CHALMERS

Automated Linear Controller Design For Mildly Nonlinear Systems Using Quantitative Feedback Theory.

Roozbeh Kianfar

Department Of Signals And Systems,

Division Of Systems, Control and Mechatronics.

Chalmers University Of Tech. Göteborg, Sweden.

EXE054/2009

CONTENTS

1 ABSTRACT

A method to design simple linear controllers for mildly nonlinear systems is presented. In order to design the desired controller, the behavior of the nonlinear system is approximated with a set of linear systems which are derived through linearizations. Classical local linearization is carried out around stationary points. However, in order to obtain a better approximation of the nonlinear system selected non-stationary points are taken into account as well. This set of linear models are considered as an uncertainty description for a nominal plant. Quantitative Feedback Theory (QFT) may be used to guarantee that the specifications is fulfilled for all linear models in such an uncertainty set. Traditionally QFT design is carried out in a Nichols diagram by loop shaping of the nominal linear plant. This task is highly dependent on the experience of the designer and is difficult for unstable systems. In order to facilitate this task, an optimization algorithm based on Genetic algorithm is used to automatically synthesize a fixed structure controller. To illustrate and evaluate, the method is applied to a Wiener system and two nonlinear Bioreactor benchmark problems. In result of this type of design we succeeded to improve the robustness and transient behavior of the nonlinear systems. Furthermore, different criteria can be chosen as the objective function for to optimization to fulfill a given set of criterion.Simulation and phase-plane analysis are used to select the non-stationary points.

Keywords: Nonlinear, QFT, loop shaping, linearization, non-stationary point, genetic algorithm.

2 INTRODUCTION

2.1 Motivation

For most control problem there are several solutions, and every one of them might be interesting from a specific perspective. In the process industry ease of implementation is without doubt one of the most important aspects of automatic control. Provided the performance is acceptable, fixed structure and linear controllers, such as PID controllers, are therefore advantageous even though the process itself may be nonlinear. In line with this, the aim of the work presented here is a semi-automized method for determination of fixed structure low order linear controllers for mildly nonlinear single input single output (SISO) processes.

Depending on the character of a nonlinear process there are many methods for designing nonlinear controllers, such as Feedback linearization, Sliding control, Adaptive control and Model predictive control (c.f. [11]). However, in many control systems there is little support for these methods and operators are untrained in their use. As a consequence, most mildly nonlinear plants are controlled by linear controllers, mainly PID controllers, which are either tuned experimentally or synthesized for a specific operating point. Because of the nonlinearities the system will have deteriorating properties and may become unstable when operated too far away from the design point. The idea here is to find a controller parameterization that gives a robust system in the sense that it has an acceptable performance in a large operating region.

Since there is an abundance of efficient methods for synthesis of linear controllers from linear models, the use of a linear process model is in many cases motivated. In general these linear models are derived from Taylor expansion of nonlinear system at stationary points. Then, the controller designed for these linear systems is applied to nonlinear system in small regions around stationary points. In practice in many cases it is required that the feedback system works in a wider operating window than only a small region around stationary points. Finding suitable linear models for a nonlinear system which can be used in a wider operating window is still an open research area. One way is to use a linear model and treat the nonlinearities as model uncertainties. Schweickhardt and Allgöver [14, 15, 16] use this to define the best linear model as the one with the smallest gain of the uncertainty. In [15] they pursue by determining the linear controller such that the Small gain theorem can be used to guarantee stability. The drawbacks are difficulties in the computation of the nonlinearity measure (uncertainty gain), that the process needs to be stable, and that the use of the Small gain theorem introduces conservatism in the resulting solution. Basically, the problem of conservatism and its connection to the nonlinearity measure originates from the fact that the gain is considered for signals that the controllers might neither use nor apply. To some degree this can be taken into account by bounding the input amplitude [16].

Olesen *et al.* also used the idea of treating the nonlinearities as uncertainties and disturbances to show that only a few linear controllers are needed in gain scheduling control of the temperature in an exothermic tank reactor. They use model linearizations to generate a set of transfer functions that can be interpreted as a model uncertainty description. This is then followed by a controller design using Quantitative Feedback Theory (QFT) to guarantee robustness specifications for all transfer functions in the set. By adding non-stationary linearization points the robustness and operating window for each controller could be made significantly larger. The use of off-equilibrium linearization has also been shown to improve performance of gain scheduling control when the controller parameters are interpolated [10].

QFT was originally developed for linear systems with uncertainties (c.f. [9]), but there are also extensions to nonlinear systems with uncertainties, which are based on finding so-called equivalent linear models for the nonlinear system (see [1] and [2] and references therein). The main idea in the first nonlinear QFT technique was to replace the nonlinear system by a set of LTI systems for a set of acceptable outputs. However, this requires the knowledge of what specific input signal that generates the desired output, which currently limits its use. Prof. Horowitz also developed another method for nonlinear QFT, based on the replacement of nonlinearities by an equivalent disturbances set and a simple set of equivalent LTI systems [2].

Basically, standard linear QFT design is based on loop shaping the nominal loop transfer function such that for each frequency considered it does not violate frequency dependent Horowitz Sidi (HS) bounds. A drawback of standard QFT for linear systems is that the manual loop-shaping in the Nichols chart highly depends on the experience of the designer. During the last decade solutions to how to automate this step have therefore been proposed. Basically, they rely on optimization where the bounds constrain the search space. The optimization problem, however, is generally non-convex. Chait *et al.* convexifies the HS-bounds and solve the problem using linear programming. However, the use is limited because the method requires that the closed loop poles are known beforehand. Another approach is to use a global optimization routine. Nataraj *et al.* propose an interval analysis, and Chen *et al.* use genetic algorithm. In both of them the high frequency gain of controller considered as the objective function to be minimized. To improve the accuracy for a given numerical effort Fransson *et al.* [5] use a combination of a global (DIRECT method) and a local optimization routine. They also used different optimization criteria as the objective functions for different frequencies. It should be noted that optimized control of uncertain linear systems can also be determined using the structured singular value for the constraints, as in [6] and [17]. However, for SISO systems with less than 8 parameters to optimize the use of HS-bounds can in general be recommended [17].

2.2 Scope

The method presented here for nonlinear processes is based on the manual method used by Olesen *et al.*, combined with an optimization using genetic algorithm, which has the advantage that no initial guess is required - a valuable property from an automation point of view. Based on systematically selected simulations of the nonlinear system new linearization points are added to the set of transfer functions until performance and robustness are no longer improved. This method is then applied to a Wiener system studied by [15] and the problem is solved for two different optimization criterion. It has been shown that by choosing an appropriate cost function for different objectives, the result can be improved. The method is also applied to a nonlinear benchmark problem, an unstable bioreactor [13]. Some solutions have been proposed for this problem such as [4], though it appears as if no linear controller for the process has been evaluated earlier. The PID controller derived with the method presented here performs well over the operating window and also compare well to the sliding mode controller by Mehmed *et al.* We also applied our method to a fourth order nonlinear CSTR benchmark problem. In this problem non-stationary points are selected automatically using simulation of open loop system for small steps as the references and the results are compared to H_∞ solution given in [15]. The main goal in this work is to present a simple method for designing linear controllers for nonlinear system. The main differences between the presented method and gain scheduling are i) In this work by using optimization algorithm we tried to facilitate the controller design procedure and ii) to exploit a robust controller design method, we tried to increase the robustness of controller. In result of this type of design the number of required controllers is decreased compare to gain scheduling.

2.3 Thesis organization

The organization of this thesis is as follows:

Chapter 3: This chapter is devoted to a theoretical background on Quantitative Feedback Theory and Genetic algorithm.

Chapter 4: In this chapter, the automated controller synthesis using the Genetic algorithm for both linear and nonlinear systems is described. Two examples are presented to show the efficiency of the algorithm for linear systems. Then, a method to automatically design simple linear controllers for mildly nonlinear systems is presented. The method is successfully applied to a linear system followed by a static nonlinearity. It is also shown how different optimization criterion can improve our design.

Chapter 5: In this chapter the proposed method is applied to an unstable bioreactor benchmark problem. It is shown that linearization around the non-stationary points, in addition to stationary points improves the robustness.

Chapter 6: In order to further evaluate the method, a fourth order CSTR benchmark problem is selected. Non-stationary points are added to the uncertainty set through the simulation of the open loop system. The results are compared to an H_∞ design for the best linear model according to [15].

3 BACKGROUND

3.1 Quantitative Feedback Theory as a tool of design and analysis

QFT is a method for design and analysis of feedback control of uncertain systems, originally developed by I. Horowitz [9].

The philosophy behind QFT is that we do not need feedback in the control design unless the uncertainties in the plant parameters or disturbance uncertainties are more than the acceptable performance uncertainties. In other words, if the amount of these uncertainties is less than the acceptable amount of performance uncertainties we do not need feedback and an open loop control is sufficient. The amount of required feedback depends on the interaction of three sets $P = \{$uncertainties in the system$\}$, $D = \{$uncertainties due to the disturbance$\}$ and $A = \{$the acceptable uncertainties in the performance of system$\}$. The concepts of controllabilty and observability which play an important role in Modern control are inherently in the design procedure. Hence, there is no need for explicit test of controllabilty and observability. QFT can cope both parametric and unstructured uncertainties as well. An uncertain plant with parametric uncertainty can be defined as

$$P(s) \in \{P(s,q)\, q \in R^n \tag{3.1}$$

where q are then uncertain parameters. A plant with unstructured multiplicative uncertainty is written

$$P(s) = P_{nom}(s)(1 + M(s)),\ |M(s)| \leq m(s) \leq 1 \tag{3.2}$$

where $M(s)$ is an asymptotically stable transfer function. Uncertainties and specifications should be translated into the frequency domain, and one arbitrary plant transfer function P_{nom} is considered as the nominal one. Instead of simultaneous design for all the loop transfer functions defined by the uncertainties, the design can then be carried out only for the nominal loop transfer function, $L_{nom}(j\omega) = P_{nom}(j\omega)G(j\omega)$, where $G(j\omega)$ is the controller. This nominal loop transfer function is a frequency dependent function and should satisfy frequency dependent constraints for each frequency. In QFT for a single input single output system (SISO) we will use a two degree of freedom (DOF) controller (see Fig. 3.1), in one DOF controller the system response transfer function $T(s)$ and the sensitivity function $S(s)$ always depend on each other because $T(s) + S(s) = 1$, which results in limitation in the design procedure. If we add a prefilter $F(s)$ to the system the equation changes to $S = 1 - (T/F)$ where F is a free function. For a system which we can measure the output (Y) and input (R) there is no need for the controller to have more than two degree of freedoms. In QFT the design process is carried out for transfer functions and therefor, there is no need to use a state space description.

In other words, for instance for a single input single output (SISO) system we have access only to one of the states so there is no need to deal with the state space and large matrices. Another feature of QFT is the use of the nominal loop transfer function $L_{nom}(j\omega)$ instead of the nominal sensitivity function $S_{nom}(j\omega) = \frac{1}{1+L_{nom}(j\omega)}$ (in contrast to H_∞). As the Bode, one of the father features in feedback amplifier theory mentioned in [8], cost of feedback is mostly paid for bandwidth. The bandwidth of L_{nom} is determined by the interaction of the two sets of A and P, which were described before. The main reason for choosing L_{nom} over S_{nom} in QFT is that in the frequency range $[\omega_c, \omega_G]$where ω_c and ω_G are cross over frequency of L_{nom} and G respectively, the sensitivity function S_{nom} is very insensitive. The effect of noise N at the plant input U in the high frequency range where $|L_{nom}| \ll 1$ is

$$T_N = \frac{-U}{N} = \frac{G}{1 + L_{nom}} \approx G = \frac{L_{nom}}{P_{nom}}$$

The large noise amplification over the high frequency range ($[\omega_c, \omega_G]$) can saturate the system and as mentioned before the sensitivity function is insensitive in this important frequency range. In QFT design for a single input single output system, the first step is to define the plant uncertainties $P(j\omega)$ and $T(j\omega) = \frac{L(j\omega)}{1+L(j\omega)}$ tolerances quantitatively. Then we can easily find the resulting bounds on the nominal loop transfer function L_{nom}. As can be seen, the bounds on $T(j\omega)$ cause bounds on the L_{nom} in the Nichols diagram. Finally, in QFT design the loop transfer function should be shaped such that it satisfies frequency dependent boundaries ($B(j\omega)$) so called Horowitz-Sidi bounds. In QFT design, the trade off between cost of feedback (bandwidth), its benefits and the order of compensator is clear to the designer. Under the assumption that the set $\{P(\omega)\}$ is a connected set in the complex plane and the plant $P(s)$ is a strictly proper transfer function with a fixed excess of poles over zeros the stability is guaranteed for all $P(s) \in \{P_i(s)\}$. The formal proof for the foregoing claim is given in [8].

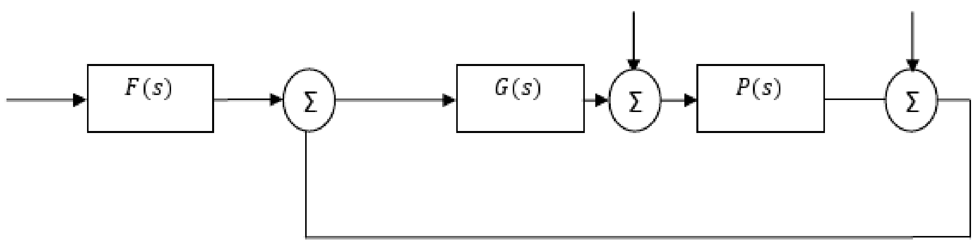

Figure 3.1. Two degree of freedom controller.

In general the design procedure in QFT includes two parts:

- Design the feedback compensator $G(s)$.
- Design the prefilter $F(s)$.

Now, we explain how these two task are carried out in practice.

3.1.1 Design the feedback compensator $G(s)$

The purpose of designing a feedback compensator $G(s)$ is to reduce the closed loop uncertainties such that they lie within the permissible envelope of the specifications. This task is carried out through the following steps:

- Define the uncertainties in the process by a set of transfer functions $P_i(s)$. These uncertainties can be either parametric or multiplicative. One of the plant transfer functions is selected to be the nominal one. Then we calculate the so called *templates* for selected frequencies. $p_i(j\omega_k), j = 1, ..n, \omega_k = \omega_1, ...\omega_N$ The template or value set shows the plant uncertainties at each specific frequency. The selection of these frequencies should be carried out with special care, because a large number of frequencies result in a large number of templates and causing calculation difficulties.

- Formulate closed loop specifications, such as servo specification and sensitivity specifications. These specifications should be given in frequency domain. Specifications in time domain is therefore translated to frequency domain using The QFT toolbox (Qsyn) for example.

- Use the templates and specifications to calculate the corresponding Horowitz-Sidi bounds for the specifications. Here we show the procedure for a simple case of servo specification.

$$a(\omega) \leq \left| \frac{F(j\omega)G(j\omega)P(j\omega)}{1 + G(j\omega)P(j\omega)} \right| \leq b(\omega) \tag{3.3}$$

  As there are no uncertainties in $F(s)$ the specification above indicates that the complex number $G(j\omega_k)$ at the frequency w_k must be determined such that

$$\frac{\max_i \left| \frac{G(j\omega_k)P_i(j\omega_k)}{1+G(j\omega_k)P_i(j\omega_k)} \right|}{\min_i \left| \frac{G(j\omega_k)P_i(j\omega_k)}{1+G(j\omega_k)P_i(j\omega_k)} \right|} \leq \frac{b(\omega_k)}{a(\omega_k)} \tag{3.4}$$

  The above equation is called the tolerance specification at the frequency ω_k. For some large values of $G(j\omega)$ the above equation can be satisfied and for some value smaller than a specific value it might not be satisfied. Hence, there exists borders $B_i(j\omega)$ in the complex plane between the permissible and impermissible values of L_{nom}, and those are called Horowitz-Sidi tolerance Bounds. In a quite similar approach we can define the Horowitz-Sidi bounds in respect to other specifications such as sensitivity function specification or input disturbance rejection. It is worth to mention again that if the equation:

$$\frac{\max_i |P_i(j\omega_k)|}{\min_i |P_i(j\omega_k)|} \leq \frac{b(\omega_k)}{a(\omega_k)} \forall \omega_k \tag{3.5}$$

  is satisfied for all frequency there is no need for feedback.

- Given the nominal plant and the Horowitz-Sidi bounds, exploit loop shaping techniques to shape the nominal loop transfer function such that it satisfies all the Horowitz-Sidi bounds. The design is carried out in frequency domain in Nichols diagram. This task requires a designer with enough experience in this area. Hence, a computer program that can do this task automatically can play a quite constructive role in finding a good design. When a loop transfer function that do not violate these H-S bounds is found, we can move on to the next phase.
- Check the stability of the closed loop system for all plants with Nyquist criterion.

3.1.2 Design of prefilter $F(s)$

If the system response is not within the acceptable servo specification envelope, a prefilter $F(s)$ is needed prior to the loop.

Over the last two decades Quantitative feedback theory is applied to many real engineering problems, such as process control systems, idle speed control for an automotive fuel injection engine, flight control, hydraulic actuator, etc.

3.2 Background on Genetic Algorithm

There exist many optimization and computational methods which are inspired by the biological evolution. Genetic algorithm (GA) is one of the most popular algorithms which belong to these evolutionary methods. It is a stochastic optimization method which is inspired by biological evolution based on Darwin's theorem. At first we begin by giving some common definitions which are used in the Genetic algorithm, then we explain how the Genetic algorithm works and explain about its features.

- *Fitness* function is the function we want to minimize. In standard optimization algorithm, it is called the objective function, cost function or lost function.
- An *individual* is any point or a vector with the equal length as the number of optimization variables that the fitness function is calculated for. The value of fitness function for each individual is called *score*.
- The *population* is a matrix which consists of individuals. For instance if the population size is 50 and there are 4 optimization variables in the fitness function, the population is represented by a 50-by-4 matrix. At each iteration a series of computations is performed on the current population to produce a new population which is called new generation.
- In a population, *diversity* shows the average distance between individuals. If a population has individuals with a large average distance, it is a high diversity

population and the population with a low average distance among individuals is a low diversity population. Diversity is an important factor in the Genetic algorithm because populations with high diversity makes it possible for the algorithm to cover a larger region of space.

- In the current population the genetic algorithm selects the individuals with better fitness values to produce the next generation, which is called the *children* or the *offspring*.

In an optimization problem that is solved using GA, the optimization variables are encoded in a string called *chromosomes*. Each chromosome consists of a number of *genes* that determine the characteristic of encoding scheme. The Algorithm is initialized with a set of random individuals called initial population. Individuals with a better fitness values from one population are taken to form new populations. These new solutions are called offspring. In each iteration, the individuals with better fitness survive and are used to reproduce the next generation. This iteration continues until a specific condition, e.g the maximum number of iterations, or a tolerance for acceptable fitness, is satisfied. Here we briefly introduce the different components of the Genetic algorithm and discuss about the interaction of these components with each other.

- Encoding Scheme.
- Fitness assignment.
- Crossover.
- Mutation.

Encoding Scheme: the representation of individual genes in a chromosome can be chosen in several different ways. One choice is binary encoding where genes take the values of 0 or 1. Another method is real number encoding, where genes take any value in the sector [0, R]. These chromosomes contain information about the solution of the problem.

Fitness assignment: The evaluation of an individual leads to a fitness assignment, which conveys information on the performance of the individuals. The simplest possible fitness assignment consists of simply assigning the value obtained from the evaluation without any transformations. This value is known as the raw fitness.

Crossover: Crossover is one of the most important operations in GA. In this stage crossover selects genes from parent chromosomes and creates a new offspring. In other words it allows partial solutions from different regions of the search space to be assembled into a complete solution of the problem.One easy way to do this operation is to choose randomly some crossover points. Then everything before this point copy from a first parent and the rest copy from another parent and the combination makes a new offspring.

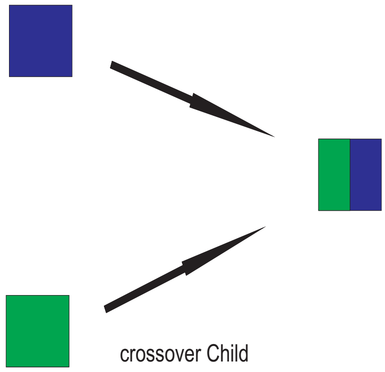

Figure 3.2. Crossover operation.

Mutation: Mutation is another important operator in GA. This operation takes place after the crossover. The main goal of mutation is to prevent all solutions from ending up at local minima. It randomly changes the genes of individual parents. For example in the case of binary encoding we can switch a few randomly chosen bits from 1 to 0 and vice versa.

Figure 3.3. Mutation operation.

In addition to the two types of crossover and mutation offspring, there is another type which is called elite children. Elite offsprings have the best fitness values among all individuals in the current generation. These individuals automatically are passed to the next generation.

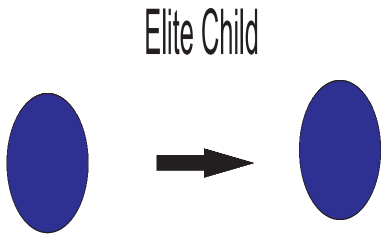

Figure 3.4. Elite child.

In genetic algorithm, there are several stopping criteria for the algorithm such as number of specific iteration or specific tolerance value. It is good to note that the operation of genetic algorithm is completely different from a random search. For instance the mutation, which provide new offsprings that GA can work with is random but selection of better individual is not random. For more comprehensive discussion about this issue the readers are referred to the references and study of schema theorem.

Genetic algorithm is a powerful global optimization method which can handle both constrained and non-constrained optimization problem. The constraints can be in the form of linear equality or inequality. It also cope with nonlinear equality and inequality constraints with bounds on the optimization variables. The Matlab toolbox for genetic algorithm is used in our work. This toolbox uses the Augmented lagrangian Genetic algorithm to solve nonlinear constraint problems. The optimization problem can be written as below

$$\min_{\theta} J(\theta)$$

subject to:

$$\begin{aligned} nc_i(\theta) &\leq 0 \ \forall i \\ nceq_i(\theta) &= 0 \ \forall i \\ A \cdot \theta &\leq b \\ Aeq \cdot \theta &= beq \\ lb &\leq \theta \leq ub \end{aligned}$$

where $nc_i(\theta)$ are the nonlinear inequality constraints, $nceq_i(\theta)$ are nonlinear equality constraints, $A\theta$ and $Aeq\theta$ represent the linear inequality and equality constraints respectively.lb and ub are lower and upper bounds on the optimization variables respectively. As we discussed earlier, the augmented lagrangian genetic algorithm can handle an optimization problem with nonlinear and linear constraints with bounds on the optimization variables. The algorithm deals with the linear constraints and bounds separately from nonlinear constraints. In order to solve such a problem, a subproblem is formulated using the Lagrangian and the penalty parameters. The subproblem is shown below:

$$\Psi(\theta, \lambda, s, \rho) = J(\theta) - \sum_{i=1}^{n} \lambda_i s_i \log(s_i - nc_i(\theta)) + \sum_{i=n+1}^{nt} \lambda_i nc_i(\theta) + \frac{\rho}{2} \sum_{i=n+1}^{nt} nc_i(\theta)^2, \quad (3.6)$$

where λ is a vector of nonnegative Lagrange multiplier estimates, s is the nonnegative shift vector and ρ is the penalty parameter. The algorithm starts with an initial value of ρ called the initial penalty. The genetic algorithm starts minimizing a sequence of this type of subproblems. If the desired accuracy is reached the Lagrangian multipliers are updated. Otherwise, the algorithm imposes a larger penalty parameter into the problem. (in genetic algorithm toolbox it is called PenaltyFactor). These steps are continued until the stopping criteria is satisfied.

4 AUTOMATED CONTROLLER SYNTHESIS USING GENETIC ALGORITHM

4.1 Controller Synthesis for Linear Systems

As we mentioned in the previous chapter, one difficulty in designing a controller using QFT is the manual loop-shaping. To overcome this difficulty, we can use an optimization algorithm to find a controller that not only satisfies the specifications but is also optimized with respect to a desired criterion.In practice, QFT loop-shaping is carried out in a Nichols diagram for a finite number of frequencies, $\Omega = \{\omega_k\}$. If we assume that the controller has a fixed structure

$$G(s) = \frac{\theta_m s^m + \theta_{m-1} s^{m-1} + ... + \theta_0}{s^n + \theta_{m+n} s^{n-1} + ... + \theta_{m+1}} \tag{4.1}$$

where θ is the parameter vector to be determined by optimization. The Horowitz-Sidi bounds at each frequency ω_i are denoted $B_i(\angle L_0(j\omega_i, \theta), \omega_i)$. These bounds have different shapes and may be single-valued or multiple valued, depending on the specifications. In general though, they are non-convex.

The objective here is to synthesize a controller such that:

- The Horowitz-Sidi bounds at each frequency ω_i are not violated.
- The nominal loop-transfer function is stable.
- The controller has low complexity.
- The controller is optimized with respect to the desired criterion.

In most QFT literature, the aim is to minimize the high frequency gain of the controller. In this work we follow that tradition. However, for one of the examples in this chapter, it has been shown that for different targets we can use different criteria to optimize the results. For instance the cost function $J(\theta)$ can be the high frequency gain of the controller or low frequency disturbance rejection, as in [5], or any other criteria.

In the next step, The Horowitz-Sidi bounds are translated to nonlinear constraint inequalities as below:

$$ub_i(\theta) = B_i(\angle L_0(j\omega_i, \theta), \omega_i) - |L_0(j\omega_i, \theta)| \leq 0 \tag{4.2}$$

$$lb_i(\theta) = |L_0(j\omega_i, \theta)| - B_i(\angle L_0(j\omega_i, \theta), \omega_i) \leq 0 \tag{4.3}$$

where ub_i and lb_i are upper and lower single-valued bound constraints. Multiple valued bounds are split into one upper and one lower bound. There are in general no analytical functions for these bounds and in this work we derive them numerically using the QSYN toolbox for Matlab [7]. The nominal closed loop transfer function stability imposes one more constraint to the problem: the roots λ of $1 + L_0 = 0$ should be in the LHP. Hence, the problem can now be formulated as

$$\min_{\theta} J(\theta)$$

subject to:

$$\begin{aligned} ub_i(\theta) &\leq 0 \ \forall i \\ lb_i(\theta) &\leq 0 \ \forall i \\ Re[\lambda(1 + L_0)] &\leq 0 \end{aligned}$$

This problem is classified in the global optimization category with nonlinear constraints, a class that classical gradient based optimization methods are generally not suited for. However, Genetic algorithm, which is a powerful evolutionary method with the ability to handle the nonlinear constraints, is a good candidate to solve this problem.

Advantages of this method are:

- There is no need for an initial guess.
- There is no need to determine optimization variable search space in advance, though it is still possible to do so.
- The structure of the controller can be determined by the designer to fit the target control system.
- It is possible to use the solution which is derived from the Genetic algorithm in a classical local optimization method to improve the solution. [5]

In order to illustrate the efficiency of this automated synthesis, a manual design example from the QSYN-manual is compared with the solution derived by optimization. In the Qsyn manual, two different controllers are presented for the following example. We also use the same controller structure with unknown coefficients and try to find these variables using our optimization algorithm.

Example 1 An uncertain plant, with parametric uncertainty is given:

$$P(s) = K.\frac{s + a}{1 + 2\zeta s/\omega_n + s^2/w_n^2} \tag{4.4}$$

where $K \in [2, 5]$, $a \in [1, 3]$, $\zeta \in [0.1, 0.6]$ and $\omega_n \in [4, 8]$.

The design specifications are:

$$M_T \leq 0.1 \tag{4.5}$$

$$T_s \leq 10s \tag{4.6}$$

$$\|S(j\omega)\| = \frac{1}{|1 + G(j\omega)P(j\omega)|} \leq 6dB \tag{4.7}$$

where M_T is the maximum overshoot of the closed loop step response, T_s is the settling time and $S(j\omega)$ is the sensitivity function. In the first case, the structure of $G(s)$ is

$$G(s) = \frac{\theta_3 s^3 + \theta_2 s^2 + \theta_1 s + \theta_0}{s^4 + \theta_6^3 + \theta_5 s^2 + \theta_4} \tag{4.8}$$

i.e. the same as the controller presented in the manual. First, the design specifications (4.5) and (4.6) are translated to the frequency domain. In QSYN this task is carried out through an approximation of the closed loop system by a low order system, from which the correspondence between time and frequency domain is found.

The objective function to be minimized is the high frequency gain of controller, i.e. $J(\theta) = \theta_3$. The optimization variables are the coefficients of $G(s)$, i.e. $\theta = [\theta_0, \theta_1, ...\theta_6]^\top$. The numerical values for Horowitz-Sidi bounds are calculated using Qsyn and are fed to the genetic algorithm toolbox In the Genetic algorithm toolbox we set the population size to 60 while the initial population is left blank, i.e. no initial guess is made and the initial range is considered $[0; 1]$. If we have a rough idea about the minimal point for a function in advance, we can set initial range such that the minimum point lies somewhere within the initial range. For instance, for an optimization problem, we know that the minimal point is around $[0, 0]$ the best choice for initial range is $[-1; 1]$. However, the genetic algorithm is still capable to find the minimum points for the cases that the initial range is not optimal.The other parameters in the toolbox set as the default. The optimization time highly depends on the number of optimization variables and population size, for this specific example with 6 variables and initial population size of 60, it is around 30 min.

As can be seen from Fig. 4.1 and 4.2, the automatically synthesized controller satisfies all specifications, that is, the nominal loop transfer function for the selected frequencies is outside the sensitivity HS bounds and above the bounds for the servo specifications.The nominal loop transfer function $L_0(j\omega)$ is stable and the high frequency gain of the controller in the automated design is less than the solution given in the manual.

Next, we solve the problem again but choosing the controller structure as:

$$G(s) = \frac{\theta_4 s^4 + \theta_3 s^3 + \theta_2 s^2 + \theta_1 s + \theta_0}{s^5 + \theta_8 s^4 + \theta_7 s^3 + \theta_5 s^2 + \theta_5} \tag{4.9}$$

and the result is illustrated in Fig. 4.3. Again we can observe that the solution which is derived using genetic algorithm satisfies all specifications and minimize the high frequency gain of the controller.The foregoing examples showed how the Genetic algorithm can be used to solve the loop-shaping problem for an uncertain linear system. In Fig. 4.4 and 4.5 the Matlab toolbox for the Genetic algorithm and some characteristics of the solution and features of toolbox for the mentioned example are shown.

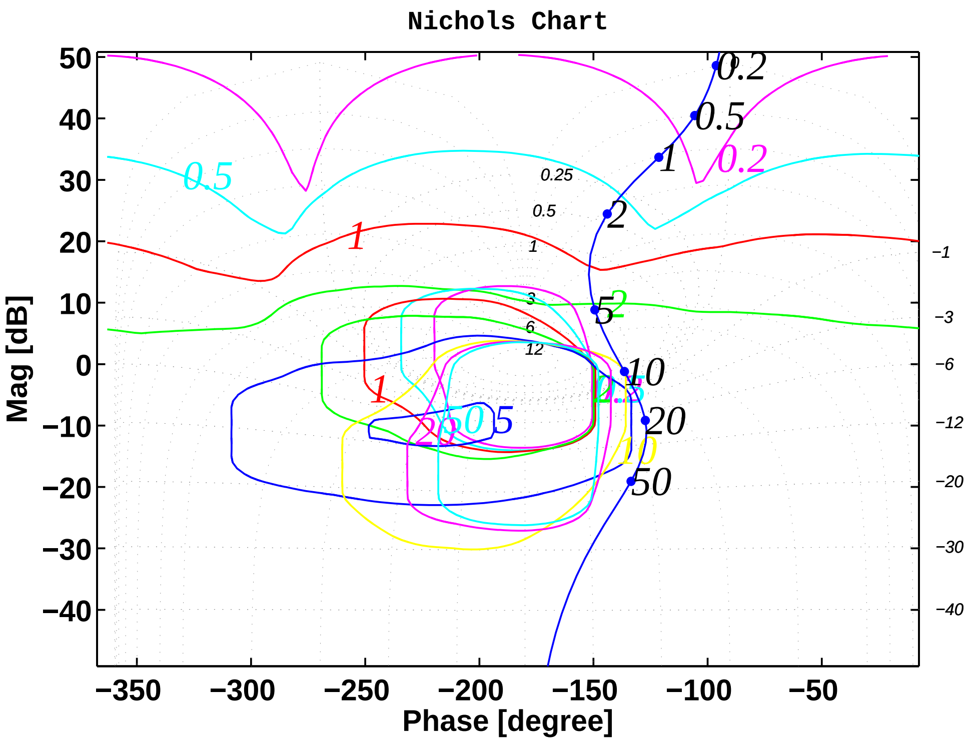

Figure 4.1. QFT solution for manual loop-shaping. The closed curve in the middle are sensitivity H-S bounds and the hat shape bounds are tolerance bounds.

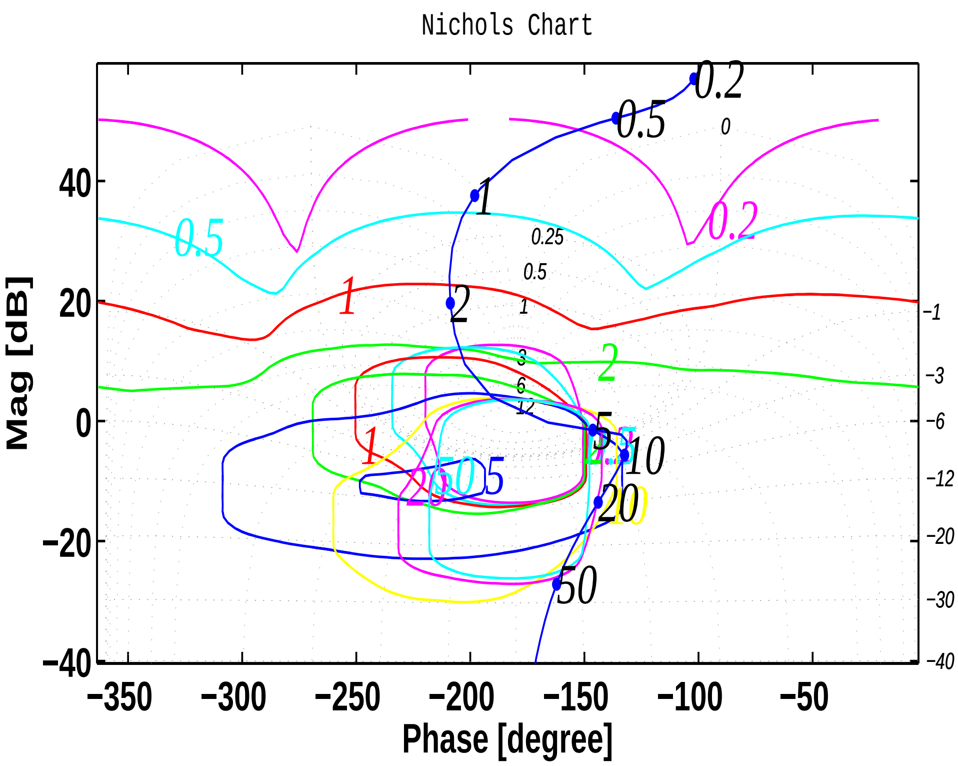

Figure 4.2. Automated loop-shaping

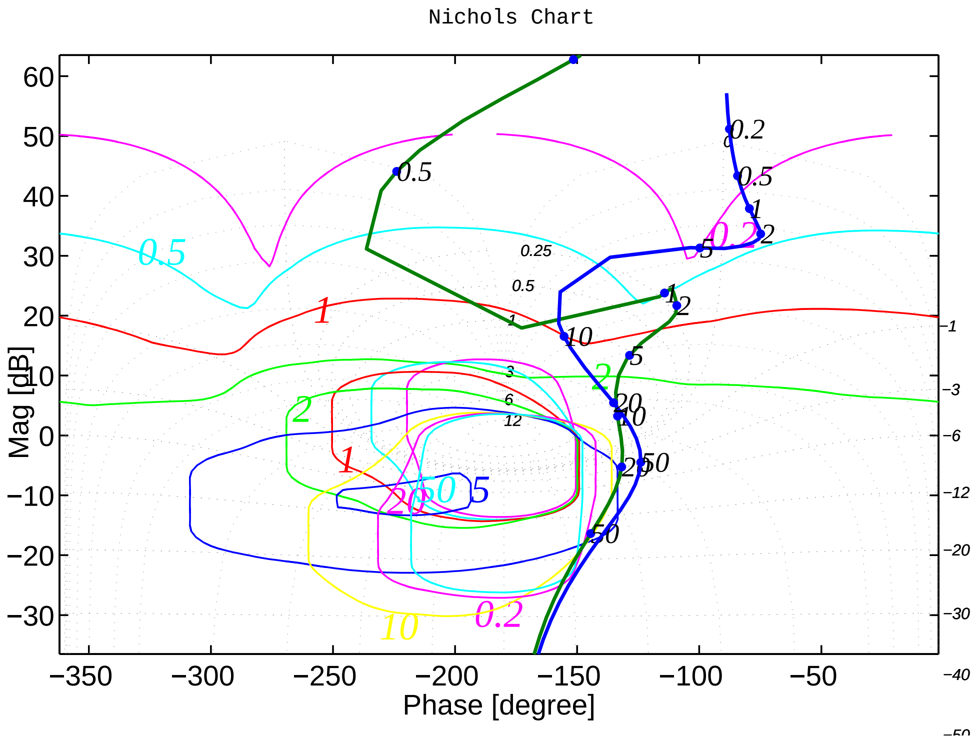

Figure 4.3. The green curve is the automated design and the blue one is the manual design.

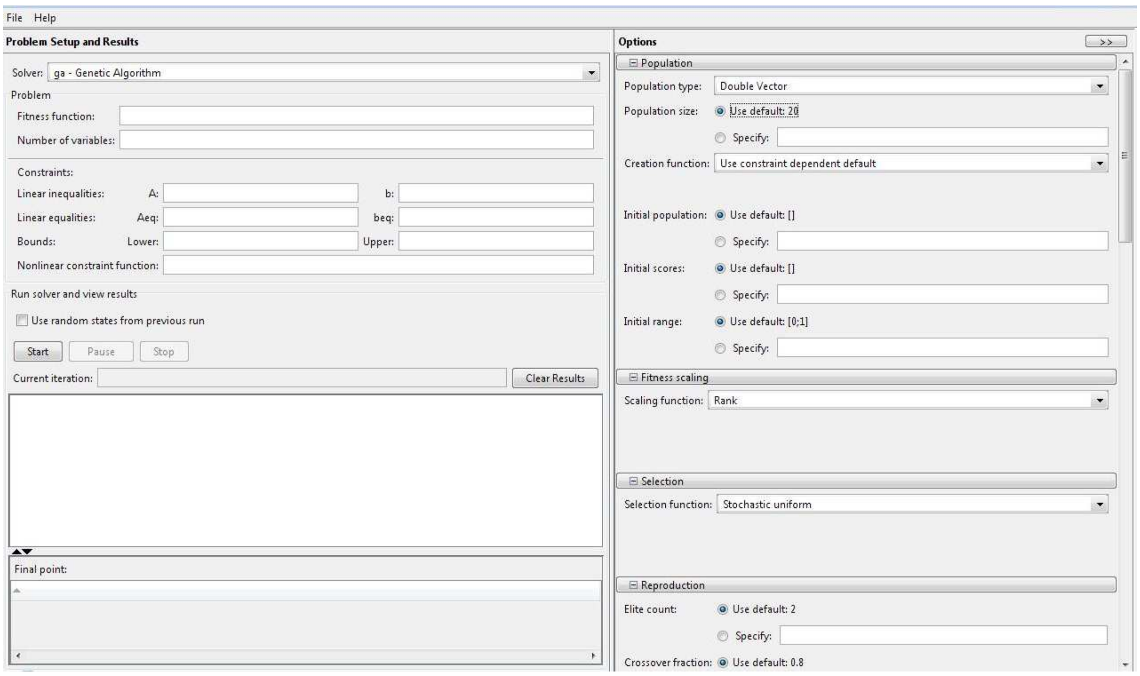

Figure 4.4. Matlab Toolbox GUI for Genetic Algorithm

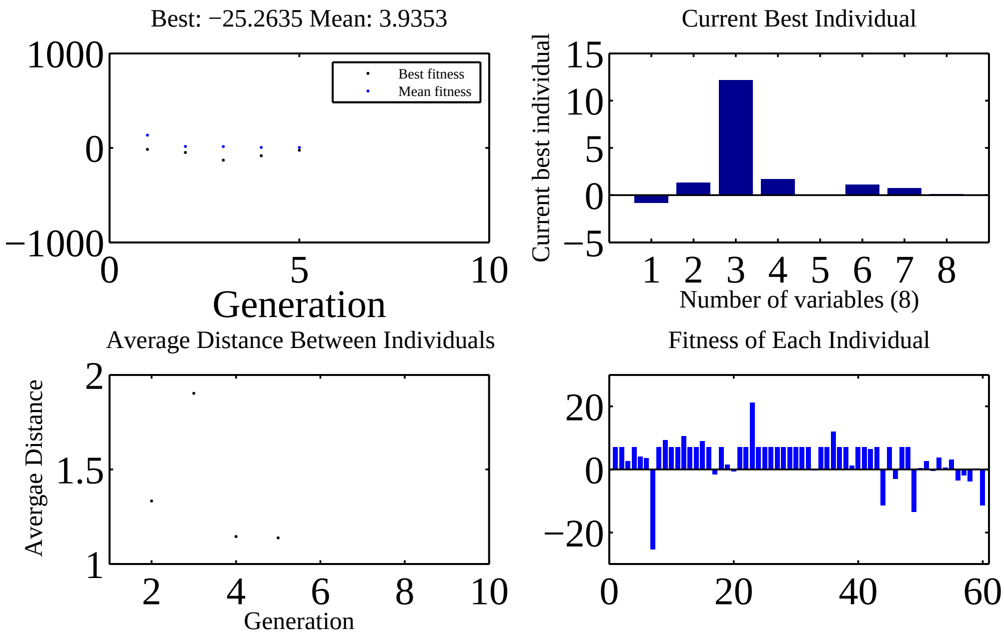

Figure 4.5. Different characteristics of a solution by genetic algorithm

4.2 Controller Synthesis for Nonlinear Systems

So far, we have shown how the genetic algorithm can be used to automate the loop-shaping in the Nichols diagram for LTI systems. Next step is to transform the nonlinear system into a LTI problem inorder to solve it with GA. The ultimate goal is to present a controller that not only works in a small region around the equilibrium points but also is robust to rather large deviations from equilibria.

Quantitative feedback theory is a useful method for design and analysis of uncertain linear systems. There are also methods to design QFT controller for nonlinear systems. However, a so-called equivalent linear model needs to be found then. To find this equivalent linear model is required a good knowledge about which input signal will yield the desired output. For an unstable system this is not a trivial task (cf. the issues in finding the best linear model [15]). Different methods of finding this equivalent linear model are discussed in [1]. In general the methods can be divided into a global and a local approach. Because of difficulties dealing with the global approach, the local approach is our interest here in this work.

For a nonlinear system:

$$\dot{x} = f(x,u) \tag{4.10}$$
$$y = h(x,u) \tag{4.11}$$

local linearization around $(\bar{x},\bar{u})$ gives:

$$\begin{aligned} \Delta\dot{x} &= A(\bar{x},\bar{u})\Delta x + B(\bar{x},\bar{u})\Delta u \\ &+ R(\bar{x},\bar{u}) \end{aligned} \tag{4.12}$$

where

$$A = \frac{\partial f}{\partial x}(\bar{x}, \bar{u}) \quad (4.13)$$

$$B = \frac{\partial f}{\partial u}(\bar{x}, \bar{u}) \quad (4.14)$$

and Δx and Δu are the deviations from $\bar{x}$ and $\bar{u}$ respectively.

In classic linearization this task is carried out only for stationary points which are derived from:

$$f(\bar{x}, \bar{u}) = 0 \quad (4.15)$$

$$\bar{y} = h(\bar{x}, \bar{u}) \quad (4.16)$$

which implies $R(\bar{x}, \bar{u}) = 0$. Linearization around non-stationary points introduces a constant $R(\bar{x}, \bar{u})$ and due to this term, properties such as stability are meaningless, because these points are reached only during transients. Equation (4.12) approximates the possibe transient dynamics of the nonlinear system when the trajectory is close to $(\bar{x}, \bar{u})$. In [10], it is shown with a phase plane analysis for a second order system, that the linearization around these non-stationary points approximates the flow of a nonlinear system well. Under the assumption that $R(\bar{x}, \bar{u})$ is small enough, R can also be considered as a process disturbance. The nonlinear system is then approximated by a family of LTI systems and a set of process disturbances.

Another method of linearization is dynamic linearization around some nominal trajectory, but the problem with this method is that the resulting system is a linear time variant system (LTV), which is not suitable for QFT analysis. The advantages of using a combination of linearization around stationary and non-stationary points over the other methods are:

- We have a better approximation of the nonlinear system compared to the classic method. If for a moment we forget about the nonlinear system and assume we have an uncertain linear model instead, then we have a more comprehensive description of the uncertainties in the system.
- If the non-stationary points are selected appropriately the resulting system might have a wider region of attraction for the trajectories of the system.
- The resulting system is a LTI sysem.

In this work the design is carried out according to the following steps:

1. Define the specifications and the cost function to be minimized.
2. Determine equilibria and linearize around them to get the initial set $\{P_{0i}(j\omega_k)\}$.

3. Determine the relevant non-stationary points in the desired operating window, linearize around them and add the new templates $\{P_i(j\omega_k)\}$.

4. Translate the specifications into Horowitz-Sidi bounds in the Nichols chart.

5. Decide the structure of the controller.

6. Run the optimization algorithm using GA.

7. Simulate the system with initial conditions in the desired operating window.

8. If the response becomes unstable, go back to the step 3 and repeat the algorithm again.

In this, we add work not only non-stationary points to the linearization points set, but we also try to find an effective way for selecting the best non-stationary points. For a second order system, this task can be carried out by using phase-plane analysis. It means that the stationary points are calculated first. Then the operating window in the state space is determined. In the next step we start to deviate from the stationary curve and select the non-stationary points in the desired operating window close to the stationary points curve. we continue this task iteratively until improvement ceases. It means that by adding non-stationary points to the linearization points set, the uncertainties in the plant increases in such a way that makes it impossible for genetic algorithm to find a feasible solution. In chapter 5, this method is applied to a second order unstable benchmark problem and the result shows how linearization around these non-stationary points improved the robustness of our controller. This method is not limited to only second order systems. For cases when the plant has an order larger than two, instead of using phase-plane analysis, the simulations are used to select the relevant non-stationary points. For stable systems one possible way to pick the non-stationary points is to simulate the open loop system for small steps in the input. In other words, if the input signal $u \in [\underline{u}, \overline{u}]$ then we can simulate the open loop for different input in the allowable interval and save the trajectories which are evolved in the desired operating window. Then we can use points on these trajectories as the non-stationary points. The proposed method is applied to a fourth order CSTR benchmark problem (see Chapter 6.). The method is not restricted to stable systems. For an unstable system with the order greater than two, at first a stabilizable controller can be designed for the main operating window. Then, we can follow the same procedure as the stable case.

Example 2

In [15], an example which compares the capability of two controllers in rejecting disturbances is presented. One of the controllers is designed for the *best linear model* and another one is designed for a linear model derived through classic linearization. Here, we applied the proposed method, based on QFT, on the same nonlinear system and compared the results with the two previous ones.

The nonlinear model is a simple Wiener system. The Wiener system is given by the series connection of the linear system $P(s) = \frac{1}{(s+1)^3}$ followed by a static nonlinearity $f(x) = x + x^3$. The control signal is limited to be $u \in [-2, 2]$.

First, the nonlinear system is linearized around stationary and relevant non-stationary points. The effect of linearizing at different points is only in the DC gain of the system. It can be interpreted that we can replace the nonlinear system with an uncertain linear system with uncertainty only in the DC gain. Then, in the next step the servo and disturbance rejection specifications are translated to the frequency domain. Finally, Genetic algorithm is used to synthesize a PID controller for this system such that the high frequency gain of the controller is minimized.

The two controllers presented in [15], are the controller designed for the best linear model

$$u = (0.2 + \frac{0.12}{s})e \tag{4.17}$$

and the controller derived from local linearization:

$$u = (1 + \frac{0.6}{s})e \tag{4.18}$$

The controller designed using QFT and linearization at stationary and non-stationary points is

$$u = (\frac{0.397s^2 + 0.562s + 0.334}{s})e \tag{4.19}$$

In this example the nonlinear system is a third order system, Hence we cannot use phase plane analysis to decide the selection of non-stationary points. One possible way is to use simulation. Basically, we start with designing a controller only for the desired operating point. Then we define the desired operating window and perturb the system with different step signals as the output disturbances. For the trajectories that stay in the operating window, we can collect the data and add points from trajectories to the set of non-stationary points. As can be seen in Fig. 4.7 and 4.8 adding non-stationary linearization points improves the robustness and performance considerably. The derived PID controller also compares well to the controller based on the best linear model [15].

In [12] different optimization criteria for tuning the PID controllers are presented. To improve the performance in the foregoing example we can minimize ouput disturbance rejection criterion instead of the high frequency gain of the controller. i.e,

$$J_{LF} = \|S(s)\|_\infty \tag{4.20}$$

$$\tag{4.21}$$

where $S(s)$ is the sensitivity function.

The foregoing criterion is a measure of the system's ability to handle LF output disturbances. The controller derived for this criterion is:

$$u = (\frac{0.588s^2 + 0.614s + 0.336}{s})e \tag{4.22}$$

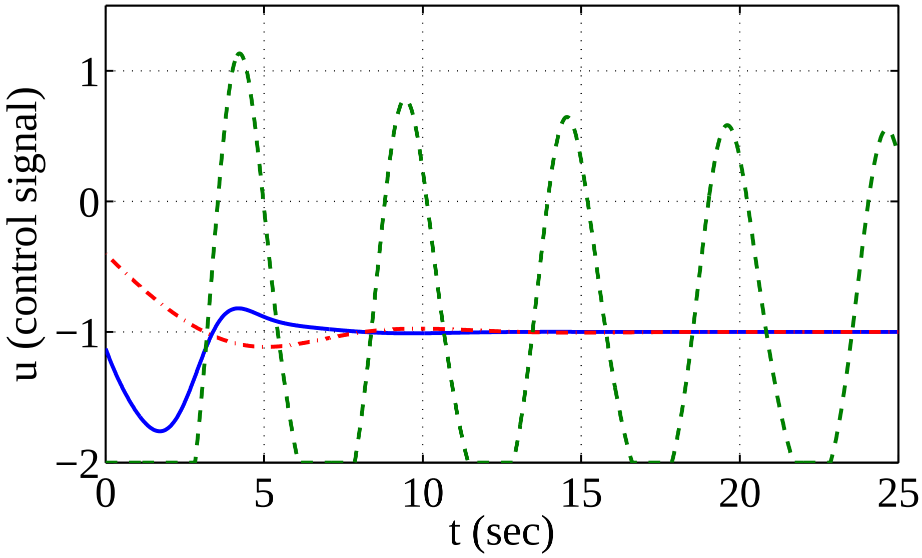

Figure 4.6. Closed loop responses to a step with unit height as output disturbance. The solid curve (blue) corresponds to the QFT controller, the dashed curve (red) is the response corresponding to the best linear model, and the dashed-dot curve (green) is the closed loop response with the controller based on local linearization.

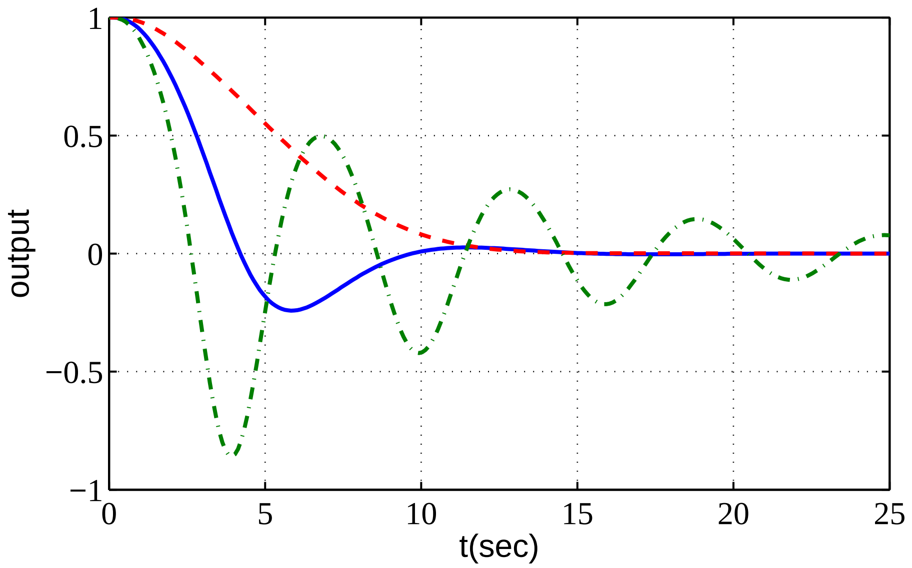

Figure 4.7. Closed loop responses to a step with unit height as output disturbance. The solid curve (blue) corresponds to the QFT controller, the dashed-dot curve (red) is the response corresponding to the best linear model, and the dashed curve (green) is the closed loop response with the controller based on local linearization.

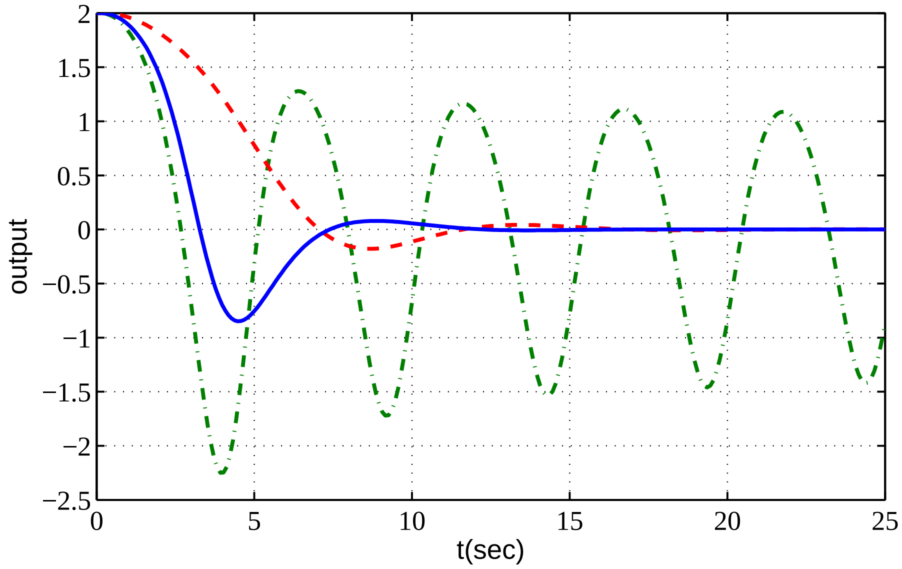

Figure 4.8. Closed loop responses to the step with height two as the output disturbance. The solid curve (blue) corresponds to the QFT controller, the dashed curve (red) is the response corresponding to the best linear model and the dashed-dot curve (green) is the closed loop response with the controller based on local linearization.

Criterion	$H_\infty norm$
$J_{HF} = K_D$	1.17
$J_{LF} = \|S(s)\|_\infty$	1.11
Best linear Model	1.14

Table 4.1. H_∞ norm for different optimization criterion.

Fig. 4.9 and 4.10 illustrate the output and control activity for different controllers to the output disturbance. As can be seen, with this new criterion, we could further improve the performance of the system.

One might ask if it is fair to compare a PI controller with a PID controller design. To answer this question, we should note that the idea here is not to compare our PID design with the PI design in [15]. With this comparison we would like to show that, the outcome from the presented method is as satisfactory as the result from the method given in [15].

To show the importance of the optimization criterion in our design, the Wiener system is simulated for a process disturbance instead of the output disturbance this time. Another controller is also designed to reject the LF process disturbance. The following criterion is used in the optimization algorithm:

$$J_{LF} = \|S(s)P(s)\|_\infty \tag{4.23}$$

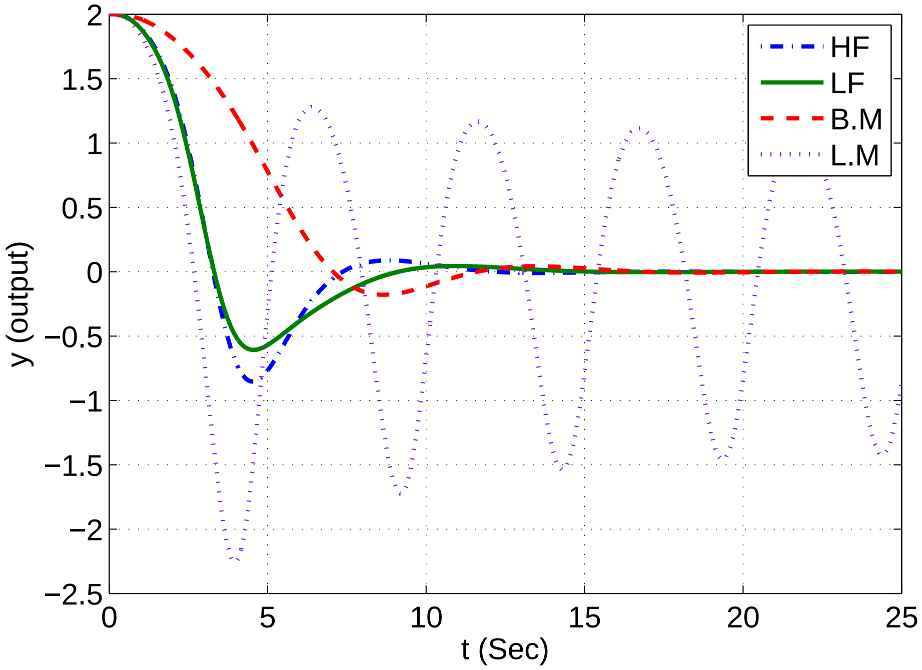

Figure 4.9. The dashed curve (red) is the closed loop response to the step with height two as the output disturbance for the best linear model, the dashed-dot curve (blue) is the response for the QFT designed controller with the high frequency gain of the controller as the cost function to be minimized, the solid curve (green) is the output for the QFT controller with LF output disturbance rejection as the objective function and eventually the dotted curve(purple) is the controller for the local linear model.

where $S(s)$ is the sensitivity function and $P(s)$ is the plant transfer function. The derived controller is then

$$u = (\frac{0.684s^2 + 0.584s + 0.526}{s})e \tag{4.24}$$

Fig. 4.11 and 4.12 show that the designed controller for the LF output disturbance rejection might not have the same behavior for other purposes. In other word, we observe that although the designed controller for the best linear model shows a good behavior in rejecting the output disturbance but its response is deteriorated when a process disturbance is applied to the system. The presented method in this work not only improve robustness but can be easily optimized with our algorithm with respect to different desired criterion. Hence, robustness, simplicity and flexibility are of the characteristics of the proposed algorithm.

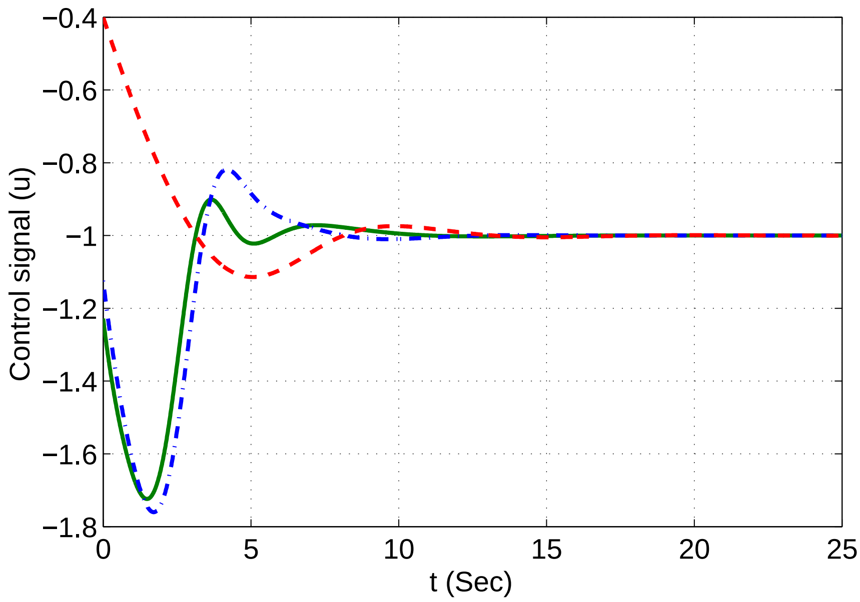

Figure 4.10. The dashed curve (red) is the control signal for the controller designed for the best linear model, The solid curve (green) is the control signal for the PID controller design to minimize the effect of LF output disturbance and the dashed-dot curve (blue) is the control signal for the PID controller with the high frequency gain of the controller as the cost function

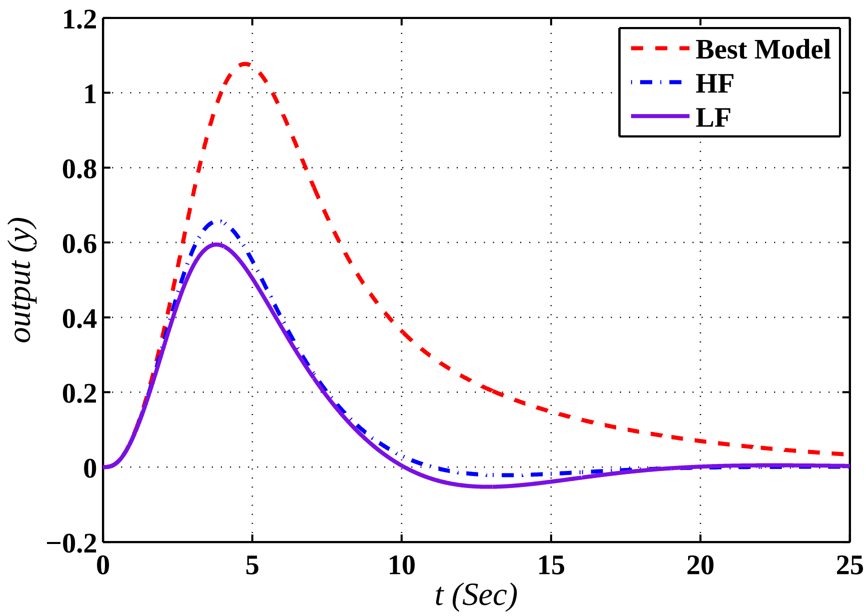

Figure 4.11. Closed loop responses to a step with height one as the process disturbance. The solid curve (purple) corresponds to the QFT controller, the dashed curve (red) is the response corresponding to the best linear model, and the dashed-dot curve (blue) is the closed loop response with the controller based on local linearization.

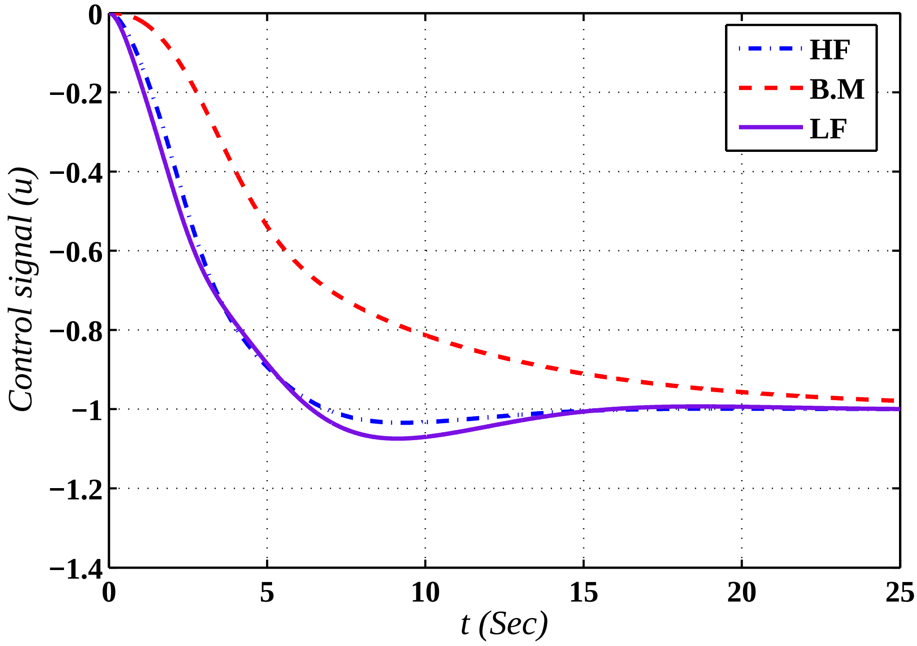

Figure 4.12. The solid curve (purple) is the control signal corresponds to the QFT controller which is designed with respect to process disturbance rejection criterion, the dashed curve (red) is the control signal corresponding to the best linear model, and the dashed-dot curve (blue) is the control signal for the controller which has the minimum high frequency gain.

5 BENCHMARK PROBLEM 1

5.1 MODELING

In order to further evaluate the performance and characteristics of our method, we have selected a Bioreactor Benchmark problem as the plant to be controlled due to its interesting characteristic [13]. Although this process is rather simple and has only two state variables, it is difficult to control due to strong nonlinearity. The bioreactor is a continuous flow stirred tank reactor (CSTR) with water and cells (e.g., yeast or bacteria) which consumes nutrients ('substrate') and produce products (both desired and undesired) and more cells. The stated control problem is tracking a desired amount of cell mass.

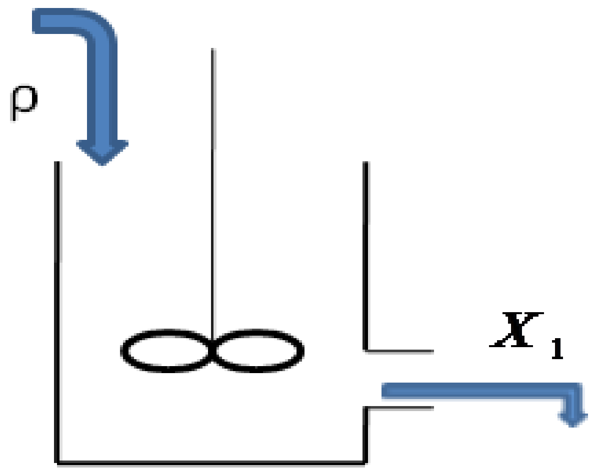

Figure 5.1. Bioreactor with ρ as input and x_1 as output

The state space equations of the plant are:

$$\dot{X}_1 = -X_1\rho + X_1(1 - X_2)e^{X_2/\gamma} \tag{5.1}$$

$$\dot{X}_2 = -X_2\rho + X_1(1 - X_2)e^{X_2/\gamma}\frac{1+\rho}{1+\rho-X_2} \tag{5.2}$$

where X_1 is dimensionless cell mass and X_2 is nutrient conversion, defined as $(S_F - S)/S_F$, where S_F is the concentration of nutrient in the feed to the reactor and S is the concentration (of nutrient) in the reactor. The constraints on the state variables are:

$$\Omega : 0 \leq X_1, X_2 \leq 1$$

ρ is the control signal, which is the flow rate through the reactor ($0 \leq \rho \leq 2$). To have a better understanding of these equations we will explain each term below:

The first term ($-X_1\rho$) in (5.1) is the amount of cells leaving the tank and the second term $X_1(1-X_2)e^{\frac{X_2}{\rho}}$ represents cell growth in the tank. The rate is proportional to the current amount of cell and depends nonlinearly on the amount of X_2. In (5.2), the first term ($-X_2$) is the amount of nutrient leaving the tank and the second term

$$X_1(1-X_2)e^{X_2/\gamma}\frac{1+\beta}{1+\beta-X_2}$$

is the rate by which the nutrient is metabolized. The constants β and γ determine the rate of cell growth and nutrient consumption. From the equations we may also deduce that cell growth in moderate nutrient concentrations is faster than at very high or low conversion.

This system is a challenging benchmark because it is highly nonlinear and for some values of ρ limit cycle is unavoidable, see Fig. 5.2. The system is also unstable, as can be seen in the phase portrait in Fig. 5.3. It can be noted that the system has one stable and one unstable eigenvalue in this area so the equilibrium points are saddle points. The system response is very sensitive to parameter variation. It means that a small error in the model can cause a large change in the control problem.

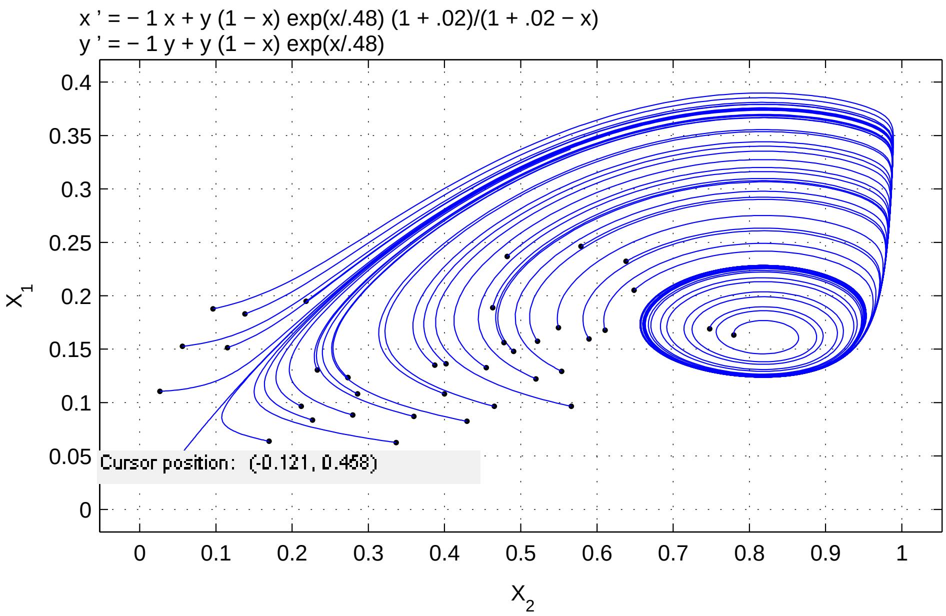

Figure 5.2. Limit cycle for $\rho = 1$

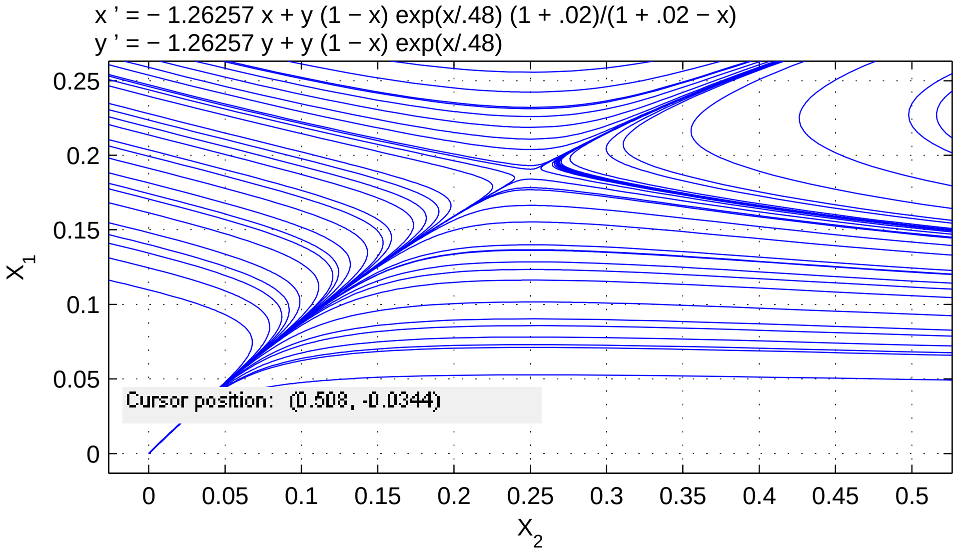

Figure 5.3. saddle point equilibrium for $\rho = 1.26$

5.2 CONTROL DESIGN

According to our design procedure we begin by linearizing the system around its stationary curve. To obtain stationary points we need to solve (5.1) and (5.2) at a steady state, which gives

$$\dot{X}_1 = 0 \Rightarrow \rho_{ss} = (1 - X_2)^{X_2/\rho} \quad (5.3)$$

$$\dot{X}_2 = 0 \Rightarrow X_2 = 0 \text{ or } X_2 = x_1 \frac{1+\beta}{1+\beta-X_2} \quad (5.4)$$

From the plot in Fig. 5.4, we observe that in a steady state we cannot achieve any value larger than $\frac{1+\beta}{4} = 0.255$ for cell mass. As mentioned, the main goal is to track a desired cell mass (X_1). We limit our design to $0 \leq X_1 \leq 0.255$ and $0 \leq X_2 \leq 0.51$ (This counts as our operating window). The nonlinear plant is linearized around stationary points in this window and one of the linearized plant model is selected as the nominal one.

As the QFT design should be carried out in frequency domain, all the servo specifications are translated from time domain into frequency domain. An output disturbance rejection constraint is also applied to the system in the form of a constraint on the sensitivity function: $\|S\| \leq 3$. The Matlab toolbox Qsyn [7] is used to calculate the corresponding Horowitz-Sidi bounds for these specifications. In Fig. 5.5 the nominal plant together with the Horowitz-Sidi bounds is portrayed for different frequencies. Clearly, the nominal plant is unstable and violates all the Horowitz-Sidi bounds. We also observe that the gain uncertainty of the template, especially for frequencies smaller than the bandwidth frequency, is larger than the tolerance specification (see Fig. 5.6). This means that the problem cannot be solved in open-loop with feedforward. Feedback control is needed to reduce the uncertainty within the

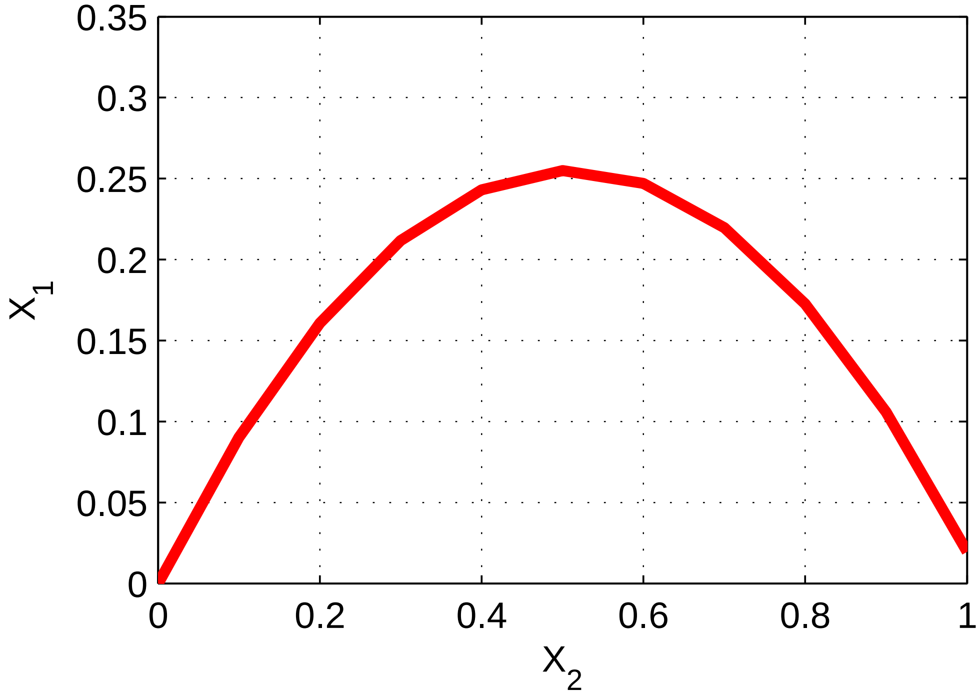

Figure 5.4. Stationary points for the benchmark bioreactor

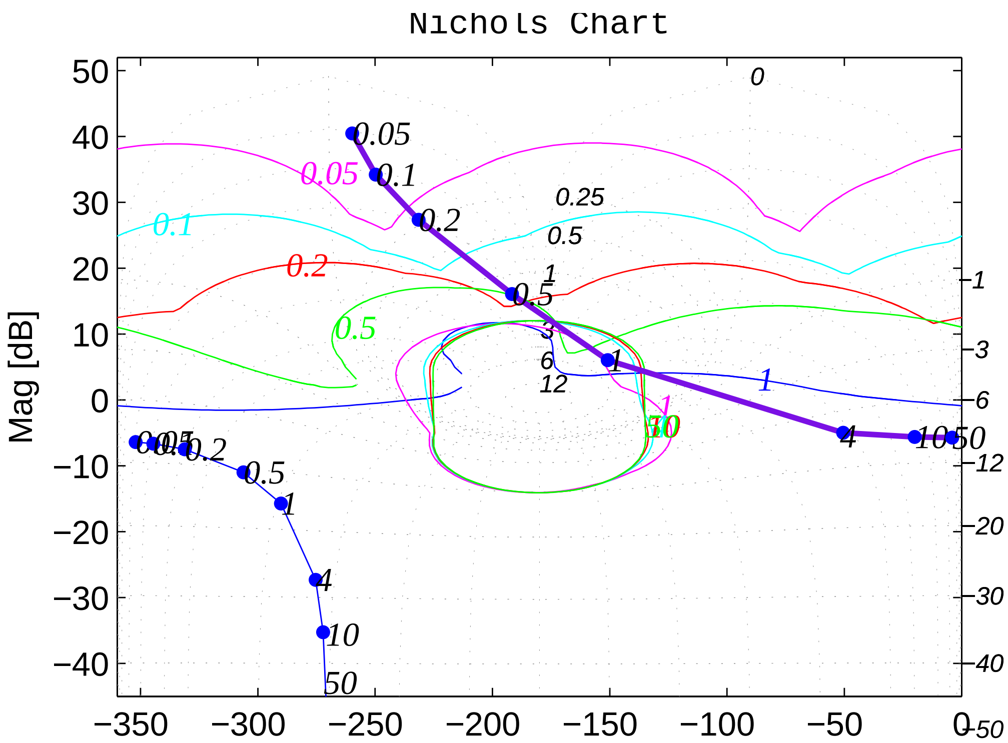

Figure 5.5. The blue curve to the left is nominal loop transfer function before design and the pepper thick curve to the right is the loop transfer function after design for only stationary points

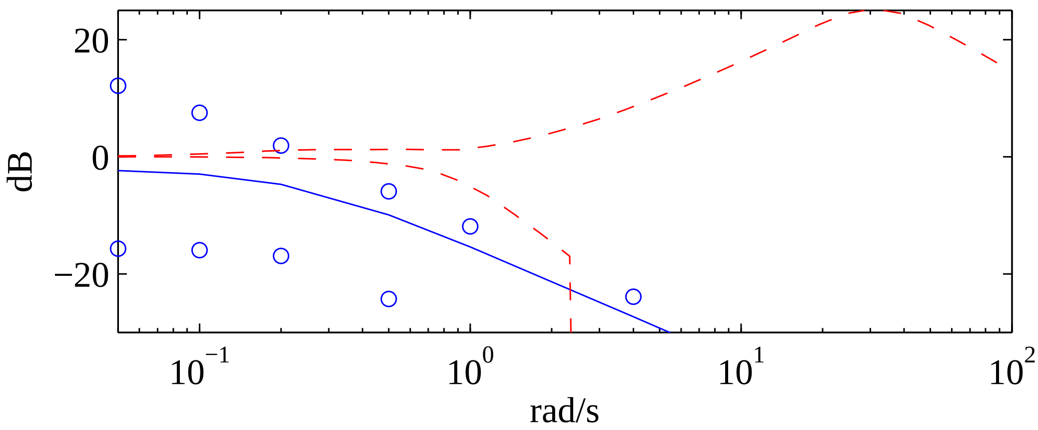

Figure 5.6. The blue curve is nominal plant. Small circles, show the uncertainties defined by the template. The red envelope is servo specification.

acceptable envelope. The idea here is to automatically design a PID controller in such a way that the closed loop system becomes stable and fulfill the specifications for all frequencies. Preferably it should also be robust to initial error or deviations from a steady state. The controller has the the following (ideal) transfer function:

$$G(s) = \frac{K_D s^2 + K_P s + K_I}{s} \tag{5.5}$$

The optimization variables are $\theta = [K_D, K_P, K_I]^\top$, and the objective function we minimize is K_D (high frequency gain of controller) subject to the following specifications:

- Servo specification

$$a(\omega) \leq \left| \frac{F(j\omega)G(j\omega)P(j\omega)}{1 + G(j\omega)P(j\omega)} \right| \leq b(\omega) \tag{5.6}$$

- Sensitivity specification

$$\left| \frac{1}{1 + G(j\omega)P(j\omega)} \right| \leq 3 \tag{5.7}$$

Fig. 5.5 shows the nominal loop transfer function after design of the PID controller. From the plot we can see that the system becomes stable and the nominal loop transfer function satisfies the specifications for all frequencies. However, we cannot claim that it has the desired performance on the original nonlinear system unless we test our design through simulation. When we simulate the system from an initial condition in a region close enough to the stationary points, the system response is satisfactory but for larger perturbations from equilibrium points in initial condition

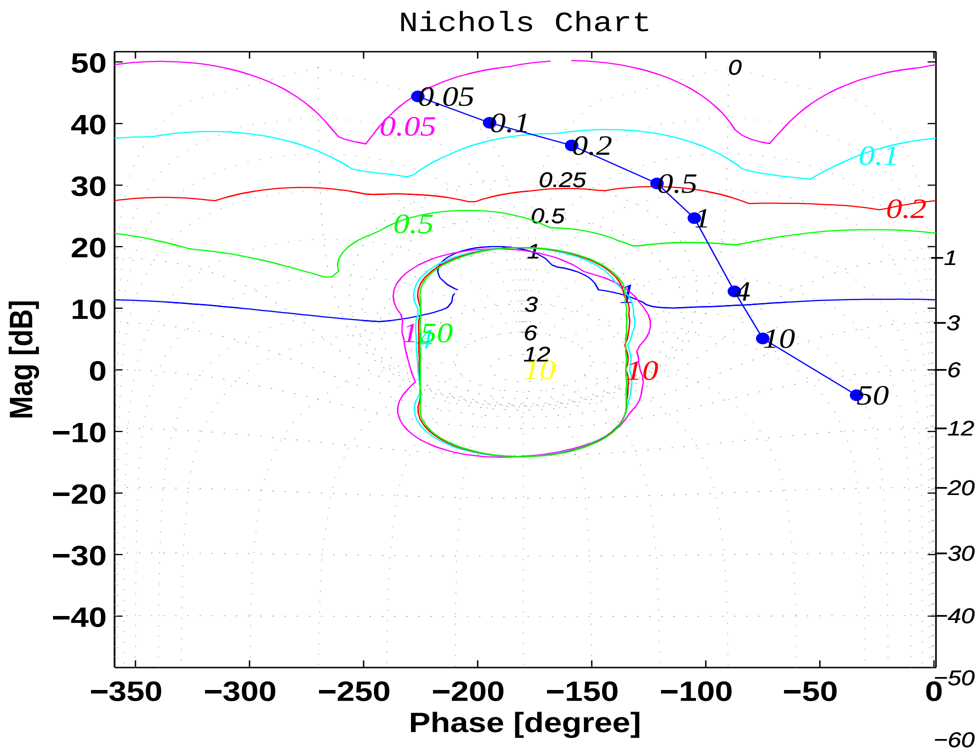

Figure 5.7. The blue curve is nominal loop transfer function after designing the controller

the system becomes unstable. These new non-stationary points are then added to the set of linearization points iteratively. This imposes tougher boundaries on the nominal loop transfer function $L_0(j\omega_k)$ in the Nichols chart (see Fig. 5.7). We should note that by selecting these new non-stationary points we aimed to make our design more robust. The results shows the improvement in robustness but one should have in mind that there is limitation for selecting these non-stationary points. One possible limitation is that if we pick these non-stationary points too far from the equilibrium curve (or operating points) the uncertainty in the plant will increase such that might make it impossible for the optimization algorithm to find a feasible solution. One good way to overcome this problem might be to start with a reasonable size of operating window in the state-space and try to design for that and then iteratively enlarge the window. The problem is then solved with genetic algorithm once more. We did not consider any initial guess for the optimization algorithm and the population size is set to 50. After five iteration the algorithm found the solution. This process took almost 20 min. The simulation results for this new controller, the former one and a sliding mode solution [4]is presented in the next section. In Fig. 5.8 the gain extent of the closed loop system together with the uncertainty in the template is portrayed. We see that after designing the feedback the uncertainty is reduced to an acceptable level. We also conclude that there appears to be no need to design a prefilter $F(s)$.

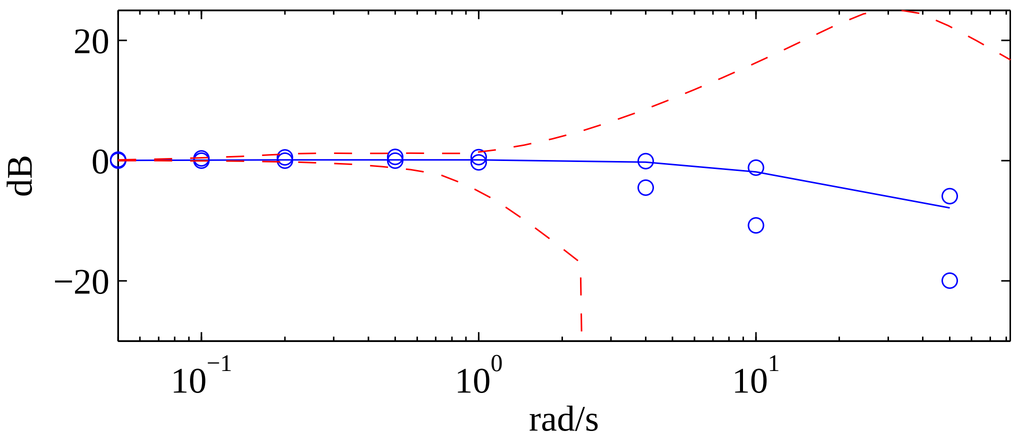

Figure 5.8. The blue curve is nominal plant, the small circles show the uncertainty in the template, and the red envelope is the servo specification.

5.3 Simulation results

Simulations were carried out in Simulink for different initial values and different square waves as reference signal. First of all the two controllers, one derived from linearization around only stationary points and another one from linearization around both stationary and non-stationary points, are simulated for two different initial values. For an initial condition close enough to the stationary curve both controllers work, but as can be seen in Fig. 5.9 for an initial condition $x_1(0) = 0.15$ and $x_2(0) = 0.3$ the controller designed for stationary points only results in large overshoots. In Fig.14 we perturbed the system harder by giving an initial condition $x_1(0) = 0.09$ and $x_2(0) = 0.4$. For this rather large deviation from the stationary curve the first controller gives an unstable response but the second one is more robust and shows a satisfactory response. In [4] a sliding mode controller is designed for this system. In Fig. 5.11 and 5.12 that sliding mode controller is compared to the PID controller designed with the QFT method. The minimum and maximum values of the square wave are close to the maximum values that the system can reach. The systems are simulated for two different initial values that cause large initial errors in the control. As can be seen in Fig. 5.11 and 5.12 the PID controller response has an acceptable response though a significant overshoot. However, from an implementation point of view the PID controller is clearly preferable.

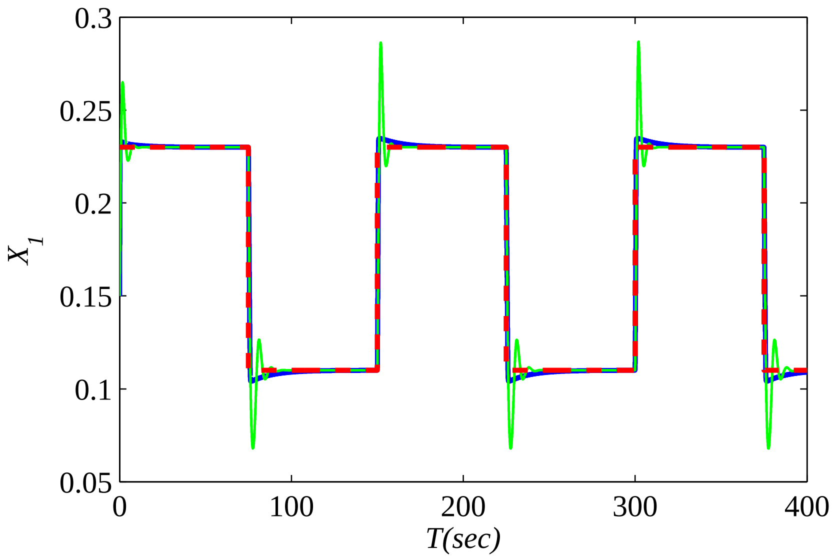

Figure 5.9. Red dashed line is reference signal, blue curve is for controller designed using non-stationary points and the green response is the response from controller designed for stationary points only.

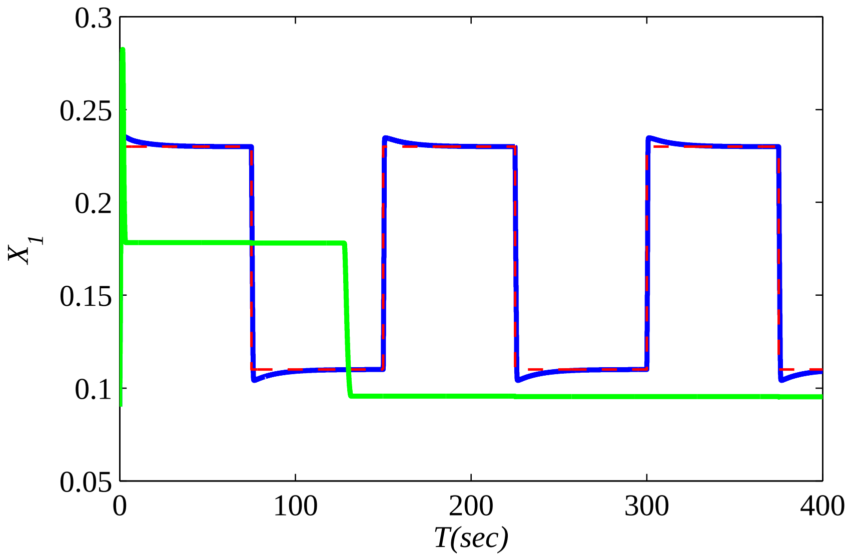

Figure 5.10. Red dashed is reference signal, blue curve is for the controller designed with QFT for non-stationary points and the green response is the response for the controller designed for stationary points only.

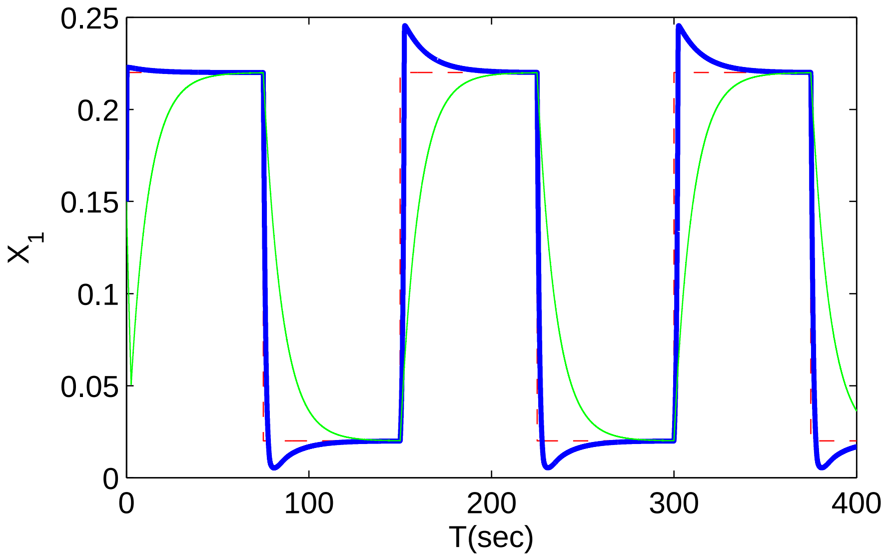

Figure 5.11. Red dashed line is the reference signal, blue is for the PID controller and the green one is for the sliding mode controller response. $x_1(0) = 0.15, x_2(0) = 0.2$

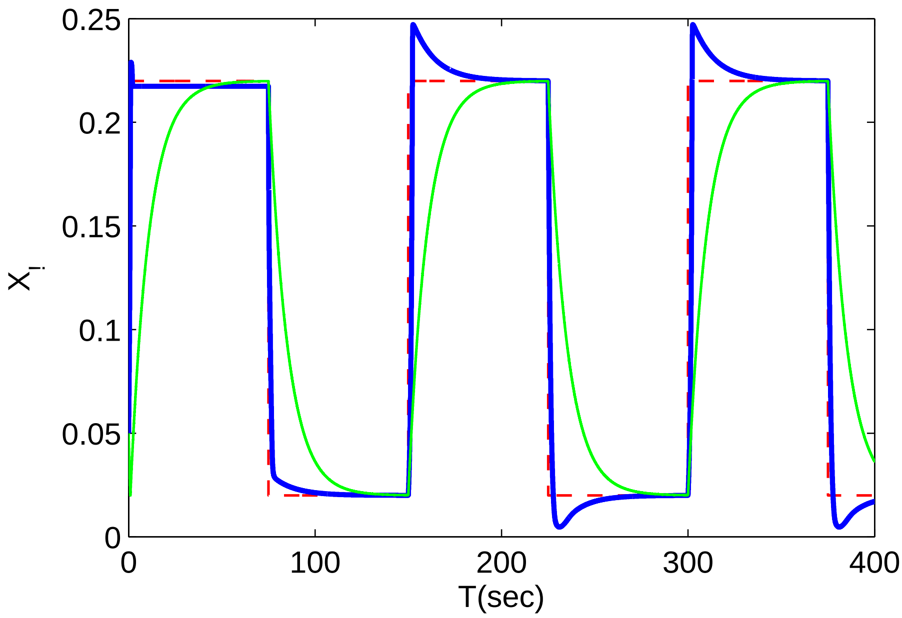

Figure 5.12. Red dash line is the reference signal, the blue one is the QFT controller and the green one is the sliding mode controller response. $x_1(0) = 0.05, x_2(0) = 0.3$

6 BENCHMARK PROBLEM 2

6.1 MODELING

The example which is considered in this chapter is a fourth order continuous stirred tank reactor (CSTR). In this process the desired product is cyclopentenol (substance B) which is produced from cyclopentadiene (Substance A). In addition to the desired product two other undesired products, dicyclopentadience (D) and cyclopentanediol (C), are produced as shown below.

$$A \xrightarrow{K_1} B \xrightarrow{K_2} C \tag{6.1}$$

$$2A \xrightarrow{K_3} D \tag{6.2}$$

The state space equations of the systems are

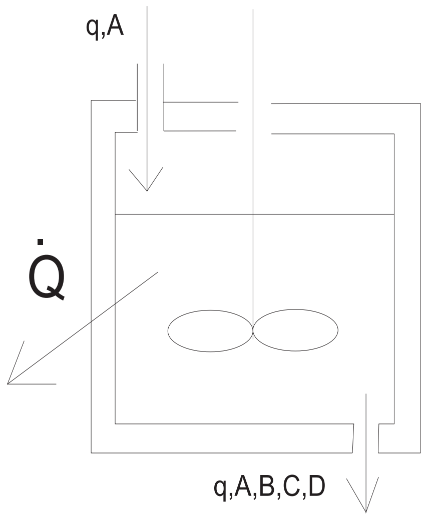

Figure 6.1. Bioreactor with q as input and c_B as output

$$\dot{c}_A = \frac{q}{V_R}(c_{A0} - c_A) - K_1 c_A - K_3 c_A^2 \tag{6.3}$$

$$\dot{c}_B = -\frac{q}{V_R} c_B + K_1 c_A - K_2 c_B \tag{6.4}$$

$$\begin{aligned} \dot{T} &= \frac{q}{V_R}(T_0 - T) + \frac{K_W A_R}{\rho C_p V_R}(T_c - T) - \frac{1}{\rho C_p}(K_1 c_A \Delta H_{R,AB} \\ &+ K_2 c_B \Delta H_{R,BC} + K_3 c_A^2 \Delta H_{R,AD}) \end{aligned} \tag{6.5}$$

$$\dot{T}_c = \frac{1}{m_c C_{pc}}(\dot{Q} + K_W A_R (T - T_c)) \tag{6.6}$$

Model Parameters	*Main operating point*
$K0_{1,2} = (1.287 \pm 0.04) \cdot 10^{12} h^{-1}$	$c_{A\|s} = 1.235 mol/l$
$K0_3 = (9.043 \pm 0.27) \cdot 10^9 1/(molAh)$	$c_{B\|s} = 0.9 mol/l$
$E_{A1,2}/R = 9758.3K \pm \Delta$	$T_{\|s} = 134.14°C$
$E_{A3}/R = 8560.0K \pm \Delta$	$T_{c\|s} = 128.95°C$
$\Delta H_R^{AB} = 4.2 \pm 2.36 kJ/molA$	$q = 188.3h^{-1}$
$\Delta H_R^{BC} = -11.0 \pm 1.92 kJ/molB$	$\dot{Q}_{\|s} = -4495.7 kJ/h$
$\Delta H_R^{AD} = -41.85 \pm 1.41 kJ/molA$	$c_{a0\|s}$ mol/l
$\rho = 0.9342 kg/l$	$C_p = 3.01 kJ/kgK$
$C_{pk} = 2kJ/kgK$	$A_R = 0.215m^2$
$V_R = 10.01$	$m_k = 5.0kg$
$T_0 = 130°C$	$K_W = 4032$

Table 6.1. H_∞ norm for different optimization criterion.

where K_1,K_2 and K_3 are the reaction rate coefficient given by

$$K_i = K_{i0} e^{E_i/(T/°C+273.15)}, i = 1, 2, 3$$

c_A and c_B are the concentrations of the substances A and B respectively. T is the temperature of the bioreactor and T_c is the temperature in the cooling jacket. $\dot{Q}$ is the heat flow which is removed from the coolant and finally q is the feed flow to the reactor containing substance A with the concentration c_{A0}. The values for the constant parameters and the main steady state operating point are given in Table 6.1. Generally, this control problem is treated as a MIMO system with concentration of substance B and temperature in the reactor (T) as the outputs, and inflow q and heat flow $\dot{Q}$ as the control input [3]. However, we consider $\dot{Q}$ constant (steady state value) and try to track the desired set point concentration c_B using q only. Hence, the problem is translated to a single input single output problem with q as the input an c_B as the output. The desired operating window is considered in a suboptimal region relatively close to the maximum yield , i.e;

$$0.8 \le c_B \le 1$$

The stationary curve for the two states (c_A and c_B) is illustrated in Fig. 6.2. (Q,T and T_c are kept at their steady state values). Although this process shows input multiplicity for some values of c_A and c_B but there is no need to be concern about that in our desired operating window.

6.2 Controller design

In [15] the nonlinear system is approximated by a best linear model. With simulation of the open loop system, they have shown that their best linear model is superior to the locally linearized model in approximating the behaviour of the nonlinear system. Finally, in their work they have designed an H_∞ controller ($C(s)$)

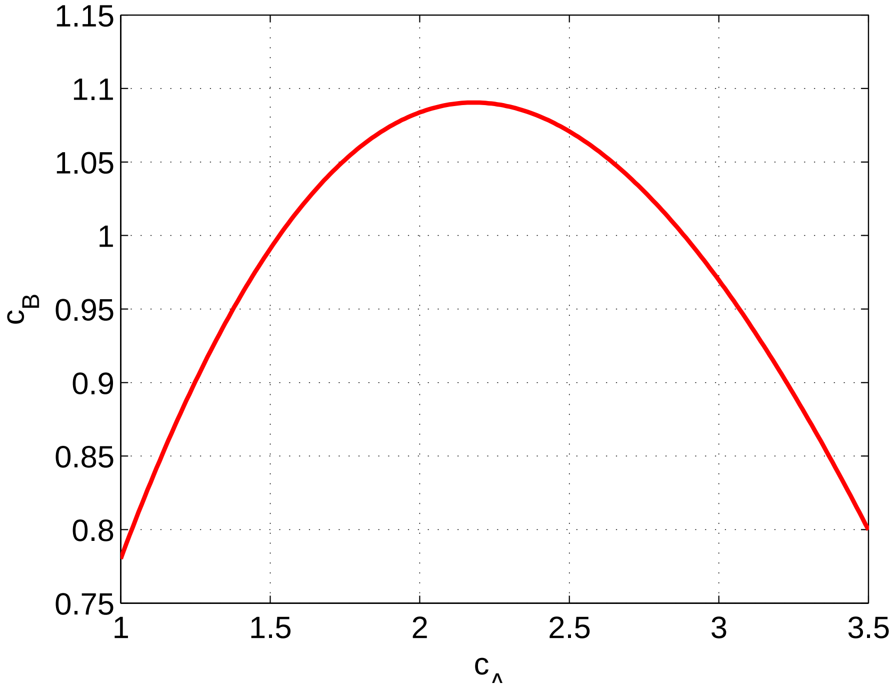

Figure 6.2. Stationary points

which is followed by a PI controller to track the desired set points for concentration of substance B. The control signal is limited to $88.3 \leq q \leq 288.3$ and the controller is as below:

$$C(s) = \frac{1.040 \cdot 10^3 s^3 + 1.020 \cdot 10^6 s^2 + 4.274 \cdot 10^7 s + 3.563 \cdot 10^8}{s^4 + 1.024 \cdot 10^3 s^3 + 8.289 \cdot 10^4 s^2 + 9.958 \cdot 10^5 s + 2.683 \cdot 10^6} \quad (6.7)$$

$$u = (1 + \frac{25}{s})C(s)e \quad (6.8)$$

where e is the control error and u is the controller output.

In our work we tried to find a lower order controller which shows similar behavior as their H_∞ controller designed for the Best linear model. A similar method that is shown in the two previous chapter is used to design a simple linear controller. The design procedure can be summarized by the following steps:

- First the nonlinear system is linearized around its stationary points for $c_B \in [0.8, 1]$.
- To add non-stationary points to the uncertainty set, the open loop system is simulated for small steps within $q \in [88, 288]$ (see Fig. 6.3). The trajectories of the responses are saved and used as the non-stationary points in the linearization.
- The servo and sensitivity specifications are translated into the frequency domain and HS-bounds are generated for the Nichols diagram. The sensitivity

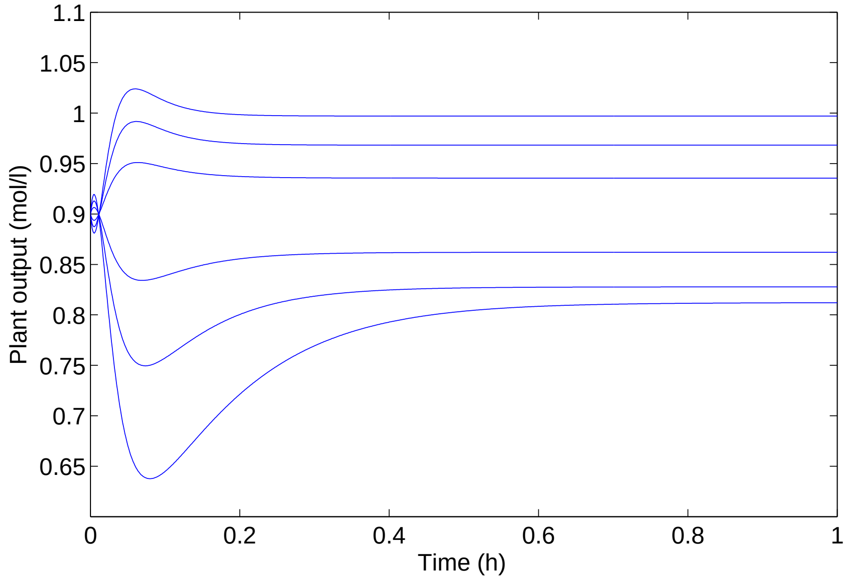

Figure 6.3. Step responses of open loop system for different step sizes.

function should be $\|S(s)\| \leq 2.5$ and for the servo specification a settling time $T_s = 2h$ and maximum overshoot $|M_T| \leq 10$ percent are specified. As can be seen in Fig. 6.4 and 6.5 The results of adding these non-stationary points to the set produces a bit tougher Horowitz-Sidi bounds in the Nichols diagram (c.f.Fig. 6.4 and 6.5.

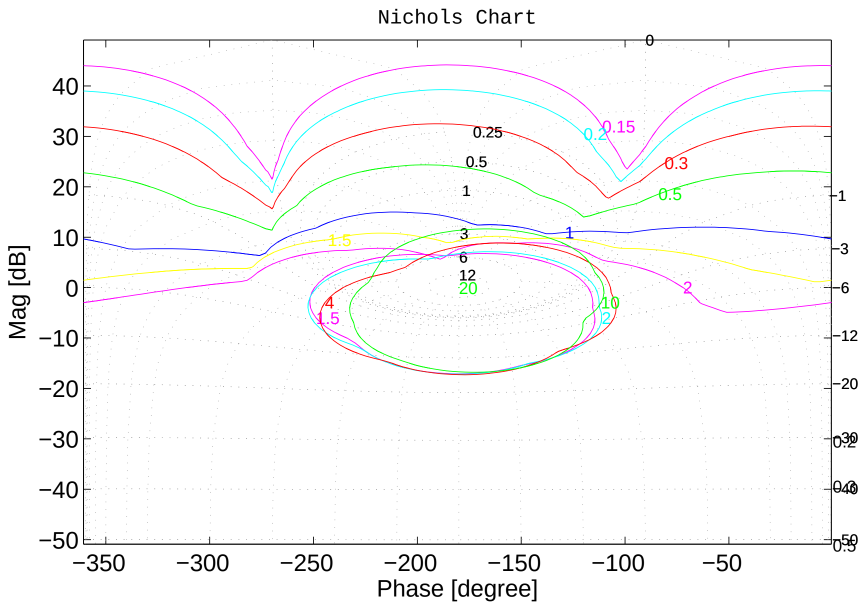

Figure 6.4. Horowiz-Sidi Bounds for templates derived from only stationary points

- Genetic algorithm is used to do loop-shaping automatically for a fixed structure

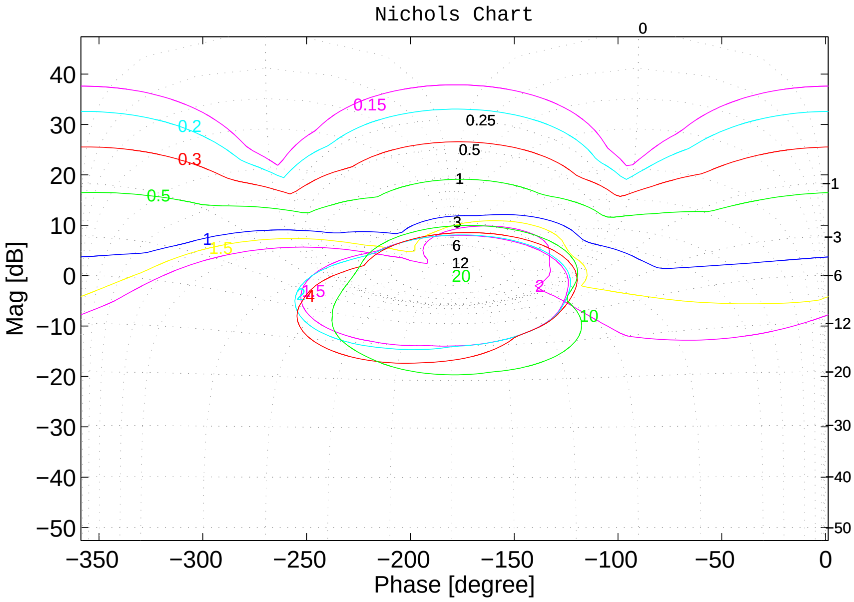

Figure 6.5. Horowiz-Sidi Bounds for templates derived from stationary and non-stationary points

fourth order controller.

$$G(s) = \frac{\theta(1)s^2 + \theta(2)s + \theta(3)}{s(s^3 + \theta(4)s^2 + \theta(5)s + \theta(6))}, \tag{6.9}$$

where x is the optimization variable which is determined by the optimization algorithm. The objective function of optimization algorithm is $J = \theta(1)$ which is the high frequency gain of the controller.

The designed controller has the transfer function as below:

$$G(s) = \frac{3358.454s^2 + 63076.673s + 87689.366}{s(s^3 + 14.323s^2 + 21.262s + 28.202)} \tag{6.10}$$

The simulation results for the controller designed using QFT and the controller designed based on H_∞ are illustrated in Fig. 6.6 and 6.7. It can be seen that the controllers have a very similar response. From the implementation point of view, using a fourth order controller is an advantageous over using a fifth order controller. Here we used the high frequency gain of the controller as the objective function to be minimized as for the Wiener problem. Furthermore, it is possible to use different criteria for different objectives. This possibility increases the flexibility of the method and in contrast to the Best linear model method we do not need to do complex calculations to find the best linear model. In summary, we can mention the features of this method in general and compared to other methods.

- Control of nonlinear plant using linear controller.

- Using our automated controller synthesis makes the design procedure very simple.
- Selection of non-stationary points carried out through simulation automatically.
- The designer has control over the structure and order of the controller. In result of this advantage, we succeed to design a controller with lower order than the H_∞ controller in [15].
- By linearizing around a set of stationary and non-stationary points, we improved the approximation of the nonlinear system. Hence, there is no need to calculate the best linear model with complicated calculation.
- Contrary to the proposed method in [15], our method is neither limited to stable plants nor to plants with time-delay.

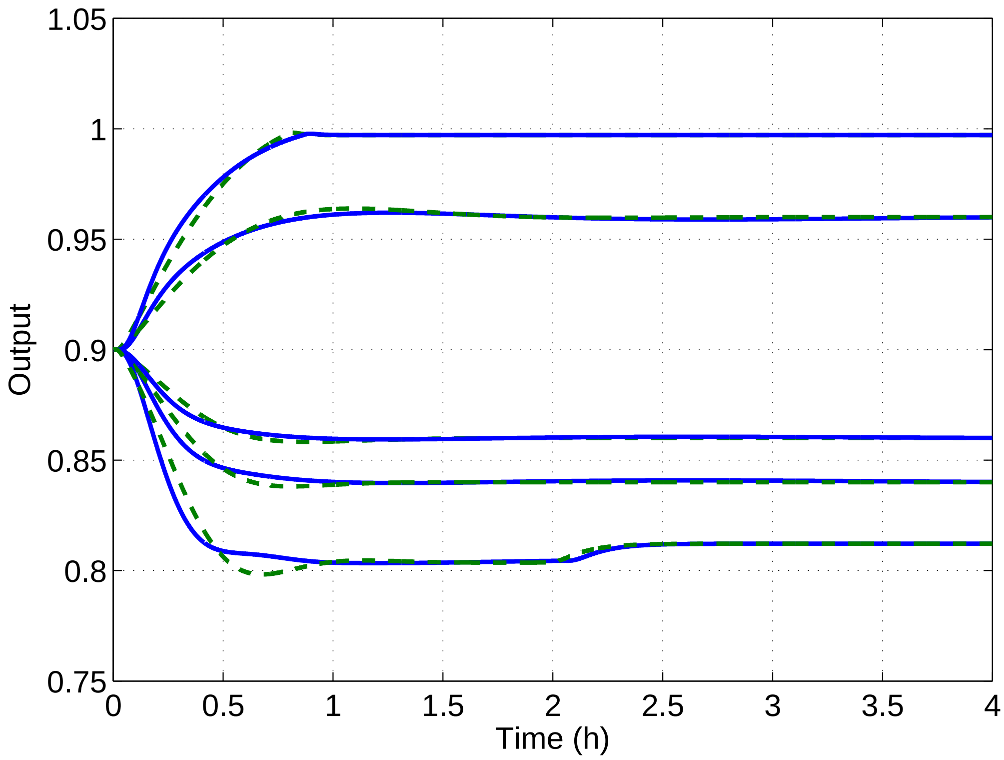

Figure 6.6. The dashed curves (green) are the closed loop step responses of the H_∞ controller which is followed with a PI controller and the solid curves (blue) are the closed loop responses for the controller designed using QFT.

The loop transfer function for both cases of H_∞ and QFT controllers are shown in the Nichols diagram. It is interesting to see that the H_∞ controller has a suprisingly similar loop transfer function to the QFT controller. We note that the phase of the QFT controller is slightly better than the phase for the H_∞ controller.

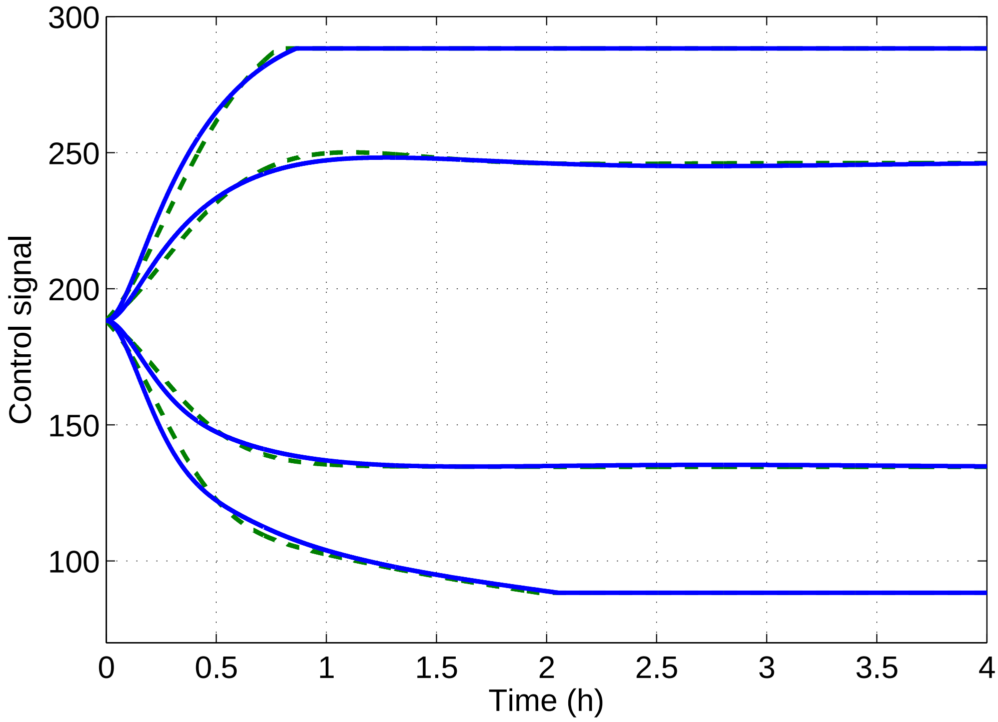

Figure 6.7. The dashed curves (green) are the control activity signals of the H_∞ controller which is followed with a PI controller and the solid curves (blue) are the control signals for the controller designed based on QFT.

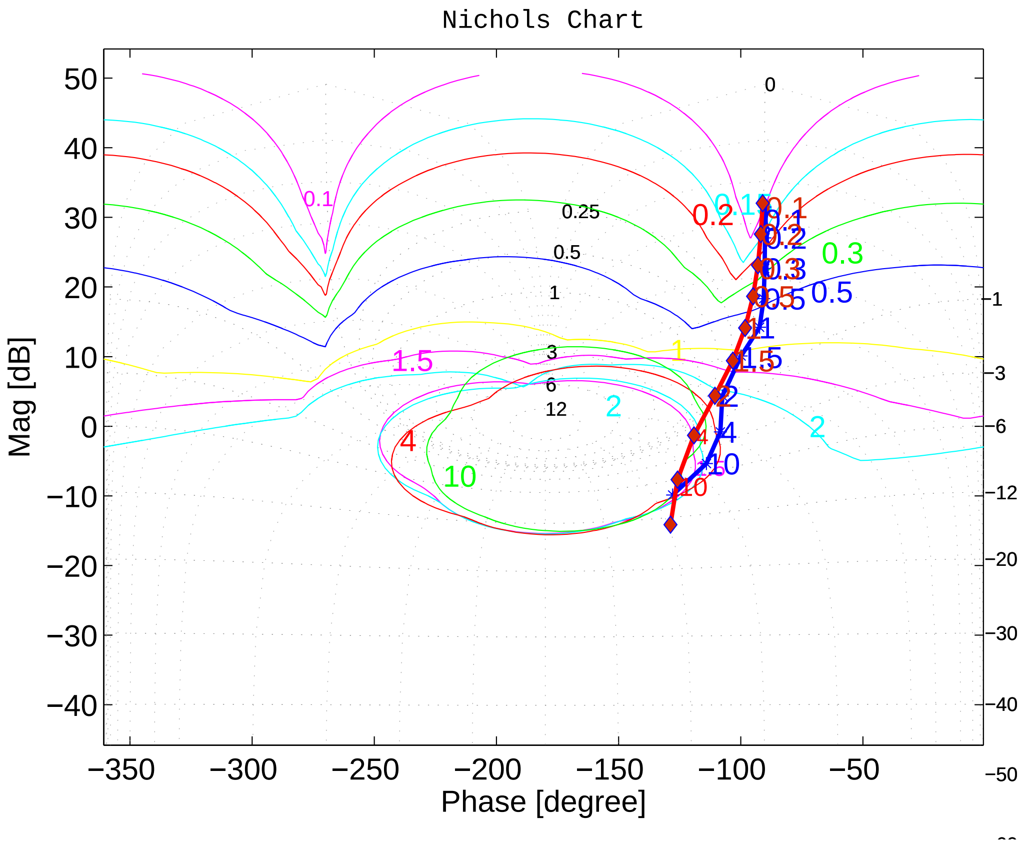

Figure 6.8. The blue curve is loop transfer function for the nominal plant and designed controller based on QFT. The red curve is the loop transfer function for the designed controller based on best linear model.

7 Conclusion

7.1 Conclusion

In this work a method based on linear QFT is used to design simple linear controllers for mildly nonlinear systems. The design is based on local linearization of the nonlinear system. In addition to the classical linearization around only equilibrium points, non-equilibrium points are taken into account as well. One nominal loop transfer function is considered as the nominal plant and the rest are treated as an uncertainty description for the nominal one. In order to facilitate the manual loop shaping in the Nichols diagram, the loop-shaping problem is translated to an optimization problem with Horowitz-Sidi bounds as the optimization constraints. It has been shown by choosing an appropriate cost function for different control target, the result is improved. To solve the optimization problem with non-convex and nonlinear constraints, genetic algorithm is used. The efficiency of this algorithm has been shown in some examples. The linearization around non-stationary points improved both transient response and robustness of our design. This fact is shown through an example which is a second order unstable bioreactor benchmark problem. The method is also successfully applied to a fourth order CSTR benchmark problem. In this problem the selection of non-stationary points is carried out through simulations of the open loop system for small step sizes as references.

7.2 Future Work

In this work genetic algorithm is used to find the coefficients of a fixed structure controller transfer function. It is possible to include the controller order selection in the optimization algorithm. One important aspect in the presented method is the selection of non-stationary points. In this work, this selection is carried out through phase plane analysis for second order systems, simulation of open loop system for stable plants, and simulation of closed loop system for unstable plant. (A controller which stabilizes the main operating window is used in the simulation). If this task is carried out on basis of solid theory it may result in a very powerful and useful method to feedback design of nonlinear systems.

Bibliography

[1] A. Bafios, O. Yaniv, and F.J. Montoya. Nonlinear QFT based on local linearization. *International Journal of Control*, 2003.

[2] A. Banos. Nonlinear quantitative feedback theory. *International Journal of Robust And Nonlinear Control, 17:181-202*, 2007.

[3] H. Chen, A. Kremling, and F. Allgöwer. Nonlinear predictive control of a benchmark cstr. *Proceeding of 3rd Eurpean Control Conference, Rome, Italy*, Sep 1995.

[4] Mehmet Onder Efe. Model refernce sliding mode control for bioreactor benchmark problem. *International Workshop on Variable Structure Systems*, 2008.

[5] C. Fransson, T. Wik, B. Lennartson, M. Saunders, and P. Gutman. Nonconservative robust control: Optimized and constrained sensitivity functions. *IEEE Transactions on Control Systems Technology, 17:298-308*, May 2009.

[6] C. M. Fransson, B. Lennartson, T. Wik, and K. Holmström. Multi criteria controller design for uncertain MIMO systems using nonconvex global optimization. In *40th IEEE Conference on Decision and Control*, volume 1-5, pages 3976–3981, Orlando, FL, USA, Dec 2001.

[7] P.O. Gutman. *Manual for the toolbox for Robust Control System Design for use with Matlab*, May 1996.

[8] Issac Horowitz. Survey of quantitative feedback theory (qft). *International Journal of Robust and Nonlinear Control*, 2001.

[9] I. Horwitz. *Quantitative Feedback Design Theory.* QFT Publications., 1993.

[10] T. A. Johansen, K. J. Hunt, P. J. Gawthrop, and H. Fritz. Off-equilibrium linearisation and design of gain-scheduled control with application to vehicle speed control. pages: 167-180. *Control enginnering practice*, 1998.

[11] H. Khalil. *Nonlinear systems.* Prentice Hall., 2003.

[12] B. Kristiansson and B. Lennartson. Robust tuning of PI and PID controllers. *IEEE Control Systems Magazine*, Feb 2006.

[13] W. T. Miller, R. S. Sutton, and P. J. Werbos. *Neural Networks for Control.* MIT Press, pp.387-402, 1990.

[14] T. Schweickhardt and F. Allgöwer. Linear control of nonlinear systems based on nonlinearity measures. *J. Process Control*, 17:273–284, 2007.

[15] T. Schweickhardt and F. Allgöwer. A robustness approach to linear control of mildly nonlinear processes. *International Journal of Robust Control*, 17:1163–1182, 2007.

[16] T. Schweickhardt and F. Allgöwer. On system gains, nonlinearity measures, and linear models for nonlinear systems. *IEEE transactions on automatic control*, 54(1):62–78, 2009.

[17] T. Wik, C. M. Fransson, and B. Lennartson. Feedforward feedback controller design for uncertain systems. In *42th IEEE Conference on Decision and Control*, pages 5328–5334, Maui, Hawaii, USA, Dec 2003.