Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

Simen Bjorvand

Control structures with embedded process knowledge

Master's thesis in MTKJ

Supervisor: Sigurd Skogestad and Cristina Zotica

June 2020

NTNU
Norwegian University of
Science and Technology

Simen Bjorvand

Control structures with embedded process knowledge

Master's thesis in MTKJ
Supervisor: Sigurd Skogestad and Cristina Zotica
June 2020

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

In this thesis, a new methodology for transforming the nonlinear process into a linear process that gives perfect disturbance rejection and decoupling with the use of a manipulated variable (MV) transformation has been studied. This new method is similar to the feedback linearization methodology, but is simpler to apply, and does not transform the nonlinear system into a chain of integrators, which gives increased control limitations. Different methods of designing transformed MVs v, which achieves these goals have been proposed. The two primary strategies are the static transformation and the linear transformation. The static transformation is designed from a static model and does not require implicit output feedback, however, it does not transform the nonlinear processes into linear ones when applied to dynamic systems. The linear transformation is derived from dynamic models and does linearise the nonlinear processes, however, it requires implicit feedback. Two case studies were carried out for an extraction process and a mixing tank system to evaluate different strategies to design transformed MVs. With no model error, perfect disturbance rejection was achieved with both the linear and static transformation. Despite not linearizing the process the static transformation performed well for slow closed-loop dynamics and was shown to be robust to both model error and measurement delay. The linear transformation was best for faster closed-loop dynamics but was shown to be less robust to model error and measurement delay. A multivariable (MIMO) process was considered for a continuous stirred tank reactor (CSTR). With no model errors the new method gave perfect decoupling, however, the new methodology might not be too robust to model errors. A condensation process was considered as well where a cascade structure was applied together with the new MV transformation, to simplify the model equations, which was shown to work well.

Preface

This thesis concludes the master program in Chemical Engineering at the Norwegian University of Science and Technology NTNU.

I would like to thank my supervisor Sigurd Skogestad for his involvement and interest in my work, and the willingness to discuss. I will also thank my co-supervisor Cristina Zotica for great guidance and contributions. I would also like to thank my fellow master students in the systems group for a great year, making every day a day to look forward to.

Table of Contents

List of Tables

List of Figures

Nomenclature

Common

x	=	State vector
u	=	Input vector
u	=	Disturbance vector
y	=	Output vector
v	=	Transformed input vector
r	=	Relative order of system
y_s	=	Output setpoint
e	=	Output setpoint deviation
K_c	=	Controller gain
τ_I	=	Integral time
τ_c	=	Closed-loop time constant
T_{12}	=	Input-input decoupling block from input 1 to input 2
T_{21}	=	Input-input decoupling block from input 2 to input 1

Chapter 1

u_0	=	PID controller bias
τ_D	=	Derivative time
c	=	Controller transfer function
k	=	Process gain
τ	=	Open-loop time constant
$k^{'}$	=	Integral process gain
θ	=	Time delay

Chapter 2

A	=	Linear transformation tuning matrix
v_L	=	Linear transformation manipulated variable
T	=	Inverse linear transformation tuning matrix
v_0	=	Static transformation manipulated variable
v_N	=	Nonlinear transformation manipulated variable
v_G	=	General transformation manipulated variable

Chapter 3

F	=	Acid water mixture feed stream
E	=	Extract product stream
S	=	Solvent feed stream
R	=	Raffinate product stream
M_E	=	Extract holdup
M_R	=	Raffinate holdup
x_{AE}	=	Acid mole fraction in Extract product stream
x_{WE}	=	Water mole fraction in Extract product stream
x_{AF}	=	Acid mole fraction in acid water mixture feed stream
K	=	Water acid equilibrium constant in organic phase
τ_r	=	Residence time in the organic phase
K_{c1}	=	Proportional gain for total hold up controller
K_{c1}	=	Proportional gain for raffinate hold up controller
v_{i1}	=	Transformed method designed with method i dependent on x_{AF} and E
v_{i2}	=	Transformed method designed with method i dependent on R and E
v_{i2}	=	Transformed method designed with method i dependent on R and S

Chapter 4

q_1	=	Inlet flow 1
q_2	=	Inlet flow 2
q	=	Outlet flow
c_1	=	concentration of inlet flow 1
c_2	=	concentration of inlet flow 2
M	=	Mixing tank volume
ρ	=	Density
τ_r	=	Residence time in the mixing tank
v_0	=	Zeroth transformed variable, is the physical input and represents the basecase
v_1	=	First transformed variable
v_2	=	Second transformed variable, static transformation
v_3	=	Third transformed variable, linear transformation
v_4	=	Fourth transformed variable, ratio control
v_5	=	Fifth transformed variable, feedback linearization
k	=	Process gain
τ	=	Open-loop time constant
$k^{'}$	=	Integral process gain
θ	=	Output measurement time delay
K_u	=	Input gain mismatch
u_r	=	Actual effect of input
K_d	=	Disturbance gain mismatch
d_r	=	Actual effect of disturbance

Chapter 5

q_f	=	Feed stream
q	=	Outlet stream
A	=	Reactant chemical component
B	=	Product chemical component
c_{Af}	=	Concentration of component A in feed stream
c_A	=	Concentration of component A in product stream
k	=	Second order reaction coefficient
V	=	Reactor volume
ρ	=	Density
A_d	=	Diagonal tuning matrix for linear transformation

Chapter 6

w_d	=	Saturated steam stream
w	=	Water stream
m	=	Hold up on cold side
T	=	Temperature on the cold side
T_d	=	Temperature on the hot side
p_d	=	Pressure on the hot side
c_p	=	Heat capacity
T_{dref}	=	Reference temperature
p_{dref}	=	Saturated steam pressure at T_{dref}
λ	=	Enthalpy of vaporization at T_{dref}
A	=	Heat transfer area
U	=	Heat transfer coefficient
z	=	Valve position
k	=	Valve coefficient
T_0	=	Inlet temperature on the cold side
T_{d0}	=	Inlet temperature of saturated steam
p_{d0}	=	Inlet pressure of saturated steam
E	=	Internal energy
Q	=	Transfered heat
R	=	Ideal gas constant
A_z	=	Tuning parameter for linear transformation when z is considered the physical input
v_{Lz}	=	Linear transformed MV for the case when z is considered the physical input
A_{wd}	=	Tuning parameter for linear transformation when w_d is considered the physical input
v_{Lwd}	=	Linear transformed MV for the case when w_d is considered the physical input
A_{Td}	=	Tuning parameter for linear transformation when T_d is considered the physical input
v_{LTd}	=	Linear transformed MV for the case when T_d is considered the physical input

Chapter 1

Introduction

1.1 Motivation

Control of non-linear systems can be a challenging and difficult task which is important as many processes have nonlinear behavior. There are many different techniques for controlling non-linear systems. One method is to create a linear approximation of the system around some operating point using the Jacobian and controlling this system using linear control theory, with either PID-controllers or linear model predictive control[14]. This however requires that the linearized system is a good approximation in the entire operation range which is not the case for many highly nonlinear systems. In this thesis, a new methodology that fully linearizes nonlinear systems over the entire operation range is looked into, such that the linear system can be controlled using linear control theory. This new methodology also introduces disturbance rejection, and for MIMO (multiple inputs multiple outputs) systems decoupling. This is done with a manipulated variable transformation.

In literature, many different techniques for controlling non-linear processes have been proposed. Nonlinear model predictive control (NMPC), or adaptive control[1] are two control techniques that use different approaches to control nonlinear processes. There also exist techniques in literature with a similar goal of transforming nonlinear systems into linear systems. The feedback linearization[7] is a nonlinear control technique that uses differential approaches[8] to transform nonlinear systems into linear ones. Input-output linearization[6] is a similar approach to feedback linearization which linearizes the map from the inputs to the outputs, leaving parts of the model nonlinear. Other similar methods include internal decoupling[3], and elementary nonlinear decoupling[2].

In this thesis, this new methodology will be applied to several case studies, where different properties of this new methodology will be studied. In the first chapter of this thesis some background theory will be presented, followed by a presentation of the new methodology in chapter two. In chapters, three to six, four case studies will be conducted. Chapter seven

is a discussion, with the last chapter being the conclusion.

1.2 Theory

1.2.1 Feedback linearization

Feedback linearization is covered in numerous literatures [6, 7, 12, 13]. Feedback Linearization is a class of nonlinear control techniques that transforms a nonlinear model into a linear one which can be controlled using linear control theory, which was researched in the 1970s and 1980s[12]. The feedback linearization techniques work by transforming a non-linear system of n^{th} order to a chain of r integrators, where r is the relative order of the system. The relative order for a non-linear system is the number of times the output must differentiate with respect to time for the input to explicitly appear in the equation. For a linear system, the relative degree is the difference between the number of poles and the number of zeros in the system. The chain of integrators which is formed gives a linear map which allows for linear control theory to be applied to design a control system.

Assumptions

The main assumptions which must be fulfilled for feedback linearization are

- All the states can be either measured or estimated.
- The nonlinear model must be linear in the inputs, meaning the model equation must be on the form $\dot{x} = a(x) + b(x)u$
- It can only be applied to minimum phase systems, meaning there are no right-hand plane zeros or time delay.
- The process must be dynamic, meaning it cannot be used for static processes.

Feedback linearization or *state space linearization* as it is called in [6] and *Input-output linearization* are two similar approaches. State-space linearization creates a linear mapping between the transformed inputs and the states linearizing the entire system. The drawback of state-space linearization is that it does not consider nonlinearities in the state-output mapping. State-space linearization is used more for stabilizing purposes where the output is not necessarily stated beforehand, however, this is usually not the case for process control applications.

The feedback linearization procedures for linearizing non-linear systems contains two steps. The first step is a nonlinear coordinate transformation using differential geometry. The second step is introducing state feedback. To explain how the nonlinear coordinate transformation is achieved, a concept known as the Lie derivative is useful.

Considering the non linear single-input-single-output (SISO) model

$$\frac{dx}{dt} = f(x) + g(x)u \tag{1.1}$$

$$y = h(x) \tag{1.2}$$

where y is the output, x is the n state variables and u is the physical input. The nonlinear vector spaces $f(x)$ and $g(x)$ maps the n-dimensional state vector x to an n-dimensional space. The non-linear scalar function $h(x)$ maps the n-dimensional state vector x to a scalar. Assume $f(x)$, $g(x)$ and $h(x)$ are C^{∞}, which means there exist a partial differential of any order, and it is continuous. Using the Lie derivative this non linear model can be transformed into a form called *Byrnes-isidori* normal form[4], with a non linear coordinate transformation.

Lie-derivative

Considering the scalar function $h(x)$ and the vector field $f(x)$, the Lie derivative of $h(x)$ along $f(x)$ is

$$L_f h(x) = \frac{\partial h}{\partial x} f(x) \tag{1.3}$$

The Lie derivative is itself a scalar so repeating Lie derivatives is possible. Considering the vector field $g(x)$, the Lie derivative of $L_f h(x)$ along $g(x)$ is

$$L_g L_f h(x) = L_g(L_f h(x)) = \frac{\partial L_f h(x)}{\partial x} g(x) \tag{1.4}$$

Subsequent Lie derivatives of a scalar function along the same vector field is possible,

$$L_f L_f h(x) = L_f^2 h(x) \tag{1.5}$$

The k^{th} Lie derivative the scalar function $h(x)$ along $f(x)$ is denoted $L_f^k h(x)$, with the special case of the 0^{th} Lie derivative being $L_f^0 h(x) = h(x)$.

For non-linear systems on the form given in eqs. (1.1) and (1.2) the relative degree r of the system is is the smallest integer for which the Lie derivative

$$L_g L_f^{r-1} h(x) \neq 0 \tag{1.6}$$

for $k < r - 2$

$$L_g L_f^k h(x) = 0 \tag{1.7}$$

for all x in some neighbourhood of a defined operating point x_0,

1.2.2 Input-output linearization

Input-output linearization creates a linear mapping between the transformed inputs and output leaving parts of the original system non-linear.

Using the Lie derivate on the nonlinear model in eqs. (1.1) and (1.2) it can be transformed into the following form

$$\begin{pmatrix}\xi\\ \eta\end{pmatrix} = \Phi(x) = \begin{pmatrix} h(x)\\ L_f h(x)\\ \vdots\\ L_f^{r-1}h(x)\\ t_1(x)\\ \vdots\\ t_{n-r}\end{pmatrix} \tag{1.8}$$

The transformation $\Phi(x)$ is assumed to be invertible for a neighbourhood of the defined operating point x_0. $\Phi(x)$ is split into two parts, ξ which is the linearized dynamics of the system, and η which are the parts of the system kept nonlinear. r is the relative degree of the system. The functions $t_i(x)$ are found from the solution of

$$L_g t(x) = 0 \tag{1.9}$$

From this coordinate transformation $y = h(x) = \xi_1$. The time derivative of y is

$$\begin{aligned}\frac{dy}{dt} &= \frac{dh(x)}{dt} = \frac{\partial h(x)}{\partial t}\frac{dx}{dt}\\ &= \frac{\partial h(x)}{\partial t}(f(x) + g(x)u)\\ &= L_f h(x) + L_g h(x)\\ &= L_f h(x)\end{aligned} \tag{1.10}$$

which means $dy/dt = \xi_2$. The last step is due to $L_g L_f^k h(x) = 0$ for $k < r$, and in this case $k = 0$. Similarly $d^2y/dt^2 = \xi_3$ up to $d^{r-1}y/dt^{r-1} = \xi_r$. The Byrnes-Isisdori normal form is found by differentiating ξ and η with respect to time, giving

$$\begin{aligned} y &= \xi_1\\ \frac{d\xi_1}{dt} &= \xi_2\\ \frac{d\xi_2}{dt} &= \xi_2\\ &\vdots\\ \frac{d\xi_{r-1}}{dt} &= \xi_r\\ \frac{d\xi_r}{dt} &= L_f^r h(x) + L_g L_f^{r-1} h(x) u\\ \frac{d\eta_1}{dt} &= L_f t_1(x)\\ &\vdots\\ \frac{d\eta_{n-r}}{dt} &= L_f t_(n-r)(x)\end{aligned} \tag{1.11}$$

This transformed system forms a chain of integrators. $L_f^r h(x)$, $L_g L_f^{r-1} h(x)$ and $L_f t_i(x)$, can be reformulated in terms of ξ and η by using the transformation given in eq. (1.8), which yields

$$\alpha(\xi, \eta) = L_f^r h(\Phi^-(\xi, \eta)) \tag{1.12}$$

$$\beta(\xi, \eta) = L_g L_f^{r-1} h(\Phi^-(\xi, \eta)) \tag{1.13}$$

$$q_i(\xi, \eta) = L_f t_i(h(\Phi^-(\xi, \eta)) \tag{1.14}$$

For the transformed model a transformed MV v can be formulated which linearizes the mapping between the transformed input v and the output y using state feedback,

$$u = \frac{v - \alpha(\xi, \eta)}{\beta(\xi, \eta)} \tag{1.15}$$

This yield the system

$$\frac{d\xi_r}{dt} = \frac{d^r y}{dt^r} = v \tag{1.16}$$

which is linear and linear control theory can be used to design a control system. A limitation of this method is that the subsystem $d\eta/dt = q(\xi, \eta)$ called the *zero dynamics*, which must not be mixed up with zero dynamics of a linear system, must be stable.This method can be further extended to include disturbance decoupling[8], and MIMO systems[8].

The main limitations of the feedback linearization methods are[19]

- Requires the model to be inverted, which makes it not robust to model error.
- Challenging to extend to multivariate systems, as it requires a type of nonrobust nonlinear decoupling.
- Process constraints are not handled explicitly.
- All the states must be either measured or estimated.
- It cannot deal with uncertainty in RHP zeros and time delays.
- It cannot be applied to static systems.
- Transforms the system to a chain of integrators which adds control limitations.

For mechanical systems with few states which are easily measured or estimated these limitations may be acceptable. This is usually not the case for process control systems, which can be characterized by many states which are difficult to measure. Besides, mechanical systems are often integrating systems from the beginning so transforming them into a chain of integrators does not give extra control limitations. However, there are some processes for which feedback linearization or input/output linearization can be applied. Input-output linearization can be was applied to a ph-neutralization control problem in [5] and for a batch reactor example in [10]. A variety of input/output linearization was applied to a continuous stirred tank reactor example in [11].

1.2.3 SIMC-tuning rules

Different types of controller exist which can be used to control processes. A commonly used technique to control processes is feedback control. Feedback control is a type of control where measurements of the *control variable* (CV) is feed into a controller and based on this measurement (deviation from its setpoint) a control decision in the *manipulated variable* (MV) is made by the controller. A commonly used feedback controller is the PID controller or its variants the PI controller or P controller. PID is an acronym for *proportional*, *integral* and *derivative*, which are the three main components of this controller. The PID controller is a setpoint tracking controller, which means it tries to track its control variable to a given setpoint value. The equation for a PID controller is given below.

$$u(t) = u_0 + K_c(e(t) + \frac{1}{\tau_I}\int_0^t e(\tau)d\tau + \tau_D\frac{de(t)}{dt}) \tag{1.17}$$

In this equation $u(t)$ is the manipulated variable, u_0 is the *controller bias*, K_c is the *controller gain*, τ_I is the *integral time*, τ_D is the derivative time and $e(t)$ is difference between the setpoint and the measured value of the control variable also called error

$$e(t) = y_s(t) - y(t) \tag{1.18}$$

Primarily in this thesis PI controllers will be used, where the derivative term is dropped. Equation (1.17) was written in the time domain, but frequency domain representation is useful and is for a PI-controller

$$\frac{u}{e}(s) = c(s) = K_c(1 + \frac{1}{\tau_I s}) = K_c(\frac{\tau_I s + 1}{\tau_I s}) \tag{1.19}$$

For the PID controller three tuning parameters were introduced, being K_c, τ_I and τ_D. For a PI controller the derivative time is dropped, however two tuning parameters must still be selected. Several techniques have been proposed to successfully tune PID-controllers to obtain stability and fast control. In this section the SIMC-tuning rules[17] will be presented.

The SIMC-tuning rules are simple tuning rules to tune PID controllers for linear first order process and integrating processes with time delay. The SIMC-tuning rules are based on IMC tuning rules, and the idea is to design the closed loop response of the system.

Consider a linear first order response process with the transfer function $g(s)$ from u to y

$$g(s) = \frac{k}{\tau s + 1}e^{-\theta s} \tag{1.20}$$

where k is the process gain, τ is the open-loop time constant, and θ is time delay. The SIMC tunings for a first order process are

$$K_c = \frac{1}{k}\frac{\tau}{\tau_c + \theta} \tag{1.21}$$

$$\tau_I = min(\tau, 4(\tau_c + \theta)) \tag{1.22}$$

for an integrating process on the form

$$g(s) = \frac{k^{'}}{s} e^{-\theta s} \tag{1.23}$$

where $k^{'}$ is the integrating process gain, the SIMC tunings are

$$K_c = \frac{1}{k^{'}} \frac{1}{\tau_c + \theta} \tag{1.24}$$

$$\tau_I = 4(\tau_c + \theta) \tag{1.25}$$

By using the SIMC-tuning rules the amount of tuning parameters have been reduced from two for a PI-controller too one being the closed loop time constant τ_c. The shape of the setpoint response is dependent on how this tuning parameter is selected. If it is selected to be small, the response will be fast and disturbances will be quickly be rejected, however this might not be robust. Selecting τ_c to be large will give stable and robust control. A trade off between the two must be made, *Skogestad*[17] recommends choosing $\tau_c = \theta$ as a good compromise.

The SIMC tuning rules for tuning PI-controllers can only be tuned using first order response process models or integrating process models, however in [17] simple analytical rules for reducing models is proposed. The rule in question is *Skogestads half-rule* where higher order process models can be reduced to first order or integrating processes with time delay. Consider the following higher order process model

$$g(s) = k \frac{\prod_{i=1}^{n} T_i s + 1}{\prod_{j=1}^{N} \tau_j s + 1} e^{-\theta s} \tag{1.26}$$

where $N > n$, $T_i > T_{i+1} \forall i \in \{1, \ldots, n-1\}$, and $\tau_j > \tau_{j+1} \forall j \in \{1, \ldots, N-1\}$. In the half rule first order Taylor approximation of the exponential function is used

$$e^{as} \approx 1 + as \tag{1.27}$$

$$e^{-as} = \frac{1}{e^{as}} \approx \frac{1}{1 + as} \tag{1.28}$$

This approximation is used to approximate the neglected zero dynamics, and the neglected pole dynamics as effective time delay. The half rule is that the largest neglected denominator time constant is divided in half, where half of it is added as extra effective time delay, and half of it is added to the smallest non-neglected time constant. If the half rule were to be applied on the higher order process in eq. (1.26) to approximate it as a first order process with time delay, the approximated process would be

$$g_{approx} = k \frac{1}{\tau_e s + 1} e^{-\theta_e s} \tag{1.29}$$

$$\tau_e = \tau_1 + \frac{\tau_2}{2} \tag{1.30}$$

$$\theta_e = \theta + \frac{\tau_2}{2} + \sum_{j=3}^{N} \tau_j - \sum_{i=1}^{n} T_i \tag{1.31}$$

1.2.4 Decoupling

Mulitvariable systems or multiple inputs multiple outputs (MIMO) systems are system with more than one CV and more than one MV. For these systems it is no longer straight forward which MV should be used to control each CV, as the MVs can affect both CVs. In addition the CVs could be dependent on each other and a change in one will lead to the others changing as well. With a lot of these interactions between the different CVs and MVs the system can be said to be highly coupled. For system which are not highly coupled single loop control loops can be used. For these systems the input-output pairing could be made randomly, however a random selection might lead to unstable behaviour.

A strategy to find the ideal input-output pairing is a strategy known as the *relative-gain analysis*[15] where the realtive gain array (RGA) is constructed. In multivariable processes there will be an open-loop gain from each input to each output. This is the change in the given output variable given a unit change in the given input, given that all the other input does not change. In multivariable processes there is also a closed loop gain from each input to each output. The closed loop gain is the change in the given output given a unit change in the given input, given that all other outputs does not change. So the difference between the open-loop gain and the closed-loop gain is that for the open-loop the other inputs are kept constant while the other states can change, while for the closed loop the opposite is the case, where the other inputs can change, but the other output are kept the same. The relative gains are the ratio between the open-loop and the closed-loop gains. To select the ideal pairing and array containing all the relative gains for the process is constructed, and pairings are made where the relative are close to 1, since this means the open-loop and closed loop-gains are the same.

However for some processes the interactions in the process are too severe so satisfactory control is not achieved with the best pairing. For these processes decoupling can be applied. Decoupling is a strategy where the interactions in multivariable processes are canceled by the use of computing networks. Decoupling can be applied to both the outputs and the inputs of the process, but here only input decoupling will be presented. When decoupling is applied to the inputs the individual controllers will be able to manipulate all the inputs.

Consider the following MIMO process with two inputs and two outputs

$$y_1 = g_{11}u_1 + g_{12}u_2 \tag{1.32}$$
$$y_2 = g_{21}u_1 + g_{22}u_2 \tag{1.33}$$

Decoupling can be achieved by constructing decoupling blocks as in fig. 1.1

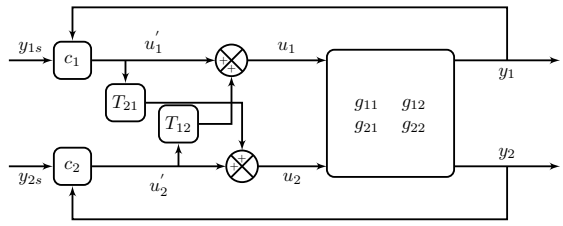

Figure 1.1: Input decoupling by the use of decoupling blocks T_{21} and T_{12}.

In the figure above a decoupled closed loop system is created where the input u_1 and u_2 are decoupled using decoupling blocks T_{21} and T_{12}

$$u_1 = u_1^{'} + T_{12}u_2^{'} \tag{1.34}$$

$$u_2 = T_{21}u_1^{'} + u_2^{'} \tag{1.35}$$

where $u_1^{'}$ and $u_2^{'}$ are the controller outputs. Inserting these into eqs. (1.32) and (1.33) yields

$$y_1 = (g_{11} + T_{21}g_{12})u_1^{'} + (T_{12}g_{11} + g_{12})u_2^{'} \tag{1.36}$$

$$y_2 = (g_{21} + T_{21}g_{22})u_1^{'} + (T_{12}g_{21} + g_{22})u_2^{'} \tag{1.37}$$

The decoupling blocks T_{21} and T_{12} are then designed such that the system becomes a set of SISO systems, this is done by selecting T_{12} and T_{21} as

$$T_{12} = -\frac{g_{12}}{g_{11}} \tag{1.38}$$

$$T_{21} = -\frac{g_{21}}{g_{22}} \tag{1.39}$$

By using these decoupling blocks eqs. (1.36) and (1.37) becomes

$$y_1 = g_{11}(1 + \frac{g_{12}g_{21}}{g_{11}g_{22}})u_1^{'} \tag{1.40}$$

$$y_2 = g_{22}(1 + \frac{g_{12}g_{21}}{g_{11}g_{22}})u_2^{'} \tag{1.41}$$

which are two SISO systems. This form of decoupling does have some problems associated with it though. The first one is that it is not straight forward on how to initialize the controllers, finding the initial biases of the controllers. The second one is that the transfer functions from the controller outputs $u_1^{'}$ and $u_2^{'}$ to the process outputs y_1 and y_2, which is used to tune the controllers are convoluted and complex. Both of these reasons can be overcome but the third problem is more difficult to solve. The third problem is that this structure does not deal well with constrained operations. These problems are all solved by introducing a slightly altered structure, as seen in figure fig. 1.2.

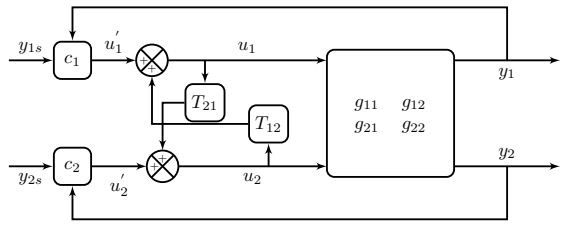

Figure 1.2: Inverted implementation if input decoupling by the use of decoupling blocks T_{21} and T_{12}.

The difference in this formulation is that the process inputs and not the controller outputs are feed back into the decoupling blocks,

$$u_1 = u_1^{'} + T_{12}u_2 \tag{1.42}$$

$$u_2 = T_{21}u_1 + u_2^{'} \tag{1.43}$$

Using the same decoupling blocks as for the other formulation eqs. (1.32) and (1.33) becomes

$$y_1 = g_{11}u_1^{'} \tag{1.44}$$

$$y_2 = g_{22}u_2^{'} \tag{1.45}$$

This structure retains the original transfer functions, making controller tuning an easy task, and control constraints are no longer a problem.

Chapter 2

New proposed methodology

The new MV transformation method was first discussed in [19]. In this new method, non-linear systems are transformed into a linear system by designing transformed MVs, which linearizes the mapping between the transformed input and the output. This is very similar to feedback linearization which achieves a similar goal, however, in this new methodology, there is a new simple methodology for designing transformed MVs which achieves this goal. With this new methodology, the aim is to transform the nonlinear systems into first-order systems, to avoid the control limitations faced when using feedback linearization which transforms the system into a chain of integrators.

2.1 Assumptions

For this methodology, an assumption is that the relative order of the system r must be 1, unlike feedback linearization which can be applied to systems with a higher relative order. However, this assumption can be relaxed if cascade is used as will be elaborated later in the chapter. This methodology can also be applied on square systems meaning the number of inputs must be the same as the number of outputs. It is also assumed all disturbances are measured.

2.2 New methodology

With the new methodology a non-linear process

$$\frac{dx}{dt} = f(x, u, d)$$

is transformed into a linear process

$$\frac{dx}{dt} = Ax + v$$

by introducing a new transformed manipulative variable v

$$v = b(x, u, d)$$

The new transformed system is both linear and decoupled from disturbances, and can be controlled using linear control theory. The block diagram for this methodology is given in fig. 2.1, where the newly introduced transformed MVs v is manipulated by a controller. The physical input vector u is calculated in a calculation block where u is calculated by solving a set of algebraic equations given the output y, disturbances d, and the transformed MVs v. d is the disturbance vector, y is the output vector, y_s is the setpoints for the outputs, and e is the error vector being the difference between the output and its setpoint.

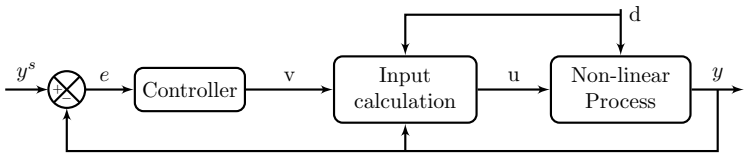

Figure 2.1: Block diagram of the new proposed method

This block diagram can be divided into two different parts. The block diagram in fig. 2.2 representing the transformed linear process decoupled from disturbances, with a controller manipulating v.

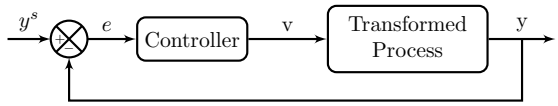

Figure 2.2: Block diagram of transformed system with a control loop.

The second block diagram is shown in fig. 2.3, which is the components of the transformed process, where disturbances are entirely taken care of internally by the calculation block.

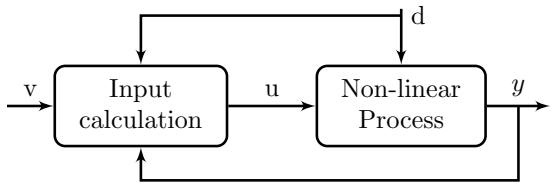

Figure 2.3: Block diagram for the linear, and disturbance decoupled process.

For the block diagrams above it was assumed that the outputs y and the states x were the same, however, this methodology can be extended for a process where this is not the case

and the outputs are a function of the states

$$y = h(x)$$

Assuming $h(x)$ is invertible with respect to y the following block diagram given in fig. 2.4 can be used where the MV transformation is combined with an output transformation as well.

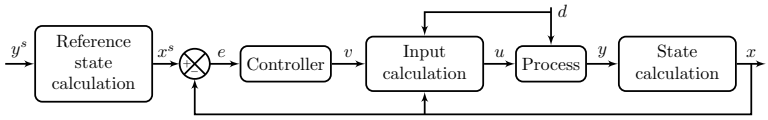

Figure 2.4: Block diagram of the new proposed method with output transformation.

2.3 Manipulated variable transformation

The methods to design MV transformations are dependent on the system in question, and in the following paragraph some strategies and subsequent categories of MV transformations will be discussed and elaborated. These strategies are based on internal communication with Sigurd Skogestad[18]. Before the strategies will be elaborated some notation must be established.

- u := Physical/original input vector
- v := Transformed input vector
- d := Measured disturbance variables vector
- x := State variables of the system, which are split into x_1 which are the control variables, and x_2 which are measured states which affects x_1 but are not control variables.
- y := Output variables, $y = x_1$, requires $n_y = n_u$.

2.3.1 Input transformation for dynamic systems, v_L

The first strategy is one which creates a linear and possibly decoupled system with perfect disturbance rejection, from a non-linear system. Consider the non-linear mathematical model on the form

$$\frac{dy}{dt} = f(y, x_2, u, d) \tag{2.1}$$

This model can be linearised with the following transformed variable v_L

$$v_L = f(y, x_2, u, d) - Ay \tag{2.2}$$

where A is a constant matrix for multivariable system and a scalar constant for scalar systems. Inserting eq. (2.2) into eq. (2.1) yields the linear system

$$\frac{dy}{dt} = Ay + v_L \tag{2.3}$$

which can be controlled using linear control theory. The tuning parameter A can be chosen arbitrary, but there are certain proposed strategies which might be preferred.

- The first strategy is to select A such that it is the Jacobian of the model function f at the nominal operating point, $A = df/dy|_*$. The advantage of choosing this parameter is, that the transformed MV v_L becomes independent of the output y locally around the nominal operating point. The selection of this parameter will only result in initial decoupling.

- The second strategy for selecting A, which will result in perfect decoupling, is to select a diagonal matrix. This matrix could be the diagonal matrix of $df/dy|_*$, which while not giving local independence from y, it will retain the local dynamics of the non-linear system, which might give better performance than an arbitrarily selected diagonal matrix. It should be noted that for eq. (2.3) to be a self-regulating system, A must be selected such that the sign of all the diagonal elements are less than zero, $A < 0$.

- The third strategy is to select A to be zero, $A = 0$. This leads to the special case similar to feedback linearisation, where the linearised system in eq. (2.3) becomes

  $$\frac{dy}{dt} = v_{fL}$$

  and

  $$v_{fL} = f(y, x_2, u, d)$$

  While this is arguably the simplest transformation, it transforms the system into an integrating one, which is undesirable when designing a feedback controller.

An alternate formulation of this linear transformation dubbed v_{L0} is to multiply the both the linear system and v_L with the parameter $T = -A^{-1}$. This results in the linear system

$$T\frac{dy}{dt} = -y + v_{L0} \tag{2.4}$$

and

$$v_L = Tf(y, x_2, u, d) + y \tag{2.5}$$

For scalar systems the transformations v_L and v_{L0} will result in identical behaviour, only different gains.

For a multivariable system on the other hand it is a different story. Both v_L and v_{L0} linearise the system and introduces perfect disturbance rejection. However, the interaction between the inputs and outputs is different as the two different formulations give different coupling, given that A is selected such that it is not a diagonal or zero matrix. v_{L0} gives static decoupling, which adds a degree of predictability to the system, as the system could be controlled by directly manipulating v_{L0}. However, since the system will most likely be controlled by an automatic controller this is not necessarily a large advantage. v_L might give less dynamic coupling than v_{L0}, meaning it might be better coupled with a controller

compared to v_{L0}.

As previously stated this methodology can only be applied on systems with a relative order of 1. This means for multivariable systems where it is only of interest to control some outputs y, and not the internal states x_2, the dynamic functions for the outputs must be explicitly a function of the input u. For systems where the dynamic equations for the states x_2 are explicitly a function of the input, but dynamic equations for the output are not this assumption is not fulfilled, however as will be discussed later this can be bypassed with the use of a cascade structure.

It is assumed that the MV transformation $v = f_L(y, x_2, u, d)$ is invertible with respect to u for a neighbourhood around the nominal operating point, such that the physical input u can be calculated as $u = f^{-1}(v, y, x_2, d)$.

2.3.2 Input transformation for static systems, v_0

The second strategy to design transformed MVs which introduce perfect feedforward control is for static systems on the form

$$\frac{dy}{dt} = f(y, x_2, u, d) = 0 \tag{2.6}$$

If the model in eq. (2.6) can be reformulated on the form

$$y = r(x_2, u, d) \tag{2.7}$$

a transformed MV v_0 can be defined as the left hand side of eq. (2.6), with

$$v_0 = r(x_2, u, d) \tag{2.8}$$

This MV gives perfect disturbance rejection and perfect decoupling for the static system

$$y = v_0 \tag{2.9}$$

The subscript indicates that transformation is independent of the output y, and that the transformation is static. For this strategy to be feasible the number of independent static equations describing the system must be equal to the number of outputs.

While this transformation is intended for static systems it can also be applied to dynamic systems for different types of reason. This reason might be that the transformation is easier to realize, or that only a static model of the system in question is available to design the transformed MV. At steady-state, the transformed system obtained from this transformation will be decoupled, linear, and independent of disturbances, but dynamically this might not be the case.

Considering the special case of a non-linear dynamic system on the form

$$\frac{dy}{dt} = A(x_2, u, d)y + b(x_2, u, d) \tag{2.10}$$

This model is non-linear with respect to x_2, u and d, but linear with respect to y. To apply the static MV transformation, the model is solved for y at steady state yielding the transformed MV

$$v_0 = T(x_2, u, d)b(x_2, u, d) \tag{2.11}$$

where $T = -A^{-1}$, inserting this into eq. (2.10) yields the transformed system

$$T(x_2, u, d)\frac{dy}{dt} = -y + v_0 \tag{2.12}$$

This transformed system gives good dynamic disturbance rejection, this especially true at steady-state where it will give perfect disturbance rejection. For process where the goal is to control the system to remain at steady-state this transformation should work excellent.

2.3.3 Non-linear and general transformation for dynamic systems, v_N and v_G

The two main strategies are the linear transformation and the steady-state transformation, but two more transformation will be briefly mentioned. These two are the non-linear transformation v_N and the general transformation v_G. The non-linear transformation can be applied to systems on the form,

$$\frac{dy}{dt} = a(y) + b(y, x_2, u, d) \tag{2.13}$$

where $a(y)$ is some non-linear function of the output y. Selecting $b(y, x_2, u, d)$ to be the MV transformation

$$v_N = b(y, x_2, u, d) \tag{2.14}$$

the system can be transformed to a non-linear system with perfect disturbance rejection and initial decoupling.

$$\frac{dy}{dt} = a(y) + v_N \tag{2.15}$$

The general transformation can be applied on systems on the form

$$\frac{dy}{dt} = a(y, x_2, d, u) + b(y, x_2, u, d) \tag{2.16}$$

where similarly the transformed MV is selected as

$$v_G = b(y, x_2, u, d) \tag{2.17}$$

This transformation does not have any general properties and it might have limited use.

2.4 Input calculation

In this new method, the physical input u is in some form calculated from the transformed input v, the measured states x, and the measured disturbances d. This can be done in several ways. One approach is to use a calculation block. This a mathematical block that

back-calculates u from the model equation of the transformed model. This can be done analytically which requires $v = b(x, u, d)$ to be invertible with respect to u, or u can be solved numerically from $v = b(x, u, d)$.

2.5 Cascade control

There are three cases where a cascade structure is desirable to use with this new methodology. The first case is to be able to apply the new methodology on systems with relative order higher than one. For the linear case, it is assumed the relative order of the system is one, such that the transformed input v_L is explicitly a function of the physical input u. For systems where this is not the case, an MV transformation might still be realized in combination with cascade control. So for a system on the form

$$\frac{dy}{dt} = f_y(y, x_2, d) \tag{2.18}$$

$$\frac{dx_2}{dt} = f_{x_2}(x_2, u, d) \tag{2.19}$$

No MV transformation can be can constructed to control y based on f_y which is a function of the physical input u. However given that x_2 is assumed measured the setpoint of x_2 can be used as an input in an outer loop, and an MV can be constructed which is a function of x_2 instead

$$v_L = f_y(y, x_2, d) - Ay \tag{2.20}$$

This MV can be used in cascade structure in two ways,

1. The first is to use a calculation block which calculates the setpoint to x_2 from the inverse of $v_L = f_y(y, x_2, d) - Ay$ and an inner loop changes u to match this setpoint.

2. The second is to directly use v_L as the setpoint and have a calculation block which calculates the v_L of the process and has an inner loop manipulate the input u to make the v_L of the process match its setpoint.

With a sufficiently fast slave loop, the performance should be similar to the case without cascade. This cascade structure can also be applied to a static system where the methodology should be similar.

The second case to use cascade with this new methodology is the case where the model equation can be simplified due to extra measurements. If flow measurements are available, the flow can be used as the input instead of the valve position when designing transformed MVs.

The third case is to use it as an alternative to using a calculation block to calculate the input u, as seen in fig. 2.5. In this variant, the physical input u is not calculated by a calculation block but instead by a slave control loop, which tries to control v calculated from a calculation block to match v_s given by the outer loop. If the inner loop is tuned

tightly it should give similar control as the calculation block. This implementation have the advantage that $v = b(x, u, d)$ does not need to be invertible with respect to u.

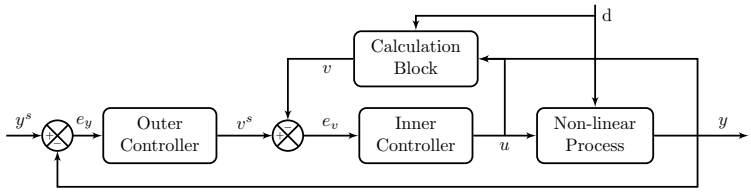

Figure 2.5: Block diagram for the new proposed methodology with cascade.

Chapter 3

Case study 1: Extraction process

3.1 Model description

In this case a liquid-liquid extraction process is to be evaluated as seen in figure 3.1. This example is from [9]. In this process, acid is transferred from the water/acid feed stream (F) to the extract stream (E), with the use of a solvent (S). The extraction takes place in a tank consisting of an organic layer (ME) and a water layer (MR). The excess water is removed in the raffinate stream (R). It is assumed that the acid is completely extracted and that the raffinate stream is pure water. In the organic layer, it assumed that the acid will bring some water with it in perfect equilibrium. The equilibrium can be described with an equilibrium constant (K) of 1/3.

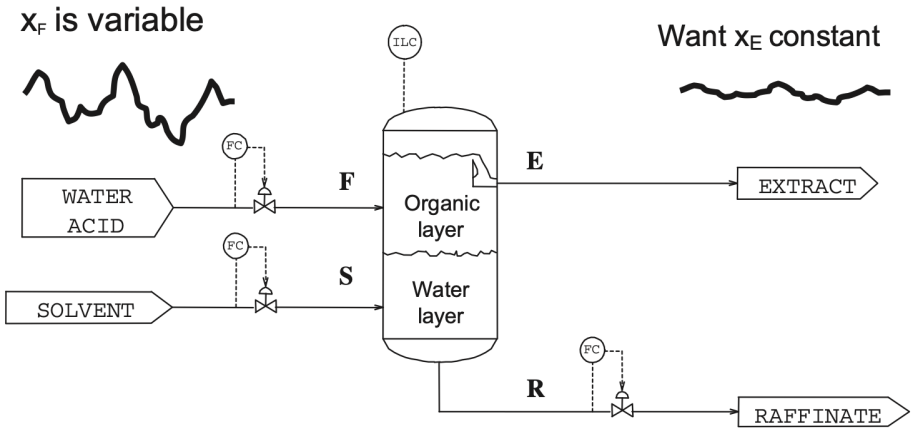

Figure 3.1: System description of liquid-liquid extraction process, from [9].

Of the four streams in this process the water acid feed stream F is used as the MV, and it is assumed the feed can be directly manipulated.

$$u = F$$

The four states in this process are the molar fraction of acid in the organic layer x_{AE}, the molar fraction of water in the organic layer x_{WE}, the holdup of extract M_E and the holdup of raffinate M_R

$$\begin{aligned} x_1 &= M_E \\ x_2 &= M_R \\ x_3 &= x_{AE} \\ x_4 &= x_{WE} \end{aligned}$$

The molar fraction of acid in the organic layer is the CV in this case study,

$$y = x_{AE}$$

The main disturbances considered in this case study is the mass fraction of acid in the acid water feed stream x_{AF}, and the solvent feed stream S

$$\begin{aligned} d_1 &= x_{AF} \\ d_2 &= S \end{aligned}$$

The solvent stream is the throughput of process, and change in throughput is considered a disturbance.

The residence time in the extraction phase of the process $\tau_r = M_E/E$

Table 3.1: Nominal operating values of extraction process

Variable	Value	Unit
M_E^*	10000	mol
M_R^*	10000	mol
x_{AE}^*	0.214	-
x_{WE}^*	0.071	-
F^*	100	$\mathrm{mol\,s^{-1}}$
S^*	100	$\mathrm{mol\,s^{-1}}$
E^*	140	$\mathrm{mol\,s^{-1}}$
R^*	60	$\mathrm{mol\,s^{-1}}$
x_{AF}^*	0.3	-
K^*	0.333	-
τ_r^*	71.429	s

Component and total mass balances, alongside the equilibrium equation are used to describe the system and forms the model equations. First the acid component balance is

$$\frac{d(x_{AE}M_E)}{dt} = x_{AF}F - x_{AE}E \tag{3.1}$$

The water component balance is

$$\frac{d(x_{WE}M_E + M_R)}{dt} = (1 - x_{AF})F - x_{WE}E - R \tag{3.2}$$

The total mass balance is

$$\frac{d(M_E + M_R)}{dt} = F + S - E - R \tag{3.3}$$

The acid water equilibrium in the organic layer is

$$\frac{x_{WE}}{x_{AE}} = K \tag{3.4}$$

The total inventory (M_E + M_R) is controlled by the overflow of E. To control the water level (M_R) R is used. P-control is used for simplicity. The extract and raffinate flows are then described by the following equations

$$E = E^* + K_{c1}(\Delta M_E + \Delta M_R) \tag{3.5}$$

$$R = R^* + K_{c2}\Delta M_R \tag{3.6}$$

where ΔM_E and ΔM_R are the deviation from the setpoint for M_E and M_R,

$$\Delta M_E = M_{Es} - M_E \tag{3.7}$$

$$\Delta M_R = M_{Rs} - M_R \tag{3.8}$$

where M_{Es} and M_{Rs} are the setpoints for M_E and M_r, E^* and R^* are the steady state biases for E and R, and K_{c1} and K_{c2} are the controller gains for the two P-controllers.

For the sake of simplifying the model, it is assumed that the mass inventories are constant. This is achieved by selecting the controller gains K_{c1} and K_{c2} sufficiently high. In this case study the controller gains are selected to be

$$K_{c1} = -1\mathrm{s}^{-1} \tag{3.9}$$

$$K_{c2} = -1\mathrm{s}^{-1} \tag{3.10}$$

With this assumption and using the equilibrium property given in eq. (3.4), the model can be simplified to

$$\frac{dx_{AE}}{dt} = \frac{1}{M_E}(x_{AF}F - x_{AE}E) \tag{3.11}$$

$$\frac{dx_{AE}}{dt} = \frac{1}{KM_E}((1 - x_{AF})F - Kx_{AE}E - R) \tag{3.12}$$

$$0 = F + S - E - R \tag{3.13}$$

An interesting property of this model is that two dynamic equations must be fulfilled simultaneously, as both eqs. (3.11) and (3.12) describe the change the output $y = x_{AE}$ with respect to time. This reduces the degrees of freedom of the system.

3.2 MV transformations

3.2.1 Disturbance rejection

In [9] the objective was to design a control structure that operated the process optimally. The economic goal was to reduce the variations in the acid concentration of the extract stream, when subject to varying acid feed concentration and changing throughput S. The easiest solution is to measure x_{AE} and use feedback control on the water acid feed stream F. However if measurements of the concentration are missing or sufficiently rare, this is not possible. Feedforward control could also be applied however this requires measurements of x_{AF}, and if measurements of x_{AE} are not available for control, it is reasonable to assume measurements of x_{AF} are unavailable as well, making feedforward control infeasible as well. An alternative solution is to find some other control objective that fulfills the economic objective. In [9] *self optimising control* was applied to design a new control variable based on available measurements.

Designing transformed MVs can be thought of as designing a new control objective for the physical input. Instead of having a feedback controller manipulating the physical input u to control the output y, the physical input u is manipulated either by a calculation block or an inner feedback loop controlling the input transform v. If the goal is to control the process at steady state an outer loop which manipulates v to control the output y is then not necessary. The MV transformation can then be used as the alternative control objective. If the goal is to control the process at steady state, the steady state methodology from section 2.3.2, can be applied to find an appropriate MV transform. The simplified model in eqs. (3.11) to (3.13) is simplified further to obtain the steady state model

$$0 = x_{AF}F - x_{AE}E \tag{3.14}$$

$$0 = (1 - x_{AF})F - Kx_{AE}E - R \tag{3.15}$$

$$0 = F + S - E - R \tag{3.16}$$

By solving eq. (3.14) with respect to the state x_{AE}, the static MV transformation can be found to be,

$$v_{01} = \frac{x_{AF}F}{E} \tag{3.17}$$

This transformation allows for control of x_{AE} if measurements are slow or not available. However this transformation requires measurements of x_{AF}, and as previously discussed if measurements of x_{AE} are unavailable it is likely x_{AF} is not measured either. Using eqs. (3.14) and (3.15) an expression for $x_{AF}F$ independent of x_{AE} can be found which can be inserted into eq. (3.17) to create a new MV transformation independent x_{AF}. Inserting eq. (3.14) into eq. (3.15), the following equation is obtained

$$0 = (1 - x_{AF})F - Kx_{AF}F - R \tag{3.18}$$

which when solved for $x_{AF}F$ yields the following expression

$$x_{AF}F = \frac{F - R}{1 + K} \tag{3.19}$$

which can be inserted into eq. (3.17) to obtain

$$v_{02} = \frac{1}{1+K}\frac{F-R}{E} \tag{3.20}$$

If measurements of the extract stream E happens to not be available but measurements of the solvent stream S are eq. (3.13) can be rewritten as

$$E = F + S - R \tag{3.21}$$

and inserted into eq. (3.20) to obtain

$$v_{03} = \frac{1}{1+K}\frac{F-R}{F+S-R} \tag{3.22}$$

which independent of both x_{AF} and E. Since $1/(1+K)$ is a constant it can be dropped from both v_{02} and v_{03}, however if it remains the setpoint of v_{02} and v_{03} will be the same as the desired concentration of x_{AE}. If $1/(1+K)$ is omited v_{03} can be reformulated as

$$v_{03}^{'} = \frac{1}{1+\frac{S}{F-R}} \tag{3.23}$$

controlling v_{03} to be a constant value is equivalent to keeping $(F-R)/S$ to a constant value, which was one of the proposed self-optimising control strategies proposed in [9].

3.2.2 Setpoint change

If measurements of x_{AE} are available, an outer loop manipulating v could be added such that the setpoint of x_{AE} can be changed. The steady state MV v_0 might not be the best selection for this purpose, and alternative MV transformations can be constructed. The original dynamics of the non-linear system are not retained with the steady state transformation. By modifying the transformation, the original dynamics can be retained. This modification can be found by evaluating the model eq. (3.11), which is on the form

$$\frac{dy}{dt} = a(x,d)y + b(x_2,u,d) \tag{3.24}$$

where

$$a(x,u,d) = -\frac{E}{M_E} \tag{3.25}$$

and

$$b(x,u,d) = \frac{x_{AF}F}{M_E} \tag{3.26}$$

By selecting $v_G = b(x,u,d)$ as the transformed MV, a transformed system which retains the original dynamics is obtained. The drawback is however that it does not linearize the system, alongside introducing bad feedforward action. A better alternative is to select the linearized MV transformation described in section 2.3.1, which will by selecting the

tuning parameter $A = df_1/dy|_*$, retain the nominal steady state dynamics of the system. With

$$\left.\frac{df_1}{dx_{AE}}\right|_* = -\frac{E^*}{M_E^*} \tag{3.27}$$

the linear MV transformation is

$$v_{L1} = \frac{x_{AF}F}{M_E} - (\frac{E}{M_E} - \frac{E^*}{M_E^*})x_{AE} \tag{3.28}$$

A simpler linear transformation is the feedback linearization variant of the linear transformation described in section 2.3.1 If the simplified model in eqs. (3.11) to (3.13) is used, the transformed MV v_{FL0} is

$$v_{FL1} = \frac{x_{AF}F - x_{AE}E}{M_E} \tag{3.29}$$

For the steady state case it was possible reformulate the MV transformations, given missing measurements of disturbance x_{AF} and of missing measurements of E, and still retain perfect disturbance rejection. This is possible for the dynamic case as well. eqs. (3.11) and (3.12) both describes the change in the output with respect to time, so by putting the right hand side of both equations the following expression is gotten

$$Kx_{AF}F - Kx_{AE}E = (1 - x_{AF})F - Kx_{AE}E - R \tag{3.30}$$

If this equation is solved for $x_{AF}F$, eq. (3.19) appears. This means v_{L1} and v_{FL1} can both be reformulated such that they are independent of the x_{AF}, yielding,

$$v_{L2} = \frac{1}{1+K}\frac{F-R}{M_E} - (\frac{E}{M_E} - \frac{E^*}{M_E^*})x_{AE} \tag{3.31}$$

and

$$v_{FL2} = \frac{1}{M_E}(\frac{F-R}{1+K} - x_{AE}E) \tag{3.32}$$

Both can be made independent of E as well which yields the following transformations

$$v_{L3} = \frac{1}{1+K}\frac{F-R}{M_E} - (\frac{F+S-R}{M_E} - \frac{E^*}{M_E^*})x_{AE} \tag{3.33}$$

and

$$v_{FL3} = \frac{1}{M_E}(\frac{F-R}{1+K} - x_{AE}(F+S-R)) \tag{3.34}$$

3.2.3 Analysis of proposed transformed MVs

Primarily three methods of designing transformed variables have been applied. The three types of MV transformations which were designed are the static transformation v_0, the linear transformation v_L and the special case of the linear transformation that was dubbed feedback linearization v_{FL} due to it transforming the nonlinear process into an integrating

process, which were for all three in three forms depending on the available measurements. The transformed system obtained with v_0 is

$$\frac{dx_{AE}}{dt} = \frac{E}{M_E}(-x_{AE} + v_0) \tag{3.35}$$

This transformation fails at linearizing the system, but it introduces perfect disturbance rejection at steady state. The system obtained with v_L is

$$\frac{dx_{AE}}{dt} = -\frac{E^*}{M_E^*}x_{AE} + v_L \tag{3.36}$$

This transformation successfully linearizes the non-linear system. The system obtained with v_{FL} is

$$\frac{dx_{AE}}{dt} = v_{FL} \tag{3.37}$$

which is an integrating system. The steady-state transform is not expected to perform that well for systems where the setpoint is changed. The transformed system in eq. (3.35) is non-linear, with the the dynamic input gain being dependent on the resident time

$$\tau_r = E/M_E$$

which changes with respect to E and M_E. While M_E is assumed to be constant, E is indirectly dependent on the input F, meaning the dynamic gain input gain is dependent on the input. This can lead to potentially bad control if the input is manipulated to aggressively. However, if the input is cautiously manipulated this might not be a problem. So the steady-state transform will probably perform best for large closed-loop time constants, and worst for small closed-loop time constants. So for this transformation to work well when doing setpoint changes the process must be a lot faster than the controller.

The linear transformation is expected to perform the best overall, for all closed loop time constants, as this transformation successfully transforms the non-linear system into a linear one. However, the performance is dependent on the selection of the tuning parameter A. For SISO-systems this is the same as selecting the open-loop time constant of the linearized system. In this case, A was selected such that the open-loop time constant is the same as the open-loop time constant for the nonlinear system at the nominal steady-state, and should probably overall result in a good performance. The feedback linearization special case is the special case where A is selected to be zero. While this transformation transforms the non-linear model into a linear one, it is an integrating system. Integrating systems are difficult to control, especially with PI-controllers, so the performance is expected to be worse than the linear transformation where A is non zero, but this transformation might outperform the steady-state transform for a fast closed-loop time constant.

Effect of simplified model

It is assumed all these three methods will bring perfect disturbance rejection, however since they are all design with the simplified model given eqs. (3.11) to (3.13) where the

inventories M_E and M_R are assumed to be perfectly controlled, perfect disturbance rejection is probably not achievable. While the control of the inventories is fast they are not instant, as well as being controlled with P-controllers. This means that there will be steady-state offsets in the control of the inventories when the operations change, and M_E and M_R will change slightly. The changing inventory and the noninstantaneous change in the inlet and outlet flow of the extractor will affect the disturbance rejection capabilities of the different proposed control structures.

Different variations of the MV transformations were proposed based on the available measurements. These were derived using the assumption of perfectly controlled inventories. With perfect inventory control, it would be reasonable to expect the different variations to behave identically, however since this is not the case the different cases will probably behave differently.

3.3 Case studies

The goal of this case study is to first test the disturbance rejection capabilities of the different MV transformations without an outer loop. The second goal is to add an outer control loop and test which structures perform best with different closed-loop constants.

3.3.1 Open-loop disturbance rejection

The steady state transformation was designed for open-loop disturbance rejection, and shall be tested for for disturbances in x_{AF} and S. Three variations of v_0 were proposed, v_{01}, which requires measurements of x_{AF} and E. If Measurements of x_{AF} are not available v_{02} could be used instead which requires measurements of E and R. If measurements of E are not available v_{03} could be used which requires measurements of R and S. When doing disturbance step tests in this case study the input F was manipulated using a calculation block, which calculates F from the inverse of eqs. (3.17), (3.20) and (3.22), which are

$$F = \frac{v_{01}E}{x_{AF}} \tag{3.38}$$

$$F = (1+K)v_{02}E + R \tag{3.39}$$

$$F = \frac{(1+K)v_{03}S}{1-(1+K)v_{03}} + R \tag{3.40}$$

Integrated absolute error (IAE) is a performance measurement that can be used to evaluate the performance of a control structure, which will be used as the performance measurement for all the case studies in this thesis. The lower the IAE the better the performance. The IAE is calculated using the following expression

$$IAE = \int_0^t |(y_s(\tau) - y(\tau))|d\tau \tag{3.41}$$

Where y_s is the setpoint of the output y.

The performance of these three variations of the steady state performance for disturbances in x_{AF} and S are given table 3.2 bellow.

Table 3.2: Open loop disturbance step test for disturbances in x_{AF} and S, with transformed MVs

DV	Δd	v_{01}	v_{02}	v_{03}
x_{AF}	0.03 [-]	0.0138	0.0346	0.0346
x_{AF}	- 0.03 [-]	0.0169	0.0422	0.0422
S	10 $[\mathrm{mol\,s^{-1}}]$	0.0111	0.0402	0.0208
S	- 10 $[\mathrm{mol\,s^{-1}}]$	0.0135	0.0487	0.0252

The step test responses are plotted in figs. 3.2 to 3.5 where it can be seen that the disturbances are rejected when v_0 are kept constant, with all three variations. From table 3.2 v_{01} performs the best for disturbances in both DVs. v_{02} and v_{03} performs identically for disturbances in x_{AF}, which is a disturbance which is not measured for any of them. For disturbances in S v_{03} which requires measurements of S performs better than v_{02} which does not use measurements of S.

With a perfect model, it could be expected that the disturbances would be perfectly rejected. However this is not the case as previously discussed in section 3.2.3, as it was assumed the inventories were perfectly controlled when designing the transformed MVs. This model simplification when deriving the control structure results in a model mismatch, in which the feedforward action rejected without feedback. With the extra model error introduced with the v_{02} and v_{03}, the feedforward action was slightly worsened but still successful.

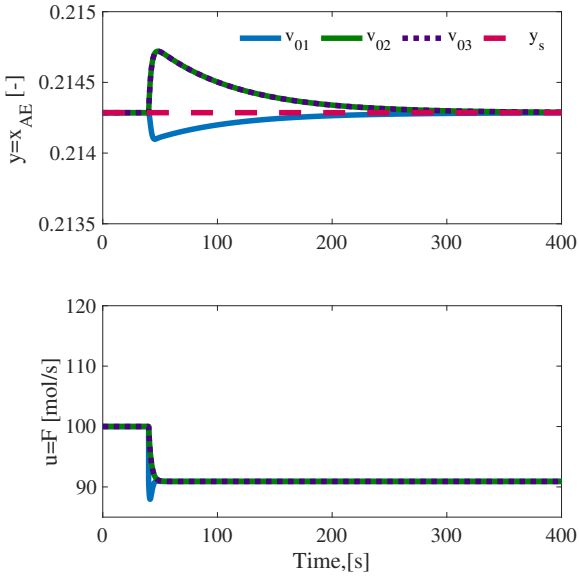

Figure 3.2: Step test in DV x_{AF} of 0.03 [-] at time = 40 s for v_{01}, v_{02} and v_{03}. y_s is the setpoint of x_{AE}.

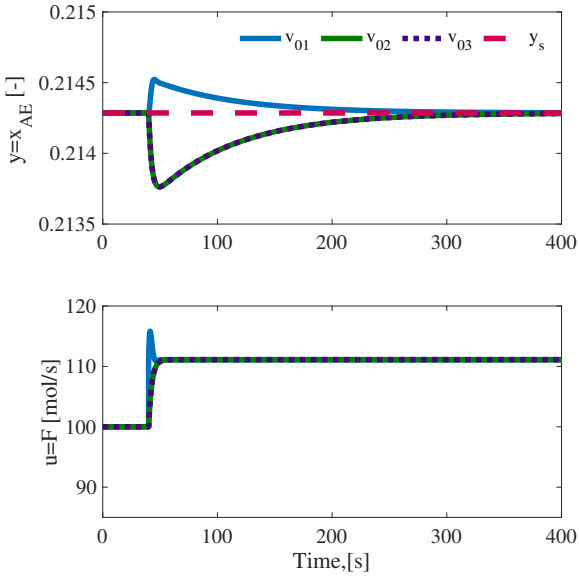

Figure 3.3: Step test in DV x_{AF} of -0.03 [-] at time = 40 s for v_{01}, v_{02} and v_{03}. y_s is the setpoint of x_{AE}.

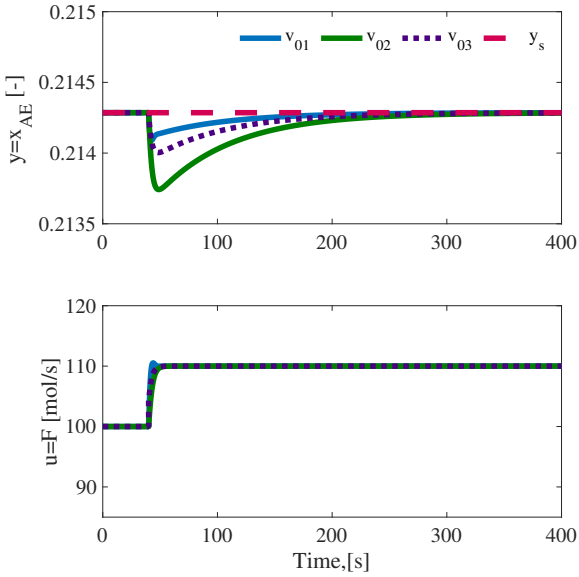

Figure 3.4: Step test in DV S of 10 $\mathrm{mol\,s^{-1}}$ at time = 40 s for v_{01}, v_{02} and v_{03}. y_s is the setpoint of x_{AE}.

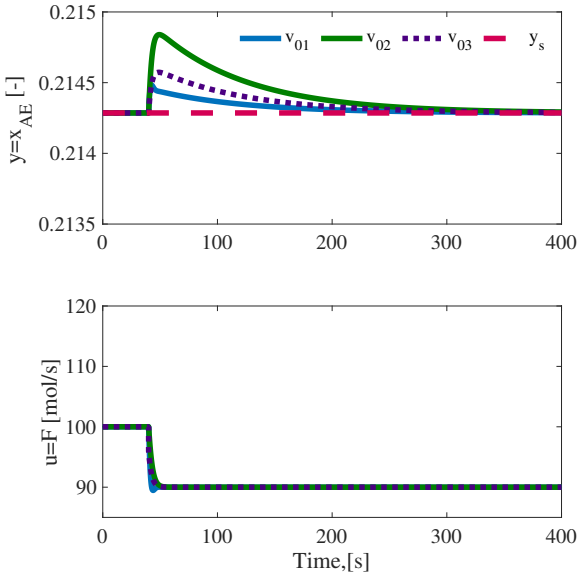

Figure 3.5: Step test in DV S of -10 $\mathrm{mol\,s^{-1}}$ at time = 40 s for v_{01}, v_{02} and v_{03}. y_s is the setpoint of x_{AE}.

3.3.2 Closed-loop setpoint changes

It was in the previous section showed that MV transformations can be used to design a control system which rejects disturbances without feedback, for differing sets of measurements, even with a simplified model. But more control structures were proposed if changing the setpoint is desired. An outer loop is introduced where a PI-controller manipulates v. The different proposed MVs will all be tested for three different closed-loop constants τ_c, and find out for which tunings the different transformed systems performs the best. Three different variations of the transformed MVs were proposed for the three transformed MVs, depending on the available set of measurements. They will be compared with each other to test if they perform similarly.

PI-controller tuning

The PI-controllers will be tuned using the SIMC-rules[17], using the transformed models given in eqs. (3.35) to (3.37). The transformed model achieved using the static transformation v_0 is not linear, and is linearized around the nominal operating point to achieve a linear model, which can be used to derive controller tunings. The linearized systems in terms of deviation variables are

$$\frac{\Delta x_{AF}}{dt} = \frac{1}{\tau_r^*}(-\Delta x_{AF} + \Delta v_0) \tag{3.42}$$

$$\frac{\Delta x_{AF}}{dt} = -\frac{1}{\tau_r^*}\Delta x_{AF} + \Delta v_L \tag{3.43}$$

$$\frac{\Delta x_{AF}}{dt} = \Delta v_{FL} \tag{3.44}$$

where the deviation variables are

- $\Delta x_{AF} = x_{AF} - x_{AF}^*$
- $\Delta v_i = v_i - v_i^*, \quad \forall i \in \{0, L, FL\}$

The linearized systems are transformed into the frequency domain with a Laplace transformation to obtain the transfer functions from v to x_{AF}. Dropping the deviation notation the transfer functions are,

$$\frac{x_{AF}}{v_0}(s) = g_0 = \frac{1}{\tau_r^* s + 1} \tag{3.45}$$

$$\frac{x_{AF}}{v_L}(s) = g_L = \frac{\tau_r^*}{\tau_r^* s + 1} \tag{3.46}$$

$$\frac{x_{AF}}{v_{FL}}(s) = g_{FL} = \frac{1}{s} \tag{3.47}$$

Using the SIMC rules the controller gain K_c and integral time τ_I can be found for the three different control structures. There is assumed there is no time delay when deriving the controller tunings. The controller tunings are given in table 3.3 bellow.

Table 3.3: Controller tunings for PI-controller in extraction process

MV	K_c	Unit	τ_I	Unit
v_0	$\frac{\tau_r^*}{\tau_c}$	-	τ_r^*	s
v_L	$\frac{1}{\tau_c}$	s^{-1}	τ_r^*	s
v_{FL}	$\frac{1}{\tau_c}$	s^{-1}	$4\tau_c$	s

In the SIMC-rules the integral time τ_I is selected as the smallest of τ and $4(\tau_c + \theta)$. However when selecting τ_I for v_0 and v_L this was not considered as τ_I was selected to be τ_r^*. For v_0 this was done due to the assumption that this control structure will perform better when the closed loop time constant τ_c is greater than the open loop time constant τ_r. Similarly it is assumed v_L will perform better when the closed loop time constant is greater than $\tau_r/4$. Despite this all the control structures will be tested for the closed loop time constant being

- τ_c = 10 s
- τ_c = 100 s
- τ_c = 1000 s

The selection of these closed loop time constants is to test the different structures to a time constant much lower than the nominal dynamics, $\tau_c = 10\text{s} << \tau_r^* = 71.429\text{s}$, a time constant which is similar to the nominal dynamics, $\tau_c = 100\text{s}$ and a time constant which is much greater than the nominal dynamics, $\tau_c = 1000\text{s} >> \tau_r^* = 71.429\text{s}$.

To get the physical input F from the transformed input v calculation blocks are used, just as was the case for the open loop disturbance rejection case. The inverse functions for v_{L1}, v_{L2}, v_{L3}, v_{FL1}, v_{FL2} and v_{FL3} are

$$F = (\frac{E}{M_E} - \frac{E^*}{M_E^*})\frac{M_E x_{AE}}{x_{AF}} + \frac{v_{L1} M_E}{x_{AF}} \tag{3.48}$$

$$F = (1+K)M_E((\frac{E}{M_E} - \frac{E^*}{M_E^*})x_{AE} + v_{L2}) + R \tag{3.49}$$

$$F = \frac{(1+K)M_E(v_{L3} - \frac{E^*}{M_E^*}x_{AE}) + R(1-(1+K)x_{AE}) + (1+K)x_{AE}S}{1-(1+K)x_{AE}} \tag{3.50}$$

$$F = \frac{M_E v_{FL1} + x_{AE}E}{x_{AF}} \tag{3.51}$$

$$F = (1+K)(M_E v_{FL2} + x_{AE}E) + R \tag{3.52}$$

$$F = \frac{(1+K)M_E v_{FL3} + R(1-(1+K)x_{AE}) + (1+K)x_{AE}S}{1-(1+K)x_{AE}} \tag{3.53}$$

Non of these inverse function appears to have problems with approaching singularity for certain values of v, however the inverse function of v_{03} given in eq. (3.40) approaches singularity as v approaches $(1/(1+K) = 3/4$. This can cause problems for this control structure when the outer loop is aggressively tuned.

Setpoint changes with measurements of x_{AF} and E

The process will be subject to large setpoint changes, both positive and negative. First, the transformations where x_{AF} and E are both measured will be considered. The performances are given in table 3.4.

Table 3.4: Setpoint changes for v_{01}, v_{L1} and v_{FL1}

IAE		
	New setpoint	
$\tau_c = 10$	0.5 [-]	0.1 [-]
v_{01}	94.5371*	2.5260
v_{L1}	2.8664	1.1426
v_{FL1}	4.1055	1.6652
$\tau_c = 100$		
v_{01}	28.5725	11.5072
v_{L1}	28.5721	11.4288
v_{FL1}	41.6623	16.7649
$\tau_c = 1000$		
v_{01}	285.7216	114.2885
v_{L1}	285.7214	114.2886
v_{FL1}	416.8355	167.7750

When tuned with a fast closed loop time constant v_L performed the best, with v_0 performing the worst. This is the expected result, as v_0 was designed with a static model. When doing a positive setpoint change to 0.5, the process went unstable with v_0. v_{FL} performed worse than v_L as expected. The setpoint changes for $\tau_c = 10$ can be seen in figs. 3.6 and 3.7.

*Unstable

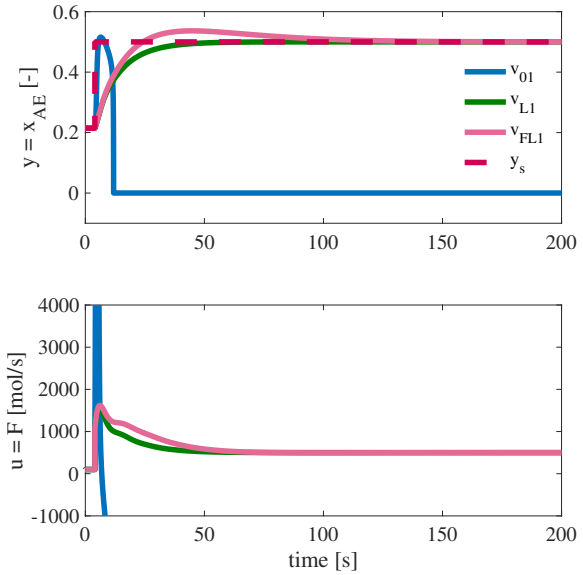

Figure 3.6: Closed loop setpoint change from x_{AF} = 0.214 to 0.5 [-] at time = 4s, for v_{01}, v_{L1} and v_{FL1}, with close loop time constant τ_c = 10s. No physical limitations are put on the physical inputs u in the simulation. While not visible the physical input u goes towards negative infinity as time increase for v_{01}.

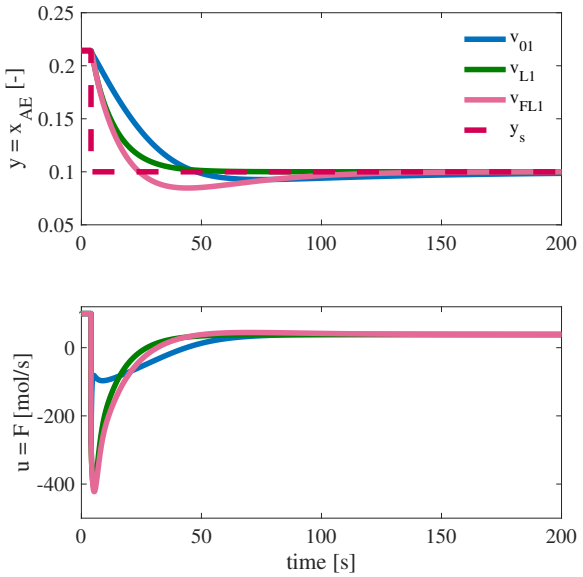

Figure 3.7: Closed loop setpoint change from x_{AF} = 0.214 to 0.1 [-] at time = 4s, for v_{01}, v_{L1} and v_{FL1}, with close loop time constant τ_c = 10s. No physical limitations are put on the physical inputs u in the simulation.

For a closed-loop time constant similar to the nominal open-loop time constant v_0 and v_L performed almost identically, with v_L performing slightly better for the negative setpoint change. This is interesting as v_0 was predicted to perform best for large closed-loop constants and not as well for smaller closed-loop time constants. As expected $v_F L$ performed worse than v_L. The setpoint changes for $\tau_c = 100$ can be seen in figs. 3.8 and 3.9.

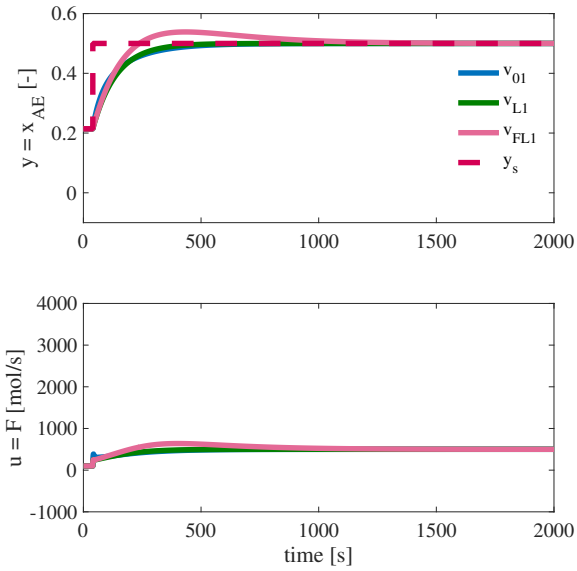

Figure 3.8: Closed loop setpoint change from x_{AF} = 0.214 to 0.5 [-] at time = 40s, for v_{01}, v_{L1} and v_{FL1}, with close loop time constant τ_c = 100s. No physical limitations are put on the physical inputs u in the simulation.

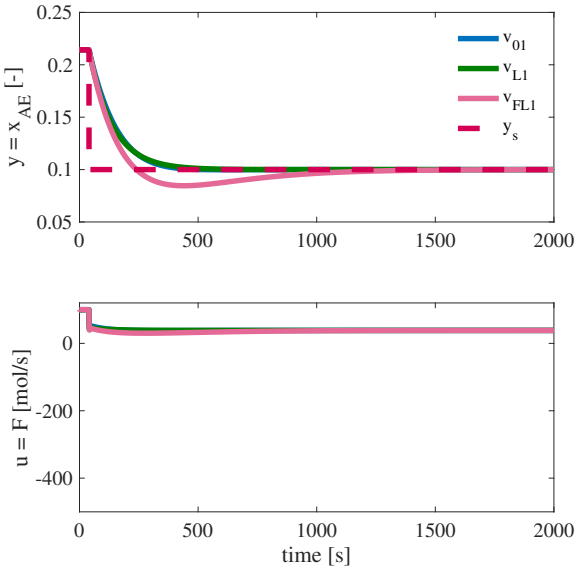

Figure 3.9: Closed loop setpoint change from x_{AF} = 0.214 to 0.1 [-] at time = 40s, for v_{01}, v_{L1} and v_{FL1}, with close loop time constant τ_c = 100s. No physical limitations are put on the physical inputs u in the simulation.

For large closed loop time constants v_0 and v_L performs practically identically, so for slow outer loops the steady state transform performs as well as the linear transform. Just as for the smaller closed loop constants $v_F L$ performs the worst. The setpoint changes for $\tau_c = 1000$ can be seen in figs. 3.10 and 3.11.

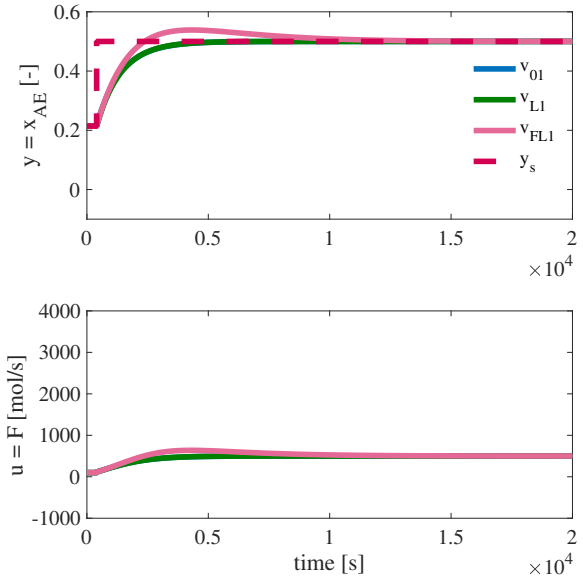

Figure 3.10: Closed loop setpoint change from x_{AF} = 0.214 to 0.5 [-] at time = 400s, for v_{01}, v_{L1} and v_{FL1}, with close loop time constant τ_c = 1000s. No physical limitations are put on the physical inputs u in the simulation.

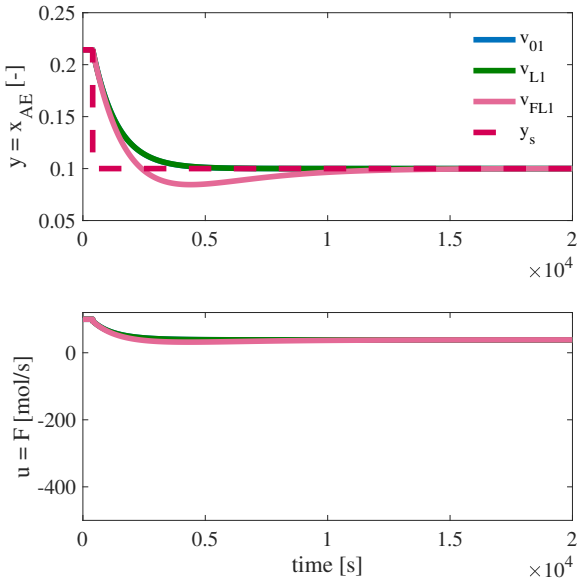

Figure 3.11: Closed loop setpoint change from x_{AF} = 0.214 to 0.1 [-] at time = 400s, for v_{01}, v_{L1} and v_{FL1}, with close loop time constant τ_c = 1000s. No physical limitations are put on the physical inputs u in the simulation.

Setpoint changes with alternative measurements

For this process three variations of the three MVs v_0, v_L and v_{FL} were proposed, v_{i1} which requires measurements of x_{AF} and E, v_{i2} which requires measurements of E and R and v_{i3} which requires measurements of R and S. In the previous section, the performances of the different MVs were tested with changing closed-loop constants, for the variants where x_{AF} and E were measured. For the closed-loop time constant similar to the nominal closed-loop time constant, all three performed well, so the different transformed MV will be tested for the different available measurement sets, to see if performance differ much. The performances for different measurement sets are given tables 3.5 to 3.7.

Table 3.5: Setpoint changes for v_{01}, v_{02} and v_{03}, for closed loop time constant $\tau_c = 100$s.

	New setpoint	
	0.5 [-]	0.1 [-]
v_{01}	28.5725	11.5072
v_{02}	28.5718	11.5215
v_{03}	28.5720	11.5198

Table 3.6: Setpoint changes for v_{L1}, v_{L2} and v_{L3}, for closed loop time constant $\tau_c = 100$s.

	New setpoint	
	0.5 [-]	0.1 [-]
v_{L1}	28.5721	11.4288
v_{L2}	28.5718	11.4287
v_{L3}	28.5718	11.4287

Table 3.7: Setpoint changes for v_{FL1}, v_{FL2} and v_{FL3}, for closed loop time constant $\tau_c = 100$s.

	New setpoint	
	0.5 [-]	0.1 [-]
v_{FL1}	41.6623	16.7649
v_{FL2}	44.1179	17.2154
v_{FL3}	43.4366	17.1334

The different variations based on the different available measurements performed very similar, when the closed loop time constant τ_c was 100 s. This is the case for v_0, v_L and v_{FL} for both negative and positive setpoint changes. The biggest differences were for v_{FL}, where v_{FL1} performed the best. The closed loop setpoint changes can be seen in figs. 3.12 to 3.17, where it as apparent that the different variations perform almost identical.

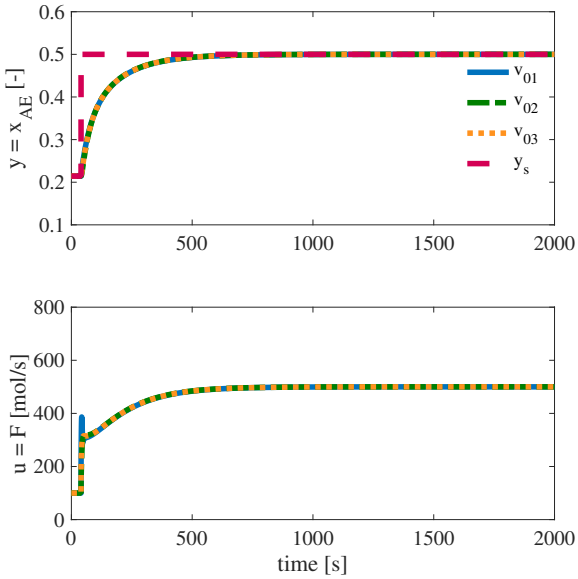

Figure 3.12: Closed loop setpoint change from x_{AF} = 0.214 to 0.5 [-] at time = 40s, for v_{01}, v_{02} and v_{03}, with close loop time constant τ_c = 100s. No physical limitations are put on the physical inputs u in the simulation.

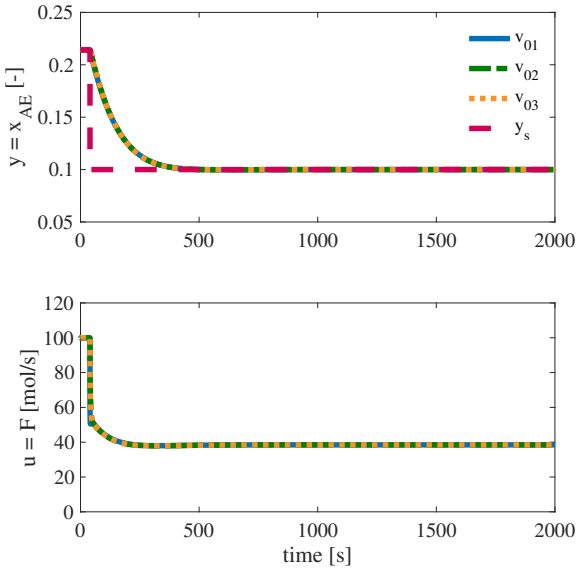

Figure 3.13: Closed loop setpoint change from x_{AF} = 0.214 to 0.1 [-] at time = 40s, for v_{01}, v_{02} and v_{03}, with close loop time constant τ_c = 100s. No physical limitations are put on the physical inputs u in the simulation.

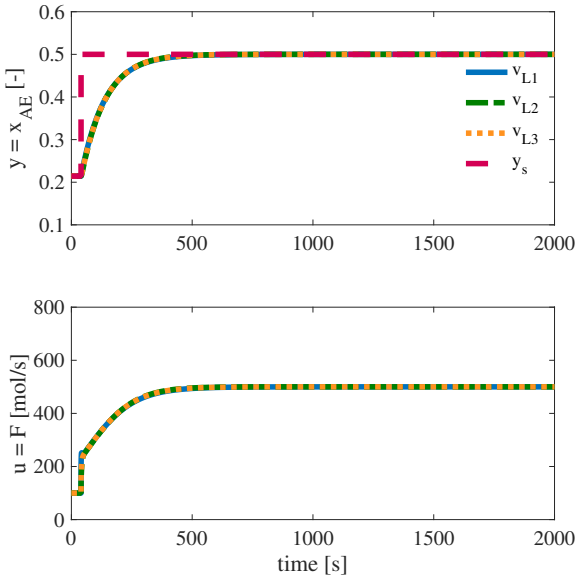

Figure 3.14: Closed loop setpoint change from x_{AF} = 0.214 to 0.5 [-] at time = 40s, for v_{L1}, v_{L2} and v_{L3}, with close loop time constant τ_c = 100s. No physical limitations are put on the physical inputs u in the simulation.

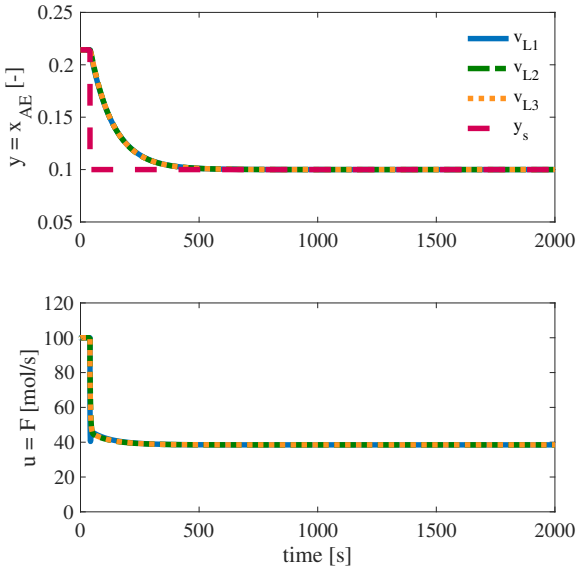

Figure 3.15: Closed loop setpoint change from x_{AF} = 0.214 to 0.1 [-] at time = 40s, for v_{L1}, v_{L2} and v_{L3}, with close loop time constant τ_c = 100s. No physical limitations are put on the physical inputs u in the simulation.

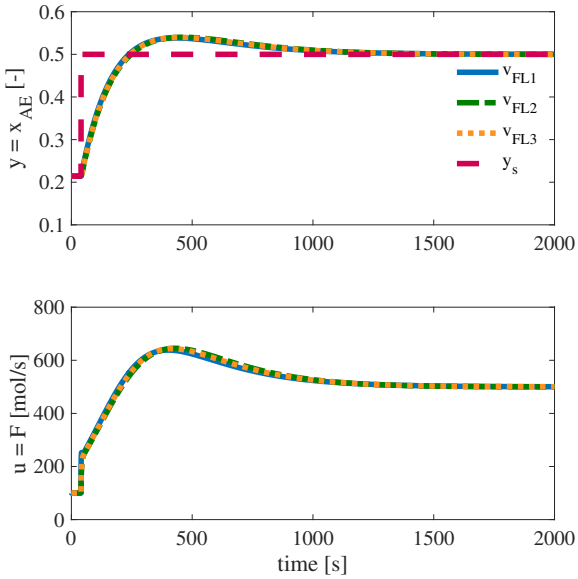

Figure 3.16: Closed loop setpoint change from x_{AF} = 0.214 to 0.5 [-] at time = 40s, for v_{FL1}, v_{FL2} and v_{FL3}, with close loop time constant τ_c = 100s. No physical limitations are put on the physical inputs u in the simulation.

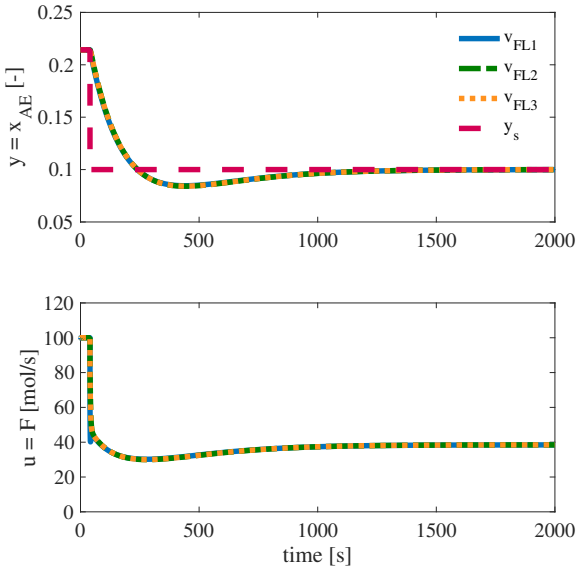

Figure 3.17: Closed loop setpoint change from x_{AF} = 0.214 to 0.1 [-] at time = 40s, for v_{FL1}, v_{FL2} and v_{FL3}, with close loop time constant τ_c = 100s. No physical limitations are put on the physical inputs u in the simulation.

3.4 Analysis of transformed MV design with model simplification

One of the proposed properties of the new MV transformation methodology is to introduce perfect disturbance rejection. This requires that all disturbances are measured. This is not always the case. In this case study, it was shown for the extraction tank example that it was possible to design a control system that rejects disturbances that are not measured with the new MV transformation methodology. To do this a model simplification was made when designing the transformed MVs. While this simplification made it such that perfect disturbance rejection was no longer possible as showed in section 3.3.1, different versions of the same MV could be designed which measured different sets of disturbances and other physical properties. All the different combinations rejected the disturbances without state feedback. This was also true for v_{02} which measured non of the two disturbances variables but still managed to reject disturbances in both of them.

In this case study, three different methodologies of designing transformed MVs were tested for three values of the closed-loop time constant. It was expected that for the static transformation v_0 would perform best for slow closed loop dynamics. The structure did perform well for slow closed-loop dynamics, however, it unexpectedly performed surprisingly well for closed-loop time dynamics similar to the open-loop dynamics. It should be noted though that the static transformation does not transform the non-linear system into a linear one, and that the open-loop dynamics change with changing operation. As expected the static case performed poorly for fast closed-loop dynamics. The linear transformation and feedback linearization cases both performed as expected. The model simplification does not seem to have affected the performances much when doing setpoint changes. It was also showed that the different formulations based on changing sets of measurements did not change the performance significantly. Overall the linear transformation performed best, but for cases where the closed-loop time constant is not too small, the static case worked as well. It should be noted that besides the model simplifications the different control structures were not tested for model error like output measurement delay, which would affect the linear case more as the transformation is dependent on the output.

Chapter 4

Case study 2: Mixing tank

4.1 Model description

A case study is conducted to test and compare different transformation strategies of a non-linear system into a linear system by the use of an input transformation. The robustness of the input transformation to model error and measurement delay will be considered. The new systems are linear with respect to the transformed input and states, but not necessarily with respect to the physical inputs and the disturbances. The input transformation is embedded with knowledge about the system and ideally introduces feedforward action and improved disturbance rejection.

To conduct this case study a mixing tank system is evaluated, with two inlet streams q_1 $[\mathrm{m}^3\,\mathrm{s}^{-1}]$ and q_2 $[\mathrm{m}^3\,\mathrm{s}^{-1}]$, and one outflow q $[\mathrm{m}^3\,\mathrm{s}^{-1}]$. Inlet stream q_1 has a concentration of c_1 $[\mathrm{kmol}\,\mathrm{m}^{-3}]$, inlet stream q_2 has a concentration of c_2 $[\mathrm{kmol}\,\mathrm{m}^{-3}]$, with c_1 being greater than c_2. The inventory M $[\mathrm{m}^3]$ (M is used to denote the volume and not V to not confuse it with the transformed MVs later) is assumed to be perfectly controlled, using q. Perfect mixing in the tank is assumed, and the density ρ is assumed to be constant as well.

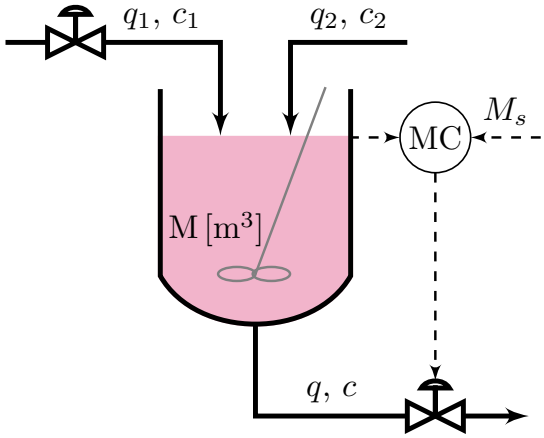

Figure 4.1: Flowsheet of the mixing tank process, with inflows q_1 and q_2 with concentrations c_1 and c_2. Outflow q with concentration c is used to perfectly control the inventory M.

The original input of the process is the first inlet flow:

$$u = q_1\ [\mathrm{m^3\,s^{-1}}]$$

The mixing tank concentration is used as state and process output:

$$y = c\ [\mathrm{kmol\,m^{-3}}]$$

The main disturbances are the second inlet flow, and the concentration of the two inlet flows:

$$d_1 = q_2\ [\mathrm{m^3\,s^{-1}}]$$
$$d_2 = c_1\ [\mathrm{kmol\,m^{-3}}]$$
$$d_3 = c_2\ [\mathrm{kmol\,m^{-3}}]$$

The residence time of the mixing tank is $\tau_r = \frac{M}{q_1+q_2}$.

The nominal operating conditions of the system are given in the table below,

Table 4.1: Nominal operating values of mixing tank process

Variable	Value	Unit
c^*	1.1	$\mathrm{kmol\,m^{-3}}$
q_1^*	1	$\mathrm{m^3\,s^{-1}}$
q_2^*	9	$\mathrm{m^3\,s^{-1}}$
M^*	10	$\mathrm{m^3}$
c_1^*	2	$\mathrm{kmol\,m^{-3}}$
c_2^*	1	$\mathrm{kmol\,m^{-3}}$
c_s^*	1.1	$\mathrm{kmol\,m^{-3}}$
τ_r^*	1	s

Two equations can be formulated to describe the system, the static mass balance

$$0 = \rho_1 q_1 + \rho_2 q_2 - \rho q \tag{4.1}$$

which can be further simplified due to the assumption of constant density ρ to

$$q = q_1 + q_2 \tag{4.2}$$

and the component balance

$$\frac{d(Mc)}{dt} = q_1 c_1 + q_2 c_2 - qc \tag{4.3}$$

which can be simplified due to the assumption of perfect inventory control to

$$M\frac{d(c)}{dt} = q_1 c_1 + q_2 c_2 - qc \tag{4.4}$$

Inserting eq. (4.2) into eq. (4.4) the model equation is found

$$M\frac{dc}{dt} = q_1 c_1 + q_2 c_2 - (q_1 + q_2)c \tag{4.5}$$

4.1.1 MV transformations

The model derived in eq. (4.5) is a non-linear model. To design a control structure for this system transformed manipulated values (MVs) can be constructed which introduce feedforward control and linearize the system. Five different transformed inputs have been proposed and are to be evaluated and compared with each other and the basecase. These five and the base case are:

$$v_0 = q_1 \tag{4.6a}$$

$$v_1 = c_1 q_1 + c_2 q_2 \tag{4.6b}$$

$$v_2 = \frac{c_1 q_1 + c_2 q_2}{q_1 + q_2} \tag{4.6c}$$

$$v_3 = q_1 c_1 + q_2 c_2 - (q_1 + q_2 + A)c; \qquad A = \left.\frac{df}{dc}\right|_* = -(q_1 * + q_2 *) \tag{4.6d}$$

$$v_4 = \frac{q_1}{q_2} \tag{4.6e}$$

$$v_5 = q_1(c_1 - c) + q_2(c_2 - c) \tag{4.6f}$$

The transformed MVs are not derived using a physical intuition of the system, but from a mathematical approach. Despite this some of these transformations do have physical interpretations, which can give a physical intuition as to why some transformations perform well or not.

The zeroth input v_0 is just the base case where the physical input u is directly manipulated by the controller. Feedforward control will not be combined with this control structure,

just feedback, so perfect disturbance rejection is not expected.

The first transformed MV v_1 is the simplest transformed MV which can constructed from the model equation in eq. (4.5), which is on the form

$$M\frac{dy}{dt} = a(u,d)y + b(u,d) \tag{4.7}$$

where $b(u,d)$ is selected to be the transformed input. This transformation, while similar to the special case of the static MV transform as seen in eq. (2.11), is classified under the general MV transform as given in eq. (2.17). This particular MV does however not introduce any implicit state feedback. The physical interpretation of v_1 is the component inflow rate into the mixing tank.

While v_1 was categorized as a general MV transform, it can be modified slightly to create an MV transform categorized as a static transform. This slight modification yields v_2, which is $-b(u,d/a(u,d)$. The physical interpretation of this MV is the concentration of the combined inlet stream into the mixing tank.

The third MV transformation is the linear MV transformation as outlined in section 2.3.1, where the scalar A is selected to be $df/dy|_*$ as suggested. This method of transformation creates a linear system, but the transformation is dependent on the state, while weakly near the nominal operation point due to the selection of A. It is interesting to test how well this control structure will perform when subject to model error and operating far away from the nominal operating point. This MV does not have a clear physical interpretation.

The fourth transformed input v_4 is the ratio between the input and the disturbance. This is a form of feedforward control known as ratio control.

The fifth transformed MV was the special case of linear transformation where A was selected to be zero. This transformation was dubbed as feedback linearization transformation as it is reminiscent of feedback linearization.

The inverse transform from v to u is given as,

$$u_0 = v_0 \tag{4.8a}$$

$$u_1 = \frac{v_1 - c_2 q_2}{c_1} \tag{4.8b}$$

$$u_2 = q_2\frac{v_2 - c_2}{c_1 - v_2} \tag{4.8c}$$

$$u_3 = \frac{v_3 + q_2(c - c_2) + Ac}{c_1 - c} \tag{4.8d}$$

$$u_4 = v_4 q_2 \tag{4.8e}$$

$$u_5 = \frac{v_5 + q_2(c - c_2)}{c_1 - c} \tag{4.8f}$$

It should be noted that the inverse transformation from v_2 to u is singular when $v_2 = c_1$, and that the inverse transforms from v_3 and v_5 to u are singular when $c = c_1$. While it seems unlikely that the inverse transformations of v_3 and v_5 will ever become singular as that requires the physical input q_1 to be infinite, however as c_1 approaches c so does the transformations approach singularity. The physical implications are that when c becomes large and the transformations reach the singularity, the physical input q_1 will become very large, which in turn leads c increasing, which in turn leads to q_1 increasing as well and so forth.

This may cause problems in the case studies when using v_2, v_3 and v_5 as MVs. It should also be noted that the inverse transforms from v_2 to u only give a positive feasible control input u when v_2 is greater than c_2 and lesser than c_1, with the reasonable constraint that u must be positive at all time. When pairing these MV transformations with feedback controllers, the feedback controllers might need to be conservatively tuned as to avoid these problems.

The transformed inputs are inserted into the model given in equation 4.5 to obtain six new model equations

$$M\frac{dc}{dt} = q_2(c_2 - c) + v_0(c_1 - c) \tag{4.9a}$$

$$M\frac{dc}{dt} = -c(q_1 + q_2) + v_1 \tag{4.9b}$$

$$M\frac{dc}{dt} = (q_1 + q_2)(-c + v_2) \tag{4.9c}$$

$$M\frac{dc}{dt} = Ac + v_3 \tag{4.9d}$$

$$M\frac{dc}{dt} = q_2(c_2 - c) + q_2 v_4(c_1 - c) \tag{4.9e}$$

$$M\frac{dc}{dt} = v_5 \tag{4.9f}$$

4.1.2 Theoretical analysis of transformed systems

One of the goals of these MV transformations is to transform a non-linear system into a linear one which can easily be controlled. The transformations for v_1 and v_2 both transforms the system into a system linear in the outputs, but not in the inputs. This non-linearity is expected to be noticed when the process is operated far away from the nominal operating point. This should be seen in the shape of response when large setpoint changes are made. v_1 and v_2 are quite similar but result in two transformed systems with an important difference. For the transformed system obtained from the v_1 transformation the initial gain from v to y is independent from u and d, while the transformed system obtained from the v_2 transformation the steady-state gain from v to y is independent from u and d. Due to this the v_2 transformation might more successfully reach its new steady-state far away from its

nominal operating point, but the initial response might cause some problems. When the dynamics of the system are important, which is the case when the closed-loop response time τ_c is fast, the initial response might be more important and v_2 might outperform v_1, but when the dynamics are not important, which is the case when the closed-loop response time is much slower than the dynamics, the steady-state gain might be more important and it is expected v_2 will outperform v_1.

The transformations v_3 and v_5 both transform the non-linear system into linear systems, but both v_3 and v_5 are dependent on the output feedback, so the output measurement delay should have an impact on the response. v_3 was designed such that when operating close to the nominal operating point the effect of the output should be minimized. Following this property, it is expected when making a small setpoint change, even with large amounts of output measurement delay the response should be close to a first-order response. When making a large setpoint change this property will no longer hold, and output measurement delay should affect the response.

For linear systems, a frequency analysis could be performed to analyze the effect of measurement delay. The MV transformations v_3 and v_5 both transform the nonlinear model to linear models so it could be assumed that a frequency analysis could be performed to analyze the effect of the measurement delay. However, this is not possible as the implicit output feedback into the calculation of u renders the transformed systems nonlinear when subject to measurement delay. Due to this a frequency analysis can not be used to analyze the effects of the implicit feedback, but a prediction of the effect will be discussed based on the inverse function from v to u.

In fig. 4.2 the transformation from v_3 to u given in eq. (4.8d) is plotted for different values of the output c. It can be seen that the value of the state affects what the physical input u will be. When there is output measurement delay the measured output will not match the actual output then. If the measured output is lower than the actual output the physical input which will be acted upon the system will be lower than the "correct" input. If the actual output is 1.5, but the measured output is 1.1 the physical input will be around $10\,\mathrm{m^2\,s^{-1}}$ too low, as seen in the figure. Similarly, if the actual output is 1.5 but the measured one is 1.8 the physical input will be around $30\,\mathrm{m^2\,s^{-1}}$ too high. If a positive setpoint change is made, this means until the output has reached its first peak, the output will be measured to be too small, which means the physical input will be smaller. This will lead to the controller in the outer loop having a more aggressive response to compensate. After the peak has been reached the physical input will be too aggressive, and the controller in the outer loop will compensate in the opposite direction. So measurement time delay will probably lead to a slow initial response followed by possibly destabilizing oscillations, depending on the amount of measurement delay.

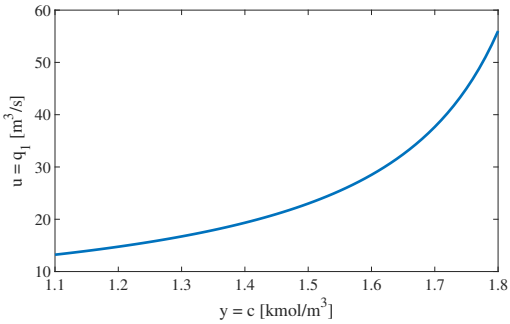

Figure 4.2: The transformation from v_3 to u for v_3 = 22 $\mathrm{kmol\,m^{-3}\,s^{-1}}$ and c varying between 1.1 and 1.8 $\mathrm{kmol\,m^{-3}}$, with all other parameters being at the nominal steady state values.

v_5 is as mentioned earlier, also dependent on the state and is expected to have similar problems relating to output measurement delay as v_3. It should also be noted that the linearised system obtained from this transformation is an integrating one. This type of system is difficult to control with feedback control, as a PI-controller will with this type of system give a second-order response and not a first-order when subject to a setpoint change. With just a P-controller the setpoint response will be first order, but with model error, this system will have a steady-state offset.

4.1.3 Theoretical analysis of disturbance rejection capabilities of transformed systems

The improved disturbance rejection is embedded into the calculation block, where the physical input is calculated from the transformed input, disturbances, and state feedback. If the model used to design the transformed MVs is perfect, perfect disturbance rejection can be achieved if the MV transformation is designed with perfect disturbance rejection in mind. To predict the disturbance rejection capabilities of each control strategy, the different transformed MVs and the transformed systems will be evaluated with respect to their disturbance rejection capabilities.

The disturbance rejection capabilities introduced by the v_1 transformation will be discussed. To do this the transformed model equation given in eq. (4.9b) will be evaluated. This equation is on the form

$$M\frac{dy}{dt} = a(u,d)y + v_1 \tag{4.10}$$

where $a(u,d) = -(q_1+q_2)$. At steady-state this control structure will try to keep $a(u,d)y$ and not y constant which is the desired response, so perfect disturbance rejection is not expected, but it might improve the disturbance rejection compared to the base case. To make a prediction about how different disturbances will affect the system eq. (4.8b) describing the transformation from v_1 to u will be evaluated. The initial response is an interesting aspect as feedback will reject the disturbance eventually. With a positive disturbance in q_2,

in which the inflow of the second flow increases. To keep the total amount of substance to a constant value this control structure will decrease u. The desired response to an increase in q_2 would be to increase u proportionally, however here the initial response is to lower u. This will make the disturbance rejection for disturbances in q_2 worse. With a positive disturbance in c_1, the initial response will be for u to decrease proportionally. While this will probably not give perfect disturbance rejection it will give improved disturbance rejection compared to the base case. For a negative disturbance in c_2 the initial response will be to increase u since less of the substrate is entering the tank with inlet stream 2. This will probably make the disturbance rejection for a disturbance in c_2 better compared with the base case.

v_2 was derived using the steady-state method, and will therefore with a perfect model perfectly reject disturbances at steady state. v_3 and v_5 were both derived using the linear transformation method which gives both perfect dynamic and steady-state disturbance rejection given they are designed with a perfect model.

The physical interpretation of v_4 is the ratio between inflow q_1 and q_2, so it is expected that a disturbance in q_2 will be perfectly rejected. As the transformed input does not know the effect of c_1 and c_2 on the system it is not expected that this formulation will have improved disturbance rejection for disturbances in c_1 and c_2 compared to the base case.

4.2 Control tuning

In this case study the transformed MVs will be paired with feedback controllers. The controllers in question are PI-controllers. The controller will be tuned using linear control theory, even for the non-linear transformed systems. To do this the model equations found in eq. (4.9) were linearised around the nominal operating point of the system. The linearised model equations formulated using deviation variables are

$$\begin{aligned} M\frac{d\Delta c}{dt} =& -(q_1^* + q_2^*)\Delta c + (c1^* - c^*)\Delta v_0 \\ &+ (c_2^* - c^*)\Delta q_2 + q_2^*\Delta c_2 + v_0^*\Delta c_1 && \text{(4.11a)} \\ M\frac{d\Delta c}{dt} =& -(q_1^* + q_2^*)\Delta c + \Delta v_1 - c^*\Delta q_1 - c^*\Delta q_2 && \text{(4.11b)} \\ M\frac{d\Delta c}{dt} =& (q_1^* + q_2^*)(-\Delta c + \Delta v_2) && \text{(4.11c)} \\ M\frac{d\Delta c}{dt} =& A\Delta c + \Delta v_3 && \text{(4.11d)} \\ M\frac{d\Delta c}{dt} =& -(q_1^* + q_2^*)\Delta c + q_2^*(c_1^* - c^*)\Delta v_4 \\ &+ q_2^*\Delta c_2 + q_1^*\Delta c_1 && \text{(4.11e)} \\ M\frac{d\Delta c}{dt} =& \Delta v_5 && \text{(4.11f)} \end{aligned}$$

where the deviation variables are

- $\Delta c = c - c^*$
- $\Delta v_i = v_i - v_i^*, \quad \forall i \in \{0, 1, 2, 3, 4, 5\}$
- $\Delta q_1 = q_1 - q_1^*$
- $\Delta q_2 = q_2 - q_2^*$
- $\Delta c_1 = c_1 - c_1^*$
- $\Delta c_2 = c_2 - c_2^*$

The linearized equations are linearized with respect to the state, transformed input, and disturbances. As previously discussed in section 4.1.3, v_2, v_3 and v_5 introduces perfect disturbance rejection which can be seen in the linearized system equations given in eq. (4.11), as they are not dependent on the disturbances. The linearized system obtained with v_4 is dependent on disturbances in c_1 and c_2, but not q_2 as expected. The linearized system obtained using the v_1 transformation is dependent on both q_2 and q_1, and since q_1 is the physical input u. The physical input Δq_1 is treated as a disturbance in eq. (4.11b), and no extra feedforward control will be used in this case study but the one introduced with the MV transformation.

The linearised dynamic equations are transformed from the time domain to the frequency domain using a Laplace transformation. The deviation variable notation is dropped for simplicity, and only the transfer functions from the transformed input to the output is considered. The system equations are formulated as either a first-order transfer function on the form

$$\frac{y}{v}(s) = \frac{k}{\tau s + 1} \tag{4.12}$$

or an integrating transfer function on the form

$$\frac{y}{v}(s) = \frac{k'}{s} \tag{4.13}$$

where the process gain k, open loop time constant τ and the integral process gain k' are given for the six system formulations in table 4.2.

Table 4.2: Gain and time constant for first order transfer function and integrating process.

MV	k	k′	τ
v_0	$\frac{c_1^* - c^*}{q_1^* + q_2^*}$ [kmol s m^{-6}]	-	$\frac{M}{q_1^* + q_2^*}$
v_1	$\frac{1}{q_1^* + q_2^*}$ [s m^{-3}]	-	$\frac{M}{q_1^* + q_2^*}$
v_2	1 [-]	-	$\frac{M}{q_1^* + q_2^*}$
v_3	$\frac{1}{q_1^* + q_2^*}$ [s m^{-3}]	-	$\frac{M}{q_1^* + q_2^*}$
v_4	$\frac{q_2^*(c_1^* - c^*)}{q_1^* + q_2^*}$ [kmol s^{-1}]	-	$\frac{M}{q_1^* + q_2^*}$
v_5	-	$\frac{1}{M}$ [m^{-3}]	-

The numeric values for the process gain k, open loop time constants and the integral process gain k' are given in table 4.3.

Table 4.3: Gain and time constant for first order transfer function and integrating process.

MV	k	k′	τ
v_0	0.09 [$\mathrm{kmol\,s\,m^{-6}}$]	-	1
v_1	0.1 [$\mathrm{s\,m^{-3}}$]	-	1
v_2	1 [-]	-	1
v_3	0.1 [$\mathrm{s\,m^{-3}}$]	-	1
v_4	0.81 [$\mathrm{kmol\,s^{-1}}$]	-	1
v_5	-	0.1 [$\mathrm{m^{-3}}$]	-

With the process gain and time constant for the six system formulations the controller gain K_c and integral time τ_I can be found using the SIMC rules. The tuning parameter τ_c representing the closed-loop time constant is not specified for the controller tunings given in table 4.4, but will be specified for different cases in section 4.3.

Table 4.4: Proportional gain and integrating time constant, found using SIMC-rules.

MV	K_c	τ_I
v_0	$\frac{M}{c_1^*-c^*}\frac{1}{\tau_c+\theta}$	$\min(\frac{M}{q_1^*+q_2^*}, 4(\tau_c+\theta))$
v_1	$\frac{M}{\tau_c+\theta}$	$\min(\frac{M}{q_1^*+q_2^*}, 4(\tau_c+\theta))$
v_2	$\frac{M}{q_1^*+q_2^*}\frac{1}{\tau_c+\theta}$	$\min(\frac{M}{q_1^*+q_2^*}, 4(\tau_c+\theta))$
v_3	$\frac{M}{\tau_c+\theta}$	$\min(\frac{M}{q_1^*+q_2^*}, 4(\tau_c+\theta))$
v_4	$\frac{M}{q_2^*(c_1^*-c^*)}\frac{1}{\tau_c+\theta}$	$\min(\frac{M}{q_1^*+q_2^*}, 4(\tau_c+\theta))$
v_5	$\frac{M}{\tau_c+\theta}$	$4(\tau_c+\theta)$

In table 4.5 the nominal values is inserted into the controller tunings to give the numeric values.

Table 4.5: Proportional gain and integrating time constant, found using SIMC-rules.

MV	K_c	τ_I
v_0	$\frac{100}{9}\frac{1}{\tau_c+\theta}$	$\min(1, 4(\tau_c+\theta))$
v_1	$\frac{10}{\tau_c+\theta}$	$\min(1, 4(\tau_c+\theta))$
v_2	$\frac{1}{\tau_c+\theta}$	$\min(1, 4(\tau_c+\theta))$
v_3	$\frac{10}{\tau_c+\theta}$	$\min(1, 4(\tau_c+\theta))$
v_4	$\frac{100}{81}\frac{1}{\tau_c+\theta}$	$\min(1, 4(\tau_c+\theta))$
v_5	$\frac{10}{\tau_c+\theta}$	$4(\tau_c+\theta)$

4.3 Case studies

To test and compare the five strategies proposed, the systems will be subjected to large disturbances in the three major disturbance variables. Since not all the transformed systems are linear with respect to the disturbances and inputs, and the controllers are tuned around the nominal operating point, it is interesting to see if the performance is good when operating is far away from the nominal operating point. Setpoint changes will also be considered. A small setpoint change where the linearisation is assumed to be valid and a larger one where the system is brought into a region where the linearisation is no longer valid will be both be evaluated and compared.

To further study the differences between different MV transformation strategies, model error will be considered. In real system a perfect model is o achieve, and some model error will be present. Model error can be represented in many different ways. One type of model error which will be considered in the case study is output measurement delay. This might cause extra problems for certain MV transformation which are dependent on the output measurement. Three values of measurement delay θ will be considered,

- Long time delay, $\theta = 3\tau_r = 3\text{s}$
- Short time delay, $\theta = 0.3\tau_r = 0.3\text{s}$
- No time delay, $\theta = 0\text{s}$

For the cases where the measurement delay is non-zero the closed loop tuning parameter τ_c used to tune the PI-controllers will be selected to be equal to the measurement time delay, and when the measurement time delay is zero it is selected to be $0.3\tau_r$,

$$\begin{aligned} \tau_c &= \theta, & \theta &> 0 \\ \tau_c &= 0.3\tau_r = 0.3\text{s}, & \theta &= 0 \end{aligned}$$

A different type of model error which will be considered in this case study is gain error, in this case study input gain error is considered in section 4.3.2, and disturbance gain error is considered in section 4.3.3

4.3.1 Uncertainty in the measurement delay

The first case which will be considered is the case where only uncertainty in the measurement delay is present, and no gain errors.

Disturbance rejection

Results

A disturbance test was conducted for the 5 transformed MVs and the basecase for the 3 different disturbance variables. The IAE is given in tables 4.6, 4.7, 4.8 and 4.9.

Table 4.6: Integrated absolute error, for step in disturbance $d_1 = q_2$ of $9\,\mathrm{m}^3\,\mathrm{s}^{-1}$. For v_5 both a PI- and a P-controller are used.

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	0.5400	6,2581	0.0000	0.0000	0.0000	0.0000	0.0000
0.3	0.0540	0,6569	0.0000	0.0000	0.0000	0.0000	0.0000
0	0.0270	0,3247	0.0000	0.0000	0.0000	0.0000	0.0000

Table 4.7: Integrated absolute error, for step in disturbance $d_1 = q_2$ of $-4.5\,\mathrm{m}^3\,\mathrm{s}^{-1}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	0.8848	3.4971	0.0000	0.0000	0.0000	0.0000	0.0000
0.3	0.0270	0.3740	0.0000	0.0000	0.0000	0.0000	0.0000
0	0.0135	0.1691	0.0000	0.0000	0.0000	0.0000	0.0000

Table 4.8: Integrated absolute error, for step in disturbance $d_2 = c_1$ of $1\,\mathrm{kmol}\,\mathrm{m}^{-3}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	0.9994	0.3474	0.0000	0.0000	0.9994	0.0000	0.0000
0.3	0.0289	0.0347	0.0000	0.0000	0.0289	0.0000	0.0000
0	0.0142	0.0174	0.0000	0.0000	0.0142	0.0000	0.0000

Table 4.9: Integrated absolute error, for step in disturbance $d_3 = c_2$ of $-0.5\,\mathrm{kmol}\,\mathrm{m}^{-3}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	2.7000	3.2832	0.0000	0.0000	2.7000	0.0000	0.0000
0.3	0.2700	0.3299	0.0000	0.0000	0.2700	0.0000	0.0000
0	0.1349	0.1643	0.0000	0.0000	0.1349	0.0000	0.0000

As predicted, when using v_2, v_3, and v_5 as transformed MVs perfect disturbance rejection was achieved for all three disturbances. When using v_4 as the transformed MV perfect disturbance rejection was achieved for disturbances in q_2 as expected. For a positive disturbance in c_1, the disturbance is rejected due to the feedback loop and happens to be better than the base case. On the contrary for a negative disturbance in c_2 the disturbance rejection is worse.

For v_1 the disturbance rejection for disturbances in q_2 is worse than the base case, as expected. This can be seen in figure 4.3 for a positive disturbance or in figure 4.4 for a negative disturbance. As expected the initial response in the input is opposite to the desired response.

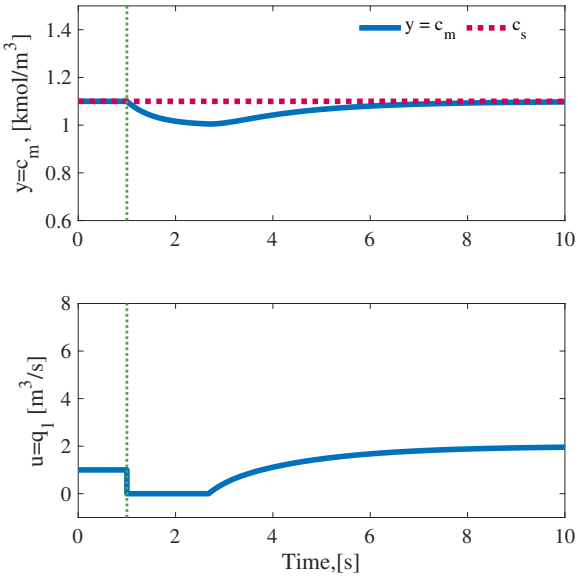

Figure 4.3: Step test in DV q_2 of 9 $\mathrm{m^3\,s^{-1}}$ at time 1 s for v_1 as the transformed MV, without measurement time delay. The physical input u is constrained to zero as the lowest value, and no input saturation is used. c_m is the measured output, and c_s is the output setpoint.

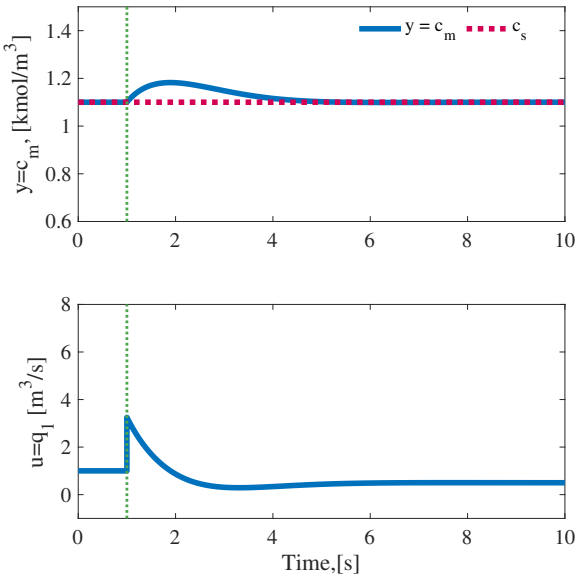

Figure 4.4: Step test in DV q_2 of -4.5 $\mathrm{m}^3\,\mathrm{s}^{-1}$ at time 1 s for v_1 as the transformed MV, without measurement time delay. The physical input u is constrained to zero as the lowest value, and no input saturation is used.

The feedforward action introduced in v_1 is expected to improve the disturbance rejection capabilities in c_1 and c_2, however improved performance is only achieved for a positive disturbance in c_1 when subject to large output measurement delay. When the system is subject to less measurement delay or even no measurement delay the base case is performing slightly better judging by the IAE given in table 4.8. The same is the case for a negative disturbance in c_2 for the three different amounts of measurement delay. This is not the expected behavior, as it was expected the feedforward action introduced by the calculation block would improve the disturbance rejection. This can be explained using figure 4.5 and 4.6. In these two figures, the system is subject to a change in the DV c_2 with a measurement time delay of 3 seconds, for the base case and the control structure where v_1 is used as a transformed MV. As predicted the feedforward action in v_1 initially makes a correction in the right direction with the input u immediately changing to approximately the halfway point between the old and the new steady-state input. The effect of this feedforward action is the output deviation having a smaller spike, but it settles slower than the base case, thus the higher IAE. Since the performance measurement IAE measures the area of the deviation from the setpoint, the base case performs better than v_1, however if another performance measurement which penalized operating far away from the setpoint more harshly, it could be argued that v_1 performs better.

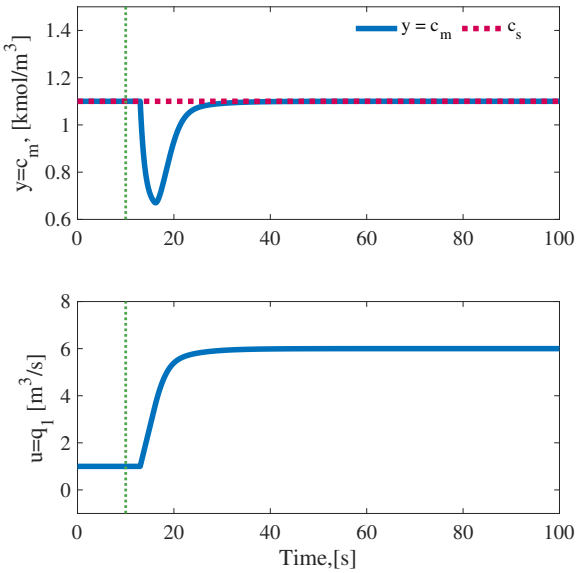

Figure 4.5: Step test in DV c_2 of -0.5 $\mathrm{kmol\,m^{-3}}$ at time 10 s for the basecase, with measurement time delay θ of 3 s.

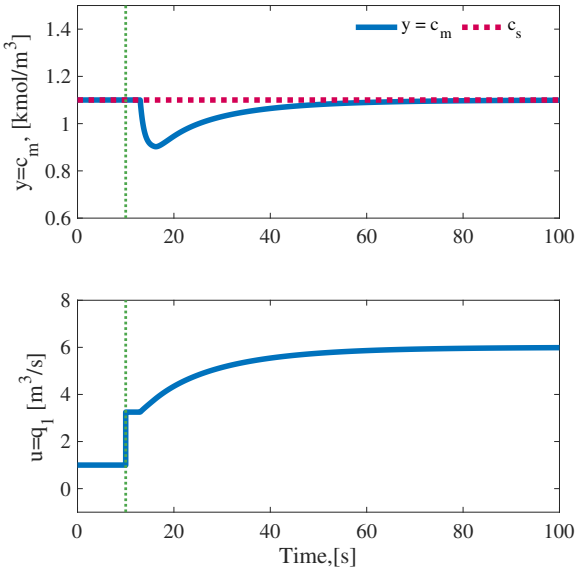

Figure 4.6: Step test in DV c_2 of -0.5 $\mathrm{kmol\,m^{-3}}$ at time 10 s for v_1 as the transformed MV, with measurement time delay θ of 3 s.

Setpoint change

With these concerns v_3 should probably overall give the best performance, as it should have good and maybe even the best performance when making small setpoint changes, even with output measurement delay as the effect of the state should be small when operating close to nominal operations. When doing a large setpoint change it should most definitely give the best performance when subject to no measurement delay, and it should still perform well, albeit less good when subject to a small amount of measurement delay. When subject to a large amount of measurement delay the performance is expected to not be so good, and oscillations are expected.

v_2 is expected to give more aggressive responses than v_1. It is also expected that for small closed-loop time dynamics, v_2 might be too aggressive leading to bad control, so it is expected v_1 might be better. Both v_1 and v_2 are expected to have troubles when subject to large setpoint changes.

v_5 is expected to overshoot when paired with a PI-controller. Measurement delay is expected to make the response worse, especially when subject to large setpoint changes, and large amount measurement delay. When controlled with a P-controller it is expected to perform quite well. When subject to a large amount of output measurement delay it is expected to still have a stable response but slow.

v_4 is expected to behave similarly to the base case and not have any other special properties.

Results

The performance for the different control structures can be found in table 4.10 for small setpoint changes, and in table 4.11 for big setpoint changes.

Table 4.10: Integrated absolute error, for setpoint change in $y_s = c_s$ from 1.1 to 1.2 $\mathrm{kmol\,m^{-3}}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	0.6752	1.5000	0.6391	0.6924	0.6752	3.5931	2.5195
0.3	0.0675	0.1500	0.0634	0.0685	0.0675	0.1354	0.0813
0	0.0337	0.0750	0.0300	0.0300	0.0337	0.0441	0.0300

Table 4.11: Integrated absolute error, for setpoint change in $y_s = c_s$ from 1.1 to 1.7 $\mathrm{kmol\,m^{-3}}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	10.7732	22.7671	3.6000	6.4003	10.7732	35.2809	21.8977
0.3	1.0747	2.2556	18.7586‡	0.5582	1.0747	0.9975	0.6154
0	0.5399	1.1882	18.7889‡	0.1800	0.5399	0.2649	0.1800

As predicted v_3 overall performed the best, especially for large setpoint changes, even when subjected to $3\,\mathrm{s}$ of measurement time delay. When subject to large measurement delay it was predicted that closed-loop setpoint response might be an oscillating response, this is however not the case as the response is just an overshoot followed by an undershoot which quickly settles as seen in fig. 4.7.

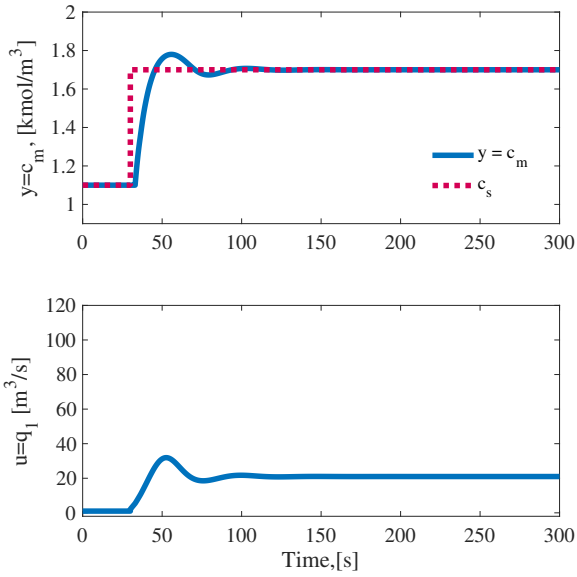

Figure 4.7: Setpoint change from c_s = 1.1 $\mathrm{kmol\,m^{-3}}$ to c_s = 1.7 $\mathrm{kmol\,m^{-3}}$, with measurement delay $\theta = 3\,\mathrm{s}$, for v_3 as the transformed MV.

Interestingly v_2 is the best for small setpoint changes, no matter what the closed-loop time was. It should be noted that with $0.3\,\mathrm{s}$ of measurement delay the response is a bit aggressive which leads to a minor overshoot, as can be seen in fig. 4.8.

‡Infeasible control

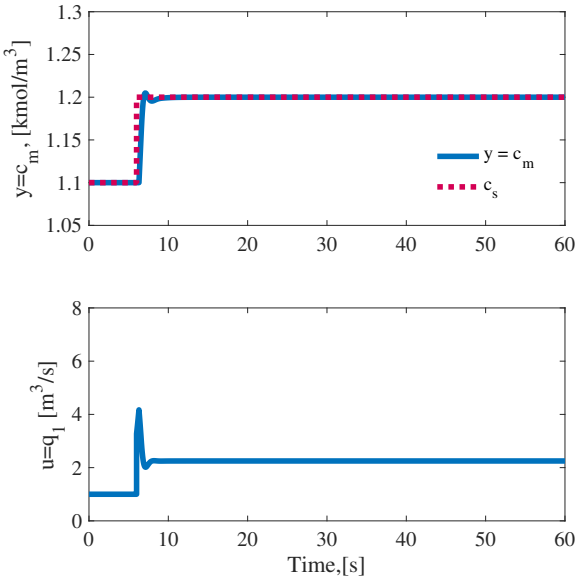

Figure 4.8: Setpoint change from c_s = 1.1 $\mathrm{kmol\,m^{-3}}$ to c_s = 1.2 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 0.3 s, for v_2 as the transformed MV.

v_2 is as predicted best when the closed-loop constant was large, which is when the measurement delay θ is large. For smaller closed-loop time constants, which is the case when the time delay is small or zero, v_2 works quite poorly when doing large setpoint changes, as the controller gives a nonphysical input u making the process operate at a zero input, as seen in fig. 4.9. This is infeasible control in the table, as the control structure fails at controlling the system. This response might seem odd, as the physical input immediately goes to zero when the setpoint change occurs, and it does not change. The reason it goes to zero is that the simulation is restrained such that the u cannot be less than zero as this is a non-physical value. If this constraint was not in place the physical input would have been less than zero. The reason for this is due to the fact the inverted function which calculates u from v_2 given in eq. (4.8c) will only yield positive values for v between c_1 and c_2. So this response is due to the control structure being too aggressive for small τ_c, as was a predicted issue. If the calculation block was implemented differently, like being constrained to only give positive values, and giving some max or min value if v is not in the region which gives a feasible input, this problem could be solved. This implementation should be coupled with anti-windup, for when v is brought out of the feasible region.

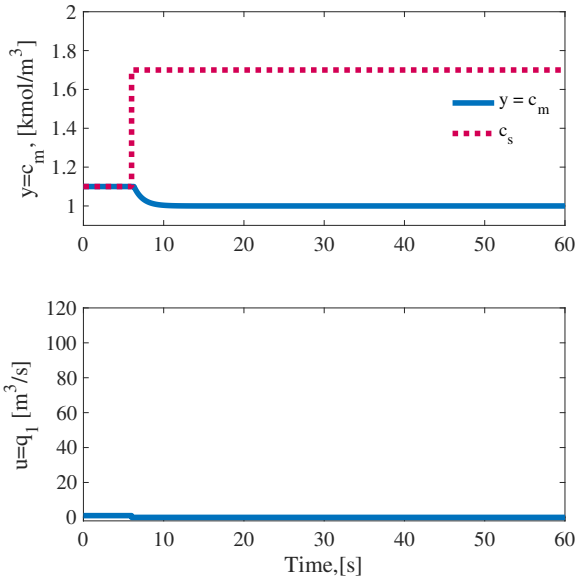

Figure 4.9: Setpoint change from c_s = 1.1 to c_s = 1.7 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 0.3 s, for v_2 as the transformed MV.

v_2 has a more conservative response than v_1 as expected, but is more conservative than the base case when tuned with the same closed-loop time constant. Interestingly v_2 and v_4 the same exact response for all the different cases.

As expected v_5 performs well when controlled with a P-controller when the measurement delay is small. When controlled with a PI-controller it performs as expected. With a large amount of output measurement delay, this structure performs the worst for both small and large setpoint changes, albeit it does not go unstable. As this formulation alongside v_3 transforms the non-linear system into a linear system, it performs well when doing large setpoint changes with small amounts of measurement delay, but not as well v_3.

4.3.2 Model error(Input gain mismatch)

The different MV transformations gave overall good performance when designed with a perfect model, and subject to only measurement delay as model error. So to further test the robustness of the different transformation strategies model error in the form of input gain error will be introduced. This is that the actual gain k_u from input u is different from the one used to derive the input transformations. This is implemented such that the model is perfect at steady state, but the difference between the actual input u_r which acts on the system and the nominal input u^* is wrong by some gain K_u. If the physical input which is given by the control system is u, the actual input u_r which acts on the system is

$$u_r = u^* + K_u(u - u^*) \tag{4.14}$$

The input gain error which will be used in this case study is K_u = 2.

Disturbance rejection

With a perfect model v_2, v_3 and v_5 introduced perfect feedforward disturbance rejection. When designed using a perfect model all of these three were expected to introduce perfect disturbance rejection. With model error, this is no longer the case. The feedforward action performed by these MV transformations might improve rejecting disturbances, or it might on the contrary worsen it, in the worst-case scenario making the process go unstable. It should be noted that the MV transformations are coupled with feedback controllers, so even though the initial feedforward action does not make the process go unstable, the feedback action might.

With an input gain mismatch of $K_u = 2$, it is expected that the feedforward action will be too aggressive since the absolute difference between the input and nominal input will be greater. This means the output will start deviating from the setpoint in the opposite direction of the one which the disturbance would have pushed it towards. With a gain mismatch higher than one it is expected that the feedforward action will overshoot. Depending on the magnitude of the gain error the feedforward action might be beneficial and reducing the impact of the disturbance. If the feedforward action is too large, the process will be brought further away from the setpoint in the opposite direction, then what the disturbance would with no feedforward action.

Since perfect feedforward action is not expected, feedback is also important to reject disturbances. Feedback action was previously discussed when doing setpoint changes with a perfect model. v_3 and v_5 are both dependent on output measurements and might lead to instability when subject measurement delay. v_3 is expected to perform better than v_5 for two reasons. It is expected the disturbances will not move the system too far away from nominal operation, and therefore v_3 will not be as heavily impaired by the measurement delay. The second is that v_5 transforms the system into and integrating one which is difficult to control. It should also be pointed out that v_5 coupled with a P-controller will never reach the steady-state but have an offset due to the model error.

Using v_1 as the transformed MV was found to not be a very good idea, for systems without gain error. For disturbances in q_2 it actively worsened the performance, and for disturbances in c_1 and c_2 the feedforward action improved the disturbance rejection, but there was not perfect disturbance rejection. With an input gain error of 2 the disturbance rejection of q_2 is expected to be even worse compared to the ideal case without input gain uncertainty. For disturbances in c_1 and c_2, the disturbance rejection might improve compared to the ideal case without input gain uncertainty.

Disturbance rejection with input gain error results

Disturbance step tests were done for the different control structures when subject to input gain error, and the performances are given in tables 4.12 to 4.15 below.

Table 4.12: Integrated absolute error, for step in disturbance $d_1 = q_2$ of $9\,\mathrm{m^3\,s^{-1}}$, when subject to input gain error of $K_u = 2$. The footnotes denote cases where the process does not settle to the new setpoint for differing reasons, and the IAE should be infinity if the simulation ran forever.

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	0.2701	5.9951	0.2681	5.9771*	0.2836	64.4555**	58.5511†
0.3	0.0270	0.5999	0.3662*	0.0279	0.3500*	0.1089	1.0582†
0	0.0135	0.2992	0.0069	0.0135	0.0067	0.0162	0.1660†

Table 4.13: Integrated absolute error, for step in disturbance $d_1 = q_2$ of $-4.5\,\mathrm{m^3\,s^{-1}}$, when subject to input gain error of $K_u = 2$. The footnotes denote cases where the process does not settle to the new setpoint for differing reasons, and the IAE should be infinity if the simulation ran forever.

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	5.4073*	4.5953	1.1916	4.7818*	0.9954	9.0962**	8.4777†
0.3	0.0155	0.3681	0.0257	0.0216	0.0270	0.0324	0.2844†
0	0.0067	0.1500	0.0129	0.0067	0.0135	0.0081	0.0650†

Table 4.14: Integrated absolute error, for disturbance in $d_2 = c_1$ of $1\,\mathrm{kmol\,m^{-3}}$, when subject to input gain error of $K_u = 2$. The footnotes denote cases where the process does not settle to the new setpoint for differing reasons, and the IAE should be infinity if the simulation ran forever.

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	6.8596*	0.1797	0.9262	0.8393	6.8595*	8.5887**	8.5887†
0.3	0.2670*	0.0126	0.0308	0.0300	0.2667*	0.0720	0.7465†
0	0.0071	0.0063	0.0154	0.0150	0.0071	0.0180	0.1557†

*Stable oscillation
**Unstable
†Steady-state offset

Table 4.15: Integrated absolute error, for step in disturbance $d_3 = c_2$ of $-0.5\,\mathrm{kmol\,m^{-3}}$, when subject to input gain error of $K_u = 2$. The footnotes denote cases where the process does not settle to the new setpoint for differing reasons, and the IAE should be infinity if the simulation ran forever.

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	2.3400	0.3000	1.2912	3.0328	2.3400	90.8393**	44.1537**
0.3	0.1350	0.0300	0.1080	0.1350	0.1350	0.3244	3.3388†
0	0.0675	0.0150	0.0540	0.0675	0.0675	0.0810	0.7007†

From the IAE tables, it can be seen that the structures dependent on the output measurement, struggles when the output measurement is large combined with large model error, at least in the form of input gain error. For v_3 this is the case for both positive and negative disturbances in q_2, where it instead of settling to the steady-state, it starts oscillating around the steady-state value and later stabilizes to stabile oscillations, as seen in fig. 4.10.

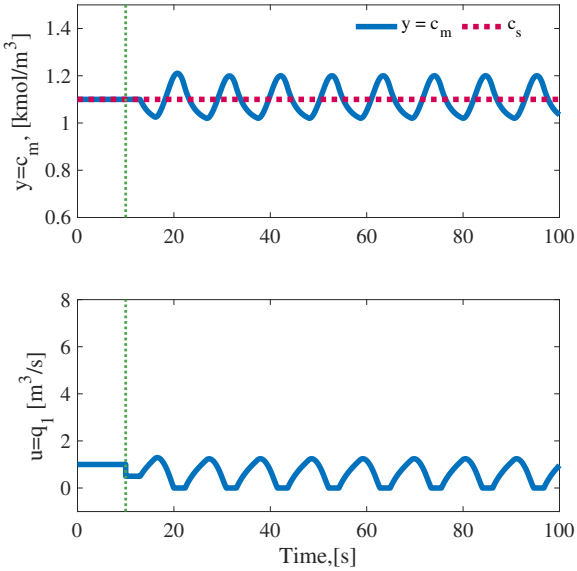

Figure 4.10: Step test in DV q_2 of -4.5 $\mathrm{m^3\,s^{-1}}$ using v_3 as the transformed MV, with measurement time delay θ of 3 s and input gain mismatch of 2.

For v_5 controlled with a PI-controller the system goes unstable for disturbances in all three disturbances when subject to large measurement delay as seen in fig. 4.11.

*Stable oscillation
**Unstable
†Steady-state offset

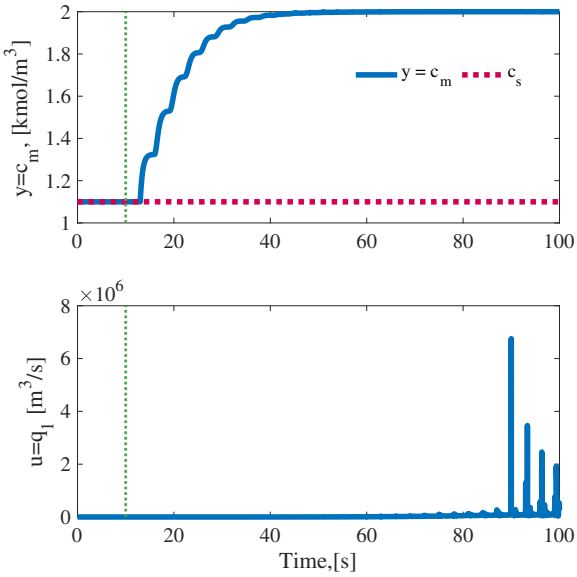

Figure 4.11: Step test in DV c_2 of -0.5 $\mathrm{kmol\,m^{-3}}$ using v_5 as the transformed MV, controlled with a PI-controller, with measurement time delay θ of 3 s and input gain mismatch of 2.

For small amounts of measurement delay, v_2 and v_3 are about as good as the base case. This is due to the feedforward action being too aggressive and not improving the performance as speculated. It should be noted that the feedback when using v_2 is quite aggressive as previously discussed, and this leads to the process not settling when subject to a positive disturbance in q_2 when the measurement time delay was $0.3\,\mathrm{s}$, but instead going to a state of stable oscillations as seen in fig. 4.12.

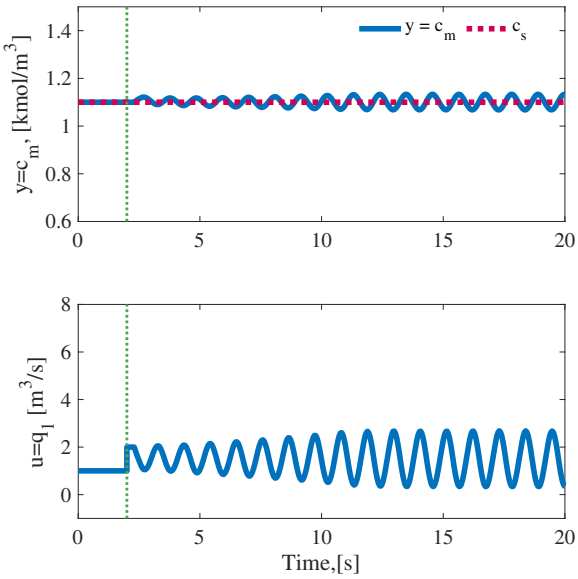

Figure 4.12: Step test in DV q_2 of 9 $\mathrm{m}^3\,\mathrm{s}^{-1}$ using v_2 as the transformed MV, with measurement time delay θ of 0.3 s and input gain mismatch of 2.

v_5 performs decently when subject to small amounts of measurement delay, but due to being an integrating process performs worse than v_3. When using a P-controller to control the system obtained when using v_5 as a MV transform the process ends up with a steady state offset as expected.

As expected v_2 is even worse at rejecting disturbances in q_2, but as speculated the gain error makes this structure better at rejecting disturbances in c_1 and c_2 by happenstance, with this structure actually being the best at rejecting disturbances in these two.

Setpoint change

What is the expected effect of an input gain error? The input which the controller "believes" it acts upon the system is different from the actual input. Without measurement delay, the outer feedback loop should compensate for this, and overall the setpoint changes should be swifter. With measurement delay, the aggressive input action due to the high gain error might lead to some oscillations or instability. This is especially true for v_2 which already gives aggressive control. v_3 and v_5 which are dependent on the output might go unstable when the measurement time delay is large, but for small setpoint changes, this should not be a large problem for v_3.

Table 4.16: Integrated absolute error, for setpoint change in $y_s = c_s$ from 1.1 to 1.2 $\mathrm{kmol\,m^{-3}}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	0.6597	0.7500	0.7278	0.8317	0.6597	110.0210**	51.8534**
0.3	0.0718	0.0750	0.0893	0.0765	0.0718	0.1658	1.1672†
0	0.0169	0.0375	0.0159	0.0150	0.0169	0.0394	0.4792†

Table 4.17: Integrated absolute error, for setpoint change in $y_s = c_s$ from 1.1 to 1.7 $\mathrm{kmol\,m^{-3}}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	5.4000	11.9480	2.9114	7.2953	5.4000	23.9777**	17.1453**
0.3	0.5399	1.1909	21.9256‡	0.4390	0.5399	1.1916	6.7036†
0	0.2700	0.5998	21.9938‡	0.0900	0.2700	0.2362	2.8750†

As expected for large measurement delay v_5 goes unstable for both small and large setpoint changes as can be seen in fig. 4.13. Interestingly v_3 does not. For small setpoint changes, all the control structures perform similarly except for v_5, though the responses are different. v_2 and v_3 have responses with an undershoot followed by a overshoot which quickly settles when there is measurement delay as seen in fig. 4.14, while v_1 and v_4 have first-order responses as seen in fig. 4.15. It should also be pointed out the base case performs as well as the systems with MV transforms for small setpoint changes, however, this is to be expected.

‡Infeasible control
**Unstable
†Steady-state offset

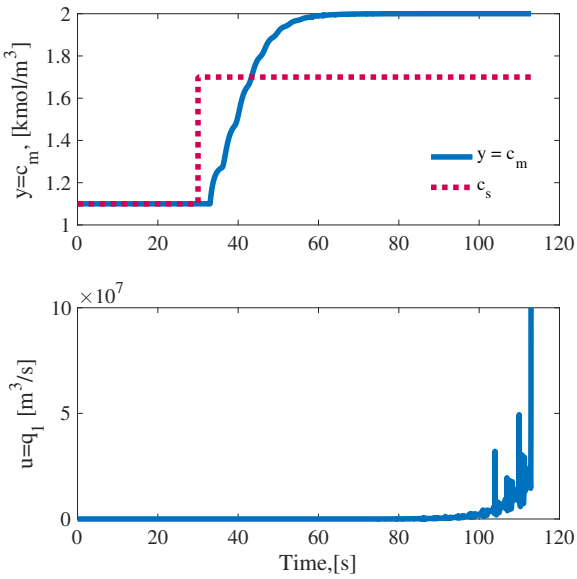

Figure 4.13: Setpoint change from c_s = 1.1 to 1.7 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 3 s, for v_6 as the transformed MV, controlled with a PI-controller and input gain mismatch of 2. The simulation crashes after around 115 second as the physical input goes to infinity.

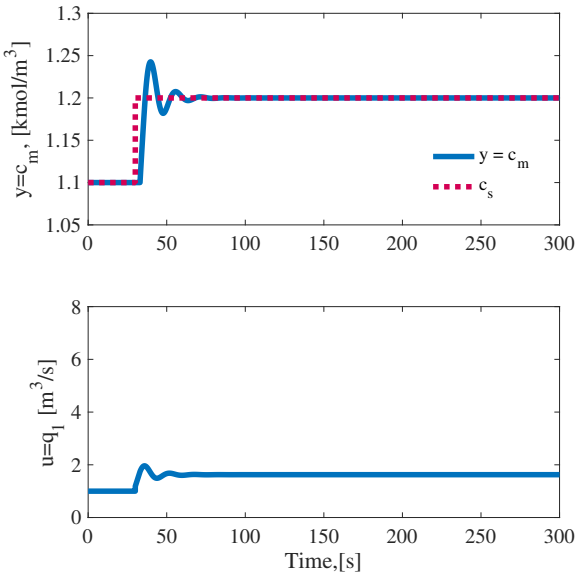

Figure 4.14: Setpoint change from c_s = 1.1 to 1.2 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 3 s, for v_3 as the transformed MV and input gain mismatch of 2.

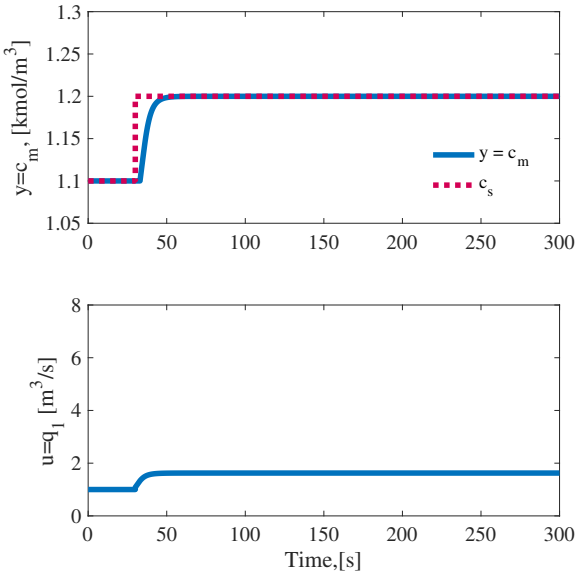

Figure 4.15: Setpoint change from c_s = 1.1 to 1.2 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 3 s, for v_1 as the transformed MV and input gain mismatch of 2.

For large setpoint changes v_3 is still stable and performs the best given that the measurement delay is small. With large amounts of measurement delay, the base case outperforms v_3.

4.3.3 Model error(disturbance gain mismatch)

Input gain mismatch is a form of model error that can be present. It could also be interesting to test the effect of disturbance gain error when doing disturbance rejection. The disturbance gain error is implemented similarly to the input gain error, with the model being perfect at the nominal steady-state, but the deviation from the nominal steady-state being wrong with some gain K_d. The real effect of the disturbance d_r which acts upon the system can be formulated as

$$d_r = d^* + K_d(d - d^*)$$

where d is the measured disturbance, K_d is disturbance gain error, and d^* is the nominal value for the disturbance value.
For this case study, a disturbance gain mismatch of 1.2 is used for all three disturbances.

Disturbance rejection

With a disturbance gain error K_d greater than 1, the feedforward action introduced with the MV transformations should still improve the disturbance rejection capabilities. So for the structures v_2, v_3 and v_5 the disturbance rejection should be quite good. v_3 and v_5 might have problems with large measurement delay, however, based on previous results v_3 should perform quite well. v_1 should overall perform worse but similar to the case with a perfect model. v_5 controlled with a PI-controller should perform worse than v_2 and v_3, and v_5 controlled with a P-controller will lead to a steady-state offset due to the model error and is therefore not a viable control strategy.

Table 4.18: Integrated absolute error, for step disturbance in $d_1 = q_2$ of $9\,\mathrm{m^3\,s^{-1}}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	0.6478	6.3845	0.0535	0.1399	0.0540	2.3112	3.5749†
0.3	0.0648	0.6791	0.0053	0.0108	0.0054	0.0260	0.1666†
0	0.0323	0.3349	0.0027	0.0054	0.0027	0.0065	0.0447†

Table 4.19: Integrated absolute error, for disturbance in $d_1 = q_2$ of $-4.5\,\mathrm{m^3\,s^{-1}}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	2.2239	4.1436	0.1848	0.9260*	0.1872	2.4123	7.1955†
0.3	0.0332	0.4400	0.0135	0.0054	0.0131	0.0130	0.0987†
0	0.0162	0.1788	0.0057	0.0027	0.0056	0.0032	0.0241†

*Stable oscillation
†Steady-state offset

Table 4.20: Integrated absolute error, for disturbance in $d_2 = c_1$ of $1\,\mathrm{kmol\,m^{-3}}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	1.5849	0.4286	0.0635	0.0573	1.5849	2.0556	5.6887†
0.3	0.0364	0.0429	0.0055	0.0051	0.0367	0.0123	0.0938†
0	0.0154	0.0214	0.0027	0.0026	0.0154	0.0031	0.0230†

Table 4.21: Integrated absolute error, for disturbance in $d_3 = c_2$ of $-0.5\,\mathrm{kmol\,m^{-3}}$

Time delay [s]	v_0	v_1	v_2	v_3	v_4	v_{5PI}	v_{5P}
3	3.2400	4.4687	0.3489	0.7239	3.2400	15.9851	33.7447†
0.3	0.3240	0.4497	0.0337	0.0540	0.3240	0.1306	0.9247†
0	0.1619	0.2238	0.0169	0.0270	0.1619	0.0324	0.2349†

As predicted the different structures perform similar to the case with a perfect model. v_2, v_3 and v_5 for the most part performs really well, with near-perfect disturbance rejection. The exceptions being v_3 for a negative disturbance in q_2 with 3 s of measurement delay, where instead of settling the system goes to a state of perpetual stable oscillations, as can be seen in fig. 4.16, and v_5 with 3 s for all disturbances.

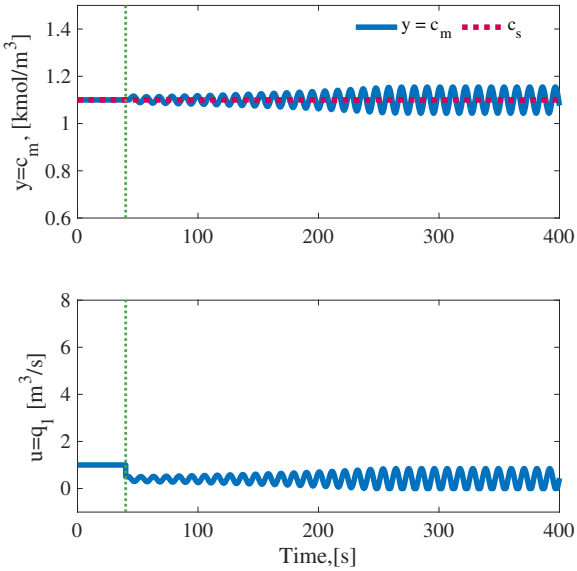

Figure 4.16: Step test in DV q_2 of -4.5 $\mathrm{m^3\,s^{-1}}$ using v_3 as the transformed MV, with measurement time delay θ of 3 s and disturbance gain mismatch of 1.2.

*Stable oscillation
†Steady-state offset

4.4 Analysis and comparison of the transformed system for the mixing example

In this case study, a SISO-mixing-tank example was considered and five different MV transformations were tested and compared. The MV transformation which overall performed the best was v_3 which was was derived with the linear MV transformation for dynamic systems method. With a perfect model, this transformation successfully transformed the non-linear system into a linear one independent of disturbances. Even with output measurement delay, this MV performed well. With model error this MV transformation still performed well when the measurement delay was not too large.

v_5 was derived similarly to v_3 using the same methodology, but selecting the tuning parameter A differently. Selecting A to be zero as done for v_5 resulted in the problem of transforming the non-linear system into an integrating one, which is difficult to control. This coupled with a stronger dependency on the output measurement made this structure overall performing consistently worse than v_3. This structure handled model error and output measurement delay in combination very poorly leading to oscillations.

v_2 was derived with input transformation for static systems methodology and arguably performed the best at disturbance rejection. This transformation does however not linearise the non-linear system and this was notable when setpoint changes were applied as this control structure was quite aggressive, to the point where it failed at performing large setpoint changes with small closed-loop time constants. v_1 is a modification of v_2 witch makes it less aggressive. This is however at the expense of making the disturbance rejection worse than no feedforward action. The transformed system obtained using v_1 does not linearize the system and is comparable to the base case when changing the setpoint.

Based on this case study it is recommendable to use the linear MV transformation for linear systems methodology for non-linear systems with changing setpoints, however, caution should be taken into consideration if there are large amounts of state measurement delay and model uncertainty. For systems where disturbance rejection is more important and setpoint changes are rare and small, the static transformation might be just as well or even better, especially with output measurement delay. However, with sufficient model uncertainty, no of the MV transformations will yield improved disturbance rejection. It should be noted that all disturbances and setpoint changes were subject from the nominal steady-state operating point, and the effect of being subjected to a disturbance or setpoint change at a different steady-state or in transit was not analyzed. Higher closed-loop time constants were also only coupled with higher measurement delay, so the effect of slower tunings of the PI-controllers was not considered by itself.

Table 4.22

Transformed	Coments - summary
General transform, v_1	- This transformation does not linearise system. - Does not give perfect disturbance rejection.
Static transform, v_2	- This transformation is best for static systems. - It gives perfect disturbance rejection. - Can be used for slow closed loop tuning, and large measurement delay.
Linear transform, v_3	- This transformation linearises system. - Gives perfect disturbance rejection. - Measurement delay worsens the performance
Feedback linearization, v_5	- This transformation linearises system. - The transformed system is an integrating system, which introduces control limitations - Gives perfect disturbance rejection. - Measurement delay worsens the performance

Chapter 5

Case study 3: Continuous stirred tank reactor

5.1 Model description

In the two previous cases the new methodology was applied to SISO-systems, and different transformation methodologies were tested and compared. One of the proposed properties of this new MV transformation is that it introduce input output decoupling between the transformed inputs and the outputs. This property will be explored in this case study.

In this case study a continuous stirred tank reactor (CSTR) with a feed stream q_f [$\mathrm{m}^3\,\mathrm{min}^{-1}$] and a product stream q_f [$\mathrm{m}^3\,\mathrm{min}^{-1}$]. The feed stream only consists component A and not component B, and have a concentration c_{Af} [$\mathrm{kmol}\,\mathrm{m}^{-3}$]. In the reactor a second order reaction occurs, where component A reacts to B. The second order reaction coefficient of the process is k [$\mathrm{m}^3\,\mathrm{kmol}^{-1}\,\mathrm{min}^{-1}$] and it is assumed that it can change.The outlet stream has the concentration c_A of component A. Unlike the mixing tank example in chapter 4 where the inventory was assumed perfectly controlled, the inventory V [m^3] must be controlled in this case. It is assumed that the density ρ [$\mathrm{kg}\,\mathrm{m}^{-3}$] is constant.The process has two valves, one for the feed stream q_f and one for the outlet stream q, for simplicity it assumed the streams can both be directly manipulated. A flowsheet of the process is given in fig. 5.1.

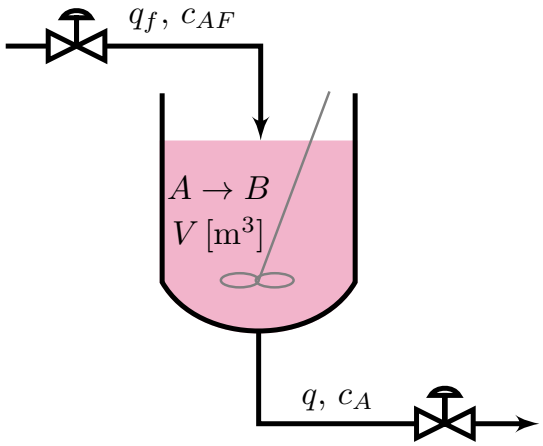

Figure 5.1: Flowsheet of the CSTR, with inflow q with concentration c_{AF}. In the reactor component A reacts to component B. The reactor has an inventory V. Outflow q has concentration c_A.

This system is a MIMO system with two inputs and two outputs, the inputs are the feed stream and the product stream,

$$u = \begin{bmatrix} q_f \\ q \end{bmatrix}$$

the states and outputs in this case are

$$y = x = \begin{bmatrix} V \\ c_A \end{bmatrix}$$

and the disturbances are

$$d = \begin{bmatrix} c_{Af} \\ k \end{bmatrix}$$

The nominal operations are given in table 5.1 bellow.

Table 5.1: Nominal operating values of CSTR

Variable	Value	Unit
V^*	4	m^3
c_A^*	0.05	$\mathrm{kmol\,m}^{-3}$
q_1^*	1	$\mathrm{m}^3\,\mathrm{min}^{-1}$
q_2^*	1	$\mathrm{m}^3\,\mathrm{min}^{-1}$
c_{Af}^*	1	$\mathrm{kmol\,m}^{-3}$
k^*	95	$\mathrm{m}^3\,\mathrm{kmol}^{-1}\,\mathrm{min}^{-1}$

There are equations describing the system which can be used to formulate a model of the system, the second order reaction which takes place in the reactor can be described by the following equation

$$A \rightarrow B; \quad r = kc_A^2 \tag{5.1}$$

where r is the reaction rate of A to B, the total mass balance is

$$\frac{d\rho V}{dt} = \rho(q_f - q) \tag{5.2}$$

and the component mass balance is

$$\frac{dc_A V}{dt} = c_{Af} q_f - c_A q - Vr \tag{5.3}$$

Constant density (ρ) is assumed so the total mass balance can be simplified to

$$\frac{dV}{dt} = q_f - q \tag{5.4}$$

the component mass balance is rewritten as

$$c_A \frac{dV}{dt} + V \frac{dc_A}{dt} = c_{Af} q_f - c_A q - Vr \tag{5.5}$$

inserting eqs. (5.1) and (5.4) into this equation the following expression is achieved

$$c_A(q_f - q) + V \frac{dc_A}{dt} = c_{Af} q_f - c_A q - V k c_A^2 \tag{5.6}$$

which when solved for dc_A/dt yields the model equation

$$\frac{dc_A}{dt} = \frac{q_f}{V}(c_{Af} - c_A) - k c_A^2 \tag{5.7}$$

This yields a model which is both non-linear and coupled. The full model is

$$\frac{dy}{dt} = \begin{bmatrix} f_1(x,u,d) \\ f_2(x,u,d) \end{bmatrix} = \begin{bmatrix} q_f - q \\ \frac{q_f}{V}(c_{Af} - c_A) - k c_A^2 \end{bmatrix} \tag{5.8}$$

5.2 MV transformations

In the two previous case studies this new MV transformation methodology was applied to SISO systems, and the linear transformation strategy outlined in section 2.3.1 was found to be the best at linearizing the non-linear system. In this case case study a MIMO system is considered and the decoupling capabilities of this new methodology will be studied and compared to input-output decoupling of a system linearized around the nominal operating point. The MV transformation can create a linear mapping between the outputs and the transformed inputs

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{5.9}$$

with the linear transformation strategy. This is accomplished by selecting the tuning parameter to be a diagonal matrix A_d. To retain the nominal dynamics of the system A_d is

selected as

$$\begin{aligned} A_d &= \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix} = diag(\frac{\partial f}{\partial x}\Big|_*) = \begin{bmatrix} \frac{df_1}{dx_1} & 0 \\ 0 & \frac{df_2}{dx_2} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & -(\frac{q_f^*}{V^*} + 2k^* c_A^*) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -9.750 \end{bmatrix} \end{aligned} \tag{5.10}$$

The transformed MVs are then

$$v_1 = f_1(x,u,d) - a_{11}V = q_f - q \tag{5.11}$$

$$v_2 = f_2(x,u,d) - a_{22}c_A = \frac{q_f}{V}c_{Af} + (\frac{q_f^*}{V^*} - \frac{q_f}{V})c_A + (2k^* c_A^* - kc_A)c_A \tag{5.12}$$

The transformed system obtain by this MV transformation is

$$\frac{dV}{dt} = v_1 \tag{5.13}$$

$$\frac{dc_A}{dt} = a_{22}c_A + v_2 \tag{5.14}$$

This transformed system is both linear and decoupled. It should be noted that the mapping from v_1 to V is that of an integrating process despite not using the feedback linearization variation of the linear MV transformation method. This is due to the mapping from the original inputs u to V not considering the coupling, were that of an integrating process.

5.2.1 MV transformation calculation block

A property which must be fulfilled for this MV transformation to be realisable is that eqs. (5.11) and (5.12) which can be written on the form

$$v = \alpha(x,d) + \beta(a,d)u \tag{5.15}$$

where

$$\alpha(x,d) = \begin{bmatrix} 0 \\ -kc_A^2 - a_{22}c_A \end{bmatrix} \tag{5.16}$$

and

$$\beta(x,d) = \begin{bmatrix} 1 & -1 \\ \frac{c_{Af} - c_A}{V} & 0 \end{bmatrix} \tag{5.17}$$

are invertible with respect to u

$$u = \beta(a,d)^{-1}(v - \alpha(x,d)) \tag{5.18}$$

This requires that $\beta(x,d)$ is non-singular in the operational range of the process. This is the case when the determinant of $\beta(x,d)$ is non-zero. The determinant of $\beta(x,d)$ is

$$D(\beta(x,d)) = \frac{c_{Af} - c_A}{V} \tag{5.19}$$

which is non-zero when $c_{Af} \neq c_A$, or when $V \neq \pm\infty$. In this process a constraint is put on the inventory V that it cannot be negative and it is unlikely that it will reach infinity if controlled well. It is similarly not likely that c_A will approach c_{Af} due to the reaction taking place in the reactor, however if the setpoint of c_A was raised sufficiently and a disturbance in the feed flow lowered c_{Af} sufficiently it is imaginable that a state of singularity could be achieved. However this is not considered likely in this case study, and eq. (5.15) is considered invertible for the operational range in this case study. The inverse of $\beta(x, d)$ is

$$\beta(x,d)^{-1} = \begin{bmatrix} 0 & \frac{V}{c_{Af}-c_A} \\ -1 & \frac{V}{c_{Af}-c_A} \end{bmatrix} \tag{5.20}$$

u is then calculated as

$$u = \begin{bmatrix} 0 & \frac{V}{c_{Af}-c_A} \\ -1 & \frac{V}{c_{Af}-c_A} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 + kc_A^2 + a_{22}c_A \end{bmatrix} \tag{5.21}$$

An interesting and unfortunate property of the MV transformation is that if the inventory V becomes zero, that is the reactor is completely emptied, the inlet feed q_f will be zero as well independent of what v is. This means if the reactor is fully emptied at one point during operation this control structure will make it so that it will not be refilled.

5.2.2 Analysis of transformed system

In this case study only one transformed MV was designed. It was designed with the linear MV transformation technique from section 2.3.1, which was shown to work well for a SISO-system in chapter 4. The transformation transforms the non-linear system into a linear and decoupled system, so it is expected this structure will perform well over a large operation range giving perfect input-output decoupling over the entire range. As explored earlier this MV transformation performs worse when subject to model uncertainty for a SISO-system. This is also expected for a MIMO-system, but how it effects the decoupling is unknown.

A type of model error which was considered in the SISO mixing tanks example was output measurement uncertainty, in the form of measurement delay. This type of model error introduces some difficulties for feedback control in general as it is difficult to control something based on feedback if the feedback is incorrect. For the new MV transformation methodology transformations dependent on output measurements are proposed. The physical inputs are calculated based on measurement of outputs and disturbances, so the physical inputs are sensitive to the output measurements uncertainty as observed in the SISO mixing tank example.

For a MIMO system it is of interest how measurement delay would affect the decoupling between the outputs and the transformed inputs. In the transformed system $y_1 = V$ are fully decoupled from $y_2 = c_A$ and v_2 and only dependent on the transformed input v_1. This transformed input is not dependent on measurements of the outputs and should not be affected by measurement uncertainty. V should therefore remain decoupled even with

measurement uncertainty. $y_2 = c_A$ is only dependent on v_2. v_2 is a function of both outputs, so measurement delay in either output variable should affect c_A. Measurement uncertainty in V will make it so that c_A is dependent on the other output, and perfect decoupling will no longer be present. This means if measurement uncertainty is only present in c_A the transformed system will still be fully decoupled. It is more likely that measurement uncertainty is present in concentration measurements than inventory measurements.

If the model uncertainty is structural such that the interaction between the outputs and inputs describing the change in the outputs are different from the model used to derive the transformed MVs, perfect decoupling is not expected. One type of structural model uncertainty is uncertainty in the gains from the inputs to the outputs. In this system there are two inputs, so an input gain error could be applied to one input or both inputs and with varying magnitude. Many different combinations of systematic model error could be tested. To evaluate the effect of an input gain error the model equations in eq. (5.8) will be evaluated,

$$\frac{dV}{dt} = q_f - q$$
$$\frac{dc_A}{dt} = \frac{q_f}{V}(c_{Af} - c_A) - kc_A^2$$

The change in V is the difference between the two physical inputs. If a gain error of the same magnitude is present for both inputs they will cancel each other out and V will still be decoupled from the other output c_A and the transformed input v_2. If the both inputs have gain errors with differing magnitude or just one of the inputs have a gain error, perfect decoupling is no longer present and V is dependent on the transformed input v_2.

The change in c_A with respect to time is only dependent on one of the inputs q_f. A gain error in q will therefore not affect c_A. So only a gain error in q_f will make c_A dependent on v_1. This means that only if there is a gain error in q_f which is different from the gain error in q, both outputs will be dependent on both transformed inputs.

5.3 Linear model simplification and decoupling

Non-linear process can also be controlled using linear control theory by linearizing the models around the operating point where the process operated around, using the Jacobian. Unlike the proposed new MV transformation method, this linearization is only valid for a small operating range around the nominal operation point. The size of this valid operational range is dependent on how non-linear the original system is. The linearized model equation is on the form,

$$\frac{d\Delta x}{dt} = A\Delta x + B\Delta u + D\Delta d \tag{5.22}$$

Where $A = df/dx|_*$, $B = df/du|_*$ and $C = df/dd|_*$. The deviation variables are,

- $\Delta x = x - x^*$
- $\Delta u = u - u^*$

- $\Delta d = d - d^*$

$$\begin{aligned} A &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dx_1} & \frac{df_1}{dx_2} \\ \frac{df_2}{dx_1} & \frac{df_2}{dx_2} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ -\frac{q_f^*}{V^{*2}}(c_{Af}^* - c_A^*) & -(\frac{q_f^*}{V^*} + 2k^* c_A^*) \end{bmatrix} \approx \begin{bmatrix} 0 & 0 \\ -0.059 & -9.750 \end{bmatrix} \end{aligned} \tag{5.23}$$

and

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} \frac{df_1}{du_1} & \frac{df_1}{du_2} \\ \frac{df_2}{du_1} & \frac{df_2}{du_2} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ \frac{c_{Af}^* - c_A^*}{V^*} & 0 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0.2375 & 0 \end{bmatrix} \tag{5.24}$$

and

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dd_1} & \frac{df_1}{dd_2} \\ \frac{df_2}{dd_1} & \frac{df_2}{dd_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \frac{q_f^*}{V^*} & -c_A^{*2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0.25 & -0.0025 \end{bmatrix} \tag{5.25}$$

Using a Laplace transformation the linearized model equations can be transformed from the time domain to the frequency domain, and the transfer function G from u to x, and transfer function G_d from d to x can be found. Dropping the deviation notation the following equation is obtained,

$$sx = Ax + Bu + Cd \tag{5.26}$$

the transfer function is then found by solving the equation for x,

$$x = (Is - A)^{-1}Bu + (Is - A)^{-1}Cu \tag{5.27}$$

where the inverse of $Is - A$ is

$$(Is - A)^{-1} = \begin{bmatrix} \frac{1}{s} & 0 \\ -\frac{a_{21}}{s(s-a_{22})} & \frac{1}{s-a_{22}} \end{bmatrix} \tag{5.28}$$

The transfer function G is

$$G = (Is - A)^{-1}B = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} \frac{1}{s} & -\frac{1}{s} \\ \frac{b_{21}s - a_{21}}{s(s-a_{22})} & -\frac{a_{21}}{s(s-a_{22})} \end{bmatrix} \tag{5.29}$$

and the transfer function G_d is

$$G_d = (Is - A)^{-1}C = \begin{bmatrix} g_{d11} & g_{d12} \\ g_{d21} & g_{d22} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \frac{c_{21}}{s-a_{22}} & \frac{c_{22}}{s-a_{22}} \end{bmatrix} \tag{5.30}$$

The transfer function g_{21} can be written on the more recognisable form

$$g_{21} = k_{21}\frac{Ts + 1}{s(\tau_{21}s + 1)}$$

where

$$k_{21} = -\frac{a_{21}}{a_{22}} = -6.090 \times 10^{-3}$$

,

$$T = \frac{b_{21}}{a_{21}} = -4$$

and

$$\tau_{21} = -\frac{1}{a_{22}} = 0.103$$

The transfer function g_{22} can also be written on a more recognisable form

$$g_{22} = k_{22}\frac{1}{s(\tau_{22}s + 1)}$$

$$k_{22} = -k_{21} = \frac{a_{21}}{a_{22}} = 6.090 \times 10^{-3}$$

and

$$\tau_{22} = \tau_{21} = -\frac{1}{a_{22}} = 0.103$$

With the transfer function G two decoupling blocks T_{12} and T_{21} can be constructed. Depending on the pairings of inputs to outputs two sets of decoupling blocks can be constructed. Without decoupling the RGA is used to make the optimal pairings which gives stable operations. When applying decoupling this is not a concern so both pairing choices will be evaluated, and the best pairing choice will be chosen. If the pairings are done on the diagonal, pairing y_1 with u_1, and paring y_2 with u_2,

$$T_{12} = -\frac{g_{12}}{g_{11}} = 1 \tag{5.31}$$

$$T_{21} = -\frac{g_{21}}{g_{22}} = Ts + 1 = -4s + 1 \tag{5.32}$$

The decoupling block T_{21} is not realisable as it has more zeros than poles. This can be fixed by introducing an additional pole, with a large absolute value.

$$T_{21} = \frac{Ts + 1}{\tau s + 1} = \frac{-4s + 1}{\tau s + 1} \tag{5.33}$$

The alternative pairing would be to pair y_1 with u_2, and pairing y_2 with u_1, which would yield the decoupling blocks

$$T_{12} = -\frac{g_{22}}{g_{21}} = \frac{1}{Ts + 1} = \frac{1}{-4s + 1} \tag{5.34}$$

$$T_{21} = -\frac{g_{11}}{g_{12}} = 1 \tag{5.35}$$

Unlike the diagonal input output pairing the of diagonal pairing gives decoupling blocks which can be implemented without modifications. This makes the off diagonal pairing seem like the better pairing, but when evaluating the transfer function G that is no longer the case. This is due to the right hand plane zero, which gives an initial inverse response.

Because of this the of-diagonal pairing will not be considered. The decoupling blocks in eqs. (5.31) and (5.33) will be used in this case study with τ in the T_{21} block being selected to be $4 * 10^{-4}$.

These decoupling blocks also requires an accurate model to give proper decoupling. These blocks are however not dependent on measurement uncertainty, unlike the decoupling from the MV transformation. They are however dependent on the structure of the model being correct, so similarly to how input gain errors will affect the decoupling in the transformed system, the effect of input gain error on the decoupling blocks derived from the linearized model will be discussed.

For a input gain error k_i on input i the transfer function from input i to output j is

$$g_{ji}^{'} = k_i g_{ji} \tag{5.36}$$

where $g_{ji}^{'}$ is the real transfer function and g_{ji} is the transfer function used to derive the decoupling blocks and controller tunings. The decoupling blocks $T_{12}^{'}$ and $T_{21}^{'}$ which would give the correct decoupling would be

$$T_{12}^{'} = \frac{k_2}{k_1}\frac{g_{12}}{g_{11}} = \frac{k_2}{k_1}T_{12} \tag{5.37}$$

$$T_{21}^{'} = \frac{k_1}{k_2}\frac{g_{21}}{g_{22}} = \frac{k_1}{k_2}T_{21} \tag{5.38}$$

$$\tag{5.39}$$

Both decoupling blocks give a wrong decoupling with a gain being the fraction between the gain errors for the two inputs. The special case where the input gain errors are the same for both inputs the decoupling would still be correct. Since the decoupling blocks are derived from a linearized model, when operating far away from the nominal operation point perfect decoupling might not be possible.

5.4 Controller tuning

Similar to the previous cases PI-controllers will be used to control the system, and the PI-controllers are tuned using the SIMC tuning rules[17]. For the transformed case the tunings are simple to derive. The transformed model equations given in eqs. (5.13) and (5.14) are transformed to the frequency domain and the transfer functions from v_1 to y_1, and v_2 to y_2 are found. These are

$$\frac{y_1}{v_1}(s) = \frac{1}{s} \tag{5.40}$$

$$\frac{y_2}{v_2}(s) = \frac{\tau_{a22}}{\tau_{a22}s + 1}; \quad \tau_{a22} = -\frac{1}{a_{22}} \tag{5.41}$$

Denoting the controller gain and integral time for the PI-controller controlling $y_1 = V$ as K_{c1} and τ_{I1}, and similarly for the PI-controller controlling $y_2 = c_A$ as K_{c2} and τ_{I2}, the

controller tunings are,

$$K_{c1} = \frac{1}{\tau_{c1} + \theta_1} \tag{5.42}$$

$$\tau_{I1} = 4(\tau_{c1} + \theta_1) \tag{5.43}$$

$$K_{c2} = \frac{1}{\tau_{c2} + \theta_2} \tag{5.44}$$

$$\tau_{I2} = \min(\tau_{a22}, 4(\tau_{c2} + \theta_2)) \tag{5.45}$$

where τ_{c1} is the closed loop time constant for $y_1 = V$, θ_1 is the delay in the closed loop for $y_1 = V$, τ_{c2} is the closed loop time constant for $y_2 = c_A$ and θ_2 is the delay in the closed loop for $y_2 = c_A$.

For the case where the model is linearized around the nominal operating point the input-output pairing must be selected. As previously discussed when designing input-output decoupling blocks, the off-diagonal pairing will not be selected do to the right plane zero in g_{21}. The input-output pairings are then $y_1 = V$ with $u_1 = q_f$, and $y_1 = c_A$ with $u_2 = q$. It should be noted that the transfer function g_{22} from u_2 to y_2 is neither that of an integrating process or a first order process, and the SIMC tuning rules can only be applied to these two systems. To apply the SIMC tuning rules g_{22} will be approximated as an integrating process with time delay by using Skogestads half rule[17]. The approximated integrating process is

$$g_{22half-rule} = \frac{k_{22}}{s} e^{-\frac{\tau_{22}}{2}} \tag{5.46}$$

Using this transfer function and g_{11} the controller tunings are

$$K_{c1} = \frac{1}{\tau_{c1} + \theta_1} \tag{5.47}$$

$$\tau_{I1} = 4(\tau_{c1} + \theta_1) \tag{5.48}$$

$$K_{c2} = \frac{k_{22}}{\tau_{c2} + (\theta_2 + \frac{\tau_{22}}{2})} \tag{5.49}$$

$$\tau_{I2} = 4(\tau_{c2} + (\theta_2 + \frac{\tau_{22}}{2})) \tag{5.50}$$

The closed loop time constants τ_{c1} and τ_{c2} will be decided later in the case study.

5.5 Case study

In this case study, the transformed system will be compared with the system linearized around the nominal operating point. It will be compared with both the linearized system without decoupling and a system with decoupling. First, the system will be tested for both positive and negative setpoint changes of both output values without model error. Both small and large setpoint changes will be considered to evaluate how well the transformed system performs compared to the linearized system both with and without decoupling. Model error will be added later to test how robust the new methodology is to model errors in a MIMO system. Similar to the SISO mixing tank case in chapter 4 two types of model error will be considered, these two being measurement uncertainty in the form of measurement delay, and input gain uncertainty.

5.5.1 Perfect model

For the case with a perfect model the closed-loop tuning constants will both be selected to be $\tau_{c1} = \tau_{c2} = \tau_r = 4\text{s}$. As previously mentioned both large and small, positive, and negative setpoint changes will be evaluated for both output variables V and c_A. A small 10% positive and negative setpoint change will be considered for both outputs, alongside a large positive 100% setpoint change and a large negative setpoint change of 50%. The performance measurement used to evaluate the setpoint change responses are IAE, and the IAE for both output variables will be considered.

It is expected that the transformed system will have perfect decoupling, even when operating far away from the nominal operating point. This might be the case for the linearized system, as the decoupling blocks are derived from a model linearization which might not be valid when operating far away from the nominal operating point. The performances for the Transformed system, linearized system and linearized system plus decoupling for a setpoint changes in $y_1 = V$ are given in tables 5.2 and 5.3.

Table 5.2: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_1 = V$ with a perfect model.

y_{1s}	4.4m^3	8m^3	3.6m^3	2.0m^3
Transformed	2.3544	23.5442	2.3544	11.7721
Linearized	6.3857	63.4469	6.3907	31.9842
Decoupled	2.3544	23.5443	2.3544	11.7721

Table 5.3: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_1 = V$ with a perfect model.

y_{1s}	4.4m^3	8m^3	3.6m^3	2.0m^3
Transformed	0.0000	0.0000	0.0000	0.0000
Linearized	0.0467	0.4583	0.0469	0.2367
Decoupled	0.0000	0.0002	0.0000	0.0001

As expected with no model uncertainty the transformed MV introduce perfect decoupling, so when the setpoint of $y_1 = V$ is changed c_A is unaffected, as can be seen in table 5.3. The linearized system with decoupling is almost as good given near perfect decoupling, with only a very small change in c_A when changing the setpoint of V far away from the nominal operating point. Both methods of decoupling greatly improves the performance compared to the linearized case without decoupling, as the system is highly coupled. This can be seen in fig. 5.2 for a negative setpoint change from $4\,\text{m}^3$ to $2\,\text{m}^3$.

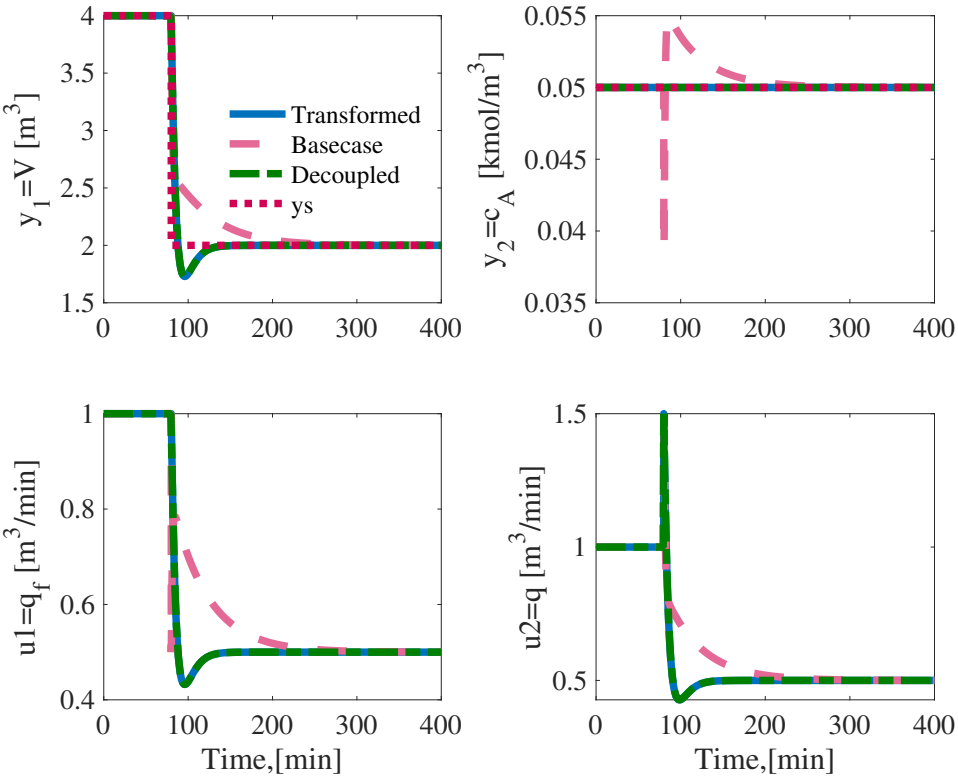

Figure 5.2: Setpoint change in output $y_1 = V$ from $4\,\text{m}^3$ to $2\,\text{m}^3$. The setpoint change occurs at time = 80 s

The performances for setpoint changes in c_A are given in tables 5.4 and 5.5.

Table 5.4: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_2 = c_A$ with a perfect model.

y_{2s}	$0.055\text{kmol}\,\text{m}^{-3}$	$0.1\text{kmol}\,\text{m}^{-3}$	$0.045\text{kmol}\,\text{m}^{-3}$	$0.025\text{kmol}\,\text{m}^{-3}$
Transformed	0.0000	0.0000	0.0000	0.0000
Linearized	13.8068	193.3166	12.4202	48.4090
Decoupled	0.0000	0.0000	0.0000	0.0000

Table 5.5: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_2 = c_A$ with a perfect model.

y_{2s}	$0.055\text{kmol}\,\text{m}^{-3}$	$0.1\text{kmol}\,\text{m}^{-3}$	$0.045\text{kmol}\,\text{m}^{-3}$	$0.025\text{kmol}\,\text{m}^{-3}$
Transformed	0.0200	0.2000	0.0200	0.1000
Linearized	0.0929	1.3000	0.0836	0.3258
Decoupled	0.0324	0.4567	0.0300	0.1261

Similarly as when doing setpoint changes in V when changing the setpoint of c_A V is fully unaffected for the transformed system, as was expected. Interestingly the transformed system gives better control of $y_2 = c_A$ for all the setpoint changes. This is due to the transformed system being a first order system, while the linearized system is that of an integrating system. It can also be seen that the transformed system is consistent in its performance independent of the setpoint change, as the performance scales linearly with the change in the setpoint. This is not the case for the linearized system with or without decoupling. This indicates that the transformation successfully transforms the non-linear system into a linear one. The setpoint change for $y_2 = c_A$ from $0.05\text{kmol}\,\text{m}^{-3}$ to $0.1\text{kmol}\,\text{m}^{-3}$ is shown in fig. 5.3.

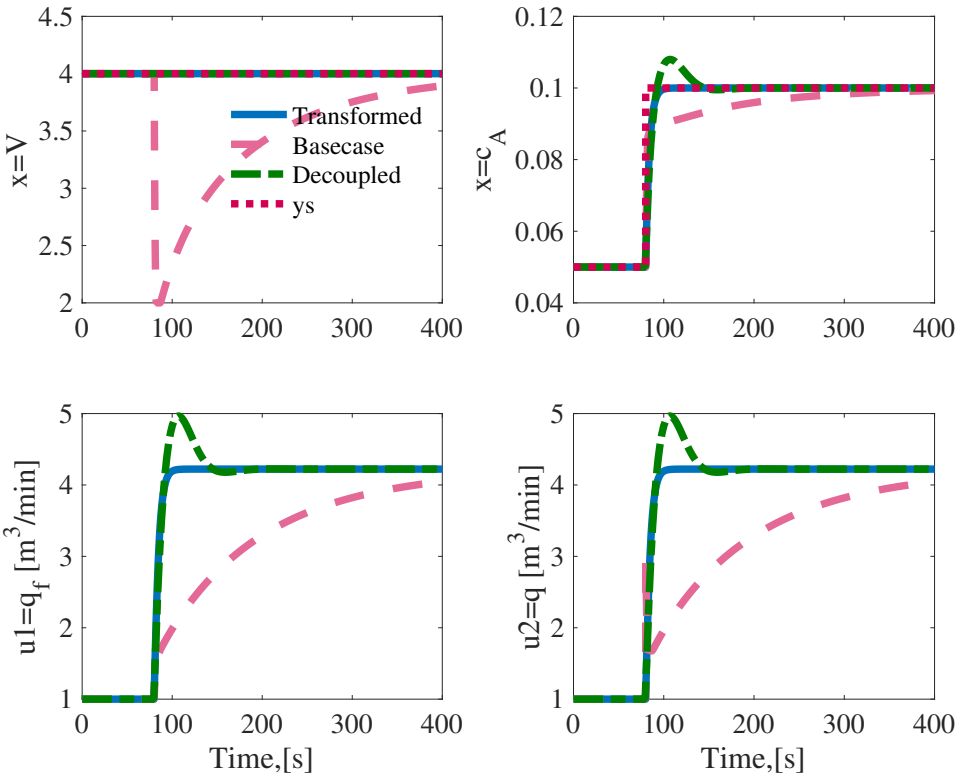

Figure 5.3: Setpoint change in output $y_2 = c_A$ from$0.05\text{kmol}\,\text{m}^{-3}$ to $0.1\text{kmol}\,\text{m}^{-3}$. The setpoint change occurs at time = 80 s

5.5.2 Measurement uncertainty

For systems with no model uncertainty, the new methodology introduces perfect input-output decoupling as predicted and perfectly linearizes the system. A system with a perfect model and no uncertainties is an idealized scenario, so this MV transformation should also be tested for how well it performs with model uncertainty.

The first type of model error which will be considered is output measurement delay. In this case only measurement delay for the second output $y_2 = c_A$ is considered. Measurement delay similar to the dynamics of the system will be considered, so the output measurement delay is $\theta_m = 4\text{s}$. The performances are given in tables 5.6 to 5.9

Table 5.6: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_1 = V$ with measurement delay $\theta_m = 4$s for $y_2 = c_A$.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	2.3544	23.5442	2.3544	11.7721
Linearized	6.5202	64.0044	6.5384	31.9931
Decoupled	2.3542	23.5437	2.3542	11.7719

Table 5.7: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_1 = V$ with measurement delay $\theta_m = 4$s for $y_2 = c_A$.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	0.0000	0.0000	0.0000	0.0000
Linearized	0.1695	1.6453	0.1704	0.8601
Decoupled	0.0000	0.0004	0.0001	0.0005

Table 5.8: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_2 = c_A$ with measurement delay $\theta_m = 4$s for $y_2 = c_A$.

y_{2s}	$0.055\mathrm{kmol\,m}^{-3}$	$0.1\mathrm{kmol\,m}^{-3}$	$0.045\mathrm{kmol\,m}^{-3}$	$0.025\mathrm{kmol\,m}^{-3}$
Transformed	0.0000	0.0000	0.0000	0.0000
Linearized	13.7906	195.4687	12.4126	48.4076
Decoupled	0.0000	0.0001	0.0000	0.0001

Table 5.9: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_2 = c_A$ with measurement delay $\theta_m = 4$s for $y_2 = c_A$.

y_{2s}	$0.055\mathrm{kmol\,m}^{-3}$	$0.1\mathrm{kmol\,m}^{-3}$	$0.045\mathrm{kmol\,m}^{-3}$	$0.025\mathrm{kmol\,m}^{-3}$
Transformed	0.0451	0.6162	0.0416	8.6500*
Linearized	0.3531	4.9855	0.3179	1.2404
Decoupled	0.0821	1.0686	0.0789	0.6067

With output measurement delay in c_A the MV transformation perfectly decouples the system for both outputs as seen in tables 5.7 and 5.8 as was expected, just as the linearized model with decoupling. Since the measurement uncertainty is only in c_A and perfect decoupling is achieved, when doing setpoint changes in V the performance is identical to the case with no measurement uncertainty for the transformed system and the linearized system with decoupling. For the linearized system without decoupling due to the large degree of coupling in the system the performance is worsened as can be seen in fig. 5.4

*Stable oscillations

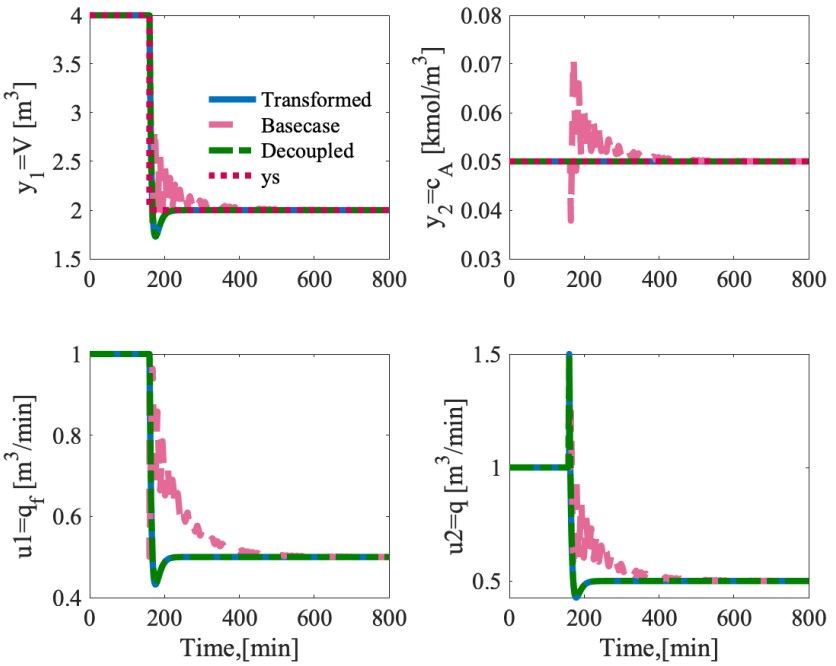

Figure 5.4: Setpoint change in output $y_1 = V$ from $4\,\mathrm{m}^3$ to $2\,\mathrm{m}^3$, with a measurement delay of 4s of output $y_2 = c_A$. The setpoint change occurs at time = 160 s.

As can be seen in the figure the decoupling greatly increases the performance, due to the process being highly coupled, even with measurement delay.

As explored in a previous case study measurement delay worsens the performance of an MV dependent on output measurement. This can be seen here in table 5.9, where the IAE for different setpoint changes are worse compared to the case without measurement uncertainty. The transformed system outperforms the linearized systems, both with and without decoupling for three of setpoint changes. It is only the large negative setpoint change from $c_A = 0.05\ \mathrm{kmol\,m^{-3}}$ to $c_A = 0.025\ \mathrm{kmol\,m^{-3}}$ where the transformed system does not perform better. In this case, the system goes into a state of stable persisting oscillations. This was one of the drawbacks of the linear MV transformation strategy as explored in chapter 4. This can be seen in fig. 5.5.

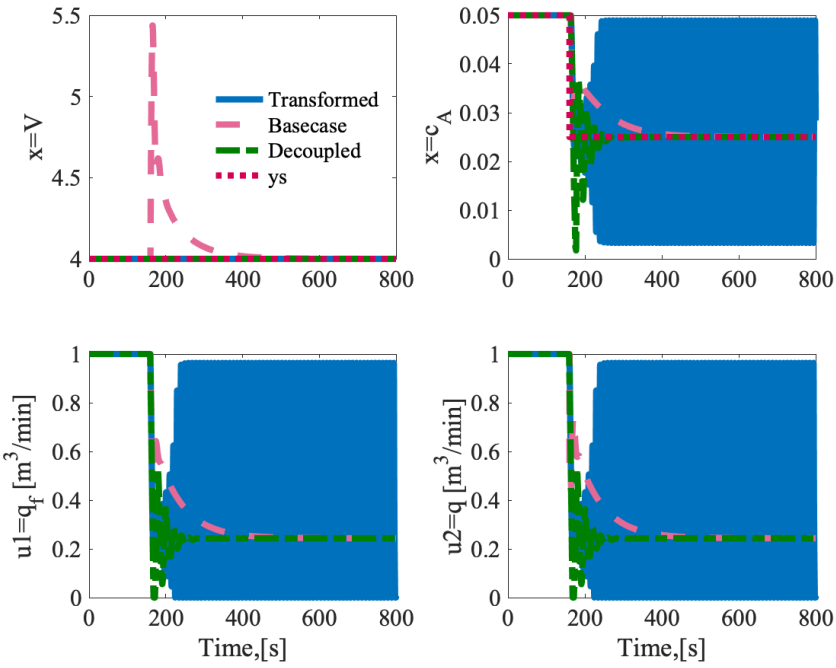

Figure 5.5: Setpoint change in output $y_2 = c_A$ from$0.05\text{kmol}\,\text{m}^{-3}$ to $0.025\text{kmol}\,\text{m}^{-3}$, with a measurement delay of 4s of output $y_2 = c_A$. The setpoint change occurs at time = 160 s

5.5.3 Structural model uncertainty

Perfect decoupling was still possible as expected with output measurement error in only c_A. The problems of MV transformations dependent on output measurements for systems with measurement delay was discussed in chapter 4, and these problems were apparent for this system as well. Measurement uncertainty is not the only model uncertainty that can be present. In this section, structural model uncertainty will be considered.

Structural model error can appear in many different forms but in this case study, input gain error will be considered. The input gain error is implemented the same way it was implemented in section 4.3.2. Different combinations of input errors can be applied to a MIMO system and several of them will be explored.

First a small input gain error of 1.1 in both inputs are considered and the performances are given in tables 5.10 to 5.13.

Table 5.10: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 1.1 in both inputs.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	2.1740	21.7936	2.1740	10.8700
Linearized	5.8095	57.8231	5.8126	29.0818
Decoupled	2.1737	21.7940	2.1737	10.8684

Table 5.11: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 1.1 in both inputs.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	0.0010	0.0048	0.0012	0.0132
Linearized	0.0429	0.4208	0.0431	0.2176
Decoupled	0.0000	0.0003	0.0001	0.0001

Table 5.12: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 1.1 in both inputs.

y_{2s}	$0.055\mathrm{kmol\,m}^{-3}$	$0.1\mathrm{kmol\,m}^{-3}$	$0.045\mathrm{kmol\,m}^{-3}$	$0.025\mathrm{kmol\,m}^{-3}$
Transformed	0.0000	0.0000	0.0000	0.0000
Linearized	12.5642	178.0695	11.2946	44.0086
Decoupled	0.0001	0.0000	0.0000	0.0000

Table 5.13: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 1.1 in both inputs.

y_{2s}	$0.055\mathrm{kmol\,m}^{-3}$	$0.1\mathrm{kmol\,m}^{-3}$	$0.045\mathrm{kmol\,m}^{-3}$	$0.025\mathrm{kmol\,m}^{-3}$
Transformed	0.0181	0.1730	0.0183	0.0931
Linearized	0.0846	1.1976	0.0760	0.2962
Decoupled	0.0299	0.4170	0.0277	0.1161

In section 5.2.2 the effect of different gain errors were explored. For systems where the input gain error was the same for both inputs it was expected that V would be fully decoupled from c_A and v_2, which is the case as can be seen in table 5.12.

It was not expected that c_A would be fully decoupled from the other outputs and transformed inputs. This is the case as seen in table 5.11. There is some decoupling, however, so the performance for the transformed MV still performs better than the linearized case

without decoupling, however, the linearized case with decoupling has near-perfect input-output decoupling being better than the transformed system as is expected. This can be seen in fig. 5.6

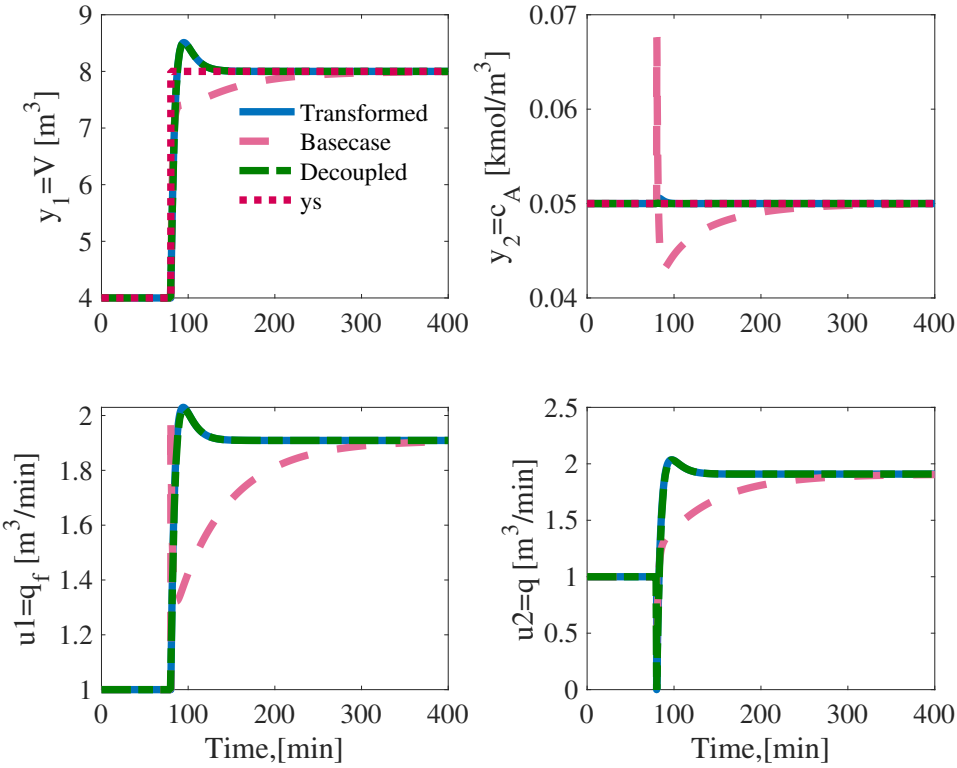

Figure 5.6: Setpoint change in output $y_1 = V$ from $4\mathrm{m}^3$ to $8\mathrm{m}^3$, with an input gain error in both inputs of 1.1. The setpoint change occurs at time = 80 s

Previously measurement delay was considered for a perfect model. Measurement delay did not affect the decoupling so the performances were similar to the case without measurement delay. With model uncertainty in the form of input gain mismatch in both inputs the MV transformation no longer gave perfect input-output decoupling for c_A. With c_A no longer being fully decoupled, it is dependent v_1 and will change when the setpoint of V is changed. c_A must then be corrected by feedback. The effect of measurement delay of c_A on the calculation block for the transformed system might make the system go unstable and should be looked into. So the effect of measurement delay and input gain error will be considered for setpoint changes in V.

Setpoint changes in c_A will not be considered as V was perfectly decoupled from the transformed input v_2. While the effect of measurement error and input gain error could be interesting to discuss, this is already considered for a SISO system in chapter 4. 4 s of output measurement delay is used. The performances are given in tables 5.14 and 5.15.

Table 5.14: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 1.1 in both inputs, and measurement delay of $4\,\mathrm{s}$ in $y_2 = c_A$.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	2.1738	21.7925	2.1739	10.8694
Linearized	6.0187	59.1692	6.0379	360.9507*
Decoupled	2.1738	21.7906	2.1738	10.8690

Table 5.15: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 1.1 in both inputs, and measurement delay of $4\,\mathrm{s}$ in $y_2 = c_A$.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	0.0021	0.0109	0.0027	0.0295
Linearized	0.1553	1.5120	0.1560	9.8372*
Decoupled	0.0000	0.0006	0.0001	0.0006

Just as the case with only measurement uncertainty, measurement uncertainty does not change performance drastically for the transformed system with a small input gain mismatch in both inputs. In fact it is the linear case without decoupling for which the performance drastically worsens when the setpoint of V is changed from $4\,\mathrm{m}^3$ to $2\,\mathrm{m}^3$ as can be seen in fig. 5.7

*Stable oscillations

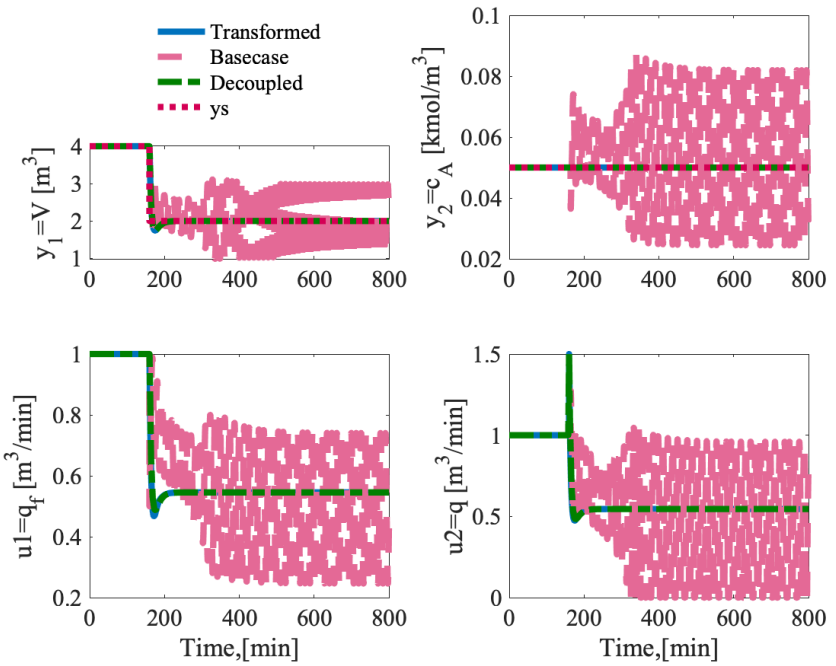

Figure 5.7: Setpoint change in output $y_1 = V$ from 4m^3 to 2m^3, with an input gain error in both inputs of 1.1 and measurement delay of c_A of 4 s. The setpoint change occurs at time = 160 s

Large gain error for both inputs

The MV transformation gave good control even with a small gain error in the two inputs, so a large gain error should be evaluated as well to test how robust this control structure is too large model uncertainty. An input gain error of 2 in both inputs will be evaluated. As shown for a small gain error in both inputs, the output V is fully decoupled from the transformed input v_2, so setpoint changes in the output c_A will not be considered. With a perfect decoupling of V, it will be perfectly controlled, and since the model equation of c_A is independent of the input q, the model of c_A will be functionally a SISO-system. The effect of a large input gain error was considered in section 4.3.2 in chapter 4 for SISO-systems, and while the process is different the same problems should be present as in that case study.

First a large input gain error with no measurement uncertainty will be considered. The performances are given in tables 5.16 and 5.17

Table 5.16: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 2 in both inputs.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	1.3012	14.5941	1.3012	6.5059
Linearized	3.1997	31.9828	3.1999	15.9997
Decoupled	1.3010	14.7879	1.3010	6.5048

Table 5.17: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 2 in both inputs.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	0.0050	0.0262	0.0063	0.0643
Linearized	0.0253	0.2526	0.0255	0.1301
Decoupled	0.0000	0.0105	0.0001	0.0001

Even with large gain errors in both input variables the transformed system is still stable and gives some decoupling from c_A to the transformed inputs, performing better than the linearized system without decoupling. The linearized system with decoupling still gives better decoupling than the MV transformation, however for the large positive setpoint change from $V = 4\,\mathrm{m}^3$ to $8\,\mathrm{m}^3$ no longer gives perfect input-output decoupling as can be seen in fig. 5.8.

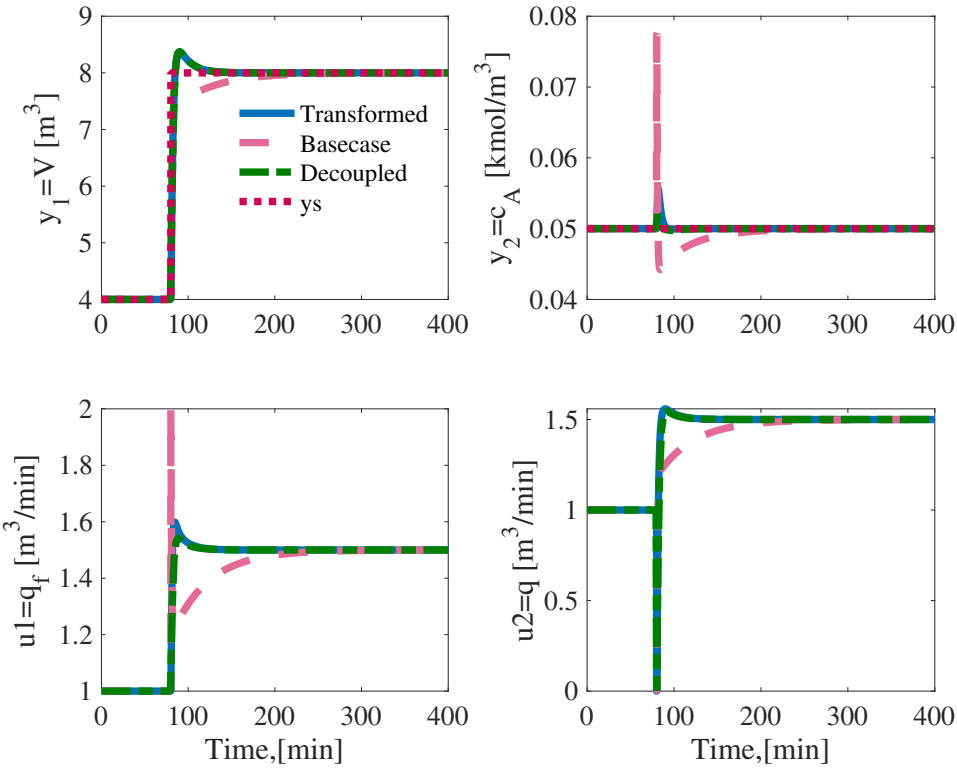

Figure 5.8: Setpoint change in output $y_1 = V$ from 4m^3 to 8m^3, with an input gain error in both inputs of 2. The setpoint change occurs at time = 80 s

For small gain errors measurement uncertainty in c_A did not greatly effect the decoupling. The performances will be considered for large input gain errors with measurement delay in c_A. The performances for setpoint changes in V are given in tables 5.18 and 5.19.

Table 5.18: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 2 in both inputs, and measurement delay of 4 s in $y_2 = c_A$.

y_{1s}	4.4m^3	8m^3	3.6m^3	2.0m^3
Transformed	1.3006	14.5458	1.3006	6.5034
Linearized	14.2390	37.0390	583.0346*	435.4469*
Decoupled	1.3011	14.7445	1.3011	6.5068

Table 5.19: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 2 in both inputs, and measurement delay of $4\,\mathrm{s}$ in $y_2 = c_A$.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	0.0249	0.1454	0.0294	0.3331
Linearized	0.2417	0.8796	13.0314*	19.7221*
Decoupled	0.0000	0.0262	0.0001	19.1051*

The combination of a large input gain error and time delay could for a SISO system lead to unstable control or a state of stable oscillations with the state dependent MV transformation. Similarly it could be expected in a MIMO system where the perfect decoupling is dependent on model accuracy, that the decoupling could lead to instability or a state of stable oscillations. This is not the case for this system as the decoupling, while not perfect improves the performance as seen in table 5.19 over the linearized case without decoupling. With the exception of the negative setpoint change in V from $4\,\mathrm{m}^3$ to $2\,\mathrm{m}^3$ the decoupling from the linearized system is the best. For the this setpoint change the decoupling is not perfect as the setpoint changes moves the system sufficiently far away from the nominal operations that decoupling based on a linearized model is no longer correct. Coupled with a high gain error and measurement delay the linearized system with decoupling goes into a state of stable oscillations, unlike the transformed system which does settle. This can be seen in fig. 5.9.

*Stable oscillations

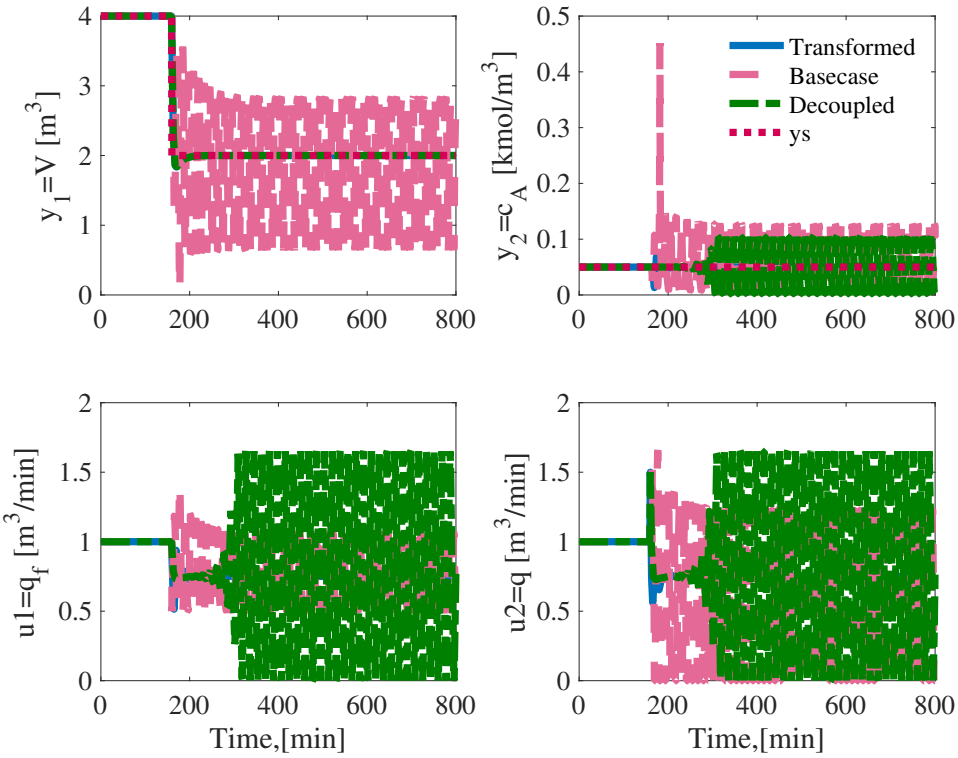

Figure 5.9: Setpoint change in output $y_1 = V$ from 4m^3 to 2m^3, with an input gain error in both inputs of 2, and measurement delat of $4\,\text{s}$ in c_A. The setpoint change occurs at time = 160 s

5.5.4 Input gain error in q

With a gain error in both inputs with the same magnitude, the decoupling for the output $y_1 = V$ was retained. An input gain error in q is expected to not affect the decoupling for $y_2 = c_A$, but only for V so this will be considered. First setpoint changes in both V and c_A will be considered to confirm that c_A is perfectly decoupled from V and v_1. First a small gain error of 1.1 in q will be considered, with the performances given intables 5.20 to 5.23.

Table 5.20: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 1.1 in q.

y_{1s}	4.4m^3	8m^3	3.6m^3	2.0m^3
Transformed	1.7793	17.8436	1.7793	8.8968
Linearized	6.3885	63.5600	6.3924	31.9866
Decoupled	1.7775	17.9016	1.7777	8.8894

Table 5.21: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 1.1 in q.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	0.0000	0.0000	0.0000	0.0000
Linearized	0.0427	0.4192	0.0428	0.2162
Decoupled	0.0045	0.0556	0.0044	0.0217

Table 5.22: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 1.1 in q.

y_{2s}	$0.055\mathrm{kmol\,m}^{-3}$	$0.1\mathrm{kmol\,m}^{-3}$	$0.045\mathrm{kmol\,m}^{-3}$	$0.025\mathrm{kmol\,m}^{-3}$
Transformed	1.2591	18.7475	1.1301	4.4009
Linearized	13.8142	194.1847	12.4225	48.4094
Decoupled	1.3447	20.4071	1.1834	4.4519

Table 5.23: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 1.1 in q.

y_{2s}	$0.055\mathrm{kmol\,m}^{-3}$	$0.1\mathrm{kmol\,m}^{-3}$	$0.045\mathrm{kmol\,m}^{-3}$	$0.025\mathrm{kmol\,m}^{-3}$
Transformed	0.0200	0.2000	0.0200	0.1000
Linearized	0.0845	1.1871	0.0760	0.2962
Decoupled	0.0386	0.4955	0.0359	0.1521

As expected c_A is perfectly decoupled for an input gain error in q as can be seen in table 5.21. This is not the case for the linearized system with decoupling as expected, however, this decoupling does improve performance compared to the control structure without decoupling. This can be seen in fig. 5.10 for a large positive setpoint change in V from $4\,\mathrm{m}^3$to $8\,\mathrm{m}^3$.

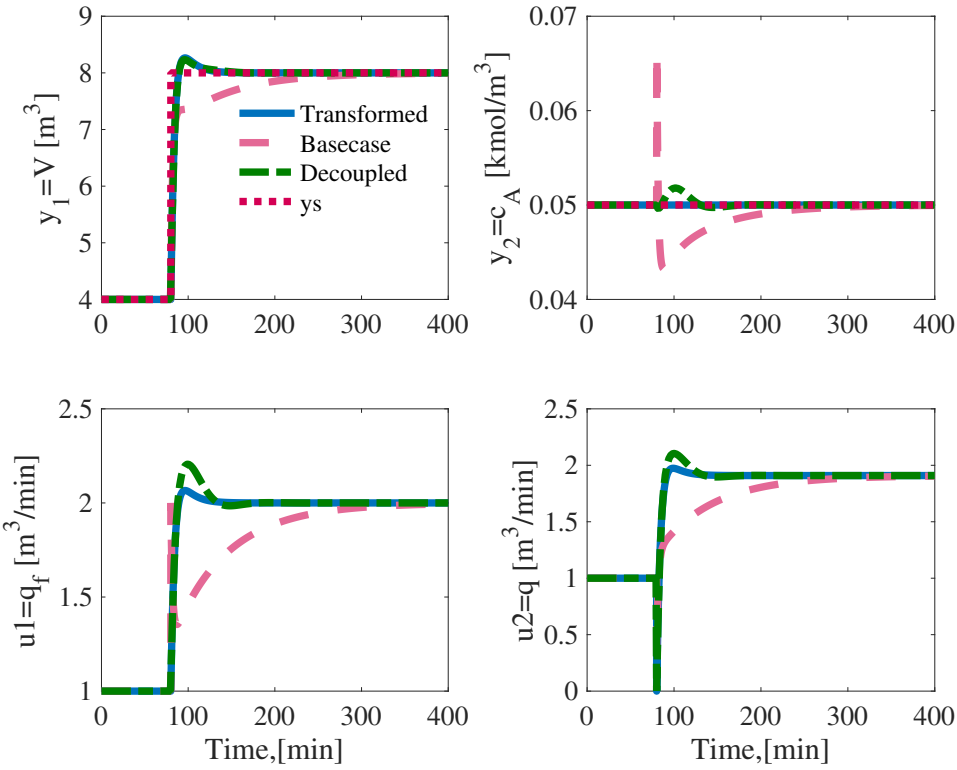

Figure 5.10: Setpoint change in output $y_1 = V$ from $4\mathrm{m}^3$ to $8\mathrm{m}^3$, with an input gain error of 1.1 in the input $u_2 = q$. The setpoint change occurs at time = 80 s

V is however not perfectly decoupled and is dependent on both transformed inputs as seen in table 5.22 for setpoint changes in c_A. Some decoupling is still present as the performance is considerably better compared to the linearized case without decoupling. For the linearized system with decoupling the decoupling is similar to the transformed system albeit slightly worse. This can be seen in fig. 5.11

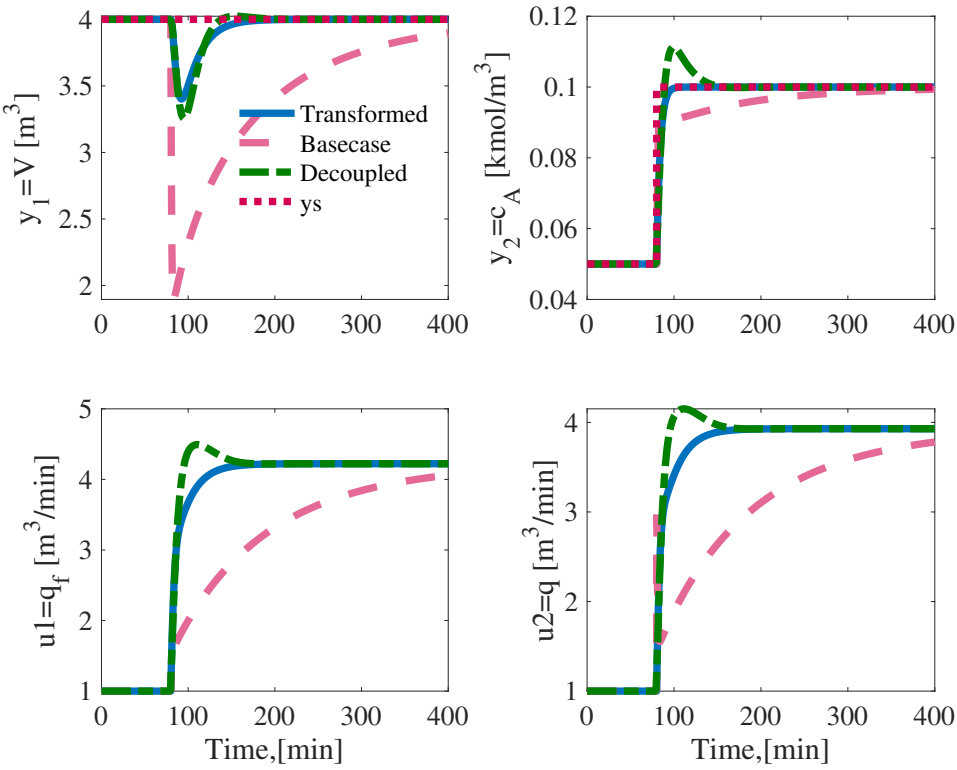

Figure 5.11: Setpoint change in output $y_2 = c_A$ from 0.05kmol m^{-3} to 0.1kmol m^{-3}, with an input gain error of 1.1 in the input $u_2 = q$. The setpoint change occurs at time = 80 s

For a small input gain error in q, c_A was still perfectly decoupled, however, V was not. When changing the setpoint of c_A, V did not remain at its setpoint and needed to be controlled. V was however not as highly coupled in the transformed system as in the original system, as V changed less from its setpoint when the setpoint of c_A was changed. This was for a small gain error in q, and it is unknown if for a larger degree of model error the decoupling will lead to instability. An input gain error of 2 in q will be considered for setpoint changes in c_A. Setpoint changes in V will not be considered since interactions between the different outputs and transformed inputs are of interest and c_A is fully decoupled from V and v_1. The performances are given in tables 5.24 and 5.25.

Table 5.24: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 2 in q.

y_{2s}	0.055kmol m^{-3}	0.1kmol m^{-3}	0.045kmol m^{-3}	0.025kmol m^{-3}
Transformed	6.9248	102.9439	6.2157	24.2051
Linearized	13.8373	197.8932	12.4290	48.4101
Decoupled	7.1377	103.1161	6.4884	24.4296

Table 5.25: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 2 in q.

y_{2s}	$0.055\mathrm{kmol\,m^{-3}}$	$0.1\mathrm{kmol\,m^{-3}}$	$0.045\mathrm{kmol\,m^{-3}}$	$0.025\mathrm{kmol\,m^{-3}}$
Transformed	0.0200	0.2000	0.0200	0.1000
Linearized	0.0466	0.6655	0.0418	0.1629
Decoupled	0.0698	0.9099	0.0649	0.2689

A large gain error in q does not lead to instability, however, V is more coupled as seen in table 5.24, where the IAE for the transformed case is around half of the linearized case without decoupling. While this structure improves performance, the increasing model error does worsen the performance as is expected for a model-based method. In fig. 5.12 a large setpoint change in c_A from $0.05\,\mathrm{kmol\,m^{-3}}$ to $0.1\,\mathrm{kmol\,m^{-3}}$ occurs and the interaction between V and v_2 can be observed as it changes a lot from the setpoint before eventually being controlled back to its setpoint. For a gain error in just q both the transformed model and the linearized model with decoupling have similar give similar decoupling of V. The linearized model with decoupling does however not give perfect decoupling of c_A which the transformed system gives for this model error.

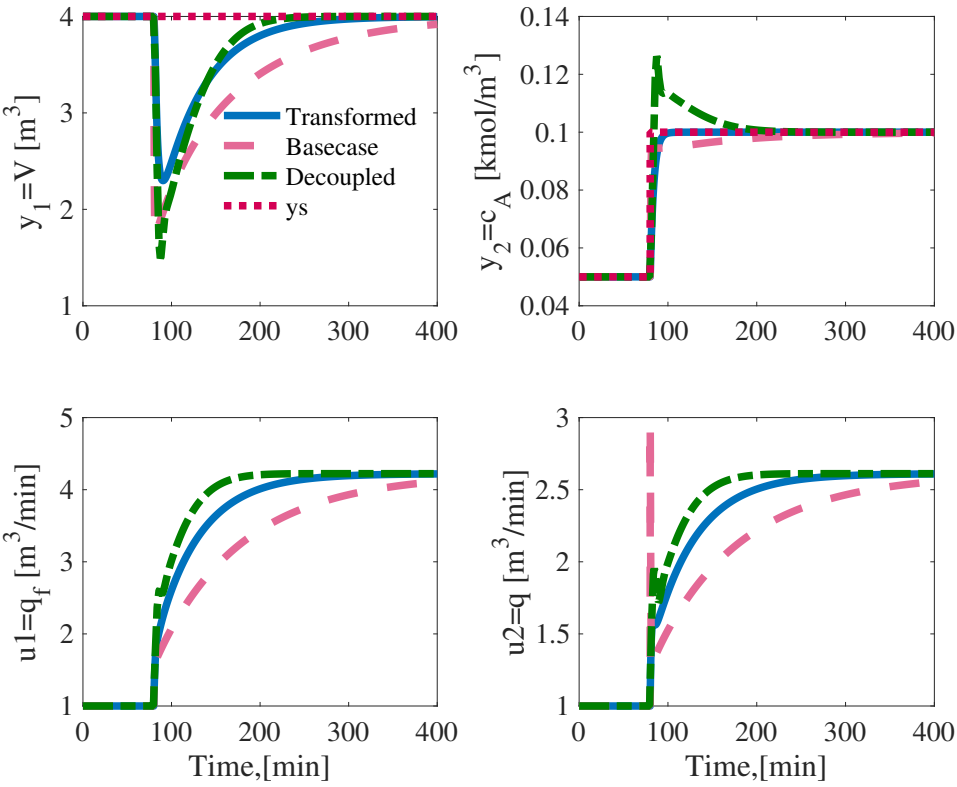

Figure 5.12: Setpoint change in output $y_2 = c_A$ from 0.05kmol m^{-3} to 0.1kmol m^{-3}, with an input gain error of 2 in the input $u_2 = q$. The setpoint change occurs at time = 80 s

5.5.5 Input gain error in $\mathbf{q}_f$

Primarily two combinations of input gain errors have been explored this far for this MIMO CSTR example. Those were input gain error in both inputs of the same magnitude where the perfect perfect decoupling of V was retained, and an input gain error in just q which retained the perfect decoupling of c_A. A third combination is an input gain error in just q_f. For this combination it was previously discussed in section 5.2.2 that perfect decoupling will not be retained for either of the two outputs. This type of model error will also be considered. It has been shown that the transformed MV is quite robust for small model error, so a large gain error will be considered from the start. A large input gain error in q_f of 2 will be considered. Setpoint changes in both c_A and V are considered and the performances are given in tables 5.26 to 5.29.

Table 5.26: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 2 in q_f.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	6.0781	62.4593	5.9427	18.3368**
Linearized	3.1976	31.8485	3.1987	15.9987
Decoupled	4.8779	48.6569	4.8822	27.9568

Table 5.27: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 2 in q_f.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	0.0110	0.0402	0.0168	0.4347**
Linearized	0.0482	0.4699	0.0486	0.2484
Decoupled	0.0240	0.2352	0.0241	0.1388

Table 5.28: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 2 in q_f.

y_{2s}	$0.055\mathrm{kmol\,m}^{-3}$	$0.1\mathrm{kmol\,m}^{-3}$	$0.045\mathrm{kmol\,m}^{-3}$	$0.025\mathrm{kmol\,m}^{-3}$
Transformed	11.0146	4.3208**	6.9519	24.2051
Linearized	6.9184	100.7695	6.2139	24.2048
Decoupled	6.9676	154.6522†	6.2179	24.2053

Table 5.29: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 2 in q_f.

y_{2s}	$0.055\mathrm{kmol\,m}^{-3}$	$0.1\mathrm{kmol\,m}^{-3}$	$0.045\mathrm{kmol\,m}^{-3}$	$0.025\mathrm{kmol\,m}^{-3}$
Transformed	0.0208	0.0903**	0.0161	0.0696
Linearized	0.0931	1.3559	0.0836	0.3258
Decoupled	0.0470	1.1864†	0.0419	0.1629

This type of model error causes many more problems for the transformed system compared to the other combinations of input gain errors. Just as expected there is not perfect input-output decoupling, and for some large setpoint changes the system goes unstable with a transformed system. This is the case for a large negative setpoint change in V from $4\,\mathrm{m}^3$ to $2\,\mathrm{m}^3$ and for a large positive setpoint change in c_A from $0.05\,\mathrm{kmol\,m}^{-3}$ to $0.1\,\mathrm{kmol\,m}^{-3}$. For the cases where the transformed system does not go unstable, it gives

**Unstable
†Tank is emptied

better decoupling than the linearized system with decoupling. The linearized model with decoupling also has difficulties, as the inventory V is fully emptied for a large positive setpoint change in c_A from $0.05\,\mathrm{kmol\,m^{-3}}$ to $0.1\,\mathrm{kmol\,m^{-3}}$. When the inventory V is emptied it is not clear how the concentration should be defined and the simulation crashes.

Since there is a coupling between both outputs and the transformed inputs for the transformed system for model error in the form of gain error of input q_f, the effect of measurement delay in c_A will also be considered. The performances are given in

Table 5.30: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 2 in q_f, and a measurement delay of $4\,\mathrm{s}$.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	209.4554*	64.6547	181476.5573**	2177383.5686**
Linearized	3.5826	34.7605	3.6144	306.5496*
Decoupled	4.8170	48.1529	4.8170	421.1594*

Table 5.31: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_1 = V$ with an input gain mismatch of 2 in q_f and a measurement delay of $4\,\mathrm{s}$.

y_{1s}	$4.4\mathrm{m}^3$	$8\mathrm{m}^3$	$3.6\mathrm{m}^3$	$2.0\mathrm{m}^3$
Transformed	2.4288*	0.2437	29.3837**	31.8632**
Linearized	0.1723	1.6640	0.1737	11.9226*
Decoupled	0.0870	0.8516	0.0877	21.8983*

Table 5.32: Integrated absolute error for $y_1 = V$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 2 in q_f and a measurement delay of $4\,\mathrm{s}$.

y_{2s}	$0.055\mathrm{kmol\,m^{-3}}$	$0.1\mathrm{kmol\,m^{-3}}$	$0.045\mathrm{kmol\,m^{-3}}$	$0.025\mathrm{kmol\,m^{-3}}$
Transformed	174.4302**	2445429.5815**	16.1368	99.0294*
Linearized	6.9055	99.5723	6.2094	24.2041
Decoupled	6.9248	103.0991	6.2157	219.5497*

*Stable oscillations
**Unstable

Table 5.33: Integrated absolute error for $y_2 = c_A$, for setpoint changes in output $y_2 = c_A$ with an input gain mismatch of 2 in q_f and a measurement delay of $4\,\mathrm{s}$.

y_{2s}	$0.055\mathrm{kmol\,m^{-3}}$	$0.1\mathrm{kmol\,m^{-3}}$	$0.045\mathrm{kmol\,m^{-3}}$	$0.025\mathrm{kmol\,m^{-3}}$
Transformed	32.2503**	60.4205**	0.2744	12.9426*
Linearized	0.3537	5.0851	0.3181	1.2404
Decoupled	0.1774	2.6417	0.1593	12.5898*

Measurement error in c_A and a large input gain error in q_f makes the transformed system unusable, as both outputs are driven into instability or a state of stable oscillations for most setpoint changes.

5.6 Summary

In this case study, the decoupling capabilities of the new MV transformation methodology was studied for a CSTR process with two outputs and two inputs. The original process was quite coupled and decoupling could improve performance. When applied to a system with no model uncertainty the MV transformation worked really well and gave perfect decoupling. Measurement uncertainty in the second output was considered, but it did not affect the decoupling which was still perfect. The structural model error however did affect the decoupling and the overall performance of the entire system. For some combinations of model errors, the decoupling was not too severely affected, and the MV transformation still improved performance, but for some combinations, the MV transformation was not robust at all, especially when subject to measurement uncertainty as well. Overall the decoupling from the MV transformation performed similarly compared with a linearized system with decoupling blocks.

*Stable oscillations
**Unstable

Chapter 6

Case study 4: Condensation process

6.1 Model description

In the following case study which will be presented, cascade structures will be applied in combination with MV transformations on a system. In the following system, different extra measurements will be used to design alternative control structures with cascade, which might be simpler than the case without cascade.

The process in question is a heat exchanger steam condensation process. Saturated steam w_d [$\mathrm{kg\,s^{-1}}$] is the hot medium, and a water stream w [$\mathrm{kg\,s^{-1}}$] enters as the medium on the cold side. It is assumed the hold up m [kg] in the cold side is constant in this process and the hold up on the hot side is assumed so small it can be neglected, both the cold side and hot side have uniform temperatures T [K] and T_d [K], where T is the temperature in the cold side, and T_d is the temperature of hot side. It is also assumed that the steam fully condensates, however, that it does not cool down any further. It is also assumed the heat capacity c_p [$\mathrm{kJ\,kg^{-1}\,K^{-1}}$] is constant. The enthalpy of vaporization is λ [$\mathrm{kJ\,kg^{-1}}$] at the reference temperature T_{dref} [K]. The heat transfer area A [$\mathrm{m^2}$] and heat transfer coefficient U [$\mathrm{kW\,m^{-2}\,K^{-1}}$] are both assumed constant. The pressure-temperature relation in the saturated steam is described by the Clausius-Clapeyron relation, and the saturated steam flow rate is described by a valve equation with a valve coefficient k [$\mathrm{kg\,bar^{-1}\,s^{-1}}$] being assumed constant. The temperature of the saturated steam flow before entering the heat exchanger is T_{d0} [K]. This example is from [16] and the flowsheet of the process is given in fig. 6.1.

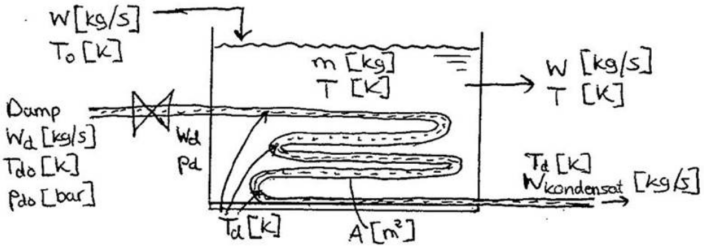

Figure 6.1: Model description of condensation process[16].

The system is a SISO system where the input is the valve position of the steam inlet valve,

$$u = z$$

The output for this case is the temperature of the cold side,

$$y = T$$

There are three primary disturbance variables in this process,

$$d = \begin{bmatrix} T_0 \\ w \\ p_{d0} \end{bmatrix}$$

This system also have one internal state, being the temperature of the steam-condensate mixture in the hot side

$$x_i = T_d$$

The nominal operations are given table 6.1 below.

Table 6.1: Nominal operating values of Heat exchanger

Variable	Value	Unit
T^*	350	K
T_d^*	370	K
z^*	-	-
T_0^*	300	K
T_{d0}^*	390	K
p_d^*	1.00	bar
p_{d0}^*	1.94	bar
w^*	10	$\mathrm{kg\,s^{-1}}$
m	1000	kg
c_p	4	$\mathrm{kJ\,kg^{-1}\,K^{-1}}$
U	1	$\mathrm{kW\,m^{-2}\,K^{-1}}$
A	100	$\mathrm{m^2}$
λ	2200	$\mathrm{kJ\,kg^{-1}}$
k	-	$\mathrm{kg\,bar^{-1}\,s^{-1}}$
R	8.314	$\mathrm{kJ\,kmol^{-1}\,s^{-1}}$
M_m	18	$\mathrm{kg\,kmol^{-1}}$
T_{dref}	370	K
p_{dref}	1.00	bar

Due to the assumptions of perfect mixing on both sides the heat exchanger is modelled as two mixing tanks, one for each side. The inventories on both the cold and warm sides are both assumed constant so no mass balances are needed to describe this. A energy balance can be formulated for the cold side

$$m\frac{dE}{dt} = E_{in} - E_{out} + Q \tag{6.1}$$

which can be written as

$$m\frac{d(mc_pT)}{dt} = wc_p(T_0 - T) + Q \tag{6.2}$$

because $c_V = c_p$ is assumed. T_0 is the inlet temperature of the inlet stream. Since m and c_p are both assumed constant the equation is further simplified to

$$\frac{dT}{dt} = \frac{w}{m}(T_0 - T) + \frac{Q}{mc_p} \tag{6.3}$$

The energy balance on the warm side can be described with following static equation due to the neglected hold up and assumption that the steam is fully condensed but not cooled any further,

$$w_d\lambda = Q \tag{6.4}$$

Due to the constant temperatures on both sides of the heat exchanger the heat transfer can be formulated as,

$$Q = UA(T_d - T) \tag{6.5}$$

The steam flow rate is described by the valve equation,

$$w_d = kz(p_{d0} - p_d) \tag{6.6}$$

where p_{d0} is the steam pressure at $T_d = T_{d0}$. The pressure temperature relation in the saturated steam is described by the Clausius-Clapeyron relation,

$$\frac{dp_d}{p_d} = \frac{\lambda}{R_s T_d^2} dT_d \tag{6.7}$$

eq. (6.7) can be solved to

$$p_d = p_{dref} e^{-\frac{\lambda}{R_s}(\frac{1}{T_d} - \frac{1}{T_{dref}})} \tag{6.8}$$

where p_{dref} is the pressure at $T_d = T_{dref}$ and

$$R_s = \frac{R}{M_m} \tag{6.9}$$

Together this set of equations form the model for the condensation process.

6.2 Transformed variables

In this system the goal is to control the temperature by manipulating the valve position of the steam inlet. The model equation describing the relation between the input and the change in the output is highly non-linear. This is due to the internal state T_d being implicitly a function of itself and the valve position z and the state T, meaning when the valve position or state change T_d instantly changes as well. This implicit relation is found by first solving eq. (6.5) for T_d

$$T_d = \frac{Q}{UA} + T$$

inserting eq. (6.4)

$$T_d = \frac{\lambda w_d}{UA} + T$$

inserting eq. (6.6)

$$T_d = \frac{\lambda kz(p_{d0} - p_d)}{UA} + T$$

and finally inserting eq. (6.8) the implicit relation is formed.

$$T_d = \frac{\lambda kz(p_{d0} - p_{dref} e^{-\frac{\lambda}{R_s}(\frac{1}{T_d} - \frac{1}{T_{dref}})})}{UA} + T$$

Due to the non-linearity this process, using the new methodology to design a transformed variable which linearizes the model might greatly improve control.

The linear methodology for designing transformed MVs will be applied, and the tuning parameter denoted A_z for this specific case is selected to be

$$A_z = \left.\frac{\partial f}{\partial T}\right|_* = -\frac{1}{m}(w^* + \frac{1}{c_p}\frac{k_1}{1+k_2}) \tag{6.10}$$

$$k_1 = \left.\frac{kz\lambda^2 p_d}{RT_d^2}\right|_* \tag{6.11}$$

$$k_2 = \frac{k_1}{UA} \tag{6.12}$$

The transformed variable is then

$$v_{Lz} = \frac{w}{m}T_0 + \frac{Q}{mc_p} - (\frac{w}{m} + A_z)T \tag{6.13}$$

and the transformed linear system is

$$\frac{dT}{dt} = A_z T + v_{Lz} \tag{6.14}$$

The numeric value of A_z is -0.0207 s^{-1}, meaning the open-loop time constant $\tau_z = -1/A_z$ for this transformation is 48.3880 s.

The valve position is calculated by calculating Q from the inverse of eq. (6.13) and then w_d from eq. (6.4). The internal state T_d is then calculated from eq. (6.5), and the valve position z can then be calculated from eq. (6.6).

6.2.1 Cascade structure with extra measurements of w_d

This is a complicated control structure and a simpler system can be formulated if a cascade structure is applied. If measurements w_d are available this can be used as the input in an outer control loop, with an inner loop manipulating the valve position z. The non-linearities from z to w_d would then be taken care of by the inner loop. By considering w_d as the input of the system the system equation is linear in the input, with non-linearities being due to state disturbance interactions. A MV transform can be introduced for this system, however the primary function of this MV transformation would be to introduce improved disturbance rejection in T_0 and w. Disturbances in T_{d0} are taken care of by the inner loop. The linear transformation technique is used to design a MV transformation, and the tuning parameter A_{wd} is selected as

$$A_{wd} = \left.\frac{\partial f}{\partial T}\right|_* = -\frac{w^*}{m} \tag{6.15}$$

For this differentiation w_d is assumed to not be a function of the state T since it is a degree of freedom. The transformed MV is then

$$v_{Lwd} = \frac{1}{m}(wT_0 + \frac{\lambda w_d}{c_p}) - (\frac{w - w^*}{m})T \tag{6.16}$$

and the transformed system is

$$\frac{dT}{dt} = A_{wd}T + v_{Lwd} \tag{6.17}$$

The numeric value for A_{wd} is -0.01 s^{-1}, meaning the open-loop constant $\tau_{wd} = -1/A_{wd}$ for this system is 100 s.

For this structure w_d is calculated from the inverse of eq. (6.16), and this calculated w_d is used as the setpoint for the inner loop.

6.2.2 Cascade structure with extra measurements of T_d

Similarly to how additional measurements of w_d can be used to create a cascade structure, extra measurements of T_d can also be used for this purpose. T_d can be used as the input for an outer loop, resulting in the following model equation for the outer loop

$$\frac{dT}{dt} = \frac{w}{m}T_0 + \frac{UA}{mc_p}T_d - \frac{1}{m}(w + \frac{UA}{c_p})T \tag{6.18}$$

This equation is linear in the input as well, so for this case as well the primary motivation to introduce a transformed MV is to improve disturbance rejection. The linear transformation methodology is once again used to design transformed variables. The tuning parameter A_{Td} is

$$A_{Td} = \left.\frac{\partial f}{\partial T}\right|_* = -\frac{1}{m}(w^* + \frac{UA}{c_p}) \tag{6.19}$$

For this differentiation T_d is assumed to not be a function of the state as it is a degree of freedom in this case. The transformed MV is then

$$v_{LTd} = \frac{w}{m}T_0 + \frac{UA}{mc_p}T_d - \frac{w - w^*}{m}T \tag{6.20}$$

and the transformed system is

$$\frac{dT}{dt} = A_{Td}T + v_{LTd} \tag{6.21}$$

The numeric value for A_{Td} is -0.0350 s^{-1}, meaning the open-loop constant $\tau_{Td} = -1/A_{Td}$ for this system is 28.5714 s.

6.3 Controller tuning

In this case study, PI-controllers will be used for the outer loop and the case without and inner loop. For the inner loops, I-controllers will be used as the relation between the control variables and inputs can be described by algebraic equations. SIMC-tuning rules will be used to tune the controllers in this case study.

The three proposed transformed systems are all on the same form but the open loop constants are different. The three model equations are all on the form

$$\tau_i \frac{dT}{dt} = -T + \tau_o v \tag{6.22}$$

where τ_o is the open loop tuning parameter for the different transformed systems. This model equation can be transformed to the frequency domain with a Laplace transform yielding the transfer function

$$\frac{y}{v}(s) = \frac{\tau_o}{\tau_o s + 1} \tag{6.23}$$

With this transfer function the PI-controller tunings can be found

$$K_c = \frac{1}{\tau_c + \theta}$$
$$\tau_I = min(\tau_o, 4(\tau_c + \theta))$$

The closed loop time constant τ_c will be chosen to be the same for all three transformed systems. It will be chosen to be 50 s, which is a value similar to τ_z which is between τ_{pd} and τ_{Td}. For the inner loops the algebraic the outer loop input and the inner loop input z will be used to find appropriate controller tunings. When the relation between input and output is purely algebraic the process is a pure time delay process. These are controlled using only I-controllers and are tuned with the SIMC-rules in the following procedure

$$K_I = \frac{1}{k_p} \frac{1}{\tau_c + \theta} \tag{6.24}$$

For the control structure where w_d is used as the inner output eq. (6.6) can be used, if linearized around the nominal operating point.

$$K_{Iwd} = \frac{1}{k(p_{d0} - p_d^*)} \frac{1}{\tau_{c2}} \tag{6.25}$$

where τ_{c2} is the inner closed loop constant. To retain a time scale separation between the loops, it will be selected to be 20 times as small as the outer closed loop constant. It is assumed there is no delays in the inner, which is why it is neglected in the tuning.

For the case where T_d is used as an inner loop output section 6.2 is linearized with respect to T_d and z around the nominal operating point of the process.

$$dT_d = \frac{\lambda k(p_{d0}^* - p_d^*)}{UA} dz - \frac{\lambda k z^*}{UA} dp_d \tag{6.26}$$

Inserting Clausius-Clapeyrons equation

$$dT_d = \frac{\lambda k(p_{d0}^* - p_d^*)}{UA} dz - \frac{\lambda k z^*}{UA} \frac{p_d^* \lambda}{R_s T_d^2} dT_d \tag{6.27}$$

Solving this equation for dT_d yields

$$dT_d = \frac{k_3}{1+k_2} dz \tag{6.28}$$

$$k_3 = \frac{\lambda k}{UA}(p^*_{d0} - p^*_d) \tag{6.29}$$

The controller tuning for the inner loop is then found to be

$$K_{ITd} = \frac{1+k_2}{k_3}\frac{1}{\tau_{c2}} \tag{6.30}$$

The inner closed loop time constant will be selected to be the same for both cascade structures.

6.4 Case study

A short case study will be conducted in this chapter, where MV transformations combined with a cascade structure will be compared with a structure that directly manipulates the input. Three structures have been proposed, two with cascade and one without. First, a setpoint change in the output T will be considered. The setpoint change conducted will be a small one from $T = 350\,\mathrm{K}$ to $T = 355\,\mathrm{K}$. Since all the transformed structures are dependent on output feedback for the MV transformation blocks some measurement time delay will be considered.

A problem these MV transformations have is that they are dependent on a good model to perform well. With model mismatch and uncertainty the MV transformation control structure will suffer as shown in previous case studies. If the model uncertainty is only for a small section of the process like uncertainty in the behavior of a valve, a cascade structure where this uncertain part of the model is removed will probably perform better. This will be tested in this case study as well, by introducing model error in the input gain of z. The manipulation of the valve position is taken care of the inner controllers in the cascade structures, so model uncertainty in this section should not affect the outer loop, and performance should not be affected much.
Input gain error in z will be implemented similar to how it was implemented in eq. (4.14) in section 4.3.2 in chapter 4. An input gain mismatch K_u of 2 will be used.

6.4.1 Setpoint changes

The setpoint changes were conducted and are given in table 6.2 below.

Table 6.2: Integrated absolute error for setpoint changes in T from 350 to 355 K for the three proposed control structures. Both a time delay of $50\,\mathrm{s}$ and an input gain mismatch of 2 in z are considered.

	$K_u = 1$		$K_u = 2$	
Structure	$\theta = 0\mathrm{s}$	$\theta = 50\mathrm{s}$	$\theta = 0\mathrm{s}$	$\theta = 50\mathrm{s}$
v_{Lz}	250.0000	587.9354	141.3691	561.2041
v_{Lwd}	259.3289	614.5101	252.6574	574.9327
v_{LTd}	250.0000	556.1422	250.0000	546.3862

Some conclusions can be made based on the results. One is that with extra measurements a cascade structure can most definitely be used, as in this case the cascade structures both perform similarly to the case without cascade as can be seen in fig. 6.2 for a setpoint change with no model uncertainty. The cascade structures which use extra measurements of T_d performs slightly better than the one which uses extra measurements of w_d.

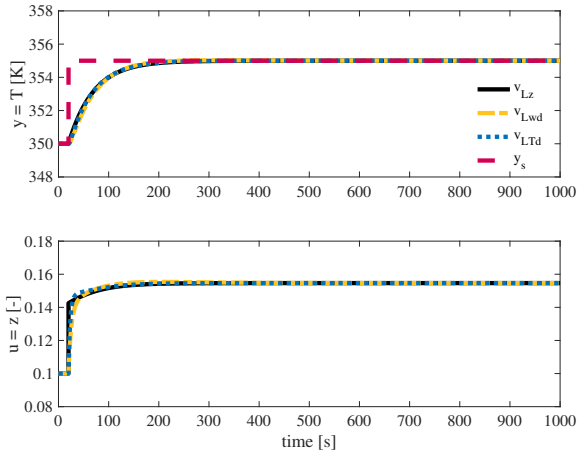

Figure 6.2: Setpoint change in T from 350 to 355K at time = 20 s.

Measurement delay as expected worsens the performance of the different structures as the IAE is around twice as much compared to the case without measurement delay. Based on fig. 6.3, the performance is still good, with small overshoots, probably due transformation being dependent on the state measurements. It should be the cascade structure with T_d performs the best, while the cascade structure with w_d performs the worst. The structure without cascade which directly manipulates the valve position performs somewhere in between the two cascade structures.

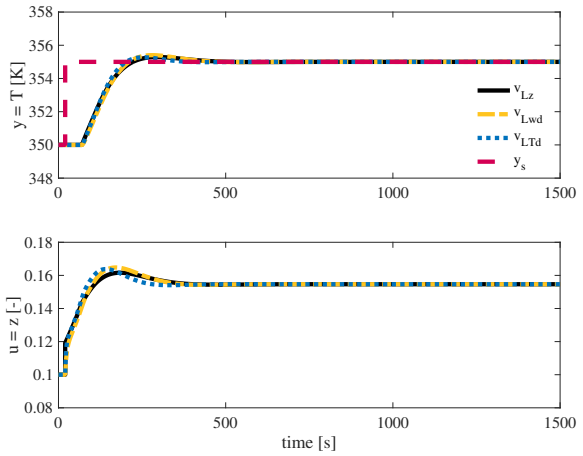

Figure 6.3: Setpoint change in T from 350 to 355K at time = 20 s, with measurement time delay of 50 s of T.

With model error in the slave loop, the performance of the cascade structure is not changed much. The performances are slightly better, due to the input gain error leading to a more aggressive response to changes in the input. The structure which directly manipulates the input is however susceptible to model errors in this part of the model. For the case without measurement delay, it improves the performance greatly as seen in fig. 6.4. For the case with measurement delay, the performance is not greatly improved. This is due to the response being too aggressive due to the combined model errors, leading to a larger overshoot as seen in fig. 6.5. If avoiding overshoot is a priority and there is model uncertainty which can be neglected with a cascade structure, cascade structures are good solutions.

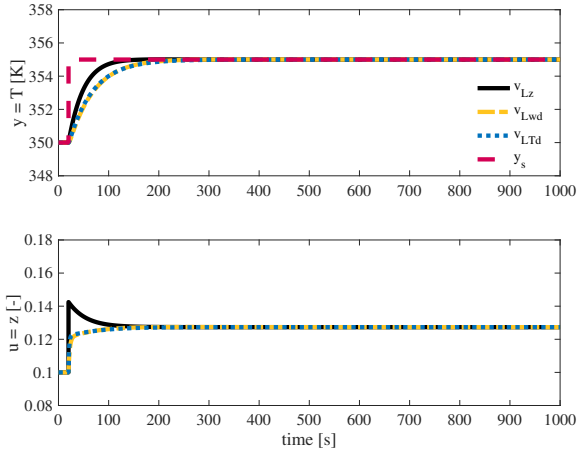

Figure 6.4: Setpoint change in T from 350 to 355 K at time = 20 s, with an input gain error in z of 2.

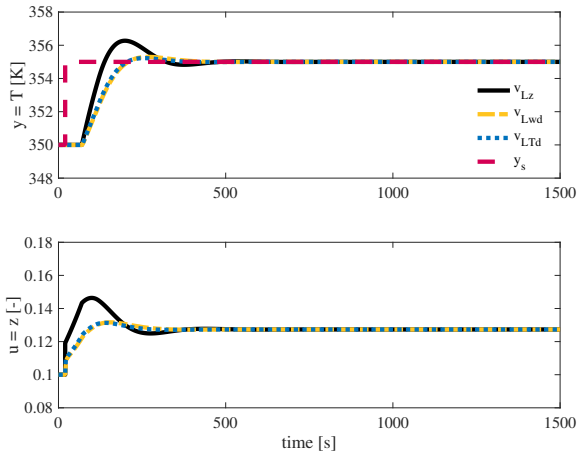

Figure 6.5: Setpoint change in T from 350 to 355K at time = 20 s, with measurement time delay of 50 s of T, and an input gain error in z of 2.

6.4.2 Disturbances

Setpoint changes were considered and the cascade structures worked as well as the structure which directly manipulates the input. Step disturbance in the three disturbance variables will be considered, with both for measurement time delay and input gain model errors in z. A strength of the transformed MVs is that if designed with a perfect model, even when subject to measurement uncertainty in the state, they give perfect disturbance rejection. This will no longer be the case for the cascade structures since the input z is no longer directly manipulated by the MV transformation. With fast inner controllers, the disturbance rejection should however be fast for these structures.

Step disturbance tests were performed in all three disturbance variables with a step disturbance 10 % from the nominal value, and the performances are given in tables 6.3 to 6.5 below.

Table 6.3: Integrated absolute error for step distrubances in T_0 from 300 to 330 K for the three proposed control structures. Both a time delay of 50 s and an input gain mismatch of 2 in z are considered.

	$K_u = 1$		$K_u = 2$	
Structure	$\theta = 0$s	$\theta = 50$s	$\theta = 0$s	$\theta = 50$s
v_{Lz}	0.0000	0.0000	489.2815	7538.4155*
v_{Lwd}	38.9891	75.1152	16.8495	28.8214
v_{LTd}	10.8390	17.5731	3.9787	4.6100

*Stable oscillations

Table 6.4: Integrated absolute error for step distrubances in w from 10 to 11 $\mathrm{kg\,s^{-1}}$ for the three proposed control structures. Both a time delay of $50\,\mathrm{s}$ and an input gain mismatch of 2 in z are considered.

	$K_u = 1$		$K_u = 2$	
Structure	$\theta = 0\mathrm{s}$	$\theta = 50\mathrm{s}$	$\theta = 0\mathrm{s}$	$\theta = 50\mathrm{s}$
v_{Lz}	0.0000	0.0000	489.2815	7538.4155*
v_{Lwd}	37.2025	75.1152	16.8495	28.8214
v_{LTd}	10,3796	17.5731	3.9787	4.6100

Table 6.5: Integrated absolute error for step distrubances in p_{d0} from 1.9351 to 2.1286 bar for the three proposed control structures. Both a time delay of $50\,\mathrm{s}$ and an input gain mismatch of 2 in z are considered.

	$K_u = 1$		$K_u = 2$	
Structure	$\theta = 0\mathrm{s}$	$\theta = 50\mathrm{s}$	$\theta = 0\mathrm{s}$	$\theta = 50\mathrm{s}$
v_{Lz}	0.0000	0.0000	489.2815	7538.4155*
v_{Lwd}	39.4684	75.1152	16.8495	28.8214
v_{LTd}	10.9795	17.5731	3.9787	4.6100

Interestingly for the the performances for the cases with model error, measurement delay, gain error or both are identical for disturbances in all three disturbances. For step disturbances with no model error and no gain errors the performances are almost the same for disturbances in all three disturbances. Because of this disturbances only disturbances in T_0 will be discussed as the system behaves identical or near identical for the other disturbances.

For a perfect model as expected the structure which directly manipulates the input has perfect disturbance rejection both with and without measurement delay. As expected this is not the case for the cascade structures, however, the disturbance rejection is very good, both with and without measurement delay in T. Some measurement delay in T did however slightly worsen the performance. Since the output T deviates a bit from its setpoint with the cascade structure, the outer loop will make small corrections to the transformed MV to further improve the disturbance rejection. With measurement delay, this correction is probably a bit late and might worsen the performance slightly compared to no correction from the outer feedback loop. Interestingly the cascade structure which uses measurements of T_d performs better than the structure which uses measurements of w_d, this can be seen in figs. 6.6 and 6.7. The cascade structures do not suffer from this problem since the model error is in the neglected part of the model.

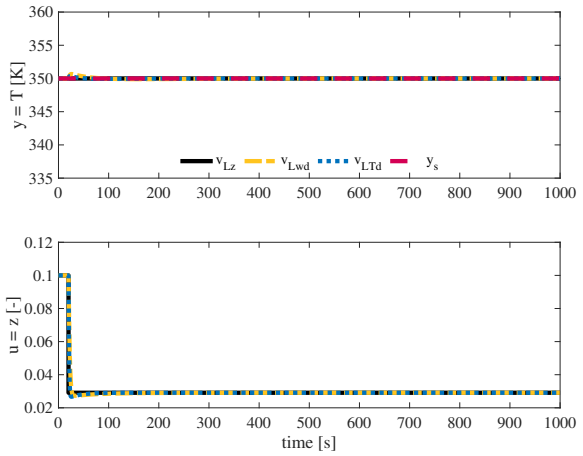

Figure 6.6: Step disturbance in T_0 from 300 to 330K at time = 20 s.

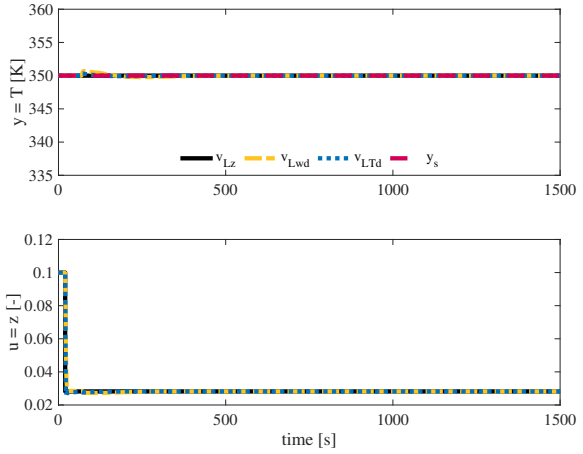

Figure 6.7: Step disturbance in T_0 from 300 to 330K at time = 20 s, with measurement time delay of 50 s .

With a gain error in the input z the structure which directly manipulates z struggles. While the disturbance is rejected for the case without measurement delay, the initial correction is too large and the output T ends up deviating quite a bit from the setpoint before being rejected by the feedback as seen in fig. 6.8. As is earlier explored these state-dependent transformation are not too robust to model error combined with measurement uncertainty. This is the case for this system as well as the disturbance is not rejected and the system goes into a state stable oscillations as seen in fig. 6.9.

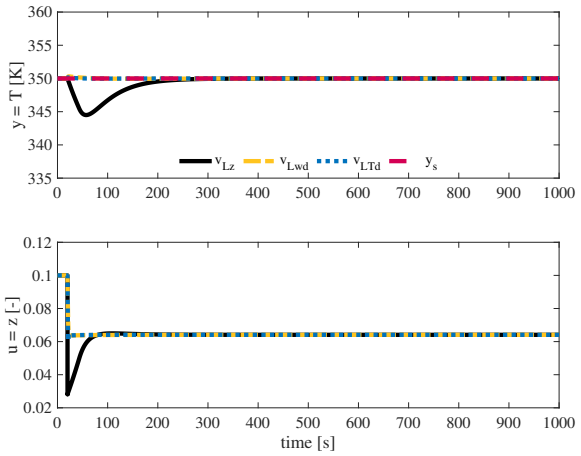

Figure 6.8: Step disturbance in T_0 from 300 to 330K at time = 20 s with an input gain mismatch of 2 in z.

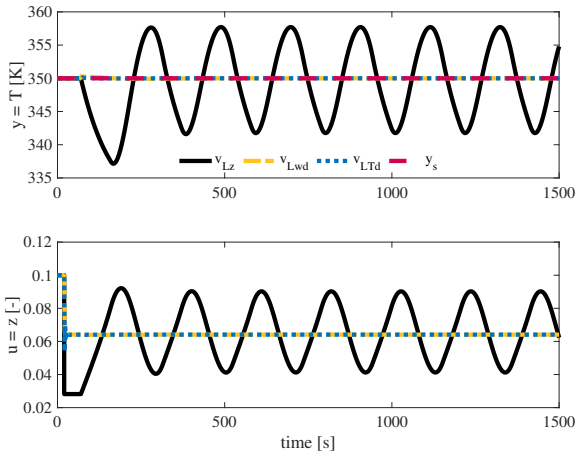

Figure 6.9: Step disturbance in T_0 from 300 to 330K at time = 20 s, with measurement time delay of 50 s and an input gain mismatch of 2 in z.

Chapter 7

Reflections

In this thesis, a new methodology for transforming nonlinear systems into linear ones by applying an MV transformation has been studied. Different methodologies to design transformed MVs were proposed. The two main design proposals were the static transformation in section 2.3.2 and the linear transformation in section 2.3.1. One of the stated aims of the new methodology is to transform the nonlinear system into a linear one which can be controlled using linear control theory. This is something the static transformation fails at, as the transformation transforms the system into a different nonlinear system. These nonlinearities come in the form of a dynamic gain which can be dependent on the physical input and/or the disturbance variables depending on the system in question. This was observed in the Extraction tank case study in chapter 3 and the mixing tank case study in chapter 4.

7.1 Static transformation

In the Extraction tank case study in chapter 3 different closed-loop responses were tested with the static MV transformation, and it was assumed that the static transformation would perform the best for slow closed loop dynamics. This was due to the transformation being derived using a steady-state simplification of the model equations, and not considering the dynamics of the system. With faster closed-loop dynamics the controller might make too aggressive input adjustment, potentially making the system go unstable. As was expected with a large closed-loop constant the static transformation works well, and for a small closed-loop constant the system goes unstable due to too aggressive changes in the input. While the input could be bounded with some physical limitation, which would be the case in a real physical process, this was not done in the case study to demonstrate the extreme effects of these transformations because the system was not fully linearised. In this case, fast and slow closed-loop dynamics are relative to the open-loop dynamics of the transformed system. While these dynamics will change with changing operations due to the nonlinearities, fast and slow closed-loop dynamics were considered in comparison to the

nominal open-loop dynamics of the system in this thesis. The static case performed well for closed-loop dynamics of similar magnitude to the nominal open-loop dynamics. This was however for no model error or measurement delay.

Model error and measurement delay was considered in the mixing tank case in chapter 4. In this case study, a challenge with the transformation arose. This challenge was that the transformation must be invertible for the entire operating range. This property is which might not be retained for MV transformation, as was the case for the static MV transformation in this case study. The range of values the transformed value v could be for the back-calculation of the input to yield positive and physical values was quite small. This meant the controller manipulating the transformed output needed to be tuned with a slow closed-loop response. When tuned with a slow closed-loop response as was the case when the measurement delay was large, this structure performed well. For the cases with less or no measurement delay, this control structure failed at controlling the process, probably due to the fast closed-loop dynamics. From the case study, some measurement delay might worsen the performance, however, that is probably not the case as it is more likely the fast controller tunings are to blame. This is reinforced by the fact that larger measurement delay did not worsen the response, with slow controller tunings. Model error in the form of an input gain error did not seem to affect the performance much. This might indicate that the static transformation is robust to model error, however, this was a simple process and a conclusion that the static transformation is robust to model error might be too early, and the more complex process should be evaluated as well. Besides, only one type of model error was considered.

7.2 Linear transformation

The second primary MV transformation which was proposed was linear transformation. This transformation gets its name from the fact that it properly linearizes the nonlinear system over the entire operation range, with the transformed system being a first-order process. This was shown in the extraction process case study in chapter 3. However, a drawback of this structure is that transformation is dependent on state measurements. This state feedback makes this transformation less robust to model error. This was shown in the mixing tank case study, the CSTR case study, and the condensation process case study in chapters 4 to 6. A combination of both large measurement uncertainty and model mismatch was found to cause the biggest robustness issues. This should be expected of a model-based approach. Whether this transformation technique is most robust to model error or measurement uncertainty is not certain, as it was the combination of both which caused most problems, so the further study could be needed.

7.3 Disturbance rejection

One of the properties of this new methodology is that it gives perfect disturbance rejection. There are two limiting factors for this property though. The first is that all disturbances must be measured and the second is that the process model must be sufficiently good. These two factors are closely related. Not all disturbances in a process are measurable, and the newly proposed method will therefore not be able to improve in rejecting these. Unmeasured disturbances can for a model-based approach be considered a form of model uncertainty or model mismatch. An unmeasured disturbance variable can be considered as a static parameter in an implementation, but when changes in this disturbance variable occur this will not be reflected in the variable transformation. This could be considered a parameter model mismatch, a form of model error that was not considered in this thesis.

Measurement uncertainty in the disturbance variables was in some form considered in this thesis in the mixing tank case in chapter 4. Measurement uncertainty comes in many forms, it can come in the form of measurement delay or measurement noise, or that the measurement is wrong by some margin. In this thesis for the SISO mixing tank case, a gain error in the effect of the disturbance on the process was considered. At the nominal operations, the disturbances variables were measured correctly but as the disturbance variables deviated from this nominal measurement, the measurement deviated linearly with some gain. This could also be considered a form of a model error where the measurements are perfect but the model is linearly worse as the disturbance deviates from the nominal measurements. For step disturbances, the new method no longer gave perfect disturbance rejection as was expected, however, the feedforward action improved the disturbance rejection. This is due to the measured change in the disturbance variables only being wrong by some magnitude, meaning the feedforward action which was performed by the control structure would only make an adjustment which was wrong by some magnitude. As long as this magnitude is not too great some improved performance should be expected with this type of measurement uncertainty.

Two more types of measurement uncertainties for disturbances were mentioned but not examined in this thesis, so the effect of them can only be speculated about. The first was measurement delay. If no feedback is present in the control structure the new proposed structure should handle measurement delays if subject to no other model uncertainty or measurement uncertainty, however, the disturbance rejection might be slow, depending on how large the measurement delay is. If there is feedback control as well, disturbance measurement delay might cause the feedback and the feedforward to fight leading to worsened disturbance rejection. If the feedback loop is much faster than the disturbance measurements, the feedback might fully reject the disturbance before the feedforward action is applied to the system. This is however not a problem unique to the new methodology, but a general challenge with feedforward control. The second type of measurement uncertainty that was not studied was measurement noise. Measurement noise could lead to the input changing a lot and unnecessarily. This is not ideal for control, but could probably be solved by filtering the measurements.

7.4 Multivariate systems

In the CSTR case in chapter 5 the new methodology was extended to a multivariate process. This was one limitation of the feedback linearization methods as it required a type of nonrobust nonlinear decoupling. The new methodology also introduced decoupling, however, is it robust was studied for the CSTR process. With a perfect model, the new methodology successfully transformed the coupled nonlinear system into a linear and decoupled system. This was however only the case for a process without model uncertainty. For more complex systems model error can come in more forms than for simple systems. This is the case for multivariate systems. In this thesis input gain error is used as the primary model error, however, for multivariate systems, there are several inputs, so many different combinations of input error can occur. A few of these were tested, and the new methodology was robust for certain model errors but not for others. If the new methodology introduces a robust nonlinear decoupling is cannot be concluded. While the new methodology was robust to some model errors, it was not to others. Model uncertainty other than input gain and output measurement delay was not considered either. Other multivariable systems as well, might be less or more robust, so more studies of the application of MV transformations on multivariable systems could be studied.

7.5 Cascade

In the condensation case study, one of the applications of a cascade structure in the MV transformation environment was studied. This was to use extra measurements to simplify the model equation when designing MV transforms. For the system in the question, the most severe nonlinearities could be eliminated with the cascade structure. For the system in question using transformed MVs might even have been unnecessary after the model reductions. Perfect disturbance rejection is no longer possible with cascade structures, but if the slave loops are sufficiently fast near perfect is. One of the benefits of using a cascade structure with extra measurements is that model mismatch in the neglected part of the model will not affect the performance. This means if there are some sections of the model with much uncertainty cascade control will improve performance. This is however not a new finding and just a property of cascade structure. The effect of model error in the parts of the model which were not neglected was not considered.

Chapter 8

Discussion

8.1 Robustness analysis

In this thesis case studies and simulation were carried out. In the case studies simulations were done with model error to gain some insight into the new methodology, and how robust it is. By doing case studies firm conclusions on the robustness of the new methodology is not attainable, as case studies can almost be considered a sophisticated form of trial and error. Due to this an analytical robustness analysis would be needed to make better conclusions as to the robustness of the new proposed methodology. With an analytical analysis a better understanding of measurement uncertainty in the outputs, and their effect on transformed manipulated variables could be made, and the robustness for multivariate systems.

8.2 Output transformation

In this thesis primarily the manipulated variable transformation was considered, however the method also included an output transformation as seen in fig. 8.1.

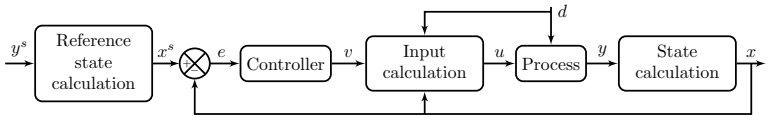

Figure 8.1: Block diagram of the new proposed method with output transformation, this is the same figure as in fig. 2.4.

This output transformation is applied for systems where the outputs are not the states but either some linear, or a nonlinear function of the states. This was not considered in this thesis as the states were considered outputs for all the case studies conducted. This extension could be useful for systems with large nonlinearities in the output from states

mapping. pH-control is an example of a process where the state to output transformation is highly nonlinear. A requisite condition for this is that the output to state transformation is possible.

8.3 Anti-windup

Anti-windup is applied to PI and PID controllers when the input saturates at some upper or lower boundary to counteract integral windup in the controllers. This was not considered in this case study. With anti-windup some harsher constraints on the input could have been placed, possibly giving better or at least stable control. This is especially the case for the static transformation when tuned with a fast closed loop constant, as this structure was quite aggressive.

Chapter 9

Conclusion

In this thesis a new proposed methodology which transforms a nonlinear process into linear first order process with a manipulated variable transformation was studied. This new methodology also introduces perfect disturbance rejection and for multivariable systems decoupling. The main assumptions for this new method is that it could only be applied to systems with a relative degree of 1, where the relative degree of a process is the amount of times the output must be differentiated with respect to time to be explicitly a function of the input. This assumption could however be loosened if a cascade structure is used. The two other main assumptions is that all states and disturbances are measured.

Two main methods of designing transformed MVs were proposed. The linear method which succeeds in transforming the nonlinear system into a linear first order process with the use of state feedback, and the static transformation which is designed using static models. The static methodology does not succeed in transforming the system into linear system, but it is not dependent on state feedback which is beneficial.

A number of case studies were conducted to test different properties of this new methodology. In chapter 3 the two main methods of designing MV transformation were tested with no model error for an extraction process. Since the linear transformation transformed the non-linear system into a linear on, and the static did not, the linear transformation was the best and could be applied for systems with fast closed loops. For slow closed loops the static transformation performed as well as the linear transformation. It was also shown that without an outer feedback loop the transformed systems managed to reject disturbances. A special case for this particular system was that the constraint that all states and disturbances needed to be measured could be relaxed, as a model simplification allowed designing transformed MVs which measured other physical properties.

In chapter 4 different proposed MV transformations were tested and compared when subject to model error and measurement delay for the mixing tank example. For this case the feedback controllers used was tuned such that the closed loop time constant was the same

as the measurement delay, with the exception being when the measurement delay was zero. Then the closed loop time constant was tuned similar to the case with a little measurement delay. Due to the fast tuning when subject too little or no measurement delay the static transformation did not perform to well when changing the setpoint. It did however excel at rejecting disturbances however. With slow controller tunings the static case performed the best, even with measurement delay and model error. The linear transformation performed the best with no measurement delay, even with model error. With measurement delay the linear transformation no longer linearized the system, however when the measurement delay was small this transformation still performed the best. The linear transformation was not robust to the combination of large model errors and large measurement delay. To conclude the linear transformation is the best strategy for fast closed loops, and little to no output measurement uncertainty. If the output measurement uncertainty is large the static case is preferable, however the controller needs to be tuned with a slow closed loop constant. Both transformation strategies gave good disturbance rejection, but the linear transformation was the least robust to model error and measurement delay.

In chapter 5 the MV transformations were applied to a multivariable system, where the decoupling capabilities of the proposed structure was applied on a CSTR example. For no model uncertainty the the new methodology gave perfect decoupling. With small model mismatch the control structure still gave some decoupling, however that dependent on the form of the model errors. For some combinations of model error this structure was not robust at all. The conclusion is that if a good model is available this new methodology can improve the performance. This is especially true for highly coupled MIMO systems. This methodology might not be too robust with respect to model error however that should be further studied for different systems and model uncertainty.

In chapter 6 cascade structures were used to simplify a complex nonlinear model by applying extra measurements, for a condensation process. In this process much of the nonlinearities were removed from the model equations with this structure. As shown in the case study this made the control structure more robust to model uncertainties, if they were in the neglected part of the model.

Overall this new methodology has introduced a simple systematic procedure for designing control structures which gives both feedforward control and decoupling. However being a model based approach somewhat good models are required.

9.1 Further work

In this thesis a new methodology for transforming nonlinear systems into linear systems were examined. As it is a new methodology there are many things left to study. A large section of this theses was to test the new methodology when subject to model uncertainty, however there were many types of model uncertainties which could be explored. An analytical robustness analysis of the new methodology should also be further studied.

There were also still many applications of this new methodology which were not explored

in this thesis. One application which could be explored is to design a control structure using this method for systems where a complex dynamic model is unavailable, but a simpler static model is. As already discussed this method could be further applied to more multivariable systems to gain further knowledge of the limitations and capabilities of the new methodology on multivariable systems. Three applications of cascade were given however only one was explored in this thesis, so the two other applications could be explores as well.

For all the processes which were considered in this thesis the output variables were measured states. The new methodology proposes a state to output transformation as well for processes where the state to output relation is nonlinear. This was not explored in this thesis and can be explored in further work.

Bibliography

[1] Karl Johan Åström and Björn Wittenmark, editors. *Adaptive Control*. Addison-Wesley, 1989.

[2] Jens. G. Balchen. Elementary nonlinear decoupling (end), a general approach to model based control of nonlinear multivariable processes. *Springer*, 1998.

[3] Jens. G. Balchen, Bernt Lie, and Ingar Solberg. Internal decoupling in non-linear process control. *Modeling, Identifiaction and Control*, 1988.

[4] Christopher I. Byrnes and Alberto Isidori. Global feedback stabilization of nonlinear systems. In *24th IEEE Conference on Decision and Control*, 1985.

[5] Michael A. Henson and Dale E. Seborg. Critique of exact linearization strategies for process control. *Journal of process control*, 1991.

[6] Michael A. Henson and Dale E. Seborg. *Nonlinear Process Control*. Prentice Hall, 1997.

[7] Alberto Isidori. *Nonlinear Control Systems*. Springer, London, third edition, 1989.

[8] Alberto Isidori, Arthur J. Krener, Claudio Gori-Giorgi, and Salvatore Monaco. Nonlinear decoupling via feedback: A differential geometric approach. *IEEE Transactions on Automatic Control*, 26(2):331–345, 1981.

[9] James J Jones and Sigurd Skogestad. An industrial and academic perspective on plantwide control. *Annual Reviews in Control*, 35:99–110, 2011.

[10] Costas Kravaris and Chang-Bock Chung. Nonlinear state feedback synthesis by global input /output linearization. *AIChE Journal*, 1987.

[11] Michael J. Kurtz and Michael A. Henson. Input-output linearizing control of constrained nonlinear processes. *Journal of Process Control*, 1997.

[12] Horacio J Marquez. *Nonlinear Control Systems*. Wiley, 2003.

[13] Henk Nijmeijer and Arjan van der Schaft. *Nonlinear Dynamic Control Systems*. Springer, New York, 1990.

[14] James B. Rawlings and David Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill, 2009.

[15] F. G. Shinskey. *Process control systems*. McGraw-Hill, third edition, 1979.

[16] Sigurd Skogestad. Eksamen i prossesregulering. Old exam, december 1997.

[17] Sigurd Skogestad. Simpel analytical rules for model reduction and pid controller tuning. *Journal of process control*, 13(4):291–309, 06 2003.

[18] Sigurd Skogestad. Control structures with embedded knowledge through input and output transformations control structures with embedded knowledge through input and output transformations. *In preporation*, 2020.

[19] Cristina Zotica, Nicholas Alsop, and Sigurd Skogestad. Transformed manipulated variables for linearization, decoupling and perfect disturbance rejection transformed manipulated variables for linearization, decoupling and perfect disturbance rejection. In *1st Virtual IFAC World Congress*, 2020.

Appendix

Appendix A

Derivation of A_z in condensation process

In the condensation process in chapter 6 the tuning parameter A_z for the linear transformation was just given. The derivation for this parameter will be given in this appendix.

$$A_z = \left.\frac{\partial f}{\partial T}\right|_* \tag{A.1}$$

$$\frac{\partial f}{\partial T} = -\frac{w}{m} + \frac{1}{mc_p}\frac{\partial Q}{\partial T} \tag{A.2}$$

$$\frac{\partial Q}{\partial T} = \lambda \frac{\partial w_d}{\partial T} \tag{A.3}$$

$$\frac{\partial w_d}{\partial T} = -kz\frac{\partial p_d}{\partial T} \tag{A.4}$$

$$\frac{\partial p_d}{\partial T} = \frac{dp_d}{dT_d}\frac{\partial T_d}{\partial T} \tag{A.5}$$

$$\frac{dp_d}{dT_d} = \frac{\lambda p_d}{R_s T_d^2} \tag{A.6}$$

$$\frac{\partial T_d}{\partial T} = \frac{1}{UA}\frac{\partial Q}{\partial T} + 1 \tag{A.7}$$

$$\frac{\partial Q}{\partial T} = -\frac{kz\lambda^2 p_d}{RT_d^2}\left(\frac{1}{UA}\frac{\partial Q}{\partial T} + 1\right) \tag{A.8}$$

$$k_1 = \left.\frac{kz\lambda^2 p_d}{RT_d^2}\right|_* \tag{A.9}$$

$$k_2 = \frac{k_1}{UA} \tag{A.10}$$

$$\left.\frac{\partial Q}{\partial T}\right|_* = -k_2 \left.\frac{\partial Q}{\partial T}\right|_* - k_1 \tag{A.11}$$

$$\left.\frac{\partial Q}{\partial T}\right|_* = -\frac{k_1}{1+k_2} \tag{A.12}$$

$$\left.\frac{\partial f}{\partial T}\right|_* = -\frac{1}{m^*}\left(w^* + \frac{1}{c_p}\frac{k_1}{1+k_2}\right) \tag{A.13}$$

$$\left.\frac{\partial f}{\partial T}\right|_* = -\frac{1}{m^*}\left(w^* + \frac{UA}{c_p}\frac{k_2}{1+k_2}\right) \tag{A.14}$$

Appendix B

Figures from mixing tank case study

B.1 Uncertainty in the measurement delay

B.1.1 Setpoint changes

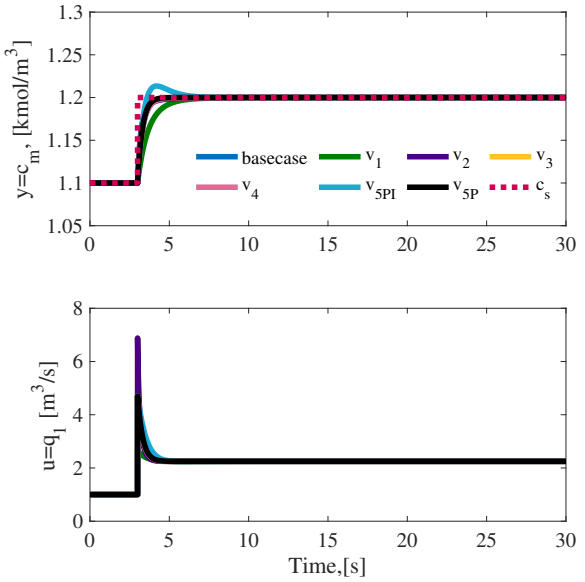

Figure B.1: Setpoint change from c_s = 1.1 kmol m^{-3} to c_s = 1.2 kmol m^{-3}.

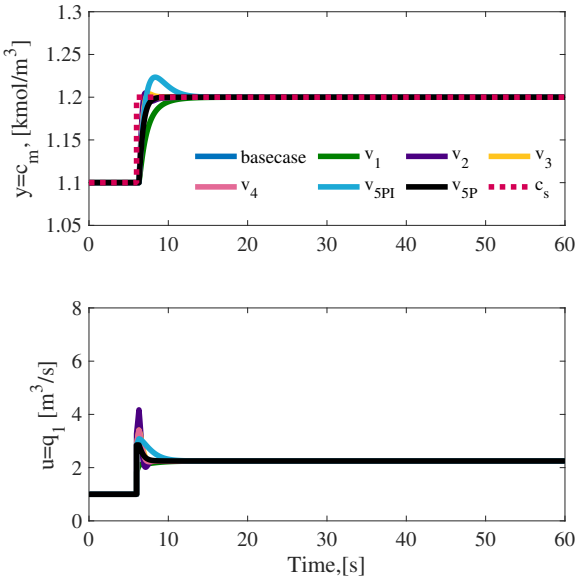

Figure B.2: Setpoint change from $c_s = 1.1\ \mathrm{kmol\,m^{-3}}$ to $c_s = 1.2\ \mathrm{kmol\,m^{-3}}$, with measurement delay $\theta = 0.3\,\mathrm{s}$.

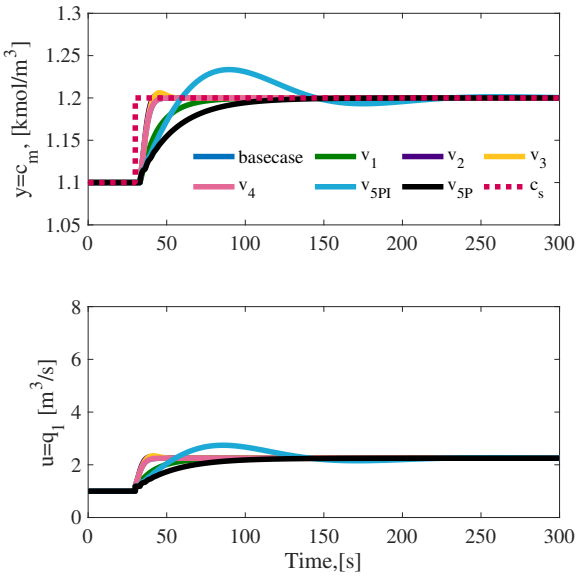

Figure B.3: Setpoint change from $c_s = 1.1\ \mathrm{kmol\,m^{-3}}$ to $c_s = 1.2\ \mathrm{kmol\,m^{-3}}$, with measurement delay $\theta = 3\,\mathrm{s}$.

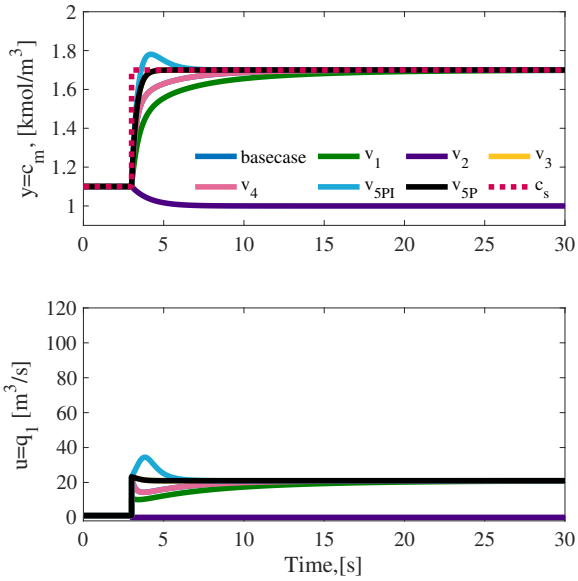

Figure B.4: Setpoint change from c_s = 1.1 kmol m^{-3} to c_s = 1.7 kmol m^{-3}.

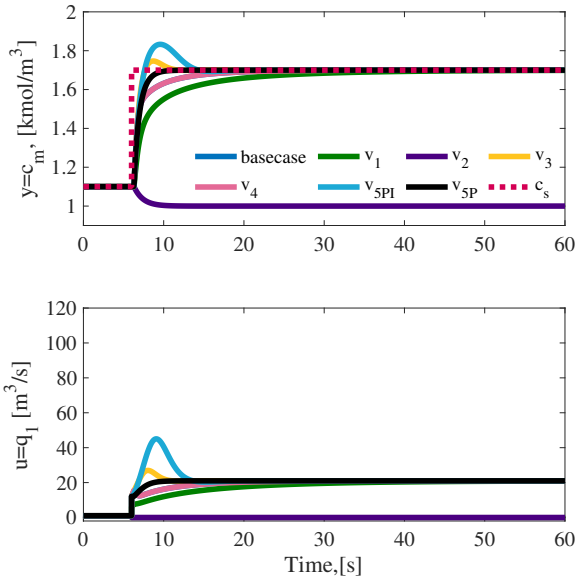

Figure B.5: Setpoint change from c_s = 1.1 kmol m^{-3} to c_s = 1.7 kmol m^{-3}, with measurement delay $\theta = 0.3$ s.

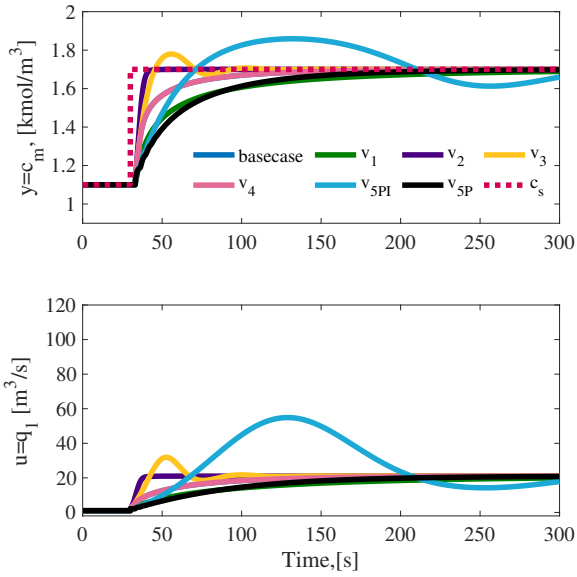

Figure B.6: Setpoint change from c_s = 1.1 $\mathrm{kmol\,m^{-3}}$ to c_s = 1.7 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 3 s.

B.2 Model error(Input gain mismatch)

B.2.1 Setpoint changes

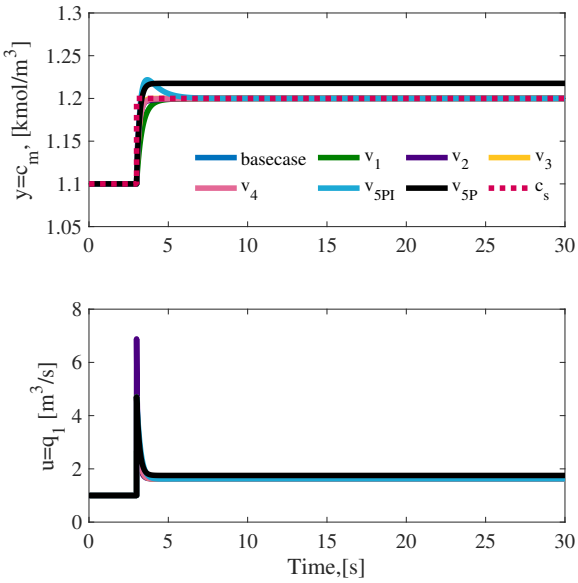

Figure B.7: Setpoint change from c_s = 1.1 $\mathrm{kmol\,m^{-3}}$ to c_s = 1.2 $\mathrm{kmol\,m^{-3}}$, with an input gain mismatch of 2..

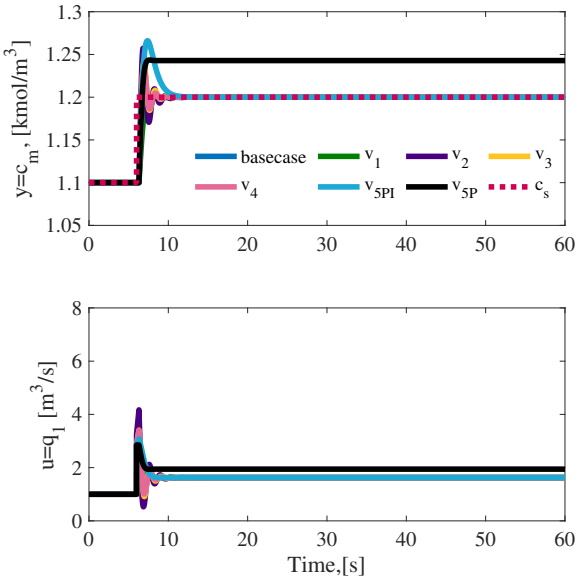

Figure B.8: Setpoint change from c_s = 1.1 $\mathrm{kmol\,m^{-3}}$ to c_s = 1.2 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 0.3 s, and an input gain mismatch of 2.

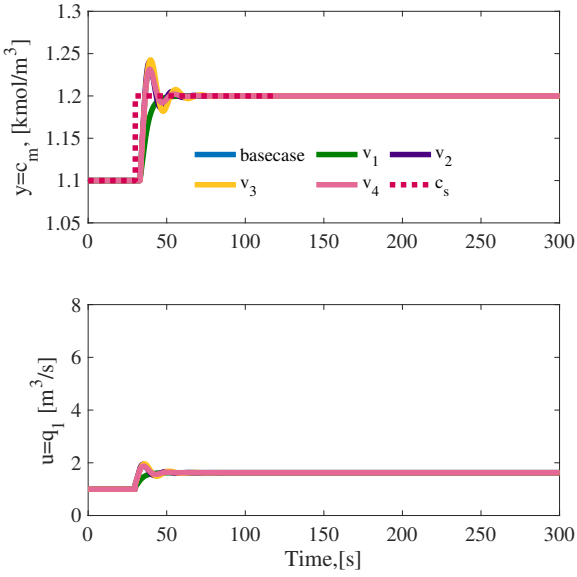

Figure B.9: Setpoint change from c_s = 1.1 $\mathrm{kmol\,m^{-3}}$ to c_s = 1.2 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 3 s, and an input gain mismatch of 2.

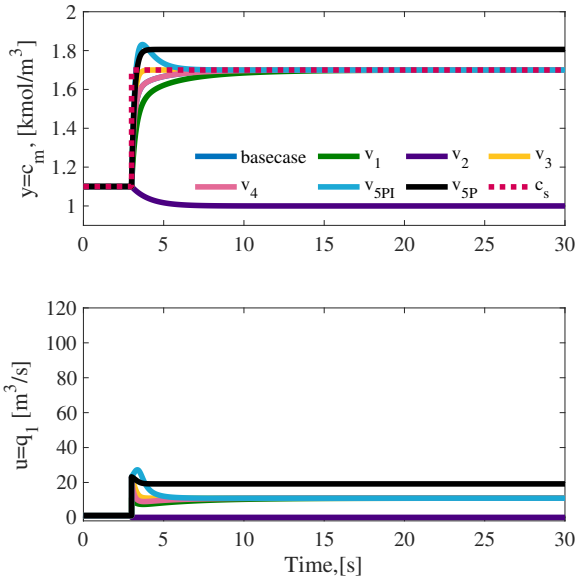

Figure B.10: Setpoint change from c_s = 1.1 $\mathrm{kmol\,m^{-3}}$ to c_s = 1.7 $\mathrm{kmol\,m^{-3}}$, with an input gain mismatch of 2.

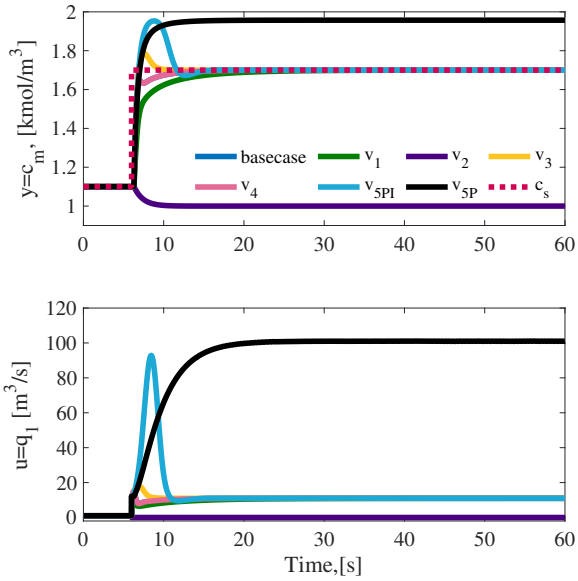

Figure B.11: Setpoint change from c_s = 1.1 $\mathrm{kmol\,m^{-3}}$ to c_s = 1.7 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 0.3 s, and an input gain mismatch of 2.

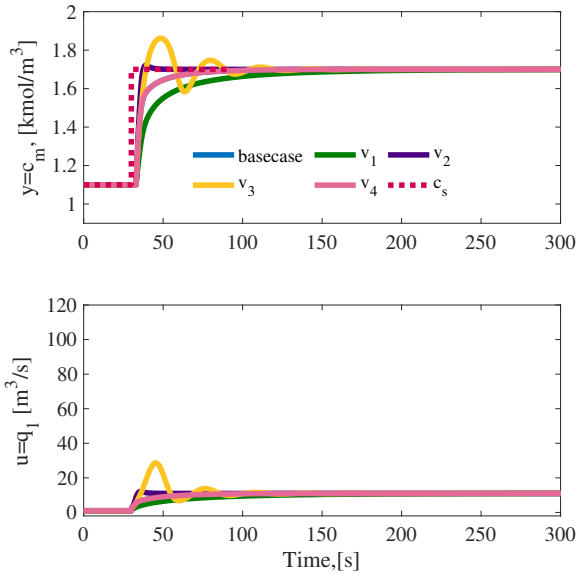

Figure B.12: Setpoint change from c_s = 1.1 $\mathrm{kmol\,m^{-3}}$ to c_s = 1.7 $\mathrm{kmol\,m^{-3}}$, with measurement delay θ = 3 s, and an input gain mismatch of 2.

Appendix C

Excerpts from Matlab code

C.1 Case study 1: Extraction process

C.1.1 Main file

```
% Task   : Transformed MVs. SISO Extraction casestudy
% compare
% Date   : 24. April 2020
% Author: Simen Bjorvand
% TODO   :
% Var    : MVs=[]; CVs=[]; States=[]  DV=[].

clc;clear;%close all
%% Plotting options
set(0,'DefaultTextFontName','Times',...
'DefaultTextFontSize',16,...
'DefaultAxesFontName','Times',...
'DefaultAxesFontSize',16,...
'DefaultLineLineWidth',2,...
'DefaultLineMarkerSize',7.75,...
'DefaultStairLineWidth',2)
%% Colors
blue    = [0 0.4470 0.7410];
red     = [0.83 0 0.33];
green   = [0 0.5 0];
indigo  = [0.2930 0 0.5078];
lilac   = [0.7383    0.5898    0.7930];
hotpink = [0.8945    0.4102    0.5898];
cerulean = [0.1094    0.6641    0.8359];

```

```
26 %% Parameters
27 %initial values for states
28 par.ME0 = 1e4;
29 par.MR0 = 1e4;
30 par.xAE0 = 3/14;
31 par.xWE0 = 1/14;
32
33 %initial values for input and disturbance variables
34 par.F0 = 100;
35
36 par.S0 = 100;
37 par.xAF0 = 0.3;
38
39 %other parameters
40 par.K = 1/3;
41 par.Kc1 = -1;
42 par.Kc2 = -1;
43 par.E0 = 140;
44 par.R0 = 60;
45
46 %tau, A and transformed input bias
47 par.tau = par.ME0/par.E0;
48 v0 = 0;         %update v0 in some help function later
49
50
51
52 %% calculate A = df/dx (x*,u*,d*)
53
54
55
56 %% Set up simulation
57 par.p    = [par.K      %1
58             par.Kc1    %2
59             par.Kc2    %3
60             par.E0     %4
61             par.R0     %5
62             par.ME0    %6
63             par.MR0    %7
64             0          %8 parameter indicating the structure
65             0];        %9 parameter indicating CV transform
                  % Parameter list
66
67 x0       = [par.ME0    %1
68             par.MR0    %2
69             par.xAE0   %3
```

```
70              par.xWE0];%4
                   % initial states
71
72  u0      = par.F0;                                          %
        physical MVs bias
73
74  d0      = [par.S0;par.xAF0];
                                          % DVs bias
75
76  y0      = stateToOutput(x0,u0,d0,par.p);
77
78  %% variables which will be varied to make up the 30 cases
79  tsim    = 400;                                             % [s],
        simulation time
80
81  tstep_y = tsim*0.02;                        % [s], step time for
        setpoint change
82
83  tstep_d = tsim*0.02;                           % [s], step time
        for disturbances
84
85
86
87  ys = y0;
88  ys = 0.5;
89
90  d1 = 1.0*par.S0;
        % change in DV 1
91  d2 = 1.0*par.xAF0;
        % change in DV 2
92
93  ds = [d1;d2];
94
95
96  tauC = 10;
97
98
99  %input gain error
100 Ku = [1.0;1.0];
101
102
103
104 %% Simulation. Get PID tuning, and run simulations
105 [Kc,KI,par] = tunings(par,tauC,1,false);
106 y0 = stateToOutput(x0,u0,d0,par.p);
```

```
107 out = sim('Extraction');
108 IAE = out.IAE(end)
109
110 colorList = {blue,green,hotpink,lilac};
111 %% plotting
112 fig=figure('visible','on');
113
114 subplot(311)
115 plot(out.time,out.x(:,3))
116 % ylim([0,0.5])
117 hold on
118 plot(out.time,out.ys,'--','color',red)
119 % plot(out.time,out.y)
120 subplot(312)
121 plot(out.time,out.u)
122 hold on
123 subplot(313)
124 plot(out.time,out.v)
125 hold on
```

C.1.2 Model file

```
1  function dxdt = model(x,u,d,p)
2  ME = x(1);    %[mol] mass of organic (extract) layer (acid,
       solvent, water)
3  MR = x(2);   %[mol] mass water (raffinate) layer (assumed
       to be pure water)
4  xAE = x(3);      % [-] mole fractions of acid in organic
       layer (extract E)
5  xWE = x(4);     % [-] mole fractions of water in organic
       layer (extract E)
6
7  F = u;                %[mol/s] water and acid feed
8
9  S = d(1);                                        % [mol/s]
       organic solvent feed
10 xAF = d(2);                          % [-] mole frocation of
       acid in feed (F)
11
12 K = p(1);                  % [-] water acid equilibrium
       constant in extract
13 Kc1 = p(2);               % [1/s] proportional gain for
       holdup p-controller
14 Kc2 = p(3);     % [1/s] proportional gain for raffinate
       holdup p-controller
15
```

```
16 E0 = p(4);                    % [mol/s] steady state bias for
       extrate outflow
17 R0 = p(5);                    % [mol/s] steady state bias for
      raffinate outflow
18
19 MEs = p(6);                                % [mol] extrate
      inventory setpoint
20 MRs = p(7);                               % [mol] raffinate
      inventory setpoint
21
22 %% mathematical model
23
24 deltaE = Kc1*((MEs-ME)+(MRs-MR));
25 deltaR = Kc2*(MRs-MR);
26
27 E = E0 + deltaE;
28 R = R0 + deltaR;
29
30 dMEdt = (1+K)*xAF*F + S - E;
31
32 dMRdt = (1-(1+K)*xAF)*F - R;
33
34 dxAEdt = (1/ME)*(xAF*(1-xAE*(1+K))*F - xAE*S);
35
36 dxWEdt = K*dxAEdt;
37
38 dxdt = [dMEdt;dMRdt;dxAEdt;dxWEdt];
39
40 logical = (dxdt == inf);
41
42 for i = 1:4
43     if(logical(i))
44         dxdt(i) = 1e7;
45     end
46 end
47
48 logical2 = (dxdt == -inf);
49
50 for i = 1:4
51     if(logical2(i))
52         dxdt(i) = -1e7;
53     end
54 end
55
56
```

```
57 end
```

C.1.3 Input calculation file

```
1  function u = calculationBlock(x,v,d,p)
2  ME = x(1);    %[mol] mass of organic (extract) layer (acid,
       solvent, water)
3  MR = x(2);   %[mol] mass water (raffinate) layer (assumed
     to be pure water)
4  xAE = x(3);       % [-] mole fractions of acid in organic
     layer  (extract E)
5  xWE = x(4);      % [-] mole fractions of water in organic
     layer  (extract E)
6
7  S = d(1);                                          % [mol/s]
     organic solvent feed
8  xAF = d(2);                          % [-] mole frocation of
     acid in feed (F)
9
10 K = p(1);                    % [-] water acid equilibrium
     constant in extract
11 Kc1 = p(2);                % [1/s] proportional gain for
     holdup p-controller
12 Kc2 = p(3);      % [1/s] proportional gain for raffinate
     holdup p-controller
13
14 E0 = p(4);                    % [mol/s] steady state bias for
       extrate outflow
15 R0 = p(5);                % [mol/s] steady state bias for
     raffinate outflow
16
17 MEs = p(6);                                    % [mol] extrate
     inventory setpoint
18 MRs = p(7);                                  % [mol] raffinate
     inventory setpoint
19
20 structure = p(8);
21
22 CVtransform = p(9);
23
24 deltaE = Kc1*((MEs-ME)+(MRs-MR));
25 deltaR = Kc2*(MRs-MR);
26
27 E = E0 + deltaE;
28 R = R0 + deltaR;
29
```

```
30
31 switch structure
32     case 1
33         u =v*E/xAF;
34     case 2
35         u = (1+K)*v*E + R;
36     case 3
37         u = (1+K)*v/(1-(1+K)*v)*S + R;
38     case 4
39         u = (ME*v + xAE*E)/xAF;
40     case 5
41         u = (1+K)*(ME*v + xAE*E) + R;
42     case 6
43         u = ((1+K)*ME*v+R*(1-(1+K)*xAE)+(1+K)*xAE*S)/(1-(1+
               K)*xAE); %vFL3#
44     case 7
45         u = ME*v/xAF;
46     case 8
47         u = ME*v*(1+K)+R;
48     case 9
49         u = ((E/ME)-(E0/MEs))*ME*xAE/xAF+v*ME/xAF;
50         if CVtransform == 1
51             u =v*E/xAF*MEs/E0;
52         end
53     case 10
54         u = (1+K)*ME*(((E/ME)-(E0/MEs))*xAE + v)+R;
55         if CVtransform == 1
56             u = (1+K)*v*(MEs/E0)/(1-(1+K)*v*(MEs/E0))*S + R
                   ;
57         end
58     case 11
59         u = ((1+K)*ME*(v-E0/MEs*xAE)+R*(1-(1+K)*xAE)+(1+K)*
               xAE*S)/(1-(1+K)*xAE);
60     otherwise
61         u = 100;
62 end
63 end
```

C.2 Case study 2: Mixing tank

C.2.1 Main file

```
1 % Task  : Transformed MVs. SISO mixing tank
2 % compare
3 % Date  : 14 January 2020
4 % Author: SB
```

```
% TODO   :
% Var    : MVs=[q1;q2]; CVs=[M;c]; DV=[q;c1;c2].

clc;clear;%close all
% profile on
%% Plotting options
set(0,'DefaultTextFontName','Times',...
'DefaultTextFontSize',16,...
'DefaultAxesFontName','Times',...
'DefaultAxesFontSize',16,...
'DefaultLineLineWidth',2,...
'DefaultLineMarkerSize',7.75,...
'DefaultStairLineWidth',2)
%% Colors
blue    = [0 0.4470 0.7410];
red     = [0.83 0 0.33];
green   = [0 0.5 0];
indigo  = [0.2930 0 0.5078];
lilac   = [0.7383    0.5898    0.7930];
hotpink = [0.8945    0.4102    0.5898];
cerulean = [0.1094    0.6641    0.8359];
%% Parameters
par.M0   = 10;                                          % [
    m3], tank holdup

par.q10  = 1;                                          % [m3
    /s] inlet flow 1
par.q20  = 9;                                          % [m3
    /s] inlet flow 2

par.taur  = par.M0/(par.q10+par.q20);                  % [s]
    tank residence time

par.c0   = 1.1;                                % [], nominal
    outlet concentration
par.c10   = 2;                 % [kg/m3], nominal inlet flow
     1 concentration
par.c20   = 1;                 % [kg/m3], nominal inlet flow
     2 concentration

par.A   = -(par.q10+par.q20);

par.v_0 = 0;

%% variables which will be varied to make up the 30 cases
```

```
43 tsim     = 600;                                          % [s],
       simulation time
44
45 tstep_c = tsim*0.1;      % [s], step time for concentration
       setpoint changes
46 tstep_d = tsim*0.1;                          % [s], step time
       for disturbances
47
48
49 par.gain_error  =  1.0;                        % input error
       gain proportions
50 par.disturbance_error = eye(4);        % disturbance gain
       error, proportions
51 par.disturbance_error(1,1) = 1.2;
52 par.disturbance_error(2,2) = 1.2;
53 par.disturbance_error(3,3) = 1.2;
54
55
56 cs1 =  1.1;                      % setpoint change for
       concentration setpoint
57 % cs1 =  1.7;
58
59 % d1 = 1.0*par.q20;
                                                    % change in DV
60 % d1 = 2.0*par.q20;
                                                    % change in DV
61 d1 = 0.5*par.q20;
                                                      % change in
       DV
62 d2 = 1.0*par.c10;
63 % d2 = 1.5*par.c10;
64 d3 = 1.0*par.c20;
65 % d3 = 0.5*par.c20;
66 d4 = 1.0*par.M0;
67 ds = [d1;d2;d3;d4];
68
69
70 par.theta  = 3.0*par.taur;
                                                    % delay
71 % par.theta  = 0.3*par.taur;
                                                   % delay
72 % par.theta  = 0.0;
                                                          % delay
73
74
```

```
%% Set up simulation
par.p  = [par.A;0];
x0      = par.c0;                          % initial condition
    for integration
u0      = par.q10;                                          %
    physical MVs bias
d0      = [par.q20;par.c10;par.c20;par.M0];
                        % DVs bias
dv      = 4;                        % step in inputs for open
    loop simulation


%% Simulation. Get PID tuning, and run simulations
[Kc,tauI,KI,par]=case3(par);
out=sim('mixingTank');
IAE = out.IAE(end)


%% plotting
fig=figure('visible','on');
subplot(3,1,1)
p1=plot(out.time,out.cm,'color',blue);
hold on
p2=plot(out.time,out.c_setpoint,':','color',red);
% ylim([1,1.2])
ylabel('y=c_{m}, [kmol/m^3]')
% xline(tstep_d,':','color',green,'LineWidth',2);
legend([p1,p2],'y = c_{m}','c_s','Location','southeast')


subplot(3,1,2)
plot(out.time,out.u)
ylabel('u=q_1 [m^3/s]')
xlabel('Time,[s]')
% xline(tstep_d,':','color',green,'LineWidth',2);

subplot(3,1,3)
plot(out.time,out.v)


%% have all the cases written down here as help functions

function [Kc,tauI,KI,par]=case0(par)
k     = (par.c10-par.c0)/(par.q10+par.q20);        % process
    gain from v to c
```

```
115 tau  = par.M0/(par.q10+par.q20);          % time constant
    for concentration
116 par.p(2) = 0;
117
118 par.v_0 = par.q10;                        % transformed
    input bias case 1
119
120
121 tauC = par.theta;          % desired closed loop time
    constant for flow loop
122 if tauC==0
123     tauC = 0.3*par.taur;
124 end
125
126
127 % PI-controller for concentration
128 Kc   = 1/k*tau/(tauC+par.theta);                % first-
    order plus delay
129 tauI = min(tau,4*(tauC+par.theta));
130 KI   = Kc/tauI;
131 end
132
133 function [Kc,tauI,KI,par]=case1(par)
134 k    = 1/(par.q10+par.q20);                     % process
    gain from v to c
135 tau  = par.M0/(par.q10+par.q20);          % time constant
    for concentration
136 par.p(2) = 1;
137
138
139 u_nom = par.q10;
140 par.v_0 = u_nom*par.c10 + par.q20*par.c20;  % transformed
    input bias case 2
141
142
143
144 tauC = par.theta;          % desired closed loop time
    constant for flow loop
145 if tauC==0
146     tauC = 0.3*par.taur;
147 end
148
149
150 % PI-controller for concentration
151 Kc   = 1/k*tau/(tauC+par.theta);                % first-
```

```
    order plus delay
152 tauI = min(tau,4*(tauC+par.theta));
153 KI   = Kc/tauI;
154 end
155
156 function [Kc,tauI,KI,par]=case2(par)
157 k    = 1;                                          % process
    gain from v to c
158 tau  = par.M0/(par.q10+par.q20);                   % time constant
    for concentration
159 par.p(2) = 2;
160
161
162 u_nom = par.q10;
163 par.v_0 = (u_nom*par.c10+par.q20*par.c20)/(u_nom+par.q20);
164                                                    % transformed
                                                       input bias
                                                       case 2
165
166 tauC = par.theta;          % desired closed loop time
    constant for flow loop
167 if tauC==0
168     tauC = 0.3*par.taur;
169 end
170
171
172 % PI-controller for concentration
173 Kc   = 1/k*tau/(tauC+par.theta);                   % first-
    order plus delay
174 tauI = min(tau,4*(tauC+par.theta));
175 KI   = Kc/tauI;
176 end
177
178 function [Kc,tauI,KI,par]=case3(par)
179 k    = -1/par.A;                                   % process
    gain from v to c
180 tau  = -par.M0/par.A;                              % time constant
    for concentration
181 par.p(2) = 3;
182
183
184 u_nom = par.q10;
185 par.v_0 = u_nom*(par.c10-par.c0) + par.q20*(par.c20-par.c0)
     - par.c0*par.A;
186                                                    % transformed
```

```
                                                    input bias
                                                    case 4
187
188 tauC = 1.0*par.theta;      % desired closed loop time
    constant for flow loop
189 if tauC==0
190     tauC = 0.3*par.taur;
191 end
192
193
194 % PI-controller for concentration
195 Kc   = 1/k*tau/(tauC+1*par.theta);                % first-
    order plus delay
196 tauI = min(tau,4*(tauC+1*par.theta));
197 KI   = Kc/tauI;
198 end
199
200 function [Kc,tauI,KI,par]=case4(par)
201 k    = 1;                                         % process
    gain from v to c
202 tau  = -par.M0/par.A;                          % time constant
    for concentration
203 par.p(2) = 4;
204
205
206 u_nom = par.q10;
207 par.v_0 = -(u_nom*(par.c10-par.c0)+par.q20*(par.c20-par.c0)
    -par.c0*par.A)/par.A;
208                                                 % transformed
                                                    input bias
                                                    case 4
209
210 tauC = 1.0*par.theta;      % desired closed loop time
    constant for flow loop
211 if tauC==0
212     tauC = 0.3*par.taur;
213 end
214
215
216 % PI-controller for concentration
217 Kc   = 1/k*tau/(tauC+par.theta);                  % first-
    order plus delay
218 tauI = min(tau,4*(tauC+par.theta));
219 KI   = Kc/tauI;
220 end
```

```
221
222 function [Kc,tauI,KI,par]=case5(par)
223 k     = (par.q20*(par.c10-par.c0))/(par.q10+par.q20);
224                                                     % process
                                                          gain
                                                          from v
                                                          to c
225 tau   = par.M0/(par.q10+par.q20);           % time constant
        for concentration
226 par.p(2) = 5;
227
228
229 u_nom = par.q10;
230 par.v_0 = u_nom/par.q20;                      % transformed
        input bias case 5
231
232
233 tauC = par.theta;          % desired closed loop time
        constant for flow loop
234 if tauC==0
235     tauC = 0.3*par.taur;
236 end
237
238
239
240 % PI-controller for concentration
241 Kc    = 1/k*tau/(tauC+par.theta);                   % first-
        order plus delay
242 tauI = min(tau,4*(tauC+par.theta));
243 KI    = Kc/tauI;
244 end
245
246 function [Kc,tauI,KI,par]=case6(par,P)
247 k     = 1/par.M0;                                   % process
        gain from v to c
248
249 par.p(2) = 6;
250
251
252 u_nom = par.q10;
253 par.v_0 = u_nom*(par.c10-par.c0) + par.q20*(par.c20-par.c0)
        ;
254                                                   % transformed
                                                          input bias
                                                          case 6
```

```
255
256 tauC = par.theta;          % desired closed loop time
    constant for flow loop
257 if tauC==0
258     tauC = 0.3*par.taur;
259 end
260
261 % PI-controller for concentration
262 Kc    = 1/k/(tauC+par.theta);                    %
    Integrated plus delay
263 tauI = 4*(tauC+par.theta);
264 KI    = Kc/tauI;
265 if P
266     KI    = 0;
267 end
268 end
```

C.2.2 Model file

```
1 function ode = model(x,u,d,p)
2 % model for level and concentration
3
4 c1     = d(2)*1;                          %[kg/kg] inlet flow
     1 concentration
5 c2     = d(3)*1;                          %[kg/kg] inlet flow
     2 concentration
6 M      = d(4)*1;                                  % [
    kg], tank holdup
7
8
9 c      = x;                                      % [kg/kg
    ], concentration
10
11 q1    = u;                                      % [kg/s
    ], physical input
12
13 q2    = d(1);                                     % [kg
    /s], disturbance
14
15
16
17 dcdt = q1/M*(c1-c) + q2/M*(c2-c);                  %
    component balance
18
19 ode = dcdt;
```

```
    % model
20
21 end
```

C.3 Case study 3: Continuous stirred tank reactor

C.3.1 Main file

```
1  % Task  : Transformed MVs. SISO flash tank casestudy
2  % compare
3  % Date  : 5. March 2020
4  % Author: SB
5  % TODO  :
6  % Var   : MVs=[zd]; CVs=[p]; States=[x]  DV=[].
7
8  clc;clear;%close all
9  % profile on
10 %% Plotting options
11 set(0,'DefaultTextFontName','Times',...
12 'DefaultTextFontSize',16,...
13 'DefaultAxesFontName','Times',...
14 'DefaultAxesFontSize',16,...
15 'DefaultLineLineWidth',2,...
16 'DefaultLineMarkerSize',7.75,...
17 'DefaultStairLineWidth',2)
18 %% Colors
19 blue     = [0 0.4470 0.7410];
20 red      = [0.83 0 0.33];
21 green    = [0 0.5 0];
22 indigo   = [0.2930 0 0.5078];
23 lilac    = [0.7383    0.5898    0.7930];
24 hotpink = [0.8945    0.4102    0.5898];
25 cerulean = [0.1094    0.6641    0.8359];
26
27 %% Parameters
28 par.ca0 = 0.05;                    % [kmol/m3]
       Concentration of component A
29 par.V0 = 4;                                    % [m3] Hold
       up CSTR reactor
30 par.q10 = 1;                                     % [m3/
       min] Inlet stream
31 par.q20 = 1;                                    % [m3/
       min] Outlet stream
32 par.caf0 = 1;              % [kmol/m3] Concentration of
       component A in feed
```

```
33 par.k0 = 95;                                % [m3/kmol/min]
       Reaction rate constant
34
35
36 par.A22 = -(par.q10/par.V0+2*par.k0*par.ca0);                %
       A22 = df2/dca|*
37 par.A21 = -(par.q10/par.V0^2)*(par.caf0-par.ca0);
      % A21 = df2/dV|*
38 par.B21 = (par.caf0-par.ca0)/par.V0;                        %
       B21 = df2/dq1|*
39
40 par.taur = par.V0/par.q10;
41
42 par.v10 = 0;                                                 %
       transformed MV 1 bias
43 par.v20 = 0;                                                 %
       transformed MV 2 bias
44
45 %% calculate A = df/dx (x*,u*,d*)
46
47
48
49 %% Set up simulation
50 par.p    = [par.A22    %1
51             0];        %2
                  % Parameter list
52 x0       = [par.V0;par.ca0];              % initial condition
       for integration
53 y0       = [par.V0;par.ca0];
      % initial output
54 u0       = [par.q10;par.q20];                                %
       physical MVs bias
55 d0       = [par.caf0;par.k0];
                                           % DVs bias
56
57
58 %% variables which will be varied to make up the 30 cases
59 tsim     = 400*1;                                            % [s],
       simulation time
60
61 tstep_y1 = tsim*0.2;     % [s], step time for concentration
       setpoint changes
62 tstep_y2 = tsim*0.2;     % [s], step time for concentration
       setpoint changes
63 tstep_d = tsim*0.2;                          % [s], step time
```

```
    for disturbances
% tsim = 45;


cas =  2.0*par.ca0;             % setpoint change for
    concentration setpoint
Vs  =  1.0*par.V0;              % setpoint change for
    concentration setpoint

d1 = 1.0*par.caf0;
                                                % change in DV
d2 = 1.0*par.k0;
                                                  % change in
    DV

ds = [d1;d2];

par.thetaca = 1.0*par.taur;

%input gain error
Ku = [2.0;1.0];



%% Simulation. Get PID tuning, and run simulations
tic
[Kc,tauI,KI,par]=transformedcase(par);
out1=sim('CSTR');
toc
tic
[Kc,tauI,KI,par]=basecase(par);
out2=sim('CSTR');
toc
tic
[Kc,tauI,KI,par]=basecase_decouple2(par);
out3=sim('CSTR_decouple2'); %this is the best decoupling
toc



%% plotting
fig=figure('visible','on');
subplot(321)
p1=plot(out1.time,out1.V,'color',blue);
```

```
hold on
p2=plot(out2.time,out2.V,'-.','color',hotpink);
p3=plot(out3.time,out3.V,':','color',green);
% p4=plot(out4.time,out4.V,'-.','color',indigo);
% hold on
% p2=plot(out.time,out.cbs,':','color',red);
% ylim([1,1.2])
ylabel('x=V')
% xline(tstep_d,':','color',green,'LineWidth',2);

subplot(322)
p5=plot(out1.time,out1.ca,'color',blue);
hold on
p6=plot(out2.time,out2.ca,'-.','color',hotpink);
p7=plot(out3.time,out3.ca,':','color',green);
% p8=plot(out4.time,out4.ca,'-.','color',indigo);
ylabel('x=c_{a}')

subplot(323)
plot(out1.time,out1.u(:,1))
hold on
plot(out2.time,out2.u(:,1),'-.','color',hotpink)
plot(out3.time,out3.u(:,1),':','color',green)
% plot(out4.time,out4.u(:,1),'-.','color',indigo)
ylabel('u1=q1 [-]')
xlabel('Time,[s]')
subplot(324)
plot(out1.time,out1.u(:,2))
hold on
plot(out2.time,out2.u(:,2),'-.','color',hotpink)
plot(out3.time,out3.u(:,2),':','color',green)
% plot(out4.time,out4.u(:,2),'-.','color',indigo)
ylabel('u2=q2 [-]')
xlabel('Time,[s]')


subplot(325)
plot(out1.time,out1.v(:,1))
ylabel('v1 [-]')
subplot(326)
plot(out1.time,out1.v(:,2))
ylabel('v2 [-]')

% subplot(427)
% plot(out3.time,out3.u1)
```

```
148 % subplot(428)
149 % plot(out3.time,out3.u2)
```

C.3.2 Model file

```
1  function ode = model(x,u,d,p)
2  V = x(1);                                          % [m3] Hold
       up CSTR reactor
3  ca = x(2);                              % [kmol/m3]
       Concentration of component A
4
5  q1 = u(1);                                          % [m3/
       min] Inlet stream
6  q2 = u(2);                                         % [m3/
       min] Outlet stream
7
8  caf = d(1);                  % [kmol/m3] Concentration of
       component A in feed
9  k = d(2);                                % [m3/kmol/min]
       Reaction rate constant
10
11 %% model equations
12
13 dVdt  = q1-q2;
14
15 dcadt = (q1/V)*(caf-ca)-k*ca^2;
16
17 error = 0;
18 if (dVdt == inf || dcadt == inf)
19     dVdt = 0;
20     dcadt =0;
21     error =-1;
22 end
23 if (isnan(dVdt) || isnan(dcadt))
24     dVdt = 0;
25     dcadt =0;
26     error =-1;
27 end
28
29 ode = [dVdt;dcadt;error];
                                                    % model
30
31 end
```

C.3.3 Input calculation file

```
function u = CalcBlock(x,v,d,p)
% model for level and concentration
V = x(1);                                          % [m3] Hold
    up CSTR reactor
ca = x(2);                              % [kmol/m3]
    Concentration of component A

caf = d(1);                  % [kmol/m3] Concentration of
    component A in feed
k = d(2);                                % [m3/kmol/min]
    Reaction rate constant

A22 = p(1);
switch p(2)
    case 0  %base case
        u = v;
%         u(1) = v(2);
%         u(2) = v(1);
    case 1
        T = [1,-1;(caf-ca)/V,0];
        b = [0;-ca*(k*ca+A22)];
        Tinv = [0,V/(caf-ca);-1,V/(caf-ca)];
        u = Tinv*(v-b);
    otherwise
        u = v;
end
end
```

C.4 Case study 4: Condensation process

C.4.1 Main file

```
% Task   : Transformed MVs. HEX
% compare
% Date   : 12. May 2020
% Author: Simen Bjorvand
% TODO   :
% Var    : MVs=[]; CVs=[]; States=[]  DV=[].

clc;clear;%close all
% profile on
%% Plotting options
set(0,'DefaultTextFontName','Times',...
'DefaultTextFontSize',16,...
'DefaultAxesFontName','Times',...
'DefaultAxesFontSize',16,...
```

```
15 'DefaultLineLineWidth',2,...
16 'DefaultLineMarkerSize',7.75,...
17 'DefaultStairLineWidth',2)
18 %% Colors
19 blue     = [0 0.4470 0.7410];
20 red      = [0.83 0 0.33];
21 green    = [0 0.5 0];
22 indigo   = [0.2930 0 0.5078];
23 lilac    = [0.7383    0.5898    0.7930];
24 hotpink = [0.8945    0.4102    0.5898];
25 cerulean = [0.1094    0.6641    0.8359];
26
27 %% Parameters
28 %initial values for states
29 par.Tinit = 350;                                    % [K]
       temperature of cold side
30
31 %initial values for input and disturbance variables
32 par.zinit = 0.1;                                   % [-] steam
       inlet valve position
33
34 par.T0init = 300;                                  % [K] cold side
       inlet temperature
35 par.winit = 10;                                              % [kg/s
       ] cold side flow
36 par.Td0init = 390;                                % [K] steam
       uppstreams temperature
37
38
39 %other parameters
40 par.m = 1000;                                           % [kg] hold
        up og cold side
41 par.cp = 4;                                  % [kJ/kgK] heat
       capacity of cold side
42 par.U = 1;                                    % [kW/m2K] heat
       transfer coefficient
43 par.A = 100;                                          % [m2] Area
       of heat transfer
44 par.lambda = 2200;
45                 % [kJ/kg] The enthalpy of vaporization of
                       medium in hot side
46 par.R = 8.314;                                     % [kJ/kmolK]
       ideal gas constant
47 par.pdref = 1.0;                  % [Bar] saturated steam
       refference pressure
```

```
48 par.Tdref = 370;                 % [K] saturated steam
       refference temperature
49 par.Mm = 18;                                  % [Kg/kmol]
       Molar mass of steam
50
51 Tdinit = 370;                                   % [K]
       Temperature of hot side
52 pdinit = 1.0;                                   % [Bar]
       Pressure of hot side
53
54 %parameters derived from the other parameters
55 Rs = par.R/par.Mm;
56 par.pd0init = par.pdref*exp(-par.lambda/Rs*(1/par.Td0init
       -1/par.Tdref));
57                                                % [Bar] steam
                                                      uppstreams
                                                      pressure
58
59 Qinit = par.U*par.A*(Tdinit - par.Tinit);
60 wdinit = Qinit/par.lambda;
61 par.k = wdinit/(par.zinit*(par.pd0init-pdinit));
62                                                % [kg/Bar*s]
                                                      valve
                                                      coefficient
63
64 % find different taus, non are "correct"
65 par.tau1 = par.m/par.winit;
66 par.tau2 = par.cp*par.m/(par.cp*par.winit+par.U*par.A);
67
68 %calculate As = df/dx|* and B = df/dz|*
69 k1 = par.k*par.zinit*par.lambda^2*pdinit/(Rs*Tdinit^2);
70 par.k2 = k1/(par.U*par.A);
71 par.k3 = par.lambda*par.k/(par.U*par.A)*(par.pd0init-pdinit
       );
72
73 par.As = -(1/par.m)*(par.winit + (1/par.cp)*(k1/(1+par.k2))
       );
74
75 par.B = (par.lambda*par.k/(par.m*par.cp)) * (par.pd0init-
       pdinit)/(1+par.k2);
76
77
78 v0 = 0;                                  %update v0 in some
       help function later
79
```

```
%% Set up simulation
par.p    = [par.m        %1
            par.cp       %2
            par.U        %3
            par.A        %4
            par.lambda   %5
            par.k        %6
            par.R        %7
            par.pdref    %8
            par.Tdref    %9
            par.Mm       %10
            par.As       %11
            0            %12 indicates the structure which is
                 used
            0];          %13 indicates cascade and which
                variable is measured
                                                              %
                                                                 Parameter
                                                                 list

x0       = par.Tinit;
                                                 % initial state

u0       = par.zinit;                                          %
    physical MVs bias

d0       = [par.T0init
            par.winit
            par.pd0init];
                                                             % DVs
                 bias

y0       = x0;

dv       = 0.1;
%% variables which will be varied to make up the 30 cases
tsim     = 1000;                                              % [s],
     simulation time

tstep_y = tsim*0.02;                        % [s], step time for
     setpoint change
```

```
112 tstep_d = tsim*0.02;                                % [s], step time
        for disturbances
113
114
115
116 ys = y0;
117 % ys = 340;
118
119 d1 = 1.0*par.T0init;
        % change in DV 1
120 d2 = 1.0*par.winit;
        % change in DV 2
121 d3 = 1.1*par.pd0init;
        % change in DV 3
122
123 ds = [d1;d2;d3];
124
125
126 tauC  = 50;
127 theta = 50;
128 TC    = tauC+theta;
129
130
131 %input gain error
132 Ku = 1.0;
133
134 % ode = model(x0+11.6927910218494455,u0+0.9,d0,par.p)
135
136 %% Simulation. Get PID tuning, and run simulations
137 % [Kc,KI,par] = tunings(par,tauC,theta,6);
138 [Kc,KI,Kci,KIi,par] = cascade_tunings(par,tauC,theta,2,2);
139 tic
140 % out = sim('HEX');
141 out = sim('HEXcascade');
142 IAE = out.IAE(end)
143 toc
144 colorList = {blue,green,hotpink,lilac};
145 %% plotting
146 % fig=figure('visible','on');
147
148 subplot(311)
149 plot(out.time,out.T)
150 % ylim([0,0.5])
151 hold on
152 plot(out.time,out.ys,'--','color',red)
```

```
153 subplot(312)
154 plot(out.time,out.u)
155 hold on
156 subplot(313)
157 plot(out.time,out.v)
158 hold on
159
160
161
162 %% figure 2
163
164 % subplot(311)
165 % plot(out.time,out.T)
166 % % ylim([0,0.5])
167 % hold on
168 % plot(out.time,out.ys,'--','color',red)
169 % subplot(312)
170 % plot(out.time,out.Td)
171 % hold on
172 % plot(out.time,out.yis,'--','color',red)
173 % subplot(313)
174 % plot(out.time,out.u)
```

C.4.2 Model file

```
1  function out = model(x,u,d,p)
2  % model for level and concentration
3  T      = x;                                    % [K]
       temperature of cold side
4
5  z      = u;                                   % [-] steam
       inlet valve position
6
7  T0     = d(1);                                % [K] cold side
       inlet temperature
8  w      = d(2);                                          % [kg/s
       ] cold side flow
9  pd0    = d(3);                                % [Bar] steam
       uppstreams pressure
10
11 m      = p(1);                                      % [kg] hold
        up og cold side
12 cp     = p(2);                              % [kJ/kgK] heat
       capacity of cold side
13 U      = p(3);                               % [kW/m2K] heat
       transfer coefficient
```

```
14 A       = p(4);                                    % [m2] Area
       of heat transfer
15 lambda = p(5); % [kJ/kg] The enthalpy of vaporization of
       medium in hot side
16 k       = p(6);                                    % [kg/Bar*s]
       valve coefficient
17 R       = p(7);                                    % [kJ/kmolK]
       ideal gas constant
18 pdref   = p(8);                  % [Bar] saturated steam
       refference pressure
19 Tdref   = p(9);                  % [K] saturated steam
       refference temperature
20 Mm      = p(10);                                   % [Kg/kmol]
       Molar mass of steam
21 % model equation
22
23 % need to solve a implicite system of equations to find Td
       (use a help
24 % function)
25 Rs = R/Mm;
26
27 pd = @(Tdvar) pdref*exp(-(lambda/Rs)*(1/Tdvar -1/Tdref));
28
29 fun = @(Tdvar) k*z*lambda/(U*A)*(pd0-pd(Tdvar)) + T - Tdvar
       ;
30
31 options = optimset('Display','off','TolFun',1e-20,'TolX',1e
       -20,'MaxIter',1e4,'MaxFunEvals',1e10);
32
33 Td  = fsolve(fun,Tdref,options);
34
35 Q = U*A*(Td - T);
36
37 wd = Q/lambda;
38
39 dTdt = (1/m)*(w*(T0-T)+Q/cp);
40
41 out = [dTdt,wd,Td];
                                                    % model
42
43 end
```

C.4.3 Input calculation file

```
1 function u = calculationBlock(x,v,d,p)
2 T       = x;                                       % [K]
```

```
    temperature of cold side
3
4   T0      = d(1);                                 % [K] cold side
    inlet temperature
5   w       = d(2);                                         % [kg/s
    ] cold side flow
6   pd0     = d(3);                                 % [Bar] steam
    uppstreams pressure
7
8   m       = p(1);                                     % [kg] hold
     up og cold side
9   cp      = p(2);                               % [kJ/kgK] heat
    capacity of cold side
10  U       = p(3);                                % [kW/m2K] heat
    transfer coefficient
11  A       = p(4);                                     % [m2] Area
    of heat transfer
12  lambda = p(5); % [kJ/kg] The enthalpy of vaporization of
    medium in hot side
13  k       = p(6);                                  % [kg/Bar*s]
    valve coefficient
14  R       = p(7);                                 % [kJ/kmolK]
    ideal gas constant
15  pdref   = p(8);                    % [Bar] saturated steam
    refference pressure
16  Tdref   = p(9);                   % [K] saturated steam
    refference temperature
17  Mm      = p(10);                                % [Kg/kmol]
    Molar mass of steam
18
19  As      = p(11);                     % [some unit] df/dt|*
    linearized constant
20
21  structure = p(12);
22  cascade   = p(13);
23  Rs = R/Mm;
24
25
26  switch structure
27      case 0
28          u = v;
29      case {1,2,3,4,5,6,7,8,9,10}
30          switch structure
31              case 1 %%no cascade z
32                  Q = cp*(v*m + (w + As*m)*T - w*T0);
```

```
33                  Td = Q/(U*A) + T;
34              case 2 %%cascade with wd
35                  Q = cp*(v*m - w*T0 + (w-10)*T);
36                  Td = Q/(U*A) + T;
37              case 3 %%cascade with Td
38                  Td = cp*( v*m - w*T0 + (w-10)*T )/(U*A);
39                  Q = U*A*(Td-T);
40          end
41          switch cascade
42              case 0 %no cascade
43                  pd = pdref*exp(-(lambda/Rs)*(1/Td - 1/Tdref
                        ));
44                  if((pd0-pd)>1e-4)
45                      u = Q/(lambda*k*(pd0-pd));
46                  elseif pd > pd0
47                      u=1;
48                  else
49                      u=1;
50                  end
51              case 1 %cascade and wd is used as inner output
52                  u = Q/lambda;
53              case 2
54                  u = Td;
55          end
56      otherwise
57          u = 0.1;
58  end
59
60  end
```

Simen Bjorvand

Control structures with embedded process knowledge

NTNU
Norwegian University of
Science and Technology