Design and Implementation of Fractional Order PID Controllers for Process Control Systems

Emmanuel Edet

A thesis submitted for the degree of Doctor of Philosophy

Department of Electronic and Electrical Engineering

University of Strathclyde

2018

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: Date:

Acknowledgements

First of all, I would like to thank my supervisor Dr. Reza Katebi, who has been an excellent mentor to me throughout the whole period of my PhD studies. In moment of sluggish progress, he was still patient with me. He has always been available to answer my questions and give me advice. I would like also to thank Dr. Sung-ho Hur for the useful conversations on how to approach problems concerning my research.

There are not enough words to thank my family and parents, to whom I owe a lifetime of gratitude, for their continuous encouragement. A big thank you to my wife, Juliet for your unflinching support. Your support gives me the confidence to take on daunting tasks. Many thanks to Shemen for being a source of strength rather than distraction during my final year. Many thanks, too, to John who offered a much-needed constructive criticism of my thesis. Finally, I am grateful to all my friends at the TIC building in university of Strathclyde for all the mutual encouragement we shared.

Abstract

New methods of designing Fractional Order PID (FOPID) controllers are developed for regulating product's purity in distillation columns. It is widely known that poorly tuned PID controllers lead to bad quality of products and accompanied reduction in profit margin in the process industry. In distillation columns, tight composition control of products with 98% purity level is not achievable with simple pressure controllers only due to sensitivity to disturbances. Therefore, FOPID controllers are proposed to solve these multivariable impurity problems. FOPID controllers have extra tuning parameters that can counteract effects of time delays in distillation columns if properly tuned. This property is exploited as a tool for improved performance. Original contributions of this thesis include the development of three new design methods for multivariable FOPID controllers and results are analysed using inverse maximum singular value of relevant sensitivity functions to assess robust stability. Several conventional PID controller design methods are also reviewed for the purpose of comparison.

Thereafter, a decentralised FOPID control system is developed for multivariable systems based on plant's critical frequency point and results show improved performance over conventional PID controllers. The proposed critical frequency point method provides very easy-to-use tuning rules similar to Cohen-Coon tuning rule for integer-order PID controllers. In addition, a new decentralised multivariable FOPID controller is also proposed based on Internal Model Control (IMC) method but settings are tuned using Biggest Log-magnitude Technique (BLT). This IMC-FOPID control design scheme overcomes the need for critical frequency point experiments. Another contribution of this thesis is the development of a novel discrete Fractional Order Predictive PI (FOPPI) control design scheme suitable for linear multivariable systems. Comparative study of these methods is presented. Simulation results prove that both top and bottoms products' purity of 98% are achievable with improved disturbance rejection when using the proposed FOPPI controller. In comparison, simpler FOPID control design schemes developed in continuous time are found to meet design objectives but at the expense of having a more conservative control action.

Table of Contents

List of Figures

List of Tables

Nomenclature

A,B,C,D	Augmented state space model (augmented coefficient matrices)
A_m	Gain Margin
α	Relative Volatility
A_p,B_p,C_p,D_p	State space model (coefficient matrices)
B_{Nn}	Bottoms flow rate at bottom(Nnth) tray
β	Centralised controller tuning parameter for proportional term
$C_{ij}(s)$	Centralised controller transfer function matrix
C_j	Decentralised controller transfer function matrix
D_1	Distillate flow rate at top tray
d_i	Disturbance in the ith loop
Δ_I, Δ_O	Input uncertainty, Output uncertainty
$\Delta \boldsymbol{u}$	Manipulated input variable (incremental)
ε	Small tuning parameter for centralised controllers
F	Detuning factor for BLT control scheme
f	Feed flow rate
G, G_j, G_p	Prediction matrix coefficients of FOPPI control law
$G_{ij}(s)$	ijth element of a MIMO process transfer function matrix
γ	Constant control signal penalising factor (MPC tuning factor)
i,j	Loop or stage numbering and iteration index
$I_{m\times m}$	Identity matrix with dimension m x m.

J	Cost function
K, K_c	Controller gain matrix (Martin-Katebi), SIMC controller gain
$k_{I,j-imc}$	Integral gain for the *j*th loop's IMC-FOPI controller
k_p, k_I, k_d	Proportional gain, Integral gain, Derivative gain of PID controller
K_p, K_i	Proportional and integral gain matrices for *L*-*T* algorithm
K_P, τ_P, L	Process Gain, time constant and time delay
$k_{p,j-imc}$	Proportional gain for the *j*th loop's IMC-FOPI controller
k_u	Ultimate gain (obtained at critical frequency point)
$L_c^{\max}$	Maximum closed loop log modulus
L_{cm}	Biggest log-modulus
L_f	Reflux flow rate
L_i	Liquid flow rate at *i*th tray
λ	Fractional-order (integral)
λ_o	Linearisation parameter defined in (3.10)
λ_{ij}	*ij*th relative gain array element
m	Number of input or output in square MIMO plant
M_i	Liquid hold-up on *i*th tray
M_p	Maximum complementary sensitivity

M_r	Resonant peak
M_s	Maximum sensitivity
μ	Fractional-order (derivative)
N	Control horizon
N_f	Feed entry plate
Nn	Number of trays in distillation column
ω_{cg}, ω_{cp}	Gain cross over frequency, Phase cross over frequency
ω_n, ω_b	Natural frequency, Operational system's bandwidth
P	Prediction horizon
$\varnothing_m$	Phase margin
q_f	Fraction of liquid in inlet-feed
r_i	Reference input at *i*th loop
$\bar{\sigma}$	Maximum singular value
s	Laplace variable
$S(s), T(s)$	Sensitivity function, Complementary sensitivity function
t, k	Continuous time, Discrete time instant
T_s	Sampling time
τ_f	Adjustable filter time constant (IMC tuning parameter)
τ_I, τ_d	Integral time constant, Derivative time constant of PID controller
$\tau_{I,j-imc}$	Integral time for the *j*th loop's IMC-FOPI controller

τ_u	Ultimate period
V_i	Vapour flow rate at *i*th tray
W	Multivariable Nyquist Array function
x_B	Mole fraction of bottom's light composition
x_D	Mole fraction of distillate's light composition
x_i	Liquid composition of light component on *i*th tray (mole fraction)
y_i	Vapour composition of light component on *i*th tray (mole fract.)
Y_i	System's *i*th output
z	Discrete domain variable

Abbreviations

BLT	Biggest Log-modulus Tuning
BMI	Bilinear Matrix Inequality
CARIMA	Controlled Auto-Regressive and Integrated Moving Average
CDU	Crude Distillation Unit
CFE	Continuous Fraction Expansion
DE	Differential Evolution
DMC	Dynamic Matrix Control
DMPC	Discrete Model-based Predictive Control
DNA	Dynamic Nyquist Array
FLCs	Fuzzy Logic Controllers
FMIGO	Fractional M_s-constrained Integral Gain Optimisation
FOPDT	First Order Plus Dead Time system
FOPI	Fractional Order Proportional-Integral
FOPID	Fractional Order Proportional-Integral-Derivative
FOPPI	Fractional Order Predictive Proportional-Integral
GA	Genetic Algorithms
GPC	Generalised Predictive Control
IAE	Integral Absolute Error
IMC	Internal Model Control
IMSV	Inverse Maximum Singular Value
ISE	Integral Squared Error
LHP	Left Half Plane

L-T	Lieslehto and Tanttu's algorithm
MIGO	M_s-constrained Integral Gain Optimisation
MPC	Model-based Predictive Control
MIMO	Multi-Input Multi-Output
NIR	Near Infra-Red *spectrometer*
NMR	Nuclear Magnetic Resonance analyser
OPI	Optimum Proportional-Integral *controller*
OR	Ogunnaike and Ray *model*
ORA	Oustaloup Recursive Approximation
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PSO	Particle Swarm Optimisation
Pt RTD	Platinum Resistance Temperature Detector
RGA	Relative Gain Array
SISO	Single Input Single Output
SOPDT	Second Order Plus Dead Time
SIMC	Skogestad Internal Model Control
SSV	Structured Singular Value
TITO	Two Input Two Output
VLE	Vapour Liquid Equilibrium
VDU	Vapour Distillation Unit
WB	Wood-Berry *model*

Chapter 1

Introduction

1.1 Control of Single Input Single Output (SISO) and Multivariable Systems

The need for regulation, stability and disturbance attenuation are fundamental motivations for design of control systems for various processes and systems. Systems with one actuating control input and a single output are generally considered to be Single Input Single Output (SISO) systems. On the contrary, systems with more than one actuating input and two or more output variables may be considered to be multivariable systems. Multivariable systems are also termed Multi-Input Multi-Output (MIMO) systems.

Today, the most dominant controller type in industry for both SISO and MIMO systems is still Proportional-Integral-Derivative (PID) controller. Large number of PID controllers can be found in chemical reactors, industrial boiler systems, water treatment plant, separation systems and many other automation systems. Industrial popularity of the PID controller is due to simplicity of design, widespread familiarity in industrial circles as well as its robust performance against sensor and actuator's limitations in practical plants (Astrom & Hagglund, 1995). The control objective of SISO systems (with PID controller) is to maintain desirable behaviour of the output variable by varying a manipulated signal. This is mainly achieved using feedback architecture. A simple diagram of SISO control system is shown in Fig. 1.1 where $G(s)$ represents the scalar process, $C(s)$ is the controller, $U(s)$ is the control signal, $Y(s)$ is the system's output while $R(s)$ represents system's reference input (desired input).

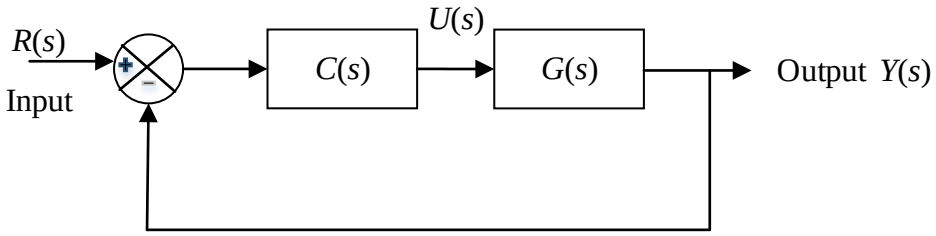


Fig. 1.1 A SISO control system showing controller C(s) and process G(s)

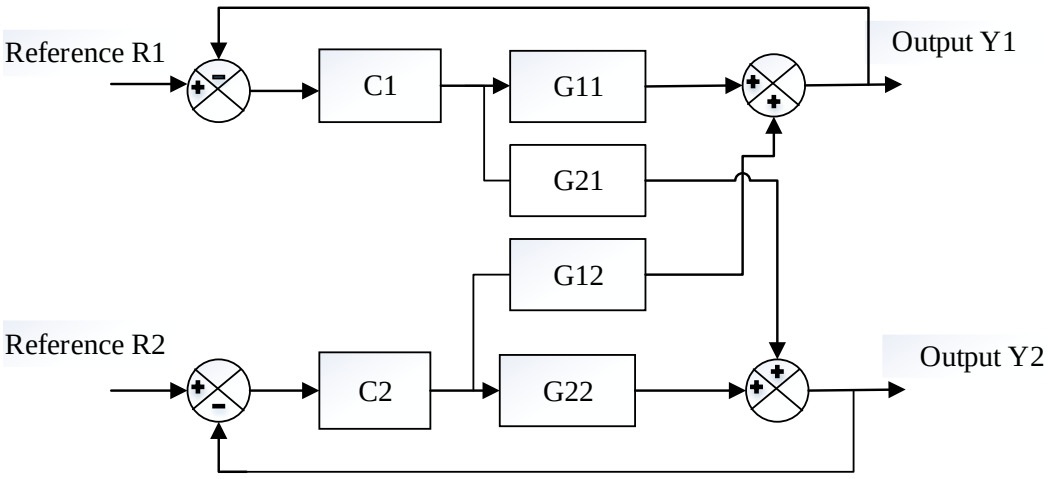


Fig. 1.2 An example of decentralised MIMO control system with two inputs and outputs

Application of feedback principle helps reduce sensitivity to noise and other process variations and also aids tracking of set-point. In many cases, Proportional-Integral (PI) controllers and PID controllers are generally a simple implementation of feedback in a SISO system as shown in *Fig.* 1.1. In a coupled MIMO case, feedback principle can be applied using multi-loop configuration as shown in *Fig*. 1.2 where the process matrix [G11 G12; G21 G22] represents the two-input two-output system; *C*1 and *C*2 represent the two decentralised controllers; *R*1 and *R*2 represent the two input signals while *Y*1 and *Y*2 are the two system's outputs. Over the past six decades, these PID controllers have been found to yield satisfactory control actions for many control problems especially, when the process dynamics are first or second order and performance requirements are not too stringent. PID controllers have been found to eliminate steady state errors through integral action while the derivative component provides anticipatory action and aids damping. Several tuning methods for this class of controllers are already well-established in industry and there is a well-established practice of installing and running these conventional PID controllers (Astrom & Hagglund, 1995). However, in extending most of these established PID controller design methods to multivariable process control, there is a major setback of multivariable interaction which limits the performance of such systems (Johnson & Moradi, 2006).

Multivariable system control has been known over many decades to be more challenging to design when compared to SISO applications. This is primarily due to presence of interaction and directions in the system (Albertos & Sala, 2004). Examples of such systems are distillation systems and industrial boiler systems. Some multivariable systems have the same number of inputs and outputs and are generally termed square systems. A functional transfer function model of MIMO systems is given below in (1.1):

$$G(s) = \left\{ G_{ij}(s) : i = 2,3,...,n;\ j = 2,3,...,m \right\}. \qquad (1.1)$$

Simulation studies in this thesis are limited to square systems with two or three inputs i.e. $i = j$; $\{n, m\} = \{2, 3\}$. This is because a simplified distillation column model contains two or three inputs and outputs. Examples of square systems considered in this work are two-input two-output distillation system e.g. Wood-Berry (WB) model and a three-by-three distillation model given by Ogunnaike and Ray (OR) (Ogannaike & Ray, 1994). Non-square systems are not considered because the scope of this thesis is limited to the design of control system for distillation column process which contains equal number of outputs and inputs. In practice, many distillation columns have a corresponding flow rate as input variable to a corresponding output (product composition) and this is naturally a square system. Non-square systems may present difficult controllability issues and may require some form of mathematical pre-treatment before linear-time-invariant design techniques are extended to them. Emphasis is placed on practical distillation column control system design hence the limitation, in scope, to square systems only.

Some mathematical tools have been developed to identify the degree of interaction in MIMO systems. For instance, the level of interaction can be estimated using indices such as Relative Gain Array (RGA) (Luyben, 1986). A large RGA value indicates high level of interaction in a particular system and that suggests the difficulty in effective pairing of manipulated and controlled variables for control design. Similarly, small RGA signifies lower level of interaction between associated variables. This information is a useful guide in pairing output variables (y) with corresponding input variables (u) for some form of multi-loop decoupled control. RGA is generally

computed as a function of frequency. It is the corresponding matrix of relative gains (G_{ij}) as given in (1.2) where *i* and *j* denotes input-output loop's numbering.

$$RGA = [G]_{ij}\left[G^{-1}\right]_{ji} \tag{1.2}$$

where: $i, j = 1, 2, 3...$

In addition to RGA analysis, physical relationship of input and output variables is also considered during selection of variable pairs.

1.2 Background of Distillation Process

Fractional distillation is a separation technique used in crude oil refining process for separation of inlet feed mixture into constituent fractions such as gasoline, kerosene and methane based on product's volatility. Two distinct column types are found in practice namely, trayed and packed distillation columns. Trayed columns are predominant in crude oil refineries (Haydary, 2009). A trayed distillation column consists of a vertical column where trays or plates are used at each stage within the column to enhance component separation. A reboiler is used to provide heat for vaporisation of liquid products from the bottom of the column. At the top, a condenser is used to cool and condense vapour from the top of the column and a reflux drum serves as a vessel to hold condensed vapour so that liquid reflux can be recycled back from the top of the column (Balchen, 1988). Column pressure is relatively kept constant using a simple proportional controller and other relevant product can be withdrawn at the relevant plate as required.

In a trayed binary distillation control system shown in Fig. 1.3, where '*i*' denotes each tray's number, output variables are molar composition of top and bottoms products. Inputs may be drawn from any of these five manipulated variables (flow rates) namely:

- Liquid flow rate at *i*th stage in Kmol/min (L_i),
- Vapour flow rate at *i*th stage in Kmol/min (V_i),
- Distillate (Product) flow rate in Kmol/min (D_i),
- Bottoms product flow rate Kmol/min (B_{Nn}),
- Reflux flow rate in Kmol/min (L_F).

This yields five potential degrees of freedom for control design in any trayed distillation system. Therefore, composition control of distillation column is naturally a multivariable control problem.

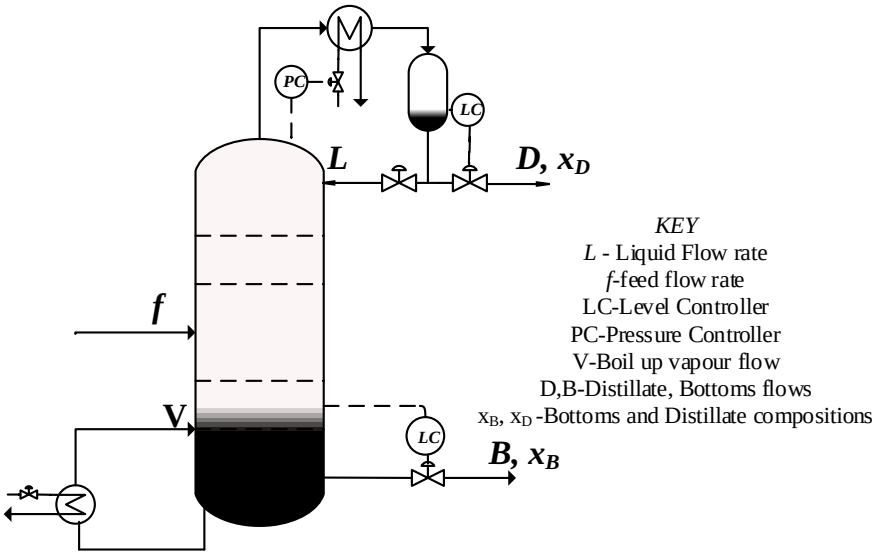


Fig. 1.3 A simple binary distillation column showing 2 outputs (top -D and bottoms - B products)

On the contrary, packed colums does not use any plates but maintains vapour-liquid contact continuously in a packed bed. The liquid flows down in the column over a packing surface while the vapour moves in the opposite direction, upwards in the column. Maintenance of good vapour -liquid distribution throughout the packed bed significantly influence the efficiency of a packed column. Comparatively:

- Packed columns are cheaper to design for distillation of corrosive mixture compared to trayed columns due to the number of plates that may be used for such applications in a trayed column.
- Trayed columns can be designed to handle a wider range of liquid and vapour flow rates than packed columns.
- The liquid hold-up is lower in a packed column compared to trays because trays have higher retention rate for liquids.
- Efficiency of a trayed column can easily be predicted unlike in packed columns because it is dependent on number of plates (flexible design parameter) (Karacan, et al., 1998).

In practice, trayed columns are found to be more common than packed columns and emphasis is placed on trayed columns although the basic theory and models of trayed column can be directly applied to packed columns too.

Three distillation model types are studied for control design namely:

- Transfer function model,
- Staged model,
- State space model.

All three considered model types are essentially linear multivariable models. Any linearisation method may be used to obtain an approximate linear multivariable model for non-linear trayed distillation column once an operating point has been identified.

In the design of distillation control systems, it is widely known that tight composition control of products with 98% level of purity is not achievable with simple proportional pressure controllers only (Balchen, 1988; Minh, 2009). This is due to sensitivity of the distillation process to feed variability, variation in input flow rate and other disturbances. Impurity concentration in end product is expected to be less than 2% in many high purity columns and this require some form of efficient control of the entire process (Minh, 2009). Additionally, safety and reliability problems can arise if liquid level in the reboiler and reflux drum drop below tolerable limit. Simple proportional controllers are identified for these inventory safety concerns. However, there is a need for implementation of composition controllers to get desired end product's quality. This is in addition to column pressure and level control systems which are typically made up of simple proportional controllers. To attenuate effects of multivariable interaction in a coupled-MIMO systems, several PID control design methods have been proposed over the years for tuning conventional PID controllers but with mixed results (Besta & Chidambaram, 2015; Chang, 2007; Chen & Marquez, 2008; Luyben, 1986). In this work, fractional order controllers are proposed to solve this multivariable distillation control problem because when properly designed, the extra tuning parameters can be tuned to counteract effects of time delays in the column (Monje, et al., 2010; Baruah, et al., 2016).

1.3 Objectives and Scope

A clear identification of an overriding set of objectives is a requisite for determining the direction of any research project and greatly influences overall probability of success of such research endeavour. The entire project plan and/or list of research activities are usually fundamental derivatives of clearly defined set of research objectives. To this end, key research questions in this thesis are defined as follows:

"*How can PID controllers and Fractional Order PID (FOPID) controllers be designed and tuned to solve a linear multivariable control problem such as composition control of distillation column with multiple time delays? If FOPID controller is selected to solve a time-delayed multivariable distillation control problem, how can it be designed and tuned to yield 98% distillate's purity and 2% bottoms impurity in the face of disturbances like 20% fluctuation in feed flow rate?"*

Many researchers can argue that design of PID controllers is now a mature field with overwhelming number of tuning methods already made available. Many of the conventional PID controller tuning methods have been reported to give satisfactory performance when the process dynamics are first or second order systems and performance requirements are modest. However, performance deteriorates when applied to solve interactive multivariable control problem such as distillation control (Astrom & Hagglund, 1995). Additionally, conventional PID controllers do not have sufficient parameters or capability to handle long dead-times in an analytical fashion (Astrom & Hagglund, 2006; Johnson & Moradi, 2006). Consequently, fractional order controllers such as FOPID controller and fractional order MPC are proposed to solve these interactive multivariable control problems. One major difference between conventional PID controller and FOPID controller is that FOPID controller is more flexible in terms of frequency domain characteristic. Given the non-restriction of orders of integral and derivative terms of the controller, more parameters are available for slope manipulation of both magnitude and phase curves at high and low frequencies. When properly tuned, this frequency domain flexibility enables FOPID controller to yield superior control performance over conventional PID controllers for a class of systems although at the expense of having more tuning parameters or

parameterised equations (Monje, et al., 2010; Baruah, et al., 2016). Flexibility of fractional order systems is also exploited in fractional order MPC to attempt to improve disturbance rejection property when compared to a standard MPC system. These frequency domain features explain the rationale behind the choice of fractional order controllers, for multivariable process control applications, over those conventional controllers where derivative orders are restricted to integers only.

One major problem associated with fractional order controllers is that there is no well-established methodology for designing MIMO FOPID controller for multivariable applications unlike SISO process control systems. Therefore, the aim of this research is to develop effective multivariable FOPID controller design schemes suitable for handling multivariable control problems like composition control of end products in a typical distillation column. This is a primary example of multivariable control problem. In addition, most distillation column models contain multiple dead-times and this introduces difficulty in composition control. FOPID controllers have long been identified as a potential control design solution for tackling control problems involving long time delays (Monje, et al., 2010).

Design and tuning of FOPID controllers for time delayed SISO processes have been developed over the years with varying degree of success. However, no established design or tuning rules have been developed for MIMO cases. Firstly, established methods of designing conventional PID controllers in a MIMO scenario is studied and adapted to design MIMO FOPID controllers. Such established design methods include Internal Model Control (IMC) and Model-based Predictive Control (MPC) design methods. A widely accepted method of tuning integer order PID controllers known as Biggest Log-modulus Tuning (BLT) method is also adapted to tune FOPI controllers. These conventional control design schemes are also considered for comparison purpose too.

In industry, many distillation columns are still operated manually without any automatic product's composition control system except simple proportional controllers for operational safety constraints like column's pressure. Therefore, a suitable automatic control system is proposed to improve purity of product's compositions in a trayed or batch distillation column. This is not limited to

conventional PID controllers only but those with fractional orders are studied and design methods are developed too. In addition, Fractional Order Predictive PI (FOPPI) control system is developed using a similar approach to conventional discrete model-based predictive control scheme. Extended simulation studies are presented to demonstrate suitability of designed controllers for tight composition control of a typical distillation column using *Matlab*/*Simulink*.

Matlab and *Simulink* are used to model a typical distillation column in order to provide a framework for control design. In a case where non-linear model is obtained, deviation variables are used to obtain an approximate linear model around a relevant operating point. Both transfer function model and state space model of distillation column are considered for controller design. Thereafter, some scalar PID controller design methods are reviewed along with their MIMO extensions in a multi-loop control system configuration. Simulation of unit step response and disturbance rejection are carried out using *Matlab* to determine suitability for multivariable control system design.

To build on robust performance of conventional PID controllers, FOPID controller design methods are developed for linear multivariable systems. Emphasis is placed on centralised multivariable controller design architecture. In addition, augmented state space model is used to develop a fractional order predictive PI control scheme for composition control of distillation column in discrete time. This allows for incorporation of future set-point information in control law formulation leading to improved set-point tracking and disturbance rejection. Simulation studies are carried out using *Matlab* and results are presented to demonstrate the benefits of proposed design schemes.

In a nutshell, objectives of this thesis are summarised as follows:

- To study both centralised controller design methods and multiloop PID control design methods such as relay feedback, Internal Model Control (IMC) and Biggest Log-modulus Tuning (BLT) methods. Given any open-loop stable distillation process, it is expected that these control design methods are implemented and results analysed for stability and robust performance,

- To evaluate and compare both performance and robust stability of these controllers using relevant sensitivity functions and uncertainty models,
- To design FOPID controllers for multivariable systems such as the experimental three-by-three distillation plant given by Ogunnaike and Ray,
- To develop a model-based predictive control method for designing FOPID controllers for a linear multivariable system.

The scope of this work is limited to control of open loop stable processes only. Open loop unstable processes are not treated primarily because distillation column, which is the primary focus of this research, is an open-loop stable process. Open loop unstable systems may require some form of pre-stabilisation before the centralised and decentralised controller synthesis methods, applicable to linear time invariant systems, are extended to them. All simulation examples are open-loop stable distillation models with two or three inputs and outputs. Distillation models considered are only those models of distillation columns set up for separation of ideal mixtures (or near-ideal mixtures) such as crude oil. There are other mixtures in practice termed azeotropic mixtures and these are liquid-mixtures that cannot be separated in a simple distillation process. This is because constituents of azeotropic mixtures have constant boiling point and cannot be altered during a simple distillation process (Arlt, 2014). Specialised distillation towers are generally required for separation of such azeotropic mixtures and the separation process typically involves a combination of techniques including distillation. Therefore, only distillation of ideal mixtures is considered where the mixture only requires one separation process only i.e. fractional distillation.

In terms of process model types used during control design, transfer function models and conventional state space models are considered in development of control design methodology. Pseudo-state space models or fractional-order system models are not considered. There are published papers for designing FOPID controllers for SISO systems with fractional order dynamics (Padula, et al., 2014). However, in this thesis, fractional order controllers are mainly used as a tool to improve control design performance and stability of conventional multivariable distillation columns. There are many examples where authors successfully utilised FOPID controllers to control integer order processes and basically exploited the design flexibility inherent within FOPID controllers (Monje, et al., 2010; Baruah, et al., 2016; Anon., 2017). However,

these methods were demonstrated for SISO systems only. In this thesis, the equivalent multivariable FOPID controller design for coupled-MIMO process control systems (with integer orders) is considered as the main focus because of the possibility of exploiting the same desirable property of fractional-order controllers (frequency-domain flexibility) for improved stability and robust performance of MIMO systems modelled with conventional integer order dynamics. It should be noted that available models of these distillation columns are conventional lumped-parameter models (ordinary differential equations with integer orders). Therefore, controller-design is aimed at practical process control systems only, with integer order models.

1.4 Preliminary Definitions

A glossary of some specialised terms found in this thesis is given in this section.

- *Centralised controller*

  A centralised controller is a fully cross-coupled multivariable controller (Johnson & Moradi, 2006).

- *Controllability*

  Input-output controllability is defined as the ability to achieve acceptable control performance, that is, to keep the outputs within specified bounds or displacements from their references in spite of unknown but bounded variations such as disturbances and plant changes, using available inputs and available measurements (Skogestad & Postlethwaite, 2005).

- *Decentralised controller*

  A decentralised controller is a multi-loop controller configured in such a way that a distinct controller term is synthesized from each major diagonal (decoupled) transfer function term. Off-diagonal (interaction terms) elements are often ignored and the structure reduces to multiple-SISO configuration (Albertos & Sala, 2004).

- *Fractional Derivative*

  Fractional-order derivative of a function $f(t)$ with order λ is defined by Riemann-Liouville as:

$$D^{\lambda} f(t) = \frac{d^n}{dt^n}\left(\frac{1}{\Gamma(n-\lambda)} \int_0^t \frac{f(\tau)}{(t-\tau)^{\lambda-n+1}} \right) d\tau, \tag{1.3}$$

where $t > 0, n-1 < \lambda < n,\ n \in \mathbb{N}$ (Monje, et al., 2010).

- *Fractional Distillation*

  Fractional distillation is defined as separation of a liquid mixture into fractions differing in boiling point by means of boiling and condensation, typically using a fractionating column (Balchen, 1988).

- *Fractional order Proportional-Integral-Derivative (FOPID) controller*

  FOPID controller may be defined in continuous form as given below:

$$C(s) = k_p + \frac{k_I}{s^{\lambda}} + k_d s^{\mu}; \qquad (1.4)$$

  where λ and μ are the integral and derivative orders; k_p, k_I and k_d are proportional, integral and derivative gains respectively (Edet & Katebi, 2016).

- *Gain margin*

  The gain margin is the amount of gain increase or decrease required to make the loop gain unity at the phase cross-over frequency where the phase angle is –180° (Seborg, et al., 2010).

- *Multivariable interaction*

  This refers to the coupling effect between different inputs and outputs in a multi-input multi-output process or system (Ogannaike & Ray, 1994)

- *Multivariable System*

  A multivariable system or process refers to systems with more than one actuating input and two or more output variables (Ogannaike & Ray, 1994). It is synonymous with multi-input multi-output system

- *Observability*

  Observability refers to the ability to infer internal states of a system from knowledge of external outputs (Wang, 2009).

- *Phase margin*

  Phase margin refers to additional phase lag that a system can tolerate before becoming unstable (Seborg, et al., 2010)

- *Robustness*

  Robustness is a measure of the resilience of a system in the face of internal changes in behaviour (Maciejowski, 1989).

- *Scalar system*

  Scalar system refers to single input single output system. There are no directions in scalar systems (Albertos & Sala, 2004).

- *Sensitivity*

  This is the measure of changes in output of a system due to small perturbations in the system (Maciejowski, 1989).

- *Stability*

  A system is said to be stable if it produces bounded output to a bounded input (Astrom & Hagglund, 1995).

1.5 Original Contributions of Thesis

Major contributions to knowledge are given as follows:

- A new method of designing fractional order PID controller (FOPID) is developed for scalar systems in chapter 4. Although the proposed procedure is similar to modified Ziegler-Nichols tuning method for conventional PID controllers in terms of requiring critical frequency point experiment, it is significantly different in terms of obtained tuning relations. Details have been published in (Edet & Katebi, 2016).
- These new FOPID controller design rules are extended to multivariable systems using sequential loop closing method in subsections of chapter 4. The *OR* distillation model is considered as process model for simulation study and FOPID controller is designed for composition control in order to improve distillate's quality. Performance is evaluated and robust stability is studied using inverse maximum singular value of sensitivity along with input/output multiplicative uncertainty models.
- IMC-BLT-FOPI controller type is developed in a multivariable setting for the first time. This is reported in chapter 5. However, fine tuning of controller parameters is achieved by BLT tuning method. Although BLT is widely studied for conventional PID controller tuning, it is modified for tuning of IMC-FOPI controllers using multiloop configuration. This new control scheme is implemented for composition control of distillation systems and it essentially overcomes the drawback of critical frequency point experiments associated

with Ziegler-Nichols type algorithms. Both *WB* and *OR* models are studied and performance is compared with optimal PID controller. Details are published in (Edet & Katebi, 2018).

- A novel predictive fractional-order control design method is developed based on augmented state space model in chapter 6. Inherent robust properties of FOPID controller are retained without losing anticipatory action and other defining features of model-based predictive control scheme (Edet & Katebi, 2018). Details of this control design scheme have been published in (Edet & Katebi, 2017).

1.6 Organisation of Thesis

In this section, organisation of the entire thesis is presented. In chapter 1, motivation, objective, outline and contributions to knowledge have been presented. In chapter 2, several methods of designing conventional PID controllers for multivariable systems are reviewed. Model-based methods such as internal model control as well as parametric model free design methods such as single frequency point methods are described in detail. Some of the reviewed methods use multiloop structure e.g. relay feedback design method. Fully cross coupled PID controller design methods such as Maciejowski or Martin-Katebi methods are also presented. A section of chapter two is dedicated to performance and robustness evaluation. Inverse maximum singular value of sensitivity functions, input multiplicative uncertainty and output multiplicative uncertainty models are considered to analyse robust stability of designed control systems. Integral Absolute Error (IAE) is used to estimate steady state error or as indicator for set-point tracking.

Chapter 3 describes system modelling. Since distillation system is the process in focus, a chapter is dedicated to modelling issues involving distillation columns. Non-linearity, sensors and general instrumentation requirements are some of the controllability and observability issues discussed. Two transfer function models are discussed and a non-linear staged model is presented too. Simulation studies are carried out on a two-by-two *WB* distillation model using some tractable methods of designing conventional PID controllers. These two linear models (*OR* and *WB*) are

thereafter used in subsequent chapters as they predominantly provide framework for control design in continuous time domain.

Chapter 4 sets out a new method of designing FOPID controller for SISO systems. Furthermore, the proposed approach is extended to multivariable systems using sequential loop closing method. A case study of distillation control is presented and results obtained via simulation studies with a three-by-three *OR* column model are presented.

In chapter 5, IMC method for designing conventional PID controllers is extended to Fractional Order PI (FOPI) controllers. The conventional IMC strategy is retained but different set of formulae are developed to set the gains of an IMC-FOPI controller. This is developed in a multivariable setting. Although initial controller settings are provided by IMC method, BLT is used to further tune these controller parameters to meet a robust specification for any linear multivariable system. Robust stability is also analysed and comparative study is presented.

Chapter 6 presents a survey of model-based predictive control design methodology. However, only one suitable predictive control algorithm is extended to design FOPI controller for both SISO and MIMO control problems in discrete time. Practical applications and limitations are also given attention. A novel fractional-order predictive PI controller design method is developed for multivariable system control.

Finally, chapter 7 states the major conclusions and summary of the reported work. Recommendations for future work concerning non-linear control of distillation system are given in this chapter. Other useful suggestions for future work are also included in this section. Appendices contain details of *Matlab* programs and *Simulink* diagrams for distillation examples reported in the thesis.

1.7 Papers and Publications

1. Edet, E. B., Katebi, R., "*Design and Tuning of Fractional-order PID Controllers for Time-delayed Processes"*, UKACC International Conference on Control. Belfast, Aug.30, 2016 - Sept 1, 2016. pp.

2. Edet, E. B., Katebi, R., ‘‘*On Fractional Predictive PID Controller Design Method’’* IFAC 2017 Conference. Toulouse, France. July 9, 2017-July 14, 2017. pp. 8555 – 8560.
3. Edet, E. B., Katebi, R., ‘‘*On Fractional-order PID Controllers*’’ Proposed Conference Paper Submitted to IFAC PID’18 Conference in Belgium June 2018. pp. 739 – 744.
4. Edet, E. B., Katebi, R., ‘‘*Fractional-order Predictive PI Controllers for Process Control Systems*’’ IET Control Theory& Applications Journal Paper (under review)

Chapter 2

Review of Multivariable PID Controllers

In this chapter, different multivariable PID controller design schemes and multi-loop PI controllers are reviewed. The aim of reviewing these established conventional PID design schemes is to identify some algorithms that may be effective in handling multivariable interactions (coupling) and use them as baseline for judging robust performance of FOPID control schemes proposed in subsequent chapters because the two controller types are similar in structure but only differ in the derivative/integral orders and frequency domain properties. In addition, some of these conventional PID control design narratives are also suitable for designing proportional-integral-derivative controllers with fractional orders with little modifications. All these design narratives are discussed in this chapter although detailed design of fractional-order PID controllers can be found in chapter 4 and in subsequent chapters. Firstly, it is imperative to understand process characteristics before designing efficient controllers to meet any pre-determined performance or stability criteria. Therefore, a generic multi-input multi-output transfer function model is briefly discussed before reviewing PID controller design methods.

A generic Multi-Input Multi-Output (MIMO) transfer function matrix can be used to represent a distillation column. Consider a MIMO transfer function model with *m* number of inputs and outputs represented with a transfer function matrix *G*(*s*):

$$G(s)=\begin{bmatrix} g_{11}(s) & g_{12}(s) & \cdots & g_{1m}(s) \\ g_{21}(s) & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ g_{m1}(s) & \cdots & \cdots & g_{mm}(s) \end{bmatrix}$$

where the elements $g_{ij}(s)$ have the form $g_{ij}(s)=G_{ij}e^{-L_{ij}s}$; $i, j = 1,2,3...m$; L_{ij} = time delay (non-negative) and $G_{ij}(s)$ represents a strictly proper, stable, scalar, real, rational function with first or second order characteristic. *G*(*s*) is assumed to be square and non-singular. Other distillation model types such as staged-equilibrium model and state space model are discussed in detail in chapter 3. A multivariable PID controller

is required to be designed to control this MIMO system $G(s)$. Inherently, the structure of the MIMO transfer function model lays bare the difficulty expected to be encountered when designing a multivariable controller for the system. This primary difficulty is associated with the presence of multiple process elements and accompanying multivariable interactions (coupling effects). These inherent limitations and multivariable characteristic are taken into consideration during the control design phase. Various methods of designing suitable multivariable PID controllers can be categorised under two major groups depending on model requirement as explained in the next section.

2.1 Grouping of MIMO PID Controllers

Existing multivariable PID controller tuning methods can be classified as model-based and non-parametric model-based methods. A design method is classified as parametric model-based method if it relies on availability of a detailed process model for controller synthesis. Non-parametric model-based design methods are based on the assumption that a detailed process model is unavailable (Katebi, 2012). Process model can be obtained from historical data and system identification or derived analytically from physical laws governing the system. Non-parametric model-based methods are suitable for online tuning of the system while parametric model-based methods are only suitable for offline tuning of the controller due to computational delays (Johnson & Moradi, 2006). PID controller design algorithms can also be grouped as follows: fixed parameter and adaptive methods; or optimisation-based methods and non-optimal algorithms. A brief classification is illustrated in Fig 2.1.

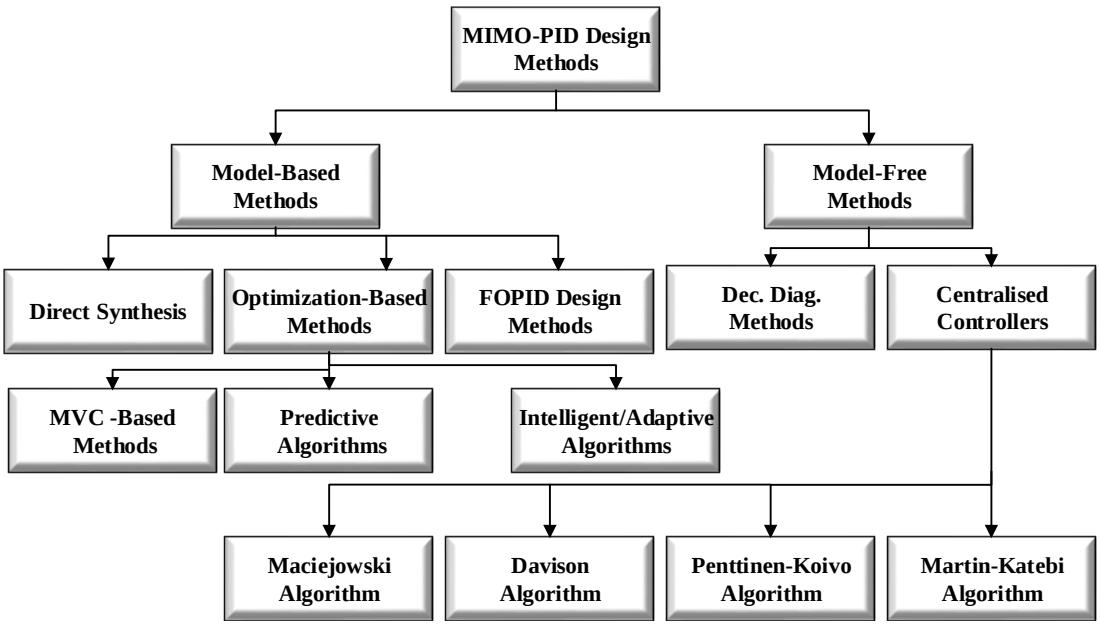


Fig. 2.1 Grouping of MIMO PID controllers

where MVC represents Minimum Variance Control; Dec. Diag. refers to Decentralised and Diagonal Elements Design Methods.

A closer examination of Fig. 2.1 reveals an increasing number of model-based methods of designing MIMO PID controllers. These methods vary from direct synthesis in time or frequency domain to optimisation-based methods (Besta & Chidambaram, 2015; Chang, 2007; Katebi & Moradi, 2001; Chen & Marquez, 2008). They are devised to meet certain performance specifications on the closed-loop time domain response or a frequency domain property (Vu & Lee, 2008). Some of the methods leave closed-loop stability to be post-checked. Several authors presented analytical solution to PID type problems and developed a direct synthesis method of designing PID controllers (Skogestad, 2012; Luyben, 1986). Other authors treated each loop of the MIMO system as a SISO and thereafter utilised Internal Model Control (IMC) to design both PI controllers and PID type controllers for the whole MIMO process (Vu & Lee, 2008; Vilanova & Katebi, 2009; Besta & Chidambaram, 2015). When a FOPDT process model is used for derivation, a PI controller is obtained using IMC method. However, when a Second Order Plus Dead Time (SOPDT) model is used as process model, a three term (PID) controller is obtained (Skogestad, 2012). Predictive PID controller design and tuning methods have been successfully developed too and it requires detailed plant model and optimisation (Katebi & Moradi, 2001).

Considering controllers with fractional orders such as Fractional Order Proportional-Integral-Derivative (FOPID) controllers, optimisation-based methods have been developed for such controllers but results have been found to be impressive predominantly in SISO applications. For instance, optimisation-based methods have been exploited to design optimal FOPID controllers using IMC framework for a class of SISO system which exhibits fractional first order plus dead-time dynamic characteristic (Padula, et al., 2014). Other optimisation based methods include: minimum variance control, intelligent control algorithms and linear matrix inequality (LMI) methods (Almeida & Coelho, 2002; Song, et al., 2014; Huynh, 2008). All these control schemes require detailed process model for controller synthesis. Generally, optimisation based methods are suitable for offline tuning due to computational delays and will not be discussed in detail. Emphasis is placed on methods that are easily adaptable to online tuning.

In contrast, there are some simple model-free methods of designing multivariable PID controllers for MIMO process control systems. Sequential loop closing technique can be combined with Ziegler-Nichols type PID tuning rules for multivariable process control (Loh, et al., 1993). This method relies on critical frequency experiments set up to identify ultimate frequency point of the plant and it is easily adaptable to autotuning. It is also implemented in a multiple-SISO loop fashion due to the decentralised approach. Decentralised approach reduces multivariable control design to a multiple-SISO loop structure where each loop handles each control variable to satisfy the design objective with respect to that variable. In addition, there are four fully cross-coupled multivariable PID controller design algorithms that will be discussed in detail in later subsections of this chapter. These four methods include Davison multivariable controller, Penttinen-Koivo PI controller, Maciejowski algorithm and Martin-Katebi multivariable controller (Davison, 1976; Katebi, et al., 2001). These are model-free methods that are easily adaptable to online tuning and do not involve extensive computational delays.

2.2 Two Major Control Structures

Two major configurations are identified for implementing multivariable PID control systems. These are centralised control structure (which is also known as fully

cross-coupled control configuration) and decentralised control structure which is also commonly referred to as multi-loop control configuration.

2.2.1. Centralised control system structure

A MIMO control system can be implemented in a fully cross-coupled or centralised feedback control structure. Each element of the transfer function matrix is equally considered for control design to reduce effects of interaction. It is also called fully cross-coupled configuration. Where this configuration is preferred, the centralised control system reduces to a simple feedback control structure. An example of a TITO centralised control system diagram is shown in Fig. 2.2. If *G*(*s*) represents a TITO distillation process transfer function matrix with FOPDT sub-models i.e.:

$$G(s)=\begin{pmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{pmatrix}; G_{ij}(s)=\frac{k_{P,ij}e^{-L_{ij}s}}{\tau_{P,ij}s+1};$$

where $i,j=1,2;\ k_P=$ steady state gain; $\tau_P=$ time constant; $L=$ dead time, a centralised PI controller of the form *C*(*s*) in (2.1) can be designed to control the TITO process.

$$C(s)=\begin{bmatrix} C_{11}(s) & C_{12}(s) \\ C_{21}(s) & C_{22}(s) \end{bmatrix}, C_{ij}(s)=k_{p,ij}\left[1+\frac{1}{\tau_{I,ij}s}\right]_{2\times 2}, \qquad (2.1)$$

where $i,j=1,2;\ k_p=$ proportional gain; $\tau_I=$ integral time.

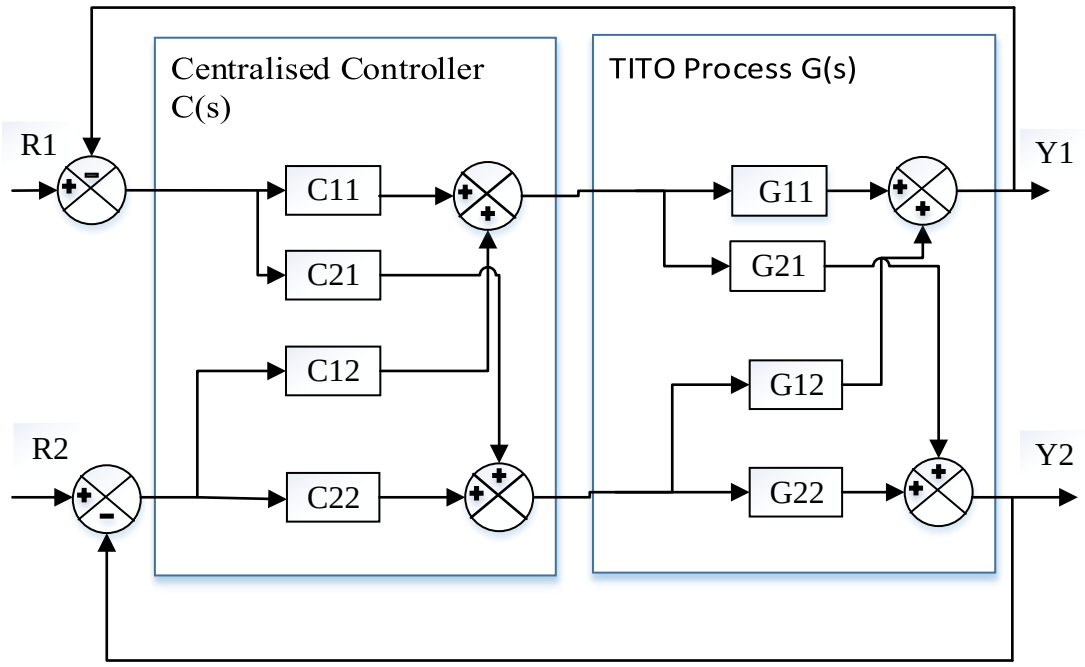


Fig. 2.2 Centralised control system C(s) with TITO process G(s)

2.2.2 Multi-loop composition control system structure

Multi-loop approach reduces the whole multivariable design problem to multiple SISO loop structure where one variable is manipulated per loop to meet desired design objective (Albertos & Sala, 2004). Only the major diagonal transfer function elements are taken into consideration during controller synthesis. A major challenge associated with any SISO based structure of multivariable control system is choice of pairing input and output variables for controller implementation. The level of cross-coupling or multivariable-interaction in distillation column complicates any basis for selection of manipulated/control variable-pair (Ogannaike & Ray, 1994). In most practical case studies of distillation column, a logical way of pairing these variables is based on physical relationship of input and output parameters. For example, pairing flows with their corresponding level measurements (Balchen, 1988). In general, input/output variables of the transfer function that will give the highest gain margin and/or phase margin are usually paired together (Morari, 2002). RGA analysis also gives an analytical methodology to variable pairing as reviewed in the introductory section. This reduces cross coupling effect. In terms of comparison with centralised architecture, it is relatively simple to tune each controller in a multi-loop arrangement

but centralised algorithms tend to give better control actions in terms of robust performance for many TITO or three-input three-output systems.

A simple diagram illustrating a TITO system (2.2) controlled with multi-loop PI controllers is given in Fig. 2.3 where

$$C(s)=\begin{bmatrix} C_1(s) & 0 \\ 0 & C_2(s) \end{bmatrix};\ C_j(s)=k_{p,j}\left[1+\frac{1}{\tau_{I,j}s}\right];\ j=1,2; \qquad (2.2)$$

k_p = proportional gain; τ_I = integral time.

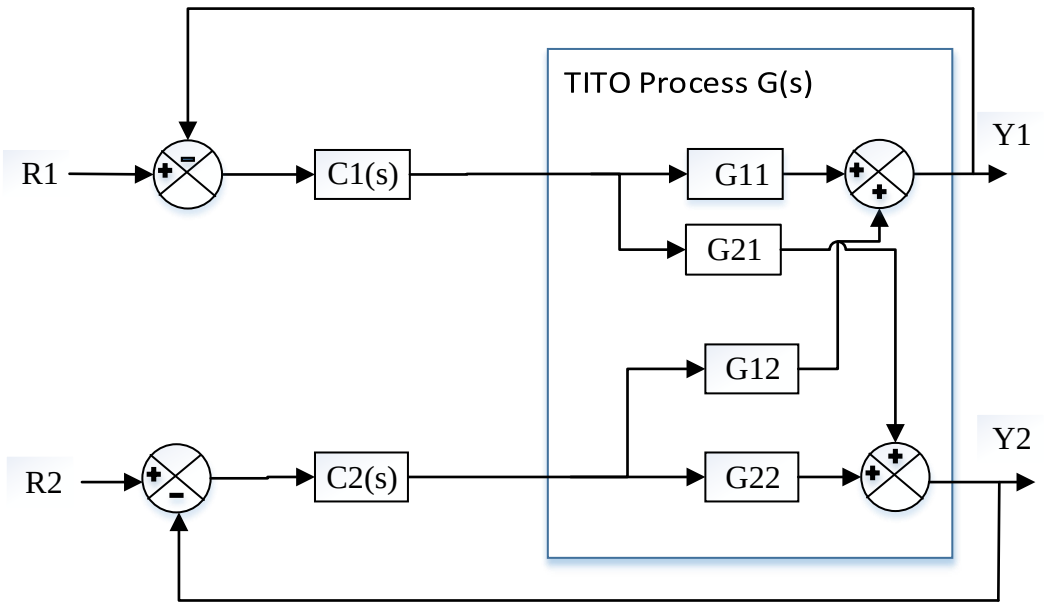


Fig. 2.3 Decentralised controllers with TITO process G(s)

2.3 Tuning of Multiloop (Decentralised) PID Controllers

Several scalar PID design and tuning methods that have been successfully extended to MIMO systems in control literature are reviewed in this section. Some of these sets of PID controller design methods include relay-feedback, IMC and BLT methods.

2.3.1 Biggest Log-modulus Tuning (BLT) Method

BLT method as proposed by Luyben is an iterative method of tuning a set of *n* multi-loop PI/PID controllers where *n* is the number of loops. In the original BLT publication, ultimate gains and ultimate periods of diagonal elements of the system's transfer function *G*(*s*) were determined experimentally as $k_{u,j}$ and $\tau_{u,j}$; $j=1,2,...,n.$

Subsequently, a Ziegler-Nichols (Z-N) setting for each loop was calculated $(k_{p,j}, \tau_{I,j})$ and final fine tuning of the conventional PID controller was carried out (Luyben, 1986). Firstly, the *j*th diagonal PI controller is given by (2.3) below:

$$C(s) = k_{p,j}(s)\left(1 + \frac{1}{s\tau_{I,j}}\right), \tag{2.3}$$

where $k_{p,j}$ = Ziegler-Nichols proportional gain; $\tau_{I,j}$ = Ziegler-Nichols integral gain. This implies that: $k_{p,j} = 0.45 \times k_u; \quad \tau_{I,j} = \tau_u / 1.2.$

A summary of the tuning procedure is presented as follows:

- Set up sustained oscillation experiment in each loop using proportional controller to determine ultimate parameters: $k_{u,j}$ and $\tau_{u,j}; \; j = 1, 2, ..., n.$
- Compute Ziegler-Nichols settings i.e.: $k_{p,j} = 0.45 \times k_u; \quad \tau_{I,j} = \tau_u / 1.2.$
- Thereafter, the characteristic function $W(\omega) = -I + \left|1 + G(j\omega)C(j\omega)\right|$ is defined where *I* is an identity matrix. The tuning factor *F* is chosen such that: *2 < F < 5*.
- Tune controller gains as follows: $k_{p,j-tuned} = \frac{k_{p,j}}{F}; \; \tau_{I,j-tuned} = F \times \tau_{i,j}.$
- The closed loop function $L(\omega) = 20 Log_{10}\left|\frac{W(\omega)}{I + W(\omega)}\right|$ is calculated and the tuning factor *F* is adjusted until $L(\omega)$ is equal to *2n* dB. Controller parameters are detuned with this new value of *F* where *n* represents number of loops.

Immediate advantages of BLT method are simplicity and very small computational demand. It can serve as a baseline to judge other model-based design methods. However, it may not meet some performance specification in certain applications except specifications are modest. It can yield high overshoot and an unstable response for highly interactive processes because of the Ziegler-Nichols tuning rule. Therefore, BLT PID tuning method, which utilizes Ziegler-Nichol's ultimate parameters, is

recommended only as a baseline for comparison of other control design schemes in terms of set-point tracking or disturbance rejection.

2.3.2 Relay Feedback Method: Sequential Loop Closing Technique.

This is similar to Ziegler-Nichols method because it uses a single frequency point information for controller synthesis but a more suitable tuning relation can be chosen for better set-point tracking. Controller implementation is achieved using a decentralised control structure by sequential loop closing. The fastest loop is first closed with a relay as a controller in order to obtain ultimate parameters before tuning the slower loops (Zhuang, 1994; Wang, et al., 1997). Once sustained oscillation is reached, the ultimate frequency (period) and ultimate gain are recorded.

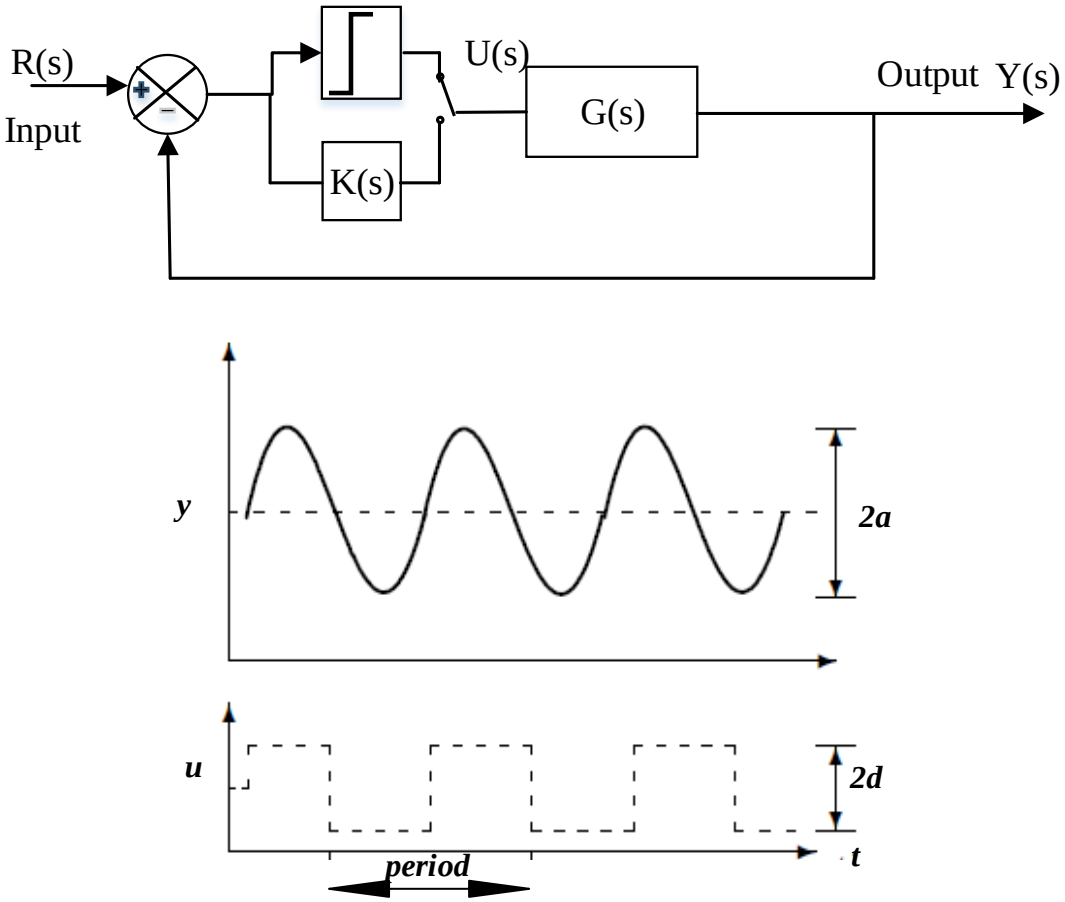


Fig. 2.4 Relay feedback control system

The term $K(s)$ represents the controller term in Fig. 2.4. Ultimate gain and ultimate period can be obtained from this experiment using $k_u = \frac{4d}{\pi a}$ where k_u = ultimate gain

and τ_u = ultimate period. Thereafter, a simple table such as Tyreus and Luyben algorithm is applied to obtain controller parameters i.e.:

$$k_p = 0.31 \times k_u,\ \tau_I = 2.2 \times \tau_u.$$

The exact procedure is repeated for the second loop with the first loop's relay replaced with the designed PI controller. Stability is achieved if the feedback control of each loop with the PI controller produces a sustained oscillation at every stage of loop closing and ultimate point parameters are correctly identified.

2.3.3 Internal Model Control (IMC)

PID parameters can be obtained by approximation of the simple feedback form of IMC-PID control structure shown in Fig. 2.5. The straightforward design approach is based on a prior process model and a low pass filter is included for robustness (Katebi, 2012). Maclaurin series or Taylor series is usually used as an approximation method to obtain a PID controller function. If the model of process is FOPDT, it naturally results in a PI controller type. However, second order model yields a conventional PID controller.

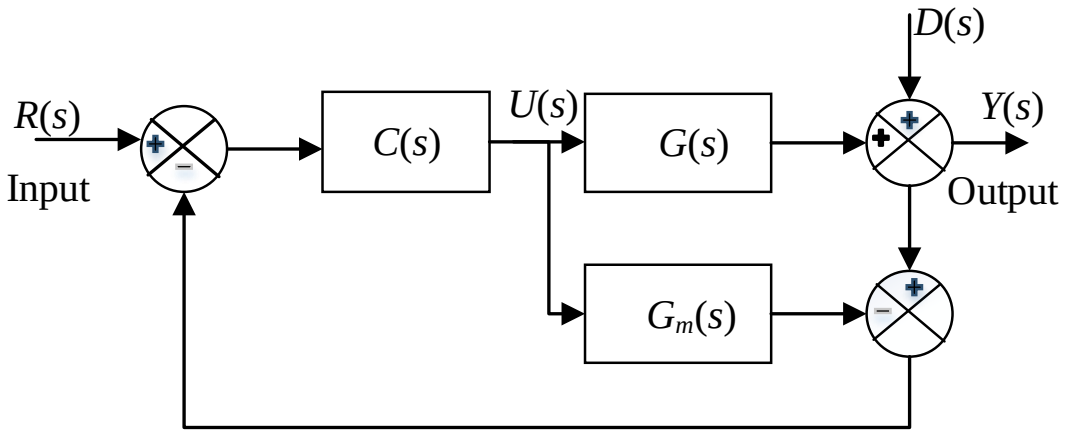


Fig. 2.5 Internal model control system

Skogestad presented another method of IMC design commonly termed as SIMC design method. Here, controller parameters are derived to meet a desired closed loop set-point specification. It retains some features of direct synthesis method. For instance, it uses process model for controller synthesis. Consider a FOPDT process $G(s) = \frac{k_P e^{-Ls}}{\tau_P s + 1}$ with internal model $G_m(s)$ as shown in Fig. 2.5 where L= Process time

delay, τ_P = Process time constant and k_P = System's steady state gain, SIMC method results in a conventional PI controller of the form (2.2) with controller gain (K_c) defined as:

$$K_c = \frac{\tau_P}{k_P(L+\tau_f)};\ \text{integral time}\ \ \tau_I = \min\{\tau, 4(L+\tau_f\}. \qquad (2.4)$$

Time constant of the filter (τ_f) is usually selected as a function of the system's time constant $\left(\tau_f = \theta\tau_p\right)$. This gives room for tuning using a small parameter (θ). θ is sometimes chosen between 0.7 and 1.5. If the model is a SOPDT system, a PID controller type is obtained with gains defined as follows:

$$K_c = \frac{\tau_1}{k_P(L+\tau_f)};\ \ \tau_I = \min\{\tau_1, 4(L+\tau_f)\}; \tau_d = \tau_2. \qquad (2.5)$$

where SOPDT model is of the form: $G(s) = \frac{k_P e^{-Ls}}{(\tau_1 s+1)(\tau_2 s+1)}$; τ_d = derivative time.

Good set-point tracking can be observed by simulating a unit step response of the closed loop control system for each design method. This is carried out for the entire controller-actuator-plant-sensor feedback control system using *WB* column in chapter three.

2.4 Centralised Controllers

In preceding sections, all PID controller design methods presented have been an attempt to extend a SISO method to multivariable control applications. These SISO-based methods do not handle interaction in an intuitive sense. If interaction is very significant, it can lead to poor control action. Consequently, fully cross-coupled or centralised PID controllers have been developed for improved performance. For instance, Lieslehto and Tanttu developed a full multivariable PID controller design and tuning scheme using IMC (Lieslehto, et al., 1993).

In Lieslehto and Tanttu's (*L-T*) algorithm, two phases of design are involved. Firstly, each constituent loop is designed independently and individual controller settings $\left(k_{p,ij}, \tau_{I,ij}\right)$ are determined using IMC. Thereafter, a centralised controller

matrix $\left(K_{p,ij}, K_{I,ij}\right)$ is obtained in the second phase. To obtain the centralised PI controller, a gain matrix containing reciprocals of scalar IMC settings is inverted. This method has been proven to be predominantly effective for TITO systems. For a TITO process model $G(s)$, a simple feedback control system can be designed using *L-T* algorithm as illustrated in Fig 2.2:

where $G(s) = \begin{pmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{pmatrix}; G_{ij}(s) = \dfrac{k_{P,ij} e^{-L_{ij}s}}{\tau_{P,ij}s + 1}; \quad i, j = 1, 2; k_P =$ process steady state gain; τ_p = process time constant; L= delay.

Let *L-T* centralised PI controller be $C(s)$; where $k_{p,ij}$ and $k_{I,ij}$ represent proportional and integral gain respectively for each *ij*th loop determined using IMC formulae. The *L-T* PI controller can be represented mathematically as given below:

$$C(s) = \begin{bmatrix} C_{11}(s) & C_{12}(s) \\ C_{21}(s) & C_{22}(s) \end{bmatrix}; C_{ij}(s) = \left[K_{p,ij} + \frac{K_{I,ij}}{s} \right]_{2\times 2}; i, j = 1, 2.$$

where gain matrices are obtained as follows: $K_{P,ij} = \begin{bmatrix} \dfrac{1}{k_{p,11}} & \dfrac{1}{k_{p,12}} \\ \dfrac{1}{k_{p,21}} & \dfrac{1}{k_{p,22}} \end{bmatrix}^{-1};$

$$K_{I,ij} = \begin{bmatrix} \dfrac{1}{k_{I,11}} & \dfrac{1}{k_{I,12}} \\ \dfrac{1}{k_{I,21}} & \dfrac{1}{k_{I,22}} \end{bmatrix}^{-1}; \quad \tau_I = \left[\frac{k_p}{k_I} \right]_{2\times 2}.$$

This algorithm is found to adequately deal with interactions in simple TITO processes. If interaction is very significant like in systems with three or more inputs and outputs, obtained gains do not handle multivariable directions properly leading to poor control action.

One limitation of L-T centralised algorithm is that it is unsuitable for larger systems where there may be three, four or more loops. There are other inherent multivariable methods of designing PID controllers for MIMO system without any restriction to the number of loops. These inherent MIMO PID controller design methods span over three decades of control research and publications. In this section,

these four fully cross-coupled MIMO PID controller design methods are presented in detail. These are:

- Davison integral controller,
- Penttinen-Koivo PI controller,
- Maciejowski multivariable controller and
- Martin-Katebi PI controller.

One common feature of these algorithms is that they do not require detailed parametric model of the plant for controller synthesis. Davison method uses an integral controller only to decouple the system at zero frequency using a constant gain compensator based on the inverse of the plant model. Penttinen-Koivo controller achieves a diagonalised plant at both low and very high frequencies. This is achieved by adding a proportional term to Davison's integral control structure.

Maciejowski developed a multivariable PI controller with the parameters (gains) calculated based on operational bandwidth frequency. Therefore, it requires frequency tests on the plant. Martin and Katebi proposed another method that retained tractable properties of Maciejowski method but without the need for plant's frequency-based experiments. A detailed implementation of these design rules is given in chapter 3. In this section, a general overview of these methods is presented.

2.4.1 Davison Integral Controller

Davison method uses an integral controller structure to derive a multivariable controller for process control. It introduces system decoupling at low frequency using a constant gain compensator based on the inverse of the plant model at zero frequency $G^{-1}(0)$ (Davison, 1976). Some simplifying assumptions are made.

- Plant model $G(s)$ is assumed to be linear, time-invariant and open loop stable.
- It is assumed that plant's steady state gain parameters are available.
- Disturbances are assumed to be constant or varying very slowly.

Therefore, Davison method reduces to finding the steady-state gain matrix of the plant for a step input (Katebi, 2012). A small tuning parameter ε is included in the controller structure for fine tuning integral gain (k_I) as shown below in (2.6):

$$u(s) = \frac{k_I}{s} e(s) \text{ where integral gain } k_I = \varepsilon G^{-1}(0). \quad (2.6)$$

The tuning parameter (ε) is usually determined online. Tuning involves making adjustments in the loops simultaneously starting with a very small positive value and increasing it until the unit step response yields maximum speed. Therefore, $G(0)$ can be obtained using step tests. In addition, since the controller is an inverse of the model at zero frequency, good decoupling at low frequencies is obtainable by Davison method (Davison, 1976). Furthermore, it guarantees asymptotic stability and asymptotic tracking for a class of disturbance given a square linear open-loop stable plant.

2.4.2 Penttinen-Koivo Method

In this method, a proportional component is added to Davison's integral controller to form a multivariable PI controller suitable for MIMO system control. Penttinen-Koivo algorithm achieves a diagonalised plant at very low and very high frequencies (Penttinen & Koivo, 1980). The control law is given below in (2.7):

$$u(s) = \left(k_p + \frac{k_I}{s} \right) e(s) \text{ where } k_p = \beta (CB)^{-1} \text{and } k_I = \varepsilon G^{-1}(0). \quad (2.7)$$

The term (β) is a constant scalar tuning parameter while k_I and k_p are integral and proportional gains respectively. *CB* is product of output and input matrices obtained from state space model of plant. In the absence of state-space model, step tests can be performed to determine *CB*. For instance, if unit step tests are carried out with each input in the matrix, *CB* can be obtained from the gradient of the output. This can be proven using Laurent series approximation. Consider a plant given as $G(s)$ where

$$G(s) = C(sI - A)^{-1} B.$$

Expanding $G(s)$ using Laurent series:

$$G(s) = \frac{CB}{s} + \frac{CAB}{s^2} + \frac{CA^2B}{s^3} + \ldots$$

Therefore, it tends to CB/s at high frequencies i.e. $\lim_{s \to \infty} G(s) \cong CB/s$

and $K_p G(s) \Rightarrow 1/s$.

In tuning the controller online, β is increased from a small positive value until the unit step closed loop response is acceptable. β can thereafter be reduced slightly and ε is increased from a small positive value until maximum speed of closed loop response is achieved.

2.4.3 Maciejowski Multivariable Controller

Maciejowski developed a multivariable PI tuning algorithm similar to Penttinen-Koivo method in (2.7) except that the controller gains (k_p, k_i) are calculated based on operational bandwidth frequency(ω_b). This is desirable because bandwidth limitation problem is associated with implementation of Penttinen-Koivo controller in real time systems. Penttinen-Koivo controller uses frequency range that is much greater than the desired frequency band. In contrast, Maciejowski's algorithm obtains PI controller parameters at operational bandwidth frequency. Proportional and integral gains are required to be proportional to the inverse of plant's dynamics at a specified frequency as shown in (2.8). A derivative term may be added to the structure if required.

$$k_I = \varepsilon G^{-1}(j\omega_b), \quad k_p = \beta G^{-1}(j\omega_b). \tag{2.8}$$

The terms (ε) and (β) are small tuning parameters typically chosen between zero and one. Frequency response is required at a single point if plant's model is available. However, if model is unavailable, practical frequency response tests should be carried out on the actual plant at the desired frequency $(\omega_b)$to yield values for gain and phase (ϕ) . If the plant is non-linear, an approximated linear result can be obtained using a low amplitude sinusoidal excitation (Katebi, 2012).

It should be noted that $G^{-1}(j\omega_b)$ term causes the solution of these equations to be complex terms, which are not physically realisable. Therefore, a real approximation of $G^{-1}(j\omega_b)$ matrix is required for final controller parameters to be realisable. Real approximation can be carried out using *Matlab*'s '*align'* function to find a constant real gain matrix (*M*), to minimise a cost function (*J*) for the entire plant with *n* loops as follows:

$$J(M,\phi) = (G(j\omega_b)M - e^{j\phi})^T (G(j\omega_b)M - e^{j\phi}) \text{ where } \phi = diag(\phi_i); i = 1,2,...,n;$$

ϕ_i = open loop phase (Vilanova & Katebi, 2009). Setting K_p to *M* produces desirable features in the multivariable system as each loop will be almost totally decoupled.

2.4.4 Martin - Katebi Method

Although Maciejowski's multivariable controller yields good response, practical frequency response experiments are required to be carried out on the plant. Martin and Katebi (2012) proposed a modified form of Maciejowski's PI controller algorithm where there is no need for frequency-based experiments. Eliminating practical frequency response experiments is meant to make control design scheme appealing to industrial practitioners. Some of the tractable properties of Maciejowski method are retained. The discretised control law is stated in (2.9).

$$u(k) = K\left(1 + \frac{T_s}{1 - z^{-1}}\right)e(k) \text{ with } K = \left(\varepsilon G(0) + (1-\varepsilon)CG_p\right)^{-1} \tag{2.9}$$

where T_s = sampling time, ε = tuning parameter, CG_p = high frequency gain product and *K*= controller gain matrix when using Martin-Katebi algorithm

According to the algorithm given in (2.9), proportional gain and integral feedback gain are obtained as a blend of the inverse of the plant's dynamics at zero $G(0)$ and high frequencies (CG_p). Therefore, if the plant has a low pass frequency characteristic, a good approximation of $G^{-1}(j\omega)$ can be obtained by selecting appropriate value for the additional controller tuning parameter ε. The parameter ε is usually fine-tuned between the interval zero and one. These four methods are selected for design of composition control system for a typical distillation system in the next chapter but firstly, some control system design issues are discussed.

2.5 Fractional Order Systems and FOPID Controllers

All the controllers reviewed before this section are conventional PID type controllers suitable for MIMO process control systems. It should be noted that another group of PID controllers with fractional orders have been developed in recent times for practical process control systems (Tepljakov, et al., 2018). These fractional order controllers are modelled with fractional-order differential equations. Traditionally, ordinary differential equations have been used as modelling tools for analytical control

design. Transfer functions, which form basis for classical control design and analysis, is a method of directly transforming ordinary differential equations in time domain to complex frequency domain to enable easy algebraic operations. State space approach tends to compute states directly in time domain using associated state equations formed from systems' differential equations. All these mathematical tools have integers as their corresponding derivative order.

However, interesting control engineering applications have been developed using differential equations of generalised order where the derivative order can be a non-integer or a fraction. One of such application of fractional calculus is the development of fractional order PID controllers. Considering time delayed process control applications, fractional order control systems have been reported to yield better performance when compared to conventional (integer order) controllers under fair comparison but at the expense of more tuning parameters (Li & Chen, 2014). The reason for superior performance of fractional order controllers have been partly due to availability of more tuning parameters and the frequency domain flexibility of fractional order system. Any generalised order differential equation such as $f = D^{\lambda}u(t) + u(t)$ is considered to be a fractional order system if the order is not an integer i.e. $t > 0, n-1 < \lambda < n,\ n \in \mathbb{N}$.

One limiting factor associated with fractional order systems is that they are irrational functions and cannot be directly realised physically. However, the fractional order derivative term can be computed approximately using some standard definitions in continuous time or discretised form. Two continuous time domain definitions that allow for computation of fractional order derivative are presented below:

- Caputo's fractional order derivative,
- Riemann-Liouville Definition.

Riemann-Liouville definition of fractional-order derivative of a function *f*(*t*) was given previously in chapter 1 i.e. (1.3)

$$D^{\lambda} f(t) = \frac{d^n}{dt^n}\left(\frac{1}{\Gamma(n-\lambda)}\int_0^t \frac{f(\tau)}{(t-\tau)^{\lambda-n+1}}\right)d\tau,$$

where $t > 0, n-1 < \lambda < n,\ n \in \mathbb{N}.$

However, Caputo gave an alternative definition of fractional-order derivative of a function $f(t)$ as:

$$D^{\lambda} f(t) = \left(\frac{1}{\Gamma(n-\lambda)} \int_0^t \frac{f^n(\tau)}{(t-\tau)^{\lambda-n+1}} d\tau \right), \text{ where } t > 0, n-1 < \lambda < n, \; n \in \mathbb{N} \qquad (2.10)$$

.

The inclusion of the nth order derivative of $f(t)$ in Caputo's definition imposes restriction compared to Riemann-Liouville form (Monje, et al., 2010). This is because it requires an absolute integration of an nth order derivative of the function $f(t)$. Also, direct application of these definitions in solving control system's problems have been limited by a common problem of selecting initial states. If a system is modelled completely using fractional calculus, it can be argued that conditions in such a system cannot be accurately described as states. Initial point is not unique, and conditions of the system can be described as pseudo-states at best. So, emphasis is placed on using the idea of fractional-order calculus as a tool to design more robust controllers for integer order systems. One example of fractional order control system is the FOPID Controller. These controllers have been applied in many systems such as mechatronic and irrigation canal control systems. Structurally, FOPID controller may be represented in the form given below in Laplace domain:

$$C(s) = k_p + \frac{k_I}{s^{\lambda}} + k_d s^{\mu}; \qquad (2.11)$$

where λ and μ are the integral and derivative orders; k_p, k_I and k_d are proportional, integral and derivative gains respectively. Therefore, five parameters are available for tuning: k_p, k_I, k_d, λ and μ. It can be observed that conventional PID controller is a subset of this generalized-order FOPID controller where λ and μ are set to one. So, FOPID is sometimes called $PI^{\lambda}D^{\mu}$ controller. Design methods for this group of controllers are discussed in chapter 4 and subsequent chapters.

2.6 Sensitivity and Uncertainty Analysis

It is important to define a suitable index for assessing the effectiveness of any given controller. Sensitivity function is a useful indicator of closed loop performance and robustness in multivariable systems. Margins of stability such as gain margin and phase margin are good indices of robust stability estimation in SISO systems but are not very suitable for multivariable system analysis and design due to presence of

multiple transfer function elements. Ideally, sensitivity of a control system is expected to fall within specified limits. Given a general feedback control system such as the system in Fig. 2.4, sensitivity function $S(s)$ can be defined as:

$$S(s) = \frac{1}{1 + C(s)G(s)}.$$

Complementary sensitivity function $T(s)$ can be defined as:

$$T(s) = \frac{C(s)G(s)}{1 + C(s)G(s)},$$

where each term retains the usual meaning as given in both Fig 1.1 and Fig. 2.4.

Maximum singular value and inverse maximum singular value analyses can be used to define sensitivity limits for MIMO systems. These indices are recommended particularly for multivariable systems and are used predominantly to assess designed control systems in this work.

2.6.1 Uncertainty

Uncertainties such as plant-model mismatch and time variations can be represented by a dynamic system block Δ. Examples of ways of handling uncertainty include:

- Unstructured additive uncertainty model,
- Input multiplicative uncertainty and output multiplicative uncertainty,
- Structured singular value.

Examples of sources of uncertainty include:

- Parametric uncertainty arises due to errors in the values of the coefficient state space matrices or transfer function matrix of the system.
- Diagonal actuators and sensors are also sources of uncertainty. Although diagonal matrices or diagonal transfer functions are used for actuator-sensor description, coupling does exist in real systems between different actuators and sensors. This coupling effect is usually ignored when using diagonal uncertainty description. For instance, in valve actuated distillation plants, diagonal input uncertainty is very significant.

- Unmodelled interconnections and known subsystems such as existence of internal cascade loops or amplifiers contribute uniquely to the overall modelling error of the system.

It is well known that high gain integral controllers can cancel out effects of static nonlinearity, parametric uncertainty and random disturbances (Maciejowski, 1989). However, unmodelled dynamics usually imply gain limitations in control system implementation. A high gain integral controller may easily result in an unstable and unsatisfactory control performance because of model mismatch. In many design solutions, trade-off between stability and performance is evident.

2.6.2 Robust Stability Analysis and Performance of MIMO control system

Robust stability analysis is required to know the degree of stability of the control system in the presence of plant-model mismatch and other uncertainties. Although many dynamic perturbations that may occur in different parts of a system can be lumped into a single perturbation block Δ, a structured method of analysing these uncertainties is beneficial to overall system design. For instance, un-modelled high frequency dynamics can be grouped together under one uncertainty block and a single metric of measurement devised to estimate robustness (Skogestad & Postlethwaite, 2005). These indices of robust stability are fundamentally based on relevant sensitivity functions and can serve as indices for comparative study. Some of these robust stability indicators are discussed below:

- Maximum Singular Value (MSV): If the maximum singular value of the complementary sensitivity function of a control system *A* is smaller than the MSV of control system *B* in all relevant frequency range, then system *A* is more robust as the stability is relatively better. MSV may be used for both structured and unstructured uncertainties and is suitable for multivariable system analysis.
- Inverse Maximum Singular Value (ISV): This is another suitable method of robust analyses for multivariable systems. If the area under the ISV curve of control system *A's* complementary sensitivity function is greater than the area under the same curve for control system *B* in all relevant frequency range, then system *A* has a greater stability region making it a more robust system. ISV may be used for both structured and unstructured uncertainty.

- Structured Singular Value (SSV): A system is stable if the SSV is less than one. With respect to internal parameter variations, SSV is relatively conservative and it is only suitable for structured uncertainties.
- Spectral radius: A system is stable if spectral radius is less than unity. It may be used for both structured and unstructured uncertainty types.

Inverse maximum singular value of complementary sensitivity function is chosen throughout this work to analyse robust stability because of suitability for multi-input multi-output (MIMO) system. It also gives a very accurate way of comparing robust stability levels of two alternative control design narratives for multivariable systems (Besta & Chidambaram, 2015). Given a process multiplicative input uncertainty $G(s)\left[I+\Delta_I(s)\right]$, Maciejowski established the stability of the system if (2.12) holds (Maciejowski, 1989).

$$\left\|\Delta_I(j\omega)\right\| < \frac{1}{\bar{\sigma}}\left\{\left[I+C(j\omega)G(j\omega)\right]^{-1}C(j\omega)G(j\omega)\right\}, \tag{2.12}$$

where $\bar{\sigma}$ is the maximum singular value of the closed loop system.

For a process multiplicative output uncertainty $\left[I+\Delta_O(s)\right]G(s)$, the closed loop system is said to be stable if (2.13) holds.

$$\left\|\Delta_O(j\omega)\right\| < \frac{1}{\bar{\sigma}}\left\{\left[I+G(j\omega)C(j\omega)\right]^{-1}G(j\omega)C(j\omega)\right\}. \tag{2.13}$$

$\Delta_I(s)$ and $\Delta_0(s)$ are assumed to be stable. A *Matlab* program can be developed to plot right hand side terms of (2.12) and (2.13) in order to reveal regions of stability for each control system. The greater the area under the curve, the greater the stability of the system. Therefore, a more robust controller will yield larger area under the curve. This way, one can judge robustness of two alternative controller types and make design decision. *Matlab* is used to simulate and analyse ISV of complementary sensitivity functions with different multivariable controller design options including fractional order controllers in subsequent chapters.

2.7 Summary

Many conventional PID control design algorithms for MIMO systems have been reviewed in this chapter. Some of these algorithms are direct extension of SISO PID design methods to multivariable systems (Palmor, et al., 1995). Since the introduction of PID controllers decades ago, several design methods have been developed and many of such methods have been adopted industrially with varying degree of commercial success. Many PID controller design patents have been obtained and commercial success of PID control technology is not in doubt.

At the core of many of these multivariable PID algorithms is the idea of using a detuning factor to reduce effects of interaction in MIMO systems (Wang, et al., 2008; Zhuang, 1994). Niederlinski modified Ziegler-Nichol's tuning rule for MIMO applications by introducing a detuning factor to meet the stability and performance specification of the system (Vu & Lee, 2008). Luyben introduced the biggest log modulus tuning method of designing PID controllers for MIMO systems. The original BLT algorithm is an iterative tuning method that extends Ziegler-Nichol's SISO rule to multivariable system using a detuning factor (Luyben, 1986). However, it gives control actions with high overshoot even in systems with moderate interaction level. This is demonstrated on a simple binary TITO distillation system in the next chapter.

Another approach to multivariable PID system control design was developed using relays. Auto-tuning strategy was also developed along with relay feedback method. Several authors recommend this method along with sequential loop closing technique (Zhuang, 1994; Palmor, et al., 1995; Loh, et al., 1993; Shiu & Hwang, 1998). In sequential loop closing method, the fastest loop is designed independently and closed first before calculating controller parameters for the remaining loops sequentially. This is implemented using a multi-loop or decentralised control structure. Although it is easily adaptable for online tuning and practical implementation, the experimental process of identifying critical frequency point can be less intuitive (Ogannaike & Ray, 1994).

Another widely reported method has been the extension of Internal Model Control (IMC) for multi-loop control systems (Vu, et al., 2007). This is a model-based design method that overcomes limitations of critical frequency point experiments.

Skogestad introduced a simple but effective method of obtaining IMC settings for SISO systems while Lieslehto and Tanttu (*L-T* algorithm) developed a multivariable PID design and tuning method with IMC (Lieslehto, et al., 1993). *L-T* algorithm gives good control actions in TITO systems but does not handle multivariable directions and interactions effectively in MIMO systems with three or more inputs and outputs.

Other multi-loop PID design methods include Dynamic Nyquist Array (DNA) and direct synthesis methods. Ho and Xu proposed the DNA method which is based on shaping the Gershgorin bands and involves a graphical procedure in the control design process (Ho & Xu, 1998). PID controllers can be synthesized in frequency domain given some well formulated stability criteria. Margins of stability such as gain margin and phase margin are used to formulate design criteria for which each controller parameter is derived to satisfy. This results in simple frequency domain parameterised algorithm. These Gain margin/Phase margin based methods yield satisfactory control actions when design specifications are modest and level of multivariable interaction is insignificant but fails to produce stable control actions for highly interactive distillation systems.

In addition, optimal solutions to multivariable PID design problem have been developed by minimising a suitable cost function. Several variants of optimisation-based methods include Linear Matrix Inequality (LMI) and minimum variance control methods. Wang developed a MIMO PID controller design scheme based on LMI optimisation with availability of state space model assumed (Wang, et al., 2006). LMI is a model-based design method that utilizes state space model in control law formulation. Some optimal methods such as Bilinear Matrix Inequality (BMI) and H_∞ do not result in PID controllers directly but require an additional procedure of model reduction in order to realise a PID controller structure. Consequently, attention have not been given to such higher order algorithms. All these optimal methods yield controllers with excellent set-point tracking but at the expense of high computational overhead.

Furthermore, intelligent control algorithms completely inspired by complex biological systems have been developed and applied to control design, parameter estimation and tuning of conventional PID controllers with integer order. These

include genetic algorithm and Particle Swarm Optimisation (PSO). Many authors have published technical papers where intelligent algorithms were successfully used to tune a set of multivariable PID controllers for MIMO applications (Mukherjee & Ghoshal, 2007; Wang, et al., 2006). For instance, Chang (2007) developed a decoupled multivariable PI controller tuning scheme using genetic algorithm. The author used a simple three parent differential crossover and uniform mutation operators in the optimisation process and results demonstrate better performance compared to BLT method. Iruthayarajan & Baskar (2009) developed a MIMO PID tuning method based on Differential Evolution (DE) algorithm. Implementation of the method on Wood-Berry distillation column for composition control produced good control performance (Iruthayarajan & Baskar, 2009). These intelligent methods compare favourably with BLT method in terms of disturbance rejection and set-point tracking measured by Integral Absolute Error (IAE). However, significant computational cost is associated with intelligent algorithms just like other optimisation-based algorithms.

PID control law have also been implemented using Fuzzy Logic Controllers (FLCs). These FLCs were previously tuned by trial and error procedure and the process was tedious and time consuming (Eranda & Mann, 2008). However, intelligent algorithms have been developed to tune FLCs. For instance, genetic algorithm has been widely used for parameter setting and tuning in fuzzy control systems. However, the drawback of complex computation involved in FLCs in a multi-loop structure implies that some decoupling methods lead to a complex rule-base. This does not offer comparative benefit over existing controllers like IMC controllers in real-time systems (Almeida & Coelho, 2002).

In recent years, many research works have identified benefits of using fractional order controllers for practical systems control (Anon., 2017; Baruah, et al., 2016). Some of the tuning strategies earlier discussed such as optimisation and intelligent methods have been utilised to tune fractional order PID controllers. Das (2012) extensively reviewed papers where intelligent methods were developed to tune fractional order PID (FOPID) controllers for scalar systems. Song (2014) transformed the problem of fractional order PID controller design to that of a static output feedback controller design in descriptor form using linear matrix inequalities (Song, et al., 2014). Furthermore, ideal decoupling, simplified and inverted decoupling techniques

associated with integer order systems have been extended to fractional order TITO processes (Li & Chen, 2014).

However, implementation of fractional order PID controllers in multivariable plants such as distillation systems with measurement and actuator delays is not yet reported. A major contribution of this thesis is therefore the development of design methods for multivariable FOPID controllers suitable for composition control of distillation column. Implementation of these fractional order controllers is demonstrated by simulation on both two-by-two and three-by-three distillation systems. Among the model-based methods of designing conventional PID controllers, IMC and MPC design algorithms have been found to be satisfactory in dealing with multivariable interactions. These two algorithms have been identified for implementing FOPID control strategy although with slight modifications in chapter five and chapter six. Specifically, Dynamic Matrix Control (DMC) algorithm has been chosen for comparative study of proposed predictive PID control system in chapter six because DMC exhibits very good predictive capability. IMC and Ogunnaike's optimal PI controllers are chosen for comparative study of a three-by-three distillation system in chapters four and five. **Table 2.1** summarises and compares these multivariable PID controller design methods.

Table 2.1 *Summary of PID Controller Design Methods*

Design Methods	**Features and Performance Comparison**	**Model Needed?**	**Experiment Required?**
Davison integral controller (Davison, 1976)	Model-free; uses integral component only; robust stability may be obtained if properly tuned; fully cross-coupled implementation.	No	Yes, except output response to step inputs is available.
Penttinen-Koivo Algorithm (Penttinen & Koivo, 1980)	PI components only; robust-performance and stability may be obtained if properly tuned; fully cross-coupled implementation.	No	Yes, except output response to step inputs is available.

The BLT Method (Luyben, 1986)	Original BLT method was based on Ziegler-Nichols; High overshoot may be obtained; Easy to understand. Multi-loop method.	No	Yes
Maciejowski Multivariable PID Controller (Maciejowski, 1989)	PI (D) components; operate within process bandwidth; bandwidth feature makes it appealing for real-time plants; robust performance and stability may be obtained if properly tuned; fully cross-coupled implementation.	No	Yes
Relay feedback tuning (Loh, et al., 1993)	Implemented in multi-SISO loop configuration; iterative tuning required; it is based on critical frequency point. Modest performance can be obtained as seen in examples; easy to adapt for online tuning.	No	Yes
Lieslehto and Tanttu IMC Algorithm (Lieslehto, et al., 1993)	Suitable for TITO systems; It is a variation of IMC; Good nominal performance can be obtained; Suitable for multi-loop method and cross-coupled (centralised) controller configurations.	Yes	No
Sequential-loop closing method (Wang, et al., 1997)	Require generalised Ziegler-Nichols or similar rules for tuning PID controller; good disturbance rejection; it may	No	Yes

	be unsuitable for demanding applications due to high overshoot; easily understood and uses multi-loop.		
Direct Nyquist Array Method (Ho & Xu, 1998)	Similar to direct synthesis; stability bounds may be used as criteria for design. Good nominal performance can be obtained. Unsuitable if there are many loops e.g five inputs/outputs.	Yes	No
Predictive PID controllers (Katebi & Moradi, 2001; Anon., 2002)	Optimal gains may be obtained although some solutions are sub-optimal; increased computational load; optimal performance within pre-defined horizons;	Yes	No
Martin-Katebi multivariable PID algorithm (Martin & Katebi, 2005)	Contains PID components; robust stability may be obtained if properly tuned; fully cross-coupled or centralised implementation. Good decoupling at high and low frequencies. Model-free.	No	Yes, except output response to step inputs is available.
Internal model control (IMC) (Vu, et al., 2007)	Straightforward just like any direct synthesis method; low pass filter may be included for robustness; many IMC variations have been developed e.g. Generalised IMC algorithm, Garcia-Morari IMC, Lieslehto-	Yes	No

	Koivo and Dong-Brosilow IMC controllers. Better nominal performance may be obtained than previous two methods at the expense of greater robustness.		
Direct synthesis (Vu & Lee, 2008)	Typically based on gain margin and phase margin for TITO systems; sensitivity bounds may be used as criteria for design. Good nominal performance can be obtained. Unsuitable if there are many loops e.g five inputs/outputs.	Yes	No
optimisation-based methods and Intelligent algorithms (Iruthayarajan & Baskar, 2009; Bouhajar, et al., 2015).	Optimal gains may be obtained although some solutions are sub-optimal; increased computational load; optimal performance within pre-defined horizons; Robustness may be sacrificed for faster response.	Yes	No
FOPID-based algorithms (Edet & Katebi, 2018; Song, et al., 2014)	Greater flexibility in design at expense of more tuning parameters.	Yes	No

Chapter 3

Distillation Modelling, Control Design and Implementation

In the preceding chapter, various PID controller design narratives have been reviewed especially, the design schemes that are suitable for multivariable process control systems. One key factor that impacts the success or failure of a controller is the level of knowledge of the process itself that is available to the control design specialist. Accuracy of system's model is a key factor that impacts efficient design and tuning of a controller if design specifications are to be met brilliantly. Sequel to this requirement, chapter 3 is devoted to modelling issues involving distillation columns, model types, sensors and actuator's constraints and implementation of designed composition controllers.

3.1 Design considerations in a Typical Distillation Column

Fractional distillation is generally carried out in a trayed or batch distillation column as explained earlier in the introductory part of chapter 1. There are operability constraints involved in routine running of distillation systems. As far as safety is concerned, these constraints require some form of simple proportional controllers to regulate them. This is required in addition to a separate products composition control system. For instance, column pressure needs to be kept fairly constant within tolerable limits and liquid levels in reflux drum and column's base have to be kept within safe limit. Sudden fluctuation in pressure is hazardous in any distillation column. For instance, sudden decrease in column's pressure can result in flashing of liquid on the trays and excessive vapour rates which floods the column while sudden increase in pressure can result in excessive condensation of vapour resulting in weeping and dumping of trays. This is widely known in industrial circles and it is routinely referred to as inventory control requirements (Balchen, 1988; Skogestad, 2007).

It has been established that this basic inventory (pressure and drum-level) control system is inadequate for operational specification of end product's quality (Balchen, 1988). So, there is need for composition control of final products to ensure relative purity of distilled fractions. In most cases, there is additional disturbances on the

system's steady state operation due to sensitivity to feed stream characteristic and changes in flow rate.

A distillation column is termed binary distillation column if only two products are obtained from the column as outputs. Examples include Wood-Berry distillation column and Minh petroleum distillation column (Minh, 2009). In these two columns, composition of top product (first output) is controlled by manipulating reflux flow rate (first input). Similarly, the composition of bottoms product (second output) is controlled by manipulating the boil-up flow rate (second input variable). It is clear that these binary distillation systems are square systems because the model consists of two input and two output variables.

There are other categories of distillation columns in existence as practical colums are not limited to binary distillation columns only. These non-binary columns are generally termed multi-component separation systems e.g. multi-staged trayed columns (Balchen, 1988). In these multi-staged trayed columns, relevant product (distilled at each tray) can be withdrawn at the relevant plate as required. For instance, in Fig. 3.1, a third product (boiling fraction *A*) is withdrawn at the middle of the column in addition to distillate and bottoms products (*D, B*). The withdrawal rate can be used as the third input (manipulated) variable to control the third product's composition (output *A*). Here, the composition of top product (first output *D*) is still controlled by manipulating reflux flow rate (first input) and the composition of bottoms product (second output *B*) is still controlled by manipulating the boil-up flow rate (second input variable). Two examples are given in this work for these multi-staged columns namely:

- Ogunnaike and Ray's distillation column (Ogunnaike & Ray, 1983).
- Giwa's three-by-three reactive distillation column (Giwa & S.Karacan., 2012).

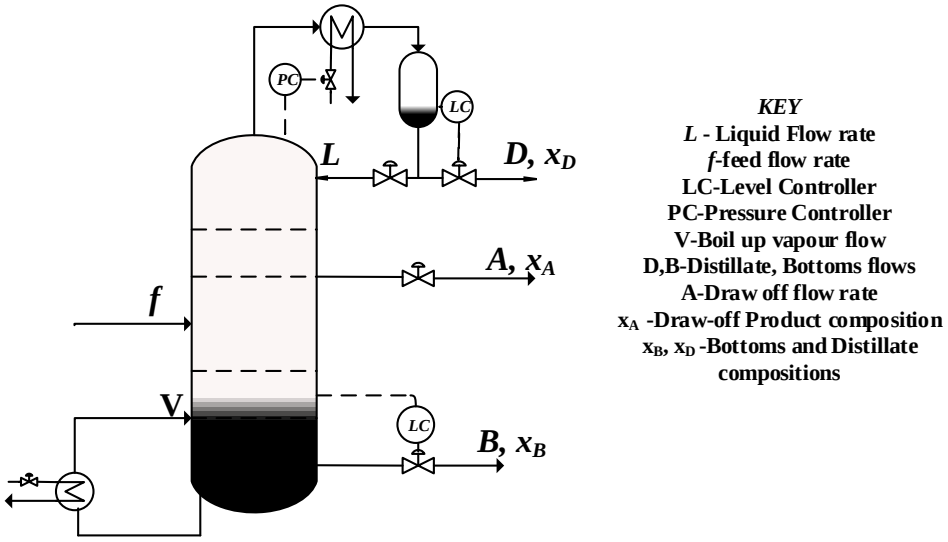


Fig. 3.1 Diagram of trayed distillation column showing three outputs (D, A, B)

Furthermore, a case study of Minh and Rani's trayed distillation column is presented in chapter six with emphasis on composition control design. It has 15 trays and was experimentally set up for petroleum distillation. Designed distillation column was expected to reach steady state product quality of 98% distillate purity at the top tray and 2% impurity of bottoms but it was never achieved with basic pressure and level control systems alone (Minh, 2009). They proposed some form of decentralised composition controller for all petroleum distillation units in the gas processing project. In this thesis, an improved centralised controller is developed for the same column and performance evaluation of both controllers is reported in chapter 6. Reflux flow rate and boil up are utilised as manipulated variables for Composition Control (CC) of both distillate and bottoms product in the binary distillation system. In a nutshell, the major factors affecting column's separation efficiency in any distillation column include:

- Relative volatility of the components (α),
- Number of trays in the column (*Nn*),
- Entry point of feed (liquid mixture) into the column (N_F),
- The ratio between liquid flow rates and vapour flow rates.

Although multicomponent distillation columns are used in practical refineries and separation systems, only a simplified model of distillation is considered in this work. An understanding of control system design for a simplified distillation column gives fundamental insight into the dynamics of complex distillation systems. For

instance, the principles of distillation of a binary mixture can be directly extended to separation of ternary mixtures in Vacuum Distillation Unit (VDU), Crude Distillation Unit (CDU) and main fractionator of practical refineries. In terms of control objectives, with respect to product composition, if the specification of purity of the distillate is higher than or equal to 98% and the impurity of the bottoms product is less than or equal to 2%, the composition control problem is said to be of tight specification (Minh, 2009).

Distillation control design is usually considered during the detailed design phase of many real-time product refining systems. Also, some model simplifying assumptions are made during the design phase. Most distillation column control systems, either conventional or advanced, assume that the column operates at constant pressure. This assumption is reasonable for modelling and design of ideal binary mixture because a simple proportional controller can keep the pressure fairly constant across a column (inventory controllers). A second assumption is that liquid and vapour are essentially in contact at the same temperature and pressure on each tray (Balchen, 1988).

Each tray in a staged distillation column is sieved and contains bubble cap for effective contact between liquid and vapour during flashing. The reboiler produces vapour which enters the column from the bottom and rises with a flow rate called the boil up flow rate (V). The feed (f) is sometimes pre-heated and introduced into the column at the middle section. As the vapour passes upwards through each tray, it bubbles upwards through the liquid. On the other hand, liquid flows downwards to lower plates in the column (Balchen, 1988). Liquid component can therefore be withdrawn as bottoms while at the top, vapour is condensed and removed as distillate. A small fraction of the liquid is fed back into the column from the top as reflux. Resistance to upward flow of vapour gives a pressure gradient across the column.

In a typical distillation column, the following process variables are identified:

- Concentration of the most volatile component in the distillate (This is affected by the top temperature),
- Concentration of the most volatile component in the bottoms product,
- Pressure at the top of the column (This is influenced by temperature at the top),

- Pressure difference between the top and bottom section of the column,
- Bottom temperature (This relates to product concentration and pressure at the bottom tray),
- Top temperature (This influences composition. Distillate's composition can be directly inferred from top tray temperature).

3.2 Process Instrumentation

In crude distillation unit of a petrochemical refinery, hundreds of flow transmitters, temperature sensors and indicators are used for measurement of process variables. Also, control valves, electric motors, pumps and compressors are required as actuators while generators and turbines are plant-wide energy inputs. Although there are several hundreds of flow meters, they serve as indicators of flow for visual display unit rather than measurements for control system. Measurement of feed, reflux and distillate flows can be carried out using suitable flow meters. Some flow meters are inaccurate and may introduce measurement errors. However, where such measurements are found to be inaccurate due to the flow meters installed, a more accurate reading can be derived from associated pump characteristics for many control applications. Sometimes, flow meters are physically blocked by tiny particles and that can interfere with accuracy of measurements.

In addition, control valves and electric motors have limits and unique characteristics, which must be considered in the design and construction of the process facility. For instance, actuator saturation must be considered during control design phase. A simple method of handling saturation constraints is control signal scaling as well as appropriate scaling of inputs and output variables. Decision on selection of measurement system types for product flow rates, composition, pressure and temperature needs to be made with respect to process specification. Typically, concentration of distillate and bottoms product can be derived from temperature measurement by appropriate location of accurate Resistance Temperature Detectors (e.g. *Platinum RTDs*) in the column. Thermocouples can also be used but it does require linearisation of output measurements.

Moreover, with advances in technology and process instrumentation, direct chemical concentration analysers are available with higher accuracy compared to

inferred data from temperature measurement. Cost-benefit analysis is also a useful decision-making tool during process facility design and selection of sensors and instruments. A brief description of required instrumentation in a typical distillation unit is presented.

3.2.1 Condenser and Associated Sensors

The condenser contains cooling water supplied by an external water pump. Flow rate is controlled by a pneumatic valve and measured by differential pressure transmitter. Temperature is also monitored using any suitable sensor. There are some desired qualities that serve as guide to selection of instruments. For instance, *Platinum RTDs* (*Pt.RTD*) are substantially linear in operation with very high sensitivity but thermocouples are more rugged and can withstand harsh environment. Thermocouples are installed in thermowells in practical plants. Level transmitter and indicators are also used for level detection and display *(Tellez-Anguiano, et al., 2009).* In summary, these factors influence selection of sensors and instruments:

- Accuracy,
- Sensitivity,
- Linearity,
- Repeatability,
- Cost,
- Resolution,
- Process compatibility in terms of signal type, instrument's size and material.

3.2.2 Chemical Sensors and Analysers

There are several chemical sensors capable of composition analysis of multicomponent stream. These chemical analysers include:

- Liquid chromatograph,
- Gas chromatograph,
- Mass spectrometer,
- UV spectrometer,
- Infra-red spectrometer,
- Near-Infra Red spectrometer (NIR),
- Nuclear Magnetic Resonance analyser (NMR),

- Chemo-luminescence analyser.

Performance characteristic or equipment specification can be found on vendor's data sheet. In addition, location of the analyser in the distillation column must be considered in the overall design. For instance, a gas chromatograph should be installed where vapour is directly obtainable and transportation delays should be analysed for effective control system application. Some distillation systems rely on column temperature for inference of approximate composition in terms of molar concentration. However, chemical analysers give more accurate measurement compared to mathematical inference from column's temperature.

3.2.3 Reboiler

Reboiler contains an electric thermo-coil which provides heat energy for vaporisation of liquid feed mixture. Reboiler is usually located at the base of the distillation column. There are RTDs or thermocouples in this unit for temperature sensing. Within the main body of the tower, which consists of several perforated plates, RTDs are also used for measuring temperature.

3.2.4 Liquid Ring Vacuum Pump

Pumps designed for refinery distillation systems are expected to be very reliable and able to safely handle explosive gases with pressure up to 15 bar (220 *psia*). This is because any failure in pump or compressor unit may result in fire hazards and production downtime and sometimes fatal loss. There are specialised pressure transmitters designed for hazardous environment. Related safety regulatory documents such as Hazard and Operability (HAZOP) guidelines can be used as a guide in selecting instrument type, capacity and specification in each unit.

3.3 Controllability

Controllability refers to an inherent property of the system concerned with the ability or inability to reach specified performance objectives given the presence of anticipated plant's variations and uncertainties. Skogestad (2005) termed it input-output controllability. In this context, controllability is defined as the ability to achieve acceptable control performance, that is, to keep the outputs (y) within specified bounds or displacements from their references (r) in spite of unknown but bounded variations

such as disturbances (*d*) and plant changes, using available inputs and available measurements (Skogestad & Postlethwaite, 2005). Any form of controllability analysis will require a system model. Accuracy of analysis of the system properties will largely depend on the accuracy of the model. In many distillation applications, a reduced-order model is usually derived for analysis and control design. Mass balance equations and thermodynamic analyses of distillation process typically result in bulky non-linear set of equations that describe system's dynamic behaviour. Such detailed model is unsuitable for controllability and stability analyses. This is due to complex levels of non-linearities. However, simple transfer function models are readily obtainable for distillation process through system identification and model fitting. Examples include two-input two-output models for binary distillation system (e.g. Wood-Berry distillation model - *WB*) and three-input three-output distillation model such as Ogunnaike and Ray's distillation system (*OR*). State space modelling technique can also be used to represents the dynamics of a distillation column.

3.4 Modelling of Distillation Column

3.4.1 Non-linear Model

Modelling and design of distillation column can be approached analytically using relative volatility of components of the binary mixture and material balance equation. However, such analytical models are derived with simplifying assumptions to have an adequate model for control design. There is no accurate model of distillation that captures all the dynamics and steady state behaviour of the process without assumptions (Rijnsdorp & Papadourakis, 1992; Tyreus, 1992; McFarlane & Rivera, 1992). This is due to complexity of variable relationships in each distillation system. Every distillation plant is unique in terms of parameterised models. Also, complex non-linear models are not suitable for most control design narratives.

However, simplified models of distillation are generally similar as they are based on fundamental operating principles. Given an *Nn*-trayed distillation column, vapour and liquid composition of light component on each tray are inter-related in a non-linear manner as given by the relative volatility equation (3.1).

$$y_i = \frac{x_i}{1+(\alpha-1)x_i};\ i = 1,2,3,...,Nn. \tag{3.1}$$

where α is the relative volatility, y_i = vapour component, x_i = liquid component on *i*th stage and *Nn* is the number of trays. Relative volatility is assumed to be constant and it is independent of pressure and composition. This non-linear relationship between vapour and liquid introduces non-linearity in the overall column model. Model types that can be used include mass-balance equations (e.g staged *column A* model), transfer function model and state space model.

3.4.2 Skogestad-Morari Column A Model

This is an example of a distillation process model derived by considering material balance and energy balance on each stage of the column. This process usually result in a non-linear model. To obtain an approximate linear model, some simplifying assumptions are made.

- Feed is assumed to be an ideal binary mixture. Azeotropic mixture is not considered.
- The feed phase condition must be known at the column pressure which is assumed to be constant.
- Total number of stages is assumed to be 41 with reboiler and condenser inclusive.
- Vapour flow is assumed to be constant at all stages.
- Vapour Liquid Equilibrium (VLE) is assumed on all trays in the column.
- Molar flows are assumed to be constant in the distillation column.
- Total condenser is assumed and liquid dynamics are linearised.

However, it does not assume constant liquid hold up on each stage implying that liquid flow dynamics is included in the model. Since the column is assumed to have 41 trays, a set of 41 differential equations describing composition dynamics on each tray were derived and simulations carried out using *Simulink* to show steady state properties of the column. Another 41 differential equations describe material hold up on each tray yielding 82 states in total. Material balance equations and steady state values of column parameters are presented.

Given that $i = 2,\ldots, Nn$ (non-feed trays) and considering light component only, let system variables be defined as follows:

- x_i = Liquid composition of light component on *i*th stage,
- y_i = Vapour composition of light component on *i*th stage,
- M_i = Liquid hold-up on *i*th stage,
- V_i = Vapour hold-up on *i*th stage,
- *Nn* = Number of stages (trays) in the column.

Skogestad (2007) and Minh (2009) derived equations for overall material balance on each tray as shown:

- Feed tray ($i=N_f$)

$$\frac{dM_{N_f}}{dt} = L_{i+1} + V_{i-1} - L_i - V_i + f. \tag{3.2}$$

Material balance of light component is given by:

$$\frac{dM_{N_f} x_{N_f}}{dt} = L_{i+1}x_{i+1} + V_{i-1}y_{i-1} - L_i x_i - V_i y_i + fz_F. \tag{3.3}$$

- At reboiler i =1; material balance of light component is given by:

$$\frac{dM_1 x_1}{dt} = L_{i+1}x_{i+1} - V_i y_i - B_i x_i. \tag{3.4}$$

Overall material balance is given by (3.5) below:

$$\frac{dM_1}{dt} = L_{i+1} - V_i - B_i. \tag{3.5}$$

- Considering the condenser, i = *Nn*.

Material balance of light component is given by:

$$\frac{dM_{Nn} x_{Nn}}{dt} = V_{i-1}y_{i-1} - L_i x_i - D_i x_i. \tag{3.6}$$

Overall material balance on condenser is given by:

$$\frac{dM_{Nn}}{dt} = V_{i-1} - L_i - D. \tag{3.7}$$

- Material balance of light component on all other trays is given by:

$$\frac{dM_i x_i}{dt} = L_{i+1}x_{i+1} + V_{i-1}y_{i-1} - L_i x_i - V_i y_i. \quad (3.8)$$

Liquid composition equation can be derived as:

$$\frac{dx_i}{dt} = \left(\frac{dM_i x_i}{dt} - x_i \frac{dM_i}{dt} \right) \Big/ M_i \quad , \quad (3.9)$$

where overall material balance is given by:

$\frac{dM_i}{dt} = L_{i+1} + V_{i-1} - L_i - V_i$ (Skogestad, 1997; Minh, 2009)

The liquid flows depend on the liquid hold up on the stage above and the vapor flow as follows:

$$L_i = L_{0i} + \frac{M_i - M_{0i}}{\tau_1} + \lambda_0 \left(V - V_0\right)_{i-1}, \quad (3.10)$$

where L_{0i} (measured in Kmol/min) and M_{0i} (Kmol) are nominal values for the liquid flow and hold up on *i*th stage. τ_1 is time constant (minute) for liquid flow dynamics on each stage. λ_0 is a constant that accounts for effect of vapor flow on liquid flow rate. If it is greater than one in a trayed column, there is a risk of getting inverse output response (Skogestad, 1997).

λ_0 is generally very small in packed columns and kept close to zero. Steady state model can be obtained from design specification of the column. Although the dynamics of the distillation column is strongly non-linear, controller design and analysis is based on a simplified model of the column linearised around the steady state operating point in Table 3.1. This is done using deviation variables. Fig. 3.2 shows simulation of column *A* at the given operating point.

3.4.2 Operating Point of Distillation Column - A

Table 3.1 shows the given operating conditions of the simplified binary distillation column *A* model.

Table 3.1 *Steady state data of column A model*

Total number of stages (*Nn*)	41
Feed tray location (N_F)	21
Feed flow rate in Kmol/min (*f*)	1
Mole fraction of feed composition (z_f)	0.5
Fraction of liquid in feed (q_f)	1
Relative volatility (α)	1.5
Liquid flow in Kmol/min (L_i)	2.706
Boil up flow rate in Kmol/min (V_i)	3.206
Distillate flow rate in Kmol/min (*D*)	0.5
Mole fraction of distillate's light composition (x_D)	0.99
Mole fraction of bottom's light composition (x_B)	0.01
Time constant for liquid flow dynamics in minutes (τ_1)	0.063
Nominal liquid hold up in Kmol (M_0)	0.5

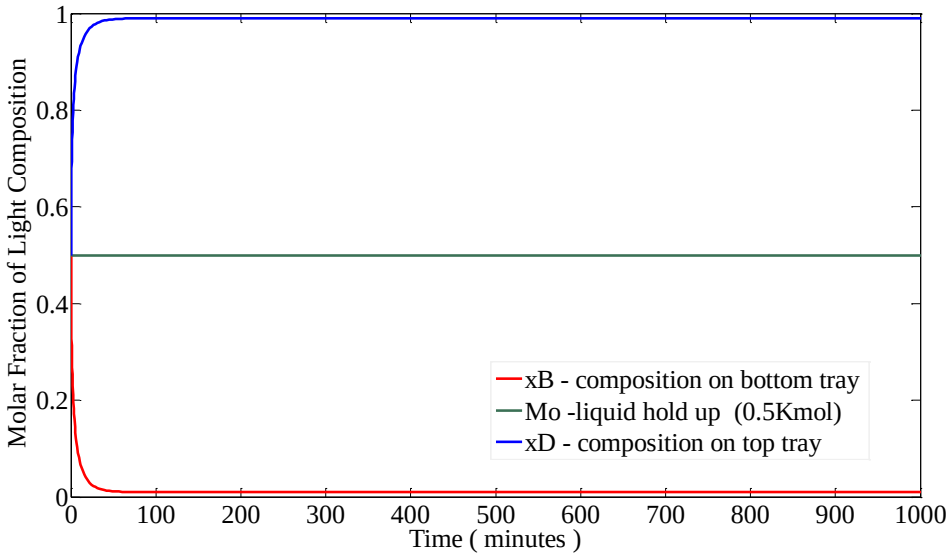


Fig. 3.2 Column A at steady state showing molar fraction of products

3.5 Linear Transfer Function Models

A second approach to distillation modelling is experimental system identification. This results in transfer function models. Examples are:

- Ogunnaike and Ray's distillation column (*OR* model).
- Wood-Berry (*WB*) Linear model: This is used in this chapter for preliminary PID controller comparisons.

3.5.1 Wood-Berry Linear model (WB)

Wood-Berry transfer function model was developed for a methanol-water pilot distillation plant. The outputs (y_1 and y_2) are mole fractions of methanol in overhead and bottoms product respectively. Variation in feed flow rate is the disturbance (*d*) term modelled. Manipulating inputs (u_1 and u_2) represent reflux flow rate and steam flow rate at the bottom (Seborg, et al., 2010). Some simplifying assumptions are given:

- Constant pressure and negligible vapour hold up: This assumption implies that a change in the vapour flow at the bottom of the column will immediately trigger a change in vapour flow at the top of the column. It is valid when vapour phase component hold-up can be neglected compared to that of the liquid phase.
- Constant molar flows: This implies that vapour flow rates in the column are equal. Wood-Berry distillation column model does not assume constant liquid hold-up and this makes it suitable for control design studies. The transfer function model is given as:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{pmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21.0s+1} \\ \dfrac{6.6e^{-s}}{16.7s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{pmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{pmatrix} \dfrac{13.8e^{-8s}}{14.9s+1} \\ \dfrac{4.9e^{-3s}}{13.2s+1} \end{pmatrix} d, \tag{3.11}$$

where

- d = Feed flow rate changes (m^3/s),
- y_1 = Overhead ethanol mole fraction,
- y_2 = Bottoms Composition,
- u_1 = reflux flow rate (m^3/s),
- u_2 = Reboiler steam pressure (kPa).

3.5.2 OR Distillation Column Model

This is a three-by-three model of distillation system developed few decades ago. A 19-plate, 12-inch diameter distillation column was experimentally set up and studied by Ogunnaike and Ray (Ogunnaike & Ray, 1983). The column had side-stream draw off as well as variable feed input with measurements taken for plate temperatures, overhead composition, reflux, feed flow rate and product lines. The distillation column was set up for ethanol-water separation as well as ternary mixtures and a three-by-three transfer function was identified as a suitable model for the experimental plant. The model is described in (3.12):

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \frac{0.66e^{-2.6s}}{6.7s+1} & \frac{-0.61e^{-3.5s}}{8.64s+1} & \frac{-0.0049e^{-s}}{9.06s+1} \\ \frac{1.11e^{-6.5s}}{3.25s+1} & \frac{-2.36e^{-3s}}{5s+1} & \frac{-0.012e^{-1.2s}}{7.09s+1} \\ \frac{-34.68e^{-9.2s}}{8.15s+1} & \frac{46.2e^{-9.4s}}{10.9s+1} & \frac{0.87(11.61s+1)e^{-s}}{(3.89s+1)(18.8s+1)} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \begin{pmatrix} \frac{0.14e^{-12s}}{(6.2s+1)} & \frac{-0.011(26.32s+1)e^{-2.66s}}{(7.85s+1)(14.63s+1)} \\ \frac{0.53e^{-10.5s}}{(6.9s+1)} & \frac{-0.0032(19.62s+1)e^{-3.44s}}{(7.29s+1)(8.94s+1)} \\ \frac{-11.54e^{-0.6s}}{(7.01s+1)} & \frac{0.32e^{-2.6s}}{(7.76s+1)} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \quad (3.12)$$

where output variables are:

- y_1 = overhead ethanol mole fraction,
- y_2 = side-stream composition,
- y_3 = bottoms composition (tray-19 temperature).

The input variables are given as:

- u_1 = reflux flow rate (m^3/s),
- u_2 = side-stream product flow rate (m^3/s),
- u_3 = Reboiler steam pressure (kPa).

The disturbance terms are:

- d_1 = Feed flow rate changes (m^3/s),
- d_2 = Feed Temperature changes (deg. Celsius).

These two transfer function models are predominantly used for controller design in this thesis. *WB* model is used to compare control actions of PID controller algorithms found to be predominant in control literature. *OR* model, which is a more interactive system, is used for demonstration of robust performance of proposed control design philosophy given in this work. In addition to these functional models, a third model type used for distillation system is the state space model. Some advanced control narratives require availability of state space model. Model-based predictive controllers introduced in chapter six is based on an augmented state space model.

3.5.3 Interaction Analysis and Variable Selection

A general definition of Relative Gain Array (RGA), as given in (1.2), shows that it is an inherent property of a system. RGA is a useful analytical tool for variable-pairing in multivariable systems such as distillation systems. A large RGA value indicates high level of interaction in a system and that suggests the difficulty in variable pairing for some form of multi-loop control. Similarly, small RGA signifies lower level of interaction between associated variables and can be paired (Skogestad & Postlethwaite, 2005). RGA is independent of plant scaling. Therefore, RGA analysis can be carried out independently before controller design. It is a function of frequency and it is an inherent property of a system.

Consider *WB* distillation system, relative gain array matrix is calculated as follows:

$$RGA = \begin{bmatrix} 0.3687 & 0.6313 \\ 0.6313 & 0.3687 \end{bmatrix}.$$

RGA is less than one. This is a relatively simple TITO system and some PID controllers are designed and tested on this system. A more interactive system is the *OR* distillation column.

The RGA of *OR* distillation system is calculated:

$$RGA = \begin{bmatrix} -0.1904 & 1.1625 & 0.0278 \\ 1.9928 & -0.1854 & -0.8074 \\ -0.8024 & 0.0229 & 1.7796 \end{bmatrix}.$$

Level of interaction is greater, and directionality is also more pronounced in this system as seen with change of sign in the matrix. It is desirable to compare centralised PID controller design methods along with multi-loop PID design methods discussed in the preceding chapter. A simple linear distillation model (*WB*) is selected for this comparative study. It is a TITO system. It is also important to understand composition control configurations in a real-time distillation system.

3.6 Control Configurations in Distillation Column

Control of product composition in a distillation column is achieved via manipulation of feed-split and fractionation using relevant flows and reflux ratio. In a binary distillation column, there are several possible configurations of manipulated variables for composition control namely:

- The energy balance structure (*L*-*V* configuration),
- *D*-*V* configuration,
- *D*-*B* configuration,
- *L*-*B* configuration,
- The double ratio configuration type such as *L/D* - *V/B* configuration,

where L_i refers to liquid flow rate at *i*th stage in the column, V_i represents vapour flow rate at *i*th stage, *D* refers to distillate (Product) flow rate and *B* represents bottoms product flow rate.

In practice, only few combinations result in a feasible control configuration despite possibility of a very large number of theoretical permutations. This is due to practical variables relationship and interactions with respect to control. The *L*-*V* and *D*-*V* methods are not self-regulating but they are generally less sensitive to feed flow rate fluctuation at high frequency. The double ratio method is self-regulating with respect to fluctuation in feed flow rate compared to the other configurations but it

requires online measurement of all flow rates which is expensive. In the *L-V* configuration, liquid flow rate (*L*) is selected to control top product concentration (distillate's composition) while vapour boil-up flow rate (*V*) is used to control bottoms composition. Energy balance (*L-V*) configuration is predominant in both experimental and industrial plants because of practical variables relationship. Consequently, it is the preferred configuration for composition control implementation in all illustrations given in this work.

Three strategies for implementing control system in a binary distillation column include: open loop control, one-point feedback control and a two-point feedback control. In the open loop case, there is no product composition control as flow rates are manually controlled by operators via valves and the column tends to drift away and be filled with light component or heavy component only. Manual monitoring is generally inadequate. This prompts the need for some form of automatic composition control system for product's quality.

In one-point control system implementation, one of the product's component is controlled (e.g. light component) with a feedback control system. This reduces it to a SISO control system. However, since top and bottoms product compositions are strongly coupled, one-point control can potentially have satisfactory control action on both distillate and bottoms' compositions in a binary distillation system if specifications are modest. Industrial implementation of one-point control with energy balance configuration in the process industry is common as it is cheaper due to fewer number of sensors and other required instruments. In contrast, both light and heavy compositions of binary mixture are controlled in a two-point control strategy (Skogestad, 1997). Liquid flow rate (*L*) and vapour flow (*V*) directly affect top and bottom compositions. In the TITO *WB* distillation control example, *L-V* configuration is used with two-point control. The reflux and steam flows are manipulated variables utilised for control of both top and bottoms' compositions.

3.6.1 Control of Process Constraints and Composition

In practical distillation plants, there are operability constraints that require simple controllers. For instance, column pressure needs to be kept fairly constant and liquid levels in reflux drum and column's base has to be kept within safe limit. This is

sometimes referred to as inventory control. Several ways of maintaining constant column pressure have been devised such as hot vapour bypass, direct pressure valve control system and vent-bleed system (Ogannaike & Ray, 1994).

Similarly, level control in the reflux drum can be achieved by manipulating distillate flow rate (top product removal rate) while the liquid level in the column base can be controlled by manipulating bottoms product flow. In a situation where the liquid level in column base is below the limit or empty, there is a likelihood of fire hazards due to overheating of the reboiler. Therefore, when liquid level in reflux drum is kept within tolerable limit, column's controllability and reduced steady state operational cost are resultant benefits. These primary operability constraints are required to be considered during the overall design review phase of the distillation system in addition to product composition control.

3.7 Multiloop PID Controller Design - *WB* Distillation Column

3.7.1 Biggest Log-modulus Tuning (BLT) Method

BLT method is the most predominant multi-loop PID controller tuning method used in literature as a baseline for comparative study of multivariable control systems. The original BLT method proposed by Luyben is considered for multi-loop PID controller design (Luyben, 1986). It is included here mainly as a baseline to judge other methods. This is because, the algorithm uses Ziegler-Nichols (Z-N) setting for controller parameters and better algorithms have been developed to build on the load rejection property of Z-N tuning rule.

Consider the first diagonal element in the model:

$G(s) = \dfrac{12.8e^{-s}}{16.7s+1}$; Each diagonal PI controller is given by equation (3.13) where each parameter retained its usual meaning as given previously.

$$C_j(s) = k_{p,j} + \frac{k_{I,j}}{s},\ j = 1, 2. \qquad (3.13)$$

Ultimate gain and ultimate period can be obtained as:

$\tau_u = 5min$, $k_u = 2.1$.

The second diagonal element is given:

$G_2(s) = \frac{-19.4e^{-3s}}{14.4s+1}$; τ_u = *14mins*, k_u = *-0.41*.

The function $W(\omega)$ is defined as $-I + |1 + G(j\omega)C(j\omega)|$.

Thereafter, closed loop function $L(\omega) = 20Log_{10}\left|\frac{W(\omega)}{I+W(\omega)}\right|$ is calculated and the tuning factor *F* is adjusted until *L* is equal to 4 dB. 4 dB point is chosen because the number of loops in *WB* column is two. With *F*=0.42, PI controller settings realised are given as follows:

$$k_p = \begin{bmatrix} 0.3750 & 0 \\ 0 & -0.0750 \end{bmatrix}, k_I = \begin{bmatrix} 0.0450 & 0 \\ 0 & -0.0031 \end{bmatrix}.$$

This controller is used as a baseline for comparative study of other controllers in terms of set-point tracking as design objective and results are presented in Fig. 3.3 - Fig.3.4.

3.7.2 Relay feedback method: Sequential loop closing

One problem with the Ziegler-Nichols based BLT method is the risk of running a real-time plant at the critical frequency point when the plant is just on the verge of instability. This experiment is not reasonably practicable when considering some critical real-time applications. Relay feedback method helps overcome this drawback while still identifying critical point parameters for control design. A relay experiment is set up in the first loop containing the first diagonal element. Loop 1 is closed and output is observed when oscillation occurs. Obtained parameters are:

τ_u = 85 – 80 = 5 *minutes*, k_u = (4 x 1)/(π x 0.375) = 3.395.

The relay is replaced with designed PI controller. Thereafter, the first loop is closed with the designed controller while setting up sustained oscillation experiment in second loop. The second loop directly contains the second diagonal element. This is the sequential loop closing technique of multivariable system. If the feedback control of each loop with the controller produces a sustained oscillation at every stage of loop closing and ultimate point parameters are correctly observed, stability is established.

Ultimate period and ultimate gain can be readily obtained as before and final controller settings are obtained using Tyreus and Luyben algorithm that relate ultimate point parameters to individual controller gains.

$$k_p = \begin{bmatrix} 0.5310 & 0 \\ 0 & -0.0780 \end{bmatrix}, k_I = \begin{bmatrix} 0.0483 & 0 \\ 0 & -0.0070 \end{bmatrix}.$$

3.7.3 Internal Model Control (IMC)

Skogestad's IMC PID commonly termed *SIMC* method is based on set-point tracking as a design objective. Controller settings are selected for output to track a given reference signal for any given first order or second order model with time delay. SIMC method results in a conventional PI controller with gains defined already in (2.5).

$$k_p = \frac{\tau_P}{k_P(L+\tau_f)};\ \tau_I = \min\{\tau_P, 4(L+\tau_f)\}. \tag{3.14}$$

A small tuning parameter θ is included when choosing filter's time constant τ_f as a function of system's time constant. $\tau_f = \theta\tau_P$.

Applying SIMC rule for the top composition loop:

$k_{p1} = 0.334,\ \tau_I = \min\{\tau_P, 4(L+\tau_f)\}$ and $\tau_I = 16.65.$

Similarly, the same algorithm is applied in second loop and obtained controller settings are given as:

$$k_p = \begin{bmatrix} 0.2609 & 0 \\ 0 & -0.0571 \end{bmatrix}, k_I = \begin{bmatrix} 0.0156 & 0 \\ 0 & -0.0039 \end{bmatrix}.$$

These controllers are simulated and performance of each PI controller is compared with the baseline method. Results of the simulation study are illustrated in Fig. 3.3 and Fig. 3.4. It is also desirable to compare these multi-loop PID controller design techniques with available centralised PI controller design techniques. Therefore, four multivariable PID control design algorithms are considered for implementation on the same distillation model.

3.8 Centralised PID Controller Design -*WB* Distillation Column

Four fully-cross coupled multivariable PID controller design methods are reviewed in this section for solving MIMO system control problems. These methods are generally grouped as centralised PID control design methods. Examples are single frequency point multivariable PID controllers and centralised IMC controllers.

3.8.1 Davison Integral Controller

This is an integral controller suitable for multivariable system and was proposed few decades ago for MIMO systems (Davison, 1976). Davison integral controller decouple a system at zero frequency using a constant gain compensator based on the inverse of the plant model. Since the controller gain is based on the inverse of the plant at steady state, good decoupling is achievable for a class of square-stable MIMO systems. The algorithm is used to design a multivariable controller for *WB* distillation column as follows:

$$u(s) = \frac{k_I}{s} e(s) \quad \text{with} \quad k_I = \varepsilon G^{-1}(0), \tag{3.15}$$

where $k_I = \varepsilon \begin{pmatrix} 0.1570 & -0.1529 \\ 0.0534 & -0.1036 \end{pmatrix}$, ε is chosen as 0.038 for a smoother response. This small tuning parameter is chosen by trial and error to have a satisfactory response speed and it is expected to be a small positive number such that: $0 < \varepsilon < 1$.

3.8.2 Penttinen-Koivo Multivariable Controller

Penttinen-Koivo multivariable controller achieves a diagonalised plant at very low and high frequencies using both integral and proportional control action. The control law is shown below in (3.16):

$$u(s) = \left(k_p + \frac{k_I}{s} \right) e(s); k_p = \beta (CB)^{-1}; k_I = \varepsilon G^{-1}(0). \tag{3.16}$$

The term β is a constant scalar tuning parameter for the proportional component. *CB* is product of output and input matrices obtained from state space model of plant. Considering the same *WB* distillation column model, obtained PI parameters are shown as:

$$k_p = \beta \begin{pmatrix} 2.7629 & -1.8546 \\ 1.2420 & -1.5722 \end{pmatrix}; \beta = 0.2;$$

$$k_I = \varepsilon \begin{pmatrix} 0.1570 & -0.1529 \\ 0.0534 & -0.1036 \end{pmatrix}; \varepsilon = 0.038.$$

It should be noted that all tuning parameters have small positive values and are chosen in a similar fashion as in the Davison algorithm i.e. $0 < \beta < 1$.

3.8.3 Maciejowski PID Controller

Maciejowski multivariable controller, defined in (2.8), is implemented using the same *WB* distillation system. The gains are as given in (3.17) where each term retains the usual meaning as defined previously in (2.8).

$$k_I = \varepsilon G^{-1}(j\omega_b), \;\; k_p = \beta G^{-1}(j\omega_b). \tag{3.17}$$

Frequency band (ω_b) of 0.2 rad/sec is selected and obtained PI controller gains for *WB* distillation model are presented as:

$$k_I = \varepsilon G^{-1}(j1.3) \cong \varepsilon \begin{pmatrix} 0.4592 & -0.3217 \\ 0.4754 & -0.0934 \end{pmatrix}, \; \varepsilon \text{ is chosen as } 0.1.$$

$$k_p = \beta G^{-1}(j1.3) \cong \beta \begin{pmatrix} 0.4592 & -0.3217 \\ 0.4574 & -0.0934 \end{pmatrix}, \; \beta \text{ is chosen as } 0.7 \text{ like in previous cases.}$$

3.8.4 Martin-Katebi Multivariable PID

According to Martin and Katebi algorithm, the control law is given as follows:

$$u(k) = K\left(1 + \frac{T_s}{1 - z^{-1}}\right)e(k); \;\; K = \left(\varepsilon G(0) + (1-\varepsilon)CG_p\right)^{-1}. \tag{3.18}$$

Each term of the controller retains the same definition as given previously in (2.9). Obtained controller gain *K* for *WB* distillation column model is:

$$K = \begin{pmatrix} 0.3439 & -0.3460 \\ 0.1416 & -0.2870 \end{pmatrix}.$$

Transient characteristic of these control systems can be observed in step response diagram of Fig. 3.3. Performance indices are also tabulated in Table 3.2.

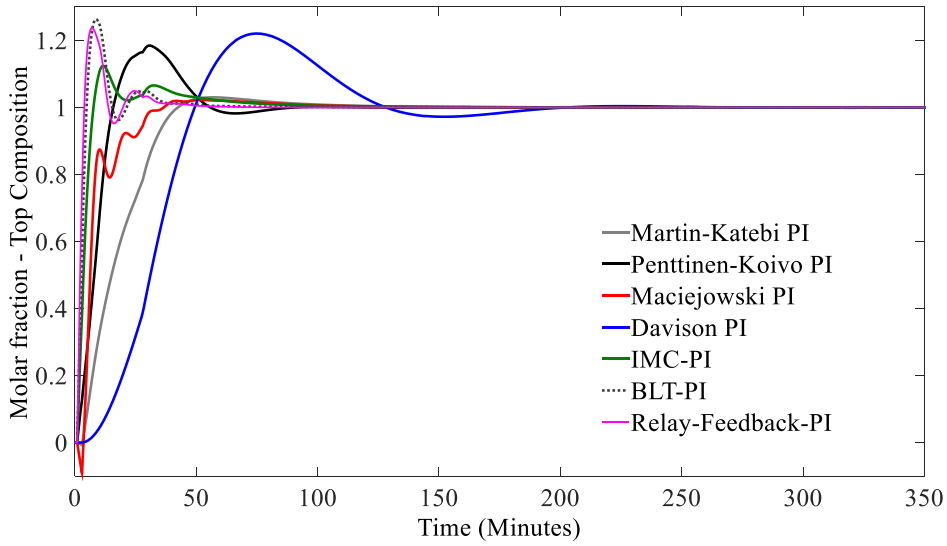


Fig. 3.3 Comparison of MIMO-PID controllers - top product

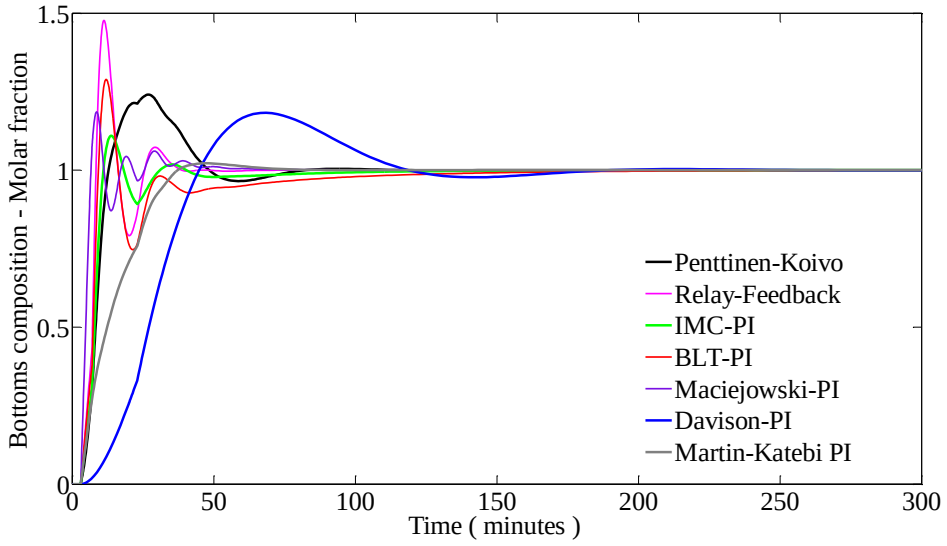


Fig. 3.4 Comparison of MIMO-PID controllers – Bottoms product

Table 3.2 *Comparison of performance in time domain*

Comparison of Control Performance	**Distillate (Top) Composition**			**Bottoms Composition**		
	% OS	***Rise Time (mins)***	***Settling Time (mins)***	***% OS***	***Rise Time (mins)***	***Settling Time (mins)***
BLT	26	5	30	30	10	40
IMC	12	7	20	10	10	32
Relay-Feedback	27	4	35	48	10	44
Davison Controller	20	50	120	19	46	100
Penttinen-Koivo	18	15	50	25	15	50
Maciejowski	0	30	40	19	7	32
Martin-Katebi	0	30	50	0	36	40

Preliminary results (as tabulated in Table 3.2) show that multi-loop methods (Relay Feedback and SIMC) yield faster response (rise time of four minutes and seven minutes respectively) but with larger overshoot compared to centralised methods (1% -20%) implying that the four centralised controllers produce a more conservative control action. In contrast, considering all three performance criteria, SIMC, Maciejowski and Martin- Katebi all gave very satisfactory performance in terms of settling time and percentage overshoot (less than 10%) consistently for both top and bottom loops. However, these results are still conservative in terms of time domain response speed for a simple TITO process. To obtain faster responses with greater robust-stability region, FOPID controllers are considered due to the extra tuning parameters that may be utilised for improved performance. In addition, these PID controllers with fractional orders are also known to counteract effects of time delays in very interactive distillation columns e.g. *OR* column (Baruah, et al., 2016; Monje, et al., 2010). If properly tuned, improved control actions are expected from fractional order controllers at the expense of one or two extra tuning parameters. It is also desirable to evaluate the performance of these controllers on a full-scale non-linear distillation column, so a brief control design review of Skogestad's non-linear distillation system (column *A*) is given in the next section.

3.19 Composition Control of Non-Linear Column *A*

It is interesting to see how centralised PID control algorithms perform in a full non-linear plant. Consequently, composition control of column *A* model is considered

here. The control objective is to regulate top composition at 0.99 (99%) thereby keeping bottoms impurity under 1% despite effects of disturbances. The disturbance signal is a 10% change in feed flow rate. Since all four centralised methods are based on linear models, an approximate form of the model is used for controller design. The model is linearised around operating point given in Table 2.1 using deviation variables. Using the exact steady state data given in Table 2.1, the un-scaled steady state gain matrix was obtained as:

$$G(0) = \begin{bmatrix} 0.8754 & -0.8618 \\ 1.0846 & -1.0982 \end{bmatrix}. \tag{3.19}$$

However, disturbance rejection studies and all control system simulations are carried out on the full 41-staged non-linear model. For instance, simulation of output response of the non-linear column with centralised composition controller when there is a 10% variation in feed flow rate (disturbance) is given in Fig. 3.6 and Fig. 3.7. The disturbance (feed flow variation) is introduced at time t=10 *minutes*. Distillate composition controller is expected to regulate composition at the nominal rate of 0.99 (99%) in the presence of feed flow rate disturbances. In this simulation study, all inputs are kept at nominal rate as given in Table 2.1 (L = 2.706 *Kmol*/*m*, V = 3.206 *Kmol*/*m*) except disturbance input f which is increased by 10%. It is desirable to study effects of these controllers in terms of disturbance rejection and output regulation. The steady state model used for controller synthesis is given as:

$$\begin{pmatrix} dy_D \\ dx_B \end{pmatrix} = \begin{bmatrix} 0.8754 & -0.8618 \\ 1.0846 & -1.0982 \end{bmatrix} \begin{pmatrix} dL \\ dV \end{pmatrix}. \tag{3.20}$$

3.9.1 Davison Multivariable Controller: Non-linear Column A Simulation

Davison's integral controller is given by:

$$u(s) = \frac{k_I}{s} e(s) \quad \text{with} \quad k_I = \varepsilon G^{-1}(0).$$

$k_I = \varepsilon \begin{pmatrix} 0.1570 & -0.1529 \\ 0.0534 & -0.1036 \end{pmatrix}$, ε is chosen as 0.038. This was chosen by trial and error. It is best to set ε to 1 before reducing it gradually to get the best or fastest response. In general $0 < \varepsilon < 1$.

On simulating Davison's controller on a full nonlinear column *A* model, it gives better output responses compared to the decentralised Skogestad's PI controller. However, other centralised algorithms give slightly better performance than Davison's integral controller. The centralised controllers regulate top composition at 0.99 and bottoms impurity at 0.01 almost perfectly except Davison's controller which deviates slightly from set-point as observed in Fig. 3.5 and Fig. 3.6.

The diagram in Fig. 3.5 has been magnified such that a slight deviation of 0.000001 molar fraction of products can be visible. Both axes are magnified. Both studies show sensitivity of control system to unmodelled non-linearity in the system. Skogestad's decentralised PI controller originally designed for column *A* is included as a baseline for comparison. Disturbance is introduced and effects of two centralised controllers can be observed in Fig. 3.7 as disturbances are rejected though very sluggishly. It is to be noted that Skogestad's PI controller, which is used here as a baseline, is a decentralised optimal PI controller obtained directly using *Matlab's* optimisation toolbox.

3.9.2 Penttinen-Koivo Controller (*Non-linear Simulation*)

Using Penttinen-Koivo method, control law is derived as follows:

$$u(s) = \left(k_p + \frac{k_I}{s} \right) e(s); k_p = \beta (CB)^{-1}; k_I = \varepsilon G^{-1}(0).$$

$$k_p = \beta \begin{pmatrix} 2.3433 & -1.0794 \\ 0.5115 & -0.8080 \end{pmatrix},\ k_I = \varepsilon \begin{pmatrix} 0.7361 & -0.5802 \\ 0.7270 & -0.5894 \end{pmatrix}.$$

Typical values of β and ε used are 0.8 and 0.6 respectively. Results of full non-linear simulation is shown in Fig. 3.5 and Fig. 3.6. Disturbance rejection property is demonstrated in Fig. 3.7 and Fig. 3.8.

3.9.3 Maciejowski Multivariable Controller (*Non-linear Simulation*)

Maciejowski controller parameters are computed at plant's operating frequency. Therefore, range of frequency of interest for the distillation column is extracted from the Bode diagram of the linearised model. A bandwidth of 0.2 rad/min is suitable for design.

$$k_p = \beta G^{-1}(j0.2) = \beta \begin{bmatrix} 0.0115 & -0.0150 \\ 0.0120 & -0.0279 \end{bmatrix}$$

where β is a small positive parameter. A typical value of 0.7 gives a fast response for this system hence it is chosen as 0.7.

$k_I = \varepsilon G^{-1}(j0.2) \cong \varepsilon \begin{bmatrix} 0.0115 & -0.0150 \\ 0.0120 & -0.0279 \end{bmatrix}$, ε chosen as 0.7 as explained previously

Simulation results are given in Fig. 3.5 and Fig. 3.6. It is compared with other centralised PID controllers.

3.9.4 Martin-Katebi Multivariable Controller (Non-linear Simulation)

The control law is defined as:

$$u(k) = K\left(1 + \frac{T_s}{1 - z^{-1}}\right)e(k),\ K = \left(\varepsilon G(0) + (1-\varepsilon)CG_p\right)^{-1}.$$

For *WB* column, matrix (K) is obtained as follows:

$$K = \begin{pmatrix} 0.2081 & -0.1035 \\ 0.0634 & -1.1535 \end{pmatrix}.$$

Fig 3.5 shows how it compares with other centralised controllers in terms of top composition regulation on the full non-linear column *A*. The output response, after 10% feed-variation (disturbance) is introduced in the system, is shown to have different settling times for two centralised controllers as shown in Fig 3.7.

Fig 3.6 shows effect of all four centralised controllers in terms of bottoms composition regulation at 0.01 impurity (99% molar percentage). The simulated outcome is magnified such that a difference of 0.00000001 unit of deviation can be visually observed. However, all four centralised controllers regulate the plant at 0.99 molar fraction for the top which correspond with 99% top purity. The centralised controllers all give smoother output compared to the decentralised controller.

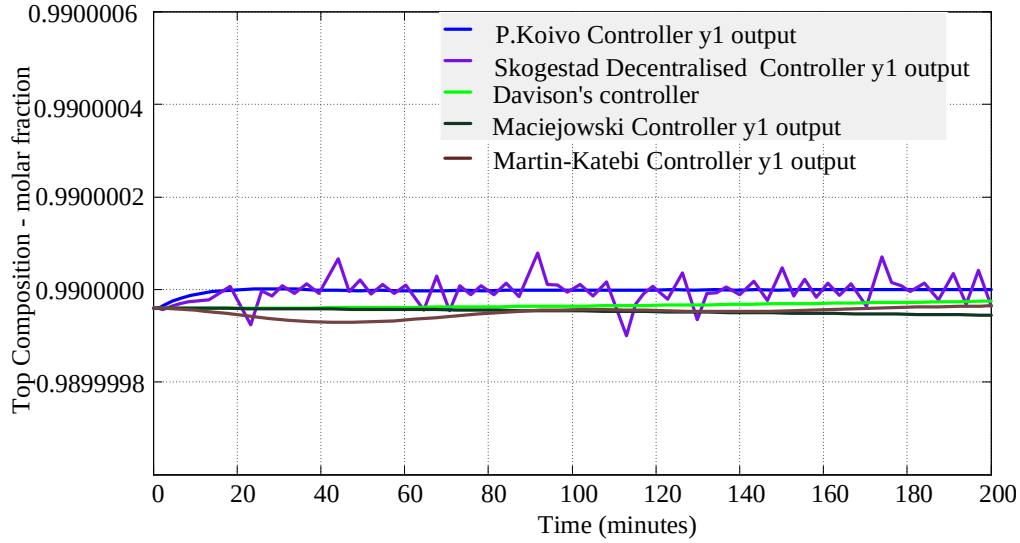


Fig. 3.5 Top composition: Comparison of different controllers

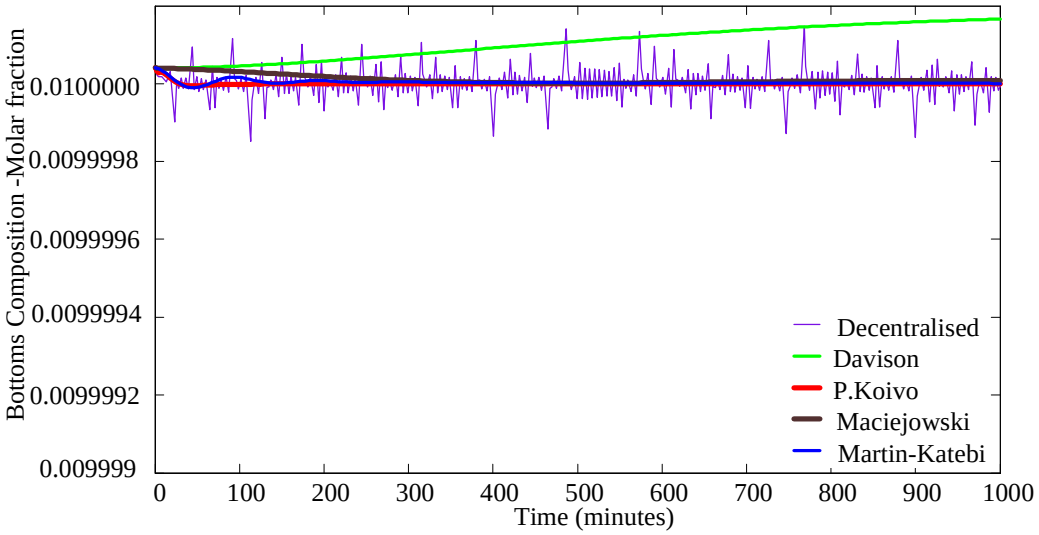


Fig. 3.6 Bottoms composition: Comparison of different controllers

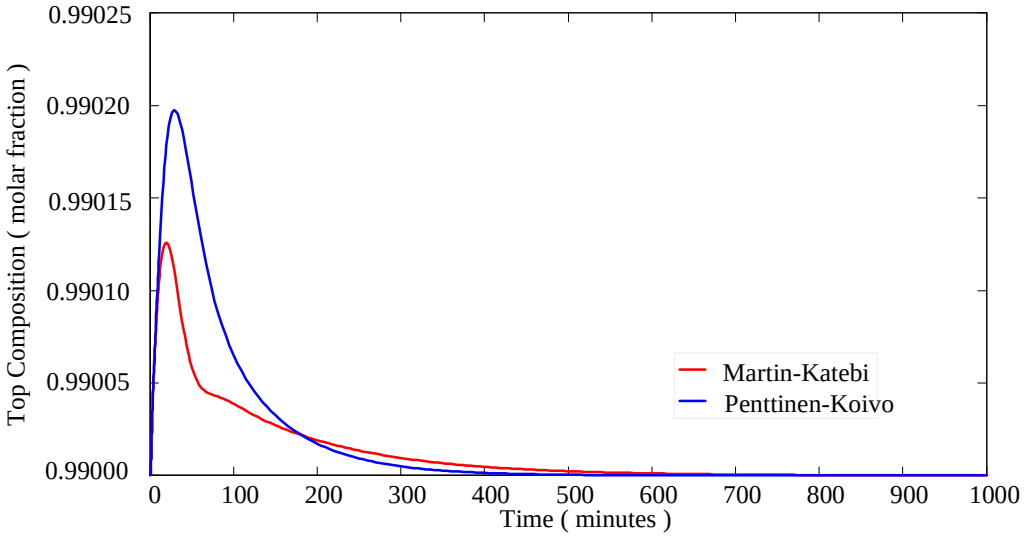


Fig. 3.7 Disturbance rejection - 10% feed flow(f) variation in top loop

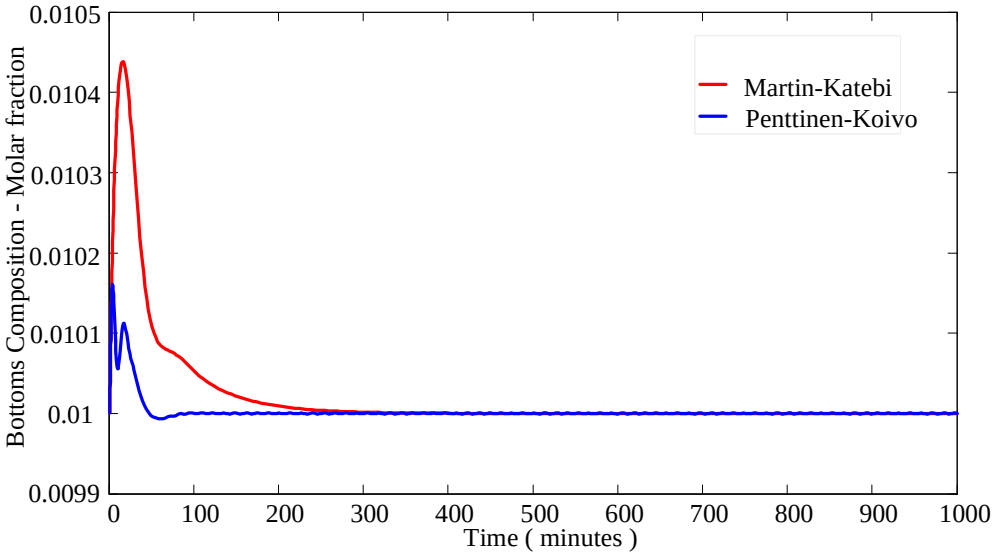


Fig. 3.8 Disturbance rejection - 10% feed flow (f) variation in bottoms loop

Martin-Katebi and Penttinen-Koivo controller performs better in terms of disturbance rejection in non-linear column *A* simulation (Fig. 3.7 and Fig. 3.8). Also, all these controllers take a long time (more than 200 mins) to recover from perturbation (recovery or settling time). Furthermore, implementation of these centralised controllers on the *OR* column, which is a highly interactive three-input three-output system with significant time delays, give oscillatory output responses. This clearly shows the limitation of conventional PID controllers with three terms: proportional, integral and derivative components, in controlling multivariable systems with three inputs and three outputs. It is well-known that FOPID controllers builds on advances

made with conventional three-term (PID) controllers (Podlubny, 1999; Hu, 2001; Lee & Chang, 2010). Although, all the authors (cited here) demonstrated these comparative advantages of FOPID controllers over conventional PID controllers for SISO systems, this thesis aims to demonstrate that improved results with FOPID controllers can be extended to MIMO systems. Therefore, FOPID controllers are considered to obtain improved control action for time-delayed MIMO systems. The extra flexibility offered by non-restriction of derivative order is expected to be exploited for improved performance. Details are reported in chapter 4 and subsequent chapters.

3.10 Summary

In conclusion, three different types of models have been presented for modelling of distillation column for the purpose of composition control design. These include:

- Staged modelling method using mass balance equations,
- transfer function models usually obtained using step response tests,
- state space model.

Staged model results in a non-linear set of equations because of non-linear vapour-liquid relationship. However, linearisation is carried out around plant's operating point using deviation variables to obtain a linear model. An example of staged model is the Skogestad's *column A* model. Transfer function models are linear time invariant models. Examples are *WB* TITO model and the three-by-three *OR* distillation model. Two state space models of distillation column considered in chapter six are Karacan-Giwa state space model and Minh petroleum distillation model. It is also well known that actuator delays and non-linearity can cause linear control system to deteriorate in real-time applications. Therefore, practical instrumentation constraints involving distillation columns have been addressed in this chapter.

Thereafter, MIMO PID controller design schemes reviewed in chapter two have been extended to composition control of a simple TITO *WB* distillation column. Both decentralised and centralised controllers have been designed for the same system to compare performance. Centralised control algorithms take interaction terms in system's transfer function into account during controller design phase. Decentralised algorithms do not account for interaction terms during control law formulation.

Simulation studies show that centralised controllers give better control action with very little or zero overshoot when compared to decentralised (multi-loop) controllers except Davison integral controller (20% overshoot).

Consequently, two centralised controllers have been extended to a non-linear staged column *A* model. Although controller synthesis is based on linearised model, simulations have been carried out on the full non-linear column *A* model with 10% feed-flow disturbance introduced. Both Penttinen-Koivo controller and Martin-Katebi PI controller deal adequately with interactions, non-linearity and disturbance rejection. However, the best response (Martin-Katebi) has a settling time of 200 minutes. This is a very conservative performance. To improve this time response, fractional-order PID controllers are proposed in subsequent chapters.

Chapter 4

Fractional-order PID Controller Design and Tuning

In the previous chapter, the design of conventional (multivariable) PID controllers have been presented and control performance evaluated on a typical binary distillation column with no disturbances (nominal performance). Simulation results of these conventional controllers have been very conservative on the two-input two-output Wood-Berry distillation column with very high rise time and settling time ranging from 40 – 150 minutes. Consequently, these conventional PID controllers cannot possibly give superior performance in a more interactive (coupled) three-input three-output distillation column (OR column) compared to the TITO system. This is because, ORA column has higher coupling effects judged by higher RGA values as earlier given in chapter 3 and there is a change of sign within the RGA matrix indicating greater control design difficulty compared to the *WB* column given in chapter 3.

i.e. RGA of *OR* distillation system is calculated as follows:

$$RGA = \begin{bmatrix} -0.1904 & 1.1625 & 0.0278 \\ 1.9928 & -0.1854 & -0.8074 \\ -0.8024 & 0.0229 & 1.7796 \end{bmatrix}.$$

While the RGA of the relatively simple TITO system (*WB* column) is less than 1 and all elements are positive as shown:

$$RGA = \begin{bmatrix} 0.3687 & 0.6313 \\ 0.6313 & 0.3687 \end{bmatrix}.$$

Therefore, FOPID controller is chosen for product's composition control of *OR* column due to its inherent frequency domain flexibility. Critical-frequency-point method of designing Fractional Order PID controller (FOPID) is proposed for multivariable process control applications such as distillation column control system in this section. Since ultimate parameter-based tuning rules applicable to conventional

PID controllers (e.g. Ziegler-Nichol's tuning rule, Cohen-Coon and Hagglund's rules) are not directly applicable to the tuning of FOPID controllers, a new algorithm is developed to tune these FOPID controllers based on critical-frequency-point test. Minimal information about process's dynamics is required and it is easily adaptable to online tuning and implementation. Results of these MIMO-FOPID controllers have been published (Edet & Katebi, 2016). Firstly, well-established methods of designing FOPID controllers for SISO systems are reviewed including methods of implementing irrational functions. More detailed review of practical applications of FOPID controllers in industry can be found in Aleksei's paper (Tepljakov, et al., 2018).

4.1 Introduction

FOPID controllers can be designed and tuned for both SISO systems and MIMO systems involving time delays (Baruah, et al., 2016; Vale´rio & Costa, 2009; Monje, et al., 2010). Considering SISO control system applications, many well-known methods of tuning this class of controllers exist. These design methods are either based on time domain parameter optimisation or frequency domain. FOPID controller parameters can be computed directly in frequency domain to meet some desired specifications such as phase margin, constant phase at gain cross over frequency and other sensitivity constraints as shown by several authors (Luo, et al., 2010; Luo & Chen, 2009; Vale´rio & Costa, 2009). Other methods of designing FOPID controllers include optimisation-based methods such as Linear Matrix Inequality (LMI), probabilistic optimisation index, and evolutionary algorithms (Lee & Chang, 2010; Song, et al., 2011; Wu, et al., 2018). All these papers present design of FOPID controllers for SISO applications only.

FOPID controller is flexible in terms of frequency domain characteristic. Given the non-restriction of orders of integral and derivative terms of the controller, more parameters are available for slope manipulation of both magnitude and phase curves at high region and lower range of frequency. When properly tuned, this frequency domain flexibility enables FOPID controller to yield superior control performance over conventional PID controllers for a class of systems. Several authors demonstrated this comparative benefit by simulation studies of both FOPID controller and conventional PID controllers (Monje, et al., 2010; Lee & Chang, 2010; Padula, et al.,

2014). Clearly, these comparative benefits can easily be understood when considering frequency domain properties of an ideal integrator compared to a fractional order type. In the case of fractional or generalised integrator, magnitude and phase properties can easily be manipulated freely using appropriate gains to meet extra gain margin or phase margin constraint. Such freedom is limited when considering a pure integrator. In terms of robustness to gain-variations and uncertainties, FOPID controller is known to give better performance compared to conventional PID controller when properly designed. For instance, loop shaping design technique can be used to design FOPID controller in such a way that the phase is kept constant around the gain cross over frequency. This robust property of FOPID controller is explored in detail in section 4.2

Considering steady state error rejection, a fractional order integrator of order $k+\lambda; (\ k \in \mathbb{N};\ 0<\lambda<1)$ is as efficient as a conventional integer order integrator of order k+1 (Monje, et al., 2010). This can easily be confirmed using final value theorem. Given a system $G(s)$:

$$\text{where } G(s) = \frac{K(a_m s^{\alpha_m} + a_{m-1} s^{\alpha_{m-1}} + ... + 1)}{s^{\lambda}(b_n s^n + b_{n-1} s^{n-1} + ... + 1)},\ \alpha_m \in \mathbb{N}.$$

Assuming a reference signal (r) of the form: $r(t) = At^{\lambda}$, $R(s) = \frac{\Gamma(\lambda+1)}{s^{\lambda+1}}$, steady state error can be analysed with both conventional integral controller and fractional order integrator. The conventional (integer order) integrator is well known to eliminate steady state error. So, steady state error analysis with equivalent fractional order integrator is presented. Position error constant is given by $K_p = \lim_{s\to 0} ks^{-\lambda}$, implying that steady state error (position) $e_p = \frac{1}{1+K_p}$. Considering velocity error constant (K_v), $\lim_{s\to 0} s.ks^{-\lambda} = \lim_{s\to 0} ks^{1-\lambda}$; steady state error (velocity) $e_v = \frac{1}{K_v}$.

Also, considering acceleration error constant (K_a), $\lim_{s\to 0} s^2.ks^{-\lambda} = \lim_{s\to 0} ks^{2-\lambda} \Rightarrow e_a = \frac{1}{K_a}$;

where e_a refers to steady state error (acceleration). Similar steady state error results are obtainable with conventional (integer order) integral controller where $\lambda = 1$.

In terms of implementation, FOPID controllers must be band limited. The fractional order integrator $\left(s^{-\lambda}\right)$ can also be implemented as $s^{-1} \times s^{1-\lambda}$ to take advantage of the action of pure integrator at very low frequencies. This effectively eliminates steady state errors at very low frequencies. Three scenarios are identified where fractional order dynamics can be applied:

- Design of fractional order controller for a conventional (integer order) plant. Examples can be found in many published papers (Baruah, et al., 2016; Tepljakov, et al., 2018)
- Design and tuning of integer order controllers for a class of systems identified with fractional order dynamics (Monje, et al., 2010).
- Design of fractional order controllers for a class of systems identified with fractional order dynamics (Luo, et al., 2010; Luo & Chen, 2009).

As explained in the motivation section of chapter 1, the problem of designing a suitable FOPID controller for a conventional multivariable process (integer order plant) is the focus of this work.

4.2 Realisation of Fractional order Terms by Approximation

Fractional order functions are not analytic functions and cannot be directly realisable (physically). To obtain a realisable controller function, some forms of integer order approximation methods are required. Continuous time models can be approximated using curve fitting, interpolation or some other forms of approximation. Considering discrete time models, continued fraction expansion or trapezoidal rule can be applied to obtain a rational approximation. For instance, consider an integral controller $C(s)$:

$C(s) = s^{-\mu}$; where $n-1 < \mu < n, n \in \mathbb{N}$.

It is not possible to find a finite order filter to fit the function for all frequencies since it is infinitely dimensional. However, a band limited implementation is feasible where a frequency range of interest can be defined for integer order approximation using suitable computational software tool like *Matlab*. Two common approximation methods are reviewed.

4.2.1 Continued Fraction Expansion (CFE)

This is an evaluation method that frequently converges more rapidly and in a larger complex plane's domain than power series expansion (Vinagre, et al., 2000). A major feature of CFE is that it inherently mirrors the poles of the system and directly yields rational functions from the approximation process. A function $G(s)$ may be represented in continued fraction expansion form as:

$$G(s) = a_0(s) + \cfrac{b_1(s)}{a_1(s) + \cfrac{b_2(s)}{a_2(s) + \cfrac{b_3(s)}{a_3(s) + \ldots}}}$$

where b and a functions are numerator and denominator terms of $G(s)$ respectively. So, $G(s)$ can be rewritten as:

$$G(s) = a_0(s) + \frac{b_1(s)}{a_1(s)} + \frac{b_2(s)}{a_2(s)} + \frac{b_3(s)}{a_3(s)} + \ldots$$

This way, an irrational $C(s)$ may be approximated with a rational function using CFE by considering the high frequency range and low frequency points only.

4.2.2 Oustaloup's Recursive Approximation (ORA) Method

Oustaloup's recursive approximation method is a frequency band limited approximation of the irrational function. It makes use of recursive distribution of poles and zeros to obtain an integer order function that approximately mimics the characteristic of the original fractional order function within the specified frequency range (ω_h, ω_l). For instance:

Given that $G(s) = s^{-\mu}, 0 < \mu < 1$, within the band (ω_h, ω_l), Oustaloup's approximation may be defined as follows:

$$H(s) = C\prod_{k=-N}^{N}\left(\frac{1+ s/\omega_k'}{1+ s/\omega_k}\right),$$

where gain $C = \left(\omega_u/\omega_h\right)^{\mu}$, $\omega_k = \left[\left(\omega_h/\omega_l\right)^{\frac{K+N+1/2+\mu/2}{2N+1}}\right]\omega_l$, $\omega_k' = \left[\left(\omega_h/\omega_l\right)^{\frac{K+N+1/2-\mu/2}{2N+1}}\right]\omega_l$,

N = Number of poles/zeros, ω_u = frequency at unit gain, ω_h = high transitional frequency limit, ω_l = low transitional frequency limit and μ = fractional order.

N is chosen reasonably before starting the approximation as it influences accuracy. The gain C is adjusted to achieve a unity gain at 1 rad/sec.

To use *Matlab* to implement ORA approximation, the band must be provided along with fractional order and expected number (*N)* of poles/zeros. These four parameters are used to approximate the fractional expression into an equivalent integer order form. This method is used predominantly in this work.

4.3 Preliminary Control Design Considerations

Frequency response data, step response and transfer function models contain information about dynamic characteristic of processes. Process knowledge is essential for formulation of control objectives and design of a FOPID controller capable of meeting predetermined design specifications. A wide range of plant is found in practice such as systems with fast and slow modes, integrating processes, open loop unstable systems, Right Half Plane (RHP) zero systems, time delay and lag systems as well as oscillatory processes. These are examples of operating conditions of real time plants. Although the list is not fully exhausted, it shows examples of complexities of control design problems typically encountered in real time systems.

Consequently, many algorithms have been developed over the years to design conventional PID controllers for various systems with varying degree of success. In some cases, meeting stringent performance specification means increased complexity of the resultant controller causing an increased computational cost. In this chapter, a method of designing FOPID controllers for controlling product compositions in a

typical distillation column is discussed. This is implemented using a simple feedback architecture.

The idea of implementing feedback in automatic control design is known to reduce sensitivity to uncertainties, model mismatch and noise. Model uncertainty and robustness have played key roles in the development of automatic control design technology. Many control design schemes have been developed over the years solely to improve robustness by reducing sensitivity to disturbances and parametric model variations. To illustrate this feedback property, consider a simple closed loop system (G) shown in Fig. 4.1 where input signals are given: R is reference signal; D is the load disturbance input and N is the measurement noise term. The output term is Y; C is the controller.

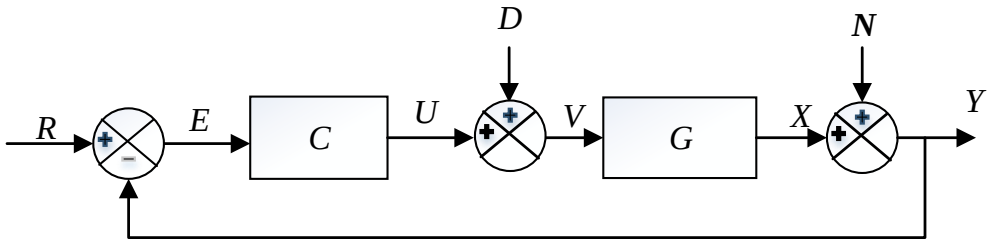


Fig. 4.1 A simple block diagram of a feedback control system

Considering all three inputs and two outputs (X and Y), four transfer functions ('s' is dropped for convenience) with significant effects can be identified:

- $X/R = \dfrac{CG}{1+CG}$; which is complementary sensitivity function $T(s)$;
- $X/E = \dfrac{G}{1+CG}$;
- $Y/N = \dfrac{1}{1+CG}$; which is sensitivity function $S(s)$;
- $U/R = \dfrac{C}{1+CG}$.

Assuming a small perturbation ∂G on the nominal plant G(s), the effect of feedback on such disturbances can be analysed by studying the sensitivity of the perturbed plant $G+\partial G$. In the open loop case, sensitivity is given by $S(s) = \dfrac{\partial G(s)/G(s)}{\partial G(s)/G(s)} = 1.$

When feedback is applied, we have a closed loop case. Sensitivity of this closed loop case is obtained as $\frac{\partial T(s)/T(s)}{\partial G(s)/G(s)} = \frac{G(s)}{T(s)} \cdot \frac{\partial T(s)}{\partial G(s)} = \frac{1}{1+C(s)G(s)}$.

Given that $C(s)G(s) \gg 1, \Rightarrow \text{Sensitivity} \ll 1.$ Therefore, sensitivity is highly reduced because of feedback. So, maximum values of both sensitivity and complementary sensitivity functions $\left(\max_{0<\omega<\infty} |S(j\omega)| \text{ and } \max_{0<\omega<\infty} |T(j\omega)| \right)$ are used as design parameters to define robustness' constraints.

Given that M_s = peak value of the system's sensitivity i.e. $\max_{0<\omega<\infty} |S(j\omega)|$ and peak value of the system's complementary sensitivity $M_p = \max_{0<\omega<\infty} |T(j\omega)|$, M_s can be deduced from Nyquist diagram as the reciprocal of the shortest distance between Nyquist plot and the critical point. The largest amount of admissible perturbation in a process is therefore given by the largest value of *Mp*. Setting a target for maximum allowable limit for these terms form a reasonable objective that can be used to derive robust controller parameters.

In addition, some control systems are designed to meet an additional requirement of robustness namely: insensitivity of phase to gain changes. This was first demonstrated by Bode's ideal transfer function. In Bode's system, changes in open loop gain introduces changes in gain cross over frequency but the phase margin remains constant over a given frequency range. Once this constraint is achieved, robustness of the system against gain variations is guaranteed within a specified interval. This is a very significant robustness feature because overshoot of system's response will remain constant within that interval. It can be challenging to design a PID controller to meet all these desirable objectives because some of these criteria are contradictory. For instance: $T(j\omega) + S(j\omega) = 1.$ Therefore, both terms cannot be simultaneously forced down to zero.

In scalar systems, it is sufficient to use open loop gain to define specification and impose constraints on the closed loop performance. Consequently, gain margin and phase margin can be easily defined for SISO systems as robustness constraints. In

multivariable systems, the concept of maximum singular values of sensitivity can be used to formulate closed loop performance requirement. There are preliminary challenges associated with formulation of control requirements for multivariable control systems just like conflicting objectives encountered in SISO control system design. These limitations are illustrated below:

- In feedback control, the open loop gain $C(s)G(s)$ is expected to be large enough to reject effects of disturbances and model mismatch. This reduces sensitivity to disturbances as $\frac{1}{1+C(s)G(s)} \ll 1.$ Within the lower frequency range, load disturbance and reference signals occupy that band implying that the maximum value of the sensitivity function $S(s)$ should be reduced to achieve good disturbance rejection. However, this increases complementary sensitivity value and can amplify measurement noise. The same scenario plays out in multivariable systems too. In MIMO systems, $\bar{\sigma}\left(\left[I+G(j\omega)C(j\omega)\right]^{-1}\right)$ is kept small but this increases $T(j\omega)$ which is linked to measurement noise.
- To meet any requirement on measurement noise attenuation, $(C(s)G(s))$ is kept as small as possible so that $\frac{C(s)G(s)}{1+C(s)G(s)} \approx \text{small}$. At high frequencies where measurement error (noise) is predominant, the maximum value of the complementary sensitivity function $T(s)$ should be reduced to attenuate high frequency noise. This conflicts with the first objective. In multivariable case, $\bar{\sigma}\left(I-\left[I+G(s)C(s)\right]^{-1}\right)$ is kept as low as possible. This also conflicts with the first objective.
- To achieve set-point tracking in MIMO systems, $\bar{\sigma}\left(I-\left[I+G(s)C(s)\right]^{-1}\right) \approx 1.$ This conflicts with the second requirement but not with the first objective.

It is evident that trade-off is required in formulating design specifications during control system design. Selection of appropriate gain margin, phase margin and sensitivity bounds can ensure an optimum performance and stability of the system in the presence of disturbance. This is not usually straightforward even in scalar systems.

In many applications, satisfaction of these design specifications usually implies an increase in the complexity of the controller thereby increasing cost. Many authors have shown that Fractional order PID controllers can meet these design objectives, although in SISO process control applications (Baruah, et al., 2016; Tepljakov, et al., 2018). This is due to the inherent frequency domain properties of FOPID controllers. Therefore, FOPID controller is proposed to solve some of these multivariable control system design problems because it retains the relatively simple structure of conventional PID controllers and produced better performance although at the expense of two extra tuning parameters. This way, improved control action can be obtained without losing the simplicity of a typical PID controller. FOPID controller is also found to be an attractive option for control design because some familiar PID controller tuning algorithms can be extended to design FOPID with very little modifications (Monje, et al., 2010; Vale´rio & Costa, 2009). This improves end user experience.

4.4 FOPID Controller Design: Review of Integral Gain Optimisation

Over the years, some optimisation-based techniques of designing conventional PID controllers have been extended to a class of fractional order controllers. Some of these optimisation methods are based on maximising robustness to parametric model variations, robustness to load disturbances and robust loop shaping. Integral gain optimisation is one of such ways of extending conventional PID design procedure to fractional order controllers. M_s constrained Integral Gain Optimisation (MIGO) method developed by Astrom, Panagopoulos and Hagglund is reviewed in this section (Astrom & Hagglund, 1995). The aim of the design is to obtain optimal values of controller parameters with the objective of minimising a sensitivity function (Sensitivity to load disturbance). This method is generally termed Fractional M_s Constrained Integral Gain Optimisation (F-MIGO) method (Monje, et al., 2010). Maximising the integral gain reduces effects of load disturbance at the output. Load disturbance effect can be quantified using Integral Squared Error (ISE).

$ISE = \int_0^\infty e(t)^2 dt$ where $e(t)$ is the error.

Due to contrasting nature of design objectives, an intermediate design parameter is formulated by imposing constraint on the peak values of both sensitivity functions.

Choosing M_s as the design parameter implies that the Nyquist plot of the transfer function must lie outside the M_s circle. Generally, the design parameter is chosen such that it encloses both the M_s circle and the M_p circle to achieve a design trade-off between these two terms. The centre of this circle C is given by:

$C = \frac{M_s - M_s M_p - 2M_s M_p^{\ 2} + M_p^{\ 2} - 1}{2M_s(M_p^{\ 2} - 1)}$ (Monje, et al., 2010). The radius of this circle is defined as R where: $R = \frac{M_s + M_p - 1}{2M_s(M_p^{\ 2} - 1)}$.

The optimal control design problem can therefore be stated as:

"*Maximising the integral gain to obtain the controller parameters such that the closed loop system is stable and the Nyquist plot of the loop's transfer function lies outside the circle with centre, C and radius, R.*" (Monje, et al., 2010)

M_s constrained Integral Gain Optimisation (MIGO) algorithm has been proven to yield optimal results for FOPDT plants. Optimal values of k_p, k_I and μ can be obtained using relevant optimisation routine to meet the specified sensitivity constraints in frequency domain i.e. $f(k_p, k_I, \omega, \mu) \geq R^2$ in *Matlab*. Finally, obtained FOPID function is realised using any approximation methods such as continuous fraction expansion or ORA method. This is primarily a SISO design method.

Other novel approaches to FOPID controller design include the CRONE control design narrative (Lanusse & Sabatier, 2011). The CRONE control design suite is a well-developed optimisation-based design method that gives optimal values of gains when you select frequency domain design criteria such as gain margin and phase margin. It has a well-developed user interface for objective selection and running of optimisation routine. However, CRONE method has the same limitation as the Fractional M_s Constrained Integral Gain Optimisation (F-MIGO) algorithm because it is limited to SISO control system design only. There are other optimisation routines that are directly based on pre-defined margins of stability such as frequency domain-based constrained optimisation.

4.5 Review of Constrained Optimisation Approach

This is another optimisation-based design method for FOPID controllers. The first step is to define design specifications expected to be satisfied by the fractional order controller. These design objectives are defined in frequency domain to take care of important control objectives: stability, performance and robustness. In practical control design and implementation, objectives are defined based on requirements of individual applications and can take any combination of these stated specifications. Controller parameters can thereafter be tuned to satisfy these design specifications.

4.5.1 Gain Margin and Phase Cross-over Frequency Specification.

Gain margin (A_m) is a primary index of relative stability in conventional control theory. A pre-defined margin can be used to formulate a robustness constraint on the system gain. Equation (4.1) defines the relationship between gain margin and phase cross over frequency. Equation (4.2) defines the gain margin constraint.

$$\left|C(j\omega_{cg})G(j\omega_{cg})\right|_{dB} = 0dB. \tag{4.1}$$

$$\left|C(j\omega_{cp})G(j\omega_{cp})\right|_{dB} = \frac{1}{A_m}, \tag{4.2}$$

where:

ω_{cg} represents the gain crossover frequency;

ω_{cp} represents the phase cross over frequency;

A_m represents the gain margin.

4.5.2 Phase Margin and Gain Cross over Frequency Specification

Phase margin and gain margin are extended to fractional order control theory as useful indices of robust stability. A controller can be designed to satisfy the phase margin condition given in (4.3).

$$\arg(C(j\omega_{cg})G(j\omega_{cg})) = -\pi + \phi_m, \tag{4.3}$$

where ϕ_m is the phase margin and 'arg(.)' represents argument.

4.5.3 *Constant Phase with Plant's Gain Variations*

Bode's ideal loop defines the criteria for absolutely stable SISO loop. If there is constant phase around the crossover frequency (useful band), robust stability against gain variations is guaranteed within that frequency range in spite of gain variations.

$$\frac{\partial \arg\left\{C(j\omega_{cg})G(j\omega_{cg})\right\}}{\partial \omega_{cg}} = 0. \tag{4.4}$$

If (4.4) is satisfied, the phase of the forward loop function will be flat around the cross over frequency. This improves robustness against gain-like variations in plant and the overshoot is nearly constant within that frequency range.

4.5.4 *High frequency noise rejection specification*

In order to ensure satisfactory measurement noise rejection, appropriate bound for complementary sensitivity function has to be defined:

$$\left\|T(j\omega)\right\|_{dB} \leq AdB \tag{4.5}$$

$$\left\|\frac{C(j\omega)G(j\omega)}{1+C(j\omega)G(j\omega)}\right\|_{dB} \leq AdB \tag{4.6}$$

for $\omega \geq \omega_t$ rad/s where $T(j\omega)$ is the complementary sensitivity function and A dB is the specified attenuation level in dB for the band $\omega \geq \omega_t$ rad/s.

4.5.5 *Output disturbance rejection specification*

Sensitivity constraint can be defined to ensure satisfactory output disturbance rejection by the controller.

$$\left\|\frac{1}{1+C(j\omega)G(j\omega)}\right\|_{dB} \leq B \text{ dB}, \tag{4.7}$$

where B dB = the specified magnitude of the sensitivity function in dB for the band $\omega \geq \omega_s$ rad/s.

The controller parameters can then be calculated and optimized to meet any combination of these design specifications. Therefore, more parameters are available for optimisation to meet more control objectives.

For instance, Monje (2010) used *Matlab*'*s* optimisation tool (fmincon) to formulate an optimisation problem that solves for these five parameters and it was a constrained non-linear optimisation problem with five unknown variables. This is primarily a SISO method. Therefore, it is desirable to develop a design procedure for FOPID controller that is suitable for multivariable process control systems.

Another major problem encountered when using the current method is that the optimisation routine requires suitable initial settings without offering a means of identifying it. A simpler way of obtaining robust values for FOPID controller gains is to use sustained oscillation experiment. The fractional order for the integral controller term can be chosen to satisfy robust performance criteria. In the case of a time delayed process, relative deadtime provides a guide for selecting fractional order and this is tabulated in Table 4.1 and referred to in all examples. This table is relevant because, considering first-order processes with dead-time, appropriate values of fractional orders to counteract effects of such time delays have been experimentally determined (Monje, et al., 2010). It can also be easily extended to multivariable applications. The proposed method is presented in the next section.

4.6 FOPID Controller Design using Critical Point Parameters

A decentralised FOPID control design method is developed based on critical frequency point information. The control scheme is suitable for SISO process control but it can also be extended to MIMO processes in a decentralised or multi-loop configuration. Ziegler-Nichols type rules are still dominant for tuning PID controllers for simple first order or second order process models. In this section, a similar technique is developed for designing FOPID controllers suitable for process control applications. The experiment is set up by using a proportional controller in cascade with a plant under a closed loop configuration. Thereafter, proportional gain is systematically increased from very small values until sustained oscillation or continuous cycling is observed at the process output. The value of the proportional gain that yields this sustained oscillation is recorded as the critical gain (k_u) while the period of oscillation is noted as ultimate period $\left(\tau_u\right)$. Relay feedback method can also be used to identify critical frequency point without running the plant up to the verge of instability.

Several other important information about the process are obtainable from these critical point measurements. For instance, the phase cross over frequency (ω_{pco}) can be obtained as: $\omega_{pco} = 2\pi/\tau_u$. If a Fractional Order PI controller (FOPI) structure given in (4.8) is chosen for process control, appropriate gains can be derived from critical point information.

$$C(j\omega) = k_p + \frac{k_I}{(j\omega)^{\lambda}}, \tag{4.8}$$

where $k_I = k_p/\tau_i$.

FOPI controller can be obtained by relocating this ultimate frequency point on the complex plane to a more desirable point. This is illustrated on the Nyquist diagram of *Fig. 4.2.* The critical point or ultimate point is the stability limit signified by the point where the Nyquist curve intersects the unit circle (-1, 0).

4.6.1 Derivation of Gains for FOPI Controller

Consider the FOPI controller given in (4.8), a desirable point on Nyquist curve is chosen as point *B* given by: $r_B e^{j(\pi+\varnothing_B)}$ where $(r_B, \varnothing_B)$ are magnitude and phase at the chosen point *B*. The controller is expected to move the ultimate point $(1/k_u, 0)$ to this desirable point. On the Nyquist plot, the ultimate point is point *A* $\left(r_A e^{j(\pi+\varnothing_A)}\right)$.

$$\varnothing_A = 0;\ r_A = \frac{1}{k_u}; \tag{4.9}$$

where $(r_A, \varnothing_A)$ are magnitude and phase at the critical point *A*.

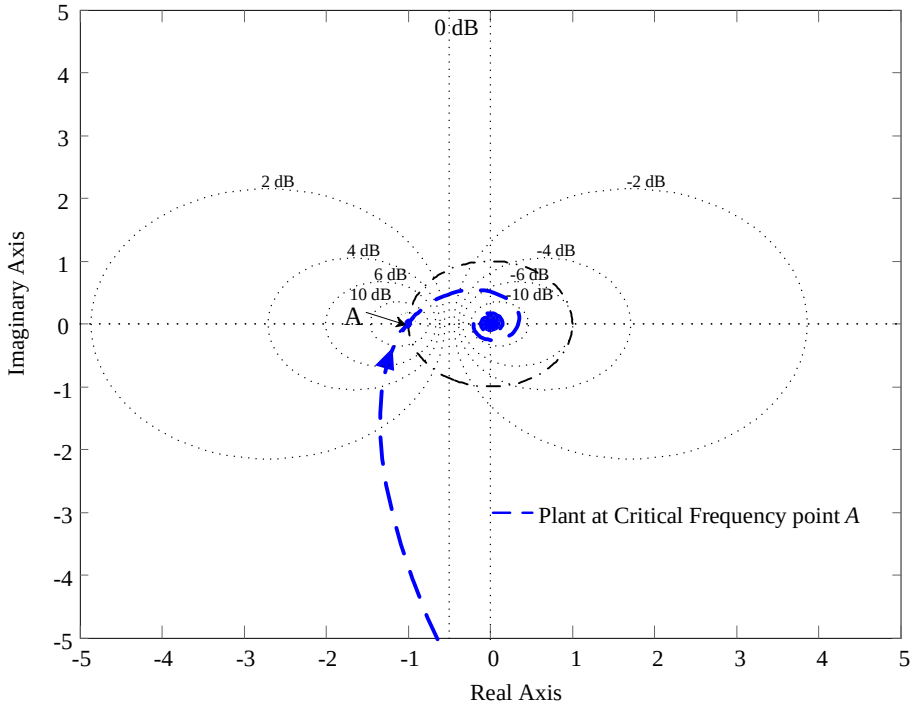


Fig. 4.2 Nyquist diagram of SISO system showing critical frequency point

Let the frequency characteristic of the controller C(s) be $r_C e^{j(\varnothing_C)}$ where r_c and $\varnothing_c$ are magnitude and phase of controller *C*(*s*). This controller is expected to relocate point *A* to point *B* implying that (4.10) holds.

$$r_A r_C e^{j(\pi+\varnothing_C+\varnothing_A)} = r_B e^{j(\pi+\varnothing_B)} \tag{4.10}$$

$$r_C e^{j(\pi+\varnothing_A)} e^{j\varnothing_c} = \frac{r_B}{r_A} e^{j(\pi+\varnothing_B)} \tag{4.11}$$

$$r_C e^{j\varnothing_c} = \frac{r_B}{r_A} e^{j(\pi+\varnothing_B-\varnothing_A-\pi)} \tag{4.12}$$

Substituting (4.9) in (4.12) and applying Euler's formula yields:

$$r_C e^{j(\varnothing_C)} = \frac{r_B}{r_A}(\cos\varnothing_B + j\sin\varnothing_B) \tag{4.13}$$

$$r_C e^{j(\varnothing_C)} = r_B k_u(\cos\varnothing_B + j\sin\varnothing_B)$$

Therefore, the controller should be chosen to satisfy these two conditions below:

$$r_C = r_B / r_A \tag{4.14}$$

$$\varnothing_C = \varnothing_B - 0 \tag{4.15}$$

In frequency domain, $C(s)$ controller structure can be expressed as follows:

$$C(j\omega) = k_p + \frac{k_I}{(j\omega)^{\lambda}},\ C(j\omega) = k_p\left(1 + \frac{1}{\tau_I (j\omega)^{\lambda}}\right).$$

To rationalize this function, consider the complex operator:

$$j^{-\lambda} = \cos\frac{\lambda\pi}{2} - j\sin\frac{\lambda\pi}{2}.$$

So: $C(j\omega) = k_p\left(1 + \frac{1}{\omega^{\lambda}\tau_I}(\cos\frac{\lambda\pi}{2} - j\sin\frac{\lambda\pi}{2})\right)$ (4.16)

$$\Rightarrow k_p\left(1 + \frac{1}{\omega^{\lambda}\tau_I}(\cos\frac{\lambda\pi}{2} - j\sin\frac{\lambda\pi}{2})\right) = r_B k_u(\cos\varnothing_B + j\sin\varnothing_B). \tag{4.17}$$

By comparing real part of (4.17), proportional gain can be obtained as shown below:

$$r_B k_u \cos\varnothing_B = k_p\left(1 + \frac{1}{\omega^{\lambda}\tau_I}(\cos\frac{\lambda\pi}{2})\right) \tag{4.18}$$

$$k_p = \frac{r_B k_u \cos\varnothing_B}{1 + \left(\frac{\tau_u}{2\pi}\right)^{\lambda}\tau_I \cos\left(\frac{\lambda\pi}{2}\right)} \tag{4.19}$$

In the same vein, imaginary part of (4.17) is compared:

$$\varnothing_B = \tan^{-1}\left(\frac{-\frac{k_p}{\omega^{\lambda}\tau_I}\sin\frac{\lambda\pi}{2}}{k_p\left(1 + \frac{1}{\omega^{\lambda}\tau_I}(\cos\frac{\lambda\pi}{2}\right)}\right)$$

$$\tan\varnothing_B = \frac{-\omega^{-\lambda}\tau_I^{-1}\sin\frac{\lambda\pi}{2}}{(1 + \omega^{-\lambda}\tau_I^{-1}\cos\frac{\lambda\pi}{2})} \tag{4.20}$$

To solve for integral gain:

$$\tau_I = \frac{-\omega^{-\lambda}\left(\tan\varnothing_B \cos\frac{\lambda\pi}{2} + \sin\frac{\lambda\pi}{2}\right)}{\tan\varnothing_B}\ ;$$
$$k_I = \frac{-k_p \tan\varnothing_B}{\omega^{-\lambda}\left(\tan\varnothing_B \cos\frac{\lambda\pi}{2} + \sin\frac{\lambda\pi}{2}\right)}. \qquad (4.21)$$

4.6.2 Justification of Design Point

In general, the desired point *B* is chosen such that:

$r_B = 0.29;\ \varnothing_B = 46°$.

This is a very desirable point because it translates to -0.2 - *j*0.21 point on the complex plane. The distance (*d*) from this point to the stability limit (-1, *j*0) is given as d where $d = \sqrt{(-0.2+1)^2 + (-0.21)^2}$. This distance is approximately 0.9. Closed loop sensitivity can be deduced from this information on the Nyquist curve of the forward loop because the maximum sensitivity M_s is the reciprocal of the shortest distance from the Nyquist curve to the critical point. On Nyquist diagram, $M_s = 1/d$.

This is desirable for robustness. For the closed loop system to be robust against variations in process dynamics, maximum sensitivity of the closed loop system is specified as: $M_s < 2$. Reasonable values of M_s ranges from 1.05 to 1.95. This indicates good robustness for many practical control applications and it is therefore recommended in this work as the design point as far as critical frequency point procedure is concerned. In terms of closed loop stability, it meets Nyquist stability criterion as the critical stability (-1, *j*0) point will not be encircled. Therefore, the fractional order PI controller can be obtained using (4.22) and (4.23).

$$k_p = \frac{0.202k_u}{1+\left(\tau_u/2\pi\right)^{\lambda}\tau_I \cos\left(\lambda\pi/2\right)} \qquad (4.22)$$

Fractional order is selected from Table 4.1. Given a time delayed system *G*(*s*):

$G(s) = \frac{k_P e^{-Ls}}{s\tau_p + 1}$; relative dead time ($T$) is expressed as follows $T = \frac{L}{(L + \tau_P)}$,

where L = dead time, k_P = process steady state gain, τ_p = process time constant.

The relative dead time provides a guide for selection of fractional order as shown in Table 4.1 (Monje, et al., 2010).

Table 4.1 *Relative dead time and fractional-order*

	Relative Dead Time (T)	Fractional Order
Selection of Fractional Order	T < 0.1	0.7
	0.1 ≤ T < 0.4	0.9
	0.4 ≤ T < 0.6	1.0
	T ≥ 0.6	1.1

$$|\tau_I| = 0.966\left((\frac{\tau_u}{2\pi})^{\lambda} (\sin\frac{\lambda\pi}{2} + 1.036\cos\frac{\lambda\pi}{2}) \right) \quad (4.23)$$

For Second order processes, derivative component can be included for a more damped response. One possible solution is to design the fractional order controller using a $PI^{\mu}D$ controller structure. It is however tuned using the same method as in the fractional order PI controller case i.e. using ultimate parameters.

4.6.3 The $PI^{\lambda}D$ Controller Structure - Derivation of Gains

The structure of the controller is given as:

$$C(j\omega) = k_p + \frac{k_I}{(j\omega)^{\lambda}} + k_d(j\omega) \quad (4.24)$$

Let the controller function $C(j\omega)$ be characterised by $r_C e^{j(\varnothing_C)}$ in frequency domain and desirable point on Nyquist plane chosen as point B (i.e. $r_B e^{j(\pi+\varnothing_B)}$) as described in the

preceding section. The controller is expected to move the ultimate point $(1/k_u, 0)$ to this desirable point as given before. On the complex plane, the ultimate point is point $A(r_A e^{j\pi})$. Therefore (4.10), (4.14) and (4.15) must be fulfilled as earlier shown. Applying (4.10) to this controller yield (4.25) below:

$$r_c e^{j\varnothing_c} = \frac{r_B}{r_A}\left(\cos\varnothing_B + j\sin\varnothing_B\right) \tag{4.25}$$

$$C(j\omega) = k_p\left(1 + \frac{1}{\tau_I\left(\cos\frac{\lambda\pi}{2} + j\sin\frac{\lambda\pi}{2}\right)} + \tau_d(j\omega)\right) \tag{4.26}$$
$$= r_B k_u\left(\cos\varnothing_B + j\sin\varnothing_B\right)$$

Equation (4.26) can be re-arranged to compare real part:

$$k_p\left(1 + \frac{1}{\omega^{\lambda}\tau_I}\cos\frac{\lambda\pi}{2}\right) = r_B k_u\cos\varnothing_B \tag{4.27}$$

$$k_p = \frac{r_B k_u\cos\varnothing_B}{\left(1 + \frac{1}{\omega^{\mu}\tau_I}\cos\frac{\lambda\pi}{2}\right)}. \tag{4.28}$$

Comparing imaginary part of (4.26):

$$k_p\left(\omega\tau_d - \frac{1}{\omega^{\lambda}\tau_I}\sin\frac{\lambda\pi}{2}\right) = r_B k_u\sin\varnothing_B. \tag{4.29}$$

Substituting (4.28) in (4.29) yields:

$$\tan(\varnothing_B) = \frac{\omega\tau_d - \frac{1}{\omega^{\lambda}\tau_I}\sin\frac{\lambda\pi}{2}}{1 + \frac{1}{\omega^{\lambda}\tau_I}\cos\frac{\lambda\pi}{2}}. \tag{4.30}$$

Solving for integral time yields:

$$\tau_I = \frac{\tau_u^{\lambda}\left(\tan\varnothing_B \cos\frac{\lambda\pi}{2} + \sin\frac{\lambda\pi}{2}\right)}{(2\pi)^{\lambda}\left(\frac{2\pi\tau_d}{\tau_u} - \tan\varnothing_B\right)}. \tag{4.31}$$

Given that the order μ is obtained as described in previous section using Table 4.1, there are now three parameters left to be calculated from two equations. One way to solve this simultaneous equation is to treat the ratio of τ_d to τ_I as a constant (Astrom & Hagglund, 2006). For instance, set $\tau_d = 0.25\tau_I$.

At point *B*: $r_B = 0.29, \varnothing_B = 46°$.

$$\tau_I = \frac{\tau_u^{\lambda}\left(1.036\cos\frac{\lambda\pi}{2} + \sin\frac{\lambda\pi}{2}\right)}{(2\pi)^{\lambda}\left(\frac{2\pi\tau_d}{\tau_u} - 1.036\right)}, \; k_p = \frac{0.202k_u}{\left(1 + \frac{\tau_u^{\lambda}}{(2\pi)^{\lambda}\tau_i}\cos\frac{\lambda\pi}{2}\right)}.$$

All four parameters can therefore be calculated.

$$\tau_I = \frac{\tau_u^{\lambda}\left(1.036\cos\frac{\lambda\pi}{2} + \sin\frac{\lambda\pi}{2}\right)}{(2\pi)^{\lambda}\left(\frac{2\pi\tau_d}{\tau_u} - 1.036\right)} \tag{4.32}$$

$$k_p = \frac{0.202k_u}{\left(1 + \frac{\tau_u^{\lambda}}{(2\pi)^{\mu}\tau_I}\cos\frac{\lambda\pi}{2}\right)} \tag{4.33}$$

The second structure of FOPID controller with fractional order derivative term is of PID^{μ} form.

4.6.4 The PID^{μ} Controller Structure – Derivation of Gains

The structure of PID^{μ} controller is given by (4.36):

$$C(j\omega) = k_P + \frac{k_I}{j\omega} + k_d(j\omega)^{\mu} \tag{4.34}$$

$$C(j\omega) = k_p\left(1 - j\frac{1}{\tau_I\omega} + \tau_d(j\omega)^\mu\right) \tag{4.35}$$

As described in preceding section, this controller is expected to move the ultimate point $(1/k_u, 0)$ to the new design point B (i.e. $r_B e^{j(\pi+\varnothing_B)}$) implying that required conditions stated in (4.10), (4.14) and (4.15) must be met. Applying these equations as before and comparing real parts of controller yields:

$$k_P\left(1+\omega^\mu\tau_d\cos(\mu\pi/2)\right) = \frac{r_B}{r_A}(\cos\varnothing_B - \varnothing_A)$$

$$k_P = \frac{\left(\frac{r_B}{r_A}\cos(\varnothing_B - \varnothing_A)\right)}{\left(1+\omega^\mu\tau_d\cos(\mu\pi/2)\right)} \tag{4.36}$$

$$k_P = \frac{\left(r_B k_u\cos(\varnothing_B - \varnothing_A)\right)}{\left(1+\omega^\mu\tau_d\cos(\mu\pi/2)\right)} \tag{4.37}$$

Considering imaginary part:

$$k_p\left(\omega^\mu\tau_d\sin(\mu\pi/2) - 1/\omega\tau_I\right) = r_B k_u\sin\varnothing_B \tag{4.38}$$

$$\tan\varnothing_B = \frac{\left(\omega^\mu\tau_d\sin(\mu\pi/2) - (1/\omega\tau_I)\right)}{\left(1+\omega^\mu\tau_d\cos(\mu\pi/2)\right)} \tag{4.39}$$

$$\tau_d = \frac{\tan\varnothing_B + \left(\tau_u/\tau_I 2\pi\right)}{\left(2\pi/\tau_u\right)^\mu\left(\sin(\mu\pi/2) - \cos(\mu\pi/2)\tan\varnothing_B\right)} \tag{4.40}$$

The ratio of integral time to derivative time is assumed to be constant i.e. $\tau_D = 0.25\tau_i$.

Equations (4.37) and (4.40) can be used to calculate these two controller gains. Fractional order is selected using relative dead time guide as explained in previous cases.

At point *B*: $r_B = 0.29; \varnothing_B = 46°$. These controller settings can further be summarized as (4.41) and (4.42) while integral time is taken as 4 times the value of derivative time.

$$k_P = \frac{0.202k_u}{\left(1+\left(2\pi/\tau_u\right)^{\mu}\tau_d\cos(\mu\pi/2)\right)} \tag{4.41}$$

$$\tau_d = \frac{1.036+\left(\tau_u/\tau_I 2\pi\right)}{\left(2\pi/\tau_u\right)^{\mu}\left(\sin(\mu\pi/2)-1.036\cos(\mu\pi/2)\right)} \tag{4.42}$$

4.7 Simulation Example 1

A simple FOPDT process is used for preliminary tests. The system is given as:

$$G(s) = \frac{12.8e^{-s}}{16.7s+1}$$

Three controllers C_1, C_2 and C_3 are designed with the following structures: PI^{λ}, $PI^{\lambda}D$ and PID^{μ} controllers respectively. This is done using the sustained oscillation method as described in this work. Obtained controllers are given as follows:

$$C_1 = 0.04\left(1+\frac{1}{1.12s^{0.7}}\right), C_2 = 0.32\left(1+\frac{0.1}{s^{0.7}}+0.35s\right),$$
$$C_3 = 0.218\left(1+\frac{1}{3.6s}+1.08s^{0.7}\right).$$

Fig. 4.4 - Fig. 4.6 show Nyquist diagram of each of these three control systems. The integral term $s^{-0.7}$ is implemented as $s^{0.3}.\ s^{-1}$ using Oustaloup Recursive Approximation (ORA) in order to obtain integer order form of the controller with similar frequency domain properties. It is observed that the controller moves the design point of the plant to a desired (stable) region. The step response of these controllers is compared in Fig 4.3.

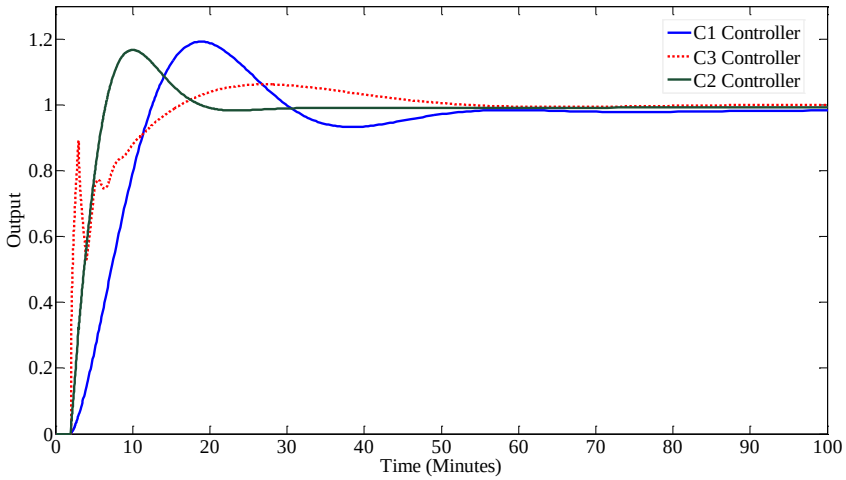


Fig. 4.3 Step response of FOPID controllers

In all three examples, notice the blue dotted curve crosses the (-1, *j*0) point showing that the process was at the verge of instability without controller. When simulating the entire feedback control system with FOPID controller, the green curve moves away from the point of instability and good margins of stability are achieved.

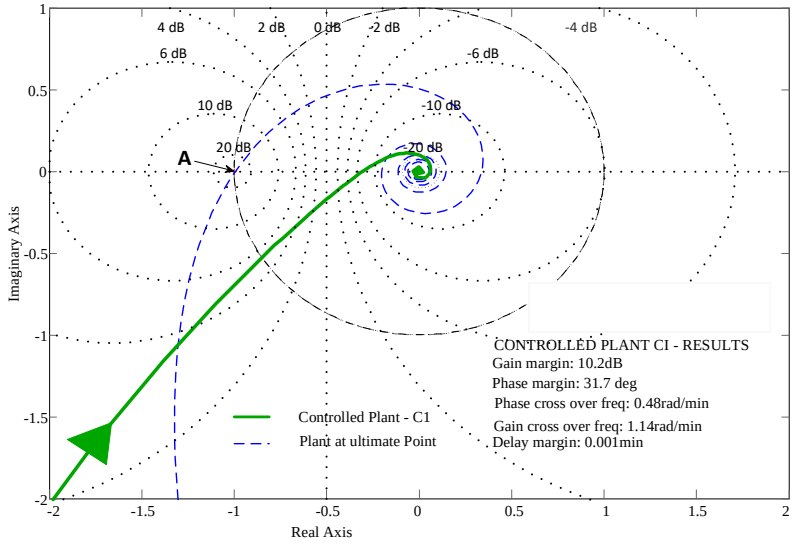


Fig. 4.4 Nyquist Diagram of FOPDT Plant with controller C1

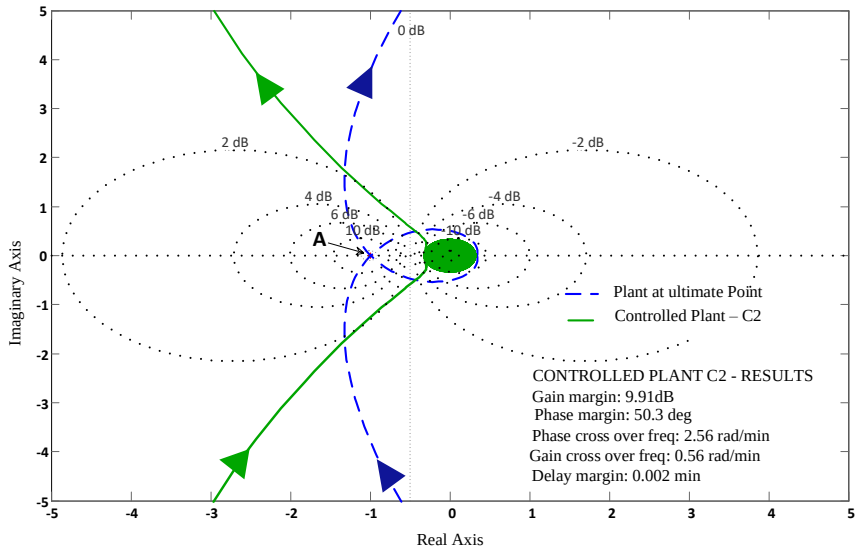


Fig. 4.5 Nyquist diagram of FOPDT plant with controller C2.

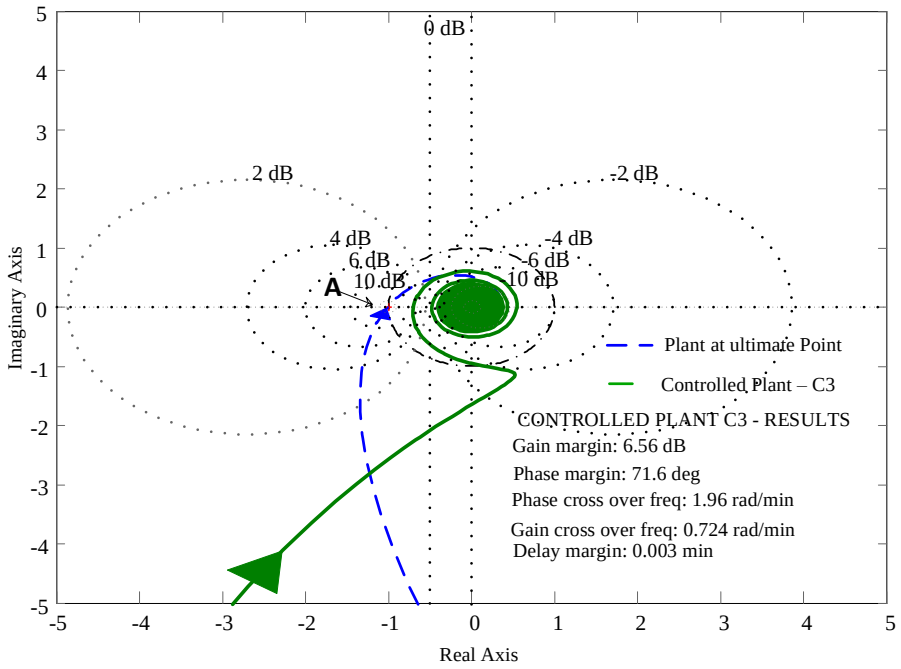


Fig. 4.6 Nyquist diagram of FOPDT plant with controller C3

Consider an integrating process such as the SOPDT example below:

$$G_2(s) = \frac{2e^{-0.2s}}{s(0.5s+1)}$$

Controllers (C_{22} and C_{11}) are designed with the following structures: $PI^{\lambda}D$ and PID^{μ} respectively. Results of simulations are shown in Bode diagram of Fig. 4.8, sensitivity diagram of Fig. 4.7 and step response diagram of Fig. 4.9.

$$C_{11} = 0.444 + (0.175/s) + 0.3s^{0.7}, C_{22} = \left(0.459 + \frac{0.2}{s^{0.7}} + 0.28s \right).$$

Consider the sensitivity M_s and complementary sensitivity T_s function:

$$M_s = \max \left| \frac{1}{1 + C(j\omega)G(j\omega)} \right|, 0 \le \omega \le \infty.$$

$$T_s = \max \left| \frac{C(jw)G(jw)}{1 + C(j\omega)G(j\omega)} \right|, 0 \le \omega \le \infty.$$

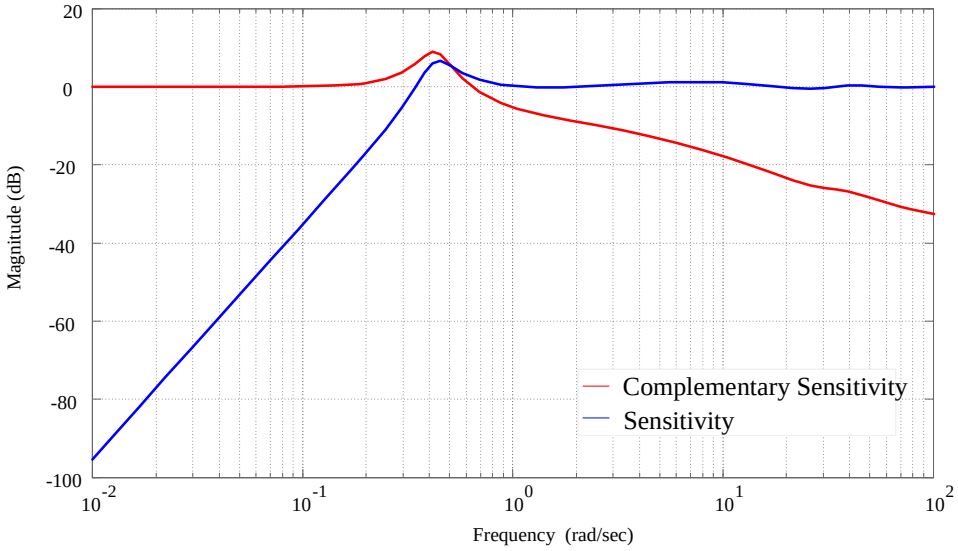


Fig. 4.7 Sensitivity diagram of integrating process with controller C11

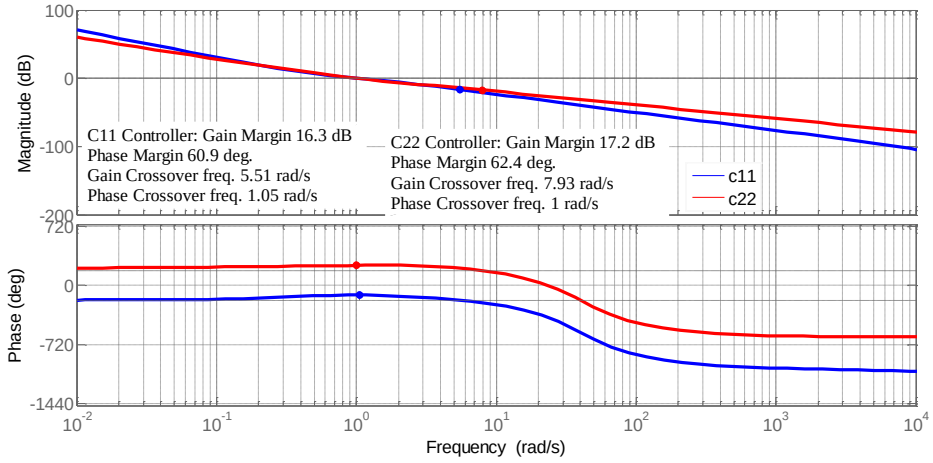


Fig. 4.8 Bode diagram of integrating process with controllers

Over a wide range of frequency, sensitivity and complementary sensitivity are illustrated in *Fig*. 4.7. These SISO based methods yield good response for the models

considered as reflected in gain margin, phase margin and delay margin for each control system.

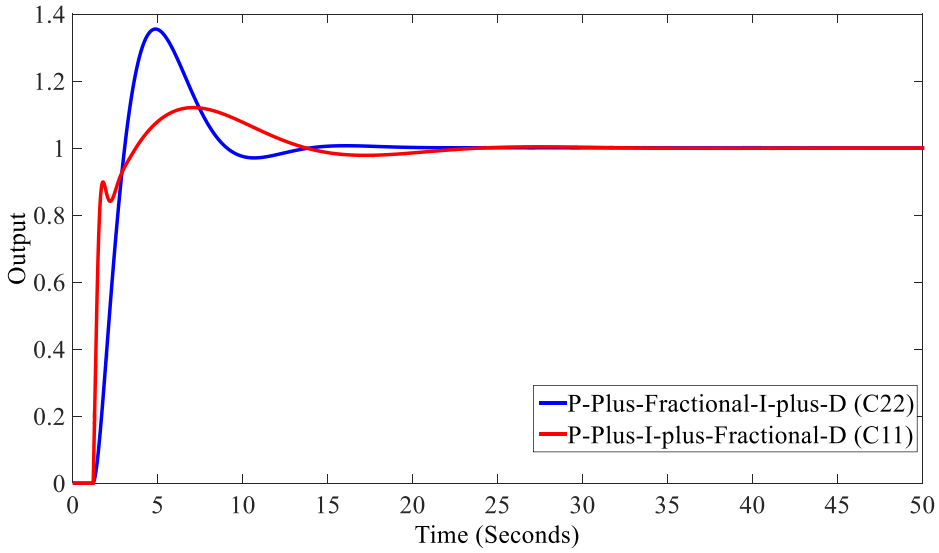


Fig. 4.9 Step response diagram of integrating process

Summary of Proposed Algorithm for SISO processes

- Obtain critical frequency point parameters using sustained oscillation or relay experiment.
- Record critical gain (k_u) and the period of oscillation (τ_u).
- For first order processes with delay, select fractional order using Table 4.1. For SOPDT models, desired fractional order can be selected to shape the frequency curve.
- For FOPI controller, use (4.23) and (4.22) to get τ_I and k_P respectively.
- For second order processes with delay, if the PID^{λ} controller structure is chosen, derivative gain can be obtained using (4.42). k_P is given by (4.41) and $\tau_I = 4\tau_d$.
- For $PI^{\lambda}D$ controller, integral gain can be obtained using (4.32) and $\tau_d = 0.25\tau_I$. k_P is given by (4.33).

In addition to good margins of stability, very little computation is required to yield desired fractional order controller. Therefore, it is extended to a more complicated process like multivariable distillation system.

4.8 Sequential Loop Closing Technique

To extend sustained oscillation method to MIMO plant control, each loop of the entire MIMO system should be treated independently using sequential loop closing technique. Sequential loop closing method has been used to design conventional PID controllers for MIMO system successfully. In conventional PID/MIMO scenario, the fastest loop is designed first using sustained oscillation experiment and after computing controller gains for the first loop, it is closed before setting up the same experiment in the next loop. This principle is extended to tune fractional order PID controllers in a multivariable setting. The difference is in the inclusion of a suitable fractional order selected for each loop using relative dead time guide or to meet M_r condition. One major advantage of this method is that very little computations are involved.

The technique handles interaction effectively if input and output variables are effectively paired and the system has limited number of loops. The first step is to design the fastest loop of the system independently before closing remaining loops sequentially. When designing controllers by sequential loop closing technique in TITO systems or systems with three inputs and three outputs, significant iteration may not be needed before convergence is reached. Each sub-transfer function of the model is also assumed to be open loop stable. Many processes in practice are found to be open loop stable (Loh, et al., 1993).

In terms of overall system stability, if the feedback control of each loop with the proportional controller produces a sustained oscillation at every stage of loop closing and ultimate point parameters are correctly measured, stability is established. The algorithm is very simple and offers very easy to use method of computing FOPID controller gains for MIMO systems. However, there could be a challenge of identifying the accurate critical frequency point of the plant or the experimentation process may take too much time than desired in some applications.

Consider a three-input three-output process as follows:

$$G(s)=\begin{pmatrix} G_{11}(s) & G_{12}(s) & G_{13}(s) \\ G_{21}(s) & G_{22}(s) & G_{23}(s) \\ G_{31}(s) & G_{32}(s) & G_{33}(s) \end{pmatrix}, G_{ij}(s)=\frac{k_{P,ij}e^{-L_{ij}s}}{\tau_{P,ij}s+1};$$

where $i, j = 1,2,3; k_P =$ steady state gain; $\tau_P =$ time constant; L=dead-time, proposed MIMO FOPI controller for this process is of the form C(s):

$$C(s)=\begin{bmatrix} C_1(s) & 0 & 0 \\ 0 & C_2(s) & 0 \\ 0 & 0 & C_3(s) \end{bmatrix}; \; C_j(s)=k_{p,j}\left[1+\frac{1}{\tau_{I,j}s}\right]; \; j=1,2,3;$$

where $k_p =$ proportional gain and $\tau_I =$ integral time.

Summary of proposed tuning procedure of FOPID controllers for MIMO processes

- Consider diagonal transfer function elements only. Choose the fastest diagonal element as loop 1. Close the loop with a proportional controller to obtain critical frequency point parameters via sustained oscillation experiment. Keep the other loops open. Critical frequency point can also be identified by connecting a relay using relay feedback method.
- Record critical gain (k_u) and the period of oscillation (τ_u) of loop 1.
- Tune loop one using the proposed algorithm given in SISO section i.e. for first order processes with delay, select fractional order using Table 4.1. For SOPDT models, use peak resonant equation to obtain suitable fractional order for frequency loop shaping.
- Use (4.23) and (4.22) to get integral time and proportional gain respectively.
- For second order processes with delay, derivative gain can be obtained using (4.42). k_p is given by (4.41). $\tau_I = 4\tau_d$ and PID^{μ} controller is realized.
- If $PI^{\lambda}D$ controller structure is considered, integral gain can be obtained using (4.32) and $\tau_d = 0.25\tau_I$. k_p is given by (4.33).
- Replace the proportional controller with the tuned fractional controller. Keep it closed and chose the next loop for the same sustained oscillation experiment.

- Tune the selected loop in the same way as loop one and close it by replacing the proportional controller with the newly designed fractional controller.
- Proceed in the same manner until all the loops are designed and closed.

Design of FOPI controller for *OR* distillation column is now considered as an example to demonstrate the procedure. The model is shown below.

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \frac{0.66e^{-2.6s}}{6.7s+1} & \frac{-0.61e^{-3.5s}}{8.64s+1} & \frac{-0.0049e^{-s}}{9.06s+1} \\ \frac{1.11e^{-6.5s}}{3.25s+1} & \frac{-2.36e^{-3s}}{5s+1} & \frac{-0.012e^{-1.2s}}{7.09s+1} \\ \frac{-34.68e^{-9.2s}}{8.15s+1} & \frac{46.2e^{-9.4s}}{10.9s+1} & \frac{0.87(11.61s+1)e^{-s}}{(3.89s+1)(18.8s+1)} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} +$$

$$\begin{pmatrix} \frac{0.14e^{-12s}}{(6.2s+1)} & \frac{-0.011(26.32s+1)e^{-2.66s}}{(7.85s+1)(14.63s+1)} \\ \frac{0.53e^{-10.5s}}{(6.9s+1)} & \frac{-0.0032(19.62s+1)e^{-3.44s}}{(7.29s+1)(8.94s+1)} \\ \frac{-11.54e^{-0.6s}}{(7.01s+1)} & \frac{0.32e^{-2.6s}}{(7.76s+1)} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

Each term is as defined previously in (3.12). To design a fractional order PI controller for this column, the first loop (fastest loop in the system based on time constant) is considered. This contain the last diagonal element which is a second order transfer function. That is:

$$\frac{y_3}{u_3} = \frac{0.87(11.61s+1)e^{-s}}{(3.89s+1)(18.8s+1)}$$

A proportional controller is connected and sustained oscillation experiment is set up. Loop 1 is closed as given in the algorithm. It should be noted that the SOPDT is first approximated to a FOPDT model for comparison purpose only in order to determine which loop to tune first. Obtained parameters are: $k_u = 12.54;\ \tau_u = 5.$ Fractional order is selected as 0.9 as described in previous sections. The controller gains are calculated using (4.22) and (4.23): $k_p = 2.2; k_I = 2.4.$ Thereafter, the proportional controller is replaced with the designed FOPI controller in loop 1. The other two loops are left open while designing the first loop.

Shortly after closing the first loop with the FOPI controller, sustained oscillation experiment is set up in loop 2. This is the loop which directly contains the second diagonal element. A proportional controller is directly connected and the value is slightly increased from small values until sustained oscillation is observed at the process output y_2.

$$\frac{y_2}{u_2} = \frac{-2.36e^{-3s}}{5s+1}$$

Obtained parameters are: k_u = -1.241; τ_u = 16. Also, 0.9 is selected as fractional order. The third loop is still open but the second loop is tuned to get: k_p = -0.13; k_I = -0.1. Thereafter, a proportional controller is connected and sustained oscillation experiment is set up in the third loop. The third loop contains the first diagonal element.

$$\frac{y_1}{u_1} = \frac{0.66e^{-2.6s}}{6.7s+1}$$

Obtained parameters are: k_u = 5.58; τ_u = 19. The same fractional order of 0.9 is selected. The other two loops are left closed and loop three is tuned to get: k_p = 0.65; k_I = 0.2. Table 4.2 shows the proposed controller (FOPI) and the gains of the base line Optimum PI controller (OPI). In summary, after designing all three diagonal controllers, a set-point change (unit step) is introduced in loop 1 with all loops closed.

Fig. 4.10 and Fig. 4.11 illustrate responses in all three loops because of a step change in $r1$. The corresponding output $y1$ rises to one while the controller forces other outputs in loops two and three to zero as expected.

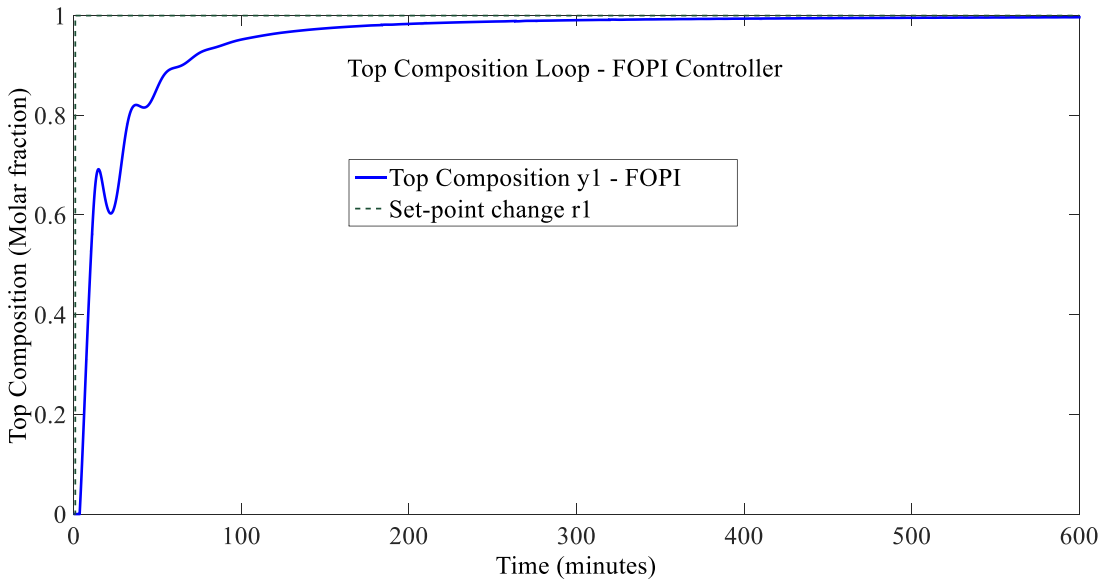


Fig. 4.10 Step change in r1 - top composition loop (y1)

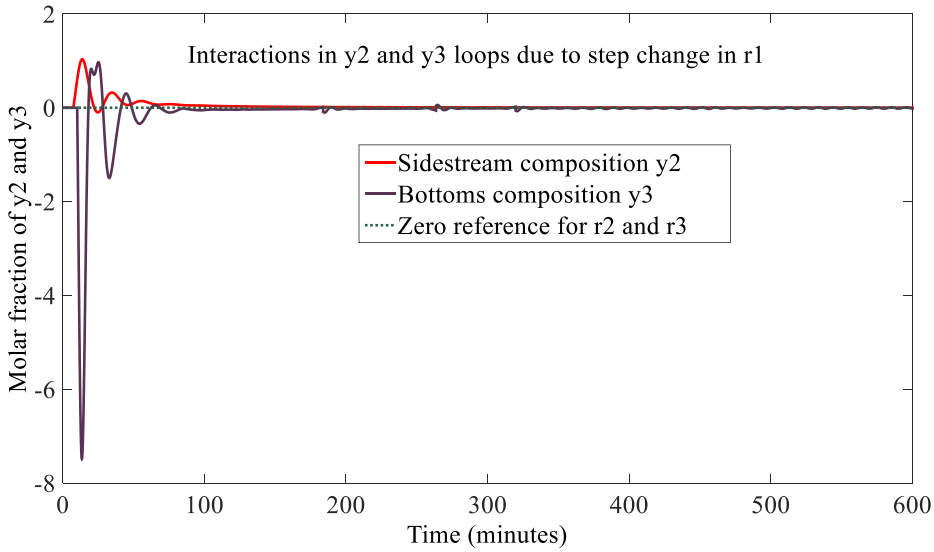


Fig. 4.11 Interactions due to step change in r1

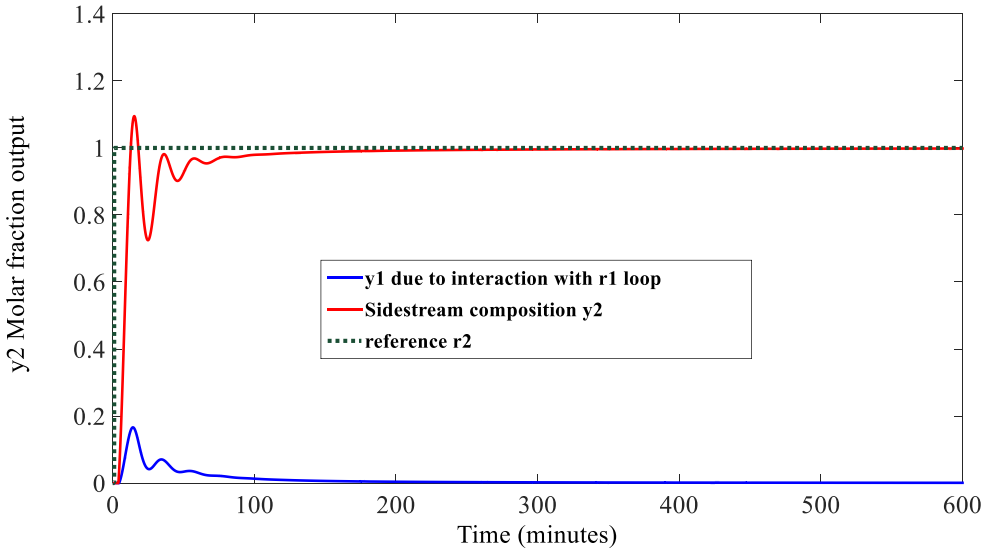


Fig. 4.12 Side-stream loop - y2 response to step change in r2

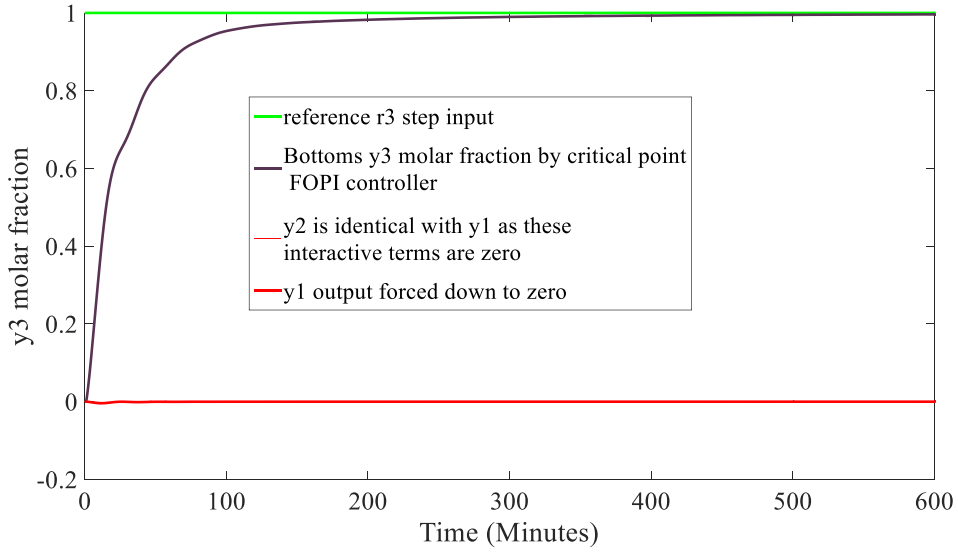


Fig. 4.13 Bottoms product loop – y3 response to step change in r3

The next step is to introduce set point change in loop two ($r2$) while all three loops remain closed. Fig. 4.12 shows that corresponding output ($y2$) rises to the set point. The controller forces interaction terms in loops one and three to zero as shown in Fig. 4.12. Finally, with all loops closed, set-point change is introduced in loop three ($r3$). The output in loop three due to the set point change is shown in Fig. 4.13. Outputs in other loops due to this set point change are also observed to be forced down to zero as expected. Each controller has good effect on interactions by reducing it to zero at steady state. This demonstrates how the proposed design method handles interaction in a multivariable system.

Table 4.2 *OR Column: OPI and proposed FOPI controller Settings*

Parameters	FOPI	Optimum PI (OPI)
$\boldsymbol{k_p}$	$\begin{bmatrix} 0.65 & 0 & 0 \\ 0 & -0.13 & 0 \\ 0 & 0 & 2.2 \end{bmatrix}$	$\begin{bmatrix} 1.2 & 0 & 0 \\ 0 & -0.15 & 0 \\ 0 & 0 & 0.6 \end{bmatrix}$
$\boldsymbol{k_I}$	$\begin{bmatrix} 0.2 & 0 & 0 \\ 0 & -0.1 & 0 \\ 0 & 0 & 2.4 \end{bmatrix}$	$\begin{bmatrix} 0.24 & 0 & 0 \\ 0 & -0.015 & 0 \\ 0 & 0 & 0.15 \end{bmatrix}$
Order λ	0.9	1

These algorithms yield control actions with a good compromise between robustness and performance. In this example, simulation has been carried out without any disturbance input. Disturbance rejection effect of the FOPID controller is studied in the next simulation example.

Considering the same *OR* distillation system, the proposed controller is simulated with significant disturbance signals introduced. A 20% step disturbance signal (*d*1) is introduced at time *t* = 500min (feed flow changes) while a 30% step disturbance signal is simultaneously introduced at *t*=600min as changes in feed temperature (*d*2). The column is simulated for 1,000 minutes and results are shown in *Fig*. 4.14- *Fig*. 4.20. Output responses are compared on Fig. 4.14, Fig. 4.16, Fig. 4.18 while control signals are compared on Fig. 4.15, Fig. 4.17 and Fig. 4.19.

It is desirable to establish how this controller compares with other well-known conventional PID control schemes. Therefore, the proposed controller (FOPI) is compared with an optimum PI controller proposed by Ogunnaike and Ray (Ogunnaike & Ray, 1983) under exact conditions. The optimum PI gains was obtained by optimisation via extended simulations as published by Ogunnaike and Ray (Ogunnaike & Ray, 1983). It is used as a baseline since it was proven to yield optimal controller settings for the distillation column. Comparison is based on recovery time from perturbation (settling time) as well as set-point tracking using integral absolute error. This is tabulated in Table 4.3

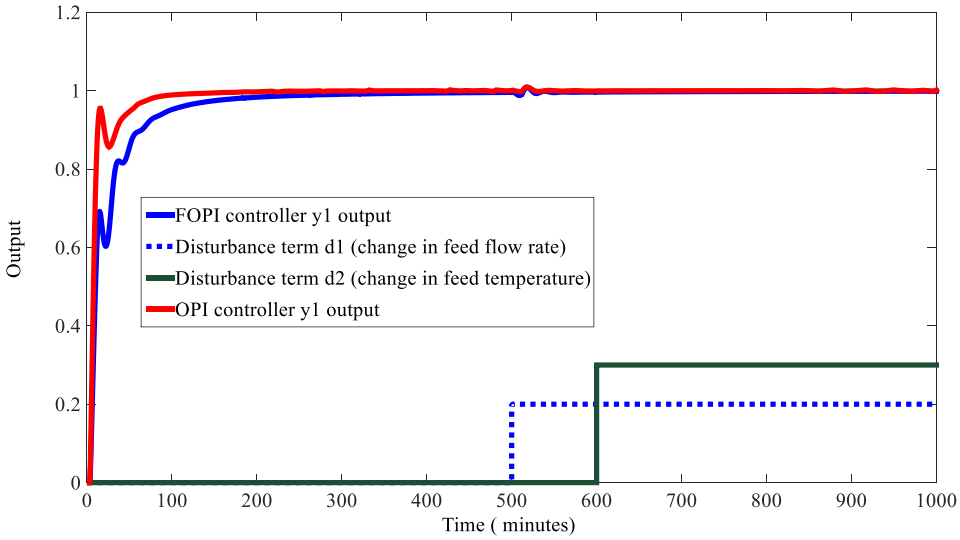


Fig. 4.14 Disturbance rejection – y1 loop

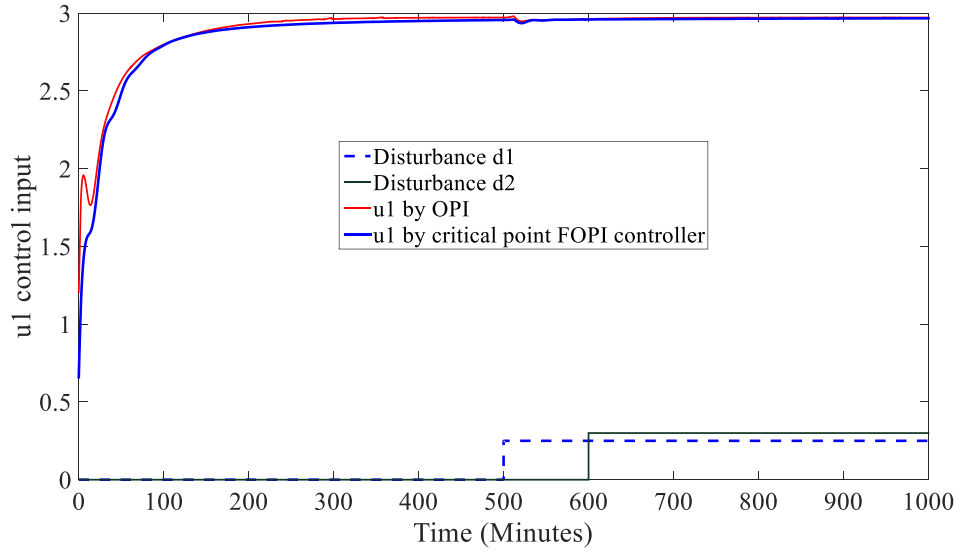


Fig. 4.15 Comparison of control effort in loop 1 – u1

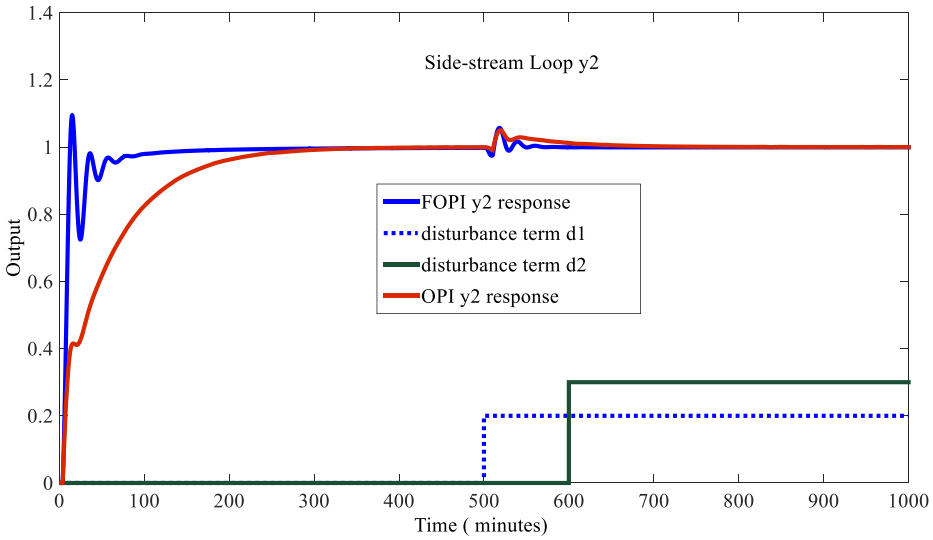


Fig. 4.16 Disturbance rejection – y2 loop

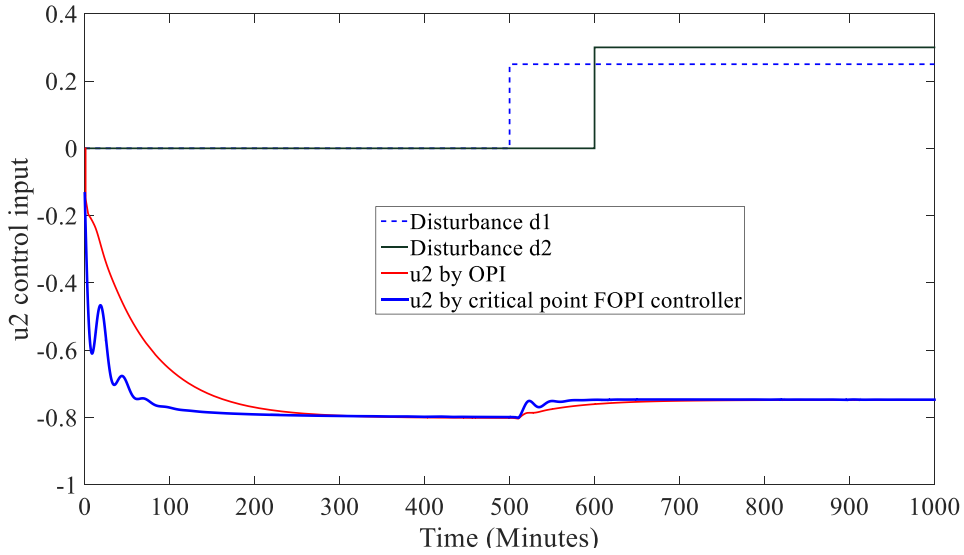


Fig. 4.17 Comparison of control effort in loop 2 – u2

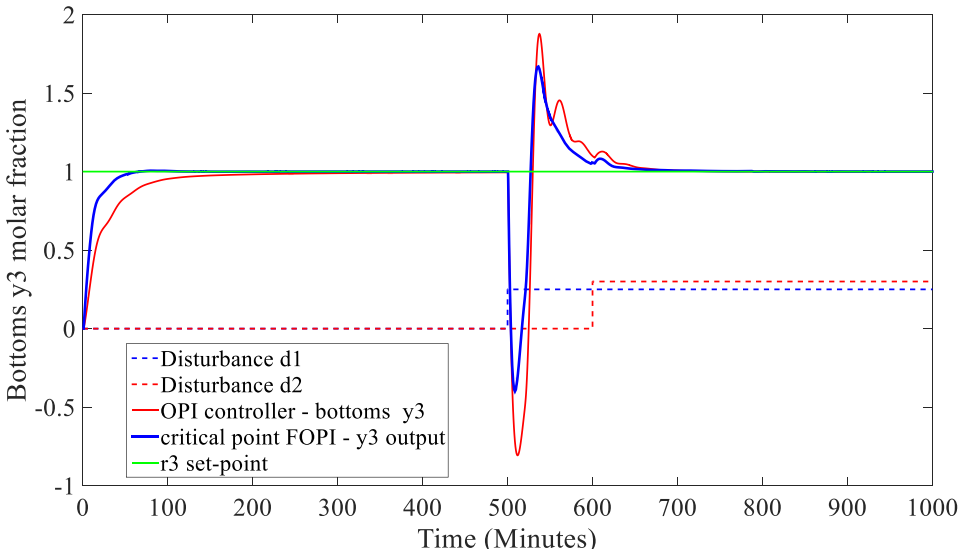


Fig. 4.18 Disturbance rejection – y3 loop

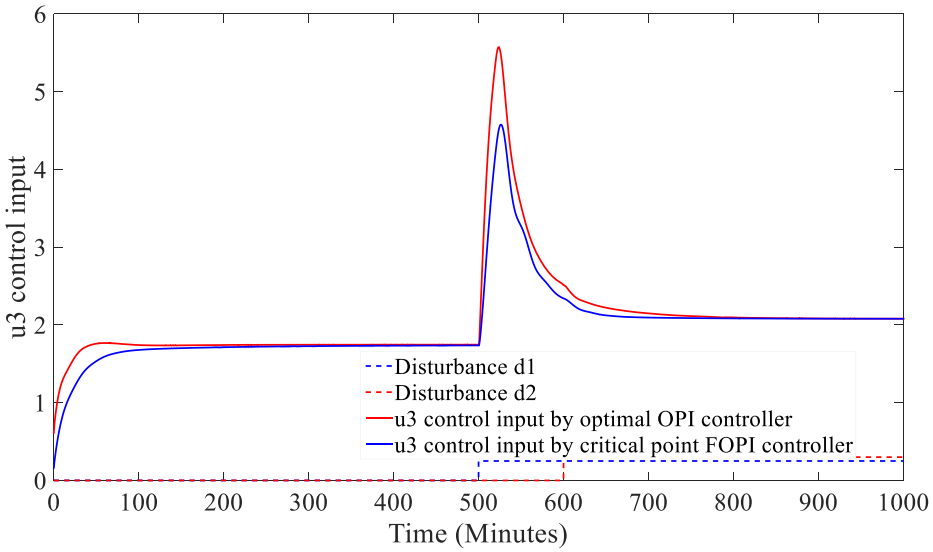


Fig. 4.19 Comparison of control effort in loop 3 – u3

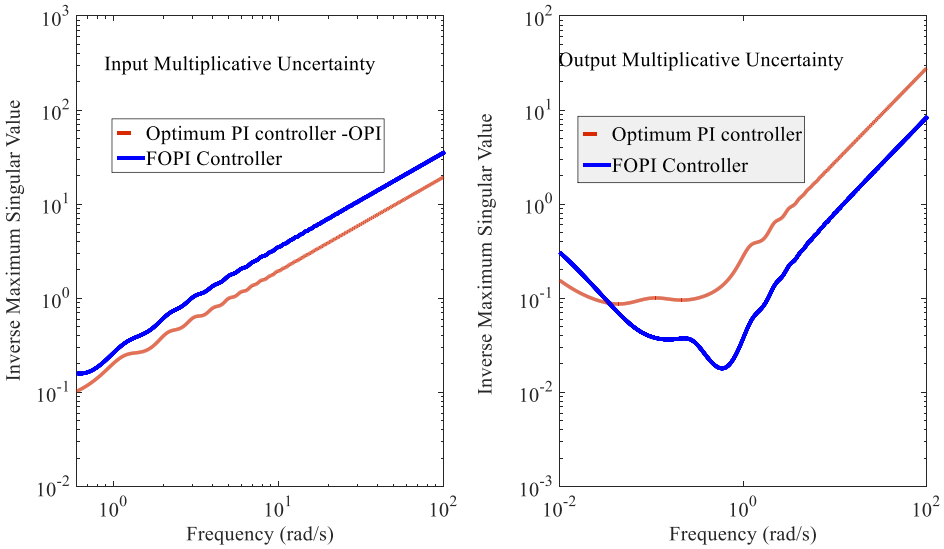


Fig. 4.20 Comparison of robust stability of the two controllers

Table 4.3 *OR Column: Performance of OPI and FOPI controllers*

Step Change	*y1*	*y2*	*y3*
FOPI - IAE	28.86	16.25	5.81
OPI - IAE	12.46	53.45	12.05
FOPI - Recovery Time (*mins*)	10	20	125
OPI – Recovery Time (*mins*)	10	80	130

It can be observed that the proposed FOPI controller compares favorably with ORA's optimum PI controller in terms of disturbance rejection and set-point tracking. Considering loops one and two (distillate and side stream product), FOPI controller reject both disturbances in a very short time (within 10 to 20 minutes). There is deterioration in the third loop due to simultaneous effects of these two disturbances in all three loops but FOPI still outperforms OPI controller in rejecting these input disturbances. Similarly, control signals of these two controllers are compared on Fig. 4.15, Fig. 4.17 and Fig. 4.19. These control efforts are found to be almost equal. Robust stability of both FOPI controller (blue line) and conventional but optimal (OPI) controller (red line) are shown simultaneously in Fig. 4.20. The constant gap between the two lines in both multiplicative-input and output sections show that they offer

different levels of robust stability. The region below each line shows enclosed region of stability. For instance, considering input multiplicative uncertainty, FOPI controller has higher region or level of robust stability (the blue line is higher than the red line for all frequency range of interest 0.01rad/s - 100 rad/s)

4.9 Conclusions

A new FOPI controller design scheme is derived analytically using critical-frequency-point parameters. Most algorithms for tuning FOPID controllers rely on optimisation routines which increases computational cost. However, this design scheme presents new and simple formulae for obtaining proportional and integral gains when using FOPI controller without using any optimisation routine. There are well established algorithms for tuning conventional PID controllers based on critical point information but very little is being published about tuning FOPID controllers using ultimate point parameters. Therefore, ultimate point parameters are used in this new algorithm for designing FOPID controllers. It is suitable for both SISO process control and MIMO applications. In MIMO case, a decentralised or multi-loop configuration is recommended using sequential loop closing technique. The procedure gives one a quick option of obtaining robust values for FOPID controller gains. However, computational cost is highly reduced as no optimisation routine is required unlike optimal PI controllers.

For instance, simulation studies carried out on the three-by-three ORA column shows that the proposed FOPI controller compares favourably with OPI controller but with settling time reduced by 70% i.e. from 90 minutes (OPI controller) to 20 minutes for output *y2* by the proposed FOPI controller. It gives similar responses in both *y1* and *y3* loops in terms of settling time but with integral absolute error reduced by 69.6% (from 53.45 to 16.25 for *y2*) and reduced by 51.8% (from 12.05 to 5.81 for *y3*) with step change in respective inputs *r2* and *r3*. OPI gives a smaller IAE in *y1* with step change in input *r1* as tabulated in Table 4.3.

Also, the proposed FOPI controller yields control actions with good compromise between robustness and performance. A method of judging robustness based on inclusion of input and output multiplicative uncertainty models have been presented. A closer look at Fig 4.20 shows that the proposed FOPI gives a more robust

control action if considering multiplicative input uncertainty because the blue line (FOPI controller) is higher than the red line (OPI controller) and that shows it has a larger robust stability region. With multiplicative output uncertainty model, optimum PI produces a slightly larger area of robust stability. However, one undesirable feature of this proposed scheme is the need for frequency response experiment for determination of critical frequency point. To overcome this limitation, another method of designing FOPID controller, without frequency response experiment, is presented in the next chapter.

Chapter 5

Design of FOPID Controllers by IMC Approach

In chapter 4, there was need for critical frequency point experiment before FOPID controllers can be synthesised. This can pose a practical limitation for certain real-time process control systems. Consequently, a new method of designing FOPID controller for multivariable process control is presented in this chapter without any need for critical frequency point experiment. The proposed algorithm is based on an established controller design method known as Internal Model Control (IMC) scheme. Results are compared with Ogunnaike's optimal PI (OPI) controller. The proposed controller shows greater robustness compared to OPI and results have been published (Edet & Katebi, 2018).

5.1 Derivation of FOPI controller Gains by IMC

A decentralised FOPID controller design method is developed based on modified internal model control approach. Final fine tuning of settings is carried out using biggest log-magnitude tuning technique. There are well-established internal model control design schemes for designing conventional PID controllers for both SISO and multivariable applications. One of such successful design methodology is the Smith's predictor proposed decades ago mainly for scalar process control application. It has been modified over the years for multivariable systems control. Other PID controller methods that use IMC technique are Tanttu and Lieslehto multivariable PI controller design method as well as power series method. These methods are limited to conventional PID controllers with integer orders and does not address FOPID controller design problem. To extend this fundamental idea of internal model to controllers with fractional orders, a new fractional order PID design method is derived. The proposed control design scheme is developed with set-point tracking as a primary design objective just like Skogestad's IMC rule for conventional SISO PID controllers.

Consider a SISO transfer function $G(s)$ with FOPDT dynamic characteristic:

$$G(s) = \frac{k_1 e^{-Ls}}{\tau_1 s + 1}, \tag{5.1}$$

where k_1 is the steady state open loop gain and τ_1 is the time constant for the same system; L is the process's dead time; if desired trajectory is represented by D, the trajectory equation is given as follows (5.2) to achieve set-point tracking:

$$D = \frac{e^{-Ls}}{\tau_f s + 1}, \tag{5.2}$$

where τ_f is the desired time constant of the system. In conventional IMC PID control system, this desired time constant provides a means of tuning the IMC PID controller. However, in the fractional-order PID controller case, using this single tuning parameter (τ_f) is insufficient. Consequently, a suitable design and tuning algorithm is developed for FOPI controllers and it takes the fractional-order dynamics of the controller into consideration. Since the desired set-point objective is specified for the entire closed-loop control system, it implies that D can also be expressed as follows:

$$D = \frac{C(s)G(s)}{1 + C(s)G(s)}. \tag{5.3}$$

The controller $C(s)$ is of the FOPI form given in (5.4) with same parameters as already defined in (2.2):

$$C(s) = k_p + \frac{k_I}{s^\mu} = k_p \left(\frac{\tau_I s^\mu + 1}{\tau_I s^\mu} \right). \tag{5.4}$$

From (5.3), we can solve for $C(s)$: $C(s) = \dfrac{D}{G(s) - DG(s)}$

$$C(s) = \frac{e^{-Ls}}{\tau_f s + 1} \div \left(\frac{k_1 e^{-Ls}}{\tau_1 s + 1} - \left(\frac{e^{-Ls}}{\tau_f s + 1} \right) \left(\frac{k_1 e^{-Ls}}{\tau_1 s + 1} \right) \right)$$

Substituting controller equation as given in (5.4) yields:

$$k_p \left(\frac{\tau_I s^\mu + 1}{\tau_I s^\mu} \right) \equiv \frac{e^{-Ls}}{\tau_f s + 1} \div \left(\frac{k_1 e^{-Ls}}{\tau_1 s + 1} - \left(\frac{e^{-Ls}}{\tau_f s + 1} \right) \left(\frac{k_1 e^{-Ls}}{\tau_1 s + 1} \right) \right)$$

$$k_p\left(\frac{\tau_I s^{\mu}+1}{\tau_I s^{\mu}}\right)\equiv\frac{\tau_1 s+1}{k_1(\tau_f s+1)-k_1+k_1 L s} \tag{5.5}$$

$$k_p\left(\frac{\tau_I s^{\mu}+1}{\tau_I s^{\mu}}\right)=\frac{\tau_1 s+1}{k_1(\tau_f+L)s}$$

To simplify (5.5), we work in complex frequency domain as $s = j\omega$.

The term:

$$j^{-\mu}=\left(\cos\frac{\mu\pi}{2}-j\sin\frac{\mu\pi}{2}\right)$$

$$\Rightarrow k_p\left(1+\frac{1}{\omega^{\mu}\tau_I}\left(\cos\frac{\mu\pi}{2}-j\sin\frac{\mu\pi}{2}\right)\right)\equiv\frac{1+j\omega\tau_1}{j\omega k_1(\tau_f+L)}. \tag{5.6}$$

Considering RHS of (5.6) and rationalising it to remove complex operator from denominator, it is further simplified as follows:

$$k_p\left(1+\frac{1}{\omega^{\mu}\tau_I}\left(\cos\frac{\mu\pi}{2}-j\sin\frac{\mu\pi}{2}\right)\right)=A-jB,$$

where $A=\dfrac{k_1\left(L-\omega^2\tau_1\tau_f\right)}{-\left(\omega^2k_1^2\tau_f^2+2\omega k_1^2L\tau_f+k_1^2L^2\right)}$,

$$B=\frac{\omega k_1\tau_f+\omega k_1 L\tau_1}{\omega^2k_1^2\tau_f^2+2\omega k_1^2L\tau_f+k_1^2L^2}.$$

Comparing real part yields:

$$k_p\left(1+\frac{\cos\frac{\mu\pi}{2}}{\omega^{\mu}\tau_I}\right)=A \tag{5.7}$$

$$k_p=\frac{A}{1+\omega^{\mu}\tau_I\cos\frac{\mu\pi}{2}} \tag{5.8}$$

Integral gain (and by extension integral time) can be obtained by comparing imaginary part:

$$-\frac{k_I \sin\frac{\mu\pi}{2}}{\omega^{\mu}} \equiv -B$$

$$k_i = \frac{B\omega^{\mu}}{\sin\frac{\mu\pi}{2}} \tag{5.9}$$

Integral gain is computed first before combining (5.8) and (5.9) to get proportional gain. Integral time can be obtained as:

$$\tau_I = abs\left(\frac{A}{k_I} - \frac{\cos\frac{\mu\pi}{2}}{\omega^{\mu}}\right)$$

These new set of formulae (5.8 and 5.9) define proportional and integral gains for the IMC-FOPI control scheme. Initial settings of controller parameters are obtained for each loop before final fine tuning. Since FOPID controller has been tested extensively for time delayed processes, a guide to selection of fractional order based on delay factor is available. Table 4.1 used in previous cases is also relevant for selection of fractional-order for the integral component of the IMC-FOPI controller. Fractional order is pre-selected before calculating other controller parameters.

These IMC gains can be extended to multivariable process control using a multi-loop configuration. A generalized multi-input multi-output system *G*(s) is considered with two or three individual loops. For a TITO system, the MIMO controller will be of the form given in (2.2) with parameters as already defined in (2.2):

$$C(s) = \begin{bmatrix} C_1(s) & 0 \\ 0 & C_2(s) \end{bmatrix};\ C_j(s) = k_{p,j} + \frac{k_{I,j}}{s^{\mu}},\ j = 1, 2. \tag{5.10}$$

If the system is a three-by-three system, the MIMO controller will be of the form *C*(*s*) as already defined in chapter two where:

$$C(s)=\begin{pmatrix} C_1(s) & 0 & 0 \\ 0 & C_2(s) & 0 \\ 0 & 0 & C_3(s) \end{pmatrix}, C_j(s)=k_{p,j}+\frac{k_{I,j}}{s^{\mu}},\ j=1,2,3. \qquad (5.11)$$

Finally, FOPI controller settings are determined individually for each *jth*-diagonal transfer function using (5.12), (5.13) and (5.14).

$$k_{I,j-imc}=\frac{B\omega^{\mu}}{\sin\frac{\mu\pi}{2}} \qquad (5.12)$$

$$\tau_{I,j-imc}=abs\left(\frac{A}{k_{I,j-IMC}}-\frac{\cos\frac{\mu\pi}{2}}{\omega^{\mu}}\right) \qquad (5.13)$$

$$k_{p,j-imc}=\frac{A}{1+\omega^{\mu}\tau_{I,j-IMC}\cos\frac{\mu\pi}{2}} \qquad (5.14)$$

where: $k_{p,j-imc}$ is the proportional gain for FOPI controller in *j*th loop.

$k_{I,j-imc}$ is the integral gain for FOPI controller in *j*th loop.

$\tau_{I,j-imc}$ is the integral time for *j*th loop's FOPI controller.

5.2 Proposed Multivariable Tuning Method (BLT- IMC)

Although the proposed IMC settings in (5.13) and (5.14) achieves good set-point tracking, further fine tuning of these settings is recommended for effective disturbance rejection. To strike that balance between performance and robustness, biggest log-magnitude tuning method is utilised to fine tune IMC-FOPI controller gains. This means that FOPI controller gains given above in (5.12) and (5.14) provide initial (starting) point for the BLT tuning process. This Proposed multivariable FOPI controller tuning method is similar to the conventional PID control design algorithm proposed by Besta and Chidambaram (Besta & Chidambaram, 2015). However, derived expressions are different in the present scheme because fractional dynamics are fully accounted for during control law formulation.

To use this FOPI controller design method, consider only diagonal process elements. Determine the IMC gains for each diagonal loop using (5.12) and (5.14). Here, the IMC tuning parameter is unused as it is set to one. The BLT detuning factor *F* is initially chosen as 0.7 if relative gain array of the system λ_{ij} < 1. If the relative gain array is greater than one, *F* is initially assumed to be 1.5. These values are found to be good starting point to deal adequately with interaction in each system. For processes with relative gain array greater than one, detuning involves decreasing control gains and increasing integral times. If derivative component is used, derivative time is decreased during the detuning process too. The procedure is briefly summarised here.

- Preliminary gains of the controller are calculated as follows:

$$k_{p,j} = \frac{k_{p,j-IMC}}{F} \tag{5.15}$$

$$\tau_{I,j} = F \times \tau_{I,j-IMC} \tag{5.16}$$

- The diagonal controller matrix is calculated as *C*(s) given in (5.11).
- Determine the corresponding multivariable Nyquist diagram of the characteristic (scalar) function *W*:

$$W(j\omega) = -1 + \left| I + C(j\omega)G(j\omega) \right| \tag{5.17}$$

where *I* represents identity matrix of corresponding size.

- Determine the multivariable closed- loop log modulus *L* as shown below in (5.18).

$$L(j\omega) = 20 \times Log_{10} \left| \frac{W(j\omega)}{1 + W(j\omega)} \right| \tag{5.18}$$

- The peak of *L* over the entire frequency range is the biggest log modulus termed $L_{\max}$.

- Finally, the factor *F* is varied (with 0.01 incrementally) until $L_{\max}$ is equal to *2n* (4 dB for two-input two-output system and 6 dB for three-input three-output system). Here, *n* is the number of independent loops in the system.
- Final gains are obtained using *F* equal to *2n*. FOPI controller is realised using (5.15) and (5.16) with the new value of tuning factor *F*.

5.2.1. Example 1- Composition control of a TITO distillation system

The diagonal transfer functions are:

$$G_1(s) = \frac{12.8e^{-s}}{16.7s+1}; G_2(s) = \frac{-19.4e^{-3s}}{14.4s+1}.$$

IMC gains are calculated as explained in the algorithm and tuned accordingly as *F* is varied until $L_{\max}$ equals *4*dB. Resultant parameter gains are tabulated in Table 5.1. The transient response for both top composition loop and bottoms composition loop are shown in Fig. 5.5 and Fig. 5.6 respectively. In Fig. 5.5, a unit step response of the system is simulated due to a step change in reference input (*r*1) while keeping *r*2 at zero. It can be observed that *y*1 rises to follow the set-point while reducing interaction in other loops to zero (*y2* = 0). In Fig. 5.6, second input *r*2 is set to 1 (unit step) while keeping *r*1 at zero. The controller ensures set-point tracking as *y*2=1 while *y*1 is simultaneously brought down to zero. In the next section, a more challenging system (ORA column) is considered. The relative gain array is greater in ORA column and that shows the greater level of interaction and difficulty of control.

5.2.2 Example 2 - Composition control of OR Distillation Column

The three diagonal transfer functions are considered independently. That is:

$$\frac{y_1}{u_1} = \frac{0.66e^{-2.6s}}{6.7s+1}, \frac{y_2}{u_2} = \frac{-2.36e^{-3s}}{5s+1}, \frac{y_3}{u_3} = \frac{0.87(11.61s+1)e^{-s}}{(3.89s+1)(18.8s+1)}.$$

Initial IMC settings are calculated as explained in the algorithm. The transfer function of the third loop is first approximated as a FOPDT model using Taylor series before calculating initial IMC settings. If the second order transfer function is used directly, a derivative component will be required. In this work, only proportional and integral gains are required using the FOPI controller structure. These gains are tuned

accordingly as F is varied until $L_{\max}$ equals 6dB. Resultant parameter gains are tabulated in Table 5.2. In addition, integral absolute error (IAE) index is used to judge performance in terms of set-point tracking and it is tabulated in Table 5.3.

Simulated results demonstrate that the proposed IMC-FOPI controller compares favourably with an optimal PI controller in terms of set-point tracking and counteracting multivariable interaction. However, computational cost is significantly reduced when using IMC-FOPI controller. Fig. 5.1 and Fig. 5.2 illustrate responses in all three loops of the column because of a step change in $r1$. The corresponding output $y1$ rises to one while the proposed controller forces other outputs in loops two and three to zero after 80 minutes. This is better than the settling time provided by the conventional Optimal PI (OPI) controller. OPI controller brings down interaction in loop two to zero after 175 minutes.

The next step is to introduce set-point change in loop two ($r2$) while all three loops remain closed. Fig. 5.3 shows that the corresponding output ($y2$) tracks the set-point too. The controller forces interaction terms in loops one and three to zero as shown in the same diagram. Finally, with all loops closed, set-point change is introduced in loop three ($r3$). The output in loop three due to the set point change is shown in Fig. 5.4. Outputs in the other two loops due to this set point change are also observed to be forced down to zero as expected. Each controller has good effect on interactions by reducing it to zero at steady state. This demonstrates how the proposed design method handles interaction in a multivariable system and how it compares with optimal PI controller.

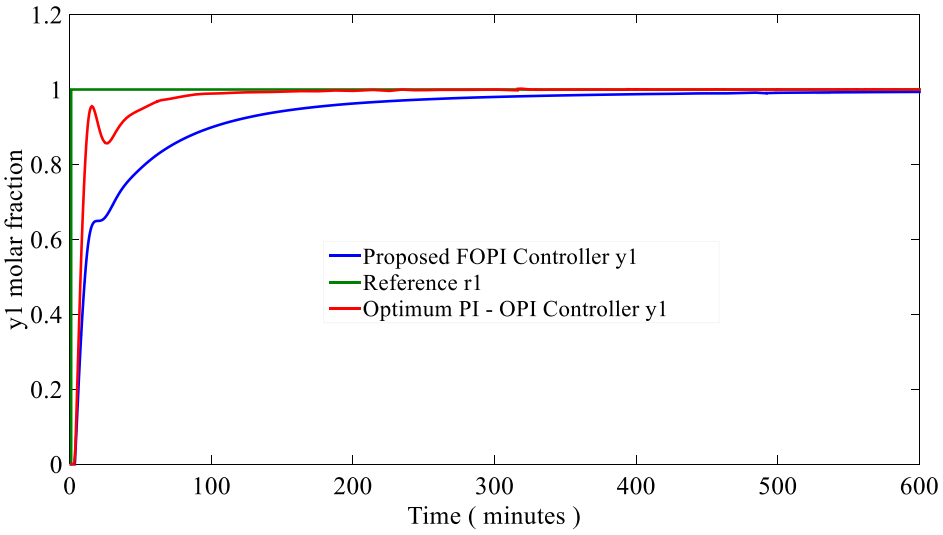


Fig. 5.1 Output - top composition set-point tracking comparison.

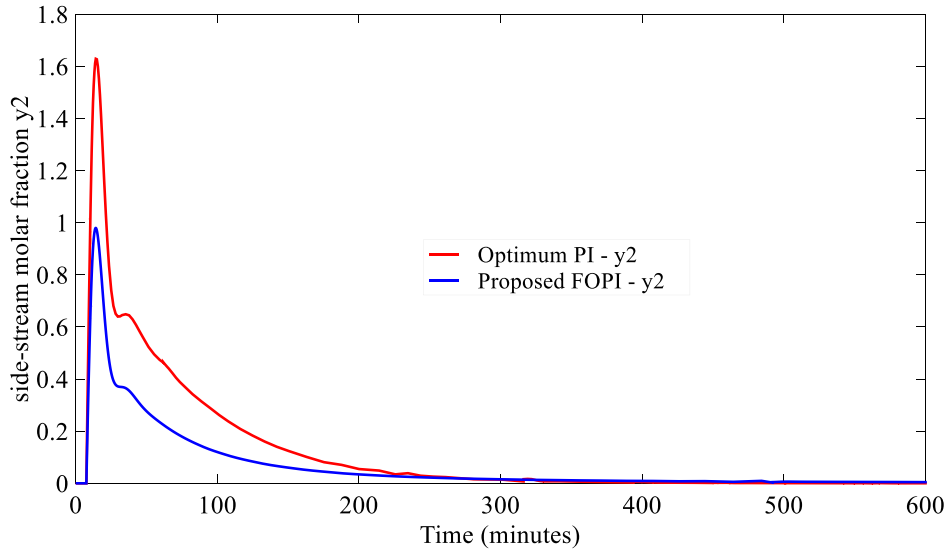


Fig. 5.2 Side-stream composition (y2) output due to input r1 set to 1.

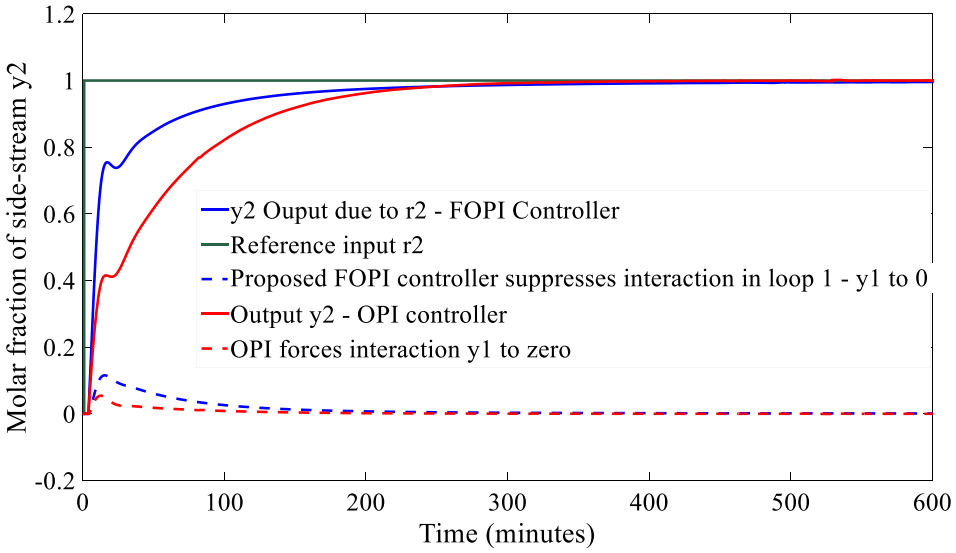


Fig. 5.3 Side-stream composition loop: Set-point tracking comparison

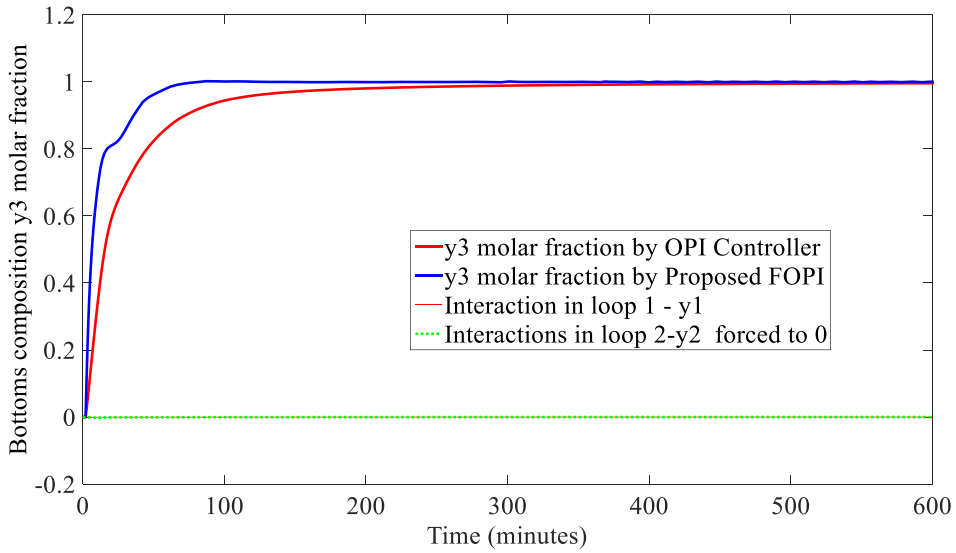


Fig. 5.4 Output: Bottoms composition set-point tracking comparison

Table 5.1 *WOBE Column: Controller Parameters (F=0.57, L=4dB)*

Parameters	IMC	BLT-IMC
k_p	$\begin{pmatrix} 0.0924 & 0 \\ 0 & -0.0152 \end{pmatrix}$	$\begin{pmatrix} 0.1621 & 0 \\ 0 & -0.0267 \end{pmatrix}$
k_I	$\begin{pmatrix} 0.0196 & 0 \\ 0 & -0.0100 \end{pmatrix}$	$\begin{pmatrix} 0.0603 & 0 \\ 0 & -0.0310 \end{pmatrix}$
order	0.7	0.7

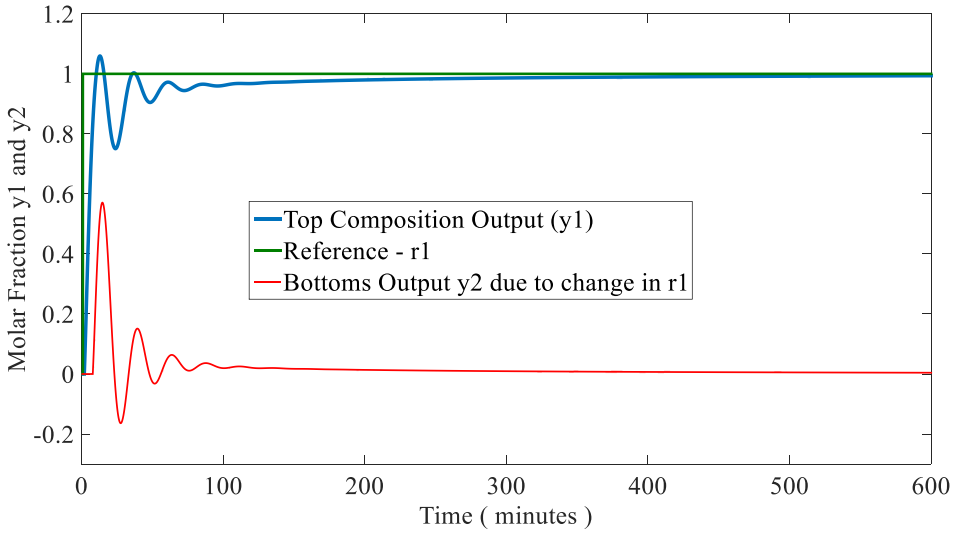


Fig. 5.5 Top composition response – input r1 is set to one.

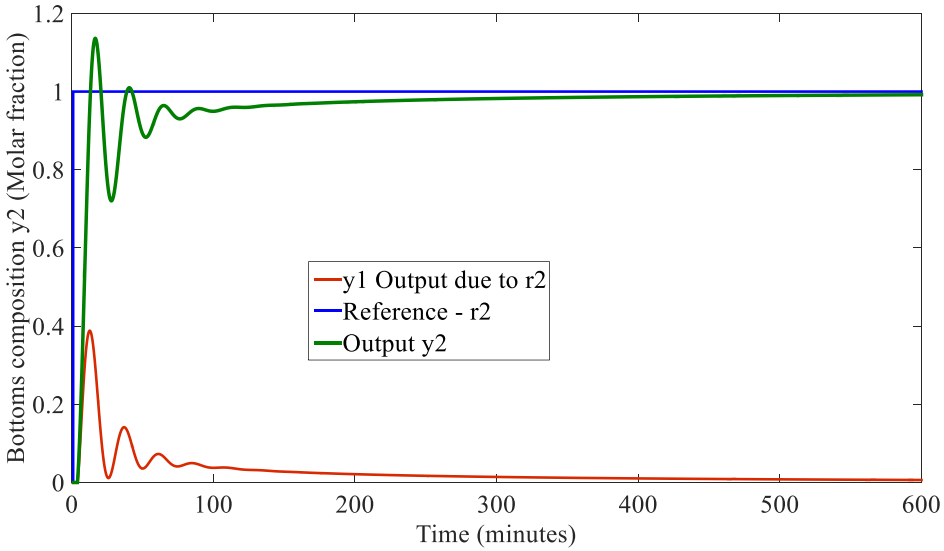


Fig. 5.6 Bottoms composition loop- input r2 is set to one. r1 is not excited

Table 5.2 *OR Column: Obtained PI Controller (F=0.46, L=6dB)*

Settings	IMC	BLT-IMC	Optimum PI (OPI)
K_p	$\begin{bmatrix} 0.4256 & 0 & 0 \\ 0 & -0.1075 & 0 \\ 0 & 0 & 0.1245 \end{bmatrix}$	$\begin{bmatrix} 0.7881 & 0 & 0 \\ 0 & -0.1991 & 0 \\ 0 & 0 & 0.2306 \end{bmatrix}$	$\begin{bmatrix} 1.2 & 0 & 0 \\ 0 & -0.15 & 0 \\ 0 & 0 & 0.6 \end{bmatrix}$
Ki	$\begin{bmatrix} 0.0423 & 0 & 0 \\ 0 & -0.0143 & 0 \\ 0 & 0 & 0.0044 \end{bmatrix}$	$\begin{bmatrix} 0.1452 & 0 & 0 \\ 0 & -0.0491 & 0 \\ 0 & 0 & 0.0151 \end{bmatrix}$	$\begin{bmatrix} 0.24 & 0 & 0 \\ 0 & -0.015 & 0 \\ 0 & 0 & 0.15 \end{bmatrix}$
Order μ	0.9	0.9	1

Table 5.3 *Controller Performance Comparisons - OR*

Step Change	y1	y2	y3
FOPI - IAE	38.4	31.0	33.9
OPI - IAE	12.42	53.48	12.06
FOPI - Recovery Time	10*m*	15*m*	90*m*
OPI-Recovery Time (*m*)	10	80	130

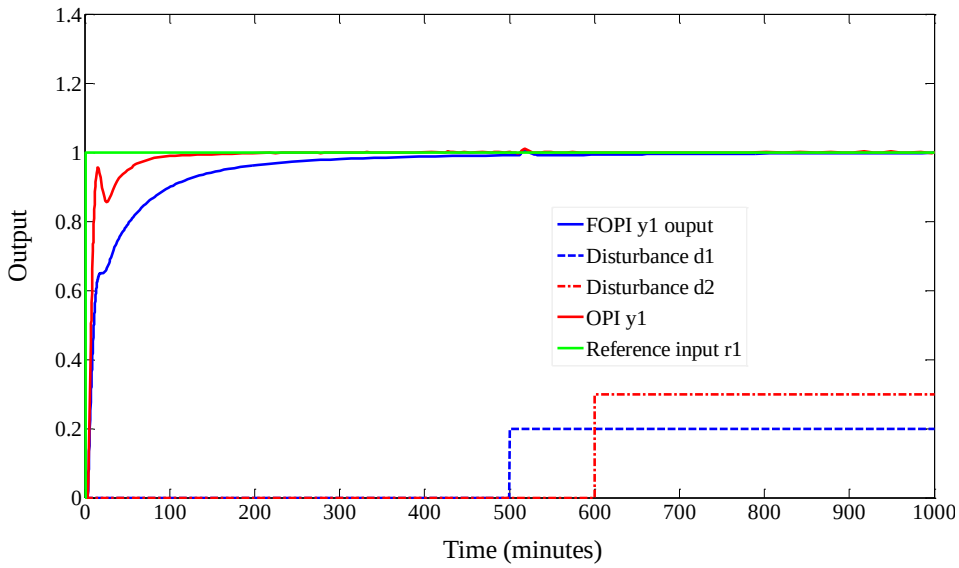


Fig. 5.7 Top composition response – input r1 is set to one

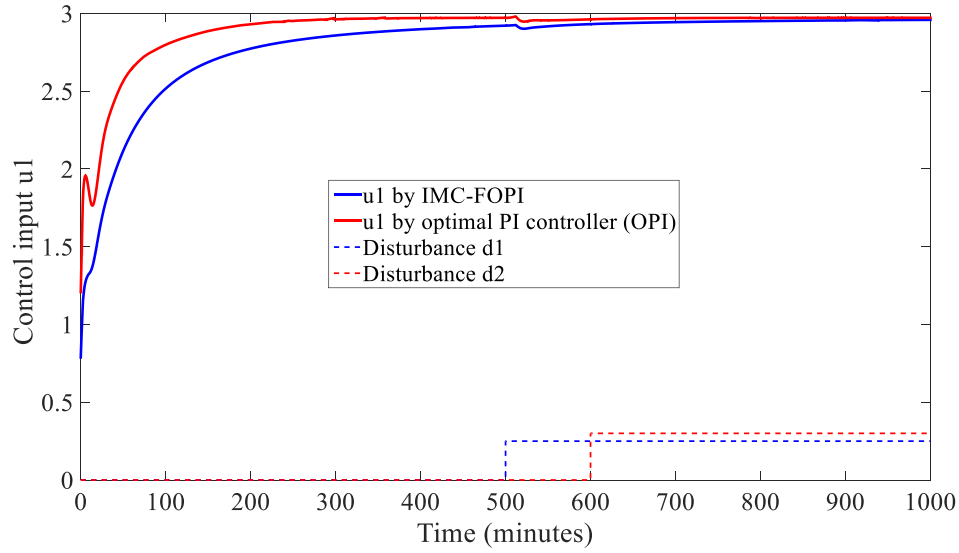


Fig. 5.8 Comparison of IMC-FOPI control effort in loop 1 – u1

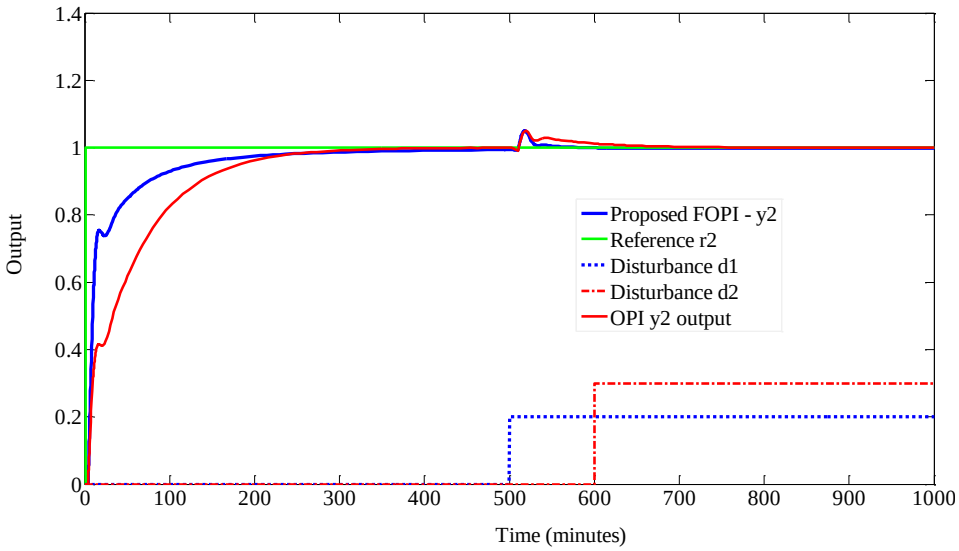


Fig. 5.9 Side-stream composition loop – input r2 is set to one

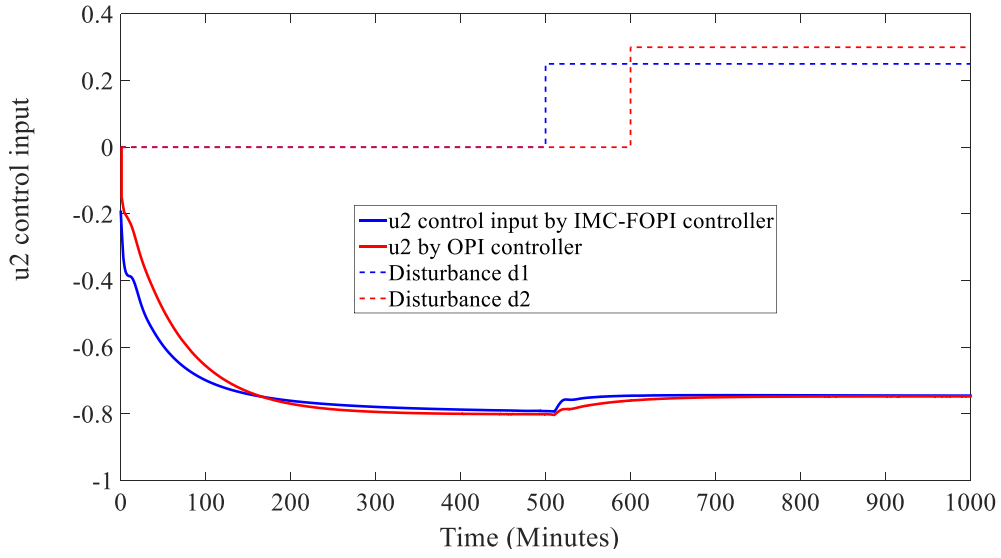


Fig. 5.10 Comparison of IMC-FOPI control effort in loop 2 – u2

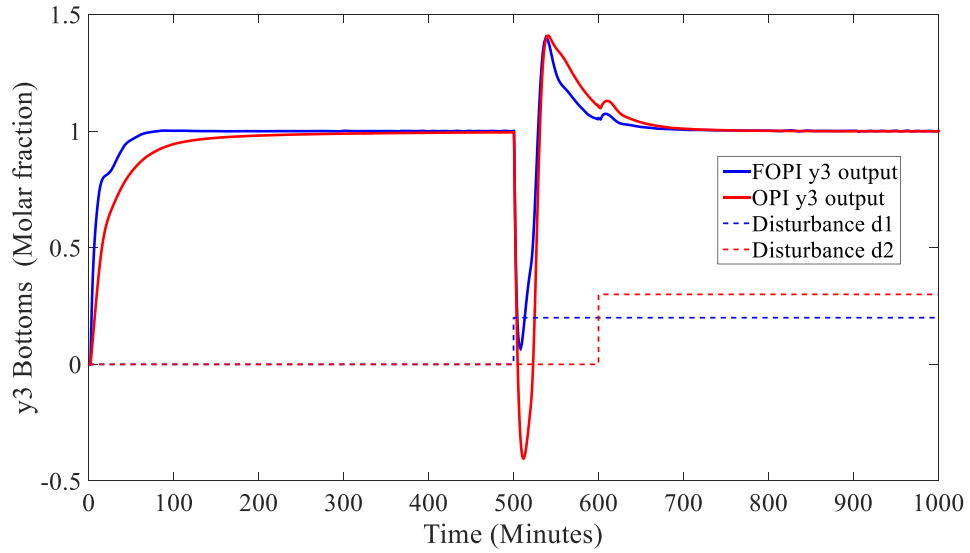


Fig. 5.11 Output bottoms composition loop – input r3 is set to one

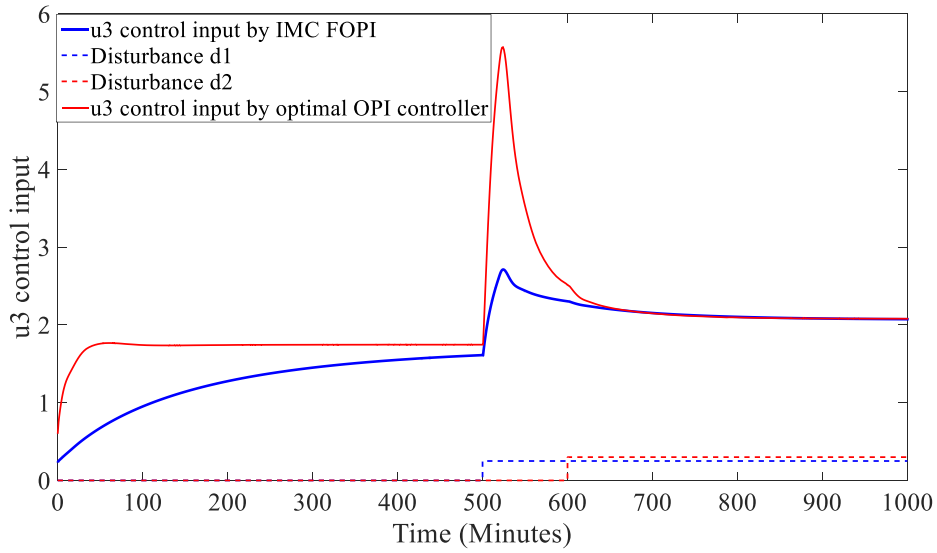


Fig. 5.12 Comparison of IMC-FOPI control effort in loop 3 – u3

5.3 Control Performance and Disturbance Rejection

Considering *OR* distillation column, the proposed controller is simulated under drastic perturbations. A 20% step disturbance signal ($d1$) is introduced at *t= 500 minutes* (feed flow changes) while a 30% step disturbance signal is simultaneously introduced at *t=600 minutes* as changes in feed temperature ($d2$). The simulation is carried out for *1000 minutes* and results are shown in Fig. 5.7 - Fig. 5.12. These diagrams (Fig. 5.7, Fig. 5.9 and Fig. 5.11) indicate how the disturbance rejection of both controllers compare in terms of output regulation. The top distillate and side-stream composition loops almost have a smooth response despite significant amount of disturbance in both feed flow rate and feed temperature (20% change in $d1$ and 30% change in $d2$). The bottoms composition loop is more sensitive to disturbance, but these controllers still attenuate their effects within 80 minutes. Fig. 5.8, Fig. 5.10 and Fig. 5.12 compares the control effort of these controllers. FOPI controller is found to use smaller control effort to achieve specified output regulation targets compared to the OPI controller.

In terms of robustness, the proposed BLT-IMC FOPI controller is compared with an optimum PI controller under exact conditions and disturbances. Inverse maximum singular value analysis of sensitivity is used to quantify robustness of the FOPI control system and results are plotted in Fig. 5.13 (blue line). The same procedure is repeated using ORA optimum PI control system within the same

frequency range (red line). The area below each curve represents stability region as each line depicts stability bounds. It can be observed that the blue line covers a greater area and that shows a greater stability region provided by the proposed FOPI controller within the frequency range of interest. Performance comparison based on recovery time from perturbation (settling time) as well as set-point tracking using integral absolute error. This is also tabulated in Table 5.4. The proposed method compares favourably with the optimum PI method as reflected in the tabulated IAE index. However, IMC is based on simple time domain and frequency response calculations and does not require any extensive optimisation routine. This reduces computational burden when compared with optimal methods like Optimum PI (OPI). In addition, it yields a more robust control system as shown by the ISV analysis in Fig. 5.13.

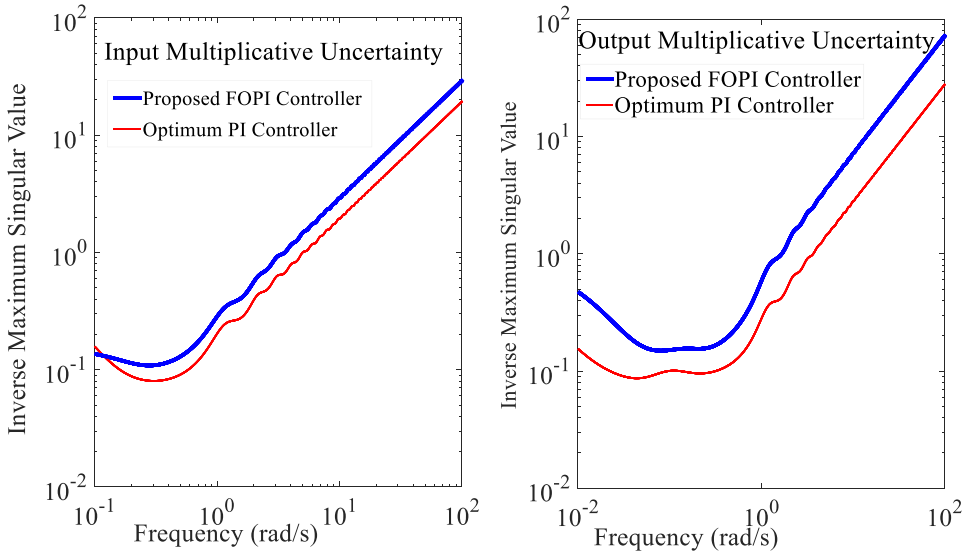


Fig. 5.13 Robust stability comparison with input and output uncertainties

5.4 Conclusions

A new multivariable control design scheme is proposed for open loop stable multivariable system in this chapter and details are summarised in a flow chart diagram of Fig. 5.14. This combines both internal model control strategy and BLT method for tuning fractional order PI controllers. The proposed control system does not depend on critical frequency point experiment. This algorithm presents new and simple formulae for proportional and integral gains when using FOPI control system. However, obtained IMC settings are further fine-tuned using biggest log-magnitude technique. Although, implementation of this IMC-FOPI controller can be done using

a fully cross-coupled or centralised controller configuration, decentralised configuration is presented because target systems of interest are expected to have many loops (more than two or three inputs and outputs). The proposed algorithm produces more robust control actions compared to conventional OPI controller with input and output multiplicative uncertainty models considered (see Fig. 5.13). Simulation studies have been carried out using the highly interactive *OR* distillation model (with two disturbance inputs of 20% and 30% variations introduced as $d1$ and $d2$). Step responses in all three loops show good set-point tracking, zero overshoot and good disturbance rejection. This compares favourably with Ogunnaike's optimum PI controller because recovery time in top composition's loop has been reduced by 50% (from 10 minutes to 5 minutes for *y1*) and reduced by 87.5% (from 80 minutes to 10 minutes for *y2*) with step change in respective inputs *r1* and *r2* while *y3* does not show any comparative improvement.

In conclusion, the proposed IMC-BLT FOPI controller gives design engineer an easier method of estimating the gains of FOPI controller for process control. Since it yields better stability bounds with respect to both input and output multiplicative uncertainty models, it is more suitable for many process control applications. All these design methods treat the controller synthesis in continuous time. In certain stringent applications where optimal control performance may be required, robust benefits of FOPI controller can be combined with predictive features of discrete model-based predictive control scheme to formulate a new Fractional Order Predictive PI (FOPPI) controller with improved set-point tracking. This predictive approach is discussed in discrete time domain. It is well-known that incorporation of set-point information in formulation of control law improves control actions especially, set-point tracking. This hybrid benefit is exploited in a new FOPPI control scheme.

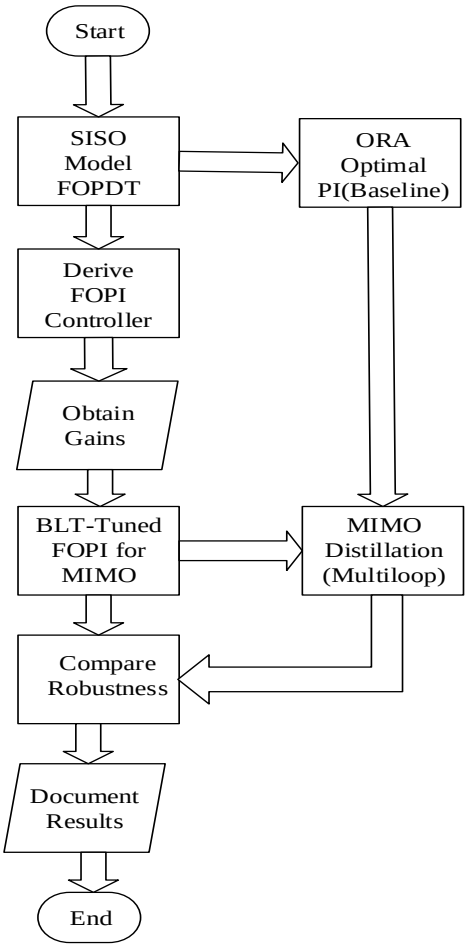


Fig. 5.14 Summary of chapter five in a flow chart

Chapter 6

Fractional-order Predictive PID Controller

In previous chapters, FOPID controllers have been designed in continuous time domain. These continuous time FOPID controllers are particularly beneficial for process control systems where parametric system models are available in continuous form e.g. Laplace Transfer function model. However, many process control applications exist where controllers are implemented in digital form. Examples include programmable logic controllers and direct use of computers in general. In such instances, continuous time signals have to be discretized and converted to digital form before control synthesis using the computer. Thereafter, discrete control signal output may be converted back to continuous time signal before application to a real time plant which works daily in continuous time. In this chapter, a new Fractional Order Predictive Proportional-Integral (FOPPI) controller is designed in discrete time domain.

This FOPPI controller shares similar features with Model-based Predictive Controller (MPC), especially, Dynamic Matric Control (DMC) algorithm. A general state space model of plant is assumed to be available and the model is used for prediction of future outputs just as it is done when using Dynamic Matrix Control (DMC) algorithm. A basic block diagram illustrating MPC algorithm is shown in Fig. 6.1 while the MPC process is illustrated in Fig.6.2. On the contrary, this novel Fractional Order Predictive Proportional-Integral (FOPPI) controller retains combined benefits of conventional predictive control algorithm and robust features of fractional-order PID controller. Results have been published (Edet & Katebi, 2017). The novelty is in the robust control action of the FOPPI control scheme. A single control narrative (FOPPI) combines predictive feature and robust stability such that anticipatory control action is ensured without compromising the robustness of a classical PID controller. Firstly, discrete MPC control technique is reviewed without any fractional dynamics. Thereafter, FOPPI controller design scheme is derived for multivariable process control systems.

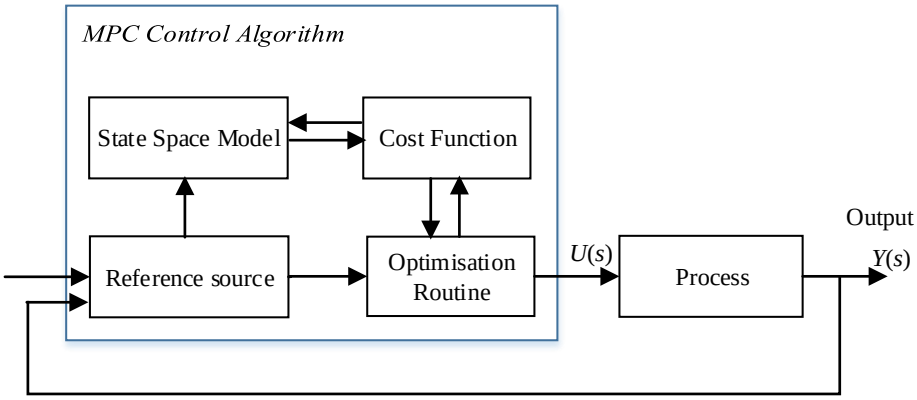


Fig. 6.1 Basic Block diagram of MPC structure

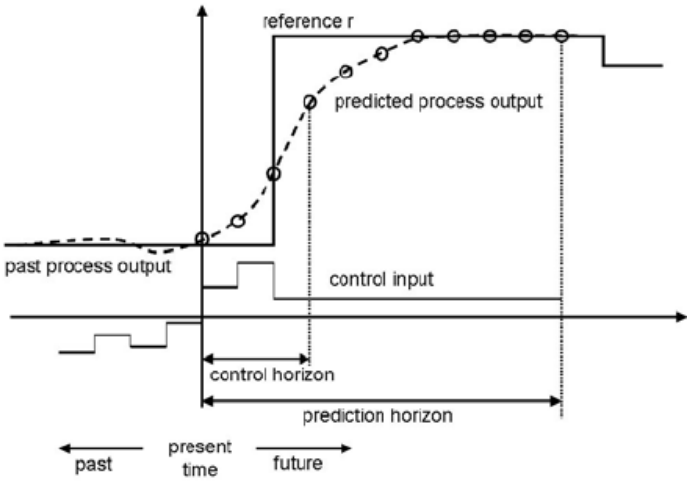


Fig. 6.2 Basic principle of MPC implementation (Esko, 2018)

6.1 Review of DMPC

Model-based Predictive Control (MPC) narrative is an inherent multivariable control design scheme that yields optimal or sub-optimal controller gains. It is implemented in a centralised control configuration. Several variants of MPC schemes have been developed and applied to systems with optimal results (Katebi & Moradi, 2001; Naeem, et al., 2005; Naeem, et al., 2004). Some of these DMPC algorithms have been modified to optimally tune conventional PID controllers with reasonable level of success (Katebi & Moradi, 2001). These DMPC algorithms include: Generalised Predictive Control (GPC), Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC) and Finite Spectrum Assignment (FSA) etc. The central feature is

retained which is incorporation of future set-point information in control law formulation. However, each algorithm differs slightly in the structure of objective function being optimised and in the type of model used for prediction of future set-point (Camacho & Bordons, 1999).

This motivates the idea of using discrete MPC algorithm to design fractional order PID controller for open loop stable MIMO systems. In this section, a brief review of discrete model predictive control procedure is given. Design of MPC based controllers for scalar process control application is considered in the first section (as illustrated in Fig. 6.2) before extending it to MIMO applications.

Consider a SISO system described by the state space equation below:

$$\begin{aligned} x(k+1) &= A_p \Delta x(k) + B_p \Delta u(k) \\ y_p(k) &= C_p \Delta x(k) + D_p \Delta u(k), \end{aligned} \tag{6.1}$$

where u is the control input; y_p is the output; x is the state variable vector and matrices A_p, B_p, C_p, and D_p are system matrix, input control matrix, output matrix and feedthrough matrix. In receding horizon control, current information of the plant is required for prediction and control implying that the input $u(k)$ cannot directly affect the output $y_p(k)$ at the same time. Therefore, $D_p = 0$ in the plant model. The state space model for a SISO process is re-written:

$$\begin{aligned} x(k+1) &= A_p \Delta x(k) + B_p \Delta u(k) \\ y_p(k) &= C_p \Delta x(k), \end{aligned} \tag{6.2}$$

where $A_p \in \Re^{n_1 x n_1}$ is n_1 x n_1 system matrix, $B_p \in \Re^{n_1 x 1}$ is n_1 x 1 input matrix, $C_p \in \Re^{1 x n_1}$ is 1 x n_1 output matrix. The vector $x(k) = \left[x_1(k), x_2(k), x_3(k), ..., x_{n1}(k)\right]^T \in \Re^{n_1}$ is state column vector while $u(k)$ and $y_p(k)$ are single input and output respectively.

In order to formulate suitable model for control derivation with integral action, the standard state space model in (6.2) is augmented as follows:

$$\Delta u(k+1) = u(k+1) - u(k),$$

$$\Delta x(k+1) = x(k+1) - x(k) = A_p \Delta x(k) + B_p \Delta u(k).$$

It can easily be observed that $\Delta u(k)$ is an input signal into the system and not $u(k)$. Therefore, a new state variable vector is defined as follows:

$$x(k) = \begin{bmatrix} \Delta x(k)^T & y(k) \end{bmatrix}^T.$$

Considering output equation:

$$\Delta y_p(k+1) = y_p(k+1) - y_p(k) = C_p x(k+1) - C_p x(k)$$
$$\Delta y_p(k+1) = C_p A_p \Delta x(k) + C_p B_p \Delta u(k)$$
$$\Delta y_p(k+1) = C_p \Delta x(k+1).$$

Therefore:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} A_p & 0_{n1}^T \\ C_p A_p & I \end{bmatrix} \begin{pmatrix} \Delta x(k) \\ y(k) \end{pmatrix} + \begin{bmatrix} B_p \\ C_p B_p \end{bmatrix} \Delta u(k) \qquad (6.3)$$

$$y_p(k) = [0_{n1} \;\; 1] \begin{pmatrix} \Delta x(k) \\ y_p(k) \end{pmatrix}$$

where $x_1 = \Delta x(k+1);\; x_2 = \Delta y_p(k+1).$

In summary, if $A = \begin{bmatrix} A_p & 0_{n1}^T \\ C_p A_p & I \end{bmatrix}$, $B = \begin{bmatrix} B_p \\ C_p B_p \end{bmatrix}$, $C = [0_{n1} \;\; 1]$, augmented state space model can therefore be stated as follows:

$$x(k+1) = Ax(k) + B\Delta u(k)$$
$$y_p(k) = Cx(k),$$

where: $x(k) = \begin{bmatrix} \Delta x(k)^T & y(k) \end{bmatrix}^T$.

This augmented state space model [*A*, *B*, *C*] is used for prediction of future states.

Let the prediction horizon be represented as '*P*' and control horizon be '*N*'. Given that $P \geq N$, future states can be predicted as follows:

$$\hat{x}(k+1) = Ax(k) + B\Delta u(k)$$

$$\Rightarrow \hat{x}(k+2) = A^2 x(k) + AB\Delta u(k) + B\Delta u(k+1)$$

This implies that the generic term:

$$\hat{x}(k+P) = A^P x(k) + A^{P-1}B\Delta u(k) + ... + A^{P-N}B\Delta u(k+N-1). \quad (6.4)$$

Similarly, the future output variables can be predicted using:

$$\hat{y}_p(k+1) = CAx(k) + CB\Delta u(k)$$

$$\Rightarrow \hat{y}_p(k+2) = CA^2 x(k) + CAB\Delta u(k) + CB\Delta u(k+1)$$

$$\Rightarrow \hat{y}_p(k+P) = CA^P x(k) + CA^{P-1}B\Delta u(k) + ... + CA^{P-N}B\Delta u(k+N-1). \quad (6.5)$$

The prediction equation can therefore be summarised as given in (6.6):

$$\hat{Y}_p = F_p \hat{X} + G\Delta U \quad (6.6)$$

where: $F_p = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^P \end{bmatrix}_{1xP}$; $\hat{Y}_p \in \Re^{P\times 1} = \left[\hat{y}_p(k+1), \hat{y}_p(k+2), ..., \hat{y}_p(k+P)\right]^T$

$$\Delta U \in \Re^{N\times 1} = \left[\Delta u(k), \Delta u(k+1), \Delta u(k+2), ..., \Delta u(k+N-1)\right]^T$$

$$G \in \Re^{P\times N} = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{P-1}B & CA^{P-2}B & CA^{P-3}B & ... & CA^{P-N}B \end{bmatrix}_{PxN}$$

The cost function to be minimised is given by J:

$$J = \sum_{i=1}^{P} \left\| R_s - \hat{Y}_p \right\|_2 Q + \sum_{j=1}^{N} \left\| \Delta U \right\|_2 R \quad (6.7)$$

where:

$$R_s = [111...1]^T_{1\times N} r(k)$$

$r(k)$ = desired reference signal

$\hat{Y}_p$ = Predicted or estimated output

R = weighting vector for the control effort

Q = state weighting matrix and I = identity matrix

$\Delta U(k) = \left[\Delta u(k), ..., \Delta u(k+N-1)\right]^T$ is the incremental control vector.

A standard solution exists for this optimisation problem assuming no input constraint and taking derivative of J. Therefore, the optimal control signal is given by:

$$\Delta U_{mpc} = K_{mpc} e(k) = K_{mpc}\left(R_s - Fx(k)\right) \tag{6.8}$$

where: $e(k)$ represents error at sampling instant 'k'. R_s is the set-point data vector while K_{mpc} is the first column of the matrix $\left[G^T G + \gamma I\right]^{-1} G^T$; $\gamma =$ tuning weight.

This is the standard MPC control technique. Receding horizon method of implementation is usually followed to obtain feedback control. In this section, the cost function given in (6.7) is modified to obtain a new fractional order predictive controller design method. Integral error correction component (with fractional order) is included in the cost function and this improves robustness or disturbance rejection property of the proposed control scheme compared to conventional MPC controller. However, implementation is done by receding horizon control technique by applying only the first column of control input and re-calculating each prediction matrix coefficient at every sampling instant. This ensures feedback control.

6.2 Review of Predictive PID Controller Based on GPC

The first attempt to build some form of integral and derivative structure into model based predictive controllers was successful as it led to design of predictive PID controllers. Today, this idea has been well accepted industrially as predictive PID controllers are now commercially available e.g. Honeywell's range of predictive PID controllers. These set of PID controllers generally retain the design objective of conventional PID controllers but with future set-point information incorporated in the

design process (Katebi & Moradi, 2001). It also retains constraint handling capability of MPC control paradigm as well as inherent MIMO capability of MPC.

Incremental model can be used to derive a predictive PID control law for a system without restriction to model order. Discrete form of integer order PID controller is presented as:

$$u(k+i)=\sum_{i=0}^{P}\left[K_p e(k+i)+K_i\sum_{n=1}^{k}e(n+i)+K_d\left(e(k+i)-e(k+i-1)\right)\right]$$

where: P is the prediction horizon. In their proposed design scheme, the predictive PID controllers consist of P parallel PID controllers where the number of PID controllers (P) equals the number of sampling instances within the prediction horizon (Katebi & Moradi, 2001). Therefore, at $P > 0$, the predictive PID control system produces similar predictive capability like MPC controllers. Control signal matching method is used to select optimal PID gains and value of P that yields best approximation to MPC control signal solution. At $P = 1$, conventional discrete PID controller is obtained.

Also: $u(k-1)=K_p e(k+i-1)+K_i\sum_{n=1}^{k-1}e(n+i)+K_d\left(e(k+i-1)-e(k+i-2)\right).$

Therefore, incremental model of the control signal is given as:

$$\Delta u(k+i)_{pid}=K_p\left(e(k+i)-e(k+i-1)\right)+K_i e(k+i)+K_d\begin{pmatrix}e(k+i)\\-2e(k+i-1)+e(k+i-2)\end{pmatrix}.$$

This can be represented in matrix form as:

$$\Delta U(k+i)_{pid}=K_{pid}\left[r(k+i)-y(k+i)\right]$$

where:

$$K_{pid}=\begin{bmatrix}k_p & k_i & k_d\end{bmatrix}\begin{bmatrix}0 & -1 & 1\\0 & 0 & 1\\1 & -2 & 1\end{bmatrix}$$

Reference signal is assumed to be zero: $r(k+i)=0$

$$\Rightarrow\ \Delta u(k+i)_{pid}=-K_{pid}Y(k+i)$$

where future output $Y(k+i)=\begin{bmatrix}y(k+i-2) & y(k+i-1) & y(k+i)\end{bmatrix}^T\in\Re^{3\times1}.$

The authors recommend generalised predictive control (GPC) algorithm for tuning the controller based on controlled autoregressive and integrated moving average (CARIMA) model of the plant i.e. $A(z^{-1})y(k) = B(z^{-1})\Delta u(k-1)$ where $y(k)$ and $\Delta u(k)$ are the output and incremental control input sequence of the plant.

Coefficients A and B are polynomials in the backward shift operator z^{-1}. These polynomials are usually defined in the form given below:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + ... + a_{Na} z^{-Na}$$
$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + ... + b_{Nb} z^{-Nb}$$

For future set-point prediction:

$$Y(k+i) = G_i \Delta\hat{u}(k) + F_i y_0(k) + G_i' \Delta u_0(k) \tag{6.9}$$

where G_i, F_i and G_i' are prediction matrices obtained from plant's CARIMA model coefficients i.e.

$$G_i = \begin{bmatrix} g_{i-3} & ... & g_0 & 0 & 0 \\ g_{i-2} & g_{i-3} & \cdots & g_0 & 0 \\ g_{i-1} & g_{i-2} & g_{i-3} & \cdots & g_0 \end{bmatrix}; \; F_i = \begin{bmatrix} f_{(i-2)1} & f_{(i-2)2} & \cdots & f_{(i-2)(Na+1)} \\ f_{(i-1)1} & f_{(i-1)2} & \cdots & f_{(i-1)(Na+1)} \\ f_{i1} & f_{i2} & \cdots & f_{i(Na+1)} \end{bmatrix}$$

$$G_i' = \begin{bmatrix} g'_{(i-2)1} & g'_{(i-2)2} & \cdots & g'_{(i-2)Nb} \\ g'_{(i-1)1} & g'_{(i-1)2} & \cdots & g'_{(i-1)Nb} \\ g'_{i1} & g'_{i2} & \cdots & g'_{iNb} \end{bmatrix}$$

Equation 6.9 is used to derive the kth control increment as follows:

$$\Delta U_{pid}(k) = -K \sum_{i=0}^{p} Y(k+i)$$
$$\Delta U_{pid}(k) = -K\left[\alpha_t \Delta\hat{u}(k) + F_t y_0(k) + G_t \Delta u_o(k)\right]$$

where: $\alpha_t = \sum_{i=0}^{p} G_i; \; F_t = \sum_{i=0}^{p} F_i; \; G_t = \sum_{i=0}^{p} G_i'$

$$\Rightarrow \Delta U_{pid}(k) = -(1 + K\alpha_t)^{-1} K\left[F_t y_0(k) + G_t \Delta u_0(k)\right] \tag{6.10}$$

Optimal values can be obtained by signal matching i.e. by minimising the objective function J:

$$J = \left\| \Delta U_{pid} - \Delta U_{mpc} \right\|_2$$

Optimal Controller gain can be obtained:

$$K_{opt} = K_0 \left[S_0 - \alpha K_g \right]^T \left(\left[S_0 - \alpha K_g \right] \left[S_0 - \alpha K_g \right]^T \right)^{-1} \tag{6.11}$$

where $S_0 = \left[F_t \;\; G_t \right]$

Individual PID parameters can be deduced by comparing (6.11) with K_{pid} matrix:

$$K_p = -2K_{opt}(1) - K_{opt}(2); K_I = K_{opt}(1) + K_{opt}(2) + K_{opt}(3); K_D = K_{opt}(1).$$

Several authors implemented these types of predictive PID controllers and demonstrated an improved control action and performance over conventional PID controllers tuned without considering future set-point information (Johnson & Moradi, 2006; Saeed, et al., 2010; Bouhajar, et al., 2015; Katebi & Moradi, 2001). Some of these predictive PID controllers turned out to be a close approximation to existing GPC or DMC algorithm as optimal gains are obtained for both SISO and MIMO process control cases (Katebi & Moradi, 2001). It is desirable to develop a similar control design methodology for FOPID controllers where future set-point information is incorporated to produce optimal (or sub-optimal) control action.

6.3 Proposed Fractional Order Predictive PI (FOPPI) Controller

6.3.1. A Variant of Fractional Predictive Control

In this section, discrete time model predictive control algorithm forms the foundation for a new control design and tuning procedure with predictive PI^{μ} controller objective function. However, this new controller is implemented using receding horizon control implementation technique. Firstly, discrete PI^{μ} control objective is formulated before minimising the objective function to obtain a fractional-order predictive PI controller.

6.3.2. PI^{μ} Controller in Discrete Time Domain

The structure of PI^{μ} controller in continuous time domain is well known. Frequency domain representation of the same FOPID controller as shown in (2.11) is also a well-known expression in Laplace domain. However, in discrete form, several authors have used different approximation methods to derive an approximate version of PI^{μ} controller. Most of these approximations are based on sampling the continuous

time signal and discretising the integral operator. Examples are Tustin approximation, forward Euler approximation, backward Euler approximation and Taylor series approximation. However, in this work, Grunwald-Letnikov (GL) definition given in (6.12) is used to handle fractional order derivatives and integrals. Although the GL definition results in infinite series, a truncated form given below in (6.13) is used extensively by several authors for deriving numerical solution. This gives a good approximation suitable for controller development. As a result, it is being used throughout this chapter to define fractional order integral or derivative.

$$D^{\alpha} f(t)_{t=k\tau} = \lim_{\tau \to 0} \tau^{-\alpha} \sum_{j=0}^{k} (-1)^{j} \binom{\alpha}{j} f(k\tau - j\tau) \tag{6.12}$$

where: $t > 0$; $n\text{-}1 < \alpha < n$; $n \in \mathbb{R}$

$\binom{\alpha}{j}$ represents binomial coefficient.

$$_{(t-L)}D_{t}^{\alpha} f(t) = \tau^{-\alpha} \sum_{j=0}^{N(t)} b_j f(t - j\tau) \tag{6.13}$$

where: b_j is binomial coefficient.

$$b_0 = 1;\ b_1 = \mu b_0; b_2 = \left(1 - \frac{1+\mu}{2}\right) b_1; b_j = \left(1 - \frac{1+\mu}{j}\right) b_{j-1}.$$

L = memory length; $N(t) = \min\left\{\left[t/\tau\right], \left[L/\tau\right]\right\}$.

Therefore, the discrete form of FOPI controller can be written as:

$$u_{pi}(k) = k_P e(k) + k_I T_s^{\mu} \sum_{j=0}^{k} b_j e(k - j) \tag{6.14}$$

where:

b_j is binomial coefficient.

$$b_0 = 1.\ b_1 = \mu b_0.\ b_2 = \left(1 - \frac{1+\mu}{2}\right) b_1.\ b_j = \left(1 - \frac{1+\mu}{j}\right) b_{j-1}.$$

k_P, k_I and μ = proportional gain, integral gain and fractional order respectively.

Equation (6.14) can be expressed in incremental form as follows:

$$\Delta u_{pi}(k) = u_{pi}(k) - u_{pi}(k-1).$$

$$u_{pi}(k-1) = k_P e(k-1) + k_I T_s^{\mu} \sum_{j=0}^{k-1} b_j e(k-1-j)$$

$$u_{pi}(k) - u_{pi}(k-1) = k_P \left(e(k) - e(k-1)\right) + k_I T_s^{\mu} \left(\sum_{j=0}^{k} b_j e(k-j) - \sum_{j=0}^{k-1} b_j e(k-1-j) \right)$$

$$\Delta u_{pi}(k) = k_P \left(e(k) - e(k-1)\right) + k_I T_s^{\mu} \left(e(k) - \sum_{j=1}^{k} \frac{1+\mu}{j} b_{j-1} e(k-j) \right)$$

$$\Delta u_{pi}(k) = k_P \Delta e(k) + k_I T_s^{\mu} \left(e(k) - \sum_{j=1}^{k} B_j e(k-j) \right) \tag{6.15}$$

where:

$$B_j = \frac{1+\mu}{j} b_{j-1} .$$

Replace the integral coefficient with a single term i.e. $K_I = k_I T_s^{\mu}$

Equation (6.15) is therefore re-written:

$$\Delta u_{pi}(k) = k_P \Delta e(k) + K_I \left(e(k) - \sum_{j=1}^{k} B_j e(k-j) \right) \tag{6.16}$$

Two distinct components are evident in (6.16) namely: a proportional term and a fractional order integral component. These two terms are sufficient to formulate a new cost function which is minimised using analytical method of optimisation in order to obtain the proposed Fractional Order Predictive PI (FOPPI) controller. $\Delta u_{pi}(k)$ is not implemented directly as given in (6.16). Direct implementation of FOPID controller has to be band-limited. In contrast, a model-based predictive control framework is used for implementation of the fractional order predictive PI controller. Consequently, control signal limits are defined inherently using constraint handling feature of MPC when necessary. Derivative component is unused because anticipatory action is inherently ensured by incorporating future set-point information in the predictive control framework.

6.3.3. Derivation of New FOPPI Controller

The first consideration in deriving a new fractional predictive control law is formulation of suitable model. This has been fully dealt with in section 6.1. Although, PID controller with fractional order is being considered, a full state space model of process is required instead of pseudo-state space model for control derivation. Therefore, augmented state space model derived previously is used for control formulation in this section. This is because systems and process models available are all defined using conventional lumped parameter models (i.e. ordinary differential equations with integer orders). Therefore, design of controllers for systems that are modelled with fractional-order dynamics is not included in the scope of study. Emphasis is placed on design of a typical distillation column for improved purity of distilled outcomes.

The augmented state space model is summarised below:

$$x(k+1) = Ax(k) + B\Delta u(k)$$

$$y(k) = Cx(k). \text{ where: } x(k) = \left[\Delta x(k)^T \;\; y(k)\right]^T.$$

$$A = \begin{bmatrix} A_p & 0_{n1}^T \\ C_p A_p & I \end{bmatrix};\; B = \begin{bmatrix} B_p \\ C_p B_p \end{bmatrix};\; C = [0_{n1} \;\; 1].$$

The control law is derived by minimising a cost function similar to DMPC procedure. However, the cost function is structured to reflect not only proportional error but fractional-integral error component too. This is expected to improve robust properties of the new control scheme, especially, disturbance rejection.

At a sample instant *k*, within a prediction horizon *P*, the aim of the control system is to bring the future predicted output $\hat{Y}(k+1)$ as close as possible to the expected set-point signal $Y_d(k+1)$ assuming the set point signal remains constant in the optimisation window. The task is therefore reduced to finding the optimum control parameter vector ΔU such that an error function between the set-point and the predicted output is minimised. Given:

P = prediction horizon; *N* = control horizon; the cost function J can be written as:

$$J = \sum_{i=1}^{P} \left\| Y_d(k+i) - \hat{Y}(k+i) \right\|_2 Q + \sum_{j=1}^{N} \left\| \Delta U(k+j-1) \right\|_2 R.$$

Replacing the error signal, the structured quadratic cost function to be minimised is of the form:

$$J = \sum^{P} \left[k_P \left(\Delta e(k+1) \right)^2 + K_I \left(e(k+1) \right)^2 + K_I \sum_{j=1}^{k} B_{j-1} \left(e(k+1-j) \right)^2 \right] + \gamma \sum^{N} \Delta u(k)^2 \quad (6.17)$$

where:

$\gamma =$ Constant control signal penalising factor. K_I component defines integral (fractional-order) error term i.e. $K_I = k_I T_s^{\mu}$.

Also, the future output $\hat{Y}(k+1)$ after P steps is the sum of free output response $\hat{Y}_p(k+1)$ and forced output $Y_f(k+1)$. That is:

$$\hat{Y}(k+1) = \hat{Y}_p(k+1) + Y_f(k+1).$$

Similarly, error signal is the difference between desired set point and predicted output:

$$e(k+1) = Y_d(k+1) - \hat{Y}(k+1).$$

Forced output is given below:

$$Y_f(k+1) = G\Delta u(k).$$

Therefore, error signal can be expressed as shown in (6.18).

$$e(k+1) = Y_d(k+1) - \hat{Y}_p(k+1) - G\Delta u(k). \quad (6.18)$$

The cost function given in (6.17) can be re-written in vector form as shown in (6.19) below:

$$J = k_P \Delta e^T \Delta e + K_I e^T e + K_I \sum_{j=1}^{k} B_j e_j^T e_j + \gamma \Delta u^T \Delta u, \quad (6.19)$$

where error signals are as expressed below:

$$e = Y_d - \hat{Y}_p - G\Delta u;$$
$$\Delta e = \Delta Y_d - \Delta \hat{Y}_p - G_p \Delta u;$$
$$e_j = Y_{dj} - \hat{Y}_{pj} - G\Delta u_j = Y_{dj} - \hat{Y}_{pj} - G_j \Delta u.$$

Error, output and control vectors are defined as follows:

$$e = \left[e(k+1), e(k+2), e(k+3), ..., e(k+P)\right]^T;$$
$$\Delta e = \left[\Delta e(k+1), \Delta e(k+2), \Delta e(k+3), ..., \Delta e(k+P)\right]^T;$$
$$e_j = \left[e(k+1-j), e(k+2-j), e(k+3-j), ..., e(k+P-j)\right]^T;$$
$$Y_d = \left[y_d(k+1), y_d(k+2), y_d(k+3), ..., y_d(k+P)\right]^T;$$
$$\Delta Y_d = \left[\Delta y_d(k+1), \Delta y_d(k+2), \Delta y_d(k+3), ..., \Delta y_d(k+P)\right]^T;$$
$$Y_{dj} = \left[y_d(k+1-j), y_d(k+2-j), y_d(k+3-j), ..., y_d(k+P-j)\right]^T;$$

$$\hat{Y}_p = \left[\hat{y}_p(k+1), \hat{y}_p(k+2), \hat{y}_p(k+3), ..., \hat{y}_p(k+P)\right]^T;$$
$$\Delta \hat{Y}_p = \left[\Delta \hat{y}_p(k+1), \Delta \hat{y}_p(k+2), \Delta \hat{y}_p(k+3), ..., \Delta \hat{y}_p(k+P)\right]^T;$$
$$\hat{Y}_{pj} = \left[\hat{y}_p(k+1-j), \hat{y}_p(k+2-j), \hat{y}_p(k+3-j), ..., \hat{y}_p(k+P-j)\right]^T;$$

$$\Delta U = \left[\Delta u(k), \Delta u(k+1), \Delta u(k+2), ..., \Delta u(k+N-1)\right]^T;$$
$$\Delta U_j = \left[\Delta u(k-j), \Delta u(k+1-j), \Delta u(k+2-j), ..., \Delta u(k+N-j-1)\right]^T.$$

In addition, the structure of prediction matrices (all three coefficients of Δu) are also given.

$$G = \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ g_3 & g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_N & g_{N-1} & g_{N-3} & \cdots & g_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_P & g_{P-1} & g_{P-2} & \cdots & g_{P-N+1} \end{bmatrix}_{P \times N}$$

Matrix *G* is based on the process model. So, it is the same *G*-matrix derived from plant's state space model as given in the introductory section of DMPC.

$$G=\begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{P-1}B & CA^{P-2}B & CA^{P-3}B & ... & CA^{P-N}B \end{bmatrix}_{P\times N}$$

Other prediction matrices (G_p and G_j) are derived from G as given below

$$G_p=\begin{bmatrix} g_1 & 0 & \cdots 0 \\ g_2-g_1 & g_1 & \cdots 0 \\ \vdots & \vdots & \vdots \\ g_P-g_{P-1} & g_{P-1}-g_{P-2} & \cdots \quad g_{P-N+1}-g_{P-N} \end{bmatrix}_{P\times N}$$

$$G_j=[G_1,G_2,....,G_k];\ j=1,2,3,\ ...,\ k$$

where

$$G_1=\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{P-1} & g_{P-2} & \cdots & \cdots & g_{P-N} \end{bmatrix}_{P\times N},$$

$$G_2=\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{P-2} & g_{P-3} & \cdots & \cdots & g_{P-N-1} \end{bmatrix}_{P\times N} ,...,\ G_k=0.$$

Substituting error signals in J as in (6.19) yield:

$$J=k_P[\Delta Y_d-\Delta\hat{Y}_p-G_p\Delta u]^T[\Delta Y_d-\Delta\hat{Y}_p-G_p\Delta u]+K_I[Y_d-\hat{Y}_p-G\Delta u]^T[Y_d-\hat{Y}_p-G\Delta u]$$
$$+\sum_{j=1}^{k}k_j[Y_{dj}-\hat{Y}_{pj}-G_j\Delta u]^T[Y_{dj}-\hat{Y}_{pj}-G_j\Delta u]+\gamma\Delta u^T\Delta u;$$

where $k_j=K_I\dfrac{1+\mu}{j}b_{j-1}=K_IB_j.$

The minimum point can be obtained by finding the gradient of *J* with respect to Δu .

$$\frac{\partial J}{\partial \Delta u} = 0;$$

$$\Rightarrow (k_P G_p^T \Delta Y_d + k_p G_p)(\Delta Y_d - \Delta \hat{Y}_p) + (K_I G^T + K_I G)(Y_d - \hat{Y}_p) + \sum_{j=1}^{k} k_j (G^T + G)(Y_{dj} - \hat{Y}_{pj})$$

$$= 2\gamma \Delta u + \Delta u(2k_P G_p^T G_p) + \Delta u(2K_I G^T G) + 2\Delta u \sum_{j=1}^{k} (k_j G_j^T G_j),$$

where $k_j = K_I \frac{1+\mu}{j} b_{j-1} = K_I B_j$.

Solving for minimum control increment:

$$\Delta u(k) = k_P G_p^T l \phi (\Delta Y_d - \Delta \hat{Y}_p) + K_I G^T l \phi (Y_d - \hat{Y}_p) + \sum_{j=1}^{k} k_j G_j^T l \phi (Y_{dj} - \hat{Y}_{pj}), \qquad (6.20)$$

where: matrix $l = [1, 0, 0, ..., 0]^T$ is *N* x 1.

$$\phi = \left(k_P G_p^T G_p + K_I G^T G + \sum_{j=1}^{k} (k_j G_j^T G_j) + \gamma I \right)^{-1}.$$

Equation (6.20) describes the new fractional predictive PI control law for the state space plant. It embeds fractional-order PI controller structure and retains the predictive capability of MPC algorithm. During implementation, at each sampling instant, only the first sample of control sequence is taken to form an incremental optimal control signal. All prediction matrices are re-calculated at the next sampling instant. Although prediction horizons and control signal penalising factor can be set as tuning parameters, only proportional and integral gain are recommended in this algorithm as tuning parameters for any necessary adjustment. This is because horizons and input penalising factor are expected to be properly pre-selected during initial control design phase.

6.3.4. Handling of Constraints

One attractive feature of using model-based predictive control narrative for control design is the ability to handle constraints in a very structured way. Constraints can be readily formulated into the cost function thereby turning the control design problem into a quadratic programming problem (Katebi, et al., 2001). Several

quadratic programming routines are available for solving constrained optimisation problem. The most common constraint is on the amplitude of input control signal allowable in a system. This type of constraint can be defined to handle input saturation problems. Therefore, the control design narrative can compensate for practical actuator saturation problems in real-time plants. Other types of constraints that may be incorporated include input rate constraint and output constraints. In this thesis, unconstrained process control examples are given for both scalar and multivariable systems.

6.3.5. SISO Simulation Example: Double Integrator Process Control

Since distillation columns are multivariable systems in nature, it is desirable to illustrate the implementation of this controller firstly, with a SISO system because of simplicity before considering multivariable distillation examples. Consider a double integrator plant (SISO system) given by:

$$x(k+1) = A_p x(k) + B_p u(k)$$
$$y(k) = C_p x(k)$$

where:

$$A_p = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad B_p = \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix}; \quad C_p = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

Let the sampling period be 0.01*s* and prediction horizon chosen to be: $P = 7$. Also, $N = 7$ and all weighting matrices chosen as unity identity matrices of proper dimension. Small values of k_P and k_I i.e. $0 < [k_P, k_I] > 1$ can be used to tune the plant until a properly damped response is obtained. In this example, k_I and k_P equal to 0.00351 and 0.7×10^{-4} are chosen with fractional order set to 0.7 in order to get a properly damped response. These small values are chosen as very small fractions by trial and error until satisfactory response is obtained. Optimisation is unconstrained too.

The resultant (incremental) control matrix is given:

$$\Delta U = 10^{-17} x \begin{bmatrix} -0.502 & -0.502 & -0.251 & -0.1255 & -0.0628 & -0.0157 & -0.002 \\ 0.1391 & 0.1391 & 0.0695 & 0.0348 & 0.0174 & 0.0043 & 0.0005 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.002 & 0.002 & 0.001 & 0.0005 & 0.0003 & 0.0001 & 0.0000 \end{bmatrix}$$

The figure below shows step response and corresponding unconstrained control signal with the proposed fractional predictive control algorithm. It yields excellent set-point tracking as shown in Fig. 6.3.

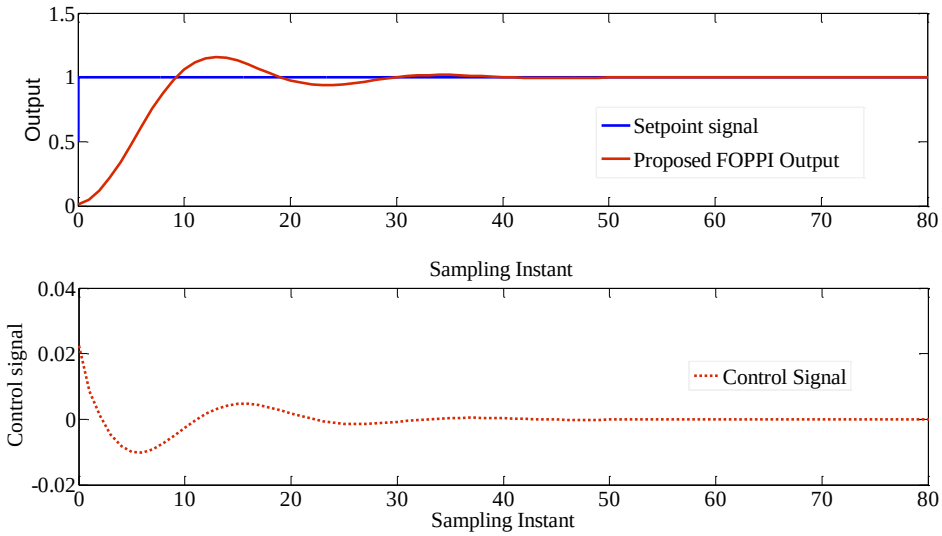


Fig. 6.3 Step response diagram

6.3.6 Disturbance Rejection: Non-Minimum Phase Control Comparison

A second example is given for SISO application of this predictive controller. Here, the system contains RHP zeros. Consider a non-minimum phase process control example where process model is described using state space with matrices:

$$A_p = \begin{bmatrix} -0.0217 & -0.3141 \\ 0.3141 & 0.7630 \end{bmatrix};\ B_p = \begin{bmatrix} 0.3141 \\ 0.2364 \end{bmatrix};\ C_p = \begin{bmatrix} -1 & 2 \end{bmatrix}.$$

The sampling period T_s is selected as 1 second. Prediction horizon chosen to be 7 and fractional order of 0.9.

The number of optimisation steps is chosen to be 7 and all weighting matrices chosen as unity identity matrices of proper dimension. Obtained incremental control matrix is given:

$$\Delta U = 10^{-12} x \begin{bmatrix} -0.5539 & -0.7005 & -0.6584 & -0.5142 & -0.3281 & -0.149 & -0.0252 \\ 0.1401 & 0.1772 & 0.1666 & 0.1301 & 0.0830 & 0.0377 & 0.0065 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.002 & 0.0003 & 0.0003 & 0.0002 & 0.0001 & 0.0001 & 0.0000 \end{bmatrix}$$

k_I and k_P equal to 0.15 yield excellent set-point tracking as illustrated in Fig. 6.4.

In order to demonstrate comparative benefit, DMC is used as a baseline with weight $R = 0.9I$. The resultant (incremental) DMC control signal:

$$\Delta U_{mpc} = 10^{-13} x \begin{bmatrix} 0.1503 \\ 0.0190 \\ 0.0479 \end{bmatrix}$$

The applied control signal is updated ($u_1 = u_0 + \Delta u_1$). Fig. 6.5 shows disturbance rejection property of these two algorithms. It is simulated over a longer timescale with a 25% step disturbance introduced at k = 50 seconds. Figure *6.3* also compares applied control signal. It can be observed that DMC controller only starts to act after effects of disturbance are already evident in the system. On the contrary, proposed FOPPI controller acts immediately and rejects the step disturbance signal. This improved disturbance rejection property is due to integral action of the new FOPPI control scheme. A high gain integral controller, in conventional sense, is known to cancel out effects of static non-linearity, parametric uncertainty and random disturbances. So, the proposed FOPPI control scheme retains robustness property of an integral controller. This extra robust feature gives FOPPI controller a major comparative advantage over conventional DMC algorithm. Also, FOPPI controller uses smaller effort as control input of the proposed fractional-order predictive controller is found to be within the range: $0 < u < 0.8$. This is smaller than the DMC signal as shown in Fig. 6.5.

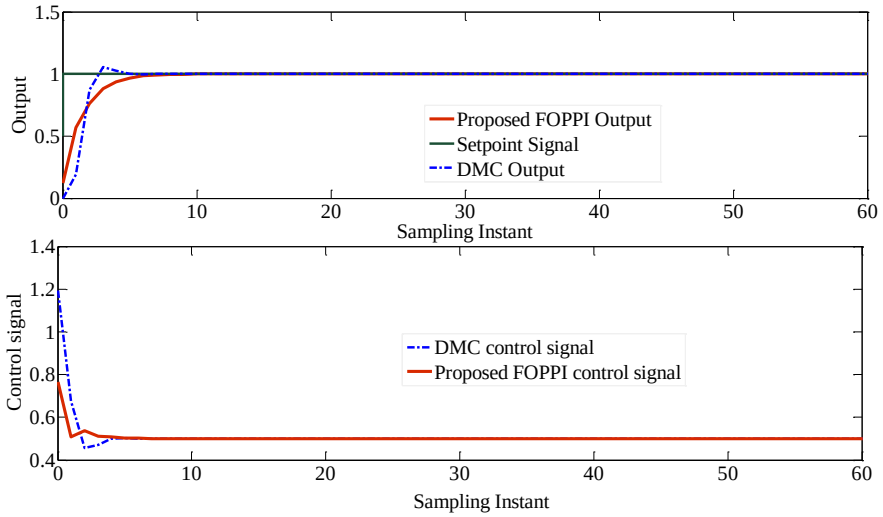


Fig. 6.4 Step response comparison

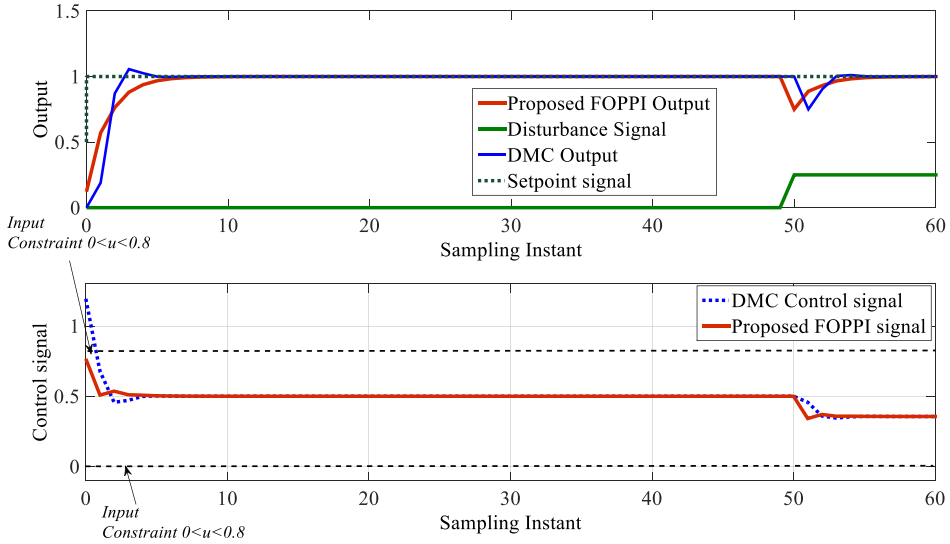


Fig. 6.5 Comparison: 10% disturbance (green) is introduced at k=50s.

6.4 Multivariable Case: Control of Distillation Process

The first step in extending this new fractional-order predictive PI controller design algorithm to MIMO system is to derive state space model of a multivariable plant. This is readily achieved using state space model because the only significant difference when compared to SISO case is in the dimension of relevant matrices. The proposed method is also able to counteract multivariable interaction in a very intuitive way using suitable predicting matrices. In formulating MIMO state space model, let:

m_1 = number of inputs,

m_2 = number of outputs,

n_1=number of states in the system.

Assumptions:

It is assumed that number of inputs (m_1) equals number of outputs (m_2) as only square systems are considered. So, m square process is considered where $m_1 = m_2 = m$. Also, assuming that this m-square system has n_1 states, each of the measurable output can be independently controlled without any steady state error using a similar FOPPI control scheme outlined for SISO state space control system. In addition, it is also assumed that noise sequence in the system is negligible. Therefore, the augmented state space system earlier derived is suitable for control design although with a commensurate increase in system dimension.

$$\begin{pmatrix} \Delta x(k+1) \\ y(k+1) \end{pmatrix} = \begin{bmatrix} A_p & 0_{n1}^T \\ C_p A_p & I_{mxm} \end{bmatrix} \begin{pmatrix} \Delta x(k) \\ y(k) \end{pmatrix} + \begin{bmatrix} B_p \\ C_p B_p \end{bmatrix} \Delta u(k)$$

$$y(k) = [0_{n1} \quad I_{mxm}] \begin{pmatrix} \Delta x(k) \\ y(k) \end{pmatrix}$$

where:

Matrix $A_p \in \Re^{n_1 x n_1}$ is n_1 x n_1 system matrix

Matrix $B_p \in \Re^{n_1 x m_1}$ is n_1 x m_1 input matrix

Matrix $C_p \in \Re^{m_2 x n_1}$ is m_2 x n_1 output matrix

I_{mxm} is identity matrix with dimension m_2 x m_2

O_{n1} is zeros matrix with dimension m_2 x n_1

$\Delta x(k) \in \Re^{n_1}$ is the state column vector

$\Delta u(k) \in \Re^{m_1}$ is the input column vector

$y(k) \in \Re^{m_2}$ is the output column vector

In summary, the augmented state space model used for MIMO predictive control is given:

$$x(k+1) = Ax(k) + B\Delta u(k)$$
$$y(k) = Cx(k). \text{ where: } x(k) = \left[\Delta x(k)^T \; y(k)\right]^T$$

$$A = \begin{bmatrix} A_p & 0_{n1}^T \\ C_p A_p & I_{mxm} \end{bmatrix};\; B = \begin{bmatrix} B_p \\ C_p B_p \end{bmatrix};\; C = [0_{n1} \;\; I_{mxm}].$$

6.4.1. Controllability, Observability and Poles of Augmented State Space System

It has been established that the proposed predictive control design scheme does not directly use original plant's state space model but controller synthesis is based on an augmented state space model. Consequently, it is important to investigate the poles (eigenvalues), controllability (state reachability) and observability of the augmented model before designing a control system. This is required in order to ensure desired closed loop performance of the designed control system. Let the characteristic equation of the augmented state space system be $C(\lambda)$.

$$\therefore C(\lambda) = \begin{vmatrix} \lambda I - A_p & 0_{n1}^T \\ -C_p A_p & (\lambda - 1) I_{mxm} \end{vmatrix}$$
$$\Rightarrow (\lambda - 1)^m \left|\lambda I - A_p\right| = 0 \qquad (6.21)$$

where λ represents eigenvalues.

Lower triangular matrix property is used to solve (6.21) for eigenvalues of the system. The determinant of a block lower triangular matrix equals the product of determinant of diagonal matrices. Therefore, eigenvalues of augmented state space model equals the union of eigenvalues of original plant (A_p, B_p and C_p) and the m number of eigenvalues, $\lambda = 1$. This embeds integral action in the resultant predictive control system as m number of integrators are embedded in the augmented state space model.

In addition, controllability of the augmented state space model is presented. Since augmented model introduces integral modes, it is desirable to analyse all sufficient conditions required for stability of these integral modes. Bay's sufficient condition for stability is achieved if minimal realisation condition of plant model is

realised. This is because, he proved that a minimal realisation is both controllable and observable.

In many practical multivariable processes, model information is available as step response data or identified Laplace transfer function models. When converting these models to state space for discrete predictive control design, a minimal realisation is required and simple codes exist for such model conversions and minimal realisation. For instance, the following Matlab program produces minimal realisation of a plant identified as *"Gmin",* converts it to state space and discretized it using zero order hold method with sampling time of 1*s* to get (A_p, B_p, C_p, D_p) model.

```
"Gmin=ss(g,'min');
T=1;
[Ac,Bc,Cc,Dc]=ssdata(Gmin);
[Ap,Bp,Cp,Dp]=c2dm(Ac,Bc,Cc,Dc,T,'zoh');"
```

According to Wang (2009), a good definition of minimum realisation of a system is given as follows:

"*A realisation of transfer function G(z) is any state space triplet* (A_p, B_p, C_p) *such that* $G_p(z) = C_p(zI - A_p)^{-1}B_p$. *If such a set* ($A_p$, B_p C_p) *exists, then* $G_p(z)$ *is said to be realisable. A realisation* (A_p, B_p C_p) *is called a minimal realisation of a transfer function if no other realisation of smaller dimension of the triplet exists*" (Wang, 2009).

If (A_p, B_p C_p) system is both controllable and observable, and has minimal realisable transfer function $G_p(z) = C_p(zI - A_p)^{-1}B_p$, then the augmented system with transfer function: $G(z) = \frac{z}{z-1}G_p(z)$ is both controllable and observable if and only if the plant model $G_p(z)$ has no zero at $z = 1$. Therefore, the sufficient condition for controllability and observability of augmented system is stated below in condition (6.22).

$$G(z) = \frac{z}{z-1}G_p(z) \tag{6.22}$$

This provides a method of analysing controllability and observability of augmented MIMO state space system. If Laplace transfer function model is used directly in Laplace domain, minimal realisation of $G(s)$ can be verified using (6.23).

$$G(s) = \frac{C_p \times \text{adj}(sI - A_p)B_p}{\left|sI - A_p\right|} \tag{6.23}$$

Every pole of $G(s)$ has to be a zero of $\left|sI - A_p\right|$ which is also eigenvalue of A_p. This is the condition for controllability and observability. The pole-polynomial is given by the term $C(s)$ where: $C(s) = \left|sI - A_p\right|$ .This is minimal realisation because zeros of $\left|sI - A_p\right|$ will not cancel any pole of $C_p \left(sI - A_p\right)^{-1} B_p$.

6.4.2. Solution to MIMO FOPPI Control problem

FOPPI control law is derived by minimising a cost function similar to the SISO derivation. At a sample instant *k*, within a prediction horizon *p*, the control system is designed to minimise error between desired set-point and predicted future output to the least possible value. It is assumed that the set point signal remains constant in the optimisation window as in the SISO case.

Let: P = prediction horizon

N = control horizon

In conventional DMPC, cost function may be defined as J such that:

$J = \sum_{i=1}^{P} \left\| Y_d(k+i) - \hat{Y}(k+i) \right\|_2 Q + \sum_{j=1}^{N} \left\| \Delta U(k+j-1) \right\|_2 R$ where each term retains their usual meaning as given previously in SISO case.

Substituting for error signal using FOPI controller error function:

$$J = \sum^{P} \left[k_P \left(\Delta e(k+1)\right)^2 + K_I \left(e(k+1)\right)^2 + K_I \sum_{j=1}^{k} B_{j-1} \left(e(k+1-j)\right)^2 \right] + \gamma \sum^{N} \Delta U^2 \qquad (6.24)$$

where:

$\gamma =$ control signal weight.

The future output after P steps is the sum of free output response and forced output.

$\hat{Y}(k+1) = \hat{Y}_p(k+1) + Y_f(k+1)$

where $\hat{Y}(k+1)$ is predicted future output; $Y_f(k+1)$ is the forced output and $\hat{Y}_p(k+1)$ is free response.

Similarly, error signal is the difference between desired set-point and predicted output:

$e(k+1) = Y_d(k+1) - \hat{Y}(k+1)$.

The forced output is given below:

$Y_f(k+1) = G\Delta u(k)$. where: $G \in \Re^{m_1 N \times m_1 N}$ is an *mN* x *mN* coefficient matrix.

$$\Delta U = \left[\Delta u(k), \Delta u(k+1), \Delta u(k+2), ..., \Delta u(k+N-1)\right]^T$$

Therefore, error signal can be expressed as given below in (6.25).

$$e(k+1) = Y_d(k+1) - \hat{Y}_p(k+1) - G\Delta u(k) \tag{6.25}$$

The cost function given in (6.17) can be re-written in vector form as done before:

$$J = k_P \Delta e^T \Delta e + K_I e^T e + K_I \sum_{j=1}^{k} B_j e_j^T e_j + \gamma \Delta u^T \Delta u \tag{6.26}$$

where error signals are as defined in previous section:

$$e = Y_d - \hat{Y}_p - G\Delta u$$
$$\Delta e = \Delta Y_d - \Delta \hat{Y}_p - G_p \Delta u$$
$$e_j = Y_{dj} - \hat{Y}_{pj} - G_j \Delta u$$

Some definitions of vectors and matrices involved in (6.26) are given:

$$e = \left[e(k+1), e(k+2), e(k+3), ..., e(k+P)\right]^T$$
$$\Delta e = \left[\Delta e(k+1), \Delta e(k+2), \Delta e(k+3), ..., \Delta e(k+P)\right]^T$$
$$e_j = \left[e(k+1-j), e(k+2-j), e(k+3-j), ..., e(k+P-j)\right]^T$$
$$Y_d = \left[y_d(k+1), y_d(k+2), y_d(k+3), ..., y_d(k+P)\right]^T$$
$$\Delta Y_d = \left[\Delta y_d(k+1), \Delta y_d(k+2), \Delta y_d(k+3), ..., \Delta y_d(k+P)\right]^T$$
$$Y_{dj} = \left[y_d(k+1-j), y_d(k+2-j), y_d(k+3-j), ..., y_d(k+P-j)\right]^T$$

$$\hat{Y}_p = \left[\hat{y}_p(k+1), \hat{y}_p(k+2), \hat{y}_p(k+3), ..., \hat{y}_p(k+P) \right]^T$$
$$\Delta\hat{Y}_p = \left[\Delta\hat{y}_p(k+1), \Delta\hat{y}_p(k+2), \Delta\hat{y}_p(k+3), ..., \Delta\hat{y}_p(k+P) \right]^T$$
$$\hat{Y}_{pj} = \left[\hat{y}_p(k+1-j), \hat{y}_p(k+2-j), \hat{y}_p(k+3-j), ..., \hat{y}_p(k+P-j) \right]^T$$

$$\Delta U = \left[\Delta u(k), \Delta u(k+1), \Delta u(k+2), ..., \Delta u(k+N-1) \right]^T$$
$$\Delta U_j = \left[\Delta u(k-j), \Delta u(k+1-j), \Delta u(k+2-j), ..., \Delta u(k+N-j-1) \right]^T$$

Also, prediction matrices are as given previously but with greater dimension:

$$G = \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ g_3 & g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_N & g_{N-1} & g_{N-3} & \cdots & g_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_P & g_{P-1} & g_{P-2} & \cdots & g_{P-N+1} \end{bmatrix}_{P \times mN}$$

Again, matrix G is obtainable from process model:

$$G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{P-1}B & CA^{P-2}B & CA^{P-3}B & ... & CA^{P-N}B \end{bmatrix}_{P \times mN} .$$

Other prediction matrices (G_p and G_j) are derived from G:

$$G_p = \begin{bmatrix} g_1 & 0 & \cdots 0 \\ g_2 - g_1 & g_1 & \cdots 0 \\ \vdots & \vdots & \vdots \\ g_P - g_{P-1} & g_{P-1} - g_{P-2} & \cdots \quad g_{P-N+1} - g_{P-N} \end{bmatrix}_{P \times mN} ,$$

$$G_j = [G_1, G_2, ..., G_k]; \quad j=1,2,3, ..., k$$

where

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{P-1} & g_{P-2} & \cdots & \cdots & g_{P-N} \end{bmatrix}_{P\times mN},$$

$$G_2 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{P-2} & g_{P-3} & \cdots & \cdots & g_{P-N-1} \end{bmatrix}_{P\times mN} \quad ,..., \; G_k = 0.$$

Substituting error signals in *J* i.e. put (6.25) in (6.26) yields:

$$J = k_P[\Delta Y_d - \Delta\hat{Y}_p - G_p\Delta u]^T[\Delta Y_d - \Delta\hat{Y}_p - G_p\Delta u] + K_I[Y_d - \hat{Y}_p - G\Delta u]^T[Y_d - \hat{Y}_p - G\Delta u]$$

$$+\sum_{j=1}^{k} k_j[Y_{dj} - \hat{Y}_{pj} - G_j\Delta u]^T[Y_{dj} - \hat{Y}_{pj} - G_j\Delta u] + \gamma\Delta u^T\Delta u$$

The minimum point can be determined by finding the gradient of *J* with respect to Δu **:**

$$\frac{\partial J}{\partial \Delta u} = 0;$$

$$(k_P G_p^T\Delta Y_d + k_p G_p)(\Delta Y_d - \Delta\hat{Y}_p) + (K_I G^T + K_I G)(Y_d - \hat{Y}_p) + \sum_{j=1}^{k} k_j (G^T + G)(Y_{dj} - \hat{Y}_{pj})$$

$$= 2\gamma\Delta u + \Delta u(2k_P G_p^T G_p) + \Delta u(2K_I G^T G) + 2\Delta u\sum_{j=1}^{k}(k_j G_j^T G_j)$$

where $k_j = K_I\dfrac{1+\mu}{j}b_{j-1} = K_I B_j$.

Solving for minimum control increment:

$$\Delta u(k) = k_P G_p^T\phi(\Delta Y_d - \Delta\hat{Y}_p) + K_I G^T\phi(Y_d - \hat{Y}_p) + \sum_{j=1}^{k} k_j G_j^T\phi(Y_{dj} - \hat{Y}_{pj}) \qquad (6.27)$$

where

$$\phi = \left(k_P G_p^T G_p + K_I G^T G + \sum_{j=1}^{k} (k_j G_j^T G_j) + \gamma I \right)^{-1}.$$

Applying the receding horizon principle as before, only the first *m* columns is taken at each sampling instant to form incremental control signal.

6.4.3. Petroleum Distillation Column Control Example

Minh (2009) developed a state space model for a distillation column set up as part of a real-time petroleum refining and gas processing unit in Malaysia. The authors used deviation from set-points (deviation variables) to describe the column. The summary of the steady state data of the column and main streams measurements are available in the paper (Minh, 2009). Precisely, the input feed (condensate) is fed into a 14 - trayed column at the middle and the mixture is separated into two components: liquefied petroleum gas (LPG) as distillate (output - overhead product) while bottom product from the column is raw gasoline. At nominal operation without any disturbance or composition control system, the system is expected to reach at least 96.54% (y_D = 0.9654) purity of distillate and 3.75% (y_B = 0.0375) impurity in raw gasoline at bottom plate. However, this objective was not achieved when experimenting without composition control as there was sensitivity to feed stream variability. The authors implemented MPC controller for regulation of product's composition.

In addition, it is desirable to raise purity level in the column to 98% distillate purity and 0.02% raw gasoline impurity level $\left(y_D \geq 0.98;\ y_B \leq 0.02 \right)$.Therefore, there is a need for composition controller to force steady state values of the system to these desired output targets. Since the authors used deviation from set-point as variables, the distillate reference set-point will be required to be 0.0261 molar fraction for distillate and -0.0275 for bottoms product reference (r_1 = 0.0261; r_2 = -0.0275) in order to track 0.9654 purity of top product (by molar fraction) and molar fraction of 0.0375 impurity of bottoms product (Minh, 2009). Notice that the purity of bottoms product in reality is not negative but the negative r_2 is only a deviation from a set-point of 0.0375. If the controller outputs track these changes, it will force steady state values of distillate concentration to at least 97% purity level at top plate and 3% impurity at bottoms plate.

This was the requirement for the liquefied petroleum gas (top product) and gasoline (bottoms product).

State space model of the process is given:

$$x(k+1)=\begin{bmatrix}-0.5105 & 0\\ 0 & -0.5105\end{bmatrix}x(k)+\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\Delta u$$
$$y(k)=\begin{bmatrix}0.0021 & -0.0031\\ -0.0026 & 0.0037\end{bmatrix}x(k) \qquad (6.28)$$

Since the authors used deviation variables, the input and output variables are defined as follows:

$$x(k)=\begin{bmatrix}y_D(k)-\bar{y}_D\\ y_B(k)-\bar{y}_B\end{bmatrix};\ \Delta u(k)=\begin{bmatrix}L(k)-\bar{L}\\ V(k)-\bar{V}\end{bmatrix};\ y(k)=\begin{bmatrix}y_D(k)-\bar{y}_D\\ y_B(k)-\bar{y}_B\end{bmatrix}.$$

where the variables with a hat (i.e. $\bar{y}_D, \bar{y}_B, \bar{L}$ and $\bar{V}$) are steady state values. It can be readily observed that the nominal plant is asymptotically stable due to location of poles in Left Half Plane (LHP). A predictive controller is designed according to our proposed algorithm for this plant. Simulation studies are carried out with the nominal plant followed by perturbed plant model with perturbed state space model coefficient. This is given below:

$$A_p=\begin{bmatrix}-0.5 & 0\\ 0 & -0.75\end{bmatrix};\ B_p=\begin{bmatrix}1.5 & 0\\ 0 & 2.5\end{bmatrix}\ C_p=\begin{bmatrix}0.0021 & -0.0031\\ -0.0026 & 0.0037\end{bmatrix}.$$

Fig. 6.6 shows that controlled outputs (y_D and y_B) track the set-point for both output variables and will consequently regulate product composition of the column to expected levels of 97% distillate purity at top tray and 3% impurity level at bottom tray. This is achieved in the midst of feed variability disturbances and plant-model mismatches. $P=80;\ N=20;\ \lambda=0.0001.$ The rise time is less than 8 Seconds when properly tuned with $k_P=0.001; k_I=0.0002.$ These tuning parameters are chosen to obtain a smooth and properly damped response as earlier explained.

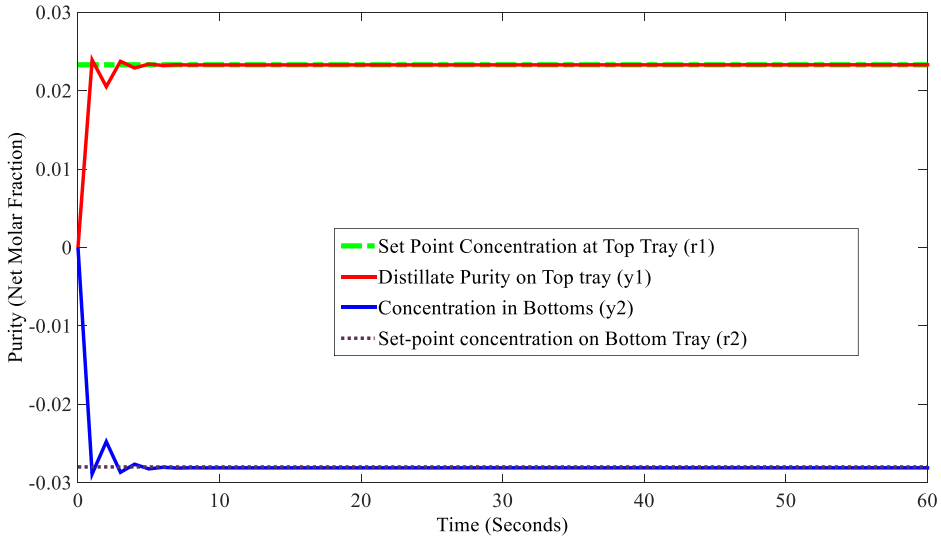


Fig. 6.6 Dual composition control of Minh distillation column.

6.4.4. A Three-by-Three Reactive Column Control Example

A second example is given to illustrate industrial application of proposed controller in distillation systems. This example is a very interactive three-by-three reactive column set up and studied by Karacan and Giwa (Giwa & S.Karacan., 2012). These authors developed state space model for an experimental column set up for esterification reaction between acetic acid and ethanol to produce an ester (ethyl acetate). The equilibrium-type esterification reaction is given in the equation below:

$$CH_3COOH + C_2H_5OH \overset{K_{eq}}{\rightleftharpoons} CH_3COOC_2H_5 + H_2O \tag{6.29}$$

Summary of steady state data of the column, main streams measurements and experimental description are available in the paper (Giwa & S.Karacan., 2012). A discretized state space model obtained using *Matlab's c2d* command is stated as follows:

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{pmatrix} = \begin{bmatrix} 0.9981 & 0.0024 & -0.0009 \\ -0.0034 & 0.9957 & -0.0008 \\ -0.0009 & -0.0052 & 0.9945 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{bmatrix} 0.0003 & 0.0013 & -0.0565 \\ -0.0005 & 0.0019 & -0.0818 \\ -0.0025 & 0.0067 & -0.2808 \end{bmatrix} \begin{pmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{pmatrix} \tag{6.30}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{bmatrix} -12.8258 & 4.0787 & 0.1685 \\ -3.7520 & -0.8266 & -1.9955 \\ -6.2674 & -17.6857 & -0.7129 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Composition of top distillate, segment draw-off and bottom plate product is inferred from temperature measurements at each tray in this experimental column. The output temperature of interest is a 70.75 ^{o}C for distillate tray. This was found to be

very important because, in the production of ethyl acetate from this esterification reaction, 70.75 ^{o}C is the optimal temperature for desired product quality of over 98% purity. Steady state values of nominal plant outcome without control system was found to be 69.89 ^{o}C for distillate product, 70.81 ^{o}C for reaction segment and 87.99 ^{o}C for bottom product. Therefore, a composition controller is required to drive the output temperature at the top of the column to 70.75 ^{o}C. Simulation was carried out with the following input (set-point changes) in the column: 0.86 step change in top temperature (from 69.89 ^{o}C steady state value to 70.75 ^{o}C), 0.5 ^{o}C change in reaction segment and 1.5 ^{o}C change in bottoms product temperature.

Fig. 6.7 shows that controlled outputs simultaneously track the set-point for all three output variables and will consequently regulate product composition of the column to expected level of 98% distillate purity. $P = 200;\ N = 50;\ \lambda = 0.0009$. The rise time is less than *10* s when properly tuned.

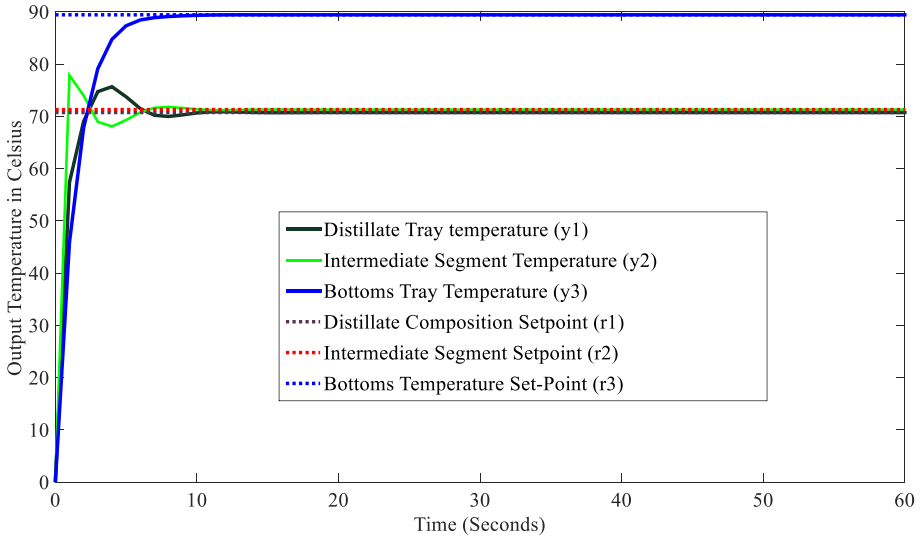


Fig. 6.7 Composition control: Step response

6.4.5. Comparison of all Three Fractional-order controllers on OR Column

FOPPI controller is implemented on the *OR* column and results are illustrated in Fig. 6.8. In addition, Fig.6.9 – Fig.6.13 shows comparison of both control actions and control signals of all three proposed fractional-order controllers. The FOPPI controller yield better control actions than FOPID controllers tuned with ultimate parameters or using IMC-BLT method although with a higher comparative computational cost. For instance, settling time has been significantly reduced by 60%

for y_2 and by 96% for y_3 and by 75% for y_1 in the *OR* column when FOPPI controller is implemented compared to IMC-FOPI and critical point FOPID controllers. These results have been tabulated in Table 6.1.

Table 6.1 *Settling Time Comparison on OR Column - Unperturbed*

Step Change	y1	y2	y3
FOPI (Critical Point)	120*m*	75*m*	150*m*
FOPI (IMC-BLT)	100*m*	150*m*	150*m*
FOPPI (Predictive)	25*m*	50*m*	5*m*
OPI (Baseline)	50*m*	200*m*	75*m*

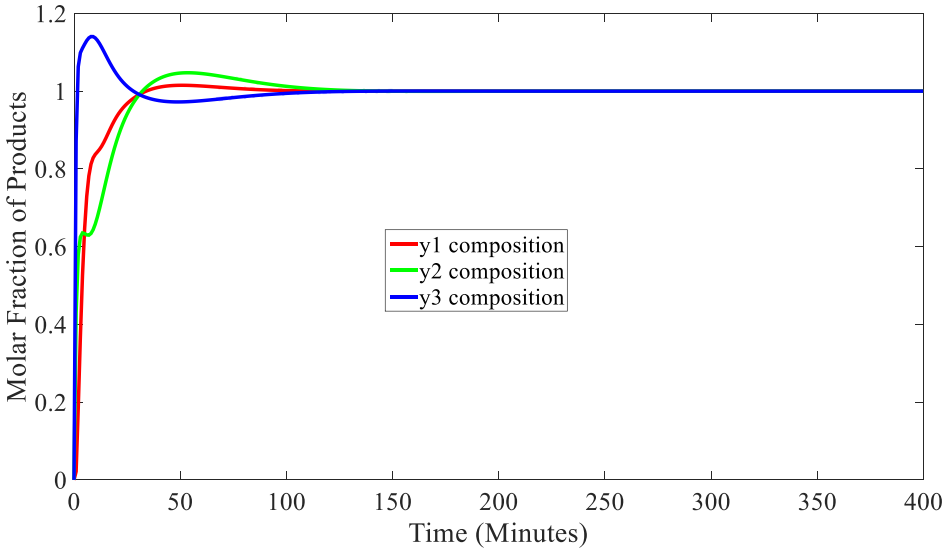


Fig. 6.8 OR distillation column control - predictive approach

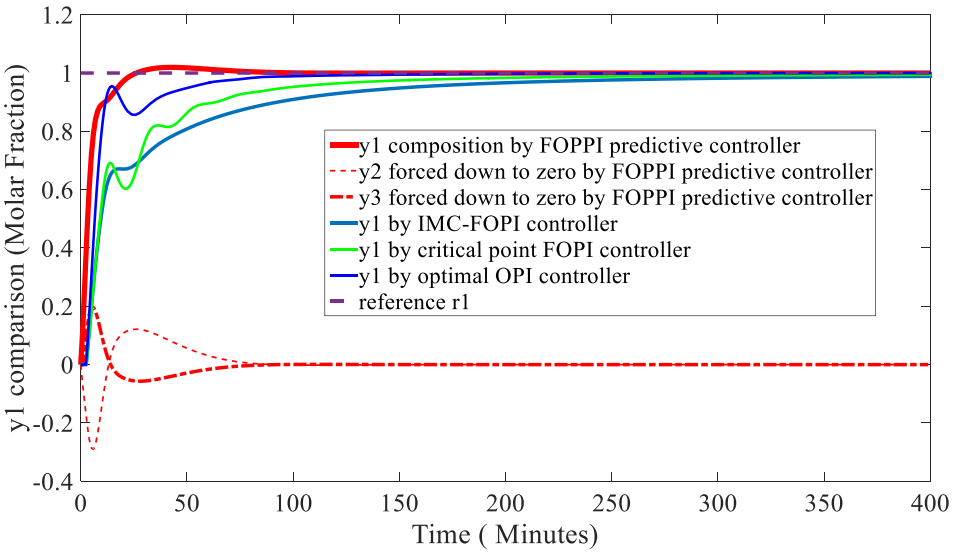


Fig. 6.9 Comparison of all proposed controllers on OR column – y1 loop

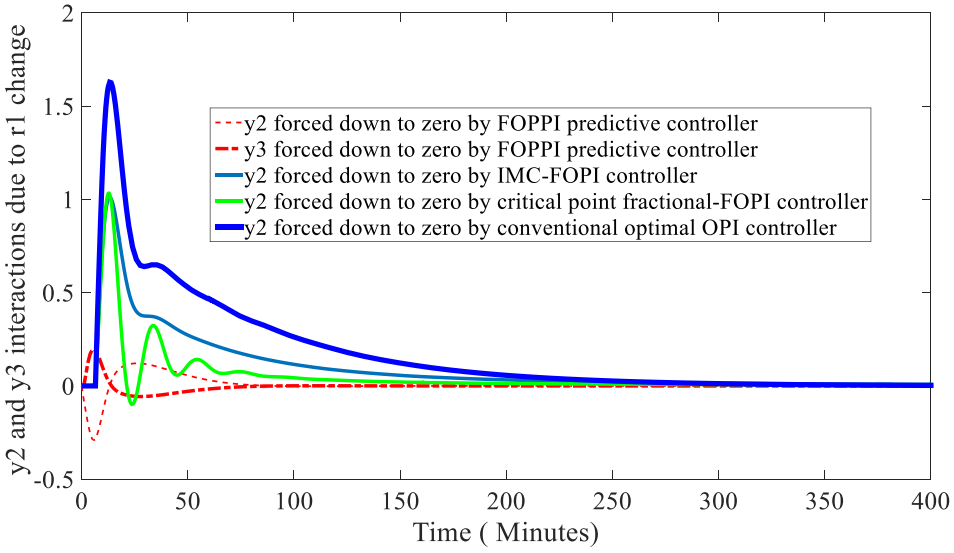


Fig. 6.10 Effect of all proposed controllers on interactions in loop 1

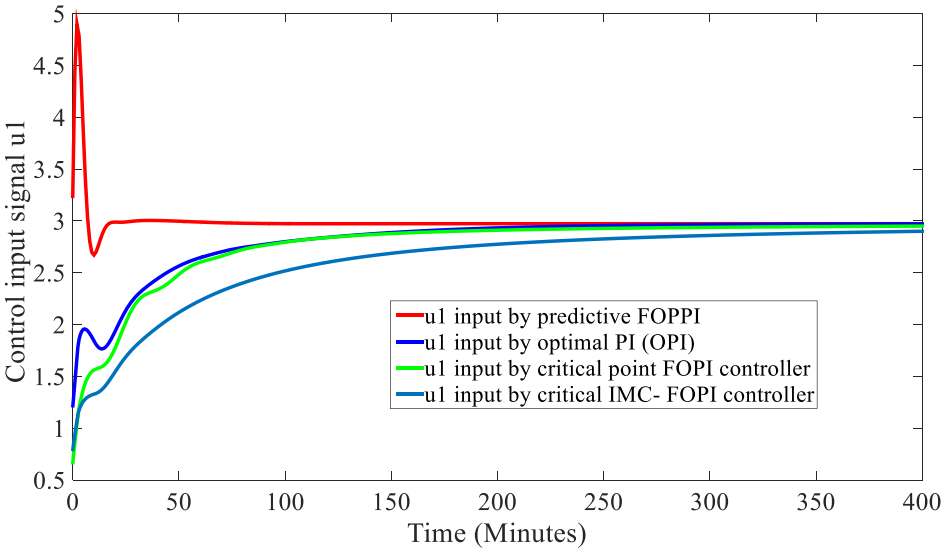


Fig. 6.11 Control signal comparison u1 - all proposed controllers

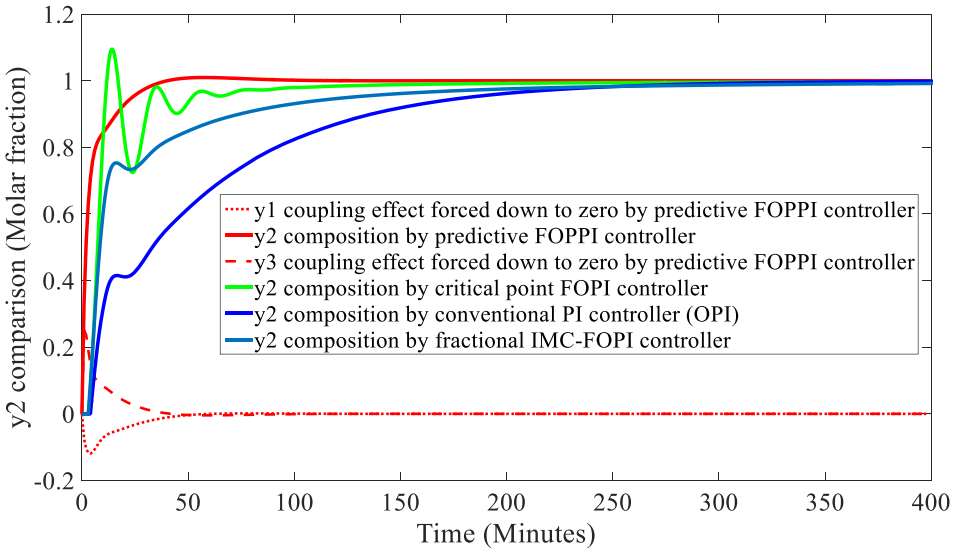


Fig. 6.12 Comparison of all proposed controllers on OR column – y2 loop

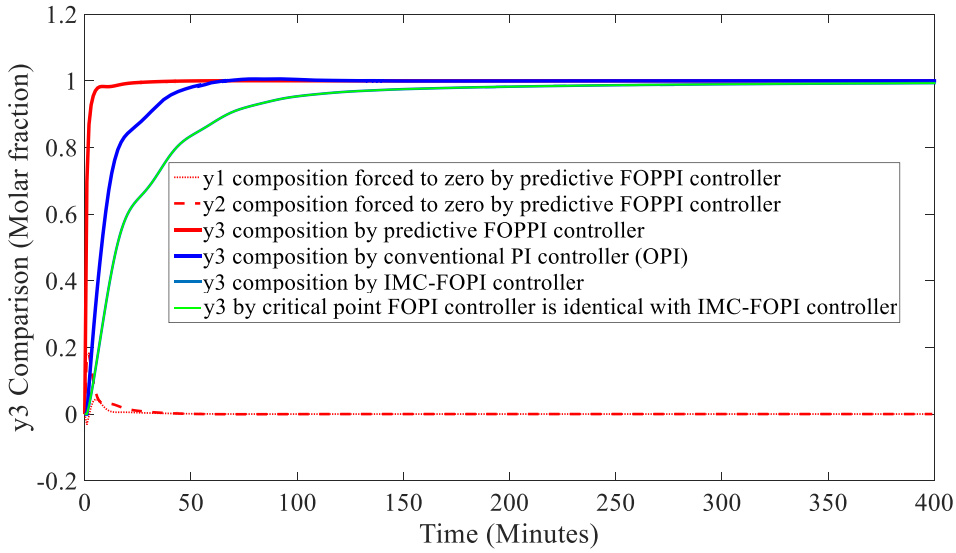


Fig. 6.13 Comparison of all proposed controllers on OR column – y3 loop

6.5 Conclusions

In conclusion, the proposed FOPPI controller ensures excellent set-point tracking and zero steady state error after the shortest possible time when compared to other fractional order PID controllers. It is also more robust than conventional MPC algorithm with better disturbance rejection capability. The proposed FOPPI control scheme assumes availability of state space model for prediction of future states and output. The control law is formulated as an analytical solution to an unconstrained

optimisation of an objective function. It is also possible to incorporate constraints during optimisation. This will turn the optimisation problem into a quadratic programming problem. The method has been extended to *OR* distillation column for the purpose of comparison as well as other distillation columns. It should be noted that *OR* model is given as Laplace transfer function model and this may make discrete time predictive control approach unattractive to it. FOPPI controller design is based on a discrete time state space model. However, Matlab's "*c2d*''command is available for discretization and "*tf2ss*'' for conversion of transfer function model to state space. A unit step response simulation is carried out with the new predictive controller in order to compare with results of fractional order PI controllers introduced in chapters four and five.

Fig. 6.8 shows reduction in both rise time and settling time when implementing predictive control of the highly interactive three-by-three *OR* column. Fig.6.9 – Fig.6.13 demonstrate these comparative advantages when there are no perturbations. The settling time for IMC and critical FOPI controllers are over 100 minutes for y_1 and y_2 while y_3 is about 150 minutes. This has been significantly reduced by more than 60% for y_2 (50 minutes) and by more than 96% for y_3 (5 minutes) and by 75% for y_1 (25 minutes) by the FOPPI controller.

Chapter 7

Summary, Conclusions and Future Work

At the beginning of this work, pertinent questions were asked about the feasibility of using controllers with fractional orders to obtain improved control actions for distillation systems as well as solving multivariable process control problems in general. These questions are re-summarised here:

- *How can PID controllers and Fractional Order PID (FOPID) controllers be designed and tuned to solve a linear multivariable control problem such as composition control of distillation column with multiple time delays?*
- *If FOPID controller is selected to solve a time-delayed multivariable distillation control problem, how can it be designed and tuned to yield 98% distillate's purity and 2% bottoms impurity in the face of disturbances like 20% fluctuation in feed flow rate?"*

Throughout the body of this thesis, these questions have been addressed and research objectives have been satisfied. New FOPID design and tuning rules have been identified and developed for composition control of distillation column and similar multivariable process control applications. These include the critical frequency point method reported in chapter four; the modified IMC-BLT-FOPI controller proposed in chapter five and the predictive (FOPPI) controller developed in chapter six for solving MIMO process control system problems. Results of all these three contributions clearly demonstrate robust stability as well as robust performance as comparative studies with optimal PI controllers have been presented too. Peer reviews of each of these proposed control design schemes have been greatly positive and three papers have been published as a direct outcome of this research effort. Itemised lists of contributions and papers are highlighted in the first chapter.

In this concluding chapter, the summary and major conclusions of this work are given in addition to suggestions for future research endeavours. Considering the highly interactive OR distillation column, Minh Petroleum distillation column and Karacan/Giwa's three-by-three column as case studies, It should be noted that the

IMC-BLT-FOPI controller proposed in chapter five and the FOPPI composition control system all yield 98% distillate's purity and 2% bottoms impurity in the face of disturbances like fluctuation in feed flow rate. This completely satisfy the aim and objectives of the research. In fact, with the FOPPI method, very fast response times are obtained even for relatively sluggish systems like distillation column. For instance, rise time of 8 *seconds* is achieved for MP column and 8 *minutes* for OR column.

7.1 Summary of Critical Point Method of Designing FOPID Controller

A new tuning method for FOPID controllers based on critical frequency point experiment has been given. This proposed design scheme presents new and simple formulae for proportional and integral gains when using FOPID controller without any need for optimisation. For a linear multivariable system, a decentralised multivariable controller is realised using these simple FOPID controller tuning rules in conjunction with sequential loop closing technique.

The control scheme is also suitable for SISO applications and in such cases, controller gains are obtained directly from ultimate point parameters using these tuning formulae. Sustained oscillation experiment is set up by using a proportional controller or relay. The value of the proportional gain that yields this sustained oscillation is recorded as the critical gain (k_u) while the period of oscillation is noted as ultimate period (τ_u). These two ultimate point parameters are used to calculate proportional and integral gains of FOPI controller.

If derivative component is required, a $PI^{\lambda}D$ controller or PID^{μ} controller structure can also be designed and tuned using critical frequency point experiment. A desirable point on the Nyquist curve is chosen as point *B* given by: $r_B e^{j(\pi+\varnothing_B)}$ where r_B and $\varnothing_B$ are magnitude and phase at the chosen point i.e. $r_B = 0.29, \varnothing_B = 46°$. These controllers are obtained such that the design point on Nyquist plot i.e. $r_A e^{j(\pi+\varnothing_A)}$ is relocated to point *B* where $r_A = 1/k_u$ and $\varnothing_A = 0.$

FOPID controllers are known to be robust in terms of performance. Considering the same *OR* distillation system, the proposed controller is simulated with significant disturbance signals introduced. A 20% step disturbance signal (*d*1) is

introduced at time t = 500 minutes (feed flow changes) while a 30% step disturbance signal is simultaneously introduced at t = 600 minutes as changes in feed temperature ($d2$). Simulated results show effectiveness of FOPID controllers in disturbance rejection as well as output regulation (See Fig. 4.14 - Fig. 4.20).

The proposed controller (FOPI) is also compared with an optimum PI controller proposed by Ogunnaike and Ray (Ogunnaike & Ray, 1983) under exact conditions. Time domain responses as well as robust stability analyses have been carried out with both controllers.

7.1.1. Performance and Robustness of Control Action

Multivariable FOPI controllers have been designed for composition control of a three - by - three distillation column (*OR* model) using ultimate frequency point measurements. The proposed method yields control actions with good compromise between robustness and performance. A method of judging robustness based on inclusion of input and output multiplicative uncertainty models have been presented. When compared with a conventional PI controller optimally tuned by Ogunnaike and Ray, the proposed FOPI gives a more robust control action if considering multiplicative input uncertainty. With multiplicative output uncertainty model, optimum PI produces a slightly larger area of robust stability. However, one demerit of this proposed control scheme is the need for frequency response experiment for determination of critical frequency point. This is overcome by using a parametric-model based method for design of FOPID controllers.

7.2 Summary of IMC-BLT Design Method

Although FOPID controller gains can easily be determined using sustained oscillation experiment, conducting the experiment itself can prove to be a limitation in certain applications because it drives the process to the verge of instability. Consequently, a new method of designing and tuning FOPI controllers is presented based on internal model and the IMC controller settings tuned using BLT approach. It is also beneficial to use this method in a fully cross-coupled or centralised configuration if the number of loops is limited to two like in TITO systems because, the cost of implementation of centralised controllers is directly proportional to the number of loops in the system (Besta & Chidambaram, 2015). However, a

decentralised controller is more suitable for applications where there are many variables and more than two control loops. In a decentralised IMC-BLT controller as proposed in this present work, only diagonal elements of the system model are considered thereby limiting the cost of implementation of sensors (measurement devices) and actual controllers. This is effective for systems with three or more number of loops.

The proposed control design scheme is developed with set-point tracking as a primary design objective. It is similar to Skogestad's IMC rule for conventional SISO PID controllers. Considering, a FOPDT model, simple tuning formulae are derived using the principle of IMC for FOPI controllers. For SOPDT models, derivative component is added as derived controller is of the FOPID structure. Finally, these IMC settings are fine tuned to yield biggest log-modulus in frequency domain. For instance, in the *OR* distillation control example, a detuning parameter (*F*) is selected to get a BLT value of 6 dB. If the system is a TITO process, *F* is varied such that a BLT value of 4 dB is obtained.

7.2.1. IMC-BLT: Performance and Robustness of Control Action

Simulation studies carried out on the highly interactive *OR* distillation model (with two disturbance inputs of 20% and 30% variations introduced as $d1$ and $d2$) shows better performance compared to conventional PI controller (OPI). With the proposed controller design scheme, step responses in all three loops show good set-point tracking, zero overshoot and good disturbance rejection. Step response comparison with optimum PI controller shows that this IMC FOPI controllers give better responses because recovery time in top composition's loop has been reduced by 50% (from 10 minutes to 5 minutes for *y1*); by 87.5% for *y2* (from 80 minutes to 10 minutes) with step change in respective inputs *r1* and *r2* while *y3* does not show any comparative improvement (See Fig. 5.7 – Fig. 5.12). This is similar to results obtained with the critical frequency point FOPID controllers.

In conclusion, the proposed IMC-BLT FOPI controller gives design engineer an easier method of estimating the gains of a FOPI controller for process control. Since it yields better stability bounds with respect to both input and output multiplicative

uncertainty models compared to OPI controller, it is more suitable for many process control applications (See Fig. 5.13).

7.2 Summary of a New FOPPI Control Design Narrative

All previous methods of designing FOPID controllers given in this chapter do not consider future set-point information during control law formulation. In fact, FOPID controllers have been designed in continuous time form and by using Laplace transform definitions only. To incorporate future set-point information in formulation of control law, a new controller scheme (a Fractional Order Predictive PI-FOPPI-controller) has been proposed. The proposed controller is defined in discrete time using approximate Grunwald-Letnikov (GL) definition given in (6.12) to handle fractional-order derivatives and integrals.

One assumption made is availability of plant's state space model for control law derivation. Although, PID controller with fractional order is being considered, a conventional state space model is required instead of pseudo-state space model for control derivation. This is because most practical process model like the distillation column comes with integer order as explained in chapter 1. The conventional state space model is augmented with integrators before deriving prediction matrices, predicted future states and future outputs. This provides framework for derivation of a new FOPPI control law. The FOPPI cost function is defined to retain predictive proportional-integral control objective and the control law is obtained by minimising this objective function. This results in improved control actions especially, set-point tracking. Hybrid benefits of dynamic matrix control and robust benefit of FOPID controller are exploited in this novel FOPPI control scheme. These MPC benefits include excellent set-point tracking, constraint handling, and handling multivariable interaction. Conventional integral action allows for better disturbance rejection. These desirable objectives are retained in this new control scheme.

In terms of control system implementation, receding horizon principle is retained in the same manner as in DMC algorithm. Only the first m columns are taken at each sampling instant to form the incremental control signal where m is the number of inputs. At the next sampling instant, prediction matrices are re-calculated and control law is re-computed. In SISO case, only the first sample is taken at each

sampling instant to form the incremental control signal and applied to the process. The entire process of computing prediction matrices and control law is repeated at the next sampling instant.

7.2.1. Proposed Predictive Approach: Performance and Robustness

In conclusion, this FOPPI controller ensures that control specifications are not only met but exceeded. For instance, in both SISO and MIMO examples, excellent set-point tracking and zero steady state error are achieved after the shortest possible time compared with other fractional order PID controllers. It is also more robust than conventional MPC algorithm with better disturbance rejection capability. When there are set-point changes and/or disturbance inputs, the controller acts immediately to reject disturbance and maintain desired output levels and this faster response times has been consistently better than conventional MPC controller in all examples. Simulation studies have been carried out on distillation columns such as Giwa-Karacan reactive distillation column, Minh petroleum distillation column and *ORA* distillation column.

In composition control of the three-by-three Giwa-Karacan reactive distillation column using proposed FOPPI controller, composition of top distillate, segment draw-off and bottom plate product are inferred from temperature measurements at each tray in this experimental column. The output temperature of interest is 70.75 ^{o}C for the distillate tray. This was found to be very important because, in the production of ethyl acetate from this esterification reaction, 70.75 ^{o}C is the optimal temperature for desired product quality of over 98% purity. In absence of composition control system, steady state values of nominal plant outcome was found to be 69.89 ^{o}C for distillate product, 70.81 ^{o}C for reaction segment and 87.99 ^{o}C for bottom product. However, composition controller is found to drive the output temperature at the top of the column to 70.75 ^{o}C. Simulation was carried out with the following input (set-point changes) in the column: 0.86 step change in top temperature (from 69.89 ^{o}C steady state value to 70.75 ^{o}C), 0.5 ^{o}C change in reaction segment and 1.5 ^{o}C change in bottoms product temperature. Similar results are also obtained in other two columns with respect to set-point tracking, output regulation and disturbance rejection.

Considering the *OR* distillation column, interesting results have been obtained when comparing control actions of all three fractional order controllers along with their control effort as observed in Fig.6.9 – Fig.6.13. The FOPPI controller yield better control actions than FOPID controllers tuned with ultimate parameters or using IMC-BLT method although with a higher comparative computational cost. For instance, settling time has been significantly reduced by more than 60% for y_2 and by more than 40% for y_3 and by 37.5% for y_1 in the *OR* column when FOPPI controller is implemented compared to IMC-FOPI and critical point FOPID controllers.

7.3 Conclusions

In this thesis, fractional order PID controller design methods have been developed for a linear multivariable system. Two of these proposed methods are decentralised in architecture. One fractional order predictive PI control scheme has been proposed in a fully cross-coupled centralised configuration. These three control design narratives are proposed for the first time and are major contributions of this present work. Comparative studies have been carried out with conventional PI control schemes on three-by-three distillation columns such as:

- Ogunnaike and Ray column (*OR*),
- Giwa and Karacan reactive distillation column (*GK*),
- Minh Petroleum distillation column (*MP*).

The first method uses critical frequency point parameters to compute FOPID controller gains. Obtained tuning rules are extended to multivariable systems using sequential loop closing method. *OR* distillation model is considered and FOPID controller is designed for composition control in order to improve distillate quality. Performance is evaluated and robust stability is studied using inverse maximum singular value of sensitivity along with input/output multiplicative uncertainty models. Superior control actions of the proposed FOPID controller is observed as settling time (after disturbance signals are introduced) was reduced to 8 minutes for y_1; 40 minutes for y_2 and 36 minutes for y_3. This is better than conventional OPI controller. In addition, a greater region of robust stability is achieved with the FOPI controller when considering input multiplicative uncertainty model.

Another contribution of this work combines IMC and BLT to derive new tuning rules for FOPI controllers in a multivariable setting. Initial controller settings are obtained using tuning relationships derived based on internal process model. However, fine tuning of controller parameters is achieved by BLT tuning method. Although BLT is widely studied for conventional PID controller tuning, it is modified for tuning of IMC-FOPI controller using multiloop configuration. This new control scheme is implemented for composition control of distillation systems and it essentially overcomes the drawback of critical frequency point experiments associated with Ziegler-Nichols type algorithms. *OR* distillation column is selected for comparative studies. Control performance is judged in terms of set-point tracking using IAE and robust stability is studied using inverse maximum singular value of system's sensitivity along with input/output multiplicative uncertainty models.

The proposed IMC-FOPI controller yields superior control action compared to optimum PI controller tuned optimally by Ogunnaike and Ray (OPI). A greater region of robust stability is achieved with the FOPI controller when considering both input and output multiplicative uncertainty model compared to OPI controller. In addition, settling time (after disturbance signals are introduced) was reduced to 8 minutes for y_1; 10 minutes for y_2 but could not simultaneously improve y_3 which has recovery time of 140 minutes.

Thirdly, a new predictive fractional-order control design method is developed based on augmented state space model. Inherent robust properties of FOPID controller is retained without losing anticipatory action and other defining features of model-based predictive control scheme. So, constraint handling, incorporation of future set point in control law formulation as well as embedded integral action are beneficial features of this new control narrative. Simulation studies are carried out on *MP* distillation column and *GK* distillation column to demonstrate effectiveness in terms of set-point tracking, output regulation and disturbance rejection.

Considering composition control study on *MP* column, the controlled outputs (y_D and y_B) tracks the set-point for both output variables and regulates product composition of the column to expected level of 98% distillate purity at top tray and 2% impurity level at bottom tray. This is achieved in the midst of feed variability

disturbances and plant-model mismatches. Excellent set-point tracking has been achieved with rise time of about 8*s*. On simulating the same control scheme with *GK* distillation column, composition controller regulates the output temperature at the top of the column to 70.75 ^{o}C which corresponds with 98% distillate purity. In conclusion, this centralised FOPPI control narrative yields faster response and generally better performance than critical frequency point or IMC method but at the expense of increased computational overhead.

For instance, considering settling time for all three controllers on the OR column when there are no perturbations, the FOPPI controller yield better control actions than FOPID controllers tuned with ultimate parameters or using IMC-BLT method although with a higher comparative computational cost. The settling time has been significantly reduced by 60% for y_2 and by 96% for y_3 and by 75% for y_1 in the *OR* column when FOPPI controller is implemented compared to IMC-FOPI and critical point FOPID controllers.

7.4 Future Work

There are several opportunities for future work based on motivations drawn from this present research study.

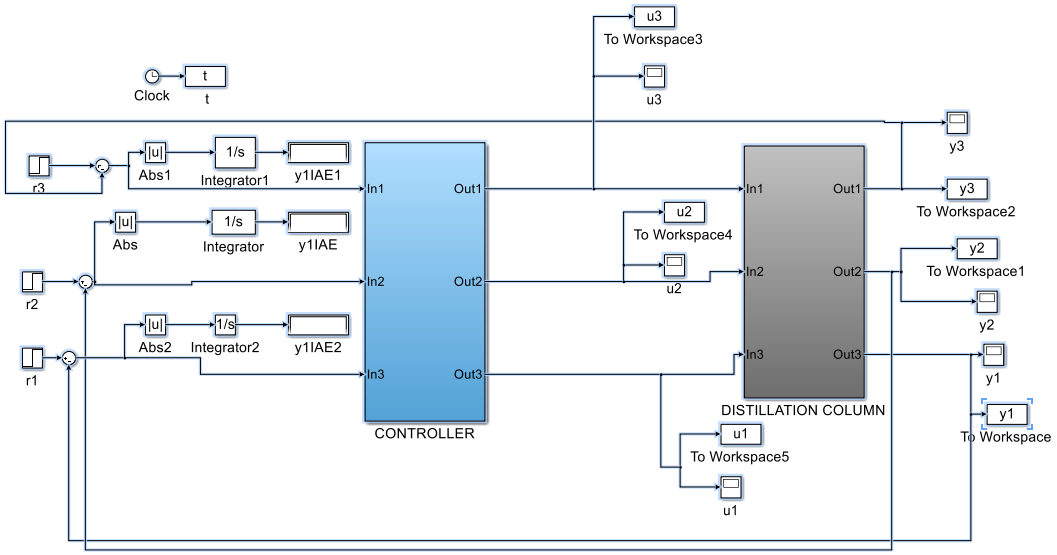
- In this thesis, multivariable FOPID controllers have been designed for distillation column models which are open loop stable in terms of dynamic characteristic. Design methods for fractional order controllers may be studied for process control systems exhibiting open-loop unstable dynamics. In addition, a generic model including systems with five or more inputs and outputs can be studied along with non-square systems. Some pre-treatment may be required before extending standard control design schemes such as MPC technique to such generic process control systems.
- Design of fractional order controllers can be studied for full non-linear process models. In addition, some controller design methods, studied in this thesis such as MPC, can be modified for non-linear systems e.g. Nonlinear Model-based Predictive Controllers (NMPC). This provides a background for some form of comparative study of nonlinear controllers for real-time processes like distillation columns, waste management systems or industrial boilers. These

processes exhibit strong nonlinear characteristic. In this thesis, linear time invariant controllers have been synthesised from linearised models of distillation columns at nominal operating points.

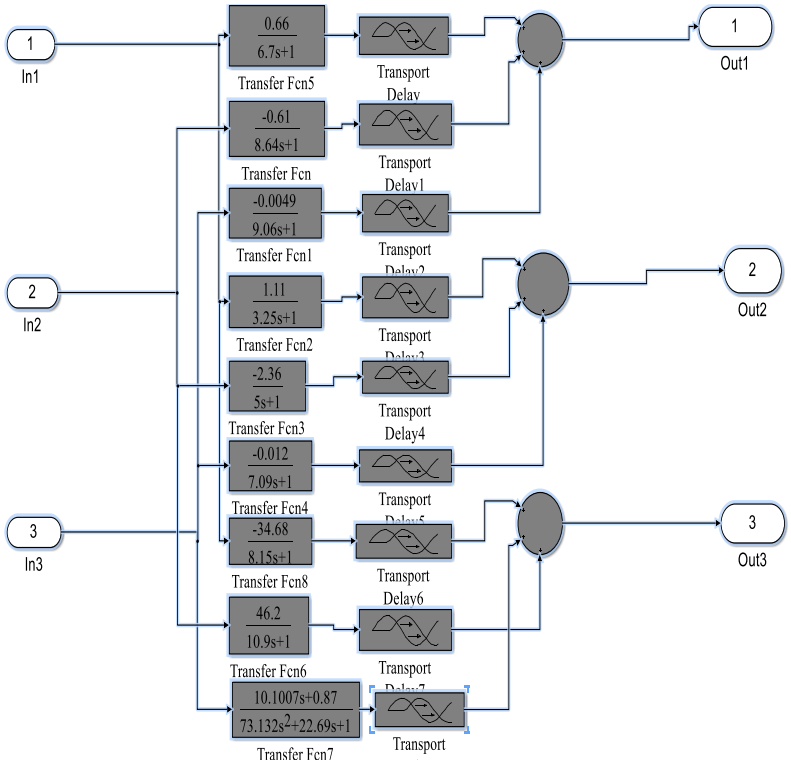
- This thesis has been limited to FOPID controller design for distillation columns where only ideal mixtures are considered for separation. Azeotropic mixtures, which are inherently more difficult to separate, have not been covered within the scope of study. Distillation of azeotropic mixtures require specialised separation towers which can only be modelled using complex mathematical models to reflect complex thermodynamic properties of such systems. This is different from regular or ideal mixtures. For instance, the order of boiling fractions may also vary with product's composition and this require further research work.
- It is also possible to design multivariable FOPID controllers for interconnected-distillation systems using similar methods proposed in this thesis, thus adding relevance for future research and practical applications. For instance, by connecting several distillation columns in series, the top product flow in one column can can act as a bottom feed to the next column and this interconnected distillation system may be studied for separation of multicomponent mixtures which are quite prevalent in industry. In this present work, a single distillation column has been considered rather than interconnected columns because, the knowledge of the dynamics and control of a typical column can be stretched to apply to interconnected columns but with some modifications (Arlt, 2014).

Appendices - A: Simulink Models of Distillation Column Control Systems

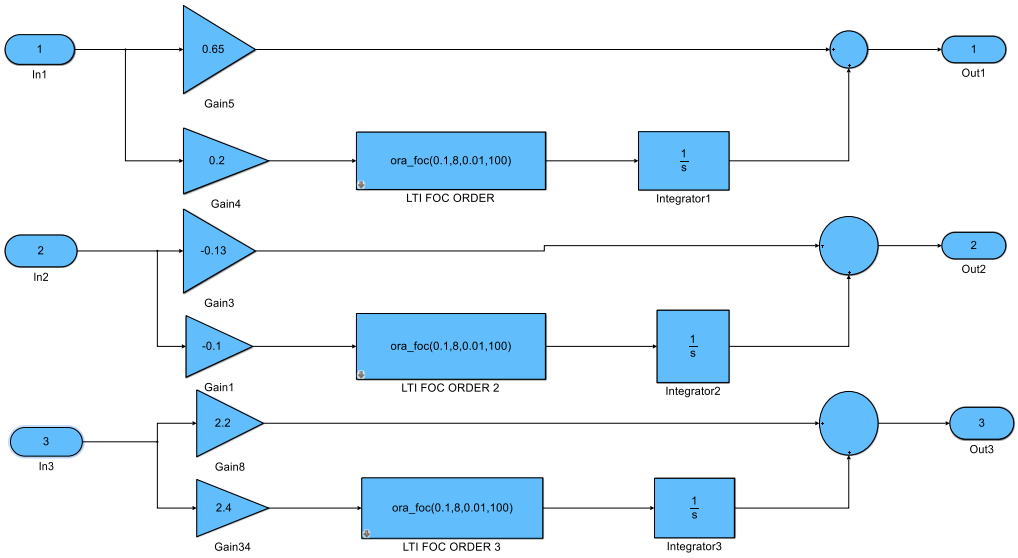
Appendix - A1: *OR* Distillation Column Control Simulink Block Diagram



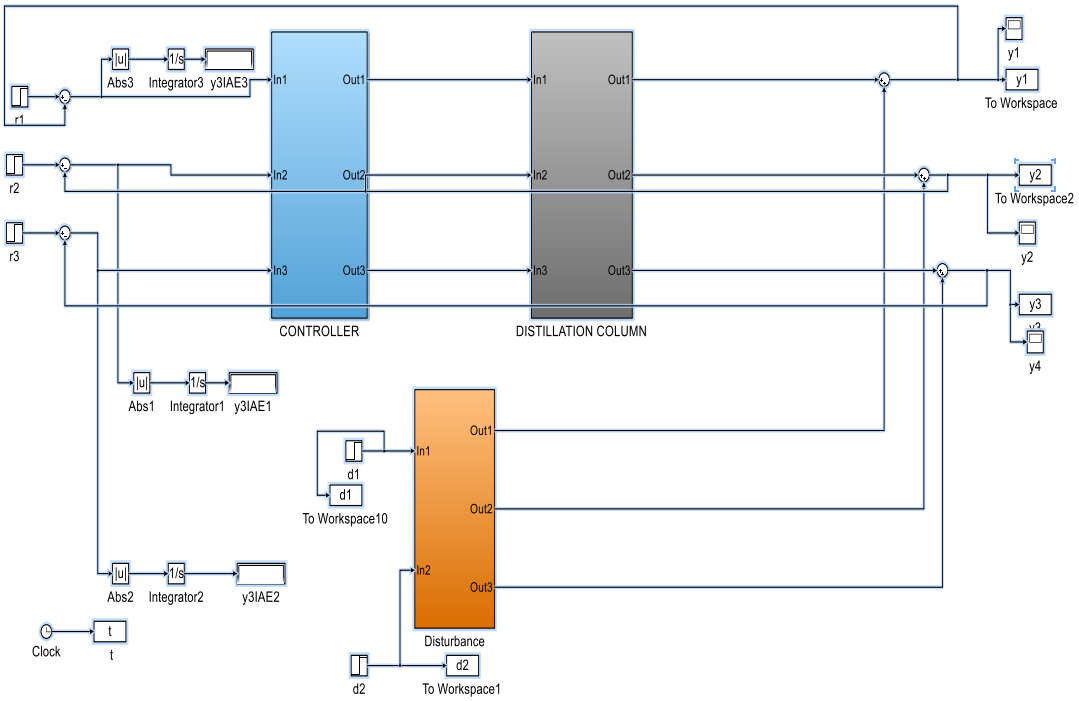
Detailed Diagram of Distillation Column Sub-system



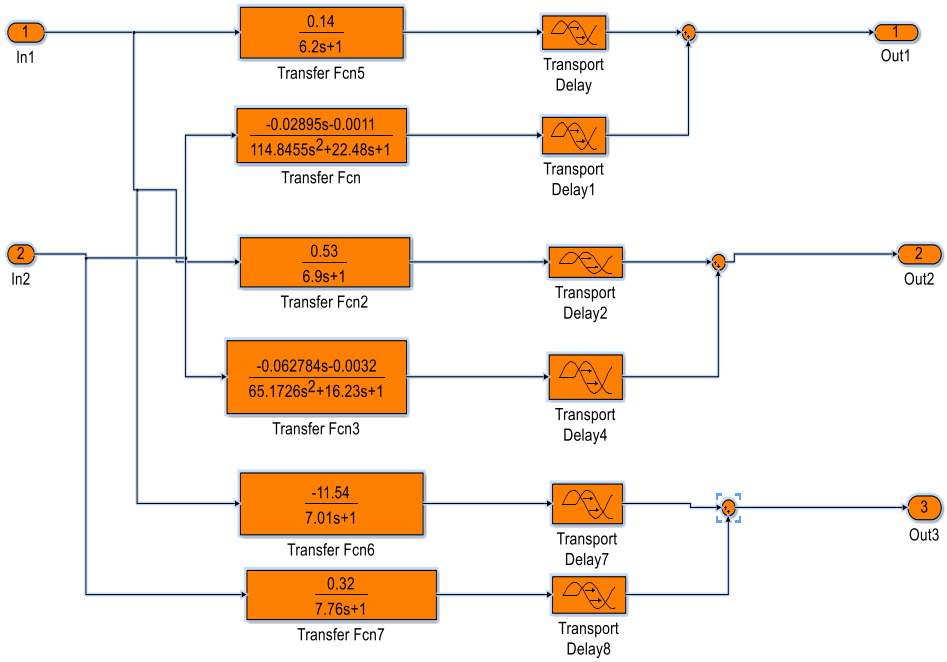
Detailed Diagram of Controller Sub-system in Example 4



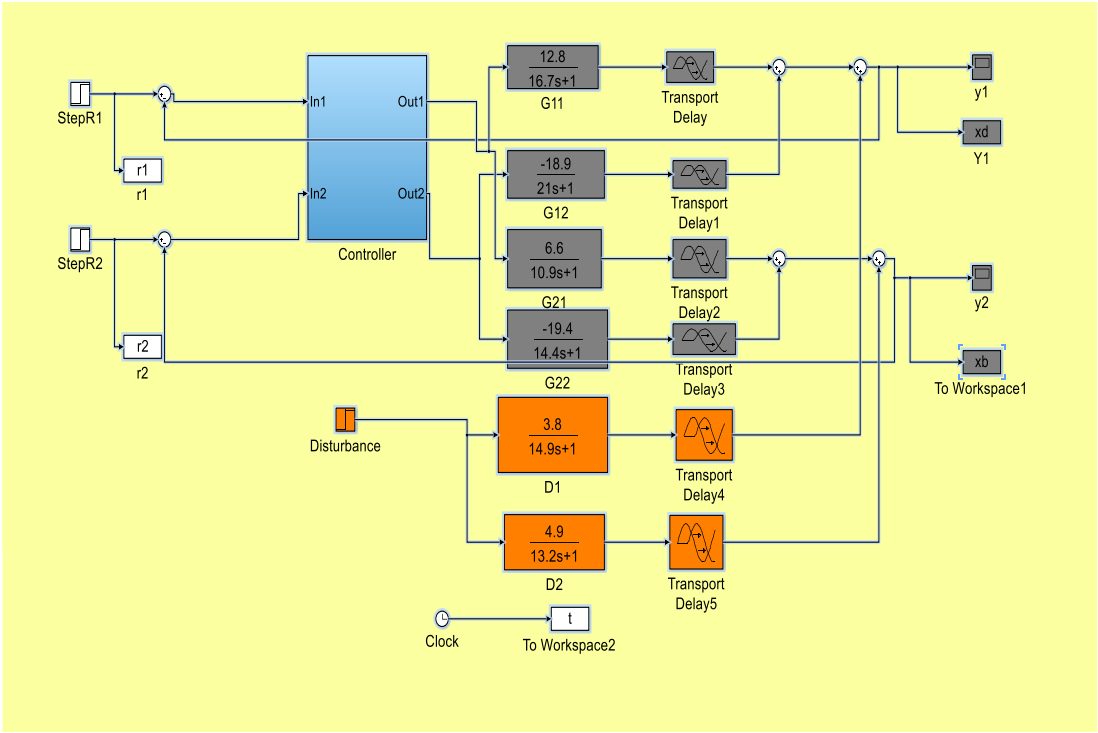
Appendix - A2: *OR* Distillation Column Control Simulink Block Diagram (With Disturbances)



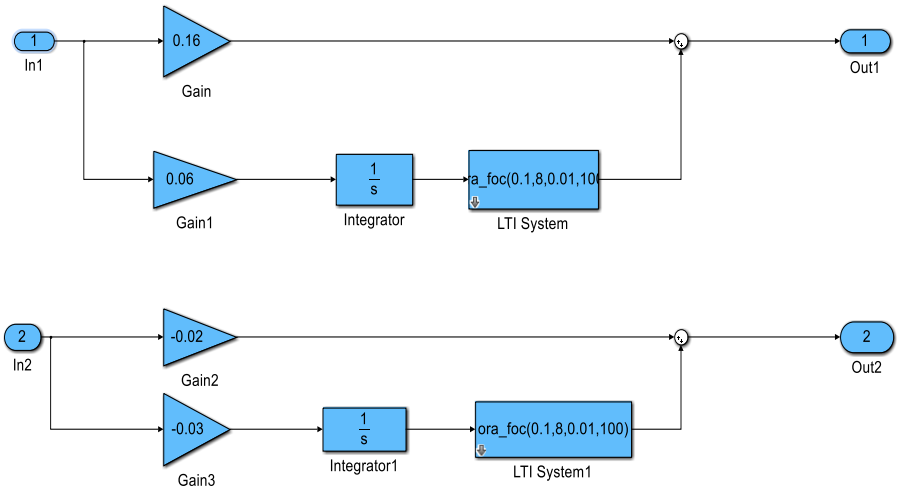
Detailed Diagram of Disturbance Sub-system is given



Appendix - A3: Simulink Block Diagram of Wood-Berry (WB) Distillation Model



Details of Controller Sub-system is given:



Appendix - B: Matlab Programs

Appendix - B1: Matlab Program for Skogestad's Column A model

```
function xprime=colamod(t,X,U)
%
% colamod - This is a nonlinear model of a distillation column with
%           NT-1 theoretical stages including a reboiler (stage 1)
plus a
%           total condenser ("stage" NT). The liquid flow dynamics
are
%           modelled by a simple linear relationship.
%           Model assumptions: Two components (binary separation);
constant
%           relative volatility; no vapor holdup; one feed and two
products;
%           constant molar flows (same vapor flow on all stages);
%           total condenser
%
%           The model is based on column A in Skogestad and
Postlethwaite
%           (1996). The model has 82 states.
%
% Inputs:    t    - time in [min].
%            X    - State, the first 41 states are compositions of
light
%                   component A with reboiler/bottom stage as X(1)
and
%                   condenser as X(41). State X(42)is holdup in
reboiler/
%                   bottom stage and X(82) is hold-up in condenser.
%            U(1) - reflux L,
%            U(2) - boilup V,
%            U(3) - top or distillate product flow D,
%            U(4) - bottom product flow B,
%            U(5) - feed rate F,
%            U(6) - feed composition, zF.
%            U(7) - feed liquid fraction, qF.
%
% Outputs:   xprime - vector with time derivative of all the states

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%-------------------------------------------------------------
% The following data need to be changed for a new column.
% These data are for "colmn A".
% Number of stages (including reboiler and total condenser:
    NT=41;
% Location of feed stage (stages are counted from the bottom):
    NF=21;
% Relative volatility
    alpha=1.5;
% Nominal liquid holdups
    M0(1)=0.5;          % Nominal reboiler holdup (kmol)
    i=2:NT-1; M0(i)=0.5*ones(1,NT-2);% Nominal stage (tray) holdups
(kmol)
    M0(NT)=0.5;         % Nominal condenser holdup (kmol)
```

```
% Data for linearized liquid flow dynamics (does not apply to
reboiler and condenser):
    taul=0.063;          % time constant for liquid dynamics (min)
    F0=1;        % Nominal feed rate (kmol/min)
    qF0 = 1;         % Nominal fraction of liquid in feed
    L0=2.70629;          % Nominal reflux flow (from steady-state
data)
    L0b=L0 + qF0*F0;     % Nominal liquid flow below feed (kmol/min)
    lambda=0;        % Effect of vapor flow on liquid flow ("K2-
effect")
    V0=3.20629;V0t=V0+(1-qF0)*F0;% Nominal vapor flows - only needed
if lambda is nonzero
% End data which need to be changed
%-------------------------------------------------------------

% Splitting the states
x=X(1:NT)';                              % Liquid composition from btm
to top
M=X(NT+1:2*NT)';                         % Liquid hold up from btm to
top

% Inputs and disturbances
LT = U(1);                                % Reflux
VB = U(2);                                % Boilup
D  = U(3);                                % Distillate
B  = U(4);                                % Bottoms

F  = U(5);                                % Feedrate
zF = U(6);                                % Feed composition
qF = U(7);                                % Feed liquid fraction

% THE MODEL

% Vapor-liquid equilibria
i=1:NT-1;    y(i)=alpha*x(i)./(1+(alpha-1)*x(i));

% Vapor Flows assuming constant molar flows
i=1:NT-1;    V(i)=VB*ones(1,NT-1);
i=NF:NT-1;   V(i)=V(i) + (1-qF)*F;

% Liquid flows assuming linearized tray hydraulics with time
constant taul
% Also includes coefficient lambda for effect of vapor flow ("K2-
effect").
i=2:NF;      L(i) = L0b + (M(i)-M0(i))./taul + lambda.*(V(i-1)-V0);
i=NF+1:NT-1; L(i) = L0  + (M(i)-M0(i))./taul + lambda.*(V(i-1)-V0t);
L(NT)=LT;

% Time derivatives from  material balances for
% 1) total holdup and 2) component holdup

% Column
i=2:NT-1;
dMdt(i) = L(i+1)          - L(i)       + V(i-1)          - V(i);
dMxdt(i)= L(i+1).*x(i+1) - L(i).*x(i) + V(i-1).*y(i-1) - V(i).*y(i);

% Correction for feed at the feed stage
```

```
% The feed is assumed to be mixed into the feed stage
dMdt(NF) = dMdt(NF)  + F;
dMxdt(NF)= dMxdt(NF) + F*zF;

% Reboiler (assumed to be an equilibrium stage)
dMdt(1) = L(2)      - V(1)      - B;
dMxdt(1)= L(2)*x(2) - V(1)*y(1) - B*x(1);

% Total condenser (no equilibrium stage)
dMdt(NT) = V(NT-1)          - LT       - D;
dMxdt(NT)= V(NT-1)*y(NT-1) - LT*x(NT) - D*x(NT);

% Compute the derivative for the mole fractions from d(Mx) = x dM +
M dx
i=1:NT;
dxdt(i) = (dMxdt(i) - x(i).*dMdt(i) )./M(i);

% Output
xprime=[dxdt';dMdt'];
```

Appendix - B2: Matlab Program to compute Robustness and Compare OPI and FOPI Controllers in Examples 4 and 5

```
% ********* Obtain Transfer function matrix of CONTROL system

% clear all
% clc
% s=tf('s');

Kcd11 = tf([1.2 0.24],[1 0]);
Kcd12 = 0;
Kcd13=0;
Kcd21 = 0;
Kcd22 = tf([-0.15 -0.015],[1 0]);
Kcd23=0;
Kcd31=0;
Kcd32=0;
Kcd33 = tf([0.6 0.15],[1 0]);
r = 1.0;
% ********* Transfer function matrix of MV system ***************
g11 = tf([0.66],[6.7 1],'iodelay',2.6);
g12 = tf([-0.61],[8.64 1],'iodelay',3.5);
g13 = tf([-0.0049],[9.06 1],'iodelay',1);
g21 = tf([1.11],[3.25 1],'iodelay',6.5);
g22 = tf([-2.36],[5 1],'iodelay',3);
g23 = tf([-0.012],[7.09 1],'iodelay',1.2);
g31 = tf([-34.68],[8.15 1],'iodelay',9.2);
g32 = tf([46.2],[10.9 1],'iodelay',9.4);
g33 = tf([10.1007 0.87],[73.132 22.69 1],'iodelay',1);
% numg11=0.66; deng11=[6.7 1];
% numg12=-0.61; deng12=[8.64 1];
% numg13=-0.0049; deng13=[9.06 1];
% numg21=1.11; deng21=[3.25 1];
% numg22=-2.36; deng22=[5 1];
```

```
% numg23=-0.012; deng23=[7.09 1];
% numg31=-34.68; deng31=[8.15 1];
% numg32=46.2; deng32=[10.9 1];
% numg33=[10.1007 0.87]; deng33=[73.132 22.69 1];
% d=[2.6 3.5 1;6.5 3 1.2;9.2 9.4 1];

G = [g11 g12 g13; g21 g22 g23; g31 g32 g33];
Gdc = [Kcd11 Kcd12 Kcd13; Kcd21 Kcd22 Kcd23; Kcd31 Kcd32 Kcd33];
% OUTPUT Uncertainty
w = logspace(-3,3,10000)';
G1 = pade(G);
h1 = ss(Gdc);
h2 = ss(G);
h5 = h2*h1;
[A2,B2,C2,D2] = ssdata(h5);
[b,a] = ss2tf(A2,B2,C2,D2,1);
[b1,a1] = ss2tf(A2,B2,C2,D2,2);
[b2,a2] = ss2tf(A2,B2,C2,D2,3);
gh11 = tf(b(1,:),a);
gh21 = tf(b(2,:),a);
gh31 = tf(b(3,:),a);
gh12 = tf(b1(1,:),a1);
gh22 = tf(b1(2,:),a1);
gh32 = tf(b1(3,:),a1);
gh13 = tf(b2(1,:),a2);
gh23 = tf(b2(2,:),a2);
gh33 = tf(b2(3,:),a2);
gh = [gh11+1 gh12 gh13;gh21 gh22+1 gh23;gh31 gh32 gh33+1];
h7 = ss(gh);
h9 = inv(h7);
h10 = h9*h5;
for i = 1:1:10000
w1(i) = w(i,1);
h12 = freqresp(h10,w1(i));
sing1(i) = max(svd(h12));
sing(i) = 1/sing1(i);
end
subplot(1,2,1),loglog(w1,sing,'--')
hold on;
% INPUT Uncertainty
w = logspace(-3,3,10000)';
G1 = pade(G);
h1 = ss(Gdc);
h2 = ss(G);
h5 = h1*h2;
[A2,B2,C2,D2] = ssdata(h5);
[b,a] = ss2tf(A2,B2,C2,D2,1);
[b1,a1] = ss2tf(A2,B2,C2,D2,2);
[b2,a2] = ss2tf(A2,B2,C2,D2,3);
gh11 = tf(b(1,:),a);
gh21 = tf(b(2,:),a);
gh31 = tf(b(3,:),a);
gh12 = tf(b1(1,:),a1);
gh22 = tf(b1(2,:),a1);
gh32 = tf(b1(3,:),a1);
gh13 = tf(b2(1,:),a2);
gh23 = tf(b2(2,:),a2);
gh33 = tf(b2(3,:),a2);
```

```
gh = [gh11+1 gh12 gh13;gh21 gh22+1 gh23;gh31 gh32 gh33+1];
h7 = ss(gh);
h9 = inv(h7);
h10 = h9*h5;
for i = 1:1:10000
w1(i) = w(i,1);
h12 = freqresp(h10,w1(i));
sing1(i) = max(svd(h12));
sing(i) = 1/sing1(i);
end
subplot(1,2,2),loglog(w1,sing,'--')
% hold on


% ********* Repeat the same program for FOPI controller

% ********* Obtain Transfer function matrix of CONTROL system

%clear all
% Define the MIMO FOPI control system
s=tf('s');

Kcd11 = 0.65+(0.2*(1/s)*ora_foc(0.1,8,0.00001,100));
Kcd12 = 0;
Kcd13=0;
Kcd21 = 0;
Kcd22 = -0.13-(0.0582*(1/s)*ora_foc(0.1,8,0.00001,100));
Kcd23=0;
Kcd31=0;
Kcd32=0;
Kcd33=2.2+(2.4*(1/s)*ora_foc(0.1,8,0.00001,100));
r = 1.0;
% ********* Transfer function matrix of MV system ****************
g11 = tf([0.66],[6.7 1],'iodelay',2.6);
g12 = tf([-0.61],[8.64 1],'iodelay',3.5);
g13 = tf([-0.0049],[9.06 1],'iodelay',1);
g21 = tf([1.11],[3.25 1],'iodelay',6.5);
g22 = tf([-2.36],[5 1],'iodelay',3);
g23 = tf([-0.012],[7.09 1],'iodelay',1.2);
g31 = tf([-34.68],[8.15 1],'iodelay',9.2);
g32 = tf([46.2],[10.9 1],'iodelay',9.4);
g33 = tf([10.1007 0.87],[73.132 22.69 1],'iodelay',1);
% numg11=0.66; deng11=[6.7 1];
% numg12=-0.61; deng12=[8.64 1];
% numg13=-0.0049; deng13=[9.06 1];
% numg21=1.11; deng21=[3.25 1];
% numg22=-2.36; deng22=[5 1];
% numg23=-0.012; deng23=[7.09 1];
% numg31=-34.68; deng31=[8.15 1];
% numg32=46.2; deng32=[10.9 1];
% numg33=[10.1007 0.87]; deng33=[73.132 22.69 1];
% d=[2.6 3.5 1;6.5 3 1.2;9.2 9.4 1];

G = [g11 g12 g13; g21 g22 g23; g31 g32 g33];
Gdc = [Kcd11 Kcd12 Kcd13; Kcd21 Kcd22 Kcd23; Kcd31 Kcd32 Kcd33];
% OUTPUT Uncertainty
```

```
w = logspace(-3,3,10000)';
G1 = pade(G);
h1 = ss(Gdc);
h2 = ss(G);
h5 = h2*h1;
[A2,B2,C2,D2] = ssdata(h5);
[b,a] = ss2tf(A2,B2,C2,D2,1);
[b1,a1] = ss2tf(A2,B2,C2,D2,2);
[b2,a2] = ss2tf(A2,B2,C2,D2,3);
gh11 = tf(b(1,:),a);
gh21 = tf(b(2,:),a);
gh31 = tf(b(3,:),a);
gh12 = tf(b1(1,:),a1);
gh22 = tf(b1(2,:),a1);
gh32 = tf(b1(3,:),a1);
gh13 = tf(b2(1,:),a2);
gh23 = tf(b2(2,:),a2);
gh33 = tf(b2(3,:),a2);
gh = [gh11+1 gh12 gh13;gh21 gh22+1 gh23;gh31 gh32 gh33+1];
h7 = ss(gh);
h9 = inv(h7);
h10 = h9*h5;
for i = 1:1:10000
w1(i) = w(i,1);
h12 = freqresp(h10,w1(i));
sing1(i) = max(svd(h12));
sing(i) = 1/sing1(i);
end
subplot(1,2,1),loglog(w1,sing)
hold on;
% INPUT Uncertainty
w = logspace(-3,3,10000)';
G1 = pade(G);
h1 = ss(Gdc);
h2 = ss(G);
h5 = h1*h2;
[A2,B2,C2,D2] = ssdata(h5);
[b,a] = ss2tf(A2,B2,C2,D2,1);
[b1,a1] = ss2tf(A2,B2,C2,D2,2);
[b2,a2] = ss2tf(A2,B2,C2,D2,3);
gh11 = tf(b(1,:),a);
gh21 = tf(b(2,:),a);
gh31 = tf(b(3,:),a);
gh12 = tf(b1(1,:),a1);
gh22 = tf(b1(2,:),a1);
gh32 = tf(b1(3,:),a1);
gh13 = tf(b2(1,:),a2);
gh23 = tf(b2(2,:),a2);
gh33 = tf(b2(3,:),a2);

gh = [gh11+1 gh12 gh13;gh21 gh22+1 gh23;gh31 gh32 gh33+1];
h7 = ss(gh);
h9 = inv(h7);
h10 = h9*h5;
for i = 1:1:10000
w1(i) = w(i,1);
h12 = freqresp(h10,w1(i));
sing1(i) = max(svd(h12));
sing(i) = 1/sing1(i);
```

```
end
subplot(1,2,2),loglog(w1,sing)
hold on
```

Appendix - B3: Matlab Program written for BLT tuning of FOPI Controllers - OR Distillation Column Control Example in Chapter 5

```
% ********* Matlab program written to tune FOPI controllers for the
%Three input three output OR distillation system. Controllers are
%tuned such that biggest log-magnitude of 6 dB is obtained.

% ********* Firstly, Obtain Transfer function elements of the system

numg11=0.66; deng11=[6.7 1];
numg12=-0.61; deng12=[8.64 1];
numg13=-0.0049; deng13=[9.06 1];
numg21=1.11; deng21=[3.25 1];
numg22=-2.36; deng22=[5 1];
numg23=-0.012; deng23=[7.09 1];
numg31=-34.68; deng31=[8.15 1];
numg32=46.2; deng32=[10.9 1];
numg33=[10.1007 0.87]; deng33=[73.132 22.69 1];
d=[2.6 3.5 1;6.5 3 1.2;9.2 9.4 1];

% kczn=[33.6 -7.5 0.1;22.4 -5.3 0.001;1945 -430.7 4.7];
% resetzn=[383.2 218.7 1592;576 75.8 182;901.8 704.9 99.7];
kczn=[0.7277 0 0;0 -0.17 0;0 0 0.5645];
resetzn=[10 0 0;0 7.5 0;0 0 28.2];


i=sqrt(-1);
w=logspace(-1,0.8,200);
s=i*w;
f=1.5;
df=0.01;
loop=0;
flagm=-1;
flagp=-1;
dbmax=-100;
 while abs(dbmax)>0.05
kc=kczn/f;
reset=resetzn*f;
% Form controller transfer function
% numgc11=kc(1,1)*[reset(1,1) 1];
% dengc11=[reset(1,1) 0];
nc11=kc(1,1)*reset(1,1)*(ora_foc(0.9,5,0.01,100)+1);
dc11=reset(1,1)*ora_foc(0.9,5,0.01,100);
c11=nc11/dc11;
[numgc11,dengc11]=tfdata(c11);
% numgc12=kc(1,2)*[reset(1,2) 1];
% dengc12=[reset(1,2) 0];
% numgc13=kc(1,3)*[reset(1,3) 1];
% dengc13=[reset(1,3) 0];
% numgc21=kc(2,1)*[reset(2,1) 1];
```

```
% dengc21=[reset(2,1) 0];
nc22=kc(2,2)*reset(2,2)*(ora_foc(0.9,5,0.01,100)+1);
dc22=reset(2,2)*ora_foc(0.9,5,0.01,100);
c22=nc22/dc22;
[numgc22,dengc22]=tfdata(c22);
% numgc23=kc(2,3)*[reset(2,3) 1];
% dengc23=[reset(2,3) 0];
% numgc31=kc(3,1)*[reset(3,1) 1];
% dengc31=[reset(3,1) 0];
% numgc32=kc(3,2)*[reset(3,2) 1];
% dengc32=[reset(3,2) 0];
nc33=kc(3,3)*reset(3,3)*(ora_foc(0.9,5,0.01,100)+1);
dc33=reset(3,3)*ora_foc(0.9,5,0.01,100);
c33=nc33/dc33;
[numgc33,dengc33]=tfdata(c33);
% Inside loop to vary frequency
nwtot=length(w);
for nw=1:nwtot
wn=w(nw);
% Process transfer function matrix (g's)
g(1,1)=polyval(numg11,s(nw)) / polyval(deng11,s(nw));
g(1,1)=g(1,1)*exp(-d(1,1)*s(nw));
g(1,2)=polyval(numg12,s(nw)) / polyval(deng12,s(nw));
g(1,2)=g(1,2)*exp(-d(1,2)*s(nw));
g(1,3)=polyval(numg13,s(nw)) / polyval(deng13,s(nw));
g(1,3)=g(1,3)*exp(-d(1,3)*s(nw));
g(2,1)=polyval(numg21,s(nw)) / polyval(deng21,s(nw));
g(2,1)=g(2,1)*exp(-d(2,1)*s(nw));
g(2,2)=polyval(numg22,s(nw)) / polyval(deng22,s(nw));
g(2,2)=g(2,2)*exp(-d(2,2)*s(nw));
g(2,3)=polyval(numg23,s(nw)) / polyval(deng23,s(nw));
g(2,3)=g(2,3)*exp(-d(2,3)*s(nw));
g(3,1)=polyval(numg31,s(nw)) / polyval(deng31,s(nw));
g(3,1)=g(3,1)*exp(-d(3,1)*s(nw));
g(3,2)=polyval(numg32,s(nw)) / polyval(deng32,s(nw));
g(3,2)=g(3,2)*exp(-d(3,2)*s(nw));
g(3,3)=polyval(numg33,s(nw)) / polyval(deng33,s(nw));
g(3,3)=g(3,3)*exp(-d(3,3)*s(nw));

% Controller transfer function matrix (gc's)
gc(1,1)=polyval(numgc11,s(nw)) / polyval(dengc11,s(nw));
% gc(1,2)=(polyval(numgc12,s(nw)) / polyval(dengc12,s(nw)));
% gc(1,3)=polyval(numgc13,s(nw)) / polyval(dengc13,s(nw));
% gc(2,1)=(polyval(numgc21,s(nw)) / polyval(dengc21,s(nw)));
gc(2,2)=polyval(numgc22,s(nw)) / polyval(dengc22,s(nw));
% gc(2,3)=(polyval(numgc23,s(nw)) / polyval(dengc23,s(nw)));
% gc(3,1)=(polyval(numgc31,s(nw)) / polyval(dengc31,s(nw)));
% gc(3,2)=(polyval(numgc32,s(nw)) / polyval(dengc32,s(nw)));
gc(3,3)=(polyval(numgc33,s(nw)) / polyval(dengc33,s(nw)));
% "eye" operation forms an identity matrix
wnyquist(nw)=-1+det(eye(size(g))+g*gc);
% Calculate lc function
lc(nw)=wnyquist(nw)/(1+wnyquist(nw));
dbcl(nw)=20*log10(abs(lc(nw)));
% End of inside loop sweeping through frequencies
end
% Pick off peak in closedloop log modulus
[dbmax,nmax]=max(dbcl);
wmax=w(nmax);
```

```
loop=loop+1;
if loop>50,break,end
% Test if +4 dB and reguess f factor
if dbmax>2
if flagp>0,df=df/2;end
flagm=1;
f=f+df;
else
if flagm>0,df=df/2;end
flagp=1;
f=f-df;
if f<10,f=10;end
end
end
```

Appendix - B4: Matlab Program written for BLT tuning of FOPI Controllers in Chapter 5 – TITO WB Distillation Model

```
% ********* Matlab program written to tune FOPI controllers for the
%TITO WB system. Controllers are tuned such that biggest log-magnitude
%of 4 dB is obtained.

% ********* Firstly, Obtain Transfer function matrix of the system

numg11=12.8; deng11=[16.7 1];
numg12=-18.9; deng12=[21 1];
% numg13=-0.0049; deng13=[9.06 1];
numg21=6.6; deng21=[16.7 1];
numg22=-19.4; deng22=[14.4 1];
% numg23=-0.012; deng23=[7.09 1];
% numg31=-34.68; deng31=[8.15 1];
% numg32=46.2; deng32=[10.9 1];
% numg33=[10.1007 0.87]; deng33=[73.132 22.69 1];
d=[1 3;1 3];

% kczn=[33.6 -7.5 0.1;22.4 -5.3 0.001;1945 -430.7 4.7];
% resetzn=[383.2 218.7 1592;576 75.8 182;901.8 704.9 99.7];
kczn=[0.0948 -0.0924;0.0157 -0.0305];
resetzn=[3.1325 2.4176;1.0448 1.0487];

i=sqrt(-1);
w=logspace(-1,0.8,200);%logspace(a,b,n) generates n points between
decades 10^a and 10^b.
%y = logspace(a,pi) generates the points between 10^a and pi.
%which is useful for DSP where frequencies over this interval go
around the unit circle.
s=i*w;
%% Detuning Factors for Centralized Controller (4.0 dB)
f=0.47331;
df=0.01;
loop=0;
flagm=-1;
flagp=-1;
```

```
dbmax=-100;
%% Main loop to vary f in BLT tuning
while abs(dbmax-2)>0.05
kc=kczn/f;
reset=resetzn*f;
% Form controller transfer function
% numgc11=kc(1,1)*[reset(1,1) 1];
% dengc11=[reset(1,1) 0];
% numgc12=kc(1,2)*[reset(1,2) 1];
% dengc12=[reset(1,2) 0];
% numgc21=kc(2,1)*[reset(2,1) 1];
% dengc21=[reset(2,1) 0];
% numgc22=kc(2,2)*[reset(2,2) 1];
% dengc22=[reset(2,2) 0];
[numgc11,dengc11,numgc12,dengc12,numgc21,dengc21,numgc22,dengc22]=
TwoPolesCentral(reset,kc);
% Inside loop to vary frequency
nwtot=length(w);
for nw=1:nwtot
wn=w(nw);
% Process transfer function matrix (g's)
g(1,1)=polyval(numg11,s(nw)) / polyval(deng11,s(nw));
g(1,1)=g(1,1)*exp(-d(1,1)*s(nw));
g(1,2)=polyval(numg12,s(nw)) / polyval(deng12,s(nw));
g(1,2)=g(1,2)*exp(-d(1,2)*s(nw));
g(2,1)=polyval(numg21,s(nw)) / polyval(deng21,s(nw));
g(2,1)=g(2,1)*exp(-d(2,1)*s(nw));
g(2,2)=polyval(numg22,s(nw)) / polyval(deng22,s(nw));
g(2,2)=g(2,2)*exp(-d(2,2)*s(nw));
% Controller transfer function matrix (gc's)
gc(1,1)=polyval(numgc11,s(nw)) / polyval(dengc11,s(nw));
gc(1,2)=(polyval(numgc12,s(nw)) / polyval(dengc12,s(nw)));
gc(2,1)=(polyval(numgc21,s(nw)) / polyval(dengc21,s(nw)));
gc(2,2)=polyval(numgc22,s(nw)) / polyval(dengc22,s(nw));
% "eye" operation forms an identity matrix
wnyquist(nw)=-1+det(eye(size(g))+g*gc);
% Calculate lc function
lc(nw)=wnyquist(nw)/(1+wnyquist(nw));
dbcl(nw)=20*log10(abs(lc(nw)));
% End of inside loop sweeping through frequencies
end
% Pick off peak in closedloop log modulus
[dbmax,nmax]=max(dbcl);
wmax=w(nmax);
loop=loop+1;
if loop>32,break,end
% Test if +4 dB and reguess f factor
if dbmax>2
if flagp>0,df=df/2;end
flagm=1;
f=f+df;
else
if flagm>0,df=df/2;end
flagp=1;
f=f-df;
if f<1,f=1;end
end
end
```

Appendix - C: Matlab Programs used in Chapter Six for Predictive Control System Design

Appendix - C1: Matlab Program written for Converting Transfer function model (OR Distillation Column) to State Space Model

```
% Enter transfer function matrix as g

s=tf('s');
g11=0.66*(exp(-2.6*s)/((6.7*s)+1));
g12=-0.61*(exp(-3.5*s)/((8.64*s)+1));
g13=-0.0049*(exp(-1*s)/((9.06*s)+1));
g21=1.11*(exp(-6.5*s)/((3.25*s)+1));
g22=-2.36*(exp(-3*s)/((5*s)+1));
g23=-0.012*(exp(-1.2*s)/((7.09*s)+1));
g31=-34.68*(exp(-9.2*s)/((8.15*s)+1));
g32=46.2*(exp(-9.4*s)/((10.9*s)+1));
g33=(exp(-s)*((10.1007*s)+0.87)/((73.132*(s^2))+(22.69*s)+1));
g=[g11 g12 g13;g21 g22 g23;g31 g32 g33];
C and minimum realisation
Gmin=ss(g,'min');
T=2.5; % Set sampling time

[Ac,Bc,Cc,Dc]=ssdata(Gmin);
[Ap,Bp,Cp,Dp]=c2dm(Ac,Bc,Cc,Dc,T,'zoh'); % Discretization

[m1,n1]=size(Cp);
[n1,n_in]=size(Bp);
```

Appendix - C2: Matlab Program written for obtaining Augmented State Space System and Predictive matrix G

```
function [G,f,Rs,H_H,H_F,H_R,A_e,B_e,C_e] % Declare function
=predictGainMIMO22(Ap,Bp,Cp,Nc,Np); % Ap Bp Cp state space
% Nc is control horizon and Np is prediction horizon
[m1,n1]=size(Cp);
[n1,n_in]=size(Bp);
% Augmentation of process state space model
A_e=eye(n1+m1,n1+m1);
A_e(1:n1,1:n1)=Ap;
A_e(n1+1:n1+m1,1:n1)=Cp*Ap;
B_e=zeros(n1+m1,n_in);
B_e(1:n1,:)=Bp;
B_e(n1+1:n1+m1,:)=Cp*Bp;
C_e=zeros(m1,n1+m1);
C_e(:,n1+1:n1+m1)=eye(m1,m1);
h(1:m1,:)=C_e;
f(1:m1,:)=C_e*A_e;
for kk=m1+1:Np
h(kk,:)=h(kk-1,:)*A_e;
f(kk,:)= f(kk-1,:)*A_e;
```

```
end

v=h*B_e;
G=zeros(Np,Nc);
G(:,1:m1)=v;
for i=m1:+m1:Nc-m1
G(:,i+1:i+m1)=[zeros(i,m1);v(1:Np-i,1:m1)]; %Toeplitz matrix
end

Rs=ones(Np,1);
H_R=G'*Rs;
H_H=G'*G;
H_F=G'*f;


% Declare predictive gain matrix function for SISO system

function [G,f,BarR,H_H,H_F,H_R,A_e,B_e,C_e]
=predictivegain1(Ap,Bp,Cp,Nc,Np);
[m1,n1]=size(Cp);%Number of outputs
[n1,n_in]=size(Bp);%Number of outputs
A_e = eye(n1+m1,n1+m1);
A_e(1:n1,1:n1) = Ap;
A_e(n1+1:n1+m1,1:n1) = Cp*Ap;
B_e=zeros(n1+m1,n_in);
B_e(1:n1,:)=Bp;
B_e(n1+1:n1+m1,:)=Cp*Bp;
C_e=zeros(m1,n1+m1);
C_e(:,n1+1:n1+m1)=eye(m1,m1);
n=n1+m1;
h(1:m1,:)=C_e;
f(1:m1,:)=C_e*A_e;
for kk = 2:Np
    h(kk,:)=h(kk-1,:)*A_e;
    f(kk,:)=f(kk-1,:)*A_e;
end
v=h*B_e;
G=zeros(Np,Nc);
G(:,1)=v;
for i=2:Nc
    G(:,i)=[zeros(i-1,1);v(1:Np-i+1,1)];
end
BarR=ones(Np,1);
H_H=G'*G;
H_F=G'*f;
H_R=G'*BarR;
```

Appendix - C3: DMPC Matlab Program

```
clc
clear all

%Get state space system
Ap=[-0.0217,-0.3141;0.3141,0.7636];%Get KENNY state space system
 Bp=[0.3141;0.2364];%proportional gain initialisation
% ki=0.009;%integral gain initialisation
```

```
% landa=0.9;%Fractional order initialisation for kenny
%Get Kj.
 Cp=[-1,2];
% Ap=[1 1;0 1];%settles at Np=7 and Ki=0.00007
% Bp=[0.5;1];Cp=[1 0];
Dp=0;Np=20;Nc=3;
%Get prediction matrices and hessian matrix noting that matrix H is
our
%matrix Gp.
%  [H,H_H,H_F,H_R,A_e,B_e,C_e] =predictivegain1(Ap,Bp,Cp,Nc,Np);
[H,f,Rs,H_H,H_F,H_R,A_e,B_e,C_e] =predictGainMIMO22(Ap,Bp,Cp,Nc,Np);
% [Np,Nc]=size(H);
[n,n_in]=size(B_e);
xm=[0;0];
Xf=zeros(n,1);
N_sim=40;

r=ones(N_sim,1);
u=0; % u(k-1) =0
y=0;
for kk=1:N_sim;
DeltaU=inv(H_H+0.1*eye(Nc,Nc))*(H_R*r(kk)-H_F*Xf);
deltau=DeltaU(1,1);
u=u+deltau;
u1(kk)=u;
y1(kk)=y;
xm_old=xm;
xm=Ap*xm+Bp*u;
y=Cp*xm;
Xf=[xm-xm_old;y];
end
k=0:(N_sim-1);
figure
subplot(211)
plot(k,y1)
xlabel('Sampling Instant')
legend('Output')
subplot(212)
plot(k,u1)
xlabel('Sampling Instant')
legend('Control')
```

Appendix - C4: DMPC Matlab Program for Reactive Column Control Example

```
clear all
Ap=[0.9972 0.0029 -0.0003;-0.0059 0.9930 -0.0003;-0.0012 -0.0062
0.9946];
Bp=[0.0002 0.0002 -0.0125;-0.0007 0.0016 -0.0697;0.0016 0.0066 -
0.2779];
Cp=[-12.8264 4.0787 -0.16808;-3.7502 -0.8260 -1.996;-6.2676 -17.6851
-0.7126];
Dp=0;
% Ap=[-0.0019 0.0024 -0.0009;-0.0034 -0.0043 -0.0008;-0.0009 -0.0052
-0.0055];Np=200;Nc=50;
```

```
% Bp=[0.0003 0.0013 -0.0566;-0.0005 0.0019 -0.0822;0.0025 0.0067 -
0.2818];
% Cp=[-12.8258 4.0787 -0.1685;-3.752 -0.8266 -1.9955;-6.2674 -
17.6857 -0.7129];landa0.0000009
% Ap=[0.9981 0.0024 -0.0009;-0.0034 0.9957 -0.0008;-0.0009 -0.0052
0.9945];
% Bp=[0.0003 0.0013 -0.0565;-0.0005 0.0019 -0.0818;0.0025 0.0067 -
0.2808];
% Cp=[-12.8258 4.0787 -0.1685;-3.752 -0.8266 -1.9955;-6.2674 -
17.6857 -0.7129];
% Dp=0;
RC=ss(Ap,Bp,Cp,Dp);
Np=200;
Nc=50;
[H,f,Rs,H_H,H_F,A_e,B_e,C_e] =predictGainMIM22(Ap,Bp,Cp,Nc,Np);
% [H,H_H,H_F,A_e,B_e,C_e] =predictGainMIMO(Ap,Bp,Cp,Nc,Np);
% [H,f,Rs,H_H,H_F,H_R,A_e,B_e,C_e]
=predictGainMIMO22(Ap,Bp,Cp,Nc,Np);
[n,n_in]=size(B_e);
[m1,n1]=size(Cp);
[n1,n_in]=size(Bp);
 xm=zeros(n1,1);
 Xf=zeros(n,1);

 N_sim=200;
r1=70.75*ones(1,N_sim+10);
r2=71.31*ones(1,N_sim+10);
r3=89.45*ones(1,N_sim+10);
% r2=-0.0275*ones(1,N_sim+10);
r=[r1;r2;r3];
n=n1+m1;
u=zeros(n_in,1);y=zeros(m1,1);
h_r=(H_F(:,4:6));
for kk=1:N_sim;
DeltaU1=inv((H_H+(0.0000009*eye(Nc,Nc))))*(((h_r*r(:,kk))-
(H_F*Xf)));
deltau=DeltaU1(1:n_in,1);

%    DeltaU2=inv(H_H+0.1*eye(Nc,Nc))*(H_R*sp(kk)-H_F*Xf);
%    deltau2=DeltaU2(1,1);
%    deltau=[deltau1;deltau2];
u=u+deltau;
u1(:,kk)=u;
y1(:,kk)=y;
xm_old=xm;
xm=Ap*xm+Bp*u;
y=Cp*xm;
Xf=[xm-xm_old;y];
end
k=0:(N_sim-1);
plot(k,y1(1,:),'r',k,y1(2,:),'g',k,y1(3,:))
```

Appendix - C5: FOPPI Matlab Program

```
clc
clear all
% Ap=[1 1;0 1];%settles at Np=7 and Ki=0.00007
```

```
% Bp=[0.5;1];Cp=[1 0];
% D=0;

%   Dp=0;

%Get state space system
Ap=[1 1 0;0 1 0;1 1 1];%settles at Np=7 and Ki=0.00007
Bp=[0.5;1;0.5];Cp=[0 0 1];%Kp=0.000099
% Dp=0;
Np=7;Nc=7;
%Get prediction matrices and hessian matrix noting that matrix H is
our
%matrix Gp.
[H,f,BarR,H_H,H_F,H_R,A_e,B_e,C_e] =predictivegain1(Ap,Bp,Cp,Nc,Np);
%  [H,F,Rs,H_H,H_F,H_R,A_e,B_e,C_e]
=predictivegain1(Ap,Bp,Cp,Nc,Np);
% [Np,Nc]=size(H);



G(1,:)=H(1,:);
sigmal=zeros(Nc,Nc);
sigmalnew=zeros(Nc,Nc);
sigmalopre=zeros(Nc,Nc);

% G1(1,:)=zeros(1,Nc);
% G2(1,:)=zeros(1,Nc);
% G3(1,:)=zeros(1,Nc);
% G4(1,:)=zeros(1,Nc);
% %Get Gp&Gj from second row
for i= 2:Np
    G(i,1)=H(i,1)-H(i-1,1);%Get Gp first column
    G(i,2:Nc)=G(i-1,1:Nc-1);%Get Gp from second row
%     G1(i,:)=H(i-1,:);%Get Gj from second row
%     G2(i,:)=G1(i-1,:);
%     G3(i,:)=G2(i-1,:);
%     G4(i,:)=G3(i-1,:);
end
ki=0.00351;
kp=0.00007;
%  ki=0.15;
% kp=0.15;%proportional gain initialisation with KENNY
% ki=0.3;%integral gain initialisation
landa=0.9;%Fractional order initialisation
%Get Kj.
 b=1;k=ki;
for i=1:Nc
    if i<2
        b(1)=-landa;
        k(1)=-landa*ki;
    else
    b(i)=(1-(1+landa)/i)*b(i-1)*ki;
    k(i)=(-(1+landa)/i)*b(i-1);
    end
end
% kj2=(1-((1+landa)/2))*ki;kj3=(1-((1+landa)/3))*kj2;kj4=(1-
((1+landa)/4))*kj3;
% k1=-(1+landa)*ki;k2=-((1+landa)/2)*kj2;k3=-((1+landa)/3)*kj3;k4=-
((1+landa)/4)*kj4;
```

```
for i= 1:Nc
%     i
%%calling Gj generator
Gj =matrixG1(H,Nc);
sigmalnew=(k(i)*(Gj'*Gj));

H=Gj;

sigmal=sigmalnew+sigmalopre;
sigmalopre=sigmalnew;
% sigmal=k(j)*Gj;
end
%%
%Get the summation term - last term of the control law -Product of
KjGj...


%sigmal=(k1*(G1'*G1))+(k2*(G2'*G2))+(k3*(G3'*G3))+(k4*(G4'*G4));
Xxx=(kp*(G'*G))+(ki*(H'*H))+sigmal+(0.18*eye(Nc,Nc));
%Get the teta function which is the only inverse operation
[n,n_in]=size(B_e);
xm=zeros(size(Bp));
Xf=zeros(n,1);
N_sim=80;
r=ones(1,N_sim);%desired reference
l11=eye(Nc);
l=l11(1,:);
l2=l';
u=0;
y=0;D=0;v=0;
 m=eye(2);
for kk=1:N_sim
%  v=[zeros(N_sim/2,1);0.1*ones(N_sim/2,1)];
%    v=[zeros(N_sim/80,1);ones((79*N_sim/80),1)];
   deltaU=(ki*(Xxx\H')*l2)*((r(kk)*H_R)-
(H_F*Xf))'+(kp*(Xxx\G')*l2)*((r(kk)*H_R)-
(H_F*Xf))'+((Xxx\sigmal)*l2)*((r(kk)*H_R)-(H_F*Xf))';
    dU=deltaU(1,1);
u=u+dU;
 xm_old=xm;
xm=Ap*xm+Bp*u;
    y=Cp*xm;
    Xf=[xm-xm_old;y];
    u1(kk)=u;
    y1(kk)=y;
end
k=0:(N_sim-1);
figure
subplot(211)
plot(k,y1,k,v)
xlabel('Sampling Instant')
legend('Output Signal')
subplot(212)
plot(k,u1)
xlabel('Sampling Instant')
legend('Control Signal')
```

Appendix - C6: FOPPI Matlab Program showing Disturbance Rejection Property of the Controller

```
clc
clear all
% [Ap,Bp,Cp,Dp]=state space system model
Ap=[-0.0217,-0.3141;0.3141,0.7636];%Get KENNY state space system
Bp=[0.3141;0.2364];%proportional gain initialisation
% ki=0.009;%integral gain initialisation
% landa=0.9;%Fractional order initialisation for kenny
%Get Kj.
Cp=[-1,2];
  Dp=0;

%Get state space system
% Ap=[1 1 0;0 1 0;1 1 1];%settles at Np=7 and Ki=0.00007
% Bp=[0.5;1;0.5];Cp=[0 0 1];%Kp=0.000099
% Dp=0;
Np=7;Nc=7;
%Get prediction matrices and hessian matrix noting that matrix H is
our
%matrix Gp.
 [H,H_H,H_F,H_R,A_e,B_e,C_e] =predictivegain11(Ap,Bp,Cp,Nc,Np);
% [Np,Nc]=size(H);


G(1,:)=H(1,:);
sigmal=zeros(Nc,Nc);
sigmalnew=zeros(Nc,Nc);
sigmalopre=zeros(Nc,Nc);

% G1(1,:)=zeros(1,Nc);
% G2(1,:)=zeros(1,Nc);
% G3(1,:)=zeros(1,Nc);
% G4(1,:)=zeros(1,Nc);
% %Get Gp&Gj from second row
for i= 2:Np
    G(i,1)=H(i,1)-H(i-1,1);%Get Gp first row
    G(i,2:Nc)=G(i-1,1:Nc-1);%Get Gp from second row
%     G1(i,:)=H(i-1,:);%Get Gj from second row
%     G2(i,:)=G1(i-1,:);
%     G3(i,:)=G2(i-1,:);
%     G4(i,:)=G3(i-1,:);
end
 ki=0.15;
kp=0.15;%proportional gain initialisation
% ki=0.3;%integral gain initialisation
landa=0.9;%Fractional order initialisation
%Get Kj.
 b=1;k=ki;
for i=1:Nc
    if i<2
        b(1)=-landa;
        k(1)=-landa*ki;
    else
    b(i)=(1-(1+landa)/i)*b(i-1)*ki;
    k(i)=(-(1+landa)/i)*b(i-1);
```

```
    end
end
% kj2=(1-((1+landa)/2))*ki;kj3=(1-((1+landa)/3))*kj2;kj4=(1-
((1+landa)/4))*kj3;
% k1=-(1+landa)*ki;k2=-((1+landa)/2)*kj2;k3=-((1+landa)/3)*kj3;k4=-
((1+landa)/4)*kj4;

for i= 1:Nc
%     i
%%calling Gj generator
[Gj] =matrixG11(H,Nc);
sigmalnew=(k(i)*(Gj'*Gj));

H=Gj;

sigmal=sigmalnew+sigmalopre;
sigmalopre=sigmalnew;
% sigmal=k(j)*Gj;
end
%%
%Get the summation term - last term of the control law -Product of
KjGj...


%sigmal=(k1*(G1'*G1))+(k2*(G2'*G2))+(k3*(G3'*G3))+(k4*(G4'*G4));
Xxx=(kp*(G'*G))+(ki*(H'*H))+sigmal+(0.2*eye(Nc,Nc));
%Get the teta function which is the only inverse operation
[n,n_in]=size(B_e);
xm=zeros(size(Bp));
Xf=zeros(n,1);
N_sim=100;
r=ones(N_sim,1);%desired reference
l11=eye(Nc);
l=l11(1,:);
l2=l';
u=0;
y=0;D=0;v=0;
 m=eye(2); td=0.00005; k11=0.000001:0.000001:0.0001;

%Introduce 25% disturbance signal 'v'
 v=0.25*heaviside(k11-td);
for kk=1:N_sim

   deltaU=(ki*(Xxx\H')*l2)*((r(kk)*H_R)-
(H_F*Xf))'+(kp*(Xxx\G')*l2)*((r(kk)*H_R)-
(H_F*Xf))'+((Xxx\sigmal)*l2)*((r(kk)*H_R)-(H_F*Xf))';
    dU=deltaU(1,1);
u=u+dU;
 xm_old=xm;
xm=Ap*xm+Bp*u+v(kk)*m(:,1);
    y=Cp*xm;
    Xf=[xm-xm_old;y];
    u1(kk)=u;
    y1(kk)=y;
end
k=0:(N_sim-1);
```

Appendix - C7: FOPPI Matlab Program for Reactive Distillation Column Control Example

```
%Get state space system

Ap=[0.9972 0.0029 -0.0003;-0.0059 0.9930 -0.0003;-0.0012 -0.0062
0.9946];
Bp=[0.0002 0.0002 -0.0125;-0.0007 0.0016 -0.0697;0.0016 0.0066 -
0.2779];
Cp=[-12.8264 4.0787 -0.16808;-3.7502 -0.8260 -1.996;-6.2676 -17.6851
-0.7126];
Dp=0;

Np=60;
Nc=15;
%
[H,f,Rs,H_H,H_F,H_R,A_e,B_e,C_e]=predictGainMIMO22(Ap,Bp,Cp,Nc,Np);
[H,f,Rs,H_H,H_F,A_e,B_e,C_e] =predictGainMIMO22(Ap,Bp,Cp,Nc,Np);

G(1,:)=H(1,:);
sigmal=zeros(Nc,Nc);
sigmalnew=zeros(Nc,Nc);
sigmalopre=zeros(Nc,Nc);
% kp=-1.95;%proportional gain initialisation
% ki=-2.9;%integral gain initialisationMalaysia
 kp=0.000000049;%proportional gain initialisation
 ki=0.00000049;%integral gain initialisation

landa=0.7;
for i= 2:Nc
    G(i,1)=H(i,1)-H(i-1,1);%Get Gp first column
    G(i,2:Nc)=G(i-1,1:Nc-1);%Get Gp from second row
%     G1(i,:)=H(i-1,:);%Get Gj from second row
%     G2(i,:)=G1(i-1,:);
%     G3(i,:)=G2(i-1,:);
%     G4(i,:)=G3(i-1,:);
end

b=1;k=ki;
for i=1:Nc
    if i<2
        b(1)=-landa;
        k(1)=-landa*ki;
    else
    b(i)=(1-(1+landa)/i)*b(i-1)*ki;
    k(i)=(-(1+landa)/i)*b(i-1);
end
end
for i= 1:Nc
%     i
%%calling Gj generator
Gj =matrixG1(H,Np,Nc);
sigmalnew=(k(i)*(Gj'*Gj));

H=Gj;

sigmal=sigmalnew+sigmalopre;
```

```
sigmalopre=sigmalnew;
% sigmal=k(j)*Gj;
end

% y=zeros(m1,1);
% u=zeros(n_in,1);
Xxx=(kp*(G'*G))+(ki*(H'*H))+sigmal+(0.00018*eye(Nc,Nc));
[m1,n1]=size(Cp);
[n1,n_in]=size(Bp);
y=zeros(m1,1);
u=zeros(n_in,1);
xm=zeros(n1,1);
N_sim=2000;
r1=zeros(1,N_sim+10);
r2=zeros(1,N_sim+10);
r3=ones(1,N_sim+10);
sp=[r1;r2;r3];
n=n1+m1;
% Xf=[zeros(2,1);zeros(2,1)];
Xf=[xm; y];
h_R=(H_F(:,4:6));
for kk=1:N_sim;
% deltaU=((kp*inv(Xxx)*G'))*((H_R*sp(:,kk))-(H_F*Xf));
 deltaU=((ki*(Xxx\H'))*((h_R*sp(:,kk))-
(H_F*Xf)))+((kp*(Xxx\G'))*((h_R*sp(:,kk))-
(H_F*Xf)))+(((Xxx\sigmal))*((h_R*sp(:,kk))-(H_F*Xf)));
deltau=deltaU(1:n_in,1);

%    DeltaU2=inv(H_H+0.1*eye(Nc,Nc))*(H_R*sp(kk)-H_F*Xf);
%    deltau2=DeltaU2(1,1);
%    deltau=[deltau1;deltau2];
u=u+deltau;
u1(:,kk)=u;
y1(:,kk)=y;
xm_old=xm;
xm=Ap*xm+Bp*u;
y=Cp*xm;
Xf=[xm-xm_old;y];
end
k=0:(N_sim-1);
figure
plot(k,y1(1,:),'r',k,y1(2,:),'g',k,y1(3,:))
% plot(k,y1(1,:),'r',k,y1(2,:),'g')
```

Appendix - C8: FOPPI Matlab Program for Minh Petroleum Distillation Control Example

```
clear all
Ap=[-0.5105 0;0 -0.5105];%MalaysiaColumn
Bp=eye(2);
Cp=[0.0021 -0.0031;-0.0026 0.0037];
%SlovakiaColumnStartsHere
% Ap=[-0.2616 0;0 -0.2616];%MalaysiaColumn
% Bp=eye(2);
% Cp=[0.0040 -0.0070;-0.0011 0.0017];
Np=80;
```

```
Nc=20;
[H,f,Rs,H_H,H_F,A_e,B_e,C_e] =predictGainMIM22(Ap,Bp,Cp,Nc,Np);
% [H,f,BarR,H_H,H_F,H_R,A_e,B_e,C_e]
=predictivegain1(Ap,Bp,Cp,Nc,Np);
%
[H,f,Rs,H_H,H_F,H_R,A_e,B_e,C_e]=predictGainMIMO22(Ap,Bp,Cp,Nc,Np);
[n,n_in]=size(B_e);
[m1,n1]=size(Cp);
[n1,n_in]=size(Bp);
 xm=zeros(n1,1);
 Xf=zeros(n,1);
N_sim=150;
% r=ones(N_sim,1);
% u=0;
% y=0;
r1=0.0261*ones(1,N_sim+10);
r2=-0.0275*ones(1,N_sim+10);
r=[r1;r2];
n=n1+m1;
u=zeros(n_in,1);y=zeros(m1,1);
h_r=(H_F(:,3:4));
for kk=1:N_sim;
DeltaU1=inv((H_H+(0.00002*eye(Nc,Nc))))*(((h_r*r(:,kk))-(H_F*Xf)));
deltau=DeltaU1(1:n_in,1);

%    DeltaU2=inv(H_H+0.1*eye(Nc,Nc))*(H_R*sp(kk)-H_F*Xf);
%    deltau2=DeltaU2(1,1);
%    deltau=[deltau1;deltau2];
u=u+deltau;
u1(:,kk)=u;
y1(:,kk)=y;
xm_old=xm;
xm=Ap*xm+Bp*u;
y=Cp*xm;
Xf=[xm-xm_old;y];
end
k=0:(N_sim-1);
plot(k,y1(1,:),'r',k,y1(2,:),'g')
```

References

Albertos, P. & Sala, A., 2004. *Multivariable Control Systems.* London: Springer-Verlag.

Almeida, O. & Coelho, A. A. R., 2002. *A Fuzzy Logic Method of Autotuning a PID Controller: SISO and MIMO Systems.* Barcelona, IFAC 15th Triennial World Congress. pp. 1-6.

Anon., 2002. The MIMO Predictive PID Controller Design. *Asian Journal of Control,* 4(4), pp. 452-463.

Anon., 2017. Design of a Fractional Order Frequency PID Controller for an Islanded Microgrid: A Multi-Objective Extremal Optimization Method. *energies,* 10(10), pp. 1502-1520.

Arlt, W., 2014. Azeotropic Distillation. In: A. Gorak & O. Zarko, eds. *Distillation (Equipment and Processes).* USA.:Academic Press, pp. 247-259.

Astrom, K. & Hagglund, T., 1995. *PID Controllers: Theory, Design and Tuning.*. 2nd ed. Durham, USA.: ISA.

Astrom, K. J. & Hagglund, T., 2006. *Advanced PID Control.* New York: International Society of Automation.

Balchen, J. G., 1988. *Process Control:Structure and Applications.* New York: Van Nostrand Reinhold.

Baruah, G., Majhi, S. & Mahanta, C., 2016. *Fractional order PID controller design for an SOPDT Model by online tuning method.* Hyderabad,India., IEEE Xplore/Indian Control Conference 2016 pp. 450-455.

Besta, C. S. & Chidambaram, M., 2015. Tuning of Multivariable PI Controllers by BLT. *Chemical Engineering Communications,* 203(4), pp. 527-538.

Bouhajar, S. et al., 2015. Trajectory Generation using Predictive PID Control for Stable Walking Humanoid Robot. *Procedia Computer Science,* Volume 73, pp. 86-93.

Camacho, E. & Bordons, C., 1999. *Model Predictive Control.* London: Springer-Verlag.

Chang, W., 2007. A multi-crossover genetic approach to multivariable PID controllers tuning. *Expert systems with Applications,* Volume 33, pp. 620-626.

Chen, T. & Marquez, H., 2008. Robust Controller Design and PID Tuning for Multivariable Processes. *Asian Journal of Control,* 4(4), pp. 439-451.

Davison, E., 1976. The robust control of a servomechanism problem for linear time-invariant multivariable systems. *IEEE Transactions on Automatic Control* , 21(1), pp. 25-34.

Duan, X.G., Li, H.X. & Deng, H., 2008. *A simple tuning method for fuzzy PID control.* Hong Kong, IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), pp. 271-275.

Edet, E. & Katebi, R., 2016. *Design and Tuning of Fractional-order PID Controllers for Time-delayed Processes.* Belfast, 2016 UKACC 11th International Conference on Control (CONTROL). pp. 1-6.

Edet, E. & Katebi, R., 2017. On Fractional Predictive PID Controller Design Method. *IFAC-PapersOnLine,* 50(1), pp. 8555-8560.

Edet, E. & Katebi, R., 2018. *Fractional-order Predictive PI Controllers for Process Control Systems,* Glasgow: IET (Under Review).

Edet, E. & Katebi, R., 2018. On Fractional-order PID Controllers. *IFAC-PapersOnLine,* 51(4), pp. 739-744.

Eranda, H. & Mann, G. K., 2008. Design and tuning of standard additive model-based fuzzy PID controllers for multivariable process systems. *IEEE transactions on systems,MAN and Cybernetics,* 38(3).

Esko, J., 2018. *Modelling and Simulation in Advanced Control,* Finland: University of Oulu.

Ghoshal, S., 2004. Optimization of PID gains by particle swarm optimizations in fuzzy based automatic generation control. *Electric power system Research,* Volume 72, pp. 203-212.

Giwa, A. & S.Karacan., 2012. State Space Model Predictive Control of a Reactive Distillation Process. *International Journal of Pure and Applied Sciences and Technology,* 12(2), pp. 20-33.

Haydary, J., 2009. Steady State and Dynamic Simulation of Crude Oil Distillation Using ASPEN Plus and ASPEN Dynamics.. *Petroleum and Coal,* 51(2), pp. 100 - 109.

Ho, W. & Xu, W., 1998. *Multivariable PID Controller Design Based on the Direct Nyquist Array Method.* Philadelphia, American Control Conference&IEEE, pp. 3524-3528.

Hu, B.G., 2001. A systematic study of fuzzy PID controllers - Function based evaluation approach. *IEEE transactions on Fuzzy systems,* 9(5), pp. 699-711.

Huynh, T.H., 2008. *A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers.* Chengdu, IEEE International Conference on Industrial Technology, pp. 1-6.

Iruthayarajan, W. & Baskar, S., 2009. Evolutionary algorithms based design of multivariable PID controller. *Expert systems with applications,* Volume 36, pp. 9159-9167.

Johnson, M. & Moradi, M., 2006. *PID Control: New Identification and Design Methods.* London: Springer.

Karacan, S., Cabbar, Y., Alpbaz, M. & Hapoglu, H., 1998. The steady-state and dynamic analysis of packed distillation column based on partial differential approach. *Chemical Engineering and Processing,* 37(5), pp. 379-388.

Katebi, M. & Moradi, M., 2001. Predictive PID Controllers. *IEE Proceeding - Control Theory Application,* 148(6), pp. 478-486.

Katebi, M. et al., 2001. Robust Dynamic Ship Positioning Control System Design and Applications. *International Journal of Robust and Nonlinear Control,* Volume 11, pp. 1257 - 1284.

Katebi, R., 2012. Robust Multivariable Tuning Methods. In: R. Vilanova & A. Visioli, eds. *PID Control in the Third Millennium (Lessons Learned and new Approaches).* London: Springer-Verlag, pp. 255 - 279.

Lanusse, P. & Sabatier, J., 2011. PLC Implementation of a Crone Controller. *Fractional Calculus and Applied Analysis,* 14(4), pp. 505-522.

Lee, C.H. & Chang, F.K., 2010. Fractional-order PID controller optimization via improved electromagnetism-like algorithm. *Expert Systems with Applications,* 37(12), pp. 8871-8878.

Lee, J., Roh, M., Lee, J. & Lee, D., 2007. Clonal Selection Algorithms for 6-DOF PID Control of Autonomous Underwater Vehicles. In: L. Castro, ed. *Artificial Immune System.* Berlin: Springer-Verlag, pp. 182 - 190.

Lieslehto, J., Tanttu, J. & Koivo, H., 1993. An Expert System for Multivariable Controller Design. *Automatica,* 29(4), pp. 953-968.

Li, Z. & Chen, Y., 2014. *Ideal, Simplified and Inverted Decoupling of Fractional Order TITO Processes.* Cape-Town, IFAC, pp. 2897-2902.

Loh, A., Chang, C., Choon, K. & Vasnani, V., 1993. Autotuning of Multiloop Proportional- Integral Controllers Using Relay Feedback. *Ind.Eng.Chem.,* Volume 32, pp. 1102 - 1107.

Luo, Y. & Chen, Y., 2009. Fractional order [proportional derivative] controller for a class of fractional order systems. *automatica,* Volume 45, pp. 2446-2450.

Luo, Y., Chen, Y., Wang, C. Y. & Pi, Y. G., 2010. Tuning fractional order proportional integral controllers for fractional order system. *Journal of Process Control,* 20(7), pp. 823-831.

Luyben, W., 1986. Simpte Method for Tuning SISO Controllers in Multivariable Systems,. *Ind.Eng.Chem.Process Des. Dev.,* 26(3), pp. 654-660.

Maciejowski, J., 1989. *Multivariable Feedback Design.* New York: Addison Wesley.

Mann, G. K. & Harinath, E., 2008. Two-level tuning of fuzzy PID controllers for multivariable process systems. In: R. Lowen & A. Verschoren, eds. *Foundations of Generic Optimization.* Antwerp: Springer, pp. 283-311.

Mann, G. K., Hu, B.-G. & Gosine, R. G., 2001. Two level tuning of Fuzzy PID controllers. *IEEE transactions on Systems, MAN and Cybernetics,* 31(2), pp. 263-270.

Martin, P. & Katebi, R., 2005. Multivariable PID tuning of dynamic ship positioning control systems. *Journal of Marine Engineering and Technology,* Volume A7, pp. 11 - 25.

McFarlane, R. & Rivera, D., 1992. Identification of distillation systems. In: W. Luyben, ed. *Practical Distillation control.* New York: Van Nostrand Reinhold, pp. 97-140.

Minh, V. T., 2009. Modelling and Control of Distillation Column in a Petroleum Process. *Mathematical Problems in Engineering,* Volume 2009, p. 14.

Monje, C. et al., 2010. *Fractional-order Systems and Controls: Fundamentals and Applications.* New York: Springer-Verlag.

Morari, M., 2002. *Robust Process Control.* London: Prentice-Hall.

Mukherjee, V. & Ghoshal, S., 2007. Intelligent Particle Swarm optimized fuzzy PID controller for AVR system. *Electric power system research,* 77(12), pp. 1689-1698.

Naeem, W., Sutton, R. & Chudley, J., 2004. *Model predictive control of an autonomous underwater vehicle with a fuzzy objective function optimized using a GA.* Ancona, Italy, International Federation of Automatic Control, pp. 433-438.

Naeem, W. et al., 2005. An online genetic algorithm based model predictive control autopilot design with experimental verification. *International Journal of Control,* 78(14), pp. 1076-1090.

Ogannaike, B. & Ray, W., 1994. *Process Dynamics, Modelling and Control.* New York: Oxford University Press.

Ogunnaike, B. A. & Ray, W., 1983. Advanced Multivariable Control of a Pilot Plant Distillation Column. *AlChE Journal,* 29(4), pp. 632-640.

Padula, F., Vilanova, R. & Visioli, A., 2014. H-infinity optimization based fractional-order PID controllers design.. *International Journal of Robust and Nonlinear Control,* 24(17), p. 3009–3026.

Palmor, Z., Halevi, Y. & Krasney, N., 1995. Automatic tuning of decentralised PID controllers for TITO processes. *Automatica,* 31(7), pp. 1001 -1010.

Penttinen, J. & Koivo, H., 1980. Multivariable Tuning Regulators for Unknown Systems. *Automatica,* Vol. 16(4), pp. 393-398.

Podlubny, I., 1999. Fractional-Order Systems and Fractional-order PID-Controllers. *IEEE Transactions on Automatic Control,* 44(1), pp. 208-214.

Rahmati, A., Rashidi, F. & Rashidi, M., 2003. *A hybrid Fuzzy Logic and PID controller for control of non-linear HVAC systems.* Washington, *IEEE International Conference on Systems, Man and Cybernetics*. pp. 2249-2254.

Rehab, R., Remy, M. & Renotte, C., 2001. *An approach to Design Fuzzy PI supervisor for nonlinear system.* Brussels, *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*. pp. 894-899

Rijnsdorp, J. & Papadourakis, A., 1992. Approximate and simplified models. In: W. L. Luyben, ed. *Practical distillation control.* New York: Van Nostrand Reinhold, pp. 48-71.

Saeed, Q., Uddin, V. & Katebi, R., 2010. Multivariable Predictive PID Control for Quadruple Tank. *World Academy of Science, Engineering and Technology* , Volume 43, pp. 861-866.

Seborg, D., Mellichamp, D., Edgar, T. & Doyle, F., 2010. *Process Dynamics and Control.* 3rd ed. London: John Wiley&Sons.

Shiu, S.J. & Hwang, S.H., 1998. Sequential Design Method for Multivariable Decoupling and Multiloop PID Controllers. *Ind. Eng. Chem. Res.,*, 37(1), p. 107–119.

Skogestad, S., 1997. Distillation Column Dynamics and Control: A Tutorial Introduction. *IChemE,* 75(A).

Skogestad, S., 2007. The Dos and Don'ts of Distillation Column Control. *IChemE,* 85(A1), pp. 13-23.

Skogestad, S. G. C., 2012. The SIMC Method for Smooth PID Controller Tuning. In: R. Vilanova & A. Visioli, eds. *PID Control in the Third Millennium - Lessons Learned and New Approaches.* London: Springer-Verlag, pp. 147-175.

Skogestad, S. & Postlethwaite, I., 2005. *Multivariable Feedback Control.* 2nd ed. Sussex: John Wiley and sons.

Song, X., Chen, Y., Tejado, I. & Vinagre, B. M., 2011. *Multivariable Fractional PID Controller Design via LMI approach.* Milan, IFAC, pp. 13960-13965.

Tellez-Anguiano, A., Rivas-Cruz, F. & Astora-Zaragoza, C.M., 2009. Process Control Interface System for a Distillation Plant. *Computer Standards and Interface,* 31(8), pp. 471-479.

Tepljakov, A. et al., 2018. FOPID Controllers and Their Industrial Applications: A Survey of Recent Results. *IFAC-PapersOnLine,* 51(4), pp. 25-30.

Tyreus, B., 1992. Object-oriented simulation. In: W. Luyben, ed. *Practical distillation control.* New York: Van Nostrand Reinhold, pp. 72-95.

Vale´rio, D. & Costa, J. S. d., 2009. Tuning of Fractional PID Controllers with Ziegler Nichols Type Rules. *Signal Processing,* 86(10), pp. 2771-2784.

Vilanova, R. & Katebi, R., 2009. *Multi-loop PI-based control Strategies for the Activated Sludge Process.* Mallorca, IEEE Emerging Technologies and Factory Automation Conference. pp.1 - 8.

Vinagre, B., Podlubny, I., Hernandez, A. & Feliu, V., 2000. Some approximations of fractional order operators used in control theory and applications. *Fractional calculus and applied analysis,* 3(3), pp. 231-248.

Vu, T., Lee, J. & Lee, M., 2007. Design of Multi-loop PID Controllers Based on the Generalized IMC-PID Method with Mp Criterion. *International Journal of Control, Automation and Systems,* 5(2), pp. 212-217.

Vu, T. & Lee, M., 2008. *Multiloop PI/PID Controller Design Based on Direct Synthesis for Multivariable Systems.* San Francisco, World Congress on Engineering and Computer Science, pp. 1-5.

Wahab, N., Katebi, R. & Balderud, J., 2007. *Multivariable PID Control Design for Waste-water Systems.* Athens, IEEE Mediterranean Conference on Control & Automation, pp. 1 - 6.

Wahab, N., Katebi, R. & Balderud, J., 2009. Multivariable PID Control Design for Activated Sludge Process with Nitrification and Denitrification. *Biochemical Engineering Journal,* Volume 45, pp. 239-248.

Wang, J., Zhang, Y. & Wang, W., 2006. Optimal design of PI/PID controller for non-minimum phase system. *Transactions of Institute of Measurement and Control,* 28(1), pp. 27-35.

Wang, L., 2009. *Model Predictive Control System Design and Implementation using Matlab.* London: Springer-Verlag.

Wang, Q.G., Ye, Z., Cai, W.J. & Hang, C.C., 2008. *PID Control for Multivariable Processes.* Berlin: Springer.

Wang, Q.G., Zou, B., Lee, T.H. & Bi, Q., 1997. Auto-tuning of multivariable PID controllers from decentralised relay feedback. *Automatica,* 33(3), pp. 319-330.

Wu, Z. et al., 2018. Tuning for Fractional Order PID Controller based on Probabilistic Robustness. *IFAC-PapersOnLine,* 51(4), pp. 675-680.

Xu, J.X., Hang, C.C. & Liu, C., 2000. Parallel structure and tuning of a fuzzy PID controller. *Automatica,* Volume 36, pp. 673-684.

Xu, J.X., Pok, Y.M., Liu, C. & Hang, C.C., 1998. Tuning and Analysis of a fuzzy PI controller based on gain and phase margins. *IEEE transactions on systems,Man and Cybernetics,* 28(5), pp. 685-691.

Yao, L. & Wen, H.K., 2012. Design of Observer based Adaptive PID Controller for Non-linear Systems. *International Journal of Innovative Computing, Information and Control,* pp. 667-677.

Yongquan, Y. et al., 2004. *Fuzzy-Neural PID controller and tuning its weight factors using genetic algorithm based on different location crossover*. The Hague, IEEE International Conference on Systems, Man and Cybernetics, pp. 3709-3713.

Zhuang, M., 1994. PID control for TITO System. *IEE Proceedings*, 141(2), pp. 111-120.