ISSN 0280–5316
ISRN LUTFD2/TFRT--5631--SE

H_∞ centering controller for single-wheel drive equipped bogie.

Martin Nilsson

Department of Automatic Control
Lund Institute of Technology
November 1999

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
November 1999

Document Number
ISRN LUTFD2/TFRT--5631--SE

Author(s)
Martin Nilsson

Supervisor
Erik Möllerstedt

Sponsoring organisation

Title and subtitle
H_∞ centering controller for single-wheel drive equipped bogie. (En H_∞-regulator för centrering av en spårvagnsbogie.)

Abstract

A bogie without a wheel-axle, i.e. a single-wheel drive equipped bogie, is desired in a tram mainly because of its possibility for low-floor design. A centering controller for such a bogie is designed. The task of the controller is to center the tram between the tracks, and to avoid running continuously on one wheel-flange. The controller shall in some respect simulate the behavior of a bogie with axles.

The process variations are modeled with an uncertainty description, which is calculated using an optimization routine developed here. The time delays due to communication delays are modeled using Padé-approximation. It is found that gain-scheduled control is necessary, meaning that different controllers for different speeds are used. The found controllers are discretized and reduced without loosing robustness.

The resulting controller is compared with a PD controller. It is found that the H_∞-controller results in a very well-behaved system, with only small overshoot, which for most systems would be better than the PD-controlled system. In this case, however, damping is undesired, and the goal is rather to achieve an oscillating motion with short wavelengths. This unusual control goal makes the PD-controller fit better to this process.

The H_∞-analysis has proved to be a fast and easy way to analyze what can be done with a controller, and to check whether a designed controller is robust. Also, the H_∞ design method can advantageously be used as a good starting point to choose the designing parameters of a classical controller.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
51

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library 2, Box 3, SE-221 00 Lund, Sweden
Fax +46 46 222 44 22 E-mail ub2@ub2.lu.se

Acknowledgement

I would like to thank DaimlerChrysler Research and Technology in Berlin for the possibility of writing this thesis there. Special thanks to Dr. Sebastian Scheffler and Dr. Notker Amann for the help they provided.

Also, I would like to thank Adtranz Germany, especially Dr. Markus Koch, for providing the subject.

Contents:

1 INTRODUCTION ... 6

1.1 PROBLEM FORMULATION ... 6

1.2 CONTROLLER SPECIFICATION ... 7

2 MODELING THE BOGIE OF A RAIL VEHICLE ... 9

2.1 THE BEHAVIOR OF A BOGIE ... 9

2.2 SLIP ... 10

2.3 ELECTRICAL CONNECTION BETWEEN WHEELS ... 12

2.4 SPEED MEASUREMENT ... 14

2.5 THE MODEL ... 15

2.5.1 Model description ... 15

2.5.2 Characteristics of the system ... 16

2.5.3 Model parameters ... 17

3 CONTROLLING THE SYSTEM USING A SIMPLE CONTROLLER 19

3.1 PD CONTROL ... 19

3.2 FILTERED PD-CONTROL ... 20

4 THEORY ON H_∞-CONTROL ... 22

4.1 WHY H_∞-CONTROL? ... 22

4.2 MODELING ERROR DESCRIPTION ... 23

4.3 FUNCTION OF THE H_∞-CONTROLLER ... 27

4.3.1 Basic theory ... 27

4.4 DESIGN ... 29

4.4.1 Augmenting the plant ... 29

4.4.2 Choosing weighting functions ... 31

4.5 BILINEAR TRANSFORMATION ... 32

5 CHOOSING DESIGN PARAMETERS FOR AN H_∞-CONTROLLER 35

5.1 FINDING UNCERTAINTY DESCRIPTIONS ... 35

5.1.1 Uncertainty description for the full parameter-range ... 35

5.1.2 Uncertainty description for a smaller parameter-range ... 37

5.2 CONSIDERING DELAYS IN THE PROCESS ... 38

5.3 CHOOSING WEIGHTING FUNCTIONS ... 39

5.4 BILINEAR TRANSFORM ... 40

6 CALCULATING A CONTROLLER ... 41

6.1 CONTROLLER DISCRETIZATION ... 41

6.2 CONTROLLER ORDER REDUCTION ... 41

6.3 GAIN SCHEDULING ... 42

6.3.1 The fuzzy switching method ... 42

6.3.2 Hard switching ... 44

6.4 RESULTS ... 45

7 THE RESULTING CONTROLLER ... 46

8 CONCLUSIONS ON CONTROLLER DESIGN 50

1 Introduction

1.1 Problem formulation

The bogie of a rail vehicle consists of two pairs of wheels, each pair held together by a stiff axle. However, the axle limits the freedom when designing the bogie. This is especially a problem with a tram bogie, as a low floor is desired to ease boarding. The axle also limits the possibility for the tram to drive through small-radius curves, as the wheels rotate at the same speed. Because of these problems, it is desired to remove the axles in the bogie.

The reason for normally having an axle in a bogie is that it makes the bogie self-centering. When the bogie is displaced, the wheels run with different effective radii because of their conical shape. In combination with their equal rotational speed because of the axle, this makes the bogie turn towards the center, running in a sine-motion as shown in Figure 1.1.

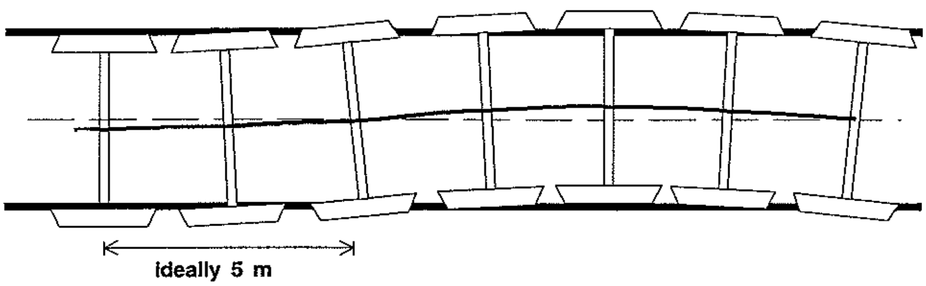

Figure 1.1: Behavior with wheel-axle

This effect is lost when the axle is removed. Instead, the bogie will go to one side of the rail and run with the wheel-flange towards the rail, thus increasing wear and losses. Therefore, the centering effect somehow needs to be reinserted. This can be done if a single-wheel drive equipped bogie is combined with a centering controller, as shown in Figure 1.2. The controller measures the difference of rotational speed between the front wheels, and can this way measure if the bogie is displaced, and if so, which way. It then has the possibility to move applied torque between the left and right sides of the bogie. This way it can steer the bogie towards the center, possibly resulting in an axle-like behavior. Such a controller shall be developed in this thesis.

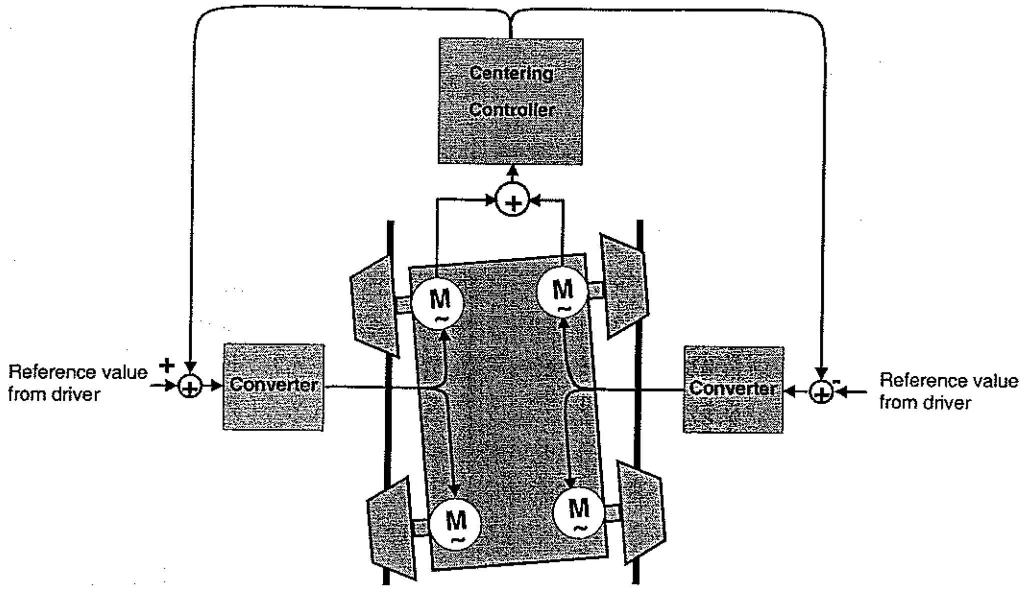

Figure 1.2: Sketch of the centering controller functionality

An advantage with the controller is that it can be turned off when turning. This gives the vehicle the possibility to drive through a curve with smaller radius than if it were equipped with axles between the wheels.

In this chapter, an overview on the problem and control goal is given. In chapter 2, the model used is explained. In chapter 3, a simple controller for comparison is designed, before the theory of a more advanced controller structure, namely H_∞, is given in chapter 4. In chapter 5 and 6 an H_∞-controller is designed, and the results are commented in chapter 7 and 8.

1.2 Controller specification

The task of the controller is to mimic the behavior of a bogie with an axle. As shown above in Figure 1.1, this means a sine-motion with a wavelength of about 20m. With an axle, the damping is very small, and this is therefore also the wanted behavior of the controlled bogie: stable but with minimal possible damping.

The stationary error of the controlled system is not important, since a track is never straight enough for any controller to reach stationarity.

These goals give the following criterias for good control:

- Short time to first-time centering (as close to 5 m as possible corresponding to a wavelength of 20 m)
- The wavelength of the sine-motion should be independent of velocity
- Damping is not primarily desired

Note that these goals, especially the desired low damping, are somewhat unusual, and that this can result in problems using some controller-design methods.

The task of centering the bogie robustly is harder than it may seem at the first look. A large vehicle, with 1.45 m between the wheels, shall be centered with a precision better than 1 mm. As a direct measurement of the displacement is not available, it has to be measured using the difference in wheel radius. In Figure 1.2, it is shown that the effective radii of the wheels differ when the bogie is not centered, due to the conical shape of the wheels. This conicity can be as low as 1/40 mm for each mm of displacement. On top of that, these 1/40 mm in radii-difference must be recognized on wheels with radii of 0.315 m.

Another problem is that the measured input to the controller, the difference in rotational speed between the motors, is of low precision.

In addition to this difficulty, the controller has to be very robust, as the dynamics of the bogie changes drastically with several parameters such as:

- Speed of the tram
- Adhesion between wheels and rail
- Conicity of the wheels

Furthermore, only 10% of the maximum torque from the motor may be used for centering control.

A model of a tram bogie, provided by Adtranz, will be used for the design. For control, a simple controller will first be tried to find out whether such controller suffices to fulfill the demands or not. Then a more advanced controller, based on H_∞-design, will be designed with hope to improve the performance, and to find out whether the more advanced controller is worth implementing or not.

2 Modeling the bogie of a rail vehicle

2.1 The behavior of a bogie

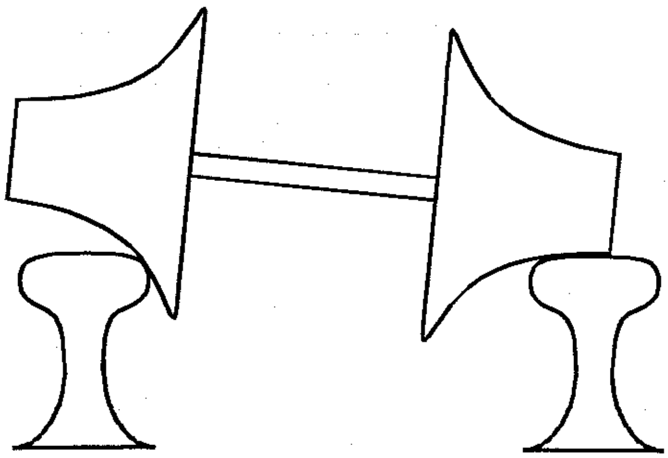

Figure 2.1: A rail-vehicle with wheel axle

A pair of wheels with an axle between them basically is shown in Figure 2.1. The bogie can move sideways on the track because of the backlash between the flanges of the wheels. The dynamics of lateral motion is complex and non-linear, as the contact angles and wheel radii are nonlinearly dependent on the displacement of the wheel-set. For simple calculations and simulations, a model, which assumes constant inclination of the wheel radii, i.e. conical wheels, and one infinitely small contact point with the rail per wheel, is normally used. This is illustrated in Figure 2.2. In the model used for controller design in this thesis, where stability for all occurring dynamics is important, the change of inclination of the wheel will then be seen as an uncertain model parameter λ which can differ between the different sides of the bogie. This is the inclination, which can have any value within a certain set, a value that in reality changes depending on where on the rail the bogie is placed. If the bogie is displaced to the left on the rail, the left wheel will be in contact with the rail at a large angle, whereas the right wheel will have a small contact angle.

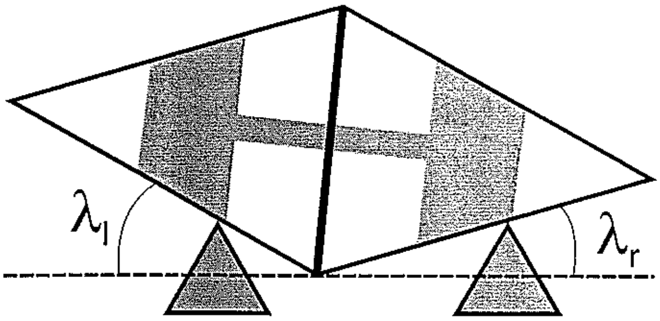

Figure 2.2: Linear model of a rail vehicle with wheel axle

2.2 Slip

Slip occurs when a torque is put on the wheel from the motor, in order to decelerate or accelerate the vehicle. Without slip, no force can be translated from the wheel to the rail. This is seen in Figure 2.3, where the creep force is plotted versus the slip. In this schematic plot, it is shown that when a certain, friction-depending, limit is reached (the maximum of the curve) no bigger traction force can be translated from the wheel, and wheel only starts spinning faster if larger torque is put one the wheels. It also shows that when dealing with lower forces, the system acts linear and slip is proportional to applied force.

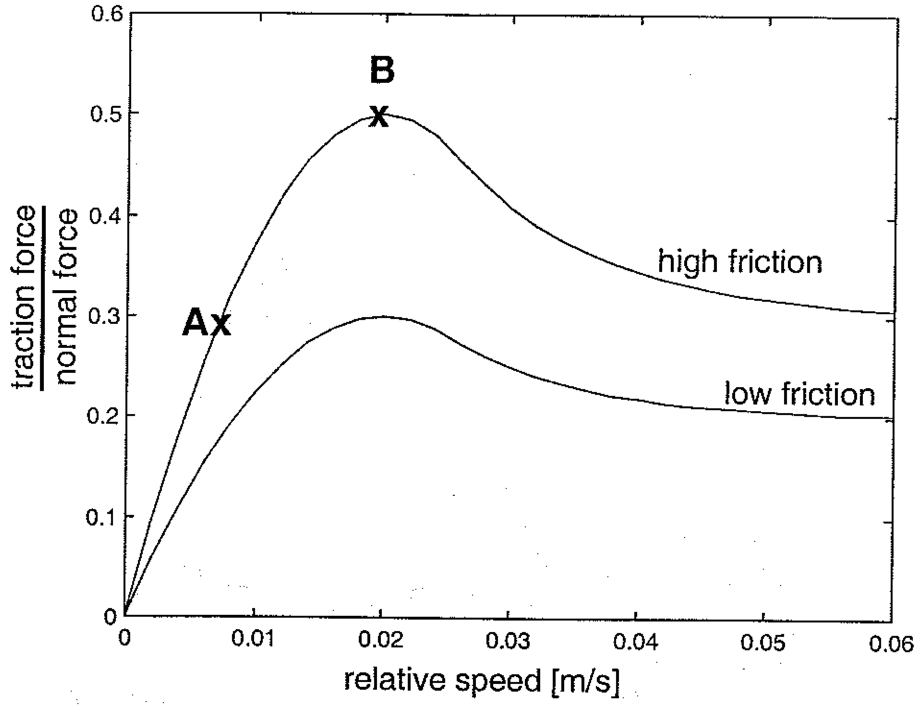

Figure 2.3: Normalized traction force versus wheel slip

In a normal rail vehicle with wheel axles, the motor controller takes the system working to a point on this curve, e.g. point A in Figure 2.3. The controller to be designed here will then move the working point upwards on one side of the bogie and downwards for the other side, as it distributes the force. If the adhesion is too low, there is a risk that the controller will make the working point reach point B, and no more force can be put on the rail. If a larger torque is applied, it only makes the wheels spin faster; the controller can not apply the desired force. In order to be able to apply maximal possible force from the wheels to the rail, an algorithm called search-logic, which identifies the friction, is used. It does this by applying an increasing torque to the wheels until they start spinning fast. It then lowers the force until the wheels stop spinning. This procedure is repeated over and over again, because the friction is not a constant parameter. It has been observed that when the controller output saturates this way, a linear controller will run the risk of being unstable or getting into limit cycles[1] due to the saturation. A possible solution to this problem is to detect this

[1] Limit cycle: the control output swings periodically back and forth, the controller being unable to stabilize the system. This effect comes from nonlinearities such as a saturation on the output.

effect with a detection algorithm and simply turn the controller off. This suggestion is not as radical as it may seem, since this state means that the centering controller hardly can control the system well, a stabilizing traction force is impossible to put on the rail. In the linear Simulink model used here, the force versus slip curve is linear, having a constant but uncertain gradient, and the controller is designed to be robust for a gradient between two values, which represents the changing gradient of the curve.

2.3 Electrical connection between wheels

In a single-wheel drive bogie, each wheel has one motor, but the two wheels on one side share a single converter, and therefore only the *sum* of the torque of the motors can be controlled. An effect that comes from this is that the forces the wheels apply to the track are not independent. This is so, because the asynchronous motors of the tram, one on each wheel, always receive the same voltage from the converter. At a fixed voltage, the output torque from the motor follows a characteristic curve, typically looking as Figure 2.4 shows.

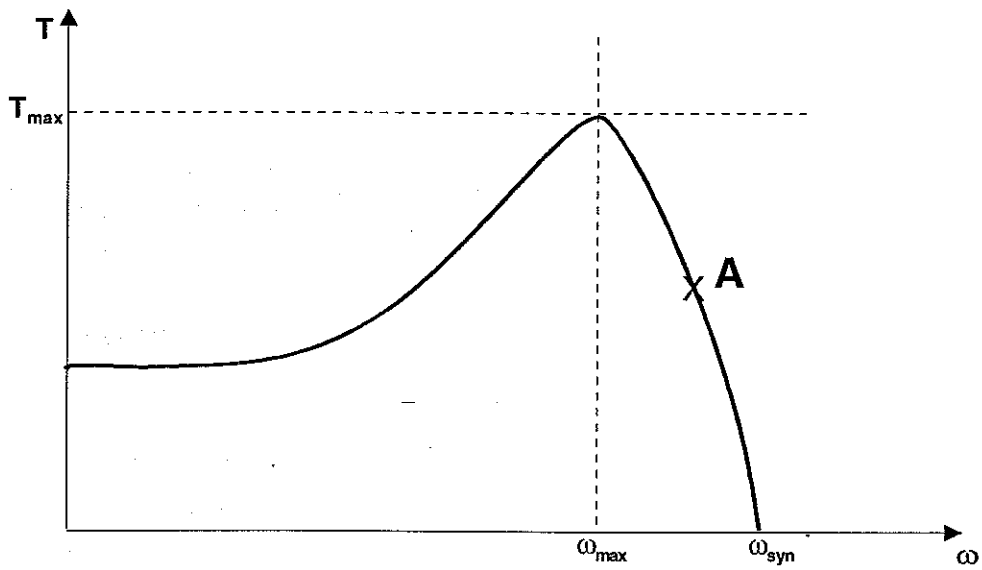

Figure 2.4: Motor torque versus rotational speed

For a certain rotational speed ω_{max}, the torque curve has a maximum T_{max}, called the pull-out torque, and for higher speeds it quickly decreases. At a certain frequency ω_{syn}, proportional to the stator frequency (the frequency of the power coming in to the motor) no torque can be generated anymore. The maximal torque is proportional to the square of the voltage and inversely proportional to the square of the stator frequency.

Normally, each motor is kept at a working point between ω_{max} and ω_{syn}, e.g. point A in Figure 2.4. This makes, in some respect, the motor self-stabilizing. If the load is increased, the rotational speed is decreased, and therefore the torque is automatically increased until an equilibrium (or ω_{max}) is reached. Another reason to keep such a working point is that the losses in the motor are larger when working below ω_{max}. In order to choose the equilibrium point, the system is controlled with a torque controller. This is considered in the model used when simulating the controlled behavior. As this controller is also limited by the one converter per pair of wheels as Figure 1.2 shows, it can not remove the fact that if the motors do not rotate at the same speed, their torque output differ as the converter always delivers the same voltage to both motors. This effect is in the model represented by the linear equation

(2.1) $$\Delta T = C_{mot} \cdot \Delta\omega .$$

C_{mot} is thus a parameter in the model, more exactly the slope on the characteristic function between ω_{max} and ω_{syn} in Figure 2.4, and ΔT is the difference of torque between the two motors on one side of the tram. Once again, this is in reality nonlinearly dependent on ω, more exactly proportional to $1/\omega^2$. Since a linear model is desired, the dependence of C_{mot} on ω is disregarded and C_{mot} is instead seen as another uncertain parameter.

A possibility to solve the whole controlling problem would be to have the same converter connected to front/rear pair of wheels instead of to left/right, i.e. use the structure that Figure 2.5 shows. The connection between the wheels would then result in a centering controller – when the wheel speeds differ, the motors would not deliver the same torque, acting like a weak wheel-axle. But a demand on the controller to be used is that it has to stick to a limited control signal, and that it must be able to handle being turned off in the curves. Designing the bogie with the converter left/right would make these demands hard to fulfil, as the connection effect can not be turned off, and the torque difference can not be limited to a certain value.

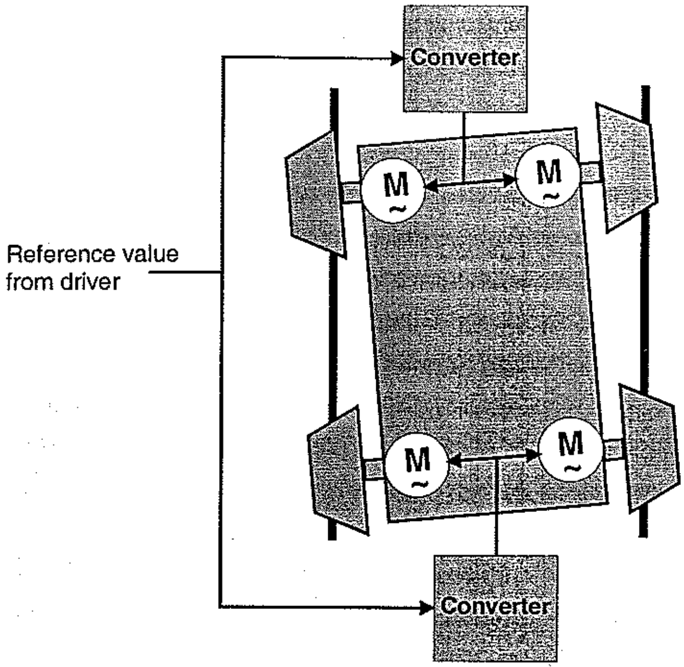

Figure 2.5: Possible controller-free self-centering structure

2.4 Speed measurement

As input to the controller, the difference in angular velocity between the front motors is used. To measure this, the two velocities are measured, and the difference is computed. Since the two measurements are much larger than their difference, the subtraction is a numerically bad operation, especially as the measured speeds are not perfect because of quantization in the sensors. Using typical parameters for this system, it can be found that the relative measurement error is increased due to this subtraction. To get a view of how bad these measurements are, a crude calculation example follows:

Assume $v=18$ m/s, displacement $y=2$ mm, conicity of wheels $\lambda=1/40$ (running rather well-centered with low conicity of both wheels)
True angular velocity is $\omega=57$ rad/s
Measured angular velocity is then $\omega_{measured}\approx57$ rad/s $\pm$ 0.05rad/s (0.1% error in the discrete measurements)
True difference in velocity $\Delta\omega\approx0.018$ rad/s
Measured difference: $\Delta\omega_{measured}=0.018 \pm 0.10$ rad/s >500% measurement error!

How bad the measurements are is of course dependent on speed, conicity of wheels and displacement, but this numerical example shows that using the difference between the rotational speeds is numerically a very dangerous solution as the 0.1% error can by difference-building be increased to over 500%.

The result is that the measurement error is much larger than the measured value. Fortunately, the measurements are in reality better than these figures, as they are strongly improved by the fact that the integrated measurement over time has the correct value (the *angle* measurement is good). This latter effect cannot be modeled as a linear time-independent uncertainty to be used in controller design. A linear time-independent uncertainty model could only include the large measurement error, resulting in so large uncertainties that it would not be possible to find a good centering and robust controller. Therefore this error is disregarded in the design, but it must be tested with simulations after the design. This is only one, but by far the most important, example of uncertainties not taken into account in the model used for controller design.

2.5 The model

2.5.1 Model description

Adtranz provided the linear model of a tram bogie used for the design. It is, all-in-all a 24^{th}-order model including:

- Lateral and longitudinal slip
- Electrical connection between the wheels
- Dynamics of the bogie (12^{th} order)
- Dynamics of the coupled drive (12^{th} order)
- The primary springs

And, only in the model used for simulations when testing:

- Motor torque controller (4^{th} order)

The model is SISO, and it has the torque reference difference between the sides of the tram as input, and as output it gives the rotational difference of the front wheels.

2.5.2 Characteristics of the system

The process has two integrators. The physical explanation of this double-integration is shown in Figure 2.6 and, in words:

- When a torque difference is put on the wheels, the bogie turns. Thus the angle between the bogie and the track is an integration of the torque difference that is put on the bogie.
- The bogie moves transversal on the track with a speed depending on the angle – the displacement is an integration of the angle, and therefore a double-integration of the moment difference.

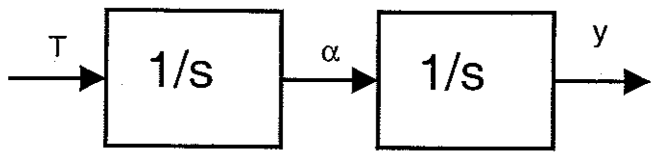

Figure 2.6: The double integration

A bode diagram of the model, schematically plotted in Figure 2.7, shows that the model basically behaves simply as *one* integrator with a roll-off of 20 dB/decade, with another behavior between 1 rad/s and 1000 rad/s. That it behaves as one integrator, although it includes a double-integrator, is because of a transfer-function zero located close to the integrating poles.

Bode diagram for a model of a tram bogie

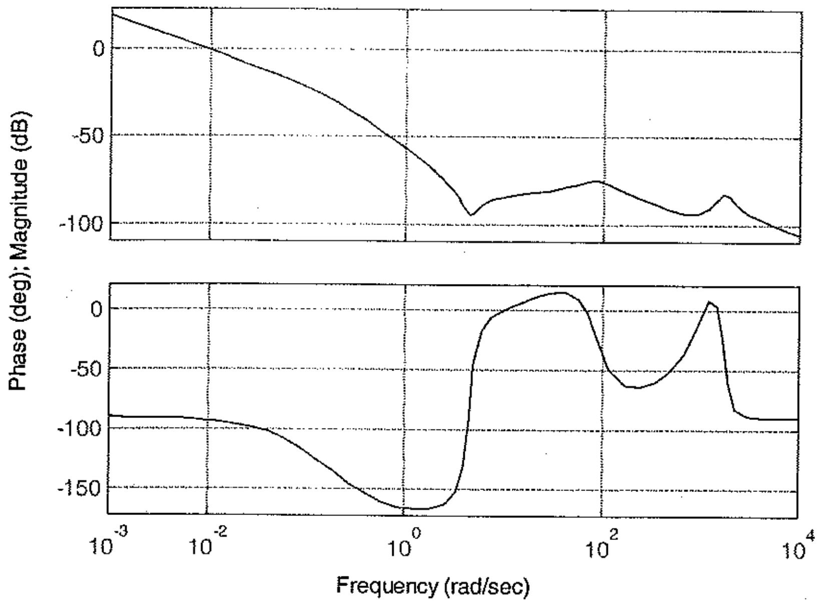

Figure 2.7: Bode plot of the model used

2.5.3 Model parameters

The model naturally includes some approximations and modeling errors, which need to be considered. This is one of the major problems involved in the controller design. The parameters in Table 2.1 were considered uncertain. These were defined in the previous sections. In this linear model, using a c_l and c_r equal to 0 would not be appropriate, since that would make it impossible for the controller to put out any output. The robustness demand is therefore limited to a smaller area which does not start at c_l=c_r=0.

Parameter	Explanation
λ_l	Inclination of the wheel-radius in y-direction, see section 2.1. In reality is λ=f(y) where y is the displacement.
c_l & c_r	Measurement on how much slip is increased for a certain increase in applied force (i.e. gradient of the force versus slip-curve in Figure 2.3). Inclination on the left side equals c_l times the maximal possible inclination. The coefficients are assumed to affect x- and y- direction similarly. Rear and front are also assumed to be equal.
v_0	Velocity of the tram.
C_{mot}	One converter gives power to a pair of wheels (left and right respectively). If the wheels do not spin equally fast, they apply different traction torque $\Delta T=C_{mot}*\Delta\omega$ where $C_{mot} = f(\omega)$. See section 2.3.

Table 2.1: Uncertain model parameters

In the cases of λ and C_{mot}, they are not uncertain in the real sense, but they are nonlinear, and the nonlinearity is modeled as changing parameters. The c_l and c_r are both uncertain and nonlinear.

The inclination of the wheel is dependent on y in such a way that both left or right lambda can not be large at the same time (the tram can only run at the flange of one wheel at the time, see Figure 2.1). Furthermore, the process is symmetric in all other aspects as long as all combinations of c_l and c_r is considered, and therefore one of the λ:s can be considered fix. In other words, the case of the tram displaced to the right does not need to be considered, as long as the case of left displacement is considered for an, in all other respects, symmetric model.

Also other parameters than the one modeled here are uncertain. For example, the wheel radius changes over time by wear. These other uncertain parameters do not drastically influence the system behavior, and therefore their effects are considered to fit into the model with the other parameters.

The controller design has succeeded when the system is stable for all possible parameter combinations, and the tram behaves well when running centered on a dry rail. For the sake of safety, if instability appears, this will need to be detected so that the controller can be turned off.

3 Controlling the system using a simple controller

3.1 PD control

A simple controller was now tuned for the system. For this assignment, an ordinary continuous PD-controller with added filtering (to remove the infinite gain at high frequencies), was chosen as equation (3.1) shows, and it comes into the process as Figure 3.1 shows.

$$(3.1) \qquad U(s) = K\left(1 + \frac{s \cdot T_d}{\frac{T_d \cdot s}{n} + 1}\right) \cdot E(s)$$

The speed difference, Δω, of the wheels is used as a measurement of the displacement, and it is therefore the error signal e. As for a completely centered tram the speed difference is zero, i.e. Δω=0, the reference value is always set to 0.

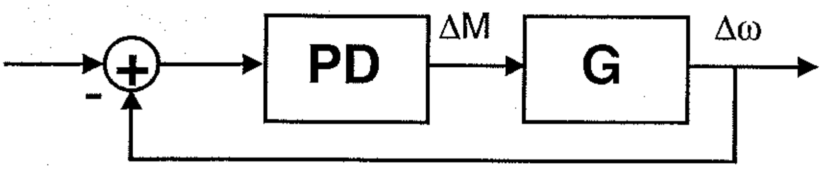

Figure 3.1: Controller structure for PD-controller

The reason for not using an integrator in the controller is the unimportancy of the stationary error mentioned earlier, which leaves no use for the integrator.

As can be seen from the simulation results, the controlled system behaves nicely and stable. The controller found here can be scaled, faster or slower and more or less damped. But a compromise has to be made, because a very fast and well-damped controller will not be very robust against modeling errors and other nonlinear effects.

The control output is not very good, which easily can be seen by looking at the control signal in Figure 3.3. It is here obvious that the system is badly damped at a high eigenfrequency at $\omega \approx 40$ rad/s. This problem has to be taken care of in the controller design. Removing the saturation on the output from the controller makes the damping of the system even worse.

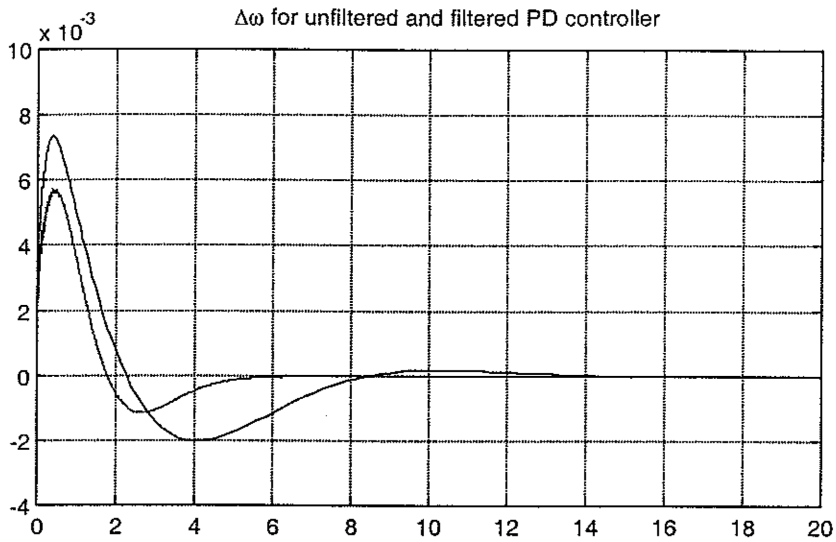

Figure 3.2: Rotational speed difference for the PD-controlled system

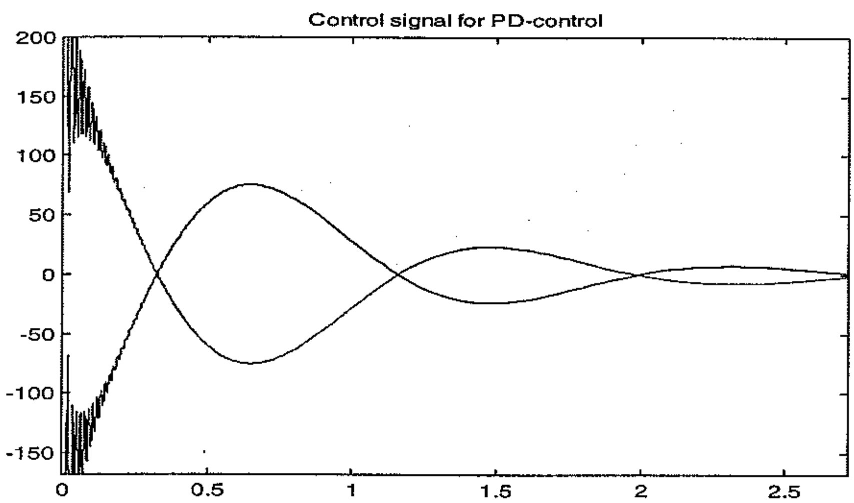

Figure 3.3: Control signal for the PD-controlled system

3.2 Filtered PD-control

To get rid of the high-frequency part of the control signal a low-pass filter (which is equivalent to a phase-lead-lag filter) was added to the controller output. The simple system showed in equation (3.2), a first order low-pass filter, was used to filter the output from the controller. Choosing T_2=28 rad/s and T_1 very large, a well-controlled system was found. The output does not differ significantly from the unfiltered output, but we got rid of the oscillations in the control signal.

$$(3.2) \qquad G_{filter}(s) = \frac{s/T_1 + 1}{s/T_2 + 1},$$

This controller at first seems to work quite well. It is quite robust, changing most of the parameters with ±50% still keeps the system acceptably fast. However, a parameter that changes widely and quickly over time is the adhesion. If the system is simulated with $\mu=0.3*\mu_{norm}$, i.e. the adhesion is reduced to 30% of the maximal possible gradient, it goes into a limit cycle with the controller hitting back and forth between the saturation points. This is a problem that must be taken care of. An easy way to do this is to make the controller slower. Doing this, a robust controller can be found. In the later comparisons with the H_∞-controller, a PD-controller tuned to be robust for a specific speed is used for comparison.

This PD-controller does not match the desired performance it is too slow. Therefore another, better, controller needs to be used. Unfortunately the idea to use an adaptive controller, or an observer/state-feedback controller, which at the first look may seem applicable here, is unfortunately not suitable because of the difficulties in observing some parameters, for example adhesion. This is where the H_∞-controller comes to use, being stable for all possible systems instead of trying to find out how the present system behaves.

4 Theory on H_∞-control

4.1 Why H_∞-control?

The H_∞-controller has evolved from Linear Quadratic Gaussian, LQG-control [4]. The LQG-controllers has proven not to be robust against modeling errors, they have no guaranteed stability margin at all, i.e. an arbitrarily small modeling error can result in instability in the controlled system. For this reason the H_∞-control is preferred for many applications, as one practically never has exact knowledge of the process. The two controller structures are, because of their common origin, similar, though the LQG-control assumes known process dynamics and white noise, while H_∞-control focuses on robust control, to ensure that the system remains stable under influence of limited modeling errors. Some characteristics of H_∞-controllers are:

1) Stability and no large performance losses when disturbances occurs
2) Keeps good performance when modeling errors occurs (robust performance)
3) "Closed-loop shaping" – the engineer decide how the closed system shall behave by choosing weighting functions in the design process, the H_∞ algorithm minimizes functions characterizing the difference between wished and real process behavior.
4) Resulting controller is often of high order, typically the same order as the process.
5) Resulting controller is hard to tune individually, which normally is done when using a standard controller on a tram. This makes the controller hard to commission if not a good process model exists
6) Choosing design parameters requires experience.
7) Controller algorithm requires a process with no integrators. Disturbances are furthermore not allowed to change the nominal stable process into an unstable process.

Note that the 5th point, that one needs a good process model, does not remove the need for a robust controller. If, for example, a perfect model of the tram bogie existed, there would still be need for a robust controller to take care of the changing model parameters. On the other hand, no individual tuning for every tram would be needed since the best controller for the model would also be the best controller for every tram.

The four first points are the features, which make the controller-type suitable for this type of process. The fifth point has the drawback that a software-implemented controller uses much processor-power. This must be taken care of by reducing the order of the resulting controller; how to do this is explained in section 6.2. The controller order can often be reduced without changing the behavior after the controller design has been done. The last point states that an H_∞-controller can not be found for this system, since it has a double-integrator. However, this problem can be solved using a transform method that, during the design, mathematically moves the poles away from the origin into the right half-plane, see section 4.5.

In the following, the basics on how the H_∞ -controller works and can be designed are explained. Proofs and theory not needed for the resulting controller designed, such as MIMO-system controllers, will be disregarded. Much of the mathematics behind it will also be left out. Readers interested in the H_∞-control theory and the proofs not needed for this exact application are recommended to look into the books mentioned in the bibliography, f x [2].

4.2 Modeling error description

The first step in designing an H_∞-controller is to describe the uncertainties of the system, which are then used to determine how robust the controller must be and thereby how fast it can be made.

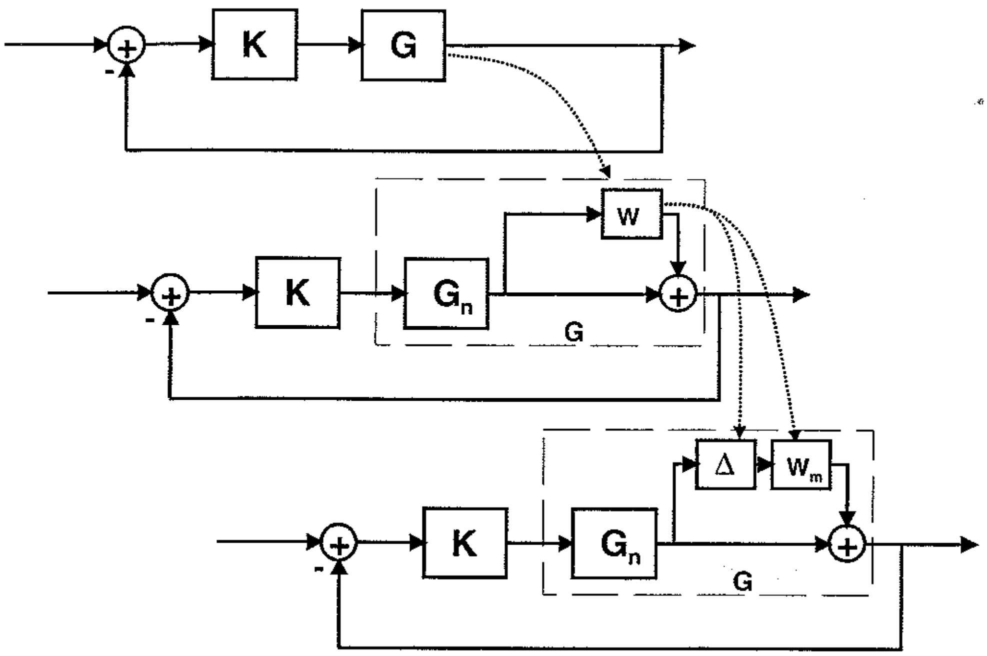

Figure 4.1: The model error description

The idea behind the uncertainty description is as follows: A feedback-loop including a controller K and the controlled system G is normally modeled as the first system in Figure 4.1 shows. But the controlled system G is not exactly known, due to modeling errors and parameter uncertainties. These errors are modeled in the mid-part of Figure 4.1 by W, where G_n is a certain nominal system and W is an unknown disturbance, depending on the actual modeling errors and parameters. Fortunately, W is not totally unknown. If the true process is known but of certain uncertainty, W is can only be within a certain set of possible transfer functions. In order to include all possible modeling errors, a Δ which follows equation (4.1), where the H_∞-norm is defined by equation (4.2) is introduced to cover for the varying characteristics of the uncertainty. If now W_m describes the largest possible disturbance for each frequency, all disturbances are covered.

(4.1) $$0 \leq \|\Delta(j\omega)\|_\infty \leq 1,$$

(4.2) $$\|\Delta\|_\infty = \sup_{\omega \in R} |\Delta(j\omega)|$$

Figure 4.2 basically shows how W_m can be found. If W is calculated for all possible modeling errors and parameter uncertainties, corresponding to the three black curves in the figure, W_m is found as the largest reachable gain in W for each frequency, marked with gray in Figure 4.2.

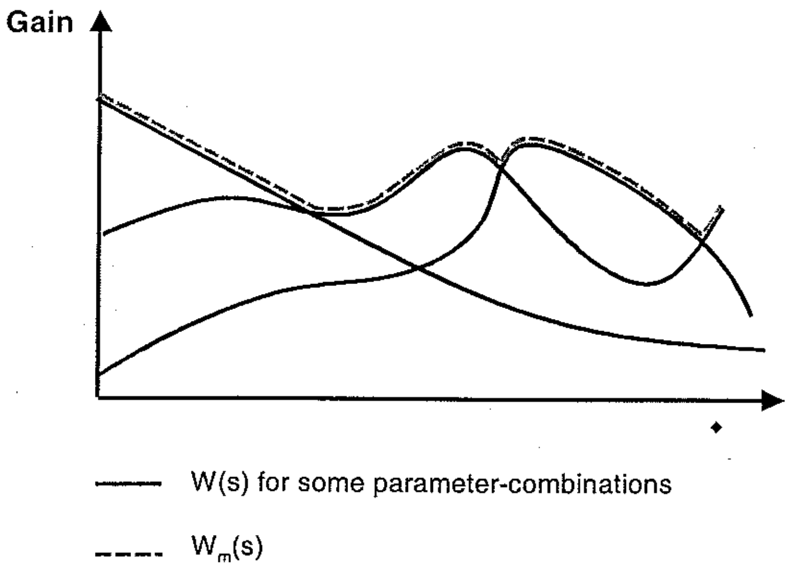

Figure 4.2: Finding W_m by calculating possible errors

There are three commonly used uncertainty descriptions: Additive, multiplicative output and multiplicative input uncertainty. These look as follows:

Multiplicative output uncertainty

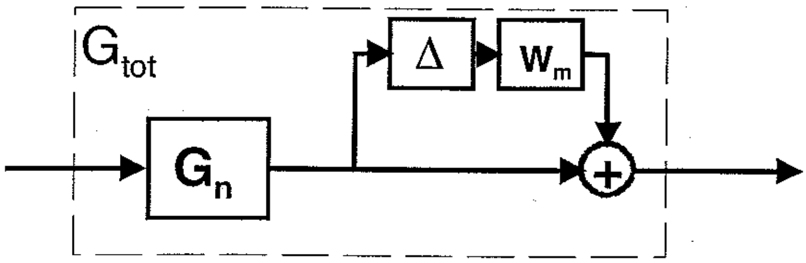

Figure 4.3: Multiplicative output uncertainty

Multiplicative input uncertainty

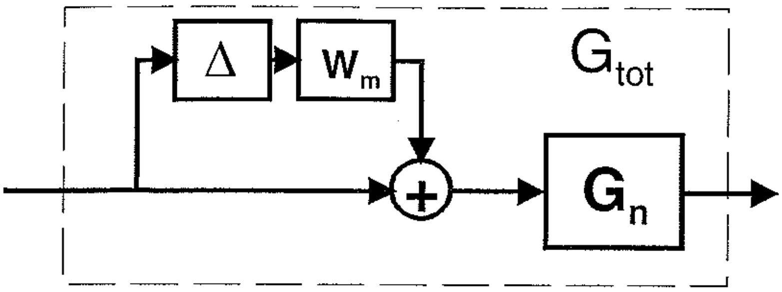

Figure 4.4: Multiplicative input uncertainty

Additive uncertainty

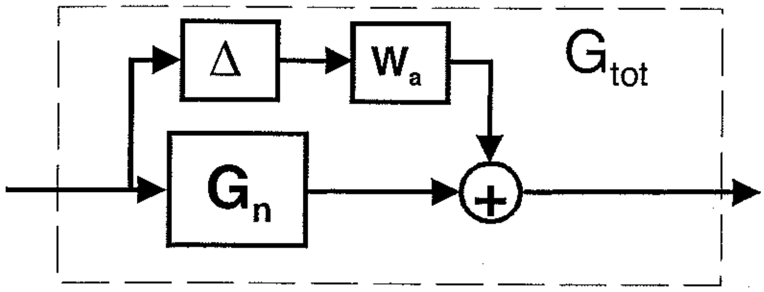

Figure 4.5: Additive uncertainty

To find W_a and W_m for a certain system, optimization can be used. For a certain parameter combination and a chosen nominal model, the disturbance can easily be calculated. The uncertainty is the maximal reachable disturbance, which then can be found by optimization.

An optimization routine that will find the uncertainties, was written so that it optimizes to find the worst-case disturbance for each frequency over a rastered frequency plane. For every combination of model parameters for a certain frequency the disturbance was calculated, and the calculated disturbance was used as the optimized variable. This way, the maximal possible disturbance for each frequency was found, and a transfer function was fitted to this.

Normally, when designing a controller, a simple uncertainty description of low order is used, since the number of states in the uncertainty description together with the plant-model decides the number of states for the controller. Therefore a simple, typically 2-3-state transfer function is used to describe the uncertainties.

Which uncertainty description that is the most suitable to use depends the system for which the controller is designed. All descriptions can cover all modeling errors, but choosing the best description can result in better control. For SISO systems, which this system is provided the displacement is regarded as the only output, output multiplicative and input multiplicative uncertainty are exactly the same. This leaves only two descriptions to be regarded in this case, multiplicative and additive uncertainty.

This way of describing the uncertainty can not only be used for system with unknown parameters, but also to get a linear description of a nonlinear system. The nonlinearity is then described as a linearity with uncertainty, resulting in a linear nominal model and uncertainty description. This is used later, when designing the controller, in order to take care of e.g. adhesion-dependence in the model.

In order to find a robust controller, G_{tot} must include *all* possible systems. However, the controller cannot be expected to handle all possible systems, e.g. is it earlier mentioned that the linear system where the gradient of the slip/force-curve is 0 is impossible to control since the control signal does not affect the system. Because of this, the parameter variations have to be kept in a set in which it is reasonable to expect the existence of an stabilizing controller. If the real parameters are outside of this set, the instability is detected and the controller shut down.

4.3 Function of the H_∞-controller

4.3.1 Basic theory

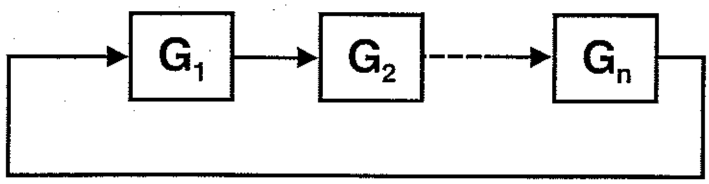

Figure 4.6: Systems connected in series

To guarantee the robustness of a closed-loop system, H_∞-theory uses a form of the small-gain theorem. This states that a closed-loop system of the form in Figure 4.6 is stable if the including systems are stable and the product of the gains of the subsystems is less than 1, i.e. inequality (4.3) is fulfilled.

(4.3) $$\left\| \prod_{k=1..n} G_k \right\|_\infty \leq 1$$

Strict inequality in the equation above assures asymptotic stability. The closed-loop system including uncertainty can be transformed into the system in Figure 4.7, where P depends on the uncertainty type used.

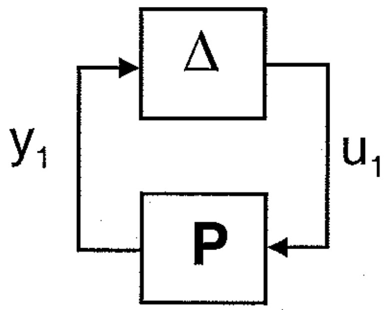

Figure 4.7: Controlled system written in standard configuration

Multiplicative uncertainty:

(4.4) $$P_{mult} = \frac{-K(s)\cdot G_n(s)}{1+K(s)\cdot G_n(s)}\cdot W_m$$

Additive uncertainty:

(4.5) $$P_{add} = \frac{K(s)}{1+K(s)\cdot G_n(s)}\cdot W_a$$

This means, since Δ(s) is normalized to have an H_∞-norm less than 1, that robust stability can be guaranteed provided Δ(s) and G_2 is stable and

(4.6) $$\|P_{mult}\|_\infty \leq 1 \quad or \quad \|P_{add}\|_\infty \leq 1,$$

also here with asymptotic stability if strict inequality applies.

Note the relation *or*, i.e. it is enough to fulfil the criteria for *one* of the uncertainties. The limitation to stable disturbances Δ(s) may seem like a problem. For example, when the disturbance makes the stable nominal system unstable, the theory does not work. This case can fortunately mostly be circumferred by using a different nominal plant, in the example use an unstable nominal plant, since the nominal plant can be chosen freely provided the uncertainty descriptions are adjusted. A limitation however remains, the disturbances must not change too much in the appearance of the plant.

4.4 Design

4.4.1 Augmenting the plant

The problem of finding a robust controller is now limited to finding a stabilizing controller such that equation (4.6) is fulfilled. To do this, the first step is to augment the plant to the plant showed in Figure 4.8, where u_1 is th ereference value, u_2 the controller output, y_2 is the control error and y_1 is a frequency-weighted measurement of control error, control signal and process output. The reason for doing this will become clear in the next section. W_1, W_2 and W_3 are here weighting functions, which are the design parameters in the later control design. They can therefore be chosen freely, but must be well-chosen in order to get good control characteristics. The fourth design parameter of the H_∞-controller is also included here, and it is the nominal model. The choice of nominal model affects the design parameters by changing the uncertainties, as well as changes the behavior of the closed-loop system since the controller is designed for the nominal system.

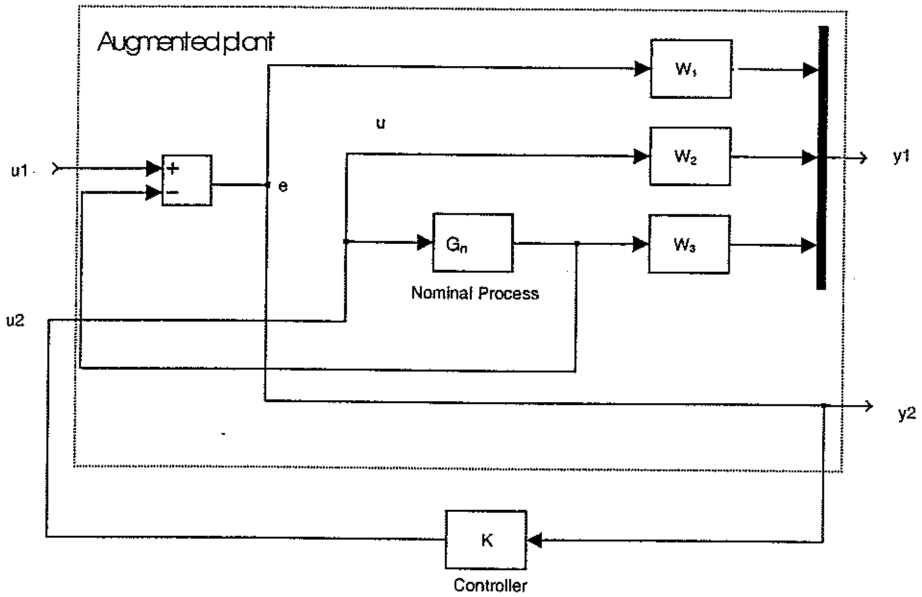

Figure 4.8: Augmentation of the plant

The reason for doing this augmentation is that the transfer function matrix from u_1 to y_1 can now be written:

$$(4.7) \qquad P_{y_1u_1} = \begin{bmatrix} W_1S \\ W_2KS \\ W_3T \end{bmatrix}$$

Where the sensitivity function S(s) is defined by equation (4.8) and the closed-loop transfer function T(s) is defined by equation (4.9)

$$(4.8) \qquad S(s) = \frac{1}{1 + K(s) \cdot G(s)}$$

$$(4.9) \qquad T(s) = \frac{K(s) \cdot G(s)}{1 + K(s) \cdot G(s)}$$

Putting this transfer function together with equation (4.6), it is seen that, provided the inverse of the weighting function $W_2(s)$ is larger than the additive uncertainty W_a or the inverse of $W_3(s)$ is larger than the multiplicative uncertainty W_m, i.e. equation (4.10) is fulfilled, robust stability is guaranteed if equation (4.11) is fulfilled for the controlled system.

$$(4.10) \qquad \left|W_2^{-1}(s)\right| > \left|W_a(s)\right| \quad or \quad \left|W_3^{-1}(s)\right| > \left|W_m(s)\right|$$

$$(4.11) \qquad \left\|P_{y_1u_1}\right\|_{\infty} \leq 1$$

The control design algorithm now has to find a stabilizing controller that fulfils this equation. How many and which weighting functions will be used is up to the freedom of choice, as long as certain regularity conditions are fulfilled. If a certain weighting function is not used, i.e. set to 0, the corresponding result can behave in any way. If e.g. W_1 is not used, it means that any sensitivity function S(s) for the controlled system is accepted.

4.4.2 Choosing weighting functions

The question arises, how to choose the weighting functions for the design. To do this, it helps to look at the weighting functions from another point of view. In the augmented plant it is easy to see that the $W_2(s)$-function can be seen as a punishment on the control signal. A larger W_2 forces a smaller control signal to keep the inequality (4.11) true. More exactly, for each frequency $W_2(s)$ defines how large the control signal may be compared to the error signal. The two different ways of viewing this weighting function expresses the known fact that a compromise between fast and robust controllers always must be done. To get a fast controller, the control signal must be allowed to be large and therefore W_2 must be small. But this also means, from the other viewpoint, that only a small additive error is allowed. Qualitatively seen, a slow controller is also a robust controller.

W_1, as seen in equation (4.7), works as a punishment on the sensitivity function, since the closed-loop system is forced to satisfy this equation. The function is therefore used to describe the inverse of the desired sensitivity function, whereas W_3 is used to describe the inverse of the desired complementary sensitivity function. This is the so-called closed-loop shaping. The control-engineer decides as controller parameters how the transfer functions shall look, and the algorithm finds a controller that fulfils the wish.

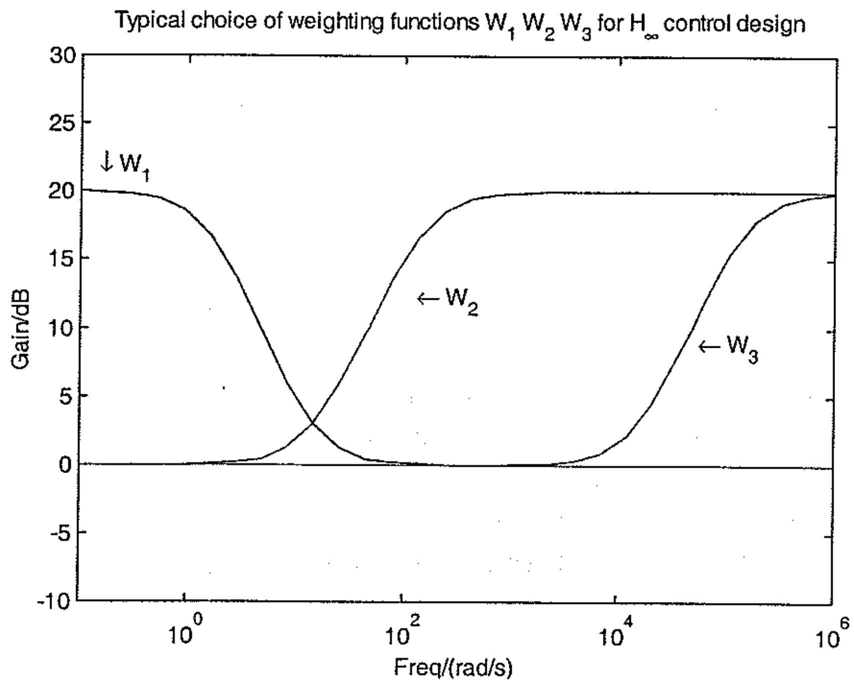

Figure 4.9: Typical choice of weighting functions for H_∞ control design

Choosing bad weighting functions may cause (4.11) to be impossible to fulfil. Therefore, normally a generalized version of the criteria is used for the controller design, equation (4.11).

$$(4.12)\quad \frac{1}{\gamma}\left\| P_{y_1 u_1} \right\|_\infty \le 1 .$$

This gives the algorithm the possibility to find a compromise if the weighting functions are badly chosen - γ is minimized, not necessarily smaller than 1. If a controller is found such that $\gamma \le 1$, the robustness is guaranteed. If $\gamma > 1$, the robustness can be checked by looking only at the part of P_{yu} that describes the uncertainty, or by simply plotting KS or (1-S) versus the inverse uncertainty, and check that the uncertainty is smaller for all frequencies.

When choosing weighting functions it is important to bear in mind that the resulting controller is of the plant model order plus the sum of all the weighting function orders. It is therefore not a good idea to try to describe an error exactly, while it will possibly result in a very high order controller. Furthermore, the possible error is seldom known exactly.

The algorithm finding the controller works by choosing a start γ, and trying to find a controller for this γ. If it does, it repeats for a smaller γ, if not, it repeats for a larger γ, a procedure called binary search. This way, the smallest possible γ for which a controller exists can be found.

4.5 Bilinear transformation

It is known that systems having poles on the complex axis causes difficulties when designing an H_∞-controller. As a matter of fact, the considered system has two integrating poles. The problem is a numerical problem arising in the MATLAB control design algorithm. This problem has to be taken care of when designing the controller. One method for doing this is the bilinear transform [5]. It suggests transforming the system using the transform in equation (4.13)

$$(4.13)\quad s = \frac{\tilde{s} + p_1}{\dfrac{\tilde{s}}{p_2} + 1}, \qquad \tilde{s} = -\frac{s - p_1}{\dfrac{s}{p_2} - 1},$$

where p_1 and p_2 are negative numbers. This transforms the poles/zeros of the system according to the scheme in Figure 4.10.

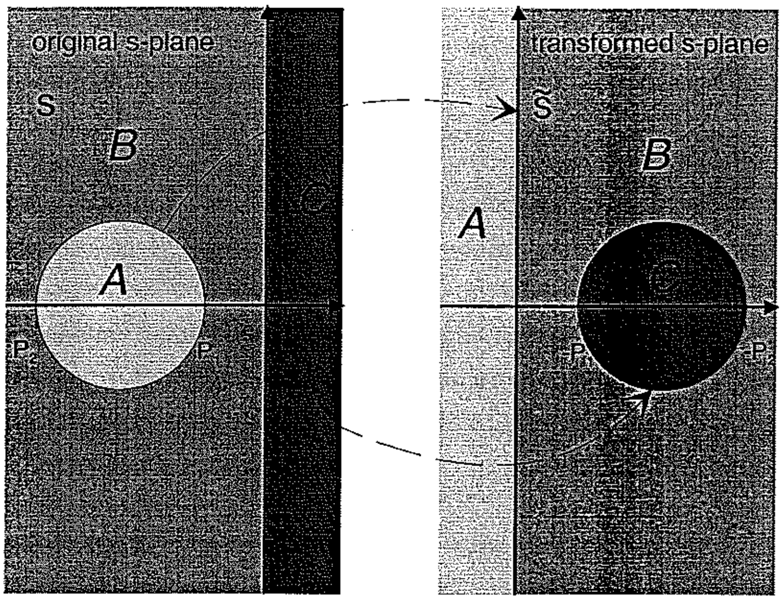

Figure 4.10: The bilinear transform

For the transformed system, which has no purely imaginary poles if sensible transform parameters are used, an H_∞ controller can then be found. This controller then has to be transformed back into the original s-plane.

Since a controller is designed for a more unstable plant than the real plant (i.e. when transforming back, a stable plant will become more stable), one can guarantee stability for the found controller. The controller found is on the other hand only sub-optimal, it has good, but not necessarily optimal, robustness.

This transform is further investigated in [5] and [6], where it also is proposed to chose a $-p_2$ much larger than the closed system bandwidth and a p_1 as

(4.14) $$p_1 \approx \frac{-2}{T_s}$$

where T_s is the desired settling time of the system. Choosing p_2 to be $-\infty$ will result in the transform moving all poles/zeros $-p_1$ to the right, since the circle has a very large radius.

The choice of parameters is limited by the fact that the weighting functions also must be transformed. They must also be stable, which means that the transformed weighting functions must be stable. Hence, all their poles must be within the circle A in the original s-plane. One result of this is that it is impossible to demand a lower bandwidth of the closed-loop system than $-p_1$.

The bilinear transform can also be used to guarantee robust performance for the controlled system. Since the controller design makes the transformed closed-loop system having all poles in section A, independent of the disturbance on the process, the poles are placed in the circle A in the back-transformed system. Choosing a suitable circle, the desired performance is guaranteed.

5 Choosing design parameters for an H_∞-controller

5.1 Finding uncertainty descriptions

5.1.1 Uncertainty description for the full parameter-range

In order to find uncertainty descriptions for the system, optimization was used. The optimization routine was written so that it makes a frequency raster, and for each frequency finds the worst-case disturbance, i.e. the uncertainty, and the worst-case process parameters. In order to avoid finding local maxima, the optimization was started from several different process parameters for each frequency. The result using additive uncertainty description G_a and multiplicative uncertainty description G_m is shown in Figure 5.1.

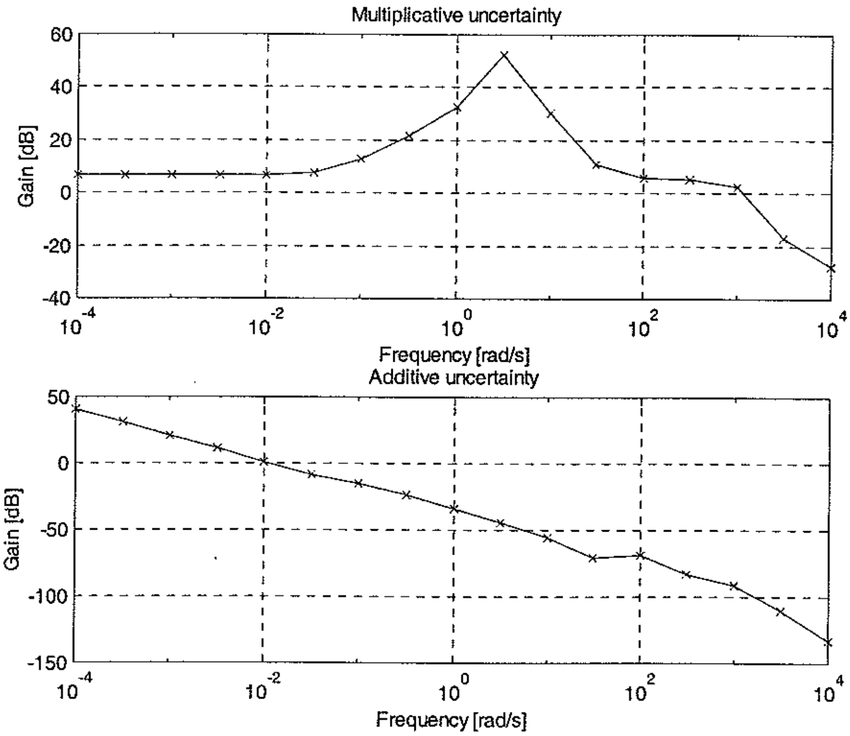

Figure 5.1: Uncertainties, full parameter-range

Since the chosen weighting function W_3, if multiplicative error description is used, must follow the equation (5.1), where G_m is the multiplicative error, the bandwidth of the closed-loop system is limited by the lowest frequency for which the multiplicative error description is larger than 0 dB. It is obvious in the plot above that this frequency is, if it exists, smaller than 10^{-4} rad/s, which is no acceptable bandwidth. Noting that a multiplicative output error larger than 0 dB corresponds to the model not being able to foresee as much as the sign of the output, the limitation of the bandwidth as the crossover frequency makes sense. It is naturally hard to control a system using such a bad model.

(5.1) $$\frac{1}{G_{cl}(s)} \approx W_3 \geq G_m$$

The uncertainty description found is dependent on which nominal system is used. One way to decrease the uncertainties is to find the nominal system that gives the smallest uncertainties. The catch with this method is that during control design the controller will be designed for the nominal system. The closed-loop shaping only considers the nominal system, and does not say anything, except from the guaranteed stability, about how a non-nominal system may behave. Since the important part of this controller is that it must behave well for a system under particular circumstances, such as dry rail, it is therefore suitable to design the controller with the corresponding parameters as the nominal system. But a compromise needs to be done, since the controller can be made faster if another system is used as nominal. The behavior for the dry-rail system must, if another nominal system is used, be checked with simulations.

Does it work better using the additive uncertainty description? It shows clearly integrating behavior in the plot in the sense that it for small frequencies decreases with about 20 dB per decade. A weighting function W_2 that has to be larger than the uncertainty description would have a pole for a small frequency, at least $<10^{-4}$ rad/s. In section 4.5, it is mentioned that the weighting functions must be stable, even after the transform. This would limit the transform parameter to $0<p_1<10^{-3}$, which is not enough for the bilinear transform to have any effect. The standard solution to this problem is to place a pole for as low frequencies as possible in W_2 and then do the design, hoping that the KS-function for the controlled system will show the integrating behavior, hence pass the robustness check. The bilinear transform unfortunately removes this possibility also.

5.1.2 Uncertainty description for a smaller parameter-range

The conclusion from the uncertainty calculation above is that the system has an uncertainty which is too large, and the only way to find a good controller is to have a smaller uncertainty. A way of making the uncertainty smaller is to use gain scheduling, which will add the possibility to limit the parameter-range in which the controller must be stable. Since an H_∞-controller is very hard to schedule continuously without loosing the guaranteed robustness property, it is not very suitable for continuous gain-scheduling. The only gain-scheduling that can be done, is using a number of different controllers, and which one to use is dependent on a measured variable. The only measurable uncertain parameter in this process is the speed of the tram v_0, and this is therefore the only possible gain-scheduling variable.

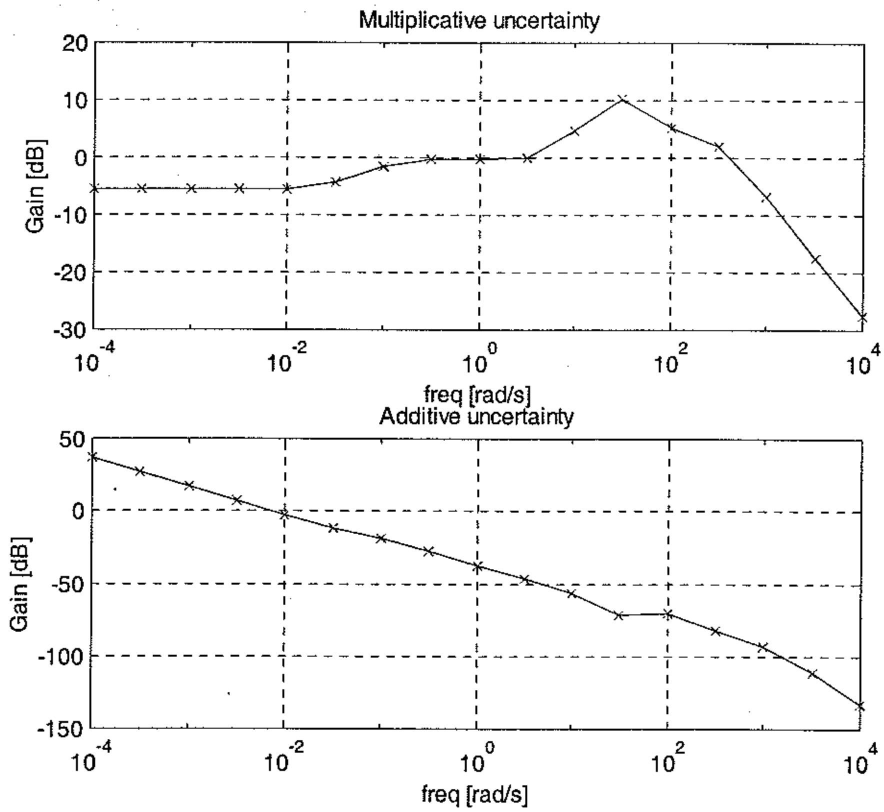

Figure 5.2: Uncertainties, reduced parameter-range

Using the same optimization routine as above, the error description was now calculated limiting v_0 to be in the area 15 m/s $\leq v_0 \leq$ 20 m/s. A smaller error description was now found, Figure 5.2. As seen in this Figure, the bandwidth is still strongly limited if multiplicative uncertainty is used for the controller design. The additive uncertainty has kept is integrating characteristics, which makes the controller design hard.

For three other areas of velocity, 3 m/s $\leq v_0 \leq$ 6.5 m/s and 6.5 m/s $\leq v_0 \leq$ 10 m/s and 10 m/s $\leq v_0 \leq$ 15 m/s the same procedure was made and qualitatively similar results were found.

Because of this, four controllers will be designed, each for a different speed range. For each one, the uncertainties are computed, as above. The weighting functions are then chosen, see section 5.3, and the system can then be controlled with a controller chosen by measuring the speed. How to do the switching is explained in section 6.3.

5.2 Considering delays in the process

The error descriptions was found assuming no delays in the process. The delays in the process come from the communication where the measurements are transferred to the CPU, and where the control signal is transferred to the motors. This communication is done through a data bus with specific time delays, which are known. To consider these time delays, their Padé approximation, which is shown in equation (5.2), is used. This is needed as the delay can not be included in a continuous model described only with proper parts. The Padé-approximation is further explained in [2].

$$G_{delay} = e^{-T_d \cdot s} \approx \frac{(1-\frac{T_d}{2n}s)^n}{(1+\frac{T_d}{2n}s)^n}, \tag{5.2}$$

The Padé approximation is, as the name says, only an approximation, but the approximation can be done with arbitrary precision.

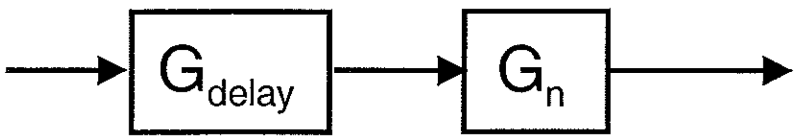

Figure 5.3: Approximating delays

This approximation comes into the system in series. The order of the approximation has to be chosen considering the trad-eoff between order of the plant (and resulting controller) and accuracy. In this case a fourth order approximation was used since it makes a good approximation for all interesting frequencies (higher frequencies disappear by discretization), and the order is of lower importance since the resulting controller must be radically reduced.

5.3 Choosing weighting functions

To choose which uncertainty description to use, trial and error is used, simply designing one controller for each type, and finding the best. It was here shown, because of the earlier mentioned bad properties of the additive uncertainty, that it is suitable to leave the weighting function W_2 out.

After considerable trial and error, the weighting functions shown in equation (5.3) were chosen.

$$(5.3)\quad \begin{aligned} w_1(s) &= \frac{s+\omega_b}{s+\frac{\omega_b}{10}} && (1) \\ w_3(s) &= 10\cdot\left(\frac{s+\omega_b}{s+10\cdot\omega_b}\right)^2 && (2) \end{aligned}$$

The possibility for a W_2, which can be seen as the additive error or a punishment on the control signal, is here not used when multiplicative disturbance is used. This gives the possibility for arbitrarily large control signals for high frequencies, but since the discretization removes the high frequencies, this is not a problem. The weighting functions W_1 and W_3 look as Figure 5.4 shows.

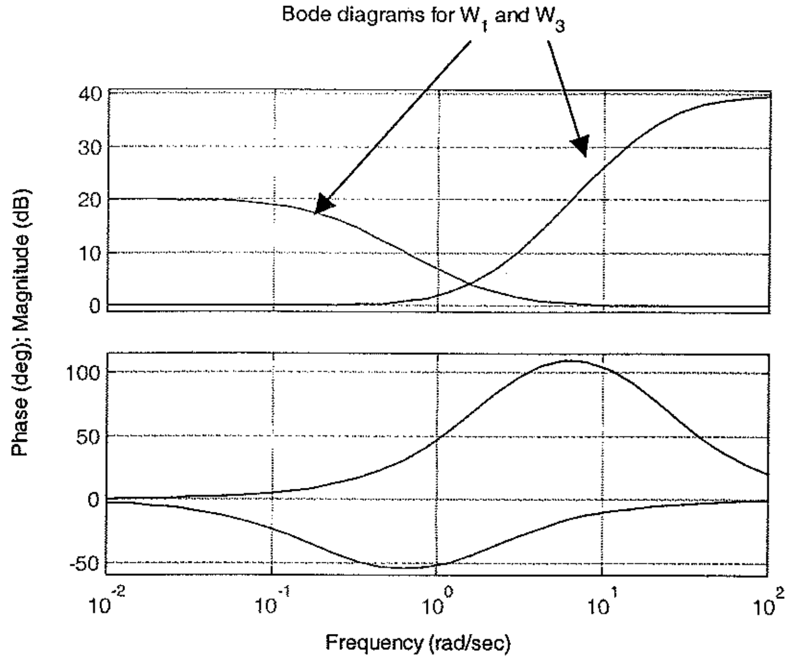

Figure 5.4: Chosen weighting functions for one controller

5.4 Bilinear transform

As earlier mentioned, the freedom of choosing transform parameter p_1 when doing the bilinear transform is limited by the weighting functions, since p_1 is not allowed to be larger than the smallest pole in any of the weighting functions. The p_1 parameter is simply chosen by trial and error within the possible limits. p_2 was chosen very big (10^8), since a well damped system was not desired. The larger p_2 is, the more poorly damped is the controlled system forced to be, since all poles are placed in circle A in Figure 4.10, which is smaller for a smaller p_2. The reason for not choosing p_2 infinitely large is purely numeric.

This transformation and choice of weighting functions result in three different controllers, each for a certain area of speed and each of 34^{th} order. This is far too much to be suitable to implement, and therefore an order reduction must be made, see section 6.2 below.

6 Calculating a controller

6.1 Controller discretization

For the real process, the controller will be implemented in discrete form and discretization therefore has to be made. As long as the sampling time is much faster than the step response of the controlled system, the discretization does not affect the behavior of the controller noticeable. The discretization was done using the Tustin approximation with frequency prewarping shown in equation (6.1), where T_s is the sampling time, and ω is a matched frequency was used.

$$(6.1) \quad s = \frac{\omega}{\tan(\frac{\omega T_s}{2})} \cdot \frac{z-1}{z+1}$$

This approximation is used because of its property to guarantee matched continuous- and discrete- time frequency responses at a given frequency. In this case, 1 rad/s was used as the matching-frequency. Testing showed that this had to be used in order to get an acceptably good discretization, which does not loose the robustness. ω was chosen to guarantee matching at the most critical frequency where the robustness margin is small.

6.2 Controller order reduction

Model reduction was performed by finding a smaller realization by balancing the controller and removing the fastest states. This resulted in all four controllers of 6^{th}-7^{th} order, without changing their behavior noticeably. A 6^{th} order controller may seem like a big step forward, but it is still too many parameters to set manually, which is done with classical controllers (PID-type) when starting to use each tram. It is therefore desired to find a controller with only 3-4 states.

The model reduction can only be done if loss of the H_∞-optimality is accepted, and this in turn can be done as the robustness demands in inequality (6.2) can be used on the resulting reduced controller to see if it is robust stable. As long as the analysis still can prove the reduced controller to be robust, and the difference in performance between the reduced and the unreduced controller is not too large, the changed behavior can be accepted. This is one of the advantages with the H_∞-analysis, that it can be used to test robustness on any system with any type of controller.

$$(6.2)\quad \frac{-K(s)\cdot G_n(s)}{1+K(s)\cdot G_n(s)} \leq W_m^{-1} \quad or \quad \frac{-K(s)}{1+K(s)\cdot G_n(s)} \leq W_a^{-1}$$

6.3 Gain scheduling

6.3.1 The fuzzy switching method

A possible solution to the gain scheduling works by using different controllers for different speeds. In order to avoid bumps in the control signal while switching from one controller to another, the fuzzy logic-idea can be used in the switching areas. Each controller is given a weighting function, which is a function of speed. At the most, the weighting functions of two different controllers may be nonzero for a specific speed, and of course the sum of all weights is 1 for all speeds. In order to ensure that the controller have a reasonable output when it at first sets any output, it is updated in an area of speed that is slightly larger than the area in which it affects the output. If it was always updated, there would be a risk for wind-up, and therefore be in a bad state when it switched in.

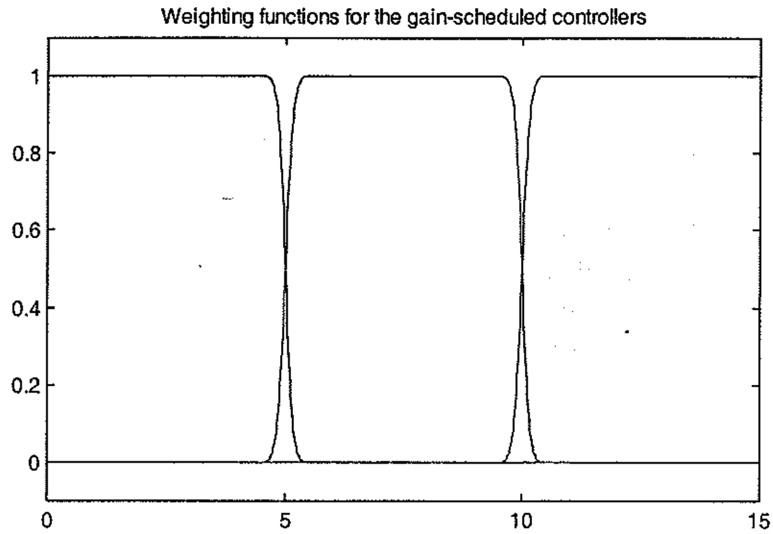

Figure 6.1: Possible weighting functions for controller-switching

When using a method like this, a question has to be answered; is the combination of two controllers robust if the two controllers are robust? For an H_∞-controller this can unfortunately only be shown numerically, and not in general, it has to be done for every combination of controllers. The idea is the same as the one used for H_∞-design: A system is robustly stable if two criteria are fulfilled:

- The nominal system is stable
- The complementary sensitivity function T follows equation (6.3)

Where G_m is the multiplicative uncertainty for the process[2].

$$T(s) \leq \frac{1}{G_m(s)}, \forall s, \quad (6.3)$$

That the second criteria is fulfilled is trivial: Both controllers result in a complementary sensitivity function which fulfills the criteria, and the complementary sensitivity function of the average of the controllers will always be in between them, which means it too fulfills the criteria. This means that only nominal stability has to be shown – robustness follows automatically!

The fulfillment of the first criteria is not trivial to show for a system of this complexity. To do this, a root locus curve was drawn sweeping v_0 within the ranges where the controllers are mixed. Noting that the curves stay within the unitary circle can be seen as a numerical proof.

This robustness proof has the limitation that it assumes perfect velocity measurement. A bad velocity measurement, for example coming from slip, results in wrong controller, or combination of controllers, being used. This can result in instability. The theory also assumes constant speed, that the control is must faster than the change in speed. Since the tram can accelerate about $1 m/s^2$, this is not fulfilled. It is therefore theoretically not proven, only likely, that the controller is stable also during an acceleration or deceleration. The latter is actually not such a big limitation as it may seem, since this limitation exists for the whole H_∞-theory.

---

[2] Note that these conditions are only sufficient, not necessary. They can therefore not be used to show that the controller is not robust

The solution presented here has the advantage that it switches softly between the controllers, provided the measurement signal is not too bad. This decreases the need to let the non-active controllers track the active controller(s). The big drawback is the difficulty to prove robustness. It can, as earlier mentioned, not be done in general, but has to be done for each combination of controllers. If one controller is replaced, the procedure has to be repeated. Therefore, another solution is desired.

6.3.2 Hard switching

The possibility to prove robustness is greatly improved if the simpler solution of hard switching is used, i.e. always only one controller is used at the time. This solution causes two new problems:

- If switching between two controllers takes place too often, it can result in instability.
- The switching between two controllers causes bumps in the control signal.

The first problem arises if a specific speed is used to switch between two controllers. If the tram runs in exactly this speed, it can cause switches every sample. This effect can be removed using hysteresis in the switching. This means that when the tram is accelerating, one speed is used to switch, and when it is decelerating another, lower, speed is used. The difference between the switching speeds should be chosen large enough to remove the possibility for switching often because of bad measurement signals or because of a swinging speed, i.e. if the repeatedly accelerating and decelerating when the driver desires a constant speed. It should on the other hand not be chosen too large, as two controllers has to be stable in this area, increasing the robustness demands by increasing the velocity-range in which each controller must be robust.

The second problem is common for all types of control when different controllers, or a manual control signal, is used. This can be solved with a tracking-scheme, letting the non-active controller track the active, in order for them to have the same control signal when the switch is made. In this case, the control signal is already very bumpy because of the bad measurement signal combined with an existing feed-through in the controller. This makes the control signal hard to track, and simulations have also shown that tracking for this system actually increases the difference between the control signal of the two controllers. This arises from the fast swinging control signal combined with that the controllers do not differ very much. The last fact makes the control acceptable with no tracking at all.

6.4 Results

The resulting controller is a gain-scheduled controller. A typical transfer function is shown in figure Figure 6.2. It does not resemble any simpler type of controller, like PID, and this will make it difficult to find rules for tuning the parameters, because lack of physical understanding of the states in the controller. Compare with a PID-controller, where it is well-known how the behavior of the closed-loop system is affected by a change of the individual controller-parameters, and tuning therefore can be made more easily.

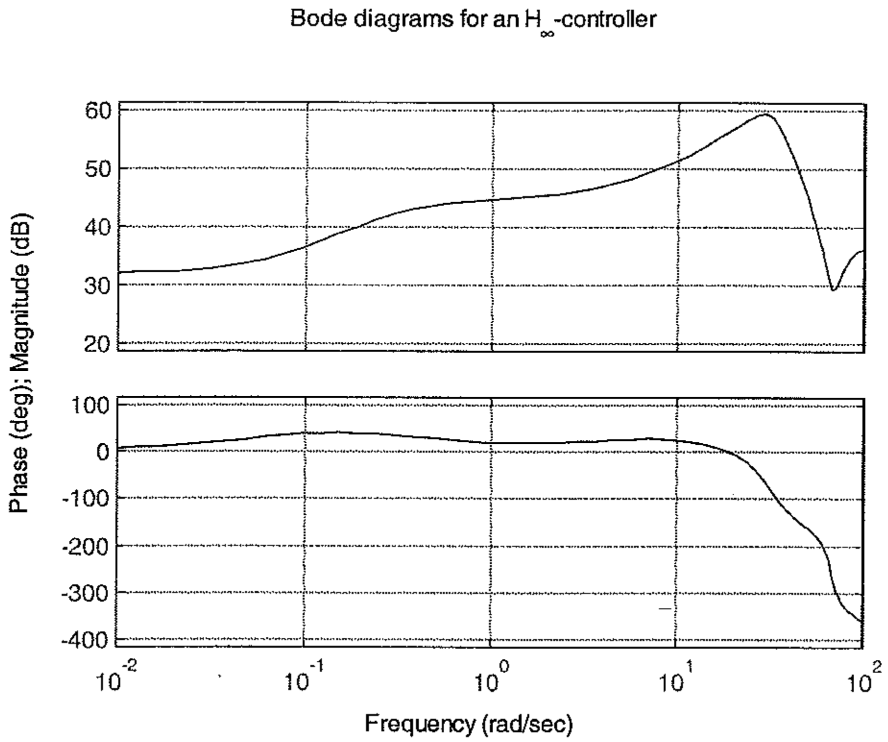

Figure 6.2: One controller, robust for 14.7-20 m/s

7 The resulting controller

The closed-loop behavior for the system controlled with the resulting H_∞-controller in the time –domain looks as Figure 7.1 shows. The simulation is here compared to the system controlled with a PD-controller, tuned to be robust for a specific speed.

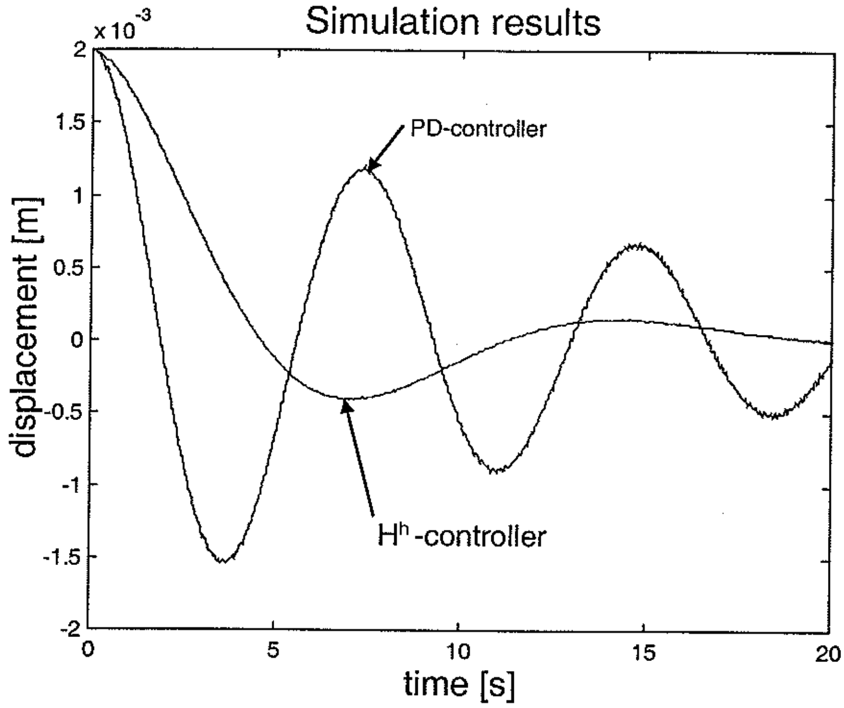

Figure 7.1: Simulation results

The simulations are done for a well-centered tram (small λ) running on dry rail (large c) starting with a 2 mm displacement.

In this plot it is seen that for most normal applications the H_∞-controller would be preferred because of its well-damped behavior. After a time of only 6 seconds, the H_∞-controller has damped the system to be kept within an error of 0.4 mm, while the PD-controller uses more than 20 seconds for the same task. But the H_∞-controller uses considerably longer time to get to reach the first zero-crossing of the bogie. However, this normally good control performance is in this application not the desired one, as explained in the controller specification.

To measure how good the controller suits this system, a wavelength for the appearing sinus-motion was calculated according to equation (7.1), where t_{c0} is the time to the first zero-crossing of the bogie.

$$\lambda_{cont} = 4 \cdot v_0 \cdot t_{c0} \quad (7.1)$$

This is the wavelength that in the beginning had a desired value of 20 m in the controller specification. When later the "speed" for a controller is mentioned, this value is used as a measurement. This value is for the H_∞-controller very much larger, and strongly dependent on speed. The wavelength's dependence of the speed for the H_∞-controller is plotted in Figure 7.2, for the case with full friction and for the case with a friction 30% of full friction. Also here it can be seen that the wavelength is not close to the wished, having longer and speed-dependent wavelength.

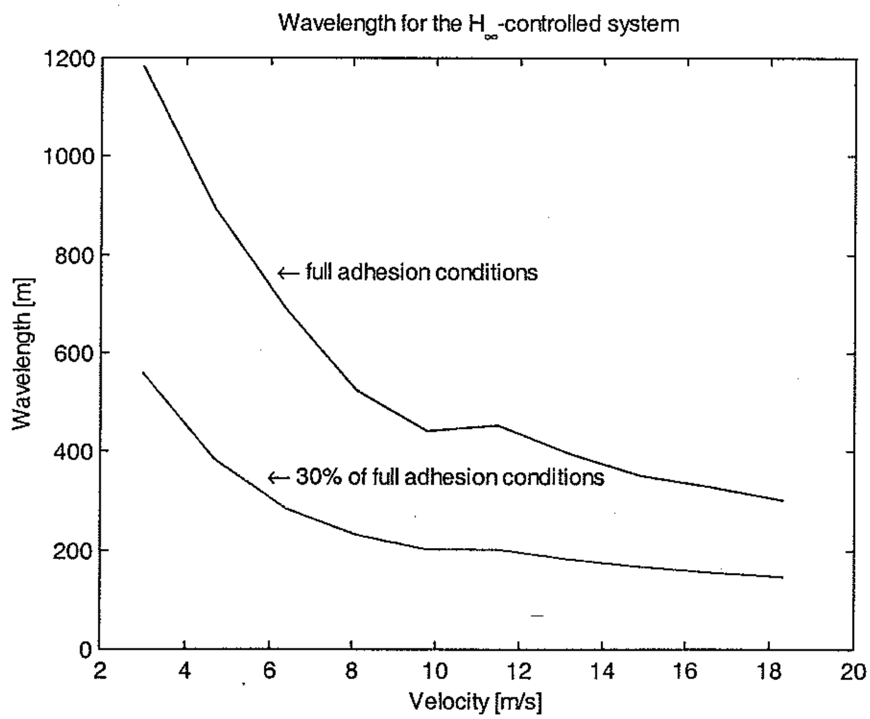

Figure 7.2: The wavelength's dependence of speed

The H_∞-controller also gives a very well-damped behavior. This is not desired, because it would make higher wear on the specific points on the wheels where the tram runs centered. This wish is untypical for normal systems to be controlled, for which the H_∞-method was developed. The obvious question then arises; can the H_∞-controller be made faster and less damped?

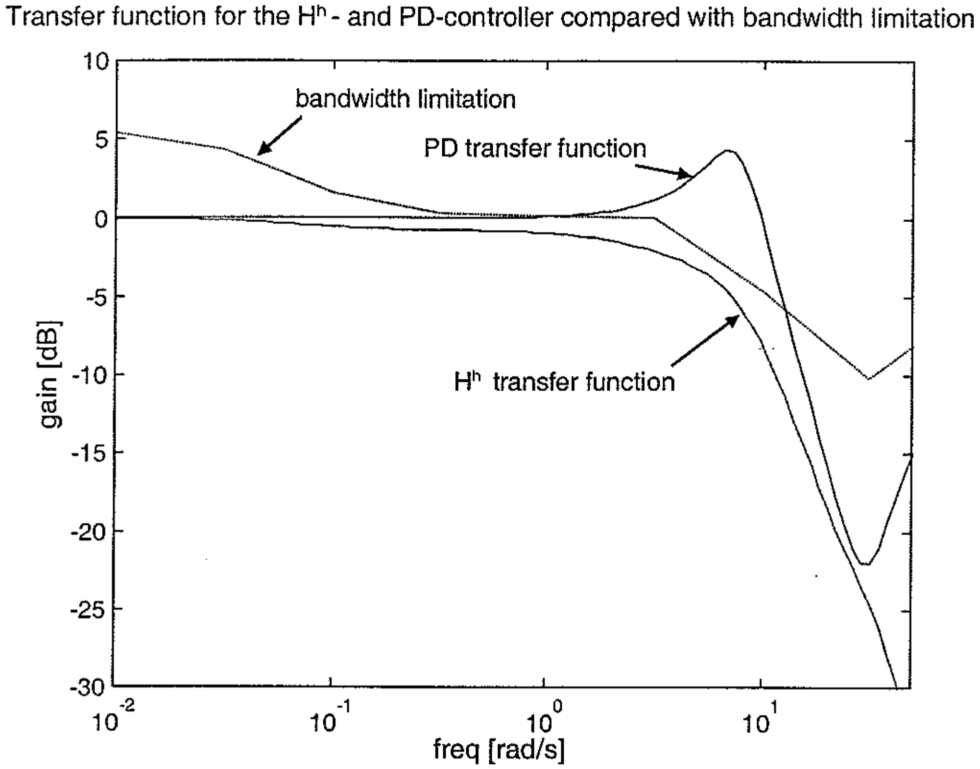

Figure 7.3: Transfer function For H_∞- and PD-controlled system

In Figure 7.3 the closed-loop transfer function using H_∞ and PD-controller, respectively, is plotted, together with the inverse of the multiplicative uncertainty. From this plot, two conclusions can be drawn:

- The H_∞-controller can not be made much faster, since it could then not be proven robust anymore with H_∞-analysis. The same conclusion is drawn if the KS-function is plotted together with the additive uncertainty. The damping can only be made smaller if speed is reduced in change, since removing damping corresponds to a transfer function >0 dB for certain frequencies. This can be done while keeping robustness if the controller is slowed down considerably, but not otherwise.
- The PD-controller is not proven robust using this method. This was done by DaimlerChrysler Research using another method, and this method does not fit into the H_∞-design.

How can the H_∞-controller be slower than the PD-controller? The explanation for this is the conservatism of the robustness demand used in the H_∞ analysis. This comes from the theorem, which says, in words:

A closed-loop system is stable if all the participating systems are stable and the H_∞-norm of their product is smaller than or equal to one

It is not difficult to find examples of stable systems not fulfilling this demand, which shows the conservatism of the theorem. The conservatism in the control design makes the H_∞-controller unnecessarily slow, which explains how the PD-controller can be faster.

As a summation of this, the H_∞ control results in a well-damped system with practically no overshoot, but this is unfortunately not desired in this, unusual, control-goal. This problem is hence not a suitable problem to use H_∞ control

8 Conclusions on controller design

In the H_∞ controller design done in this thesis, the method has shown some advantages and some disadvantages:

Disadvantages:

1. *Difficulties when gain-scheduling*
   This can make for example a PD-controller better if an uncertain process parameter is measurable, since it is more suitable for gain-scheduling.
2. *Not possible to include all uncertainties in the uncertainty description*
   In this case the large measurement errors can not be described as uncertainties, which removes the big advantage with the method: the mathematically proven robustness.
3. *Problems with unusual control goals*
   In this case the wished low damping makes the method not suitable.

Advantages:

1. *The closed-loop shaping*
   Makes it possible to automate controller design.
2. *Quick way of analyzing what can be done with a controller*
   Through analysis of the uncertainties it can quickly be checked if it is possible to make a robust controller with a certain bandwidth. Note that this can only be done one way, it is not possible to show that a certain bandwidth can not be reached. The robust PD-controller is a good example of this.

It is obvious from the previous sections that H_∞ control is not very suitable for this process, but how should a process look where it is a good idea to use an H_∞-controller?

- *Good process model*
  If a good process model exists, all the uncertainties can be precisely expressed as parameter changes, avoiding the approximation of the uncertainties done here which must use the idea that if some robustness margin is used, the system should still be robust. This

approximation removes the possibility of mathematically proven robustness.

A good process model also makes tuning unnecessary, the result from controller design can be used directly without tuning. Tuning is hard to do with an H_∞-controller, where the physical understanding for the controller-states is hard to get. Furthermore is the guaranteed robustness lost when tuning, if not earlier. Therefore an H_∞-controller is not tuned for each specific tram as a classical controller is.

- *Low-order process*

  The high order of this process forces for uncertainty-calculation using optimization, with no guarantee of really finding the worst-case. If the process is of much smaller order, this calculation can be performed analytically, finding the uncertainty as a function of the process parameters
- *No large nonlinearities or time-dependent uncertainties*

  If very big nonlinearities exist, these must be described as large uncertainties, which makes the controller very slow. For this process, the measurement errors limit the possibility to describe all uncertainties in the design since a standard description would be very large.
- *No uncertain variable measurable*

  The H_∞-theory is hard to use in combination with gain-scheduling. If gain-scheduling can be used, the controller can be made faster. If, on the other hand, no uncertain variable can be measured, gain-scheduling can not be used, which lowers the possibilities for the PD-controller to be gain-scheduled, thus improving the H_∞-controller in comparison.

The conclusion of this work is that H_∞-control has proved to have difficulties with processes with non-normal control goals. But the H_∞-analysis was found to be a fast and easy way to analyze what can be done with a controller, and to check if a controller is robust. For this process, the method is not suitable, and can not reach enough bandwidth to match the wished behavior of the closed-loop system.

Bibliography

[1] Ludyk, G:
Theoretische Regelungstechnik 2.
Springer-Verlag 1995

[2] Morari, M and Zafiriou, E:
Robust process control.
Prentice-Hall 1989

[3] Skogestad, S and Postlethwaite, I:
Multivariable feedback control, Analysis and Design.
John Wiley & Sons 1996

[4] Zhou, K; Doyle, J.C and Glover, K:
Robust and Optimal Control.
Prentice-Hall 1996

[5] Chiang, R.Y and Safonov, M.G:
H^{∞} Synthesis Using a Bilinear Pole Shifting Transform.
Journal of Guidance, Control and Dynamics, Vol. 15 No. 5, 1992, pp. 1111-1117.

[6] Chiang, R.Y and Safonov, M.G:
Robust Control Toolbox User's guide.
MathWorks, Inc 1997

[7] Kwakernaak, H:
Robust Control and H_{∞}-optimization – Tutorial Paper.
Automatics, Vol. 29 No.2, 1993, pp. 255-273.

[8] Ackermann, J:
Robuste Regelung, Analyse und Entwurf von linearen Regelungssystemen mit unsicheren physikalischen Parametern.
Springer-Verlag 1993