, IEEE Transactions on,* vol. 8, pp. 456-465, 2000.

[12] K. H. Johansson and J. L. R. Nunes, "A multivariable laboratory process with an adjustable zero," in *American Control Conference, 1998. Proceedings of the 1998*, 1998, pp. 2045-2049.

[13] Carlos, "Modeling, Simulation and Control for an Experimental Four Tanks System using ScicosLab," 2011.

[14] D. Di Ruscio, "Combined Deterministic and Stochastic System Identification and Realization: DSR-a subspace approach based on observations," *Modeling, Identification and Control,* vol. 17, p. 193, 1996.

[15] R. E. Kalman, "Contributions to the theory of optimal control," *Bol. Soc. Mat. Mexicana,* vol. 5, pp. 102-119, 1960.

[16] N. R. Ruchika, "Model predictive control: History and development," ed: IJETT, 2013.

[17] D. Di Ruscio, "Model predictive control and optimization," *Telemark University College,* 2001.

[18] B. Kim, G. N. Washington, and H.-S. Yoon, "Hysteresis-reduced dynamic displacement control of piezoceramic stack actuators using model predictive sliding mode control," *Smart Materials and Structures,* vol. 21, p. 055018, 2012.

[19] D. D. Ruscio, "Model based predictive control: an extended state space approach," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, 1997, pp. 3210-3217.

[20] Wikipedia. (April, 2016). *Mean squared error*. Available: https://en.wikipedia.org/wiki/Mean_squared_error

[21] R. Andersen, "Two-Tank control with DeltaV using MPC," *Telemark University College*, 2014.

Annexes

Annex 1: Project Task Description

Annex 2: MATLAB implementation of MPC with Integral action

Annex 3: MATLAB implementation of system identification for quadruple tank process

Annex 4: MATLAB implementation of system identification for 2 tank process

Annex 1: Project Task description

FMH606 Master's Thesis

Title: Model Predictive Control using DeltaV of the Quadruple Tank Process

TUC supervisor: David Di Ruscio

External partner: Rune Andersen. Emerson.

Task description:

1. Give a short overview of existing models for the Quadruple Tank (QT) process, both system identification based (e.g. as DSR) and first principles based models.
2. It is of interest to fit some of the models to observed input and output process data. Perform an input experiment on the QT-process and collect input and output data. Use and compare the models upon each other. The MSE and IAE indices may be used for comparison
3. Investigate the ability to use the DeltaV control system and in particular the built in MPC strategy to control the QT-process.
4. Build a real time control and communication platform between the DeltaV system and the QT-process. Perform step changes in the level references and illustrate the possible efficiency of the real time control system.

Task background:
The QT-process is recently under reconstruction and rebuilding at the TUC laboratory. It would be of great interest to analyse the process for use with the DeltaV control system, and in particular the built in MPC strategy. Both the real laboratory process and numerical simulations with MATLAB may be used as parts of the work in this thesis.

Student category:
SCE students.

Practical arrangements: This work is to be done at TUC and at the laboratory.

Signatures:

Student (date and signature):

Supervisor (date and signature):

Annex 2

```
%==========================================================================
%MatLab script for Implemetation of MPC with Integral action for the QTP
% eobsv.m, prbs1.m, q2qt.m, scmat.m, ss2h.m functions all written by [David
DiRuscio]
% Reference http://www2.hit.no/tf/fag/sce4106/ovinger/ovinger.html
% -------------------------------------------------------------------------
%HSN, Porsgrunn, Norway [29-04-2016]
%==========================================================================
clear all;
% % Initial level in the tanks [cm]
h1=12.4;
h2=12.7;
h3=1.8;
h4=1.4;
h=[h1,h2,h3,h4]';

dt = 1;
time = 0: dt: 2500;
N = length (time);
um1= 3; um2=3.2;

% Space for storing variables
Y=zeros(N,4);
U=zeros(N,2);
u=zeros(N,2);

%input using the pbrs function by Di Ruscio
U=[3*ones(N,1)+0.1*prbs1(N,30,150) 3*ones(N,1)+0.1*prbs1(N,30,150)];

%Discretising using Euler
for k=1:N
    u=U(k,:)'
    dhdt=TankSimData(time,h,u);
    Y(k,:)=h';
    h=h+dt*dhdt;
end
u;

Nid=N;
Yid=Y(1:Nid,:)
    Uid=U(1:Nid,:)

%    Trending the data

Uid=[U(1:Nid,1)-3.0 U(1:Nid,2)-3.0 ];
Yid=[Y(1:Nid,1)-12.3 Y(1:Nid,2)-12.8];

[A,B,D,E,C,F,x0]=dsr(Yid,Uid,4,0);

Ys=dsrsim(A,B,D,E,Uid,x0);

l=size(A,1);
m=size(B,2);
n=size(D,1);
% Matrics At, Bt and Dt
At=[A zeros(l,n); D eye(n,n)];
Bt=[B;zeros(n,m)];
Dt=[D eye(n,n)];
% Model predictive control algorithm,
```

```
% Prediction horizon
Np=15;
% Simulation horizon
N=615;
% Weighting matrices
Q=100*eye(2); R=0.1*eye(2);
% Observability matrix OL and Toeplitz matrix HdL calculation by ss2h
function.[David Di Ruscio]
[HdL,OL,OLB]=ss2h(At,Bt,Dt,zeros(n,m),Np,0);
FL=[OLB HdL];
Qt=q2qt(Q,Np);
Rt=q2qt(R,Np);
% Make matrices S and c in the relationship, u(k,L) %scmat function
[S,c] = scmat(m,Np);
% H as defined in equation (3.50)
H=FL'*Qt*FL+Rt;
% Defining inital values for simulation
I_val=[11.4;12.5];
Hd=D*inv(eye(4)-A)*B;
u_ss=inv(Hd)*I_val;
I=eye(4);
x_ss =inv(I-A)*B*u_ss;
% Assigning variables
u=u_ss;
h=x_ss;
u_old=u; h_old=h; y_old=D*h;
% Make the references by using the function prbs1.m[David Di Ruscio]
rand('seed',0),randn('seed',0);
ref=[11.4*ones(N,1)+0.1*prbs1(N,100,100),12.5*ones(N,1)+0.1*prbs1(N,100,100
)];
% ref=[14*ones(N,1)+0.1*prbs1(N,70,70) 12*ones(N,1)+0.1*prbs1(N,70,70)];
% Used in Input amplitude constraints implementation.
u_min=0;
u_max=5;
% Making for variables storage
r1L=zeros(15,1);
U=zeros(N-Np,2);
Y=zeros(N-Np,2);
for k=1:N-Np
y=D*h; % output equation
% Make the extended reference vector, r_(k,L) .[David Di Ruscio]
rf =ref(k+1:k+Np,:);
r1L=rf(1,:)';
for i=2:Np
r1L=[r1L;rf(i,:)'];
end
% Computing MPC control
xk=[h-h_old;y_old];
pL=OL*At*xk;
% f as defined in equation (3.50)
f=FL'*Qt*(pL-r1L);
% For constrained MPC [Equation 2.8]
a=[S;-S];
b=[u_max*ones(Np*m,1)-c*u_old;-u_min*ones(Np*m,1)+c*u_old];
% quadprog function for input amplitude constraints. [Equation 2.29]
duf=quadprog(H,f,a,b);
u=u+duf(1:m);
u_old=u;
% For plotting purpose store the variables.
U(k,:)=u';
Y(k,:)=y';
% Feed control to process.
```

```
h_old=h;
y_old=y;
h=A*h+B*u;
end
% Plotted Results
t=1:N-Np;
% Level results for Tank 1.
figure(1)
subplot(3,1,[1,2])
plot(t, ref(1:N-Np,1),'r');hold on
plot(t,Y(:,1))
ylabel('Level [cm]');
title('Reference and output level for Tank 1')
legend('Reference r_k','Output y_k')
grid on
% Input Control signal for Tank 1
subplot(3,1,3),
plot(U(:,1)),
xlabel('Time [s]');
ylabel('Voltage [V]')
title('Input signal for Tank 1')
grid on
% Level results for Tank 2.
figure(2)
subplot(3,1,[1,2])
plot(t, ref(1:N-Np,2),'r');hold on
plot (t,Y(:,2))
ylabel('Level [cm]');
title('Reference and output level for Tank 2')
legend('Reference r_k','Output y_k')
grid on
% Input Control signal for Tank 2
subplot(3,1,3),
plot(U(:,2)),
xlabel('Time [s]');
ylabel('Voltage [V]')
title('Input signal for Tank 2')
grid on
```

Annex 3

```
%==========================================================================
% DSR system Identification of quadruple tank process
%HSN, Porsgrunn, Norway [29-04-2016]
% dsr function written by [David DiRuscio]
%==========================================================================
load sys_data_09_03_2014.lvm
sys_data_09_03_2014=sys_data_09_03_2014(10:end,:);

N=size(sys_data_09_03_2014,1)
Is=1;
Y=[sys_data_09_03_2014(Is:N,2) sys_data_09_03_2014(Is:N,3)];
U=[sys_data_09_03_2014(Is:N,5) sys_data_09_03_2014(Is:N,4)];

save Y_4tank.dat -ascii Y
save U_4tank.dat -ascii U

ym=mean(Y)
um=mean(U)

Nid=15000;

itrend=2;
if itrend==1
    Yid_RD=Y(1:Nid,:)
    Uid=U(1:Nid,:)
else
    Yid_RD=[Y(1:Nid,1)-11.83 Y(1:Nid,2)-12.38];
    Uid=[U(1:Nid,1)-3.8 U(1:Nid,2)-3.4];

end

[A,B,D,E,C,F,x0]=dsr(Yid_RD,Uid,4,0);

Ys_RD=dsrsim(A,B,D,E,Uid,x0);

figure(1)
subplot(211), plot(U(:,1))
ylabel('u_1 [V]')
subplot(212), plot(U(:,2))
ylabel('u_2 [V]')
xlabel('Number of samples')

figure(2)
subplot(211), plot(Y(:,1))
ylabel('h_1 [cm]')
xlabel('[Number of samples]')
grid on
subplot(212), plot(Y(:,2))
ylabel('h_2 [cm]')
xlabel('[Number of samples]')
grid on

figure(3)
subplot(211), plot([Yid_RD(:,1) Ys_RD(:,1)])
legend ('real', 'predicted');
ylabel('h_1 [cm]')
grid on
subplot(212), plot([Yid_RD(:,2) Ys_RD(:,2)])
```

```
legend ('real', 'predicted');
ylabel('h_2 [cm]')
xlabel('[Number of samples]')
grid on

%Model performance indices calculations
err_h1=(Ys_RD(:,1)-Yid_RD(:,1));
err_h2=(Ys_RD(:,2)-Yid_RD(:,2));
IAE_dsr_h1= (sum(abs(err_h1)))
IAE_dsr_h2= (sum(abs(err_h2)))
MAE_dsr_h1= 1/Nid*(sum(abs(err_h1)))
MAE_dsr_h2= 1/Nid*(sum(abs(err_h2)))
ISE_dsr_h1= ((sum(err_h1).^2))
ISE_dsr_h2= ((sum(err_h2).^2))
RMSE_dsr_h1= 1/Nid*sqrt((sum(err_h1).^2))
RMSE_dsr_h2= 1/Nid*sqrt((sum(err_h2).^2))
```

Annex 4

```
%=========================================================================
% Identification of 2 tank process
% dsr function written by [David DiRuscio]
% -------------------------------------------------------------------------
%HSN, Porsgrunn, Norway [29-04-2016]
%=========================================================================
Load data_2_tank.csv;
N=641;
Y=[u(Is:N,2)];
U=[u(Is:N,4)];
% Identification of 2 tank process
ym=mean(Y)
um=mean(U)
Yid=[Y(1:N,1)-ym]; %trending data
Uid=[U(1:N,1)-um];  %trending data
[A,B,D,E,C,F,x0]=dsr(Yid,Uid,4,0);
Ys_RD=dsrsim(A,B,D,E,Uid,x0);
figure(1)
subplot(211), plot(U(:,1))
ylabel('u [V]')
subplot(212), plot(Y(:,1))
ylabel('h [cm]')
xlabel('[Number of samples]')
gridon

figure(2)
plot([Yid(:,1) Ys_RD(:,1)])
legend ('real', 'predicted');
ylabel('h [cm]')
xlabel('[Number of samples]')
gridon

% Model Performance calculations
err_h=(Ys_RD(:,1)-Yid(:,1));
IAE_dsr_h= (sum(abs(err_h1)))
MAE_dsr_h= 1/Nid*(sum(abs(err_h1)))
ISE_dsr_h= ((sum(err_h1).^2))
RMSE_dsr_h= 1/Nid*sqrt((sum(err_h1).^2))
```