NTNU
Norwegian University of
Science and Technology

A robust Multi-Loop tuning Procedure based at Succsessive Semidefinite Programming, that achieves optimal Performance and Failure Tolerance

Henrik Helgesen

Master of Science in Engineering Cybernetics
Submission date: June 2008
Supervisor: Ole Morten Aamo, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

Multi-loop PID controllers are widely used in the process industry even if more advanced controllers are available. Due to interactions between the loops, it is difficult to obtain optimal tuning of such systems. The objective of this project is to develop a tool for tuning multi-loop PID controllers that applies to linear MIMO systems. The tool may be based on available literature. The following tasks should be addressed.

• Literature survey: Present and discuss existing multi-loop PID tuning procedures

• Methodology: Develop a tuning procedure for multi-loop PID controllers in linear MIMO systems.

• Implementation: implement the tuning procedure in MATLAB.

• Application: Apply the tuning procedure to a suitable example.

• User interface: Use MATLAB GUI (GUIDE) to develop an interface that makes the tuning procedure user-friendly.

Assignment given: 14. January 2008
Supervisor: Ole Morten Aamo, ITK

Abstract

The desired properties of a multi-loop PID tuning procedure is to find some parameters that makes the plant robust, meet some desired performance requirements and guarantee failure tolerance.

A detailed literature survey of the different multi-loop PID tuning procedures is presented. Properties of the Independent design methods, Detuning methods, Sequential closing methods, Iterative or trial and error methods, Optimization methods and Relay feedback approaches are described in detail and discussed. Most of the tuning procedures result in a too conservative design without failure tolerance ensured.

It is shown how the integrity property may be achieved with a multi-loop H_∞ optimal tuning method. How to define and solve such a H_∞ optimal problem is presented. The desired properties of the multi-loop PID tuning procedure are obtained with this method. The method aim to solve a H_∞ optimization problem with Linear Matrix Inequality (LMI) constraints. The optimization problem is non-convex, so a Successive Semidefinite Programming (SSP) algorithm is used to find local solutions to the problem. Several initial points must be examined to aim for a global solution.

The SSP algorithm is implemented in MATLAB, and applied at a distillation column example. The implemented algorithm does not converge to a solution. Hence no simulation results that back up the theoretical work are presented.

Preface

By writing this thesis my knowledge of how plants are controlled in the process industry has increased. The challenges that the operators and the engineers face every day are severe and very difficult to solve. They really earn my deepest respect. It has been difficult, fun and valuable to dig into the literature, and try to find some improvements to this part of the special field.

My thanks goes first to Ole Morten Aamo and Hans Petter Bieker for great supervision throughout the semester. They always took the time to answer, even if some questions and subjects where small and simple.

The assignment where given by Morten Dalsmo and ABB. A special thanks goes to him for letting me have the opportunity to write about such an interesting and challenging subject as multi-loop PID tuning.

At last my thanks goes to Martin Kivle, Thomas Rognmo, Kjetil Hodne and Helle Lorentzen. They all contributed to big and small daily highlights at office G 232B. I hope I will find similar work environment in the future.

Contents

Chapter 1

Introduction

1.1 Background

1.1.1 Control structure in the industry

To be aware of the present challenges in process control, it is important to know today's industrial practice. The control configuration at an industrial plant can either be centralized or decentralized. In decentralized control several controllers are placed at different geographical places beyond the plant. This is in contrast to centralized control, where all measurements are lead into a central control unit.

Decentralized control remains popular in the industry despite of other advanced control syntheses. Examples are Model Predictive Control, outlined in [32], or strategies which lead to full multivariable controllers, found in [39]. The controller must be centralized if a full multivariable controller can be used.

In decentralized control each single measurement is paired with one single actuator. This results in N Single Input Single Output (SISO) loops. Hence decentralized control is also called a multi-loop controller. Traditionally these SISO controllers have a PI or a PID structure, shown in Section 2.1.2. This is also present in modern gas processing plants, such as Ormen Lange where all the 800 controllers are PI or PID[15].

In [29] and [39] at page 421, several reasons for this conservatism is outlined.

- A full multivariable controller requires accurate plant models. Modeling is both time consuming and expensive. Decentralized controllers are usually tuned one at a time, sometimes on-line, with minimum modeling effort.
- The tuning parameters have a direct and localized effect, so the controllers are easy to implement to the physical plant.
- Even with faster computers, the advantage of small computation load is still existing.
- They may be brought gradually into service during startup and taken gradually out of service during shutdown.

- It is possible to maintain robust stability, failure tolerance and the insensitivity to uncertainty in the inputs.
- The controllers are well known to operators, so they may do the retuning at their own if the process conditions change.

1.2 Motivation

1.2.1 The need for process control

Some general aims of industrial automation is listed in [45] at page 30. Process control is desired for safety reasons, and to make the workplace environment more humane. If a controller stabilizes a natural unstable plant, like a nuclear reactor, it is obvious that settings and configurations which ensure the safeness is of the most important concern.

A desire to minimize the environmental impact is more topical for the industry than ever. Good control strategies are important when pollution should be minimized.

The general aim for a manufacturer is to maximize the profit. Basic optimization theory outlined in [19] states that the optimal value often lies at one or several constraints. Hence the most profitable operation is obtained when a process is running at one or several of its capabilities. These constraints are associated with direct or overhead costs, product and quality specifications, or pollution minimization. Physical constraints appear at the inputs to the process as saturation characteristics. Examples are valves with finite range of adjustment, flow rates with maximum values due to fixed pipe diameters or control surfaces with limited deflection angles.[32] For instance, if a product has to be at least a certain quality in order to be useful, the cost can be minimized by making its quality just sufficient. This constraint is soft, and could be included in the objective function by a penalty term, as described at in chapter 15 in [19]. It is not critical for the system functionality if some product of bad quality is produced over a short time period. An example of a hard constraint is the compressors 'surge line'. It is desired that the operating point is as close to the line as possible, but if it operates at the wrong side, the compressor will collapse[22].

Due to model error and disturbances, the set-points can not be chosen at the real limits of its capabilities. As Figure 1.1 illustrates, the better the control system deals with these error sources, the operating point can be chosen closer to the constraints. The disturbances origin from temperature variations, liquid slugs, compositional changes from the wells, system shut downs due to maintenance activities and changes of substances in the tank. Since PI/PID are linear controllers, the tuning are based at linear models. Due to the valve characteristics, tank shapes or the chemical dynamics, the plant is nonlinear. These nonlinearities, approximations of transport delays or simplifications in the linear model are sources of model errors. Hence robust controllers need to be designed. The need for optimal disturbance dampening versus optimal set-point control is therefore clear, since the desired set-points, due to constrains, do seldom change[44]. The set-point responses are typically of secondary importance. This is in contrast to motion

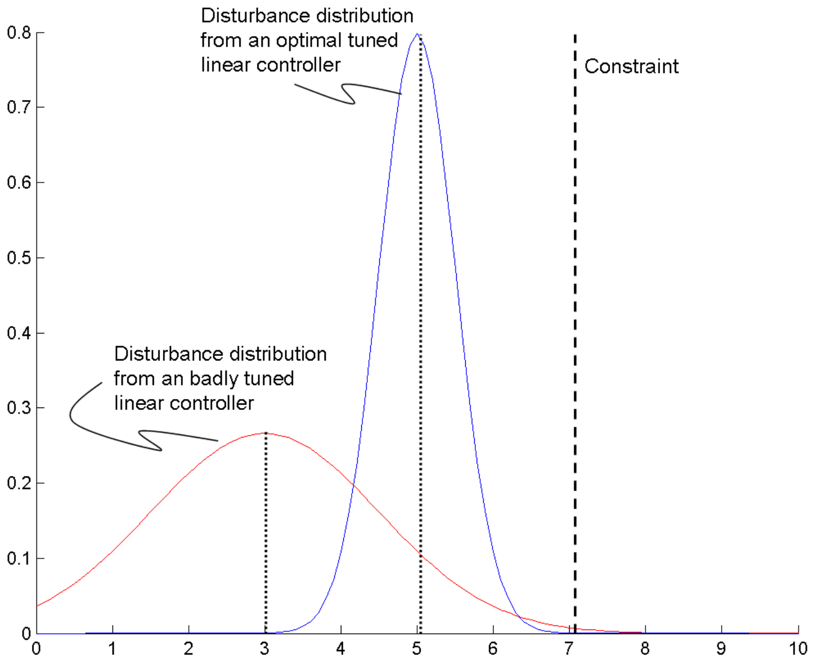

Figure 1.1: An illustration of why the set-point can be chosen closer to the real limits of the system capabilities, if the controller has optimal tuning settings. In this illustration the uncertainty is assumed to have a Gaussian distribution, which may not be the general case.

control systems such as ships, cranes or robot arms, where obviously the set-point is of most importance. Still most papers focus on the set-point control.

If the aim is to reach a desired set-point, a fast settling time is often desired. If this control imposes oscillations at the controller output, the actuator may tear needlessly. Therefore in some cases, it may be desirable with a more sluggish control. In some systems it is not required to reach a set-point at all. Buffer tanks are used to store a product between individual units in production process. Their purpose is to absorb disturbances in one unit, thus preventing the propagation of the effects to downstream units. The level of the tanks has to be controlled, so they do not become empty or full. If the controller is tuned too tight to a set-point in these systems, the buffer tanks would lose their purpose. Hence, as pointed out in [1], the buffer tanks need slow level control.

Due to the need for maintenance, shutdown of some part of the plant may be necessary. An undesired failure in the system circuits may also occur. Hence it is desired that the remaining system is stable while some loops are out of service or in manual. Stability ought to be guaranteed if some control outputs saturate. Control outputs saturate for instance when a higher flow rate than the maximum flow rate is desired. It is possible to guarantee such failure tolerance with decentralized control if the system holds some specifications called integrity.

When a controller is tuned, trade offs between performance and robustness must be made. These trade offs varies between the different control problems. It is seldom possible to achieve both properties.[39]

1.2.2 Interactions between the SISO-loops in multi-loop systems

Interactions between the different system inputs and outputs will usually occur in Multi Input Multi Output (MIMO) systems. If one of the inputs is changed, then it will affect some of, or all the outputs. If the objective is to keep a substance at a certain temperature and one of the the two flows into the tank contains a hot substance while the other contains a cold substance, they will both affect the temperature with opposite signs. Hence, multi-loop control will not directly compensate for these interactions. With a decentralized control configuration, a set-point change in one SISO loop results in disturbances in the other SISO-loops. Hence the performance in each loop will depend on all the controllers in the multi-loop system.

Interconnections are also associated with recirculation. The reflux flow back to the destination tank shown in Figure 1.2 is an example of such recirculation.

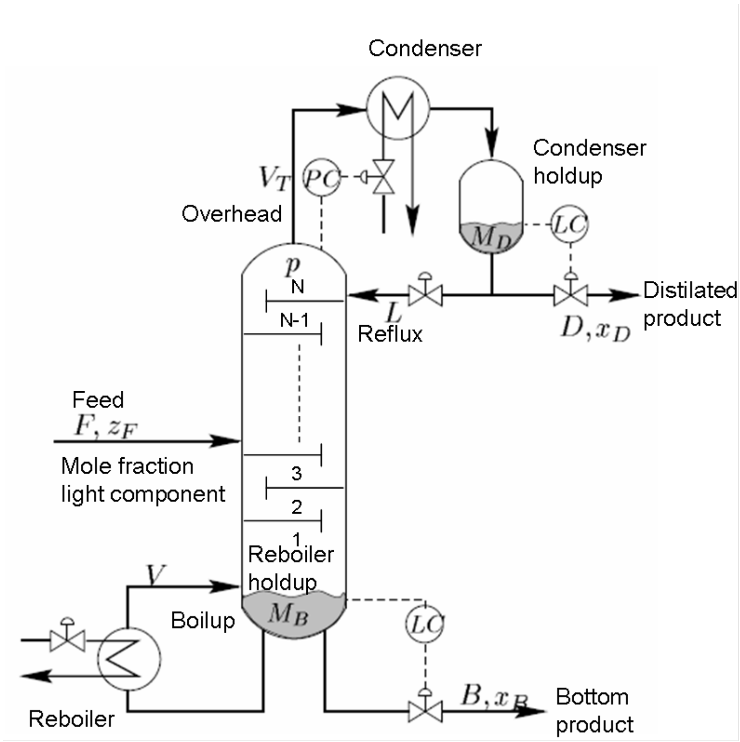

Figure 1.2: Distillation column with LV configuration

The binary distillation example in Figure 1.2 is detailed explained in [31, 38, 42, 36] and [39] at page 406. The objective is to control the top and bottom quality product. Trade offs must be considered when choosing inputs and outputs. Some interactions appear in the system no matter how the control configuration is chosen. The LV configuration, shown in Figure 1.2 is strongly interactive at steady state, but the quality control is nearly independent of the tuning of the level controls in the tanks. An alternative is the DV configuration, which is usually used for difficult separations. The steady state interactions are usually less compared to the LV configuration, but the tank-level and pressure control inflict strongly at the quality specification control.

1.3 Desired properties of the tuning approach

Since multi-loop PI/PID control remains popular in the process industry, a tuning approach for such controllers is obviously desirable. This tuning procedure should make the controller robust with respect to model errors and disturbances, and at the same time meet some performance requirements. The main concern is disturbance rejection for systems in the process industry. If some parts of the plant are shut down or some controllers saturate or fail, the rest of the system should be stable. A tuning procedure that results in such failure tolerance is desired.

The main challenge when obtaining the objectives above, are the system interactions. The tuning of each controller will affect the performance in some of, or all the other loops in a multi-loop system. The tuning procedure should handle these interactions, and preferably utilize the interaction dynamics.

Chapter 2

MIMO system properties

Some MIMO-system properties and definitions are presented in this chapter, so the reader can achieve a better understanding of the challenges concerning multi-loop PID tuning. In this thesis the notation $(\cdot)(s)$ refers to a function or a signal in the Laplace domain. At page 107 in [45] it is defined that $s = \sigma + j\omega$. In some cases only the frequency is considered. Hence the transfer function is denoted as $(\cdot)(j\omega)$. Since almost all the functions and signals in this thesis are in the Laplace domain, the notation $(\cdot)$ is the same as $(\cdot)(s)$.

2.1 MIMO system definitions

Controlling of MIMO systems are in general more difficult than for SISO systems. Some MIMO system properties and definitions according to control issues are here presented.

The transfer matrix from a vector of inputs $u(s)$ to a vector of measurements $y(s)$ is defined as $G(s)$, where the element $g_{ij}(s)$ is defined as the transfer function from input $u_i(s)$ to measurement $y_j(s)$. The transfer matrix $K(s)$ is defined as the controller. In this thesis, decentralized control in a $N \times N$ system will be the main concern, so

$$G(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) & \cdots & g_{1N}(s) \\ g_{21}(s) & g_{22}(s) & \cdots & g_{2N}(s) \\ \vdots & \vdots & \ddots & \vdots \\ g_{N1}(s) & g_{N2}(s) & \cdots & g_{NN}(s) \end{bmatrix} \tag{2.1}$$

$$K(s) = \begin{bmatrix} k_1(s) & 0 & \cdots & 0 \\ 0 & k_2(s) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & k_N(s) \end{bmatrix} \tag{2.2}$$

2.1.1 System loop transfer functions

A common way of representing the system in negative feedback interconnection is shown in Figure 2.1, where $r(s)$ is the reference value, and $e(s)$ is the reference-measure mismatch. According to [39] at page 22, the following relations hold

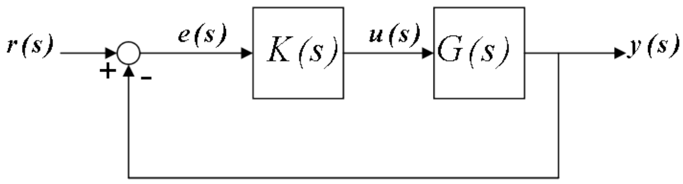

Figure 2.1: Simple block diagram of the system and the controller in a negative feedback interconnection.

$$L(s) \triangleq G(s)K(s) \tag{2.3}$$
$$S(s) \triangleq (I + G(s)K(s))^{-1} = (I + L(s))^{-1} \tag{2.4}$$
$$H(s) \triangleq (I + G(s)K(s))^{-1} G(s)K(s) = (I + L(s))^{-1} L(s) = I - S(s) \tag{2.5}$$

where $L(s)$ is called the loop transfer function, $S(s)$ is called the sensitivity function and $H(s)$ is called the complementary sensitivity function. These transfer functions relates to the different system inputs and outputs as follows

$$y(s) = L(s)e(s) \tag{2.6}$$
$$e(s) = S(s)r(s) \tag{2.7}$$
$$y(s) = H(s)r(s) \tag{2.8}$$

The notation $\text{diag}\{(\cdot)\}$ is a diagonal matrix with the diagonal elements of $(\cdot)$ at its diagonal. At page 437 in [39] some convenient definitions are given.

$$\Gamma(s) \triangleq \tilde{G}(s)G^{-1}(s) \tag{2.9}$$
$$\tilde{G}(s) \triangleq \text{diag}\{G(s)\} \tag{2.10}$$
$$E(s) \triangleq (G(s) - \tilde{G}(s))\tilde{G}(s)^{-1} \tag{2.11}$$
$$\tilde{S}(s) \triangleq (I + \tilde{G}(s)K(s))^{-1} \tag{2.12}$$
$$\tilde{H}(s) \triangleq \tilde{G}(s)K(s)(I + \tilde{G}(s)K(s))^{-1} = I - \tilde{H}(s) \tag{2.13}$$

where Γ is the Performance Relative Gain Array (PRGA), $\tilde{G}$ contains the diagonal elements of G and E is a interaction measure normalized with respect of the diagonal elements of G. The transfer functions $\tilde{S}$ and $\tilde{H}$ are the sensitivity function and the complementary sensitivity function respectively when only the diagonal elements of G

is concerned. Note $\tilde{S}$ is *not* the same as diag$\{S\}$. The sensitivity function S can be expressed by E, $\tilde{S}$ and $\tilde{H}$ as

$$S(s) = \tilde{S}(s)(I + E(s)\tilde{H}(s))^{-1} \tag{2.14}$$

2.1.2 PI and PID control

PI or PID controllers take several forms, as shown in [39] at page 56. The ideal form is expressed as

$$k_{PID\,ideal}(s) = k_c\left(1 + \frac{1}{\tau_I\, s} + \tau_D\, s\right) \tag{2.15}$$

The cascade form is

$$k_{PID\,cascade}(s) = \tilde{k}_c \frac{\tilde{\tau}_I\, s + 1}{\tilde{\tau}_I\, s}\,(\tilde{\tau}_D\, s + 1) \tag{2.16}$$

The cascade version does not allow complex zeros. If $\alpha = 1 + \frac{\tilde{\tau}_D}{\tilde{\tau}_I}$, the relationship between the ideal form and the cascade form is

$$k_c = \tilde{k}_c\,\alpha \tag{2.17}$$

$$\tau_I = \tilde{\tau}_I\,\alpha \tag{2.18}$$

$$\tau_D = \frac{\tilde{\tau}_D}{\alpha} \tag{2.19}$$

$$\tag{2.20}$$

The PID controller is improper. Hence a filter needs to be added to the controller itself or to the controller input. If the ideal form is considered, then

$$k_{PID,practical}(s) = k_c\left(1 + \frac{1}{\tau_I\, s} + \tau_D\, s\right)\frac{1}{\epsilon\tau_D\, s + 1} \tag{2.21}$$

where is typically $\epsilon \approx 0.1$. This filter may also avoid differentiation of high frequent noise. The closed loop response is not considerably changed with the filter added.

To avoid derivative kick, the reference is usually not differentiated. The practical controller is then

$$u = k_c\left[\left(1 + \frac{1}{\tau_I\, s}\right)(r - y_m) - \frac{\tau_D\, s}{\epsilon\tau_D\, s + 1}y_m\right] \tag{2.22}$$

Throughout this thesis, the control elements k_i in (2.2) are either PI or a PID controllers.

2.1.3 System scaling

It is desired that the system is properly scaled as shown in [39] at page 5. Let the unscaled system be

$$\hat{y} = \hat{G}\hat{u}, \quad \hat{e} = \hat{y} - \hat{r} \tag{2.23}$$

where Let D_e and D_u be scaling matrices with the maximum errors $e_{\max i}$ and the maximum controller outputs $u_{\max i}$ at the diagonal respectively. The scaled system then becomes

$$y = D_e^{-1} \hat{G} D_u u \tag{2.24}$$

Now u should be less than 1 in magnitude.

2.1.4 Singular value decomposition and H_∞- norm

In [39] at page 75 it is derived that any matrix $G(j\omega)$ with a fixed frequency ω can be decomposed as

$$G(j\omega) = U(j\omega)\Sigma(j\omega)V^H(j\omega) \tag{2.25}$$

where Σ is an $l \times m$ matrix with $k = \min\{l, m\}$ non-negative singular values σ_i, arranged in descending order along its main diagonal. The other elements in Σ is zero. The $l \times l$ matrix U is unitary and consists of output singular vectors u_i^s, not to be confused with the input system vector u. The $m \times m$ matrix V is unitary, and consists of the input singular vectors v_i. The vectors u_i^s and v_i represent the output and input direction respectively. Since $V^H V = I$, (2.25) can be rewritten as $GV = U\Sigma$. Since $\|u_i^s\|_2 = \|v_i\|_2 = 1$, σ_i directly gives the gain of the system $G(j\omega)$ in the i'th direction. The most interesting is the systems maximum and minimum singular values $\bar{\sigma} = \sigma_1$ and $\underline{\sigma} = \sigma_k$, and the corresponding input and output directions $\bar{v} = v_1$, $\underline{v} = v_k$, $\bar{u}^s = u_1^s$ and $\underline{u}^s = u_k^s$.

$$\bar{\sigma}(G) = \max_{u \neq 0} \frac{\|G\, u^s\|_2}{\|u^s\|_2} = \frac{\|G\, v_1\|_2}{\|v_1\|_2} \tag{2.26}$$

$$\underline{\sigma}(G) = \min_{u \neq 0} \frac{\|G\, u^s\|_2}{\|u^s\|_2} = \frac{\|G\, v_k\|_2}{\|v_k\|_2} \tag{2.27}$$

Hence, $\bar{v}$ corresponds to the direction with the largest amplification of the output with the direction $\bar{u}^s$.

The H_∞- norm defined at page 60 in [39], relates to the maxim singular value to a system matrix G as follows.

$$\|G(s)\|_\infty \triangleq \max_\omega \bar{\sigma}\left(G(j\omega)\right) \tag{2.28}$$

2.1.5 Structured singular value

The structured singular value was first defined in [18] as

Definition 1. *Let M be a given complex matrix, and let $\Delta = \operatorname{diag}\{\Delta_i\}$ denote a set of complex matrices with $\bar{\sigma}(\Delta) \leq 1$ and with a given block-diagonal structure (in which* *some of the blocks may be repeated and some may be restricted to be real.) The real non-negative function $\mu(M)$, called the structured singular value is defined by*

$$\mu(M) = \frac{1}{\min\{k_m \,|\det(I - k_m M\Delta) = 0 \text{ for structured } \Delta, \bar{\sigma}(\Delta) \leq 1\}} \tag{2.29}$$

If no such structured Δ exists then $\mu(M) = 0$.

2.1.6 Padé approximation

A plant model $G(s)$ often includes a transport delay denoted $e^{\theta\, s}$ in the Laplace domain. This makes the transfer function irrational. Hence it can not be used in calculations of linear time-invariant (LTI) systems in simulation programs such as MATLAB. An approximation of this expression is found by an n order Taylor series called Padé approximation, defined in [39] at page 127 as

$$e^{\theta\, s} = \frac{\left(1 - \frac{\theta}{2n}\, s\right)^n}{\left(1 + \frac{\theta}{2n}\, s\right)^n} \tag{2.30}$$

A higher order gives a more accurate approximation.

2.2 Interactions

As explained in Section 1.2.2 some interactions will usually occur in MIMO systems. In [39] at page 89 it is stated that the interactions only are one way for an upper triangular (only zeros at its lower off-diagonal elements) or lower triangular (only zeros at its upper off-diagonal elements) plant $G(s)$. For plants with elements at both sides of its diagonal, the interconnection is two ways. A plant is non-interacting if a change in u_i only results in a change in y_i. The following theorem is stated and proved in [39] at page 441.

Theorem 1. *Suppose the plant $G(s)$ is stable and upper or lower triangular (at all frequencies), and is controlled by a diagonal controller. Then the overall system is stable if and only if the individual loops are stable.*

Hence, one-way interactions can not introduce instability. The same cannot be concluded for two-way interactions.

In [39] at page 440 it is stated that if a matrix is diagonal dominant, then the interactions will not introduce instability. A definition of generalized diagonal dominance is defined as

Definition 2. *A matrix G is said to be generalized diagonal dominant if and only if $\mu(E) < 1$*

If some combinations of the inputs have large effect on the outputs, whereas other combinations have weak effect, the plant is said to be ill-conditioned.

2.2.1 Interaction measures

The singular values can be used determine if the plant is ill-conditioned. Define the condition number $\gamma(G)$ as

$$\gamma(G) = \frac{\bar{\sigma}(G)}{\underline{\sigma}(G)} \tag{2.31}$$

A large condition number implies an ill-conditioned plant. The scaling of $G(s)$ will affect the condition number, so the system needs to bee proper scaled.

The Rellative Gain Array (RGA) was first defined in [10]. In [35, 30, 39] it is outlined that

$$\text{RGA}(G) = \Lambda(G) = G \times (G^{-1})^T \tag{2.32}$$

where $\times$ denotes element-by-element multiplication. Based at its original definition

$$\text{RGA} = \frac{\left.\frac{\partial y_i}{\partial u_j}\right|_{u_k=0, k\neq j}}{\left.\frac{\partial y_i}{\partial u_j}\right|_{y_k=0, k\neq j}} = \frac{g_{ij}}{\hat{g_{ij}}} \tag{2.33}$$

This is the transfer matrix gain when all other loops are open divided at the transfer matrix gain when all the other loops are closed with perfect control. If the elements diverge largely from one, it implies that $\hat{g}_{ij}$ is affected by the interactions. The column and row sum of the RGA matrix is always one. An interesting property is that $RGA = I$ if $G(s)$ is upper or lower triangular. Hence RGA is a measure for two-way interaction.

According to [39] at page 437, the interactions are given by the off-diagonal elements in $G(s)$ when decentralized control is used. Hence, the magnitude of $E(s)$, defined in (2.11), is a common used interaction measure. It does not distinct the difference between one-way and two-way interactions.

2.2.2 Spectral radius

In [39] at page 518 it is remarked that the eigenvalues of a matrix A are sometimes called characteristic gains. The set of eigenvalues of A is called the spectrum of A. The largest absolute values of the eigenvalues of A is the spectral radius of A

$$\rho(A) \triangleq \max_i |\lambda_i(A)| \tag{2.34}$$

2.3 Failure tolerance, integrity and detuneability

In Chapter 1 it is explained why failure tolerance is desired for control systems. To formalize such requirements, some definitions concerning failure tolerance are here presented.

In this thesis the system has *failure tolerance* if the closed loop system remains stable when *some* loops, but not all, are brought out of service or saturate while the remaining loops still are intact. Hence, open unstable systems may have failure if the stabilizing loops are intact.

A decentralized control system is said to have integrity if the closed loop system remains stable when an arbitrary subsystem of controllers are brought in and out of service or when inputs saturate. In [39] at page 442 integrity is explained mathematically.

Definition 3. *The system possess integrity if it remains stable when the controller* $K(s)$ *is replaced by the matrix* $E\,K(s)$*, where* $K(s)$ *is the controller and* $E = \mathrm{diag}\{\epsilon_i\}$*. The diagonal elements* ϵ_i *may take on the values of* $\epsilon_i = 1$ *or* $\epsilon_i = 0$*.*

This definition requiters that the system must be open stable to possess integrity, since all the diagonal elements of E can be $\epsilon_i = 0$.

If integrity with integral control is possible, it is addressed in [37] that either of the following conditions must hold:

1. $G(0)D$ is positive definite, where

$$D = \mathrm{diag}\left\{\frac{g_{11}}{|g_{11}|}, \frac{g_{22}}{|g_{22}|}\right\}$$

2. There exists a diagonal matrix X such that $G(0)X$ is positive definite

3. Spectra of all principal sub-matrices of $G(0)$ exist and are positive

If the decentralized control system remains stable as the gains in the various loops are detuned (reduced) by an arbitrary factor $\epsilon_i \in [0, 1]$, the system is complete detunable. If complete detunability is possible with integral control, the system is said to have the decentralized integral controllability property. This is referred to as DIC, and is defined in [3] and in [39] at page 442 as

Definition 4. *The plant* $G(s)$ *is DIC if there exists a stabilizing decentralized controller with integral action in each loop such that each individual loop may be detuned independently by a factor* $\epsilon_i \in [0, 1]$ *without introducing instability.*

Since most controllers in decentralized control are PI or PID, this property is desired, but not always achieved. A stable system is DIC only if the diagonal steady state RGA elements $\lambda_{ii}(0) \geq 0$ for all i.

Furthermore it is desired that the system is stable if the model is not accurate. Such systems are said to be robust stable. In [29] robust decentralized detunable systems is defined as

Definition 5. *A closed loop system is said to be robust decentralized detunable (RDD) if each controller element can be detuned independently by an arbitrary amount without endangering robust stability.*

RDD is not the same as DIC. The system property DIC implies the existence of a decentralized controller with integral action that is decentralized detunable.

2.4 Model uncertainty

Chapter 7 and Chapter 8 in [39] consider the model uncertainties. They are grouped into two main classes.

- **Parametric uncertainty.** The model structure is known, but some of the parameters are uncertain. If α is a parameter in the model, it is bounded within a region $[\alpha_{\min}\ \alpha_{\max}]$. A common way to represent the parameter is of the form

$$\alpha = \bar{\alpha}(1 + r_\alpha \Delta) \tag{2.35}$$

  where $\bar{\alpha}$ is the mean parameter, $r_\alpha = \frac{\alpha_{\min}-\alpha_{\max}}{\alpha_{\min}+\alpha_{\max}}$ is the relative uncertainty in the parameter, and Δ is any real scalar satisfying $|\Delta| \leq 1$.

- **Dynamic uncertainty.** The lack of dynamics are present, usually at high frequencies, either trough deliberate neglect or because the physical process is not understood well enough. Representation of this type of uncertainty is best done in the frequency domain. Hence the perturbation is a complex transfer function such that $\|\Delta\|_\infty \leq 1$.

It is common to lump the different sources of uncertainty into an additive or multiplicative uncertainty model. To get a simple uncertainty model for MIMO systems, unstructured perturbations are often used. Then the Δ is a "full" complex matrix with dimensions compatible with those of the plant, where any Δ satisfying $\|\Delta\|_\infty \leq 1$ is allowed. If G_p is the plant including uncertainty, examples of representations are

$$G_p = G + w_A\Delta_A \quad \|\Delta_A\|_\infty \leq 1 \text{ - Additive uncertainty} \tag{2.36}$$
$$G_p = G(I + w_I\Delta_I) \quad \|\Delta_I\|_\infty \leq 1 \text{ - Multiplicative input uncertainty} \tag{2.37}$$
$$G_p = (I + w_O\Delta_O)G \quad \|\Delta_O\|_\infty \leq 1 \text{ - Multiplicative output uncertainty} \tag{2.38}$$
$$G_p = (I + w_{iO}\Delta_{iO})^{-1}G \quad \|\Delta_{iO}\|_\infty \leq 1 \text{ - Inverse multiplicative output uncertainty} \tag{2.39}$$

where w_A, w_I, w_O and w_{iO} are some SISO transfer functions. If Δ is not frequency dependent, the uncertainty is parametric. The form of w_A, w_I, w_O and w_{iO} will determine which parameters the uncertainty includes. Using multiplicative uncertainty, the weighting functions w_I, w_O and w_{iO} can easily be used to derive bounds of which frequency region the nominal model G can be trusted. If $|w_I| \geq 1$, then the uncertainty exceeds 100%.

The uncertainty representations (2.36) to (2.39) can also be used in SISO cases. If different SISO perturbations δ_i are lumped into a perturbation matrix Δ, it will result in some structure in the Δ matrix. Hence just perturbation blocks with diagonal structure so that $\|\Delta\|_\infty \leq 1$ is satisfied should be considered. Perturbations that arise form uncertainty or neglected dynamics in the individual actuators or in the individual sensors, are always present. This leads to a complex diagonal perturbation matrix $\Delta(s) = \text{diag}\{\delta_{ii}(s)\}$ where $|\delta_{ii}(j\omega)| \leq 1\ \forall\omega,\ i$.

2.5 System performance

An obvious way of describing the performance specification of a system is in the time domain using for instance *rise time*, *settling time* or *overshoot*. But for control design purposes it may be convenient to represent the performance specifications in the frequency domain. Responses form sinusoids of any frequencies are here considered, and not only the step response. Some of the frequency domain measures are gain and phase margins, defined at page 32 in [39] or at page 289 in [45], or maximum peaks of S and H defined at page 35 in [39] as M_S and M_H respectively.

$$\|S\|_\infty = M_S \quad \|H\|_\infty = M_H \tag{2.40}$$

A large value of M_S and M_H (typically larger 4) indicates poor robustness and performance.

Since $e = y - r = S\,r$ it is desired to have S as small as possible. But since real systems are strictly proper, $L \to 0 \Leftrightarrow S \to 1$ as $\omega \to \infty$. At intermediate frequencies one cannot avoid a peak value of M_S larger than 1. Thus there is an intermediate frequency range where performance is degrading, and M_S is the worst case measure of this degradation. Since the smallest distance between $L(j\omega)$ and -1 is M_S^{-1} in the Nyquist curve, a smaller peak value of $S(j\omega)$ gives better robustness.

An relationship between the time and frequency domain peaks is pointed out at page 37 in [39]

2.5.1 Bandwidth

The bandwidth is a measure of what frequency range the control is effective. In most cases it is required tight control at steady state, so the bandwidth only concerns the upper frequency limit. In [39] tree definitions of the SISO system bandwidth are outlined in terms of $L(j\omega)$, $S(j\omega)$ and $H(j\omega)$

Definition 6. ***Bandwidth***

1. *The closed-loop bandwidth* ω_c *where* $|L(j\omega)|$ *first crosses* 1 *from above. This frequency is also defined as the gain crossover frequency.*

2. *The closed-loop bandwidth* ω_B *is the frequency where* $|S(j\omega)|$ *first crosses* $\frac{1}{\sqrt{2}} \approx 3$ *dB from below.*

3. *The closed-loop bandwidth* ω_{BH} *is the highest frequency at which* $|H(j\omega)|$ *crosses* $\frac{1}{\sqrt{2}} \approx 3$ *form above.*

where the relation between them is

$$\omega_B < \omega_c < \omega_{BH} \tag{2.41}$$

The crossover ω_c is used as a bandwidth definition due to its simplicity. Generally frequencies up to ω_B, the control will improve the performance. In the range $[\omega_B,\ \omega_{BH}]$ the control will affect the response, but not improve the performance. When $|S(j\omega)| > 1$, the control degrades performance.

For the MIMO case, the bandwidth is direction dependent. Let $\omega_{B\,wc}$ denote the "worst-case" direction, where $\bar{\sigma}(S(j\omega))$ crosses $\frac{1}{\sqrt{2}} \approx 3$ from below, and $\omega_{B\,bc}$ denote the "best-case" direction where $\underline{\sigma}(Sj\omega)$ crosses $\frac{1}{\sqrt{2}} \approx 3$ from below. The bandwidth region is then $[\omega_{B\,wc}\ \omega_{B\,bc}]$. If the bandwidth must be associated with a single frequency, then $\omega_{B\,wc}$ is used.

2.5.2 Performance weights

The sensitivity function S is often used as a performance indicator. Specifications of S, so the performance requirements are achieved, may be

1. Minimum bandwidth frequency
2. Maximum tracking error at selected frequencies
3. System type or alternatively the maximum steady state tracking error
4. Shape of S over selected frequency ranges
5. Maximum peak magnitude of S

To achieve the items addressed above, it is convenient to design a performance weighting function $w_P(s)$, and let

$$|S(j\omega)| < \left|\frac{1}{w(j\omega)}\right|,\ \forall\omega \quad \Leftrightarrow \quad \|w_P\,S\|_\infty < 1 \tag{2.42}$$

When (2.42) is satisfied, nominal performance of the system is achieved.

2.6 Stability and performance analysis

Any system can be represented as a general plant P and a controller matrix K with a perturbation block Δ, shown in Figure 2.2. The P matrix includes the uncertainty and performance weights described in Section 2.4 and Section 2.5.2. How to derive the generalized plant P depends at the system.

In the terms of Figure 2.2, w is the system input vector, z is the system output vector, v is the controller input, u is the controller output, y_Δ is the input to the perturbation block and u_Δ is the output of the perturbation block. With lower Linear Fractional Transformation (LFT), the PK representation can be lumped into a N representation as

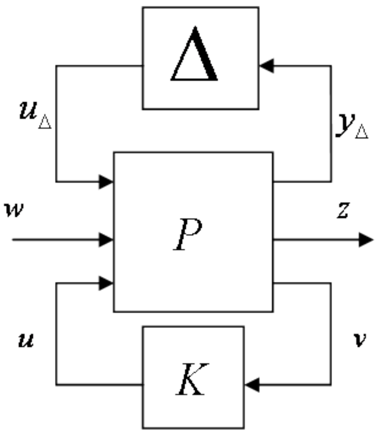

Figure 2.2: PK representation of the system with perturbation block Δ.

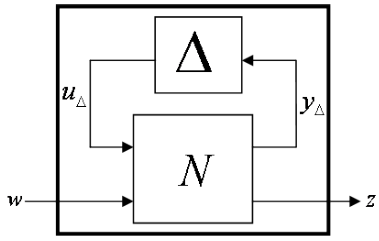

Figure 2.3: N representation of the system with perturbation block Δ.

$$N = F_l(P, K) = P_{11} + P_{12}K\left(I - P_{22}K\right)^{-1}P_{21} \tag{2.43}$$

The system is represented as in Figure 2.3

To include the perturbation into the system, an upper LFT is used.

$$F = F_u(N, \Delta) = N_{22} + N_{21}\Delta\left(I - N_{11}\Delta\right)^{-1}N_{12} \tag{2.44}$$

The instability caused by the system perturbations, is represented by the term $(I - N_{11}\Delta)^{-1}$. Hence the element N_{11} is of especially interest. In the literature this element is referred as M, and the $M\Delta$ structure in Figure 2.4 is considered in robust stability analysis.

$$M = N_{11} \tag{2.45}$$

Now Nominal Stability (NS), Nominal Performance, Robust Stability (RS) and Robust Performance (RP) can be defined in terms of the system representations above.

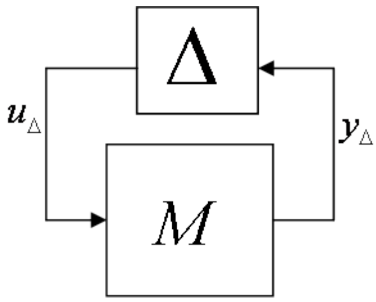

Figure 2.4: M representation of the system with perturbation block Δ.

$$\text{NS} \Leftrightarrow N \text{ is internally stable} \quad (2.46)$$

$$\text{NP} \Leftrightarrow \|N_{22}\|_\infty < 1, \quad \text{and NS} \quad (2.47)$$

$$\text{RS} \Leftrightarrow F = F_u(N,\Delta) \text{ is internally stable } \forall\Delta,\ \|\Delta\|_\infty \leq 1 \quad \text{and NS} \quad (2.48)$$

$$\text{RP} \Leftrightarrow \|F\|_\infty < 1, \quad \text{is internally stable } \forall\Delta,\ \|\Delta\|_\infty \leq 1 \quad \text{and NS} \quad (2.49)$$

2.7 Overall stability criteria for decentralized control

First the generalized Nyquist theorem is stated. This theorem applies for all MIMO systems, and is defined in [39] at page 152.

Theorem 2. Generalized (MIMO) Nyquist theorem *Let P_{ol} denote the number of open-loop unstable poles in $L(s)$. The closed-loop system with open transfer function $L(s)$ and negative feedback is stable if and only if the Nyquist plot of* $\det(I - L(s))$

1. *makes P_{ol} anti-clockwise encirclements of the orign, and*
2. *does not pass through the orign.*

Some interaction criteria that need to be satisfied to guarantee nominal stability of the overall system with decentralized control are stated in this section. Note that these criteria can be used in all the design methods, but are especially important in the independent design.

Theorem 3. *With assumptions that the plant G is stable and each individual loop is stable by itself ($\tilde{S}$ and $\tilde{H}$ are stable), the overall system is stable (S is stable):*

1. *if and only if $\left(I + E\tilde{H}\right)^{-1}$ is stable, where E is defined as in* (2.11)
2. *if and only if* $\det\left(I + E\tilde{H}\right)$ *does not encircle the origin as s traverses the Nyquist D-contour.*

3. if

$$\rho(E\tilde{H}(j\omega)) < 1,\ \forall\omega \tag{2.50}$$

4. (and (2.50) *is satisfied) if*

$$\bar{\sigma}(\tilde{H}) = \max_i \left|\tilde{h}_i\right| < \mu^{-1}(E)\ \forall\omega \tag{2.51}$$

The structured singular value $\mu(E)$ is computed with respect to a diagonal structure of $\tilde{H}$

A similar theorem is stated in [39] at page 438, only that $\tilde{H}$ is swapped with $\tilde{S}$ and E is swapped with $E_S = (G - \tilde{G})G^{-1}$.

The Structure Singular Value Interaction Measure in (2.51) was first introduced in [35]. There it is stated that this is the tightest norm bound. The definition of μ is stated in (2.29). Failure tolerance is usually obtained. In [8] it is pointed out that (2.51) is conservative because the phase information of $\tilde{H}$ is not utilized.

To derive less conservative bounds than the overall stability condition in (2.51), $\tilde{H}$ is split into a fixed part $\tilde{H}_B$ and a design part $\tilde{H}_M$.

$$\tilde{H} = \tilde{H}_B\left(I + \tilde{H}_M\right) \tag{2.52}$$

Assume that $\tilde{H}_B = \frac{1}{2}\tilde{H}$ and $\hat{H} = \frac{1}{2}\tilde{H}\left(I + \tilde{H}_M\right)$, then the stability conditions in Theorem 4 are presented in [26]

Theorem 4. *Assume that $G(s)$ and $\tilde{G}(s)$ have the same number of Right Half Plane (RHP) poles and that $\tilde{H}$ and $\hat{H}$ are stable. Then the closed-loop system $H(s,\hat{H})$ is stable if*

$$N_n\left\{0, \det\left(I + \frac{1}{2}E\tilde{H}\right)\right\} = 0 \tag{2.53}$$

where N_n is the number of encirclements of the Nyquist curve, and

$$\left|\frac{\hat{h}_i(j\omega)}{\tilde{h}_i(j\omega)} - \frac{1}{2}\right| < \mu^{-1}\left(\left[I + \frac{1}{2}E(j\omega)\tilde{H}(j\omega)\right]^{-1} E(j\omega)\tilde{H}(j\omega)\right) \quad \forall\omega \in [0,\infty\rangle\,,\ i = 1,2,..N \tag{2.54}$$

The condition (2.51) for nominal stability of decentralized control, can be generalized to include robust stability and robust performance. The complimentary sensitivity function H doses not include the uncertainty and performance weights. The transfer function matrix N in Theorem 5, stated in [40], is used to include these uncertainties into the interconnection matrix M shown in Figure 2.5. Note that the N in Theorem 5 is independent of the controller, and is a different matrix than the N in (2.43).

Theorem 5. *Let the μ interconnection matrix M be written as lower linear fractional transformation (LFT) of the transfer matrix H*

$$M = F_l(N,H) = N_{11} + N_{12}H(I - N_{22}H)^{-1}N_{21} \tag{2.55}$$

assume $\mu_\Delta(N_{11}) < 1$ *and* $\det(I - N_{22}H) \neq 0$, *then*

$$\mu_\Delta(M) \leq 1 \quad \textit{if} \quad \bar{\sigma}(H) \leq c_H \tag{2.56}$$

where c_H *solves*

$$\mu_{\hat{\Delta}} \begin{bmatrix} N_{11} & N_{12} \\ c_H N_{21} & c_H N_{22} \end{bmatrix} = 1 \tag{2.57}$$

and $\hat{\Delta} = \mathrm{diag}(\Delta, H)$

Using Theorem 5 and $H = G\tilde{G}^{-1}\tilde{H}\left(I + E\tilde{H}\right)^{-1}$ with the assumption that $N_{22} = 0$ a robust performance condition in terms of $\tilde{H}$ is outlined in [41] as

RP-condition in terms of $\tilde{H}$ Let $M = N_{11} + N_{12}G\tilde{G}^{-1}\tilde{H}\left(I + E\tilde{H}\right)^{-1} N_{21}$. Then at any frequency

$$\mu_\Delta(M) \leq 1 \quad \text{if} \quad \bar{\sigma}(\tilde{H}) \leq c_{\tilde{H}} \tag{2.58}$$

where $c_{\tilde{H}}$ solves

$$\mu_{\hat{\Delta}} \begin{bmatrix} N_{11} & N_{12}G\tilde{G}^{-1} \\ c_{\tilde{H}} N_{21} & -c_{\tilde{H}} N_{22} \end{bmatrix} = 1 \tag{2.59}$$

and $\hat{\Delta} = \mathrm{diag}(\Delta, K)$ The structure of K is block-diagonal and equal to that of $\tilde{H}$.

An equivalent robust performance condition to (2.58) using $\tilde{S}$ and $E_S = (G - \tilde{G})G^{-1}$ is stated in [41].

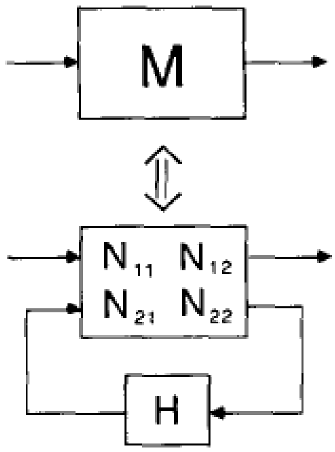

Figure 2.5: The interconnection matrix M expressed as an LFT of H. N is independent of the controller K

Alternatively to Theorem 3 is to use Greshgorgin bounds defined in [13, 21] and in [39] at page 439. Consider the Nyquist curve of $g_{ii}k_i$, and superimpose the circle of radius $\sum_{i=1,k\neq i}^{n} |g_{ki}(j\omega)k_i(j\omega)|$. This circle is the Greshgorin circle. The band composed by Greshgorin circles for all frequencies is called the Greshgorin band. Theorem 6 is stated in [8], where derivations in [35] are used.

Theorem 6. *Assume that $G(s)$ and $\tilde{G}(s)$ have the same number of RHP poles and $\tilde{H}(s)$ is stable. Then the closed loop system $H(s)$ is stable if*

$$\left|\tilde{h}_i(j\omega)\right| < \frac{|g_{ii}(j\omega)|}{\sum_{k=1,k\neq i}^{n} |g_{ki}(j\omega)|} \quad \forall i, \omega \tag{2.60}$$

or equivalent

$$|g_{ii}(j\omega)k_i(j\omega)| > \sum_{k=1,k\neq i}^{n} |g_{ki}(j\omega)k_i(j\omega)| \quad \forall i, \omega \tag{2.61}$$

This ensures diagonal column dominance.

An equivalent theorem can be stated for diagonal row dominance. The smallest value of the upper bounds in (2.60) is always more restrictive than (2.51). However (2.51) imposes the same bound on $|\tilde{h}(j\omega)|$ for each loop, whereas (2.60) give individual bounds. Hence some of the Greshgorin bounds may be less restrictive than the Structure Singular Value Interaction Measure[39].

Chapter 3

A survey of SISO and MIMO tuning methods

3.1 SISO tuning methods

Since the main concern of this thesis is multi-loop PID tuning, only some of the most important SISO tuning methods are presented. This is done because some of the multi-loop methods are modifications of the SISO tuning rules.

3.1.1 Ziegler-Nichols tuning method

To use this method for tuning the parameters in (2.15), the open loop system must be stable, and the magnitude- and phase characteristic have to fall monotonous when the frequency is increased. In [45] at page 334, the tuning method is presented in depth.

First $\tau_I \approx \infty$ and $\tau = 0$. The controller gain is increased until the output has undamped oscillations. This gain K_u, called the ultimate controller gain, and the period of the oscillations P_u are used further in the tuning process. K_u and P_u can be found analytically by

$$K_u = \frac{1}{|G(j\omega_u)|}, \quad P_u = \frac{2\pi}{\omega_u} \tag{3.1}$$

where ω_u is defined by $\angle G(j\omega_u) = -180°$. The tuning parameters are different for P, PI and PID controllers.

P controller

$$K_c = \frac{1}{2} K_u \tag{3.2}$$

PI controller

$$K_c = 0.45\, K_u, \quad \tau_I = 0.85\, P_u \tag{3.3}$$

PID controller

$$K_c = 0.6\, K_u \quad \tau_I = 0.5\, P_u \quad \tau_D = 0.12\, P_u \tag{3.4}$$

3.1.2 Tyreus-Luyben tuning

A method as simple as the Ziegler Nichols approach is the Tyreus and Luyben (TL) method, presented in [47]. Consider the ultimate gain K_u and frequency P_u in (3.1). Then the TL settings for a PI controller can be calculated as

$$K_{TL} = \frac{K_u}{3.2}, \quad \tau_{I_{TL}} = 2.2\, P_u \tag{3.5}$$

This is a more conservative approach than Ziegler Nichols. Hence it gives better performance with small values of deadtime, but very sluggish performance when dead times are large.

3.1.3 IMC and SIMC

IMC

The Internal Model Control (IMC) strategy is detailed explained in [39] at page 54. The plant G is factorized as

$$G(s) = G_m G_a \tag{3.6}$$

$$G_a(s) = e^{-\theta s} \prod_i \frac{-s + z_i}{s + z_i}, \quad \mathbb{R}(z_i) > 0,\ \theta > 0 \tag{3.7}$$

where G_m is the invertible minimum phase part, and G_a is the non-invertible all-pass part. (The time delay and Right Half Plane (RHP) zeros cannot be inverted.) Specify the desired complimentary sensitivity function H. Choose a filter $f(s)$ such that

$$H(s) = f(s)G_a(s) = G(s)K(s)\left(1 + G(s)K(s)\right)^{-1} \tag{3.8}$$

$$f(s) = \frac{1}{(\tau_c\, s + 1)^n} \tag{3.9}$$

$$\Rightarrow K = G^{-1}\frac{H}{1 - H} = G_m^{-1}\frac{1}{f^{-1} - G_a} \tag{3.10}$$

The controller inverts the minimum phase part of G. The time delay $e^{-\theta s}$ is realized by Padé approximation defined in (2.30). A small value on τ_c makes the loop fast, and a big value makes it robust due to problems at higher frequencies.

SIMC

It is common for process control purposes to approximate the plant as a first order model with time delay. Then the plant G and the desired closed loop function H is defined as

$$G(s) = \frac{k}{\tau\, s + 1}\, e^{-\theta\, s} \tag{3.11}$$
$$H(s) = \frac{1}{\tau_c\, s + 1}\, e^{-\theta\, s} \tag{3.12}$$

With (3.11) and (3.12) as basis, a PI controller can be designed with the IMC strategy. In [39] at page 57 it is purposed some tuning rules for this PI-IMC controller, called Skogestad IMC or Simple IMC (SIMC) settings

$$\tilde{K}_c = \frac{1}{k}\frac{\tau}{\tau_c + \theta} \tag{3.13}$$
$$\tilde{\tau}_I = \min(\tau, 4(\tau_c + \theta)) \tag{3.14}$$

The closed loop system will oscillate with a $\tilde{\tau}_I$ too small, but it has to be small enough to contradict ramp like effects on the output from disturbances. The system will be sensitive to such disturbances for large τ, typically $\tau \geq 8\theta$.

If G is a dominant second order process ($\tau_2 > \theta$), a PID-IMC structure is obtained, with the following recommended SIMC settings.

$$G(s) = \frac{k}{(\tau_1\, s + 1)(\tau_2\, s + 1)}\, e^{-\theta\, s} \tag{3.15}$$
$$\tilde{K}_c = \frac{1}{k}\frac{\tau_1}{\tau_c + \theta} \tag{3.16}$$
$$\tilde{\tau}_I = \min(\tau_1, 4(\tau_c + \theta)) \tag{3.17}$$
$$\tilde{\tau}_D = \tau_2 \tag{3.18}$$

With these rules the only tuning parameter is τ_c. Small τ_c gives good output performance, and large τ_c gives robustness and input usage. For robust and fast control, it is recommended that $\tau_c = \theta$.

3.1.4 Iterative continuous cycling tuning

In [26] the Iterative continuous cycling (ICC) tuning for SISO systems is explained. This method does not depend on a process model.

1. Set $\tau_I \approx \infty$ and $\tau_D = 0$. Then increase the gain until continuous cycles occurs at the output. The ultimate gain and frequency in (3.1) is then reached. Sett $k_c = 0.5\, k_u$.

2. Decrease the integral time τ_I until continuous cycles occur. Set the integral time three times this value.

3. Increase the derivate time until continuous cycles occur. Set the derivative time at one-third of this value.

Since the controller must initially be turned off to make use of the method, it can only be applied at open stable systems.

3.1.5 Relay-feedback approach

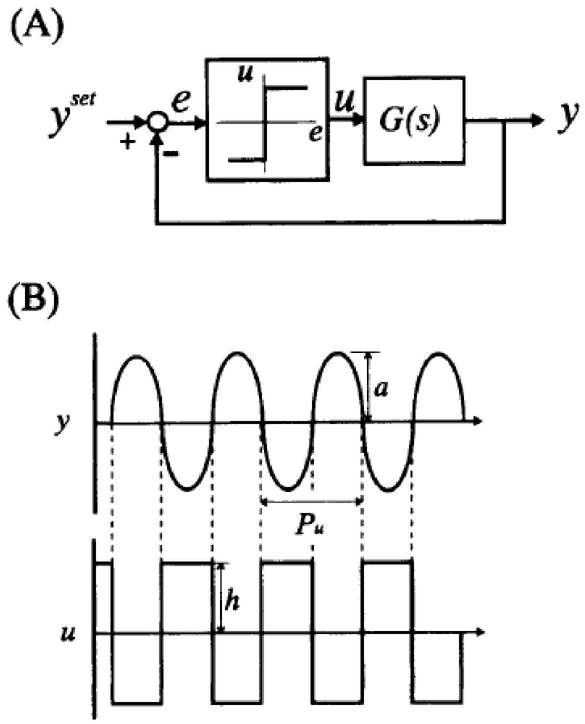

Figure 3.1: (A) Block diagram for a relay feedback system and (B) relay feedback test for a system with positive steady state gain

In Ziegler-Nichols or Tyereus and Luyben, the ultimate gain and frequency are used to determine the PI/PID settings for SISO systems. To estimate these ultimate parameters the relay feedback approach, which outlined in [6] and [12], are used. The estimation works as follows: When the output lags behind the input by $-\pi$ radians, the closed loop system may oscillate with a period P_u. Figure 3.1 shows how the block structure and the output $y(t)$ and input $u(t)$ relates to each other. Initially raze $u(t)$ with amplitude h. When $y(t)$ starts rising, change the sign of $u(t)$. Change the sign of $u(t)$ each time y is crossing its bias.

The output will oscillate with an amplitude a and a period P_u. Then the ultimate gain K_u and ultimate frequency ω_u can be determined by

$$k_u = \frac{4h}{\pi a} \tag{3.19}$$

$$\omega_u = \frac{2\pi}{p_u} \tag{3.20}$$

Hence a mathematical model of $G(s)$ is not necessary. But estimates of the transfer functions can be obtained form K_u and ω_u if desired, outlined in Chapter 3 in [6] and [12]. A structure of $G(s)$ has then to be assumed. Then a model based tuning method based at the estimated model $G(s)$ can be done, outlined in [12].

The information obtained by this method is the same as for the continuous cycling method, but the oscillations can be obtained in a controlled manner.

3.2 MIMO tuning for decentralized PI and PID controllers

In this section the control matrix has N SISO PI or PID controllers at the diagonal. The controllers have the same structure as (2.15).

First it is desired that the system is paired properly to active good performance and avoid instability caused by interactions, using the pairing rules in [39] at page 449:

1. **Prefer pairings such that the rearranged system has a RGA matrix close to identity at frequencies around the closed loop bandwidth.**
   This is to ensure diagonal dominance. To ensure stability, the rearranged plant must be triangular at crossover frequencies. This is not guaranteed by this rule. Since all triangular plants have RGA $= I$ and there is at most one choice of pairings with RGA $= I$, diagonal dominance can be checked by the condition $\mu(E_S) = \mu(PRGA - I) < 1$ at crossover frequencies.

2. **For a stable plant, avoid pairings that correspond to negative steady state RGA elements, $\lambda_{ij} < 0$.**
   The rule follows since integrity (DIC) is required for independent design, and to avoid introduction of RHP-zeros with sequential design.

3. **Prefer pairing ij, where g_{ij} puts minimal restrictions on the achievable bandwidth. Specifically the effective delay ϕ_{ij} should be small.**
   This rule favors the variable physically to each other, which makes it easier to use high-gain feedback so good performance is achieved.

Second, a design of each PID controller has to be performed. Some of the known tuning rules for decentralized PID controllers are presented in this section. The pairing should be considered regardless of which tuning method that is used, since the interactions are made as small as possible.

3.2.1 Independent design methods

Each controller element k_i is designed based on the corresponding diagonal element g_{ii} at the transfer function matrix, while satisfying some process interactions criteria. This design guarantee integrity, so failure tolerance is obtained. Due to the assumption of independent design, which does not exploit the information about other control loops, the independent design is somewhat conservative in general.

To obtain the pairing rules at the beginning of Section 3.2 are important in the independent design. Note that it is not always possible to achieve all the rules presented above. In such cases, the independent design method may fail.

After the appropriate pairing has been chosen, a controllability analysis should be preformed as in [39] at page 450, to see if it is possible to control the plant G and the disturbance model G_d with the decentralized controller K.

1. Plot the PRGA matrix $\Gamma(j\omega)$ and the Closed Loop Disturbance Gain matrix $\Gamma(j\omega)G_d$, where $G_d(j\omega)$ is the disturbance transfer function. The following criteria needs to be satisfied for each loop i.

$$|1 + L_i| > \max_{k,j}\{|\tilde{g}_{d\,ik}|, |\gamma_{ij}|\} \quad (3.21)$$

   To achieve stability, g_{ii} has to be analyzed to see if the bandwidth requirements imposed by (3.21) is achievable.

2. To avoid input constraints, plot $|g_{ii}|$ and ensure that

$$|g_{ii}| > |\tilde{g}_{d\,ik}|, \ \forall k \quad (3.22)$$

   for frequencies where $|\tilde{g}_{d\,ik}| > 1$

If the chosen pairings is controllable, the analysis based on (3.21) tells how large the loop gain $|L_i|$ must be. This can be used as a basis for designing the independent controllers $k_i(s)$. If the plant is not controllable, another choice of pairings must be considered, which most likely will not help. Another control design method than the independent procedure must then be considered.

The overall stability criteria in Section 2.7 can be used to derive bounds on PID settings for the independent design.

Robust independent design

An IMC independent design procedure is outlined in [29]. Assume the filter f in (3.9). After finding the parameter n which makes $g_{m\,ii}$ realizable, the tuning parameter $\tau_{c\,i}$ for each loop has to be determined. The parameter $\tau_{c\,i}$ has to be bounded due the interactions in the system G. Consider the RP-condition (2.58). The interconnection matrix M can be expressed as an LFT with the filter f shown in Figure 3.2. The uncertainties in $\tau_{c\,i}$ are applied as

$$\tau_{c\,i} = \frac{\tau_{c\,i}^*}{2}(1 + \Delta_{\tau_{c\,i}}), \quad |\Delta_{\tau c\,i}| \leq 1 \quad (3.23)$$

for the first order case. For higher order cases see [29].

To derive an upper bound at the filter, define

$$t_i = \frac{1}{\tau_{ci}} \tag{3.24}$$

with the uncertainty

$$t_i = \frac{t_i^*}{2}(1 + \Delta_{t_i}), \quad |\Delta_{t_i}| \leq 1 \tag{3.25}$$

The structure to the right in Figure 3.2 can be applied to Theorem 5. Note that ϵ in Figure 3.2 corresponds to τ_c.

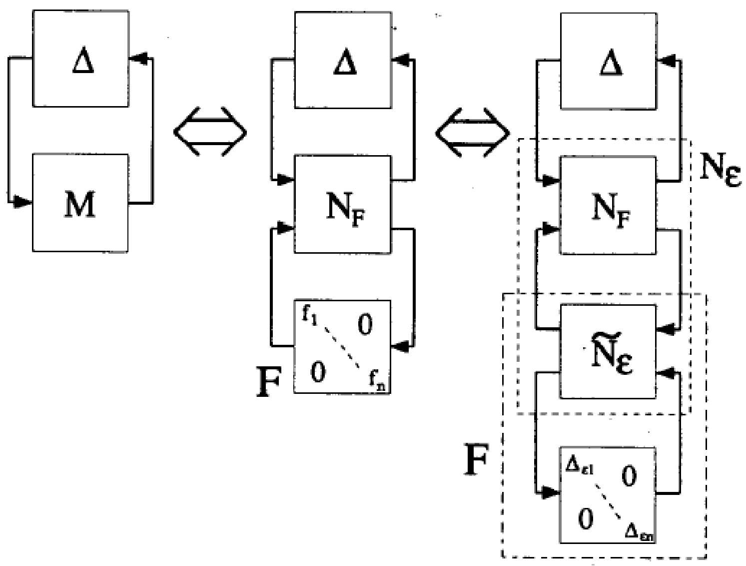

Figure 3.2: The interconnection matrix M expressed as an LFT of the IMC filter F and of the uncertainty associated with the filter time constraints. ϵ corresponds to τ_c

The procedure of the robust independent design, in [29], is as follows:

1. Define and find the structures in Figure 3.2. Different structures for N_{τ_c} and N_t should be applied.

2. Consider Theorem 5. The condition

$$\mu_\Delta(M) \leq 1 \tag{3.26}$$

is satisfied if

$$0 \leq \tau_{ci} \leq \tau_{cs}^* \quad \forall i \tag{3.27}$$

where τ_{cs}^* solves:

$$\mu_{\hat{\Delta}}(N_{\tau_c}) = 1 \quad \text{where } \hat{\Delta} = \text{diag}\{\Delta, \Delta_{\tau_c}\} \tag{3.28}$$

Similarly, let t_s^* solve $\mu_{\hat{\Delta}}(N_t) = 1$, giving the bound

$$\tau_{ci} \geq \frac{1}{t_s^*} \quad \forall i \tag{3.29}$$

3. Find the range of values of τ_{ci} so *either* (3.27) *or* (3.29) is satisfied for all frequencies.

4. Choose a value of τ_{ci} within a range of values found in Step 3, and verify the stability of M for this choice of τ_{ci}.

Effective open-loop process

To handle the interaction problems an estimate of the interactions form the other loops are designed in [16]. The resulting system is called an Effective Open-loop Process (EOP). An independent design of each EOP can then be performed. The concept of the method is here presented for a Two Input Two Output (TITO) system. The method can be extended systems with more loops, but is then more complex to use. Hence this method is not recommended for systems with 4 or more loops. The EOP's for a TITO system is defined as

$$\mathbf{g_1} = g_{11}\left\{\frac{1}{\lambda} + \frac{g_{12}g_{21}}{g_{11}g_{22}}\left(1 - \tilde{h}_2\right)\right\} \tag{3.30}$$

$$\mathbf{g_2} = g_{22}\left\{\frac{1}{\lambda} + \frac{g_{12}g_{21}}{g_{11}g_{22}}\left(1 - \tilde{h}_1\right)\right\} \tag{3.31}$$

where $\lambda = \frac{g_{11}g_{22}}{g_{11}g_{22}-g_{12}g_{21}}$ and $\tilde{h}_i$ are the complementary sensitivity functions for loop number $i = 1, 2$.

In general the SISO open loop function $l(s) = k(s)\,g(s)$, having integration in $k(s)$, can be written as

$$l(s) = \frac{\kappa(s)e^{-\theta s}}{s} \tag{3.32}$$

so for each loop, $\tilde{h}_i$ can be written

$$\tilde{h}_i = \frac{\kappa_i \frac{e^{-\theta_i s}}{s}}{1 + \kappa_i \frac{e^{-\theta_i s}}{s}} \tag{3.33}$$

In [17] the function $\kappa(s)$ has been found as

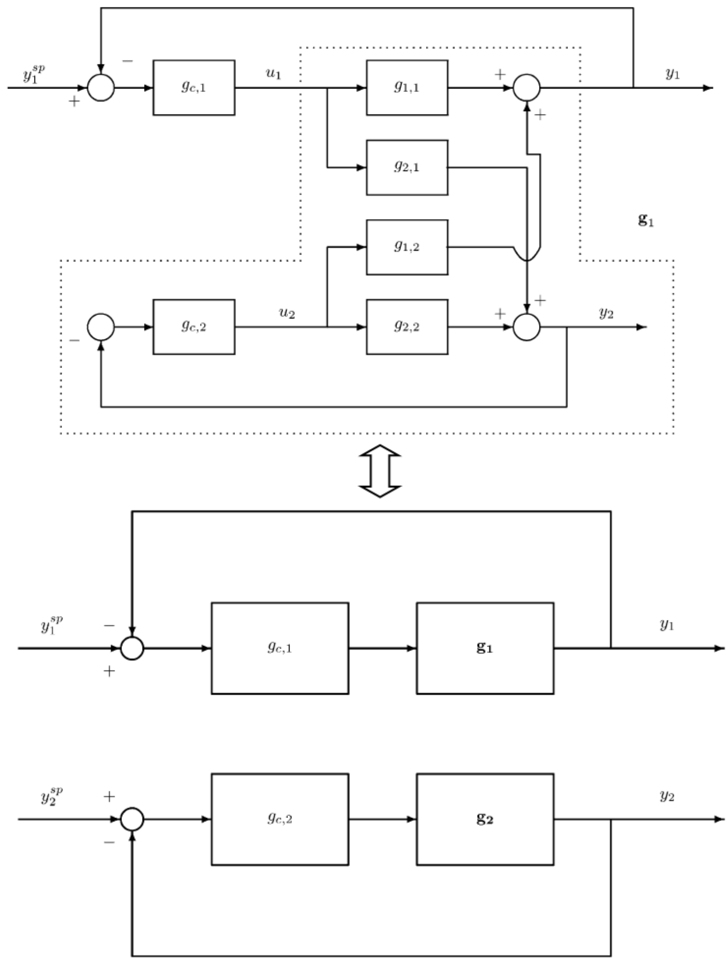

Figure 3.3: The interactions form other loops are lumped into an Effective Open-loop Process $\mathbf{g_i}$ for each loop number i. The controller $g_{c\,i}$ corresponds to the controller k_i in this thesis.

$$\kappa(s) = \frac{k_o(1 + a\theta\, s)}{\theta} \tag{3.34}$$

Small value for k_o gives robust performance, and large values give better performance. It was found in [16] that $a = 0.4$ is appropriate. Then the estimate $\tilde{h}_i^*$ of $\tilde{h}_i$ can be derived as

$$\tilde{h}_i^*(s) \approx \frac{\frac{k_{o\,i}(1+0.4\theta_i\, s)}{\theta_i\, s}}{1 + \frac{k_{o\,i}(1+0.4\theta_i\, s)}{\theta_i\, s}} \tag{3.35}$$

The parameter k_o can be used to assign the importance of each loop. Using $k_o = 0.6$, the loops are weighted equal. Estimates of the EOP can be derived if (3.35) is applied into (3.30) and (3.31)

$$\mathbf{g_1}^* = g_{11}\left\{\frac{1}{\lambda} + \frac{g_{12}g_{21}}{g_{11}g_{22}}\left(1 - \tilde{h}_2^*\right)\right\} \tag{3.36}$$

$$\mathbf{g_2}^* = g_{22}\left\{\frac{1}{\lambda} + \frac{g_{12}g_{21}}{g_{11}g_{22}}\left(1 - \tilde{h}_1^*\right)\right\} \tag{3.37}$$

The equations (3.36) and (3.37) can be used to design the controller in the open loop system $k_i\,\mathbf{g_i}^*$ independently.

Define the deviation from the real system as a multiplicative modeling error and the closed loop system in terms of the estimated EOP's

$$\delta\mathbf{g_1}^* = \frac{\mathbf{g_1} - \mathbf{g_1}^*}{\mathbf{g_1}^*} \tag{3.38}$$

$$\delta\mathbf{g_2}^* = \frac{\mathbf{g_2} - \mathbf{g_2}^*}{\mathbf{g_2}^*} \tag{3.39}$$

$$\tilde{h}_i^o = \frac{k_i\mathbf{g_i}^*}{1 + k_i\mathbf{g_i}^*} \tag{3.40}$$

Then, if the system has integrity with integral action, overall stability is ensured if the conditions in Theorem 7 are fulfilled.

Theorem 7. *A TITO system resulted from the earlier direct design procedure will be stable, if the controllers meet the following conditions.*

1. *k_1 stabilizes g_{11} and k_2 stabilizes $\mathbf{g_2}$, or*

2. *k_2 stabilizes g_{22} and k_1 stabilizes $\mathbf{g_1}$*

3. *k_i satisfies*

$$|k_i\, g_{ii}(j\omega_{p\,i})| < 1, \quad i = 1, 2 \tag{3.41}$$

where $\omega_{p\,i}$ is the phase crossover frequency of $k_i\, g_{ii}(s)$

4. *$\tilde{h}_i^o$ satisfies*

$$\left|\tilde{h}_i^o\right| < \min_{\omega}\left\{\frac{1}{|\delta\mathbf{g_i}^*|}\right\}, \quad \forall\omega \in [0, \infty\rangle\,, i = 1, 2 \tag{3.42}$$

The SISO methods in Section 3.1 can be used directly at $\bar{\mathbf{g}}_\mathbf{i}^*$. Alternatively, a tuning approach purposed in [16] can be used. Representations of $\mathbf{g_i}^*$ as a first or second order transfer function can be deriven. The parameters must then be identified.

PI/PID settings for EOP as first order dynamics

Represent $\mathbf{g_i}^*$ as $\bar{\mathbf{g}}^*$

$$\bar{\mathbf{g}}^* = \frac{k_p\, e^{-\theta\, s}}{\tau\, s + 1} \tag{3.43}$$

Use this representation to apply the following settings for the PID controller defined in (2.16)

$$\begin{aligned} \tilde{k}_c &= \frac{k_o \tau}{k_p \theta} \\ \tilde{\tau}_I &= \tau \\ \tilde{\tau}_D &= 0.4\, \theta \end{aligned} \tag{3.44}$$

For PI controllers, just ignore the derivate time constant $\tilde{\tau}_D$

PI/PID settings for EOP as second order dynamics

Represent $\mathbf{g_i}^*$ as $\bar{\mathbf{g}}^*$

$$\bar{\mathbf{g}}^* = \frac{k_p(\tau_3\, s + 1)\, e^{-\theta\, s}}{\tau_2^2\, s^2 + 2\tau_2^2 \zeta + 1} \tag{3.45}$$

Use this representation to apply the following settings for the PID controller defined in (2.16)

$$\begin{aligned} \tilde{k}_c &= \frac{2 k_o \tau_2 \zeta}{k_p \theta} \\ \tilde{\tau}_I &= 2\tau_2 \zeta \\ \tilde{\tau}_D &= \frac{\tau_2}{2\zeta} \end{aligned} \tag{3.46}$$

3.2.2 Detuning methods

Biggest log modulus for PI and PID controllers

The detuning method presented below for PI controllers were first presented in [50]. An extension to this method, including derivative action and tuning based at the importance of the different loop performances, is presented in [34].

In terms of (2.3), the closed loop log modulus for SISO systems is defined in [50] as

$$L_c = 20\,\log\left|\frac{GK}{1+GK}\right|, \tag{3.47}$$

If $L_c^{\max} = \max_\omega L_c > 2$ dB, then the GK contour is far enough away from the point $(-1, 0)$ in the Nyquist plot defined in [45] at page 195, and the SISO tuning is acceptable.

In order to use the Nyquist method for MIMO systems, W is defined as

$$W(s) = -1 + \det(I + GK)(s), \tag{3.48}$$

and applied in a Nyquist plot. The multivariable closed-loop log modulus L_{cm} is defined as

$$L_{cm} = 20 \log \left| \frac{W}{1+W} \right|, \tag{3.49}$$

For a $N \times N$ system, the criteria

$$L_{cm}^{\max} = 2N\text{dB}, \tag{3.50}$$

is proposed. This is defined as the "Biggest Log modulus Tuning" (BLT), and gives a reasonable compromise between stability and performance. It is important to apply the Nyquist plot of W, because L_{cm} (and L_c) merely denotes nearness to the critical point $(-1, 0)$. Hence an unstable system could be tuned even if $L_{cm}^{\max} = 2\,N$ dB. To achieve the criteria in (3.50), the procedure below is proposed in [50] for PI controllers.

PI BLT tuning

1. The settings for each individual loop is found using some SISO-tuning rule as Ziegler-Nichols, SIMC or Tyreus and Luyben presented in 3.1.
2. The control gain k_{ci} and τ_{Ii} is defined as

$$k_{ci} = \frac{k_{SISO_i}}{F}, \qquad \text{for } i = 1, 2 ... N \tag{3.51}$$

$$\tau_{Ii} = F\,\tau_{I-SISO_i}, \qquad \text{for } i = 1, 2 ... N \tag{3.52}$$

   where k_{SISO_i} and τ_{I-SISO_i} is the SISO-loop settings for loop number i. Assume factor F, that typically vary form 2 to 5.
3. When a factor F is found so (3.50) is satisfied, the tuning settings are acceptable.

In [50] the Ziegler-Nichols settings are used as the SISO settings. The system will be more stable and more sluggish if F is large. The same "detuning" factor F can be applied in all loops, but if some variables need tighter control, different weighting factors F_i can be used in the individual loops. This procedure guarantees that the system is stable with all controllers in automatic, and also that each loop is stable if all others are in manual. Hence integrity is not guaranteed for this procedure.

PID BLT tuning
This method, presented in [34], incorporates derivative action as follows;

1. Choose a detuning factor $F_D \geq 1$
2. Compute τ_{Di} for loop number i as

$$\tau_{Di} = \frac{\tau_{D-SISO_i}}{F_D} \tag{3.53}$$

   where τ_{D-SISO_i} is the Ziegler-Nichols value for τ_{D_i}.

3. Calculate W as (3.48) and $L_{cm}^{\max}$
4. Change F_D until $L_{cm}^{\max}$ is minimized, maintaining $F_D \geq 1$.
5. Reduce F, and calculate k_{ci}, τ_{Ii},W and $L_{cm}^{\max}$ until $L_{cm}^{\max} = 2\,N$
6. Repeat step 4 and 5 until no further reduction in F is possible.

By this procedure the derivative term is tuned so they have the greatest effect near the closed-loop resonant frequency. This recursive procedure converged for all cases studied.

BLT loop weighting
The Integral Total Error (ITE) for the i'th controlled variable u_i can be computed as

$$\int_0^\infty e_i(t)\,dt = \frac{u_i(\infty)\tau_{I\,i}}{K_{c\,i}} \tag{3.54}$$

The steady state value of $u(\infty)$ drives $y(\infty)$ to r can be expressed as

$$u(\infty) = G^{-1}(0)y(\infty) \quad \text{where } y(\infty) = [0, ..., 0, 1, 0..., 0] \tag{3.55}$$

Then [34] propose a way of computing the absolute ITE's for disturbance in each set-point and for a load disturbance as

$$J_i = \text{abs}\left(\frac{\tau_{I\,i}}{K_{c\,i}}\right)\sum_{j=1}^{N}\left\{\text{abs}\left[\frac{g_{ij}^{-1}(0)}{N}\right]\right\} \tag{3.56}$$

It is desirable that the absolute ITE's of each loop are equal. The individual detuning factor F_i for each loop is then

$$F_i = F\sqrt{\frac{J_{\max}}{J_i}} \tag{3.57}$$

where $J_{\max} = \max_i J_i$. The PI controller settings are then computed as

$$k_{ci} = \frac{k_{SISO_i}}{F_i}, \tag{3.58}$$

$$\tau_{Ii} = F_i\,\tau_{SISO_i}, \tag{3.59}$$

For a 2×2 system, it is shown that this procedure result in a tighter y_2 loop, and a looser y_1 loop than the tuning in [50]

Combination of PID BLT and loop weighting tuning
This procedure is presented in [34].

1. The weighting factor $\sqrt{\frac{J_{\max}}{J_i}}$ is found for each loop.

2. The PID BLT tuning approach is used with individual $F_{D\,i}$ factors for each loop

$$F_{D\,i} = F_D \sqrt{\frac{J_{\max}}{J_i}} \tag{3.60}$$

This method has proved to provide an improvement over the method in [50].

The Nyquist contour must be checked every time F is changed, since the DIC or RDD property is not guaranteed.

Detuning factor based on a diagonal dominance index

Define

$$g_{ki}(j\omega) = a_{ki}(\omega) + jb_{ki}(\omega) = r_{ki}(\omega)\, e^{j\phi_{ki}(\omega)} \tag{3.61}$$

$$R_i(\omega) = \sum_{i=1,k\neq i}^{n} |g_{ki}(j\omega)| \tag{3.62}$$

When the relations in (3.61) and (3.62) are applied into the squared of (2.61), the interaction criteria that needs to be satisfied to guarantee closed loop stability, according to Theorem 6, is

$$k_{c\,i}^2\left(r_{ii}^2(\omega) - R_i^2(\omega)\right) + k_{I\,i}^2\frac{r_{ii}^2(\omega) - R_i^2(\omega)}{\omega^2} + 2\left(k_{c\,i}a_{ii}(\omega) + (k_{I\,i}\frac{b_{ii}(\omega)}{\omega}\right) + 1 > 0 \tag{3.63}$$

The equation (3.63) is used to derive regions for the parameters $k_{c\,i}$ and $k_{I\,i} = \frac{k_{c\,i}}{\tau_{I\,i}}$ so closed loop stability is ensured.

To guarantee stability for $\tilde{H}(s)$, bounds for the individual SISO systems g_{ii} has been derived in [44] as

$$k_{c\,i} = -\frac{a_{ii}(\omega)}{r_{ii}^2(\omega)} \tag{3.64}$$

$$k_{I\,i} = -\omega\frac{b_{ii}(\omega)}{r_{ii}^2(\omega)} \tag{3.65}$$

where the definition in (3.61) have been used.

The performance may vary significantly within the stability region. The tuning procedure purposed in [8] is as follows:

1. Calculate the stability region for each PI controller by obtaining the intersection of (3.63), (3.64) and (3.65). The ultimate gain k_{ui} and the ultimate frequency ω_{ui} lies at the stability boundary shown in Figure 3.4 when the integral gain $k_{I\,i}$ is fixed to zero. Due to process interactions, the ultimate gain and frequency is different when only a SISO loop with g_{ii} is considered.

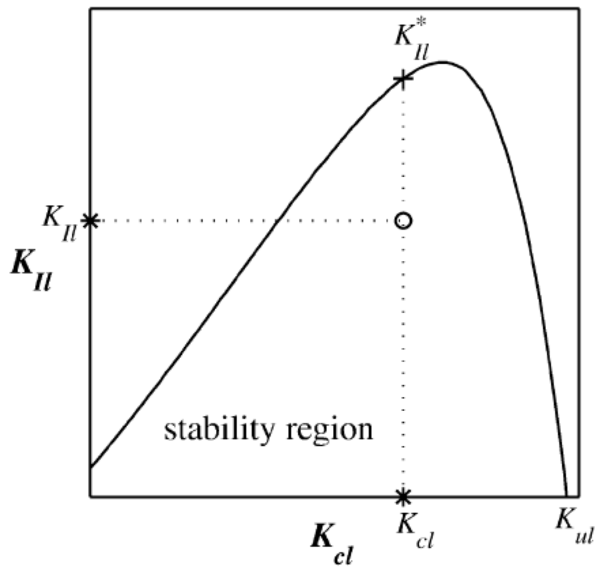

Figure 3.4: Stability region found by (3.63), (3.64) and (3.65).

2. A detuning factor F_i is calculated for each loop. Use the *column diagonal dominance index* at the ultimate frequency ϕ_{ui}, defined as

$$\phi_{ui} = \phi_i(\omega_{ul}) \tag{3.66}$$

$$\phi_i(\omega) = \frac{|G_{ii}(j\omega| - \sum_{i=1,k\neq i}^{n} |G_{ki}(j\omega)|}{|G_{ii}(j\omega|} = 1 - \frac{R_i(\omega)}{r_{ii}(\omega)} \tag{3.67}$$

to determine which F_i to use.

$$F_i = \begin{cases} 0.75 & \text{if } \phi_{ui} \leq -1.5 \\ 0.375 - 0.25\phi_{ui} & \text{if } -1.5 < \phi_{ui} \leq -0.5 \\ 0.5 & \text{if } -0.5 < \phi_{ui} \leq -0 \\ 0.5 - 0.25\phi_{ui} & \text{if } 0 < \phi_{ui} \leq 1 \end{cases} \tag{3.68}$$

The regions of F_i are based at empirical studies of simulation results of 14 cases outlined in [8].

3. The controller settings applied are

$$k_{ci} = k_{ui}F_i \tag{3.69}$$

$$k_{Ii} = k_{Ii}^*F_i \tag{3.70}$$

where the gain k_{Ii}^* is the integral gain that gives marginal stability with the corresponding k_{ci} calculated in (3.69). In Figure 3.4 it is shown that this is at the stability boundary in the (k_{ci}, k_{Ii}) plot. The resulting stability gains are located inside the stability bound, since F_i is always less than one.

The overall stability region for (k_{ci}, k_{Ii}) for the ith PI controller is the intersection of (3.63) and the region bounded by (3.64) and (3.65). Hence this tuning procedure could be categorized in the Independent design methods. But due to the fact that a detuning factor is used to guarantee that the parameters are within the overall stability region, it has been grouped under the Detuning methods.

When different control settings were tested in [48] this procedure was suitable due to reference tracking and disturbance rejection of control of a four tank system.

A simplification of this method, only valid for diagonal dominant systems, is presented in [9]. To find the ultimate gain for each loop, k_{ui}, use the boundary values of (3.63) with proportional controller only. Hence fix $k_{Ii} = 0$.

$$k_{ci}^2 \left(r_{ii}^2(\omega) - R_i^2(\omega) \right) + 2k_{ci}a_{ii}(\omega) + 1 = 0 \tag{3.71}$$

Since only a real-valued solution of k_{ci} is valid, the ultimate gain and ultimate frequency for the ith Greshgorin band can be expressed as

$$\omega_{ui} = \{\omega \,|\min |k_{ci}| \; \forall k_{ci} \in \text{ real valued solutions of (3.71)}\} \tag{3.72}$$

$$k_{ui} = k_{ci}(\omega_{ui}), \quad k_{ci} \in \text{ real valued solutions of (3.71)} \tag{3.73}$$

When the ω_{ui} and k_{ui} are determined the Ziegler-Nichols or Tyreus-Luyben, presented in Section 3.1, with a detuning factor F can be applied.

3.2.3 Optimization methods

If the control parameters can be optimized, the controller will give superior performance due to handling of all the interactions in the MIMO systems. It can also be used to stabilize an unstable plant. Unfortunately, the optimization problem may be both hard to design and hard to solve due to non-convexity. Few of the normal benefits of decentralized control mentioned in Section 1.1.1 is offered. Hence, as pointed out in [39] at page 429, full multivariable controllers, which lead to better performance, should be considered instead. Exceptions are when centralized control can not be used, e.g. for geographical reasons or when the existing decentralized control configuration is required for some other reason. Hence this approach is not common in practice. This method can only be used for off-line tuning, because of the time consuming computation load. A process model and simulator tests may therefore be required.

Genetic optimization

In [20] a detailed overview of Genetic Algorithms is outlined. Figure 3.5 shows the pseudo code for the Genetic Algorithm.

In [49] a genetic approach for PI tuning for a decentralized controller is proposed. The objective function can be based on some specified performance objectives or constraints. The response could be compared to a desired shape of $H(s)$ in (2.5) to some specified input. Peak overshoot, settling time, rise time, steady-state offset, velocity lag etc. may

```
t=0;
initialize P(t); % P(t) is the population at time t
evaluate P(t);
while (termination contition not satisfied) do
    t = t+1;
    select P(t);
    recombine P(t);
    evaluate P(t);
end;
```

Figure 3.5: Psedocode of a Genetic Algorithm.

be used to evaluate the performance. For an input reference at the jth input, define upper and lower performance objectives, f^l_{ij} and f^u_{ij}, for the ith output. In [32] these are called zone objectives. The objective function to be minimized is

$$J_0(K_P, T_I) = \max_{1\le j\le n}\left\{\sum_{i=1}^{n} w_{ij} J_{ij}(K_P, T_I)\right\} \tag{3.74}$$

where w_{ij} is a weighting factor to distinguish the importance of the different output patterns, and

$$J_{ij}(K_P, T_I) = \int_0^{t_{\max}} \left[\max\left\{f^l_{ij}(t) - y_i(t), 0\right\} + \max\left\{y_i(t) - f^u_{ij}(t), 0\right\}\right] dt \tag{3.75}$$

is the objective function for the ith process output and the jth set point pattern. This makes the objective function flexible.

Due to the fact that the control structure is fixed, the existence of a general solution for arbitrary performance objectives is not guaranteed.

H_∞ optimal robust PID controller

A robust multi-loop PID controller design is outlined in [2].

An output multiplicative uncertainty model, shown in (2.38), is considered. The matrix $\Delta(s)$ is all possible complex perturbations. It is assumed that the system $w_O(s)\Delta(s)$ is stable, and $\|\Delta\|_\infty$.

Then the H_∞ problem can be stated as

$$\min_{K\in K_{PID}} -\gamma \tag{3.76}$$

$$\text{subject to} \tag{3.77}$$

$$\|H_{cl}(s)\|_\infty = \left\| \begin{array}{c} w_O(s)H(s) \\ \gamma w_P(s)S(s) \end{array} \right\|_\infty < 1 \tag{3.78}$$

$$H_{cl}(s) \in RH^\infty \tag{3.79}$$

where $w_O(s)H(s)$ ensures Robust Stability, and $\gamma w_P(s)S(s)$ ensures the best nominal performance possible according to the performance weight w_P selected. The optimal level of γ indicates the level of performance achieved. If $\gamma > 1$, better performance is obtained while $\gamma < 1$, it is not possible to meet the performance specification. Equation (3.79) ensures that $H_{cl}(s)$ is analytic in the RHP, and is therefore stable.

The H_∞ problem in (3.76) is converted into a state space realization using the Bounded Real Lemma outlined in [2]. The optimization problem is then

$$\min_{\mathbf{x}} -\gamma \tag{3.80}$$

$$\text{subject to} \tag{3.81}$$

$$\left[\begin{array}{ccc} A_{cl}^T(\mathbf{x})X(\mathbf{x}) + X(\mathbf{x})A_{cl}(\mathbf{x}) & X(\mathbf{x})B_{cl}(\mathbf{x}) & C_{cl}^T \\ B_{cl}^T(\mathbf{x})X(\mathbf{x}) & -I & D_{cl}^T \\ C_{cl} & D_{cl} & -I \end{array} \right] < 0 \tag{3.82}$$

$$X = X^T \tag{3.83}$$

$$X > 0 \tag{3.84}$$

where A_{cl}, B_{cl}, C_{cl} and D_{cl} are the state space realization of the closed loop system. The matrix X is a symmetric positive definite optimization variable.

A method called Successive Semidefinite Programming is used to iterate to a solution of (3.80). The optimization problem in (3.80) is not convex, hence the purposed algorithm only guarantees to find a local optimum. Several starting points have be used to aim for a global solution. This tuning method works only for time continuous systems, and any time delays have to be approximated by a Padé approximation shown in (2.30).

3.2.4 Sequential loop closing methods

Sequential design involves closing and tuning one loop at a time with the previously designed controllers implemented. Initially all loops are open, so the system must be open stable. A new transfer function has to be derived after each closing, where the control information of the closed loops are included. The method is suited for on-line tuning using a relay, presented in Section 3.2.6. The pairing rules in the beginning of this chapter are still desirable, but may be violated. Then the interactions are used to achieve desired performance. Some advantages and disadvantages of the sequential design procedure are presented in [30].

Advantages of sequential design

1. Each step in the design procedure involves designing only one SISO controller.

2. If stability has been achieved after the design of each loop, then the system will remain stable if loops fail or are taken out of service in the reverse order of how they were designed.

3. Stability is ensured during startup if the loops are brought into service in the same order as they have been designed.

Disadvantages of sequential design and solutions

1. The quality of the final controller may depend on the order of the individual SISO loop design. Hence, the fast loops are closed first, because the loop gain and phase in the bandwidth region of the fast loops is insensitive to the tuning of the lower loops. The response in a fast loop may still be sensitive to the tuning of a slower loop, if the interactions between the slow and fast loop is severe.

2. The closing of subsequent loops may alter the response of previously designed loops, and thus make iteration necessary. Hence estimates of how the undesign loops will affect the output of the loops to be design can be made. How to include these estimates in the sequential design process is found in [30] and outlined below.

3. The transfer function $g_{ii}(s)$, which is used to design the i'th SISO loop, may contain RHP zeros that do not correspond to RHP transmission zeros in $G(s)$. Hence this will affect the order of the closing, since RHP zero induce bandwidth limitation outlined in [39] at page 45. However the RHP zeros of the individual $g_{ii}(s)$ may disappear when other loops are closed. Hence it may be possible to neglect these limitations if the loop are included in the design at a later stage.

4. Due the nature of the tuning procedure, failure tolerance is not guaranteed if the inner loops fail.

While the independent design presented in Section 3.2.1 is suited for loops that are decoupled in space (the interactions are small), the sequential design is suited for loops that are decoupled in time. If the previous loops are considerably faster than the one to be designed, then perfect control may be assumed for the inner loops. Then the previous controllers can be set to infinity when the transfer function for the loop to be designed is derived. This is explained in [39] at page 430.

Sequential design with estimates of undesigned loops

To evaluate the effect of undesigned loops, the following functions are defined when the k'th loop is to be closed. The system matrix $(\cdot)_k$ is the upper left subsystem of $(\cdot)$ with dimension $k \times k$. Then

$$K_k \triangleq \text{diag}(k_1, k_2 ... k_k) \tag{3.85}$$

$$S_k \triangleq (I + G_k K_k)^{-1} \tag{3.86}$$

$$H_k \triangleq G_k K_k (I + G_k K_k)^{-1} \tag{3.87}$$

$$\hat{G}_k \triangleq \text{diag}(G_k, G_{ii}) \qquad i = k+1, k+2 ... N \tag{3.88}$$

$$\hat{S}_k \triangleq \text{diag}(S_k, \tilde{S}_i) \qquad i = k+1, k+2 ... N \tag{3.89}$$

$$\hat{H}_k \triangleq \text{diag}(H_k, H_i) \qquad i = k+1, k+2 ... N \tag{3.90}$$

$$\hat{E}_k \triangleq (G - \hat{G}_k)\hat{G}_k^{-1} \tag{3.91}$$

Then

$$S = \hat{S}_k (I + E_k \hat{H}_k)^{-1} \tag{3.92}$$

$$\tag{3.93}$$

where $(I + E_k \hat{H}_k)^{-1}$ is the estimated interactions from the undesigned loops. In [30] the SISO controllers have been design so that the objective function

$$\min_{k_i} \|W_P S W_D\| \tag{3.94}$$

is minimized where W_P and W_D are some weighting functions. This could be done by a H_2- or H_∞-norm. If (3.92) is inserted in (3.94), then

$$\|W_P S W_D\| = \left\|W_P \hat{S}_k (I + E_k \hat{H}_k)^{-1} W_D\right\| = \left\|W_P \hat{S}_k \hat{W}_D\right\| \tag{3.95}$$

where

$$\hat{W}_D = (I + E_k \hat{H}_k)^{-1} W_D \tag{3.96}$$

The method outlined in [30] can be summarized as follows.

1. Determine the order of the closing by estimating the required bandwidth in each loop. Estimate the individual loop design in terms of their complementary sensitivity $\tilde{H}$. If the system is scaled, as in Section 2.1.3, it is required that for $r_i < 1$

$$|[S_{ij}]| < 1 \tag{3.97}$$

   respectively. The following applies for frequencies lower than the bandwidth;

$$\Gamma \approx (I + E\tilde{H})^{-1} \qquad \omega < \omega_B \tag{3.98}$$

$$\tilde{S} = (\tilde{G}K)^{-1} \tag{3.99}$$

$$e = y - r = -\tilde{S}\Gamma r \tag{3.100}$$

where $\Gamma = \{\gamma_{ij}\}$ is the PRGA in (2.9). The performance requirement in (3.97) can be rewritten as bounds on the individual loop gains $G_{ii}k_i$ as

$$\left|\frac{\gamma_{ij}}{G_{ii}k_i}\right| < 1 \Leftrightarrow |G_{ii}k_i| > |\gamma_{ij}| \qquad \omega < \omega_B \tag{3.101}$$

This requirement is very helpful when the bandwidth estimation is needed.

2. Design the controller k_1 by considering output 1 only. Use the objective

$$\min_{k_1} \left\| W_P^{n_p \times 1} S_1 \hat{W}_D^{1 \times n_D} \right\| \tag{3.102}$$

analogous to (3.95), where $\hat{H}_k = \tilde{H}$. The function S_1 needs to be stable.

3. Design controller k_k by considering outputs 1 to k. Use

$$\min_{k_k} \left\| W_P^{n_p \times k} S_k \hat{W}_D^{k \times n_D} \right\| \tag{3.103}$$

where $\hat{H}_k = \text{diag}(H_{k-1}), \tilde{H}_i$ for $i = k, k+1 \ldots N$. H_{k-1} is used instead of H_k, since it is independent of k_k. The function S_k needs to be stable.

4. Design the last controller k_n. This is done by considering the overall problem in (3.94). The function S needs to be stable.

Other functions than S could be used to evaluate closed loop performance. Since S_k needs to be stable at each step, a limited degree of failure tolerance is guaranteed. Since the system is required to be stable after closing each loop, it is not possible in general to obtain the optimal decentralized controller with this approach.

Robust sequential design

A sequential design procedure presented in [28], that ensures robustness for systems with input multiplicative uncertainty. The complementary sensitivity function H are defined with LFT as

$$H = N_{11}^{hh} + N_{12}^{hh} T^h \left(I - N_{22}^{hh} T^h\right)^{-1} N_{21}^{hh} \tag{3.104}$$

where N^{hh} and T^h consists of estimates of the sensitivity and the complimentary sensitivity functions of the undesigned loops shown in [28]. In [18] it is shown how the M in (2.45) are represented as a LFT with the complementary sensitivity H. Then (3.105) is put into this representation, and M is expressed as the LFT

$$M = N_{11}^{h} + N_{12}^{h} T^h \left(I - N_{22}^{h} T^h\right)^{-1} N_{21}^{h} \tag{3.105}$$

The elements N_{ij}^h contain the elements N_{ij}^{hh}. Then the robust conditions can be stated analogous to (2.58).

$$\bar{\sigma}(\tilde{h}_i) < \psi_h^k \quad \forall i = k+1, ..., N,\ \omega \in [0, \infty\rangle \tag{3.106}$$

where ψ_h^k solves

$$\mu_{\{\Delta_I, \Delta_P, \bar{H}^k\}} \begin{bmatrix} N_{11}^h & N_{12}^h \\ \psi_h^k N_{21}^h & \psi_h^k N_{22}^h \end{bmatrix} = 1 \tag{3.107}$$

where $\tilde{h}_i$ is the elements in (2.13). Note that $\bar{H}^k$ is not the same as the estimate $\hat{H}_k$ in (3.90). Equal bounds can be derived for $\tilde{S}_i$.

The fastest loop is still closed first. This design ensures robust performance, and due the nature of sequential design, the possibility of less conservative design than the robust independent design is present. But still, integrity is not guaranteed.

3.2.5 Iterative or trial-and-error methods

Iterative methods are much like sequential loop tuning, since all loops start open, and each loop are tuned sequentially. The tuning of each loop is then redone with all the other loops closed. This is repeated until the controller settings converge. This method is also called trial- and- error methods.

Multi-loop iterative continuous cycling method

In [26] a trial and error method uses a multi-loop version of the ICC in Section 3.1.4 is presented. The multi-loop ICC method for PI controllers redefines the Nyquist array method to provide less conservative stability conditions. It is pointed out in [8] that this method does not guarantee closed loop stability due to the nature of the continuous cycling method.

Assume that the decentralized controller $K(s)$ has the form

$$K(s) = K_c(I + \frac{1}{s}K_I) \tag{3.108}$$

where $K_c = \text{diag}\{k_{c1}, k_{c1}...k_{cN}\}$ is the proportional gain matrix, and $K_I = \text{diag}\left\{\frac{1}{\tau_I 1}, \frac{1}{\tau_I 2}, ...\frac{1}{\tau_I N}\right\}$. The method presented in [26] is as follows:

1. Set $K_I = 0$ and use the condition in (2.51) to design the proportional gain matrix K_c. Improve this gain once by applying (2.54) using $\widetilde{H}$ with the previous calculated proportional controllers. Set the K_c to the half of the gains calculated also.

2. The characteristic equation for the system with PI control is

$$\det(I + GK) = \det\{I + GK_c\} \det\left\{I + (I + GK_c)^{-1} GK_c K_I \frac{1}{s}\right\} = 0 \tag{3.109}$$

Then K_{0I} can be design by applying the rearranged process

$$[I + GK_c]^{-1} GK_c \frac{1}{s} \tag{3.110}$$

to (2.51) with K_c from Step 1. Set $K_I = 0.4\, K_{0I}$.

3. To ensure a gain margin of 2, K_c is adjusted again. Condition (2.54) is used at the process with integral action

$$G\left(I + K_I \frac{1}{s}\right) \tag{3.111}$$

with $\tilde{H}$ designed in step 1 and 2. Condition (2.54) is applied two times iteratively.

This method is less conservative, but more complex to use than the method outlined in 3.2.2. Due to the iterations, the integrity is not ensured with this method.

3.2.6 Multi-loop relay feedback approaches

To use the tuning methods presented above, the transfer matrix $G(s)$ must be known. The transfer matrix can be obtained by mathematical modeling of the system, or by using system identification techniques.

In [6] at page 99, it is stated that for multivariable systems, a relay feedback test can be preformed while having the rest of the controllers turned off. Then the parameters for each individual element in the corresponding column can be identified. For a $N \times N$ system, all the transfer functions g_{ij} are found after N tests. In principle all the identification techniques for SISO systems described in [12] can be applied. When $G(s)$ is obtained, all the tuning methods presented above can be used, such as the independent design method or the detuning methods.

In MIMO systems, due to system interactions, there are infinite numbers of ultimate points $(k_{u\,1}, k_{u\,2}...k_{u\,N}, \omega_u)$ consisting of the ultimate gain in each loop and the corresponding frequency. Hence simple rules based at the ultimate points can not be used directly as pointed out in [11]. Tuning based at different critical points leads to different performance. The ultimate parameters must be obtained by sequential relay tests, or replacing all the controllers with relays.

If a sequential or iterative approach is used, it is not necessary to find all the g_{ij} elements. A sequential design is presented in [27], and a iterative design is presented in [43] and [6] at page 123. The iterative method using relay feedback, presented in [43] and [6], is here restated.

1. Close the loops after how fast they are. Start with the fastest loop

2. Perform the relay feedback test on loop 1 and design the corresponding controller using Ziegler-Nichols. Close the loop.

3. Perform the relay feedback test at loop k, while loop 1 to $k-1$ is closed and design the corresponding controller using Ziegler-Nichols. Repeat this until all loops are closed.

4. Perform the relay feedback test at each loop again, starting with the fastest one, while every other loop are closed. Re-tune the corresponding controller using Ziegler-Nichols. Repeat this step until the parameters converge.

It is pointed out that it is typically needed $2N - 1$ relay tests for this method. The method in [27] is basically the same, only the controllers are not retuned.

In [11], all controllers are replaced with relays at the identification phase, and limited cycles with the same period in all loops are generated. Each loop is weighted by importance. A method that converge to a desired ultimate point $(k_{u\,1}, k_{u\,2}...k_{u\,N}, \omega_u)$ based at the weighting is obtained. Based on this ultimate point, Ziegler-Nichols settings are applied for each controller.

3.3 Discussion and conclusion of the MIMO tuning survey

Ideally, all the tuning methods should have been tested with high interactive test benches with model uncertainty, and compared to each other. This would have revealed the strengths and weaknesses clearly. In [48] some tuning methods have been tested to a high interactive plant. The BLT method with Ziegler-Nichols and Tyerus and Luyben settings, the sequential relay method with Ziegler-Nichols and Tyerus and Luyben settings, ideal relay method with BLT[1], the simultaneous relay method and the Gershgorgin band method using Ziegler-Nichols settings were compared. The Gershgorgin band method presented last in Section 3.2.2 and the ideal relay method showed the best performance in the case study of a high interactive plant. Such a comparison is not done here. The methods presented above are discussed based at the presentation above.

As mentioned in the introduction, it is desired to have robust controllers with disturbance rejections and failure tolerance. Almost every publications dealing with PI/PID tuning treats set-point reaching. Hence, which procedure that handles disturbance rejection best, cannot be determined. Interaction handling and integrity can be discussed independently whether the examples deal with set-point tracking or disturbance rejections.

Most of the methods rely on a mathematical model of the system. But not all tuning methods need a detailed model. Sometimes the model includes uncertainty and performance weighs to take account for model errors, and ensure a specific performance. But of course, model errors could also occur in the weighting functions, and the robust stability and performance properties rely on good model descriptions.

Independent design methods

The main advantage of independent design is that integrity is automatically obtained. To use this method, the system must be open stable and pairings of inputs and outputs

[1] It is not clear in [48] "Ideal Relay tuning" is, but based at literature such as [6] it is probably a rely feedback test performed in each SISO- loop while the others are open. From the diagonal model achieved, the BLT settings are used.

must be chosen so the controllability test in Section 3.2.1 is satisfied. The method is conservative, because only necessary overall stability conditions are taken into account and it does not exploit information in other loops. Hence it is suited for plants where interactions are low, but difficult for strongly interactive plants. The EOP design method uses an estimate of interactions to include these effects in the independent loop design to utilize better performance. But this method is very complex for systems with four or more loops. The robust independent approach ensures that the system is RDD. Here the uncertainty and performance weights in Section 2.4 and Section 2.5 are included in the design. This method relies on a mathematical model.

Detuning methods

Detuning procedures are simple to use. Open stable systems are considered. By detuning the controllers, the interactions from the other loops are reduced. Hence a trade off between performance versus stability is made. The BLT method is often used as comparison for performance of newer methods. A drawback is that it often leads to a too conservative design. This design only ensures stability when every controller is intact or when only one controller is intact. Consequently, integrity is not guaranteed. Uncertainty and performance weights are not used. Hence robustness and performance properties are not quantified. But it is clear that if the detuning factor is high, the tuning is conservative, but more robust, since the influence of the other loops is reduced.

Sequential closing methods

The sequential closing method is simple, but may result in too conservative tuning. The system must be open stable. It can be used for interactive plants where the system can be decoupled in time. Hence, the method is suited when response time differences in the closed loops are acceptable. Desired robustness and performance can be achieved for the overall system, and for the sub-systems generated in the closing procedure. Failure tolerance is not guaranteed when the inner loops fail, but stability and performance are guaranteed when the loops are brought in and out in the design order. It is suited for on-line tuning.

Iterative or trial and error methods

Iterative or trial and error methods are similar to the sequential loop closing method. Several of the advantages and disadvantages for the sequential closing method also applies for this method. Since retuning of the loops are done until the parameters converge, this gives generally less conservative settings than the sequential tuning. But since the settings are altered after all loops are closed, this implies that performance and stability are not guaranteed when the loops are brought in and out of service in the design order. Overall stability is not guaranteed with the multi-loop ICC method.

Optimization methods

Optimization methods offer superior interaction handling, but is time consuming. Preferably, accurate models of the process and the uncertainty descriptions are needed. With decent uncertainty weights, the plant model is not restricted to be accurate. But modeling error could occur in the uncertainty weights, and the obtained settings should be tested with a plant simulator if such exists. Hence these procedures are suited for off-line tuning. Due to non-convexity, the optimization problems may be hard to solve. Since this is a full coordinated design, all the interactions are taken into account when the settings are found. Consequently integrity is most likely not obtained. It is well suited for finding settings that stabilize the plant. With a H_∞- control framework the robust stability and nominal performance with respect to some weighting functions can be found.

Relay feedback approaches

With the auto-tuning approach, a process model is not needed. The system must be open stable. This is a very practical method, since no modeling is required. The integrity property is not mentioned in any of the publications dealing with this tuning method. When the ultimate points are found sequentially, this method should have the same properties as the sequential loop closing method. In general the ultimate points are related to the various interactions. If some loops are shut down, the ultimate points may be different, and the settings obtained may also differ. Hence it can be argued that the settings do not give integrity. It is stressed that this is an assumption, and has not been proved.

Overall conclusion and further work

The only method where failure tolerance is guaranteed is the independent method. But this method is not suited for high interactive plants, and may lead to a conservative design. With an appropriate model of the system and the weighting functions, the H_∞ optimization method in [2] handles interactions superiorly and guarantee robust stability and nominal performance. Since all the interactions are considered at once, integrity is not guaranteed. It could be interesting to find a similar optimization problem that gives settings so the remaining system is stable while some loops are shut down. If integrity with integral control is possible, such an optimization algorithm should give the optimal performance, integrity and robust stability at the same time. This is of course under the assumption that suitable uncertainty weights are chosen. Then all the requirements of the PI/PID multi-loop control system in Section 1.3 are achieved. Hence, the procedure in [2] is investigated to see if so improvements can be made such that failure tolerance can be guaranteed.

Chapter 4

H_∞ optimal multi-loop tuning

Conventional H_∞ methods outlined in [39], leads to a full MIMO controller with several states. A full multivariable PID controller can be obtained if model reduction are performed on the conventional H_∞ design, shown in [46]. In [33] and [7], the H_∞ norm of the weighted sensitivity function and the complementary sensitivity function is minimized to derive the PID settings for SISO systems. These SISO designs are extended to handle multi-loop control design, outlined in [2]. This optimization algorithm can be used to achieve nominal performance and robust stability with superior interaction handling.

In this chapter, it is explained how to use the H_∞ multi-loop tuning approach, presented in [2], so failure tolerance is guaranteed. This method should guarantee a system with integrity if such a property is possible with integral control. First a detailed survey of the method is presented to get the proper insight of the method.

4.1 Detailed survey of H_∞ optimal PID tuning approach

4.1.1 Defining the optimization problem

In this procedure, an output multiplicative uncertainty model with a performance weight at the sensitivity function S is considered as shown in Figure 4.1. The matrix $\Delta(s)$ is a full complex matrix, restricted by

$$\|(\Delta)\|_\infty < 1 \tag{4.1}$$

The system $w_O(s)\Delta(s)$ is assumed to be stable.

Any system can be written with the $PK\Delta$ representation in Figure 2.2. If the plant is nominal, $\Delta = 0$. Conventional H_∞ problems find a controller K so the lower LFT of the PK system is minimized, as shown in [39] at page 357.

$$\min_K \|F(P,K)\|_\infty = \min_K \|N\|_\infty \tag{4.2}$$

The system N is derived in (2.43), and is shown in Figure 2.3. This leads to full MIMO controllers with several of states.

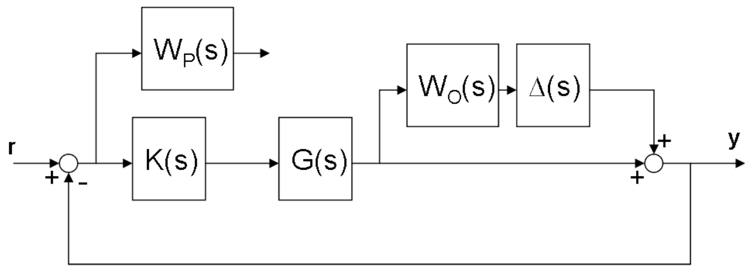

Figure 4.1: System block diagram of output multiplicative uncertainty

In (2.46) to (2.49) nominal stability, nominal performance, robust stability and robust performance are defined. Theorem 8 defines robust stability when the perturbation Δ is a full matrix. The $M\Delta$ structure in Figure 2.4 is assumed, where M is defined in (2.45).

Theorem 8. RS for unstructured perturbations. *Assume that the nominal system $M(s)$ is stable (NS) and that the perturbations $\Delta(s)$ are stable. Then the $M\Delta$- system is stable for all perturbations satisfying $\|\Delta\|_\infty < 1$ (i.e. RS) if and only if*

$$\|M\|_\infty < 1 \tag{4.3}$$

Hence NS, NP and RS can be stated as

$$\begin{aligned} \text{NS} \quad &\Leftrightarrow \quad N \text{ is internally stable} && (4.4)\\ \text{NP} \quad &\Leftrightarrow \quad \|N_{22}\|_\infty < 1, \quad \text{and NS} && (4.5)\\ \text{RS} \quad &\Leftrightarrow \quad \|N_{11}\|_\infty < 1 \ \ \forall\Delta,\ \|\Delta\|_\infty \leq 1 \quad \text{and NS} && (4.6) \end{aligned}$$

If the system in Figure 4.1 is converted to a PK structure shown in Figure 4.2, the N matrix can be derived as

$$N(s) = \begin{bmatrix} w_O(s)H(s) & w_O(s)H(s) \\ w_P(s)S(s) & w_P(s)S(s) \end{bmatrix} \tag{4.7}$$

Then the nominal performance and robust stability conditions can be defined as

$$\begin{aligned} \text{NP} \quad &\Leftrightarrow \quad \|w_P(s)S(s)\|_\infty < 1, \quad \text{and NS} && (4.8)\\ \text{RS} \quad &\Leftrightarrow \quad \|w_O(s)H(s)\|_\infty < 1 \ \ \forall\Delta,\ \|\Delta\|_\infty \leq 1 \quad \text{and NS} && (4.9) \end{aligned}$$

Let K_{PID} be the set of all possible stabilizing decentralized PID controllers. If only the output z_1 in Figure 4.2 is considered and if the controller K is restricted to be in the set K_{PID}, then (4.2) can be defined as

$$\min_{K \in K_{PID}} \|w_P(s)S(s)\|_\infty \tag{4.10}$$

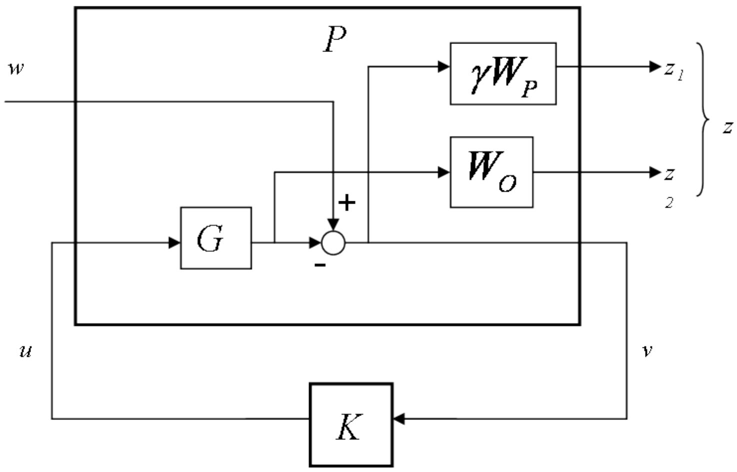

Figure 4.2: General P-K control configuration. K denotes the controller and P denotes the plant.

subject to

$$\|w_O(s)H(s)\|_\infty < 1 \tag{4.11}$$

Note that the robust stability condition is stated as a constraint. The minimization problem in (4.10) analogous to

$$\min_{K \in K_{PID}} -\gamma \tag{4.12}$$

subject to

$$\|w_O(s)H(s)\|_\infty < 1 \tag{4.13}$$
$$\|\gamma w_P(s)S(s)\|_\infty < 1 \tag{4.14}$$

By introducing the minimization variable γ, the best nominal performance related to the weighting w_P will be achieved. The robust stability will never be better than necessary, but the condition must be achieved if the problem is feasible. If $\gamma < 1$ the performance condition related to the weighting w_P can not be achieved. If $\gamma = 1$ the performance is achieved and better performance is achieved if $\gamma > 1$.

To simplify the calculations, $H_{cl}(s)$ is defined as

$$H_{cl}(s) = \begin{bmatrix} W_O(s)H(s) \\ \gamma w_P(s)S(s) \end{bmatrix} \tag{4.15}$$

Then the minimization problem is stated as

$$\min_{K \in K_{PID}} -\gamma \tag{4.16}$$

subject to

$$\|H_{cl}(s)\|_\infty = \left\| \begin{array}{c} w_O(s)H(s) \\ \gamma w_P(s)S(s) \end{array} \right\|_\infty < 1 \tag{4.17}$$

This is a more conservative minimization problem than (4.12), since

$$\|H_{cl}(s)\|_\infty \geq \max\left\{\|w_O(s)H(s)\|_\infty, \|\gamma w_P(s)S(s)\|_\infty\right\} \tag{4.18}$$

If the plant can be represented as (4.7), then

$$\bar{\sigma}\left[\begin{array}{c} w_O(s)H(s) \\ \gamma w_P(s)S(s) \end{array}\right] \leq \mu_{\hat{\Delta}}(N) \leq \sqrt{2}\bar{\sigma}\left[\begin{array}{c} w_O(s)H(s) \\ \gamma w_P(s)S(s) \end{array}\right] \tag{4.19}$$

This relation is stated in [39] at page 327.

In [39] at page 319 the robust performance condition is expressed with the structure singular value as

$$\text{RP} \;\Leftrightarrow\; \mu_{\hat{\Delta}}(N) < 1 \;\; \forall\omega, \hat{\Delta} = \text{diag}\{\Delta, \Delta_P\} \text{ and NS} \tag{4.20}$$

where Δ_P is the perturbation in the feedback as shown in Figure 4.3.

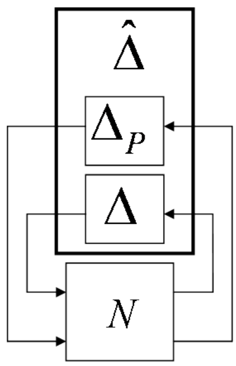

Figure 4.3: A $N\hat{\Delta}$ representation of the system for a robust performance withcondition.

By combining (4.19) and (4.20), robust performance is achieved with respect to the performance weight γw_P if

$$\text{RP} \;\Leftrightarrow\; \|H_{cl}\|_\infty \leq \frac{1}{\sqrt{2}} \text{ and NS} \tag{4.21}$$

Note that robust performance with respect to w_P is only achieved if $\gamma \geq 1$. This is of course a more conservative requirement than (4.20). Hence, if robust performance must be achieved, (4.21) can be used as a condition in (4.16) instead of (4.17). In this thesis only the nominal performance requirement in (4.16) is considered.

4.1.2 Representing the optimization problem with linear matrix inequalities

Control tuning problems can be framed in terms of matrix inequality constraints as shown in [4]. To solve the problem in (4.16), equation (4.17) is converted into a Linear Matrix Inequality by the Bounded Real Lemma (BRL) defined in [2] and in [4] at page 26.

Lemma 1. *Consider a continuous-time transfer function $H(s)$ of (not necessarily minimal) realization $H(s) = D + C(sI - A)^{-1}B$. The following statements are equivalent.*

1. $\left\|D + C(sI - A)^{-1}B\right\|_\infty < \gamma_B$ *and* $H(s)$ *is stable* $Re\{\lambda_i(A)\} < 0$
2. *There exists a real symmetric positive definite solution X to the LMI*

$$\begin{bmatrix} A^T X + XA & XB & C^T \\ B^T & -\gamma_B I & D^T \\ C & D & -\gamma_B I \end{bmatrix} < 0 \tag{4.22}$$

where the matrix inequality represents a partial ordering and indicates negative definiteness ($<$), negative semidefiniteness ($\leq$), positive definiteness ($>$) or positive semidefiniteness ($\geq$).

The minimization problem in (4.16) with matrix inequalities is then

$$\min_{\mathbf{x}} -\gamma \tag{4.23}$$

subject to

$$\begin{bmatrix} A_{cl}^T(\mathbf{x})X(\mathbf{x}) + X(\mathbf{x})A_{cl}(\mathbf{x}) & X(\mathbf{x})B_{cl}(\mathbf{x}) & C_{cl}^T \\ B_{cl}^T(\mathbf{x})X(\mathbf{x}) & -I & D_{cl}^T \\ C_{cl} & D_{cl} & -I \end{bmatrix} < 0 \tag{4.24}$$

$$X = X^T \tag{4.25}$$

$$X > 0 \tag{4.26}$$

where

$$H_{cl}(s) = D_{cl} + C_{cl}\left(sI - A_{cl}\right)^{-1} B_{cl} \tag{4.27}$$

and the notation $(\cdot)(\mathbf{x})$ is used to point out that the matrix depends of the decision variables. The PID control parameters and the elements of the symmetric positive definite matrix X are the decision variables in $\mathbf{x}$. The LMI constraints ensure that the system is stable. The matrices A_{cl}, B_{cl}, C_{cl} and D_{cl} has to be defined with the state space realizations of G, W_O, W_P and K. The relations between the transfer functions and the state-space realization for the given systems are

$$G(s) = D_g + C_g\,(sI - A_g)^{-1}\,B_g \tag{4.28}$$
$$W_O(s) = D_{wo} + C_{wo}\,(sI - A_{wo})^{-1}\,B_{wo} \tag{4.29}$$
$$W_P(s) = D_{wp} + C_{wp}\,(sI - A_{wp})^{-1}\,B_{wp} \tag{4.30}$$
$$K(s) = D_k + C_k\,(sI - A_k)^{-1}\,B_k \tag{4.31}$$

where $W_O = w_O I$ and $W_P = w_P I$. First a state-space representation of the P matrix in Figure 4.2 is derived. If $x(t)$ is the state space vector, the state-space of $P(s)$ is

$$\dot{x} = A\,x + B_1\,w + B_2\,u \tag{4.32}$$
$$z = C_1\,x + D_{11}\,w + D_{12}\,u \tag{4.33}$$
$$v = C_2\,x + D_{12}\,w + D_{22}\,u \tag{4.34}$$

where $z = [z_1\ z_2]^T$. Hence, it is desired to represent

$$z_1(s) = \gamma W_P(s)\,w(s) - \gamma W_P(s) G(s)\,u(s) \tag{4.35}$$
$$z_2(s) = W_O(s) G(s)\,u(s) \tag{4.36}$$
$$v(s) = w(s) - G(s)\,u(s) \tag{4.37}$$

as the Laplace transformation of (4.33) and (4.33)

$$z = (C_1(sI - A)^{-1}B_1 + D_{11})\,w + (C_1(sI - A)^{-1}B_2 + D_{12})\,u \tag{4.38}$$
$$v = (C_2(sI - A)^{-1}B_1 + D_{21})\,w + (C_2(sI - A)^{-1}B_2 + D_{22})\,u \tag{4.39}$$

The equations (4.28), (4.29) and (4.30) are used in (4.35), (4.36) and (4.37).

$$\begin{aligned} z_1(s) = {} & \gamma(D_{wp} + C_{wp}\,(sI - A_{wp})^{-1}\,B_{wp})\,w(s) \\ & -\gamma(D_{wp} + C_{wp}\,(sI - A_{wp})^{-1}\,B_{wp})(D_g + C_g\,(sI - A_g)^{-1}\,B_g))\,u(s) \end{aligned} \tag{4.40}$$

$$z_2(s) = (D_{wo} + C_{wo}\,(sI - A_{wo})^{-1}\,B_{wo})(D_g + C_g\,(sI - A_g)^{-1}\,B_g)\,u(s) \tag{4.41}$$
$$v(s) = w(s) - (D_g + C_g\,(sI - A_g)^{-1}\,B_g)\,u(s) \tag{4.42}$$

These system equations are rearranged so

$$C_1 = \begin{bmatrix} \gamma D_{wp} C_g & \gamma C_p & 0 \\ D_{wo} C_g & 0 & C_{wo} \end{bmatrix} \quad C_2 = \begin{bmatrix} -C_g & 0 & 0 \end{bmatrix} \tag{4.43}$$

$$(sI - A)^{-1} = \begin{bmatrix} (sI - A_g)^{-1} & 0 & 0 \\ (sI - A_{wp})^{-1} B_{wp} C_g (sI - A_g)^{-1} & (sI - A_{wp})^{-1} & 0 \\ (sI - A_{wo})^{-1} B_{wo} C_g (sI - A_g)^{-1} & 0 & (sI - A_{wo})^{-1} \end{bmatrix} \tag{4.44}$$

$$B_1 = \begin{bmatrix} B_g \\ -B_{wp}D_g \\ B_{wo}D_g \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 \\ B_{wp} \\ 0 \end{bmatrix} \tag{4.45}$$

$$D_{11} = \begin{bmatrix} \gamma D_{wp} \\ 0 \end{bmatrix} \quad D_{12} = \begin{bmatrix} -D_{wp}D_g \\ D_{wo}D_g \end{bmatrix} \tag{4.46}$$

$$D_{12} = I \quad D_{22} = -D_g \tag{4.47}$$

To obtain the A matrix form (4.44), a Gauss-Jordan operation is performed at

$$\left[\begin{array}{ccc|ccc} (sI - A_g)^{-1} & 0 & 0 & I & 0 & 0 \\ (sI - A_{wp})^{-1}B_{wp}C_g(sI - A_g)^{-1} & (sI - A_{wp})^{-1} & 0 & 0 & I & 0 \\ (sI - A_{wo})^{-1}B_{wo}C_g(sI - A_g)^{-1} & 0 & (sI - A_{wo})^{-1} & 0 & 0 & I \end{array}\right]$$

then

$$(sI - A) = \begin{bmatrix} (sI - A_g) & 0 & 0 \\ B_{wp}C_g & (sI - A_{wp}) & 0 \\ -B_{wo}C_g & 0 & (sI - A_{wo}) \end{bmatrix} \tag{4.48}$$

Hence

$$A = \begin{bmatrix} A_g & 0 & 0 \\ -B_{wp}C_g & A_{wp} & 0 \\ B_{wo}C_g & 0 & A_{wo} \end{bmatrix} \tag{4.49}$$

It is here assumed that $G(s)$ are strictly proper, and $D_g = 0$. Then $D_{22} = D_{12} = 0$. Then by using (2.43), [2] represents the state-space matrices in (4.27) as

$$A_{cl} = \begin{bmatrix} A + B_2D_kC_2 & B_2C_k \\ B_kC_2 & A_k \end{bmatrix} \tag{4.50}$$

$$B_{cl} = \begin{bmatrix} B_1 + B_2D_kD_{21} \\ -B_2D_{21} \end{bmatrix} \tag{4.51}$$

$$C_{cl} = \begin{bmatrix} C_1 & 0 \end{bmatrix} \quad D_{cl} = D_{11} \tag{4.52}$$

Then A_{cl}, B_{cl}, C_{cl}, and D_{cl} can be represented with the system matrices of G, W_O, W_P and K as

$$A_{cl}(\mathbf{x}) = \begin{bmatrix} A_g - B_gD_k(\mathbf{x})C_g & 0 & 0 & B_gC_k(\mathbf{x}) \\ -B_{wp}C_g & A_{wp} & 0 & 0 \\ B_{wo}C_g & 0 & A_{wo} & 0 \\ -B_kC_g & 0 & 0 & A_k(\mathbf{x}) \end{bmatrix} \tag{4.53}$$

$$B_{cl}(\mathbf{x}) = \begin{bmatrix} B_gD_k(\mathbf{x}) \\ -B_{wp} \\ 0 \\ B_k(\mathbf{x}) \end{bmatrix} \tag{4.54}$$

$$C_{cl} = \begin{bmatrix} \gamma D_{wp} C_g & \gamma C_p & 0 & 0 \\ D_{wo} C_g & 0 & C_{wo} & 0 \end{bmatrix} \quad D_{cl} = \begin{bmatrix} \gamma D_{wp} \\ 0 \end{bmatrix} \tag{4.55}$$

In [2] the element $c_{1\,11} = \gamma D_{wp} D_g$ instead of $\gamma D_{wp} C_g$ as shown in (4.43). The result in [2] is also repeated when C_{cl} is written. When $D_g = 0$ for strictly proper systems, $c_{cl\,11} = 0$.

The controller K can be any controller, but is here restricted to be a PI or a PID. For sate-space representation, the PID controller has to be proper, so (2.21) is used. If k_i is the PID controller in loop i, x is the state vector, e is the controller input and u is the controller output, the state-space of the PID controller is

$$\begin{aligned} \dot{x} &= A_{ki}\, x + B_{ki}\, e \\ u &= C_{ki}\, x + D_{ki}\, e \end{aligned} \tag{4.56}$$

The controller can be represented in the canonical form presented in [45] at page 93 as

$$A_{ki} = \begin{bmatrix} 0 & 1 \\ -\alpha_1 & -\alpha_2 \end{bmatrix} \quad B_{ki} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.57}$$

$$C_{ki} = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \quad D_{ki} = d \tag{4.58}$$

If (2.21) is rewritten to

$$k_i(s) = \frac{k_{ci}\tau_{Ii}\tau_{Di}\, s^2 + k_{ci}\tau_{Ii}\, s + k_{ci}}{s(\tau_{Ii}\epsilon\tau_{Di}\, s + \tau_{Ii})} \tag{4.59}$$

it is seen that one of the two poles in the controller is restricted to be zero. This restrict $\alpha_1 \equiv 0$. Then

$$A_{ki} = \begin{bmatrix} 0 & 1 \\ 0 & -k_{i,1} \end{bmatrix} \quad B_{ki} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.60}$$

$$C_{ki} = \begin{bmatrix} k_{i,2} & k_{i,3} \end{bmatrix} \quad D_{ki} = k_{i,4} \tag{4.61}$$

For a multi-loop PID controller, this gives the following matrices

$$A_k = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -k_{1,1} \end{bmatrix} & 0 & \cdots & 0 \\ 0 & \begin{bmatrix} 0 & 1 \\ 0 & -k_{1,2} \end{bmatrix} & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & \begin{bmatrix} 0 & 1 \\ 0 & -k_{1,N} \end{bmatrix} \end{bmatrix} \tag{4.62}$$

$$B_k = \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} & 0 & \cdots & 0 \\ 0 & \begin{bmatrix} 0 \\ 1 \end{bmatrix} & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} \tag{4.63}$$

$$C_k = \begin{bmatrix} \begin{bmatrix} k_{2,1} & k_{3,1} \end{bmatrix} & 0 & \cdots & 0 \\ 0 & \begin{bmatrix} k_{2,2} & k_{3,2} \end{bmatrix} & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & \begin{bmatrix} k_{2,N} & k_{3,N} \end{bmatrix} \end{bmatrix} \tag{4.64}$$

$$D_k = \begin{bmatrix} k_{4,1} & 0 & \cdots & 0 \\ 0 & k_{4,2} & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & k_{4,N} \end{bmatrix} \tag{4.65}$$

where

$$k_{1,i} = \frac{1}{\epsilon\tau_D} \tag{4.66}$$

$$k_{2,i} = \frac{K_c}{\tau_I\,\epsilon\tau_D} \tag{4.67}$$

$$k_{3,i} = \frac{K_c}{\epsilon\tau_D}\left(1 - \frac{\tau_D}{\epsilon\tau_D}\right) \tag{4.68}$$

$$k_{4,i} = \frac{K_c\,\tau_D}{\epsilon\tau_D} \tag{4.69}$$

If $A_{cl} \in R^{n\times n}$ and the control system has N loops, then the decision variables $\mathbf{x} = x_1 ... x_{4N+n(n+1)/2}$. The relationship between the decision variables $\mathbf{x}$ and the control matrix variables are

$$k_{1,i} = x_i \qquad i = 1...N \tag{4.70}$$

$$k_{2,i} = x_{i+N} \qquad i = 1...N \tag{4.71}$$

$$k_{3,i} = x_{i+2N} \qquad i = 1...N \tag{4.72}$$

$$k_{4,i} = x_{i+3N} \qquad i = 1...N \tag{4.73}$$

$$\tag{4.74}$$

Then the PID controller parameters can be deriven from (4.66)-(4.69). The last $n(n+1)/2$

variables relates to the symmetric auxiliary matrix variable X as

$$X = \begin{bmatrix} x_{4N+1} & x_{4N+2} & x_{4N+3} & \cdots & x_{4N+n-1} & x_{4N+n} \\ x_{4N+2} & x_{4N+n+1} & x_{4N+n+2} & \cdots & x_{4N+2n-3} & x_{4N+2n-1} \\ x_{4N+3} & x_{4N+n+2} & x_{4N+2n} & \cdots & x_{4N+3n-2} & x_{4N+3n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{4N+n-1} & x_{4N+2n-3} & x_{4N+2n-2} & \cdots & x_{4N+n(n+1)/2-2} & x_{4N+n(n+1)/2-1} \\ x_{4N+n} & x_{4N+2n-1} & x_{4N+3n-3} & \cdots & x_{4N+n(n+1)/2-1} & x_{4N+n(n+1)/2} \end{bmatrix} \tag{4.75}$$

4.1.3 Successive Semidefinite Programing

The problem in (4.23) is non-convex, because $\mathbf{x}$ forms bilinear terms in (4.24), shown in [23]. Hence conventional semidefinite programming approaches in e.g. [25] cannot be used. A Successive Semidefinite Programing approach is presented in [2] to overcome these problems.

If $A_{cl}^T(\mathbf{x})X(\mathbf{x})$ forms bilinear terms, and $A_{cl}^T(\mathbf{x}) = A_{cl\,0}^T$ and $X(\mathbf{x}) = X_0$ are feasible solutions to (4.23), then $A_{cl}^T(\mathbf{x})X(\mathbf{x})$ is approximated as

$$\begin{aligned} A_{cl}^T(\mathbf{x})X(\mathbf{x}) &= A_{cl\,0}^T X_0 + A_{cl\,0}^T \delta X(\mathbf{x}) + \delta A_{cl}^T(\mathbf{x}) X_0 + \delta A_{cl}^T(\mathbf{x}) \delta X(\mathbf{x}) \\ &\approx A_{cl\,0}^T X_0 + A_{cl\,0}^T \delta X(\mathbf{x}) + \delta A_{cl}^T(\mathbf{x}) X_0 \end{aligned} \tag{4.76}$$

where $\delta A_{cl}^T(\mathbf{x}) = A_{cl}^T(\mathbf{x}) - A_{cl\,0}^T$, $\delta X(\mathbf{x}) = X(\mathbf{x}) - X_0$ and

$$\left\|\delta A_{cl}^T\right\| \leq \epsilon \quad \|\delta X\| \leq \epsilon \tag{4.77}$$

where ϵ is an arbitrary small positive number such that the approximated solution of (4.76) holds. Then (4.23)

$$\min_{\mathbf{x}} -\gamma \tag{4.78}$$

subject to

$$\begin{bmatrix} \begin{array}{c} A_{cl\,0}^T X_0 + A_{cl\,0}^T \delta X + \delta A_{cl}^T X_0 + \\ X_0 A_{cl\,0} \delta X A_{cl\,0} + X_0 \delta A_{cl} \end{array} & X_0 B_{cl\,0} + X_0 \delta B_{cl} + \delta X B_{cl\,0} & C_{cl\,0}^T + \delta C_{cl}^T \\ B_{cl\,0} X_0 + \delta B_{cl} X_0 + B_{cl\,0} \delta X & -I & \delta D_{cl}^T + D_{cl\,0}^T \\ \delta C_{cl} + C_{cl\,0} & \delta D_{cl} + D_{cl\,0} & -I \end{bmatrix} < 0 \tag{4.79}$$

$$\delta X = \delta X^T \tag{4.80}$$

$$X_0 + \delta X > 0 \tag{4.81}$$

where δA_k, δB_k, δD_k and δX are functions of the decision variables $\mathbf{x}$. The optimization problem (4.78) is convex, and standard semidefinite programing techniques given in

[25] can be used. It is shown in Appendix B how to represent (4.79) with the system and control matrices. Note that (B.8) differ from the result obtained in [2] due to the difference in the element $c_{1\,11}$ in (4.43).

Then the Successive Semidefinite Programing procedure in [2] can be used to find a local minimum to (4.23) as follows:

1. Choose initial values of $A_{k\,0}$, $C_{k\,0}$, and $D_{k\,0}$. Fix $A_k = A_{k\,0}$, $C_k = C_{k\,0}$, and $D_k = D_{k\,0}$. Solve the resulting LMI problem (4.23) and use the solution $(\tilde{X}, \tilde{\gamma})$ as initial value $X_0 = \tilde{X}$ and $\gamma_0 = \tilde{\gamma}$. Set the initial solution radius as ϵ_0, the maximum number of iterations as $n_{\max}$, convergence tolerance as ζ, and preset iteration counter $k = 0$.

2. Solve (4.78) with the initial values $A_{k\,0}$, $C_{k\,0}$, $D_{k\,0}$, X_0 and γ_0. Assume at kth iteration, the solutions δA_k, δC_k, δD_k, δX and γ^k are obtained with their radii restricted by

$$\|\delta A_k\| \le \epsilon^k \quad \|\delta C_k\| \le \epsilon^k \quad \|\delta D_k\| \le \epsilon^k \quad \|\delta X\| \le \epsilon^k \tag{4.82}$$

3. Compute $\tilde{A}_k = A_{k\,0} + \delta A_k$, $\tilde{C}_k = C_{k\,0} + \delta C_k$, $\tilde{D}_k = D_{k\,0} + \delta D_k$, $\tilde{X} = X_0 + \delta X$.
   (a) If $\tilde{A}_k$, $\tilde{C}_k$, $\tilde{D}_k$ and $\tilde{X}$ are feasible solutions to (4.23)
      i. If $\left\|\gamma^k - \gamma^{k-1}\right\| \le \zeta$, then acceptable solution is obtained. Proceed to 4.
      ii. If $\left\|\gamma^k - \gamma^{k-1}\right\| > \zeta$, then $\tilde{A}_k$, $\tilde{C}_k$, $\tilde{D}_k$ and $\tilde{X}$ are used as the reference values for the next approximated optimization problem. Let $A_{k\,0} = \tilde{A}_k$, $C_{k\,0} = \tilde{C}_k$, $D_{k\,0} = \tilde{D}_k$, $X_0 = \tilde{X}$, and $\epsilon^{k+1} = \alpha\epsilon^k$ (where α is user-specified tuning parameter, typically $\alpha = 0.95$), and go to step 2.
   (b) If $\tilde{A}_k$, $\tilde{C}_k$, $\tilde{D}_k$ and $\tilde{X}$ are not feasible solutions to (4.23),
      i. If $k > n$, then (4.23) is infeasible in the neighborhood of ($A_{k\,0}$, $C_{k\,0}$, $D_{k\,0}$, X_0) and stop.
      ii. If $k \le n$, then reject these solutions and choose $\epsilon^{k+1} = \beta\epsilon^k$ (where β is another user-specified tuning parameter, typically $\beta = 1.05$) and go to step 2.

4. The state-space representation of the multi-loop PID controller is obtained as $K(s) = \tilde{C}_k(sI - \tilde{A}_k)^{-1}\tilde{B}_k + \tilde{D}_k$. Use (4.66)-(4.69) to obtain the controller parameters.

This tuning method works only for time continuous systems, and any time delays have to be approximated by a Padé approximation shown in (2.30).

4.1.4 Discussion of the H_∞ optimal tuning method

The optimization problem in (4.23) is not convex. Hence the purposed algorithm only guarantees to find a local optimum. Further analysis in the literature must be done to extend the algorithm to guarantee a global optimum. A brute force solution is to compare several local optimums when different initial values is applied in Step 1.

The relations in (4.50), (4.51) and (4.52) are neither proved here nor in [2]. But the reader is encouraged to prove these equations by using (2.43) and the matrix inversion lemma shown in [24] page 364.

To verify the algebraic results, the distillation process in [2] was considered. The P matrix was found in MATLAB using `sysic`. Then the function `lft` was performed at the controller K and P. The transfer functions obtained where compared with the results obtained using (4.50), (4.51) and (4.52) for the distillation process with a controller. Some of the transfer functions calculated with `sysic` and `lft` had one extra order in the numerator with a coefficient of 10^{-11} or less. This difference is neglectable and may be the result of some numerical operations in MATLAB. Hence, this indicates that (4.50), (4.51) and (4.52) are correct.

When the plant P was obtained in MATLAB using `sysic`, the same result as the calculations in this thesis was obtained. This indicates that the results in (4.43) is correct. It is not known if $c_{1\,11} = \gamma D_{wp} D_g$ is used in the simulation results presented in [2].

At algorithm step one in Section 4.1.3, the control matrices are fixed, so the constraints form no bilinear terms. Then (4.23) is convex, and starting values for X and γ can be found.

4.2 H_∞ optimal multi-loop tuning with failure tolerance

The tuning method presented in Section 4.1, may be used in the following way to guarantee integrity. Remember the mathematically description of integrity in Section 2.3, where the controller is replaced by the matrix $E\,K(s)$. Consider for simplicity a multi-loop TITO system. Then the matrix E can take four forms.

1. $E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ Both controllers are in function. (4.83)
2. $E = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ Only control loop 1 is in function. (4.84)
3. $E = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ Only control loop 2 is in function. (4.85)
4. $E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ None of the controllers are in function. (4.86)

If the system possess the integrity property, the system must be stable for all the above cases. Hence, case four requires that the system must be open stable, and is completely independent of the controller. Consequently, only the three first cases are considered in the control design, shown in Figure 4.4:

The weights W_{O1}, W_{O2}, W_{P1} and W_{P2} are user specified weights when one of the loops are turned off or fail. Due to system interactions, it is reason to believe that the performance requirement must be poorer when one of the loops fail. Then three H_∞

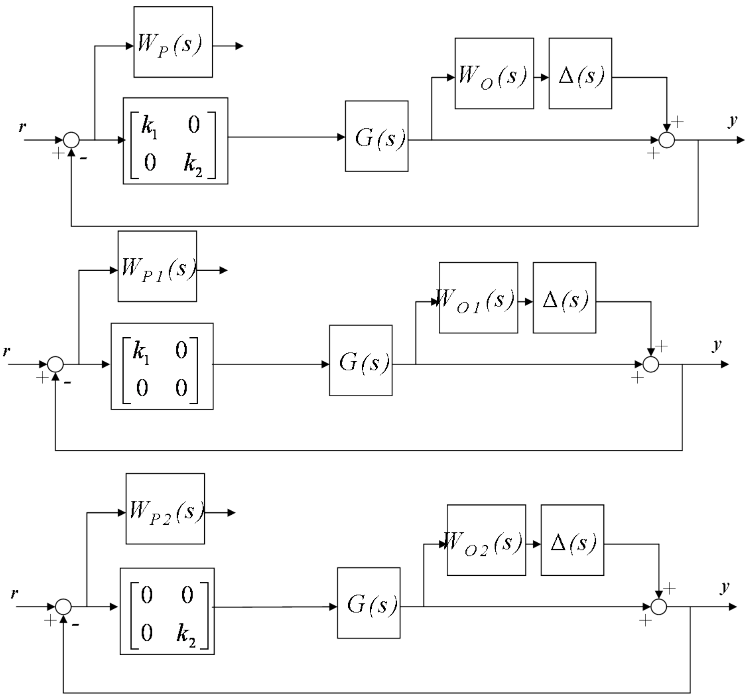

Figure 4.4: The combinations of controllers which are in or out of function.

optimization problems can be defined for each scenario. First where both the controllers are running normally

$$\min_{K \in K_{PID}} -\gamma \tag{4.87}$$

subject to

$$\|H_{cl}(s)\|_\infty = \left\| \begin{array}{c} w_O(s)H(s) \\ \gamma w_P(s)S(s) \end{array} \right\|_\infty < 1 \tag{4.88}$$

then when loop two is turned off

$$\min_{K \in K_{PID}} -\gamma \tag{4.89}$$

subject to

$$\|H_{cl\,1}(s)\|_\infty = \left\| \begin{array}{c} w_{O1}(s)H_1(s) \\ \gamma w_{P1}(s)S_1(s) \end{array} \right\|_\infty < 1 \tag{4.90}$$

and finally when loop one is turned off.

$$\min_{K \in K_{PID}} -\gamma \tag{4.91}$$

subject to

$$\|H_{cl\,2}(s)\|_\infty = \left\| \begin{matrix} w_{O2}(s)H_2(s) \\ \gamma w_{P2}(s)S_2(s) \end{matrix} \right\|_\infty < 1 \tag{4.92}$$

Suppose a solution to (4.87), (4.89) and (4.91) exists, then the nominal performance and robust stability for the chosen weights are satisfied when loop 2 and loop 1 fails respectively. If some control parameters exists such that all the conditions in (4.88), (4.90) and (4.92) are fulfilled at the same time, then nominal performance and robust stability requirement are obtained for all the three scenarios. Hence, the system has integrity, if $G(s)$ is stable.

A brute force solution to find these parameters is to stack requirement (4.87) (4.89) and (4.91) into a matrix.

$$\tilde{H}_{cl} = \begin{bmatrix} H_{cl} \\ H_{cl\,1} \\ H_{cl\,2} \end{bmatrix} = \begin{bmatrix} w_O(s)H_(s) \\ w_{O1}(s)H_1(s) \\ w_{O2}(s)H_2(s) \\ \gamma w_P(s)S(s) \\ \gamma w_{P1}(s)S_1(s) \\ \gamma w_{P2}(s)S_2(s) \end{bmatrix} \tag{4.93}$$

The optimization problem is then

$$\min_{K_{tot} \in K_{tot\,PID}} -\gamma \tag{4.94}$$

subject to

$$\left\| \tilde{H}_{cl}(s) \right\|_\infty < 1 \tag{4.95}$$

the optimization in (4.94) gives more conservative settings than (4.87), (4.89) and (4.91) since

$$\left\| \tilde{H}_{cl}(s) \right\|_\infty \geq \max \left\{ \|H_{cl}(s)\|_\infty, \|H_{cl\,1}(s)\|_\infty, \|H_{cl\,2}(s)\|_\infty \right\} \tag{4.96}$$

The three scenarios are lumped into a generalized $P(s)$ matrix shown in Figure 4.5. As seen from Figure 4.5 the system matrices can be defined as

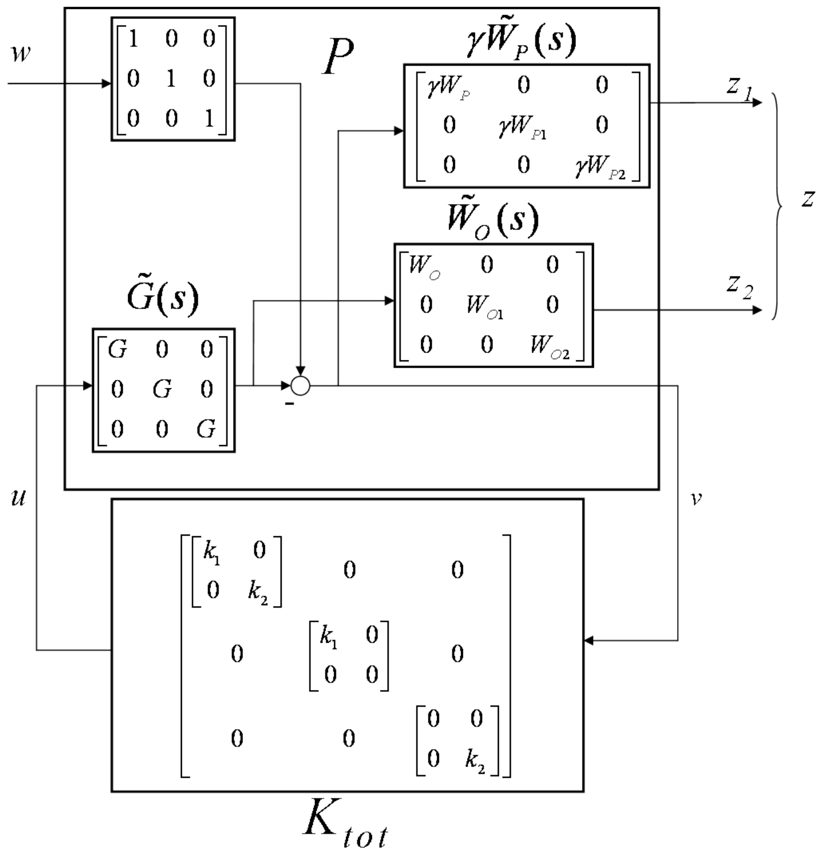

Figure 4.5: A PK representation of the system when both the controllers are turned on, and when each controller in the two loops are turned off one at the time.

$$\tilde{G} = \begin{bmatrix} G & 0 & 0 \\ 0 & G & 0 \\ 0 & 0 & G \end{bmatrix} \tag{4.97}$$

$$\tilde{W}_O = \begin{bmatrix} W_O & 0 & 0 \\ 0 & W_{O1} & 0 \\ 0 & 0 & W_{O2} \end{bmatrix} \tag{4.98}$$

$$\tilde{W}_P = \begin{bmatrix} W_P & 0 & 0 \\ 0 & W_{P1} & 0 \\ 0 & 0 & W_{P2} \end{bmatrix} \tag{4.99}$$

$$K_{tot} = \begin{bmatrix} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} & 0 & 0 \\ 0 & \begin{bmatrix} k_1 & 0 \\ 0 & 0 \end{bmatrix} & 0 \\ 0 & 0 & \begin{bmatrix} 0 & 0 \\ 0 & k_2 \end{bmatrix} \end{bmatrix} \tag{4.100}$$

Since $\tilde{G}$, $\tilde{K_{tot}}$, $\tilde{W}_O$ and $\tilde{W}_P$ are block diagonal, the weighted sensitivity function $\tilde{S}$ and the weighted complementary sensitivity function $\tilde{H}$ for the lumped system is

$$\tilde{W}_P\tilde{S} = \tilde{W}_P(1+\tilde{G}K_{tot})^{-1} = \begin{bmatrix} W_P S & 0 & 0 \\ 0 & W_{P\,1}S_1 & 0 \\ 0 & 0 & W_{P\,2}S_2 \end{bmatrix} \tag{4.101}$$

$$\tilde{W}_O\tilde{H} = \tilde{W}_O\tilde{G}K_{tot}(1+\tilde{G}K_{tot})^{-1} = \begin{bmatrix} W_O H & 0 & 0 \\ 0 & W_{O\,1}H_1 & 0 \\ 0 & 0 & W_{O\,2}H_2 \end{bmatrix} \tag{4.102}$$

Since (4.101) and (4.102) also are block diagonal, it can be shown by comparing the infinity norm of a vector and a matrix defined in (A.3) and (A.4) that

$$\tilde{H}_{cl} = \left\| \begin{matrix} \tilde{W}_O\tilde{H} \\ \gamma\tilde{W}_P\tilde{S} \end{matrix} \right\|_\infty = \left\| \begin{matrix} w_O(s)H_(s) \\ w_{O1}(s)H_1(s) \\ w_{O2}(s)H_2(s) \\ \gamma w_P(s)S(s) \\ \gamma w_{P1}(s)S_1(s) \\ \gamma w_{P2}(s)S_2(s) \end{matrix} \right\|_\infty \tag{4.103}$$

Hence, the BRL can be used at $\tilde{H}_{cl}$ in the same way as described in Section 4.1.2 to represent the constraints in (4.95) as a LMI. Since the system matrices Since $\tilde{G}$, K_{tot}, $\tilde{W}_O$ and $\tilde{W}_P$ are block diagonal, it is easy to obtain their state-space representation.

$$A_{\tilde{g}} = \begin{bmatrix} A_g & 0 & 0 \\ 0 & A_g & 0 \\ 0 & 0 & A_g \end{bmatrix} \quad C_{\tilde{g}} = \begin{bmatrix} C_g & 0 & 0 \\ 0 & C_g & 0 \\ 0 & 0 & C_g \end{bmatrix} \tag{4.104}$$

$$B_{\tilde{g}} = \begin{bmatrix} B_g \\ B_g \\ B_g \end{bmatrix} \quad D_{\tilde{g}} = \begin{bmatrix} D_g \\ D_g \\ D_g \end{bmatrix} \tag{4.105}$$

$$A_{\tilde{w}_O} = \begin{bmatrix} A_{w_O} & 0 & 0 \\ 0 & A_{w_O\,1} & 0 \\ 0 & 0 & A_{w_O\,2} \end{bmatrix} \quad C_{\tilde{w}_O} = \begin{bmatrix} C_{w_O} & 0 & 0 \\ 0 & C_{w_O\,1} & 0 \\ 0 & 0 & C_{w_O\,2} \end{bmatrix} \tag{4.106}$$

$$B_{\tilde{w}_O} = \begin{bmatrix} B_{w_O} \\ B_{w_O\,1} \\ B_{w_O\,2} \end{bmatrix} \quad D_{\tilde{w}_O} = \begin{bmatrix} D_{w_O} \\ D_{w_O\,1} \\ D_{w_O\,2} \end{bmatrix} \tag{4.107}$$

$$A_{\tilde{w}_P} = \begin{bmatrix} A_{w_P} & 0 & 0 \\ 0 & A_{w_P\,1} & 0 \\ 0 & 0 & A_{w_P\,2} \end{bmatrix} \quad C_{\tilde{w}_P} = \begin{bmatrix} C_{w_P} & 0 & 0 \\ 0 & C_{w_P\,1} & 0 \\ 0 & 0 & C_{w_P\,2} \end{bmatrix} \tag{4.108}$$

$$B_{\tilde{w}_P} = \begin{bmatrix} B_{w_P} \\ B_{w_P\,1} \\ B_{w_P\,2} \end{bmatrix} \quad D_{\tilde{w}_P} = \begin{bmatrix} D_{w_P} \\ D_{w_P\,1} \\ D_{w_P\,2} \end{bmatrix} \tag{4.109}$$

The representation of the controller K_{tot} is also quite straight forward.

$$A_{k\,tot} = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -k_{1,1} \end{bmatrix} & 0 & 0 & 0 & 0 & 0 \\ 0 & \begin{bmatrix} 0 & 1 \\ 0 & -k_{1,2} \end{bmatrix} & 0 & 0 & 0 & 0 \\ 0 & 0 & \begin{bmatrix} 0 & 1 \\ 0 & -k_{1,1} \end{bmatrix} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \begin{bmatrix} 0 & 1 \\ 0 & -k_{1,2} \end{bmatrix} \end{bmatrix} \tag{4.110}$$

$$B_{k\,tot} = \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} & 0 & 0 & 0 & 0 & 0 \\ 0 & \begin{bmatrix} 0 \\ 1 \end{bmatrix} & 0 & 0 & 0 & 0 \\ 0 & 0 & \begin{bmatrix} 0 \\ 1 \end{bmatrix} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} \tag{4.111}$$

$$C_{k\,tot} = \begin{bmatrix} \begin{bmatrix} k_{2,1} & k_{3,1} \end{bmatrix} & 0 & 0 & 0 & 0 & 0 \\ 0 & \begin{bmatrix} k_{2,2} & k_{3,2} \end{bmatrix} & 0 & 0 & 0 & 0 \\ 0 & 0 & \begin{bmatrix} k_{2,1} & k_{3,1} \end{bmatrix} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \begin{bmatrix} k_{2,2} & k_{3,2} \end{bmatrix} \end{bmatrix} \tag{4.112}$$

$$D_{k\,tot} = \begin{bmatrix} k_{4,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{4,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{4,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{4,2} \end{bmatrix} \tag{4.113}$$

If $\tilde{H}_{cl} = \tilde{C}_{cl}(Is - \tilde{A}_{cl})^{-1}B_{cl} + D_{cl}$, then the optimization can be stated as the same way as (4.23)

$$\min_{\mathbf{x}} -\gamma \tag{4.114}$$

subject to

$$\begin{bmatrix} \tilde{A}_{cl}^T(\mathbf{x})X(\mathbf{x}) + X(\mathbf{x})\tilde{A}_{cl}(\mathbf{x}) & X(\mathbf{x})\tilde{B}_{cl}(\mathbf{x}) & \tilde{C}_{cl}^T \\ \tilde{B}_{cl}^T(\mathbf{x})X(\mathbf{x}) & -I & \tilde{D}_{cl}^T \\ \tilde{C}_{cl} & \tilde{D}_{cl} & -I \end{bmatrix} < 0 \tag{4.115}$$

$$X = X^T \tag{4.116}$$

$$X > 0 \tag{4.117}$$

where

$$\tilde{A}_{cl}(\mathbf{x}) = \begin{bmatrix} A_{\tilde{g}} - B_{\tilde{g}}D_{k\,tot}(\mathbf{x})C_{\tilde{g}} & 0 & 0 & B_{\tilde{g}}C_{k\,tot}(\mathbf{x}) \\ -B_{\tilde{w}p}C_{\tilde{g}} & A_{\tilde{w}p} & 0 & 0 \\ B_{\tilde{w}o}C_{\tilde{g}} & 0 & A_{\tilde{w}o} & 0 \\ -B_{k\,tot}C_{\tilde{g}} & 0 & 0 & A_{k\,tot}(\mathbf{x}) \end{bmatrix} \tag{4.118}$$

$$\tilde{B}_{cl}(\mathbf{x}) = \begin{bmatrix} B_{\tilde{g}}D_{k\,tot}(\mathbf{x}) \\ -B_{\tilde{w}p} \\ 0 \\ B_{k\,tot}(\mathbf{x}) \end{bmatrix} \tag{4.119}$$

$$\tilde{C}_{cl} = \begin{bmatrix} \gamma D_{\tilde{w}p}C_{\tilde{g}} & \gamma C_p & 0 & 0 \\ D_{\tilde{w}o}C_{\tilde{g}} & 0 & C_{\tilde{w}o} & 0 \end{bmatrix} \quad \tilde{D}_{cl} = \begin{bmatrix} \gamma D_{\tilde{w}p} \\ 0 \end{bmatrix} \tag{4.120}$$

Then the Successive Semidefinite Programming procedure presented in Section 4.1.3 can be used to derive a local solution to (4.114).

The same technique can be used to systems with more loops. Then all the combinations of the different controllers must be placed at the block diagonal of K_{tot}. For N loops, the number of combinations C becomes

$$C = 1 + \frac{N!}{(N-1)!1!} + \frac{N!}{(N-2)!2!} + \frac{N!}{(N-3)!3!} + ... + \frac{N!}{2!(N-2)!} + \frac{N!}{1!(N-1)!} \tag{4.121}$$

This can be written as

$$C = \begin{cases} 1 + 2\left(\frac{N!}{(N-1)!1!} + ... + \frac{N!}{(N-k)!k!}\right) & k = \frac{N-1}{2}, \text{ for odd } N \\ 1 + 2\left(\frac{N!}{(N-1)!1!} + ... + \frac{N!}{(N-k)!k!}\right) + \frac{N!}{(N-\frac{N}{2})!\frac{N}{2}!} & k = \frac{N-2}{2}, \text{ for even } N \end{cases} \tag{4.122}$$

This inflicts the size of X but not the number of PID parameters. Nevertheless, the number decision variables will increase rapidly with C.

If a solution to (4.114) exists and $G(s)$ is stable, then this solution guarantees integrity but not DIC and RDD.

When a solution to (4.23) exists, and no solution to (4.114) can be found, it indicates that the stability weight is too tight, or that integrity with integral control is not possible.

If the latter is the case, then some loops must be intact for the system to be stable. To find these loops, the purposed procedure in this section may be used. Consider again a TITO system. A solution to (4.23) is found, but no solution to (4.114). To determine if one of the loops may be brought out of service, then the following optimization problems may be compared. First, derive the optimization problem where both the controllers and loop 1 are in service. Then, derive the optimization problem where both the controllers and loop 2 are in service. If a solution to the first problem is found, and no solution to the second one is found, then loop 1 is critical for the system functionality. The performance weights for such tests should be as small as possible. Such comparison can atomized with a computer program. The only inputs to the program is the plant model, the different weights and the SSP parameters presented in Section 4.1.3.

The purposed tuning method should be compared to the Independent method and the regular H_∞ method in [2]. The strengths of the purposed method are best exemplified if

1. the purposed method gives better performance than by the Independent design, and integrity is simultaneously satisfied.
2. the purposed tuning method gives integrity to a plant which is impossible to handle with the Independent design.
3. integrity or failure tolerance is not achieved with the regular multi-loop H_∞ method in Section 4.1, while integrity or failure tolerance is obtained with the purposed method.

To compare the methods, the optimization algorithm in Section 4.1.3 must be implemented in a simulation program, such as MATLAB.

Chapter 5

Implementation of the H_∞ optimization

5.1 SSP algorithm using MATLAB functions

The multi-loop H_∞ method with integrity in Section 4.2 is based at the multi-loop method presented in Section 4.1. Hence, the SSP algorithm in Section 4.1.3 must first be implemented at a computer to compare the method in Section 4.2 with other methods. For this use, MATLAB R2007b is used. The *Robust Toolbox* in this MATLAB version offers functions that solves convex LMI problems. Some modifications of the SSP algorithm has been done so the predefined LMI-functions can be used.

5.1.1 How to find the optimal value of γ

In [2] it is not explained in detail how to find the optimum value of γ. The function `mincx` solves the problem

$$\min c^T \mathbf{x} \tag{5.1}$$

subject to the LMI's

$$N^T L(\mathbf{x}) N \leq M^T R(\mathbf{x}) M \tag{5.2}$$

where $\mathbf{x}$ a the vector of scalar decision variables. To use this function to solve (4.23), γ is included in the vector of the decision variables.

$$\mathbf{x} = \begin{bmatrix} \gamma & x_1 & x_2 & ... & x_{4N+n(n+1)/2} \end{bmatrix} \tag{5.3}$$

where $x_1 ... x_{4N+n(n+1)/2}$ is defined in (4.70) to (4.75). If the matrix $N = I$, $M(\mathbf{x}) = 0$, $c = \begin{bmatrix} -1 & 0 & 0 & ... & 0 \end{bmatrix}^T$ and

$$L(\mathbf{x}) = \begin{bmatrix} A_{cl}^T(\mathbf{x})X(\mathbf{x}) + X(\mathbf{x})A_{cl}(\mathbf{x}) & X(\mathbf{x})B_{cl}(\mathbf{x}) & C_{cl}^T \\ B_{cl}^T(\mathbf{x})X(\mathbf{x}) & -I & D_{cl}^T \\ C_{cl} & D_{cl} & -I \end{bmatrix}$$

then the optimization problem (4.23) can be stated as

$$\min_{\mathbf{x}} c^T \mathbf{x} \tag{5.4}$$

subject to

$$\begin{bmatrix} A_{cl}^T(\mathbf{x})X(\mathbf{x}) + X(\mathbf{x})A_{cl}(\mathbf{x}) & X(\mathbf{x})B_{cl}(\mathbf{x}) & C_{cl}^T(\mathbf{x}) \\ B_{cl}^T(\mathbf{x})X(\mathbf{x}) & -I & D_{cl}^T(\mathbf{x}) \\ C_{cl}(\mathbf{x}) & D_{cl}(\mathbf{x}) & -I \end{bmatrix} < 0 \tag{5.5}$$

$$X = X^T \tag{5.6}$$

$$X > 0 \tag{5.7}$$

Note that this redefinition forms no new bilinear terms. Hence, (5.4) can be used in Step 1 in the SSP procedure.

Alternately, an iterative line search algorithm, in e.g. [19], could be made using `feasp`, which determines if the LMI problem is feasible and returns a value of the decision variables in the feasible area.

5.1.2 Redefinition of the magnitude of the step size ϵ

The solution to (5.9) is bounded by (4.82). The solution to `mincx` can only be bounded by the euclidean norm of the vector $\mathbf{x}$. But it is not desired to have a bound at the optimal value of γ. To bound the magnitude of δA_k, δB_k, δD_k and δX each decision variable is bounded by

$$\begin{matrix} x_i \leq \epsilon^k \\ x_i \geq -\epsilon^k \end{matrix} \quad i = 1, 2..., 4N + n(n+1)/2 \tag{5.8}$$

where N is the number of loops, and n is the row and column size of A_{cl}. The bound ϵ^k in (5.8) is not the same bound as in (4.82). The approximated semidefinite programming problem (4.78), implemented in algorithm Step 2 is

$$\min_{\mathbf{x}} c^T \mathbf{x} \tag{5.9}$$

subject to

$$\begin{bmatrix} \begin{matrix} A_{cl\,0}^T X_0 + A_{cl\,0}^T \delta X + \delta A_{cl}^T X_0 + \\ X_0 A_{cl\,0} \delta X A_{cl\,0} + X_0 \delta A_{cl} \end{matrix} & X_0 B_{cl\,0} + X_0 \delta B_{cl} + \delta X B_{cl\,0} & C_{cl\,0}^T + \delta C_{cl}^T \\ B_{cl\,0} X_0 + \delta B_{cl} X_0 + B_{cl\,0} \delta X & -I & \delta D_{cl}^T + D_{cl\,0}^T \\ \delta C_{cl} + C_{cl\,0} & \delta D_{cl} + D_{cl\,0} & -I \end{bmatrix} < 0 \tag{5.10}$$

$$\delta X = \delta X^T \tag{5.11}$$

$$X_0 + \delta X > 0 \tag{5.12}$$

$$\begin{matrix} x_i \leq \epsilon^k \\ x_i \geq -\epsilon^k \end{matrix} \quad i = 1, 2..., 4N + n(n+1)/2 \tag{5.13}$$

where δC_{cl} and δD_{cl} is functions of γ.

The function `feasp` returns a value $t_{\min} \leq 0$ if the problem is feasible, $t_{\min} > 0$ if the problem is infeasible. If the problem is strictly feasible, then $t_{\min} < 0$. If the problem is feasible but not strictly feasible, $t_{\min}$ is positive and very small. Some post-analysis may then be required to decide whether the solution is close enough to feasible. For simplicity, the solution is treated as infeasible if it is not strictly feasible in algorithm Step 3. This may lead to a bit more conservative parameters, or that the optimization problem (5.4) is infeasible even if a solution exists.

5.1.3 Detailed explanation of the SSP algorithm with MATLAB functions

The implemented SSP algorithm is as follows:

1. Choose initial values of A_{k0}, C_{k0}, and D_{k0}. Fix $A_k = A_{k0}$, $C_k = C_{k0}$, and $D_k = D_{k0}$. Solve the resulting LMI problem (5.4) by using `mincx`, and use the solution $(\tilde{X}, \tilde{\gamma})$ as initial value $X_0 = \tilde{X}$ and $\gamma_0 = \tilde{\gamma}$. Set the initial solution radius as ϵ_0, the maximum number of iterations as $n_{\max}$, convergence tolerance as ζ, the iteration counter $k = 0$, and the update coefficients to the solution radius $\alpha = 0.95$ and $\beta = 1.05$.

2. Calculate the LMI (5.10), using the matrices in Appendix B. Solve (5.9) by using `mincx` with the initial values A_{k0}, C_{k0}, D_{k0}, X_0 and γ_0.

    (a) if the solutions δA_k, δC_k, δD_k, δX and γ^k are obtained at k'th iteration, proceed to Step 3.

    (b) if (5.9) is infeasible. Exit the program.

3. Compute $\tilde{A}_k = A_{k0} + \delta A_k$, $\tilde{C}_k = C_{k0} + \delta C_k$, $\tilde{D}_k = D_{k0} + \delta D_k$, $\tilde{X} = X_0 + \delta X$. Insert the updated values into (5.5) and check feasibility, using `feasp`.[1]

    (a) If $\tilde{A}_k$, $\tilde{C}_k$, $\tilde{D}_k$ and $\tilde{X}$ are strictly feasible solutions

        i. If $\|\gamma^k - \gamma^{k-1}\| \leq \zeta$, then acceptable solution is obtained. Proceed to 4.

        ii. If $\|\gamma^k - \gamma^{k-1}\| > \zeta$, then $\tilde{A}_k$, $\tilde{C}_k$, $\tilde{D}_k$ and $\tilde{X}$ are used as the reference values for the next approximated optimization problem. Let $A_{k0} = \tilde{A}_k$, $C_{k0} = \tilde{C}_k$, $D_{k0} = \tilde{D}_k$, $X_0 = \tilde{X}$, $\epsilon^{k+1} = \alpha\epsilon^k$ and $k = k + 1$.

[1] The only variable in this problem is γ. `feasp` returns a value of γ that gives feasibility, but this value is not used further since it is not an optimal value.

A. if $k > n_{\max}$, the convergence tolerance ζ or the maximum iterations $n_{\max}$ must be increased.

B. if $k \leq n_{\max}$, go to step 2.

(b) If $\tilde{A}_k$, $\tilde{C}_k$, $\tilde{D}_k$ and $\tilde{X}$ are not feasible solutions to (5.4),

i. If $k > n$, then (5.4) is infeasible in the neighborhood of ($A_{k\,0}$, $C_{k\,0}$, $D_{k\,0}$, X_0). Exit the program and choose different initial parameters.

ii. If $k \leq n$, then reject these solutions and choose $\epsilon^{k+1} = \beta\epsilon^k$, update $k = k + 1$ and go to step 2.

4. The state-space representation of the multi-loop PID controller is obtained as $K(s) = \tilde{C}_k(sI - \tilde{A}_k)^{-1}\tilde{B}_k + \tilde{D}_k$. Use (4.66)-(4.69) to obtain the controller parameters.

The implemented SSP algorithm is found in the MATLAB script `SSPalgoritm` found at the attached software.

5.2 SSP algorithm with a case study

In [2], it is presented a case study to illustrate the multi-loop H_∞ optimization tuning method. It is tried to recreate the results obtained. The system and its output uncertainty and performance weights are here presented.

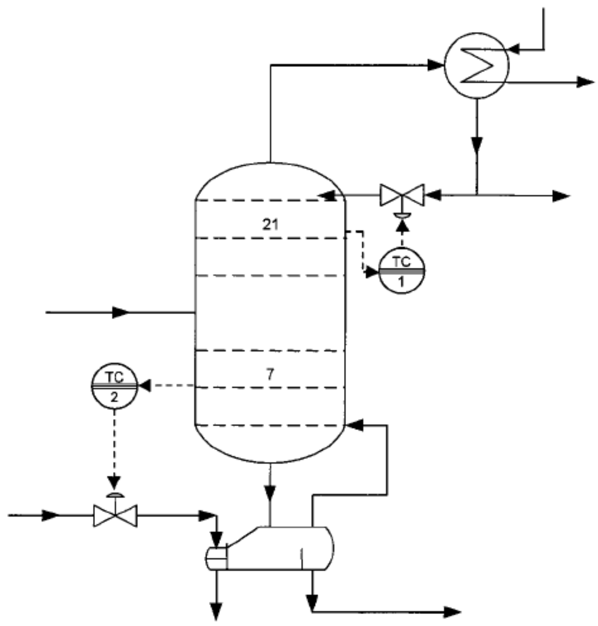

Figure 5.1: Distillation column used as a case study for the multi-loop H_∞ tuning.

Consider the distillation column shown in Figure 5.1. Tray temperatures are regulated as inferential variables for composition, by manipulating the reflux and boilup rates. The transfer function is given as follows in [2]

$$G_c(s) = \begin{bmatrix} \frac{-33.89}{(98.02\,s+1)(0.42\,s+1)} & \frac{32.63}{(99.6\,s+1)(0.35\,s+1)} \\ \frac{-18.85}{(73.43\,s+1)(0.30\,s+1)} & \frac{34.84}{(110.5\,s+1)(0.03\,s+1)} \end{bmatrix} \tag{5.14}$$

It is assumed that a suitable configuration has been found e.g. by the pairing rules presented in Section 3.2.

The output uncertainty w_O and performance weight w_P is according to [2]

$$w_O = \frac{500\,s + 1000}{3\,s + 5000} \tag{5.15}$$

$$w_P = \frac{s + 1000}{1000\,s + 1} \tag{5.16}$$

The SSP algorithm in Section 5.1 is tested at the system G_c. Since the case study used in [2], the same PID parameters should be obtained. These settings are presented in Table 5.1.

Table 5.1: Multi-loop PID parameters obtained in [2]

	k_c	τ_I	τ_D	$\epsilon\tau_D$
loop 1	-11.25	46571.70	$8.2 \cdot 10^{-10}$	$1.00 \cdot 10e - 5$
loop 2	15.49	3.87	$2.27 \cdot 10^{-7}$	$1.22 \cdot^{-5}$

The initial values to the SSP algorithm in Table 5.2 where applied.

Table 5.2: Initial parameters to the SSP algorithm

ϵ	$n_{\max}$	ζ	α	β
$1 \cdot 10^{-3}$	100	$1 \cdot 10^{-4}$	0.95	1.05

Appropriate initial values for the control parameters must satisfy the constraint $\|w_O(s)H(s)\|_\infty < 1$ in (4.11). To find such settings the individual diagonal elements where tuned according to the SIMC rule in Section 3.1.3. This gave the parameters in Table 5.3. These settings gave overall nominal stability and the $\bar{\sigma}(W_O H(j\omega))$ frequency plot are shown in Figure 5.2. From (2.28) it is seen that the peak value of $\bar{\sigma}(W_O H(j\omega))$ corresponds to its infinity norm. Hence, the SIMC settings satisfies (4.11) and should be a feasible solution to (5.4).

When these values where applied to the SSP algorithm in Section 5.1.3, the updated values of A_k, C_k, D_k and X in Step 3 never became feasible even if ϵ where increased

Table 5.3: Multi-loop PID parameters obtained with SIMC rules at the diagonal elements $g_{c\,ii}$

	k_c	τ_I	τ_D	$\epsilon\tau_D$
loop 1	-0.5785	20.0	0.4200	0.042
loop 2	0.6343	20.0	0.030	0.003

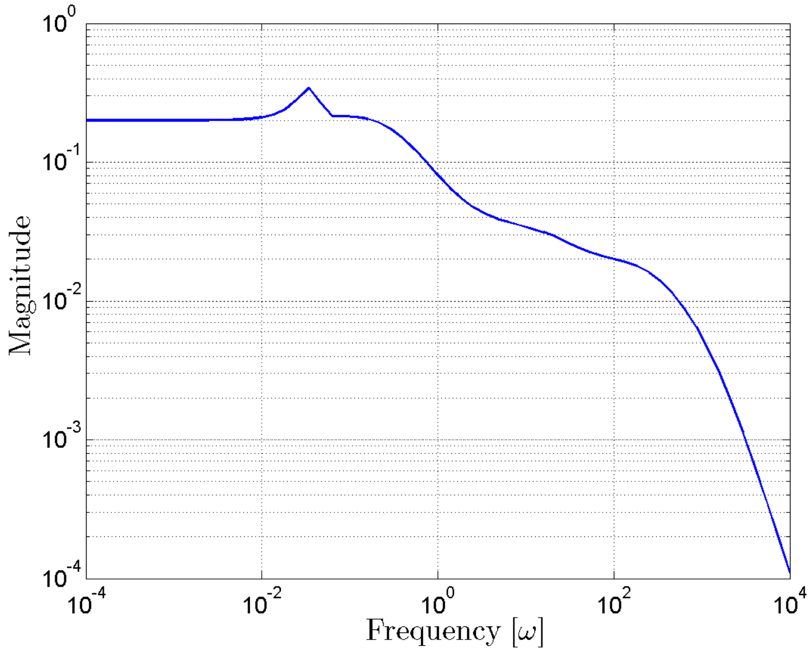

Figure 5.2: The maximum singular value of $W_O H(j\omega)$ over a frequency range when SIMC tuning rules where applied at the $g_{c\,ii}$ elements.

$n_{\max} = 100$ times. The feasibility where tested in the following way using the function `brlfeas`:

1. Initialize LMI variable γ and construct the LMI using the uprated values for A_k, C_k, D_k and X.

2. Use `feasp` at the resulting LMI, and return the value `tmin` and `xfeas`.

It could indicate that ϵ initially were set to a too large value, since the ϵ is increased every time (5.4) is not feasible. First ϵ where set to 0.1, then 0.01 and finally 0.001. None of the different initial values of ϵ gave feasible solutions to (5.4). If ϵ reached a too large value, then (5.9) where found infeasible. The value of ϵ can not be too large, due the approximated solution will be too far away from the accurate solution.

The value of $t_{\min}$ became bigger after every step. This could indicate that the problem went further away from a feasible solution, since a value of $t_{\min}$ less than zero mean that the problem is feasible. The reason for this behavior is not known. Closer analysis of the implemented algorithm must be done.

A different initial starting point where tried to see if the SIMC initial values where too far away from the settings in Table 5.1. Just a slight adjustment of the values presented in Table 5.1 where applied. The values gave nominal stability, and are shown in Table 5.4. The $\bar{\sigma}(W_O H(j\omega))$ frequency plot for these settings are shown in Figure 5.3.

Table 5.4: Multi-loop PID parameters close to the ones obtained in [2]

	k_c	τ_I	τ_D	$\epsilon\tau_D$
loop 1	-12	4600	$9 \cdot 10^{-10}$	$2.00 \cdot 10e - 5$
loop 2	14	4	$3 \cdot 10^{-7}$	$2 \cdot^{-5}$

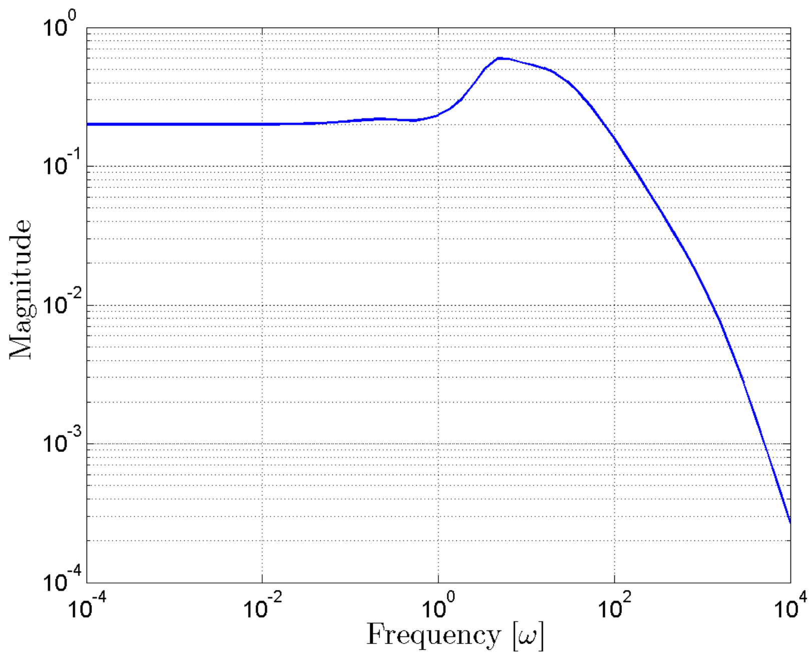

Figure 5.3: The maximum singular value of $W_O H(j\omega)$ over a frequency range.

The peak value of $\bar{\sigma}(W_O H(j\omega))$ is below one. Hence, constraint (4.11) is satisfied, and the problem in (5.4) should be feasible with these values. When $\epsilon = 0.001$ where applied, the approximated problem in (5.9) became infeasible after six steps. Also in this case the value of $t_{\max}$ were increased in each step when the feasibility of (5.4) were checked in Step 3.

The state-space representation of H_{cl} could of course be wrong. But due to the comparison performed as mentioned in Section 4.1.4 the calculation of the system matrices should be correct, and the error must be in the implementation of the SSP algorithm. This comparison is done in the MATLAB script `testSystemMatrix` in the software attached.

A better way of implementing the magnitude bound of the step values δA_k, δC_k, δD_k and δX can be investigated. The implementation presented here may be too simple.

As outlined in Section 4.1 the result presented in [2] has a different value of $c_{1\,11}$ in (4.43). This difference could have been applied in the simulations in [2]. To see if the parameters in Table 5.1 is obtained if $c_{1\,11}$ is changed to the value in [2] can be investigated. If so occurs, the SSP algorithm may be incorrect. This is a strong claim. Most likely an implementation error has been done.

Chapter 6

Further work

As mentioned, the different multi-loop methods presented in Section 3.2 have different strengths and weaknesses. To reveal these strengths and weaknesses, they should all be tested to a set of difficult case studies with uncertainty and a high level of interactions. It would then be clearer what improvements that must be done to achieve all the properties of the PID settings presented in Section 1.3.

To evaluate the tuning procedure in Section 4.2 and make it user-friendly, the following tasks should be addressed.

Debugging the implementation of the SSP algorithm Concerning the multi-loop H_∞ tuning method presented in this thesis, the implementation of the SSP algorithm must be evaluated. The reason why the variable $t_{\min}$ were increased for each step when feasibility of (5.4) were tested, must be found. Whether the extension (5.13) of the LMI constraints is a correct way to bound the magnitude of δA_k, δC_k, δD_k and δX must also be investigated.

Evaluate the purposed multi-loop H_∞ procedure The multi-loop H_∞ procedure in [2] should be tested to a system that introduce instability if one or several loops fail. Then the proposed H_∞ algorithm with failure tolerance in Section 4.2 should be applied to the same system to see if the system remains stable if same loops fail. The independent design gives integrity, but results in a conservative design. If the plant is too interactive, the independent design is not suited. Hence, the purposed method should also be compared to the independent design to see if better performance is achieved, or applied to plants where the independent design cannot be used.

Extend the multi-loop H_∞ method to handle other uncertainty descriptions The derivation of the method limited the system to be described with an output uncertainty with complex perturbations. To find other optimization problems so input uncertainty and structured perturbations are handled, may be desirable.

Design a user interface An interface should be made, so the procedure in Section 4.2 is easy to use. The user must choose the SSP parameters such as the convergence tolerance ζ, maximum iterations $n_{\max}$, the initial solution radius ϵ and the update values α and β. The nominal plant with uncertainty and performance weights should be user defined. Simulation results should be presented in a well arranged way. A list of which loops that may be brought out of service without introducing instability is a property that this interface should possess.

Chapter 7

Conclusions

7.1 Literature survey

Some main points of Section 3.3 are here repeated.

Independent design A mathematical model is needed, but not an accurate one. The system must be open stable and the loops must be somewhat decoupled to use this method. Integrity is obtained. The design method does not consider the information in the other loops. Hence it may lead to a conservative design.

Detuning This method is simple to use. The system must be open stable. The controller settings obtained are conservative and do not guarantee integrity. The robust stability and performance is not quantified since weighting functions are not included in the design.

Sequential closing To use this method the system must be open stable. The method is suited if response time differences in the closed loops are acceptable. Robustness and performance requirements of the overall system can be achieved. Failure tolerance is not guaranteed if the inner loops fail.

Iterative or trial and error The system must be open stable. Integrity is not guaranteed.

Optimization The method can be used to find stabilizing settings for an open unstable plant. A mathematical model is required. This method is time consuming, and the settings are recommended to be tested at a plant-simulator before applied at the physical plant. This method handles interactions superiorly, and offer robust controller settings. Failure tolerance is not guaranteed.

Relay feedback A plant model is not needed, but can be obtained form the relay feedback tests. The system must be open stable. Integrity is not guaranteed.

7.2 Multi-loop H_∞ optimal method

Based at the literature survey in Chapter 3, it was decided to work further with the multi-loop H_∞ optimal procedure in [2]. This tuning procedure gives superior interaction handling and ensures robust stability and nominal performance property as well. It can also be used to stabilize an open unstable plant. With the constraint $H_{cl} < \sqrt{2}$, robust performance is also ensured. The multi-loop H_∞ optimal method is outlined for a plant with an output uncertainty model restricted to unstructured perturbations. Since the optimization uses the interaction to achieve optimal performance, the loops are dependent of each other. Hence integrity is most likely not obtained using this method.

In [2] the matrix C_1 were found to be

$$C_1 = \begin{bmatrix} \gamma D_{wp} D_g & \gamma C_p & 0 \\ D_{wo} C_g & 0 & C_{wo} \end{bmatrix} \tag{7.1}$$

This result has been repeated several times in the paper. The calculations in this thesis showed that

$$C_1 = \begin{bmatrix} \gamma D_{wp} C_g & \gamma C_p & 0 \\ D_{wo} C_g & 0 & C_{wo} \end{bmatrix} \tag{7.2}$$

The results in this thesis have been backed up by comparing results obtained by MATLAB functions.

The constraints in the H_∞ optimal problem is represented with some Linear Matrix Inequaleties. The resulting non-convex optimization problem is solved by using Successive Semidefinite Programming, which iterates to a local optimum. Hence several initial values must be considered to find the possible global optimum.

7.3 Multi-loop H_∞ optimal method with failure tolerance

It is shown how to use the multi-loop H_∞ optimal method so integrity is obtained if the plant $G(s)$ is open stable. If there exists a solution to the optimization problem, all the desired properties of a PI/PID tuning approach, presented in Section 1.3, are achieved. This method could be used to derive settings for unstable plants, and reveal which loops that must be intact for the system to be overall stable.

Due to the complexity of the algorithm, this tuning approach is only suited for off-line tuning, and if a linear plant model is available. The obtained settings should be tested to an accurate simulator of the plant before applied at the real plant.

7.4 Implementation

The SSP algorithm has been tried implemented in MATLAB. To use existing LMI functions in MATLAB, some additional constraints have been applied to bound the solution radii when the optimization problem in (5.4) is solved.

It seems that the implemented algorithm iterates away from the feasible region, when the optimal step results in an infeasible solution. This is most likely attributed to an error in the implementation. This error has not been found.

Bibliography

[1] Faanes A. and Skogestad S. A systematic approach to the design of buffer tanks. *Computers and Chemical Engineering*, 24(5):1395–1401, 2000.

[2] Forbes J.F Bao J. and McLellan P.J. Robust Multiloop PID Controller Design: A Successive Semidefinite Programing Apprach. *Ind. Eng. Chem. Res*, 38:3407–3419, 1999.

[3] McLellan P.J Bao J. and Forbes J.F. A Passivity-Based Analysis for Decentralized Integral Controllability. *Automatica*, 38:243–247, 2002.

[4] Feron E. Boyd S., Ghaoui L.E. and Balakrishnan V. *Linear Matrix Inequalities in System and Control Theory*. SIAM: Philidelphia, PA, 1994.

[5] Chen C.-T. *Linear System Theory and Design*. Oxford University Press, third edition, 1999.

[6] Yu C.C. *Autotuning of PID Controllers: A Relay Feedback Approach*. Springer London, second edition, 2006.

[7] Cheng Y.M. Chen B.S and Lee C.H. A genetic apprach to mixed h_2/h_∞ optimal PID control. *IEEE Control Syst.*, 15(5):51–60, 1995.

[8] Seborg D.E. Chen D. Design of decentralized PI control systems based on Nyquist stability analysis. *Journal of Process Control*, 13:27–39, 2003.

[9] Seborg D.E. Chen D. Multiloop PI/PID controller design based on Greshgorin bands. *Journal of Process Control*, 13:27–39, 2003.

[10] Bristol E.H. On a New Measure of Interaction for Multivariable Process Control. *IEEE transactions on automatic control*, 11:133–134, 1966.

[11] Palmor Z.J. Halevi Y. and Efrati T. Automatic tuning for decentralized PID controllers for MIMO processes. *J. Process Control*, 7:119–128, 1997.

[12] Åström K.J. Hang C.C. and Wang Q.G. Relay feedback auto-tuning of process controllers – a tutorial review. *Journal of Process Control*, 12:143–162, 2002.

[13] Rosenbrock H.H. *Computer-Aided Control System Design.* Academic Press, New York, 1974.

[14] Khalil H.K. *Nonlinear Systems.* Prentice Hall, third edition, 2002.

[15] Bieker H.P. Assessment of Teoretical and Industrial Status with Respect to Tuning Support of Diagonal PID Control Structures for Oil and Gas Producing and Poduction Systems. PhD Try Lecture.

[16] Chiang C. Huang H.H., Jeng J. and Pan W. A direct method for multi-loop PI/PID controller design. *Journal of Process Control*, 13:769–783, 2003.

[17] Jeng J.C. Huang H.P. Monitoring and assessment of control performance for single loop systems. *Ind. Eng. Chem. Res.*, 41:1297–1309, 2002.

[18] Doyle J. Analysis of feedback systems with structured uncertainties. *IEE Proceedings, Part D-Control Theory and Applications*, 129(6):242–250, Nov 1982.

[19] Nocedal J. and Wright S.J. *Numerical Optimization.* Springer, 1999.

[20] Grefensette J.J. Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-16(1):122–128, jan/feb 1986.

[21] Maciejowski J.M. *Multivariable Feedback Design Design.* Wesly, Reading, MA, 1989.

[22] Gravdahl J.T and Egeland O. Centrifugal Compressor Surge and Speed Control. *IEEE Transactions on Control Systems and Technology, Man, and Cybernetics*, 7(5):567–579, sept 1999.

[23] Goh K.C. and Safonov M.G. Robust synthesis via bilinear matrix inequalities. *Int. J. of Robust and Nonlinear Control*, 6:1079–1095, 1996.

[24] Ljung L. *System Identification, theory for the user.* Prentice Hall PTR, second edition, 1999.

[25] Vandenberghe L. and Boyd S. Semidefintie programming. *SIAM Review*, 38(1):49–95, mar 1996.

[26] Cho W. Lee J. and Edgar T.F. Multiloop PI controller tuning for interacting multivariable processes. *Computers Chem. Engng*, 22(11):1711–1723, 1998.

[27] Quek C.K. Loh A.P., Hang C.P and Vasnani V.U. Auto-tuning of multi-loop proportional-integral controllers using relay feedback. *Ind. Eng. Chem. Res*, 32:1102–1107, 1993.

[28] Chiu M. and Arkun Y. A Methodology for Seqential Design of Robust Decentralized Control Systems. *Automatica*, 28(5):997–1001, 1992.

[29] Hovd M. and Skogestad S. Improved Independent design of Robust Decentralized Controllers. *J. Process Control*, 3(1):43–51, 1993.

[30] Hovd M. and Skogestad S. Sequential Design of Decentralized Controllers. *Automatica*, 30(10):1601–1607, 1994.

[31] Skogestad S. Morari M. Control Configuration Selection for Distillation Columns. *AIChE*, 33(10):1620–1635, 1987.

[32] J.M Maciejowski. *Predictive Control with Constraints*. Pearson Prentice Hall, 2002.

[33] Gimble M.J. H_∞ controllers with PID structure. *J. Dyn. Syst. Meas. Control*, 112:325–336, 1990.

[34] Yu C. Monica T.J. and Luyben W.L. Improved Multiloop Single-Input/Single-Output (SISO) Controllers for Multivariable Processes. *Ind. Eng. Chem. Res*, 27:969–973, 1988.

[35] Grosdidier P. and Morari M. Interaction Measures for Systems Under Decentralized Control. *Automatica*, 22(3):309–319, 1986.

[36] Skogestad S. Lundström P and Jacobsen E.W. Selecting the Best Destillation Control Configuration. *Ind. Eng. Chem. Res.*, 27(10):1848–1862, 1988.

[37] Campo P.J. and Morari M. Achievable closed-loop properties of system under decentralized control: Conditions involving the steady-state gain. *IEEE Trans. Automatic Control*, 39(5):932–943, 1994.

[38] Skogestad S. Dynamics and Control of Destillation Columns: A tutorial introduction. *Trans. IChemE*, 75(Part A):539–562, sept 1997.

[39] Skogestad S. and Postlethwaite I. *Multivariable Feedback Control, Analysis and Design*. Wiley, second edition, 2005.

[40] Skogestad S. and Morari M. Some new properties of the structured singular value. *IEEE Trans. Aut. Contol*, 33(12):1151–1154, 1988.

[41] Skogestad S. and Morari M. Robust Performance of Decentralized Control Systems by Independent Designs. *Automatica*, 25(1):119–125, 1989.

[42] Skogestad S. and Morari M. Understanding the Dynamic Behavior of Destillation Columns. *AIChE Journal*, 30(5):753–764, may 1990.

[43] Shen S.-H. and Yu C.C. Use relay-feedback test for automatic tuning of multivariable systems. *AlChEJ*, 40:627–646, 1994.

[44] Panagoulos H. Åström K.J. and Hägglund T. Design of PI Controllers based on Non-Convex Optimization. *Automatica*, 34(5):585–601, 1998.

[45] Balchen J.G. Andresen T. and Foss B.A. *Reguleringsteknikk*. Department of Engineering Cybernetics, fifth edition, 2003. In Norwegian.

[46] Chen T. Tan W. and Marquez H.J. Robust controller design and PID tuning for multivariable processes. *Asian J. Control*, 4(4):439–451, 2002.

[47] Luyben W.L. Tyreus B.D. Tuning PI controllers for integrator/dead time processes. *Ind. Eng. Chem. Process Des. Dev.*, 31(11):2625–2628, 1992.

[48] Sivakumaran N. Chandra J.S. Viknesh R. and Radhakrishnan T.K. A Critical Study of Decentralized Controllers for a Multivariable System. *Chem. Eng Technol.*, 27(8):880–889, 2004.

[49] Williams D. Vlachos C. and Gomm J.B. Genetic approach to decentralised PI controller tuning for multivariable processes. *IEE Proc.-Control Theory Appl.*, 146(1):58–64, jan 1999.

[50] Luyben W.L. Simple Method for Tuning SISO Controllers in Multivariable Systems. *Ind. Eng. Chem. Process Des. Dev.*, 25:654–660, 1986.

Appendix A

Some definitions

Positive definite and positive semidefinite matreces

The definition of a positive definite and positive semidefinite matrix is defined in Theorem 3.7 in [5] at page 74

Theorem 9. *A symmetric $n \times n$ matrix M is positive definite (positive semidefinite) if and only if any one of the following conditions holds*

1. *Every eigenvalue of M is positive (zero or positive)*
2. *All of the leading principal minors of M are positive (all of the leading principal minors of M are zero or positive)*
3. *There exists an $n \times n$ nonsingular matrix N ($n \times n$ singular matrix N or an $m \times n$ matrix N with $m < n$) such that $M = N^T N$*

The norm of a vector and a matrix

The induced $p-$norm for a vector x and matrix A is defined in [14] at page 647 and648 respectively.

$$\|x\|_p = (|x_1|^p + ... + |x_n|^p)^{1/p} \tag{A.1}$$

$$\|A\|_p = \sup_{x \neq 0} \frac{\|A\,x\|_p}{\|x\|_p} = \max_{\|x\|_p = 1} \|A\,x\|_p \tag{A.2}$$

when $p = \infty$

$$\|x\|_\infty = \max_i |x_i| \tag{A.3}$$

$$\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}| \tag{A.4}$$

Appendix B

Approximated Linear Matrix Inequalities

To show how the approximated LMI in (4.79) and (5.10) relates to the state-space representation of the system and control matrices, the symmetric positive definite matrix $\delta X + X_0$ is divided into 16 submatrices as

$$\delta X + X_0 = \begin{bmatrix} \delta X_{11} + X_{110} & \delta X_{12} + X_{120} & \delta X_{13} + X_{130} & \delta X_{14} + X_{140} \\ \delta X_{12}^T + X_{120}^T & \delta X_{22} + X_{220} & \delta X_{23} + X_{230} & \delta X_{24} + X_{240} \\ \delta X_{13}^T + X_{130}^T & \delta X_{23}^T + X_{230}^T & \delta X_{33}^T + X_{330} & \delta X_{34} + X_{340} \\ \delta X_{14}^T + X_{140}^T & \delta X_{24}^T + X_{240}^T & \delta X_{34}^T + X_{340}^T & \delta X_{44} + X_{440} \end{bmatrix} \tag{B.1}$$

Equation (4.79) and (5.10) are also symmetric, and is represented as

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} & A_{17} \\ A_{12}^T & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} & A_{27} \\ A_{13}^T & A_{23}^T & A_{33} & A_{34} & A_{35} & A_{36} & A_{37} \\ A_{14}^T & A_{24}^T & A_{34}^T & A_{44} & A_{45} & A_{46} & A_{47} \\ A_{15}^T & A_{25}^T & A_{35}^T & A_{45}^T & A_{55} & A_{56} & A_{57} \\ A_{16}^T & A_{26}^T & A_{36}^T & A_{46}^T & A_{56}^T & A_{66} & A_{67} \\ A_{17}^T & A_{27}^T & A_{37}^T & A_{47}^T & A_{57}^T & A_{67}^T & A_{77} \end{bmatrix} < 0 \tag{B.2}$$

Note that the elements A_{ii} is symmetric matrices. Then

$$\begin{aligned} A_{11} = \; & (A_g^T \delta X_{11} + \delta X_{11} A_g) + (A_g^T X_{110} + X_{110} A_g) - (C_g^T D_{k0}^T B_g^T \delta X_{11} + \delta X_{11} B_g D_{k0} C_g) \\ & -(C_g^T \delta D_k^T B_g^T X_{110} + X_{110} B_g \delta D_k C_g) - (C_g^T D_{k0}^T B_g^T X_{110} + X_{110} B_g D_{k0} C_g) \\ & -(C_g^T B_{wp}^T X_{120}^T + X_{120} B_{wp} C_g) - (C_g^T B_{wp}^T \delta X_{12}^T + \delta X_{12} B_{wp} C_g) \\ & +(C_g^T B_{wo}^T X_{130}^T + X_{130} B_{wo} C_g) + (C_g^T B_{wo}^T \delta X_{13}^T + \delta X_{13} B_{wo} C_g) \\ & -(C_g^T B_k^T X_{140}^T + X_{140} B_k C_g) - (C_g^T B_k^T \delta X_{14}^T + \delta X_{14} B_k C_g) \end{aligned} \tag{B.3}$$

$$
\begin{aligned}
A_{12} = \quad & (A_g^T \delta X_{12} + A_g^T X_{120}) - (C_g^T D_{k0}^T B_g^T \delta X_{12} + C_g^T \delta D_k^T B_g^T X_{120} + C_g^T D_{k0}^T B_g^T X_{120}) \\
& -(C_g^T B_{wp}^T X_{220}^T + C_g^T B_{wp}^T \delta X_{22}) + (C_g^T B_{wo}^T X_{230}^T + C_g^T B_{wo}^T \delta X_{23}^T) \\
& -(C_g^T B_k^T X_{240}^T + C_g^T B_k^T \delta X_{24}^T) + (\delta X_{12} A_{wp} + X_{120} A_{wp})
\end{aligned} \tag{B.4}
$$

$$
\begin{aligned}
A_{13} = \quad & (A_g^T \delta X_{13} + A_g^T X_{130}) - (C_g^T D_{k0}^T B_g^T \delta X_{13} + C_g^T \delta D_k^T B_g^T X_{130} + C_g^T D_{k0}^T B_g^T X_{130}) \\
& -(C_g^T B_{wp}^T X_{230}^T + C_g^T B_{wp}^T \delta X_{23}) + (C_g^T B_{wo}^T X_{330}^T + C_g^T B_{wo}^T \delta X_{33}^T) \\
& -(C_g^T B_k^T X_{340}^T + C_g^T B_k^T \delta X_{34}^T) + (\delta X_{13} A_{wo} + X_{130} A_{wo})
\end{aligned} \tag{B.5}
$$

$$
\begin{aligned}
A_{14} = \quad & (A_g^T \delta X_{14} + A_g^T X_{140}) - (C_g^T D_{k0}^T B_g^T \delta X_{14} + C_g^T \delta D_k^T B_g^T X_{140} + C_g^T D_{k0}^T B_g^T X_{140}) \\
& -(C_g^T B_{wp}^T X_{240}^T + C_g^T B_{wp}^T \delta X_{24}) + (C_g^T B_{wo}^T X_{340}^T + C_g^T B_{wo}^T \delta X_{34}^T) \\
& -(C_g^T B_k^T X_{440}^T + C_g^T B_k^T \delta X_{44}^T) + (X_{110} B_g C_{k0} + \delta X_{110} B_g C_{k0} + X_{110} B_g \delta C_k) \\
& (\delta X_{14} A_{ko} + X_{140} A_{ko} + X_{130} \delta A_k)
\end{aligned} \tag{B.6}
$$

$$
A_{15} = (X_{110} B_g D_{k0} + \delta X_{110} B_g D_{k0} + X_{110} B_g \delta D_k) + X_{120} B_{wp} + \delta X_{12} B_{wp} + X_{140} B_k + \delta X_{14} B_k \tag{B.7}
$$

$$
A_{16} = -C_g^T D_{wp}^T \gamma \tag{B.8}
$$

$$
A_{17} = C_g^T D_{wo}^T \tag{B.9}
$$

$$
A_{22} = A_{wp}^T X_{220} + A_{wp}^T \delta X_{22} + X_{220} A_{wp} + \delta X_{22} A_{wp} \tag{B.10}
$$

$$
A_{23} = A_{wp}^T X_{230} + A_{wp}^T \delta X_{23} + X_{230} A_{wo} + \delta X_{23} A_{wo} \tag{B.11}
$$

$$
\begin{aligned}
A_{24} \quad & = A_{wp}^T X_{240} + A_{wp}^T \delta X_{24} + X_{240} \delta A_k + \delta X_{24} A_k + X_{240} A_{k0} \\
& + X_{120} B_g \delta C_k + \delta X_{12} B_g C_k + X_{120} B_g C_{k0}
\end{aligned} \tag{B.12}
$$

$$
A_{25} = X_{120}^T B_g \delta D_k + \delta X_{12} B_g D_k + X_{120} B_g D_{k0} + X_{220} B_{wp} + \delta X_{22} B_{wp} + X_{240} B_k + \delta X_{24} B_k \tag{B.13}
$$

$$
A_{26} = C_p^T \gamma \tag{B.14}
$$

$$
A_{27} = 0 \tag{B.15}
$$

$$A_{33} = A_{wo}^T X_{330} + A_{wo}^T \delta X_{33} + X_{330} A_{wo} + \delta X_{33} A_{wo} \tag{B.16}$$

$$\begin{aligned} A_{34} = \; & A_{wo}^T X_{340} + A_{wo}^T \delta X_{34} + X_{340} \delta A_k + \delta X_{34} A_k + X_{340} A_{k0} + X_{130} B_g \delta C_k \\ & + \delta X_{13} B_g C_k + X_{130} B_g C_{k0} \end{aligned} \tag{B.17}$$

$$A_{35} = X_{130}^T B_g \delta D_k + \delta X_{13} B_g D_k + X_{130} B_g D_{k0} + X_{230} B_{wp} + \delta X_{23} B_{wp} + X_{340} B_k + \delta X_{34} B_k \tag{B.18}$$

$$A_{36} = 0 \tag{B.19}$$

$$A_{37} = C_{wo}^T \tag{B.20}$$

$$\begin{aligned} A_{44} = \; & C_{k0}^T B_g^T X_{140} + \delta C_k^T B_g^T X_{140} + C_{k0}^T B_g^T \delta X_{14} + A_{k0}^T X_{440} + \delta A_k^T X_{440} + A_{k0}^T \delta X_{44} \\ & + X_{140}^T B_g C_{k0} + \delta X_{14}^T B_g C_{k0} + X_{140}^T B_g \delta C_k + X_{440} A_{k0} + \delta X_{44} A_{k0} + X_{440} \delta A_k \end{aligned} \tag{B.21}$$

$$A_{45} = X_{140}^T B_g D_{k0} + \delta X_{14}^T B_g D_{k0} + X_{140}^T B_g \delta D_k + X_{240}^T B_{wp} + \delta X_{24}^T B_{wp} + X_{440}^T B_k + \delta X_{44}^T B_k \tag{B.22}$$

$$A_{46} = 0 \tag{B.23}$$

$$A_{47} = 0 \tag{B.24}$$

$$A_{55} = -I \quad \dim(\text{row number of } XB_{cl}, \text{column number of } B_{cl}^T X) \tag{B.25}$$

$$A_{56} = D_{wp}^T \gamma \tag{B.26}$$

$$A_{57} = 0 \tag{B.27}$$

$$A_{66} = -I \quad \dim(\text{row number of } D_{wp}, \text{column number of } D_{wp}^T) \tag{B.28}$$

$$A_{67} = 0 \tag{B.29}$$

$$A_{77} = -I \quad \dim(\text{row number of } C_{wo}, \text{column number of } C_g^T D_{wo}^T) \tag{B.30}$$

All the matrices $\delta(\cdot)$ are functions of $\mathbf{x}$.

Appendix C

MATLAB code

The implementation of the Successive Semidefinite Programming procedure to solve the multi-loop H_∞ optimal problem can be found in a CD attached to the thesis. It is recommended to read the text in `readme.txt` before examining the code.