Multiple Model Multiple Controller Adaptive Control

R.M.R. Bruns

DCT 2008.097

Master’s thesis

Committee: prof.dr.ir. M. Steinbuch (supervisor)

dr. A.Y. Pogromski

dr.ir. A.G. de Jager (coach)

dr.ir. J.J.M. van Helvoort (coach)

Technische Universiteit Eindhoven
Department of Mechanical Engineering
Control Systems Technology Group

Eindhoven, September, 2008
Preface

This master’s thesis is the result of my graduate project for the Control System Dynamics group at the Technische Universiteit Eindhoven. It is based upon studies conducted from March 2008 and until 2009. I would like to thank Jeroen van Helvoort for his advice, his support and the introduction to the Unflasified Control principle during the first nine months. Secondly, I would like to thank Bram de Jager and prof. M. Steinbuch for their support, discussions and supervision. Special thanks go out to Bram for reviewing this thesis. Furthermore I would like to thank my fellow students at the university for the discussions during the breaks and the pleasant time at the university. Finally I would like to thank my family for their support and advice.

Roel Bruns
September 2008
Abstract

From literature, two control methods are found to be complementary. When the advantages of the Robust Multiple Model Adaptive Control and Unfalsified control are merged together a new Multiple Model Multiple Controller Adaptive Controller should be able to deal with uncertain systems with a large quantity of realizations, like cheap mass produced systems. This new Multiple Model Multiple Controller Adaptive control method includes a supervisory control, based on the Unfalsified Control principle and robust control techniques to synthesize the multiple controllers.

The supervisory control method evaluates, by only using the input and output signal of the plant, the performance of the different controllers if they had been in the loop during the time the data was collected, this is known as fictive reference. These multiple controllers are synthesized by constructing multiple, uncertain models from identification data of multiple PATO realizations and applying the complex-$\mu$. Then, the controllers are analyzed by using mixed-$\mu$ since uncertain parameters are included in the different models and these are real valued. This results in a non-optimal controller design method where the multiple controllers do not differ enough to perform the supervisory control scheme. One of the multiple uncertain models can completely or almost completely be controlled by almost all controllers, this is not desired. By using mixed-$\mu$ to synthesize the controllers one should be able to distinguish the controllers from each other based on performance, this can be done in the future.

The multiple complex-$\mu$ controllers are not used in the supervisory control scheme after all because of these small differences. Therefore, for each measured PATO realization a $H_\infty$ controller is designed. By using the stabilizing complex-$\mu$ controller for the full uncertain model domain as initial controller the supervisory control scheme should find a well performing $H_\infty$ controller for a single PATO realization, in simulation and experiment. However, the
fictive reference and fictive error representing the multiple controllers depends on the inverse controllers. Therefore, in simulation, the same controller is observed as the best performing controller for each PATO realization because for this inverse controller the gain at the reference frequencies is the smallest. This is of course not the desired behavior.

From simulation, multiple solutions are observed. A well performing, stabilizing controller is initially switched in the loop and remains in the loop. Or, initially a destabilizing controller is used whereafter the system remains unstable. Thirdly, when started with the stabilizing controller that does not suffice the performance criterium (the complex-\(\mu\) controller for the full uncertain model domain), a destabilizing or a well performing controller is switched in the loop. Which case occurs depends on the used PATO realization. A solution to unfalsify the correct controllers, besides a different calculated fictive reference and fictive error, can be found in the change of the falsification bound. A controller dependant falsification bound that takes the transient response into account when a switch is needed for example.

Based on the experiments on the PATO set-ups, switching is cumbersome due to the computational burden of the new controller number. When started with a satisfying controller, stabilizing and well performing, the controller remains in the loop and the PATO realization is controlled well. For other initial controllers, the system becomes unstable at a switch.
Samenvatting

Naar aanleiding van een literatuur studie is gebleken dat de voordelen van twee regel strategieën complementair zijn. Indien deze voordelen van Robust Multiple Model Adaptive Control en Unfalsified control worden samengevoegd in een nieuwe, Multiple Model Multiple Controller Adaptive regelaar moet het mogelijk zijn om systemen, die een grote onzekerheid herbergen te regelen zoals in de massa productie waar nauwkeurigheid minder van belang is. Deze nieuwe Multiple Model Multiple Controller Adaptive regel methode bevat een schakelmechanisme dat de prestatie van de regelaars evalueert, gebaseerd op het Unfalsified Control principe, en het bevat robuust control technieken om de verschillende regelaars te ontwerpen.

Het schakelmechanisme evalueert, door enkel gebruik te maken van het ingang en uitgang signaal van het system, de prestatie van de verschillende regelaars die gehaald zou zijn indien deze regelaars in de lus geschakeld waren tijdens de meting, dit wordt ook wel fictieve referentie genoemd. De verschillende regelaars zijn ontworpen door meerdere onzekere modellen uit de identificatie data van verschillende PATO realisaties te maken en de complex-$\mu$ theorie toe te passen. Deze regelaars worden hierna geëvalueerd door middel van mixed-$\mu$, omdat de onzekerheid van de parameters in de modellen reëel waardig zijn. Dit resulteert in een regelaar ontwerp dat niet optimaal is. De verschillende regelaars verschillen niet genoeg om het schakelmechanisme toe te passen. Een van de onzekere modellen kan hierdoor helemaal of deels geregeld worden door bijna elke ontworpen regelaar, dit is niet gewenst. Door het gebruik te maken van regelaars die zijn ontworpen met behulp van de mixed-$\mu$ synthese, zou het mogelijk moeten zijn om het schakelmechanisme toe te passen en de verschillen in de regelaars te herkennen in de prestatie, dit kan wellicht in de toekomst uitgevoerd worden.
De verschillende regelaars, ontworpen met complex-$\mu$, zijn uiteindelijk niet gebruikt in het schakelmechanisme door deze kleine verschillen. Daarom is voor elke PATO realisatie een $H_\infty$ regelaar gemaakt. Door deze regelaars, gecombineerd met de stabiliserende regelaar voor de volledige onzekerheid van het model als initiële regelaar, in het schakelmechanisme te gebruiken zou een goede $H_\infty$ regelaar voor het PATO systeem gevonden moeten worden, zowel in simulatie als experiment. Echter, de fictieve referentie en fictieve fout behorend bij de verschillende regelaars hangen af van de inverse regelaars. Hierdoor, in simulatie, wordt telkens dezelfde regelaar als beste bezien voor elke PATO realisatie. Dit is niet het gewenste resultaat.

Meerdere resultaten kunnen uit de simulaties volgen, afhankelijk van de te regelen PATO realisatie. Als initieel een stabiliserende en goed presterende regelaar in de regellus geschakeld is, zal deze ook de regelaar blijven en voldoen aan de gestelde prestatie eisen. Als echter een destabiliserende regelaar in de lus geschakeld wordt, zal het systeem instabiel blijven. De derde mogelijkheid voor initiële regelaar is de, op voorhand bedachte optie, stabiliserende regelaar die niet aan de prestatie eis voldoet (de complex-$\mu$ regelaar voor de volledige model onzekerheid). Dit resulteert in twee mogelijkheden. Een instabiele regelaar wordt ingeschakeld en het systeem wordt instabiel of een stabiliserende en goed presterende regelaar is uiteindelijk het resultaat. Afhankelijk van de PATO realisatie is een van deze twee gevallen van toepassing. Een oplossing om de goede regelaar niet te falsificeren, buiten een andere manier om de fictieve referentie en fictieve fout berekening uit te voeren, kan gezocht worden in een falsificatie grens die per regelaar een andere waarde of verloop kan hebben en tijdens het experiment of simulatie opnieuw berekend wordt.

Tijdens experimenten op de PATO’s is gebleken dat het schakelen problemen oplevert door de rekenlast van de beste regelaar, dit nummer wordt niet meer correct berekend. Als gestart wordt met een stabiliserende en goed presterende regelaar dan blijft deze regelaar in de regellus en de PATO realisatie is naar wens geregeld. Voor andere initiële regelaars wordt het system instabiel tijdens het schakelen naar een andere regelaar.
Contents

Preface i
Abstract iii
Samenvatting v
List of Figures xi
List of Tables xiii

1 Introduction 1

2 Literature 3
  2.1 Multiple Model Adaptive Control 4
  2.2 Multiple Controller Adaptive Control 8
  2.3 Comparison 11

3 Motivation 15
  3.1 Objective 16

4 Supervisory control 17
  4.1 Unfalsified Control 17
  4.2 Fictitious Reference 18
  4.3 Performance Index 19
  4.4 Cost-Detectability 21
  4.5 Stability 21
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Controller Design</td>
<td>23</td>
</tr>
<tr>
<td>5.1</td>
<td>Generalized plant</td>
<td>24</td>
</tr>
<tr>
<td>5.2</td>
<td>Uncertainty</td>
<td>27</td>
</tr>
<tr>
<td>5.3</td>
<td>Synthesizing the controllers</td>
<td>29</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Full uncertainty controller, lower bound</td>
<td>29</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Single plant controllers, upper bound</td>
<td>33</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Multiple models, multiple controllers</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>Simulations</td>
<td>39</td>
</tr>
<tr>
<td>6.1</td>
<td>Identification</td>
<td>40</td>
</tr>
<tr>
<td>6.2</td>
<td>Controller design for single plant</td>
<td>42</td>
</tr>
<tr>
<td>6.3</td>
<td>Single plant - Single controller</td>
<td>45</td>
</tr>
<tr>
<td>6.4</td>
<td>Performance index in simulation</td>
<td>46</td>
</tr>
<tr>
<td>6.5</td>
<td>Complete controlled simulation</td>
<td>49</td>
</tr>
<tr>
<td>6.6</td>
<td>Conclusion and recommendations</td>
<td>54</td>
</tr>
<tr>
<td>7</td>
<td>Experiments</td>
<td>57</td>
</tr>
<tr>
<td>7.1</td>
<td>Single Plant - Single Controller</td>
<td>57</td>
</tr>
<tr>
<td>7.2</td>
<td>Complete controlled system</td>
<td>59</td>
</tr>
<tr>
<td>7.3</td>
<td>Conclusion and recommendations</td>
<td>62</td>
</tr>
<tr>
<td>8</td>
<td>Conclusion</td>
<td>63</td>
</tr>
<tr>
<td>9</td>
<td>Recommendations</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>69</td>
</tr>
<tr>
<td>A</td>
<td>FRF Measurements</td>
<td>71</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Multiple Model Adaptive Control schemes. ........................................ 5
4.1 Unfalsified Control scheme .............................................................. 18
4.2 Fictitious reference creation. .............................................................. 19
5.1 General control configuration. .............................................................. 24
5.2 Control scheme for $\mu$-design ............................................................ 24
5.3 Inverse of the weights to shape the desired behavior. ......................... 25
5.4 The four used weight functions. ........................................................... 26
5.5 Controlled plant with uncertainty structure. .......................................... 28
5.6 $\mu$-bound. .......................................................................................... 30
5.7 Controller for full uncertainty controller. .............................................. 30
5.8 Open-loop for full uncertainty controlled system ................................. 31
5.9 Nyquist diagram for the controlled system. .......................................... 32
5.10 Sensitivity and weight. ....................................................................... 32
5.11 Complementary sensitivity and weight. .............................................. 33
5.12 General control configuration. .............................................................. 33
5.13 Controller synthesized for a single plant. ............................................ 35
5.14 Open-loop for single plant controlled system. .................................... 35
5.15 Performance achieved by the different controllers. ............................. 37
5.16 Dividing the model ............................................................................. 38
6.1 PATO experimental setup. ................................................................. 39
6.2 Frequency response measurements. .................................................... 40
6.3 Fitted models of the frequency response functions. ............................... 41
6.4 Frequency response and fitted model. .......................... 41
6.5 Typical controller for a single plant. .......................... 42
6.6 Open-loop of the controlled frequency response and controlled model. .......................... 43
6.7 Sensitivity and weight function for a single controlled system. .......................... 43
6.8 Nyquist diagram of the controlled system and mismatched controllers. .......................... 44
6.9 Single plant Single Controller simulation, step response and third order reference signal. .......................... 45
6.10 Single plant Single Controller simulation error, matching and mismatched controller. .......................... 45
6.11 Inverse controller. .......................... 47
6.12 Performance index of a controlled system. .......................... 47
6.13 Inverse controllers. .......................... 48
6.14 Complete controlled system, error and performance index. .......................... 50
6.15 Used and unfalsified controllers. .......................... 50
6.16 Matching controller, error and performance index. .......................... 51
6.17 Used and unfalsified controllers during a stable, matching simulation. .......................... 52
6.18 Error and performance index of proper switched system. .......................... 52
6.19 Controllers of proper switched system. .......................... 53
6.20 Initially destabilizing controller used, error and performance index. .......................... 53
6.21 Initially destabilizing controller used, used controllers and unfalsified controllers. .......................... 54

7.1 Single Plant - Single Controller, output and error. .......................... 58
7.2 Single Plant - Single Controller, output and error (2). .......................... 58
7.3 Complete controlled system, output and error. .......................... 59
7.4 Complete controlled system, falsification and performance index. .......................... 60
7.5 Complete controlled system, output and error (2). .......................... 60
7.6 Complete controlled system, falsification and performance index (2). .......................... 61
7.7 Controller to be switched into the loop. .......................... 61

A.1 Identification scheme. .......................... 71
A.2 Measured sensitivity. .......................... 72
A.3 Measured process sensitivity. .......................... 72
LIST OF FIGURES

A.4 Resulting plant. .................................................. 73
List of Tables

2.1 RMMAC and UC ........................................... 13
5.1 $\mu$-values ................................................... 36
6.1 Simulation results ........................................... 55
Chapter 1

Introduction

The core of mass production is to manufacture as cheap and as fast as possible products or components that are the equal to each other. However, in general a cheap and fast produced product is not developed with great accuracy. This results in products that are the same, can do the same task, but will be different from each other. Uncertainty is introduced in the products or components. With this uncertainty, a difference in dynamical behavior can occur for certain mass produced systems. Controlling these systems can be done by identifying each system and synthesize a controller for that individual system which will require a lot of effort for a mass produced system. A second option is to identify enough systems to get a representation of the systems that can occur and construct a nominal system with an uncertainty. This last option is less time consuming but because of the uncertainty, the controller for a single system will probably not be the optimal controller. This loss of performance can be reduced by an adaptive control scheme of multiple controllers.

This work focusses on the control of a system with uncertainty by dividing the systems uncertain model in multiple uncertain models, synthesizing controllers for each model and by using a supervisory control scheme to select the best controller for a particular plant. A literature study is performed first in chapter 2, followed by the motivation in chapter 3. Thirdly the supervisory control will be explained in chapter 4, then the controllers are designed in chapter 5. Simulations are performed in chapter 6, whereafter the experiments on an experimental set-up are discussed in chapter 7. Resulting in a conclusions and recommendations in chapter 8.
Chapter 2

Literature

Early adaptive controllers, like the Model Reference Adaptive Controller (MRAC), were developed to overcome the control problem for time varying systems or systems with a large parametric uncertainty. However, in Rohrs et al. (1985) it is shown that classical systems can become unstable in the presence of plant disturbances, sensor noise and high-frequency unmodeled dynamics. Later, adaptive controllers partially evolved in multiple model techniques that deal primarily with multi-input multi-output systems systematically and are able to subdivide the parameter uncertainty into smaller parameter subsets to achieve better performance.

On the other hand, robust control theory exists that can deal with unmodeled dynamics and real parametric uncertainty. A combination of an adaptive and a robust controller could lead to a better performing controller by, for example, decreasing the parameter uncertainty interval of which a single controller developed with the robust control theory has to deal with and using adaptive control to switch between multiple robust controllers Fekri (2005). The recent developments on the so-called robust adaptive controllers all have three elements: multiple controllers, a supervisory part that chooses the best suitable controller to insert in the feedback loop, and some parameter or performance estimation. This estimation occurs in two different ways, direct or indirect. The indirect estimation of the parameters is achieved by model identification techniques while the direct approach is based on the performance of the controllers. The design and use of a controller depends, among others, on the plant assumptions and the plant identification that needs to be done to satisfy these assumptions.
In this research, two approaches of the identification element are discussed and compared. First Multiple Model Adaptive Control, which uses the indirect estimation, then Multiple Controller Adaptive Control will be outlined, including a recent developed control method with direct performance estimation, Unfalsified Control.

## 2.1 Multiple Model Adaptive Control

As mentioned before, the indirect Multiple Model Adaptive Control uses model identification techniques to validate models, and inserts the controller corresponding to the best suitable model. Model identification techniques are used to identify parameters with a large uncertainty interval. To create the control output, the input of the system to be controlled, one can use a switch to include a controller (Fig. 2.1(a)) or one can multiply the multiple controllers with corresponding probabilities (Fig. 2.1(b)) by using a posterior probability estimator (PPE), which will be explained in the following. The earlier multiple model adaptive control switching system of Narendra in, Narendra & Balakrishnan (1994) and Narendra & Balakrishnan (1997), uses a switch to include one of the controllers in the feedback loop. The created models are all identical in structure, with different initial conditions. Each identification model corresponds to a controller, yielding a stable system when connected to the plant. It is proven that the system is globally stable for any arbitrary switching sequence, under the condition that the intervals between successive switches have a nonzero lower bound. A performance monitoring function is used to improve the transient behavior of the controlled system. The proposed monitoring function consists of the square of the identification error and its integral, multiplied with design parameters to tune the proportion of steady-state and transient performance. Switching is based on this performance index of the models, while stability is assured.

A SISO supervisory controller is developed in Morse (1996, 1997). From a candidate controller set, a controller is selected by comparing the estimation errors of the model transfer functions and by placing the candidate controller corresponding to the smallest estimation error in the feedback loop. The used controllers are fixed gains, in combination with a non-adaptive integrator, that stabilize the system. To overcome chattering the supervisor is forced to remain at the controller for a certain amount of time. Due to persistent noise, the system may never
stop switching. To avoid this unwanted behavior the adaptation is simply turned off after a while to force switching to stop. Similar to the previous discussed controller in Narendra & Balakrishnan (1994), a controller is chosen on basis of the estimation error of the several models (multiestimator).

Another switching based multiple model adaptive control method is the safe switching adaptive controller Anderson et al. (2001). This adaptive feedback starts with a stabilizing controller and switches to a new model and controller when the Vinnicombe distance criterion is less or equal to a certain bound. The Vinnicombe distance is the maximal chordal distance between two models or controllers over every frequency. Switching is safe when the bound of the Vinnicombe criterion, with candidate controller, is larger than the stability margin. Each controller is build to perform well, with respect to the performance requirements, in combination with its model. But it could be that the performance of a different combination of model and controller results in a better performance than the designed combination, due to model uncertainty and disturbances. Therefore, performance needs to be evaluated after each switch. This method holds for linear, slowly time-varying systems and LTI controllers.

A new robust adaptive control method is first presented in Fekri et al. (2004a) and in Fekri et al. (2004b) and will be outlined in the following. With the use of the robust control theory, controllers can be designed with a well-defined design method. Fekri (2005) used the

![Diagram](image)

Figure 2.1: Multiple Model Adaptive Control schemes: (a) based on switching and (b) based on probabilities.
mixed-µ synthesis to construct multiple controllers for multiple, smaller parameter uncertainties such that better performance is achieved. The combination of these multiple parameter uncertainties results in the total parameter uncertainty. A controller designed for a smaller uncertainty interval yields better performance compared to a single controller designed for the total, larger parameter uncertainty. The new idea in the so-called, Robust Multiple Model Adaptive Control (RMMAC) theory of Fekri (2005), is the combination of model identification and controllers based on the mixed-µ. One of the main advantages is the prediction of the performance benefit by using the RMMAC instead of a traditional robust controller. First, a traditional robust controller is made to overcome the total parameter uncertainty, prior plant information is needed. This controller, called the “global non-adaptive robust compensator” (GNARC), is the best robust, non-adaptive controller and will be the lower-bound on the performance of the new designed robust adaptive controller. Clearly this case implies that there is, at least, one controller for the whole parameter uncertainty. Secondly, an upper-bound on the obtainable performance is created by constructing “fixed non-adaptive robust compensators” (FNARC). Here, for each parameter the best performance is computed for the case that the real parameter is known, hence implicitly assuming infinite number of models and controllers.

One of the key elements of RMMAC is the choice of the number of models used to estimate the parameter(s). While a large number of models results, in general, in a better performance it also results in high complexity and more computational effort. Fewer models can be used to keep the complexity reasonable, this implies a larger parametric uncertainty for each model. “The Kalman filters may have difficulty yielding rapid and precise model identification, i.e. slow posterior probability convergence and, perhaps, erroneous model identification.” This problem is fixed by using Extended Kalman filters that not only estimate the plant state variables in real time but one (or more) uncertain parameters as well, resulting in faster posterior probability convergence. The number of models to be used depends on the system and the desired performance, see Fekri et al. (2006a). One can just choose a number of models and compute the potential benefit of the adaptive controller or one can divide the parameter uncertainty in a more systematic manner, like the %FNARC method Fekri (2005), Athans et al. (2005), Fekri et al. (2006b). Using this method enables the user to choose the potential resulting performance level as a percentage of the FNARC resulting in an, in advance, unknown number of models.
Besides these two described methods to compute the number of models, one can use several other methods, which will not be outlined here, depending on the performance requirements and the system to control.

When the number of models and their intervals are computed, the Kalman filters are fitted to this interval. These Kalman filters construct the innovations (the difference between the predicted and observed measurement, also called residual), leading to the posterior probability estimator (PPE). The PPE calculates, in real time, the posterior conditional probability ($P_1$ to $P_N$, see Fig: 2.1(a)) that each model generates the plant output. The “global” state-estimate is then obtained by the probabilistic weighting of the local state-estimate. One of the posterior probabilities, corresponding to the model “closest” to the correct hypothesis, needs to converge to 1, even though the actual plant parameter differs from the real parameter. Proper design of the Kalman filters results in correct model identification of this PPE. To construct a Kalman filter, a nominal value of the parameter needs to be determined. This value is computed by using the Baram Proximity Measure (BPM), in contrast to the more ad-hoc method of choosing the nominal value at the middle of the parameter interval. The BPM is a measure for the “stochastic distance” between the residual of the Kalman filter and the nominal Kalman filter residual. Identification and control is separated, the Kalman filters do not drive the controllers directly and are less susceptible to errors in the local state estimates.

Concluding, the RMMAC predicts in advance the potential benefit of the adaptive controller dependant on multiple models with respect to the GNARC and FNARC, constructs multiple mixed-$\mu$ synthesis based compensators (the so-called local non-adaptive robust compensators (LNARC)) for the same parameter interval as the (systematically) chosen Kalman filters. These Kalman filters are build by using the BPM and yield residuals for the PPE. This posterior probability estimator determines the influence of each LNARC. In other words, the Kalman filters estimate the interval where the uncertain parameter is situated, leading to a feedback loop with the corresponding LNARC. The main advantage of subdividing the parameter interval is to decrease the parameter uncertainty, resulting in better performance for each individual mixed-$\mu$ synthesized controller (LNARC) with respect to the GNARC. Several simulation results, see
Fekri (2005), Fekri et al. (2007, 2006a), show a superior behavior of the RMMAC controller with respect to the classical designed robust controller, the GNARC.

2.2 Multiple Controller Adaptive Control

One of the first attempts to decrease the need for a priori information of the plant to be controlled, compared to the Model Reference Adaptive Control (MRAC) theory, is made in Martensson (1985). The research is driven by the idea of relaxing the four classical assumptions on necessary a priori information for adaptive control, i.e. the degree of the plant is known, the plant is minimum phase, the relative degree is known and the sign of the instantaneous gain is known. In Martensson (1985), a start with the simple idea of attaching the plant to a box of integrators, each with its own input and output, has been made. Static feedback is applied to the augmented plant, a dynamic feedback is achieved by searching (in a pre-defined manner) through the coordinates of the controller’s state space to stabilize the system. One of the main assumptions made is existence of a stabilizing controller, this assumption will be maintained in recently designed MCAC strategies. Although there can be an exhaustive on-line search which makes this method of control in practice useless, as also stated by the writer himself and in Fu & Barmish (1986), it is of value because of the idea that, in theory, it is possible to control a system adaptively with a certain amount of a priori information.

Further research based on previously discussed idea of control and in the search of relaxing known a priori information assumptions, in Fu & Barmish (1986) an adaptive stabilizing method without a minimum phase and known sign of the high-frequency-gain assumptions is developed. The presented control methodology starts with a compensator gain matrix and develops a monitoring function with the use of the output signal. This monitoring function “decides” when to switch to a next gain matrix. This leads to a final controller, gain matrix, which stabilizes the system and no more switching is applied. The feedback can be seen as piecewise linear time-invariant and uses a finite number of switches to achieve the desired result. Although certain assumptions are relaxed, two more are included, an upper bound on the order of the plant is known and the possible plant belongs to a compact set of possible plants. With these assumptions the controllers can be derived and Lyapunov stability is proven.
A promising multiple controller adaptive control method is the recently developed Unfalsified Control method. This control method uses a different approach compared to, for example, the classical robust control. Unfalsified Control, first introduced in Safonov & Tsao (1994), searches for a controller that suffices the pre-defined performance requirements: “It should be just as simple to directly validate the control laws from the input/output data with specified performance bounds instead of first validate a plant model with specified uncertainty bounds”.

This adaptive control method is plant-assumption-free and only uses input/output measurement data, it does not try to estimate or identify the actual system. The only assumption made is feasibility of the control problem. Starting with a candidate controller set, which (if continuous) consists of infinitely many controllers, an unfalsified controller determines whether a candidate controller satisfies the performance requirements based on the available measurement data. A candidate controller that fails to meet these user-defined performance requirements is falsified and will no longer be part of the candidate controller set used in the next falsification iteration with new measurement data. This iterative process finally results in at least one controller that is good enough to satisfy the performance requirements, if and only if feasibility holds.

In Safonov & Tsao (1994) the Unfalsified Control definition is given as:

**Definition 1** A controller $K$ is said to be falsified by measurement information if this information is sufficient to deduce that the performance specifications $(r, y, u) \in T_{\text{spec}} \forall r \in \mathbb{R}$ would be violated if that controller were in the feedback loop. Otherwise, the control law $K$ is said to be unfalsified.

Although there is not (yet) a well-defined step-by-step design principle, multiple researchers designed an Unfalsified Controller. The Unfalsified Control theory includes conditions to assure stability and convergence. Where starting with only plant input and output measurement data, a candidate controller set and performance specifications, the main restrictions apply on the candidate controller set. Different questions concerning stability and falsifying of controllers arise when the initial idea of Unfalsified Control is examined. One of those questions is; how to determine, without inserting a controller in the feedback loop, if the closed loop with that controller meets the stability and performance specifications or not? To overcome this problem a fictitious reference signal is created, from a given set of past measurement input/output
data and a candidate controller. The fictitious reference signal for this candidate controller is a hypothetical reference signal that would have produced exactly the measured data had the candidate controller been in the feedback loop with the unknown plant during the entire time period over which the used measured data is collected Paul (2005). This fictitious reference signal is used to construct a fictitious error signal, depending on the performance requirements. A desired closed loop dynamics system can be used as a weighting function for the fictitious reference signal, see Van Helvoort et al. (2007), Wang & Safonov (2005).

Based on the fictitious error signal, a cost-function determines whether the cost exceeds the performance requirement or not and thus whether the controller is falsified or not. The key element to assure this desired convergence is the cost-detectability property of the user defined cost-function, see Wang (2005). As shown in Stefanovic (2005), other sufficient criteria to assure stability and convergence are Stably Causally-Left-Invertible (SCLI) candidate controllers with a finite number of switches, assuming feasibility of the control problem. The two most difficult conditions to achieve on the choice of the candidate controller set are feasibility and the SCLI property, the latter of which is also needed to compute a fictitious reference signal and to provide a uniquely present input based on past and present output of the controller Paul (2005).

A special form of Unfalsified Control is implemented in Van Helvoort et al. (2007), based on the idea of Cabral & Safonov (2004). The continuous candidate controller set is approximated by an analytic expression of an ellipsoid. The minimum-volume outer bounding ellipsoidal Unfalsified Set is an approximation of the intersection of the ellipsoid and two parallel half-spaces. These parallel half-spaces are constructed by combining the performance requirements with the controller structure. This allows the user to lower the computational burden of continuous adaptation compared to other previously used algorithms.

An algorithm to select one of the feasible controllers is the $\epsilon$-hysteresis algorithm Morse et al. (1991). This algorithm has a build-in hysteresis that prevents the algorithm from switching, i.e. changing parameters, too quickly. It starts with an initialization where the first controller and the bound $\epsilon$ are chosen. At each time step the cost-function is evaluated. A controller is switched in the feedback loop when the cost function is smaller then the cost-function of the current
controller minus the bound $\epsilon$. Else, the current controller is maintained in the loop, see also Wang & Safonov (2005).

2.3 Comparison

Two promising methods of both the MMAC and MCAC are the recently developed RMMAC and Unfalsified Control respectively. When the advantages of both methods are combined, it could result in an interesting new controller. Although ideally all the advantages of both methods are combined without any of the disadvantages, it is not a realistic objective. The resulting dis(advantages) should depend on the users desires and priorities. The (dis)advantages will be described in the following, whereafter the main (dis)advantages of both controllers are summed in table 2.1.

Starting with the RMMAC, its ability to compute the performance gain of the adaptive controller with respect to a classically designed robust controller is a huge advantage. The designer can predict the performance and take the performance gain and the complexity of the RMMAC design in consideration. The method relies on a systematically defined design methodology for controllers, i.e. the mixed-$\mu$ synthesis, allowing the designer to systematically develop the controllers. This systematically defined construction of controllers inevitably results in the need for system identification or plant knowledge before constructing a controller. For several systems, model identification could be a huge disadvantage and difficult to overcome.

Being designed with the mixed-$\mu$ synthesis implies that the controllers are robust for the designed parameter uncertainty. As a result the RMMAC is robust for the whole parameter uncertainty. Unfortunately, extension to more than two uncertain parameters appears to be hard with the RMMAC method. The computational burden might increase to a situation where it leads to erroneous identification of the models. This is an unwanted characteristic of the identification part of the adaptive controller. Although a model-mismatch might still yield a stable system, due to the sufficient condition of the mixed-$\mu$ upper bound inequality Fekri (2005), the performance will decrease. When a model is used that does not suffice the mixed-$\mu$ upper bound inequality, i.e. excessive model mismatch occurs, the resulting system is unstable.
Because the RMMAC uses multiple models, where the performance gain by using more or less models can be computed in advance, the performance requirements highly determine the number of models. This dependency can result in an excessive number of models, which can lead to more computational burden. Because the performance requirements and the number of models and controllers can be computed in advance, the user can choose and predict if it is worthwhile developing a RMMAC for a particular system.

Like RMMAC, an unfalsified controller is a robust controller. One candidate controller is by itself not robust, but by regarding the whole set of candidate controllers, Unfalsified Control is robust when feasibility is assumed and easy to implement. The plant-assumption-free characteristic of the Unfalsified Control method enables the user to control a system by only measuring its input and output (regardless if it is open or closed loop data), this in contrast to the RMMAC method. This characteristic leads to a controller for an unknown system instead of a controller for one, two or multiple parameter uncertainties. The candidate controller set “only” needs to suffice the feasibility assumption and needs to be SCLI. But Unfalsified Control lacks a theory for developing controllers satisfying these two conditions, a check on feasibility of the candidate controller set is not available at the moment. It is of great importance to select the candidate controller set and its structure in a proper manner, like the RMMAC uses the mixed-$\mu$ synthesis. The main difficulties of Unfalsified Control design arise in creating the candidate controller set to assure feasibility and cost-detectability of the cost-function (SCLI property of the candidate controllers is necessary).

The fictitious reference signal can be a “tool” to evaluate the candidate controllers directly but it can also result in an incorrect reference signal if pole-zero cancelation did occur during the computation, which can lead to erroneous falsification of candidate controllers. But because of this fictitious reference and the unfalsified principle, only controllers that are not yet falsified with respect to a performance requirement can be included in the feedback loop. But an included controller that is not the final controller is unfalsified while switched in the feedback loop. To improve the transient behavior of the system an exponential term can be included in the performance criteria.
One of the main disadvantages of the RMMAC is the computational burden, specially when multiple uncertain parameters and a high number of models are present. An unfalsified controller can have infinitely many controllers (continuous candidate controller set) described with an analytic Unfalsified Set, like the Ellipsoidal Unfalsified Control presented in Van Helvoort et al. (2007), without causing any computational problems.

<table>
<thead>
<tr>
<th></th>
<th>RMMAC</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>Performance gain adaptive over robust</td>
<td>Plant-assumption-free</td>
</tr>
<tr>
<td></td>
<td>is known in advance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Based on a systematical defined design</td>
<td>Open/Closed loop data can be used to learn</td>
</tr>
<tr>
<td></td>
<td>line for controllers</td>
<td>Multi Dimensional</td>
</tr>
<tr>
<td></td>
<td>Multiple model benefit known in advance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model-mismatch often still remains stable</td>
<td>Analytic, fast computation</td>
</tr>
<tr>
<td></td>
<td>Robust</td>
<td>Robust</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td>Identification/Model needed</td>
<td>Selection of candidate controllers/structure</td>
</tr>
<tr>
<td></td>
<td>Multi dimensional very hard (if not possible)</td>
<td>Cost detectability of cost function (SCLI of controllers)</td>
</tr>
<tr>
<td></td>
<td>Model-mismatch, poorer performance or</td>
<td>No check on feasibility of candidate controller set</td>
</tr>
<tr>
<td></td>
<td>instability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computational burden if N is large or &gt; 3-D, erroneous identification</td>
<td>Mostly, numerical, computational burden</td>
</tr>
<tr>
<td></td>
<td>No global asymptotic stability results in</td>
<td>Disturbance-rejection not clear in advance</td>
</tr>
<tr>
<td></td>
<td>near future</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 3

Motivation

The last two decades, combinations between the classical robust control theory and adaptive systems led to new control schemes. These schemes consist of multiple models and/or multiple controllers, like the Robust Multiple Model Adaptive Control (RMMAC) and Unfalsified Control (UC) theory. These two methods in particular complete each other very well when the advantages of both controllers are constructed into a new controller. It is hard to deal with multiple uncertain parameters in the RMMAC method, while an unfalsified controller assumes a completely unknown plant and only uses the input and output of the plant to be controlled. The plant-assumption-free characteristic of UC makes it applicable for a wide range of systems. On the other hand, the design of the candidate controller set is based on the designer’s knowledge of the system and needs to suffice previously discussed properties. It is not based on a well-defined design methodology like the RMMAC, which uses the mixed-$\mu$ synthesis to develop the controllers. Before even starting to develop a controller it is of great value to know the performance gain in advance, then deciding if it is worth pursuing a controlled system with an increased complexity. The RMMAC design contains this property.

When the advantages of both methods are merged together, the resulting controller will be, in the best case, a robust adaptive controller with plant-assumption-free characteristic that is capable of dealing with multiple parameter uncertainties and predict the performance gain in advance with respect to a single robust controller by constructing the controllers in a well-defined manner. To include only the advantages of both previous developed controllers and discard all the disadvantages will not be realistic to achieve. An attempt is made to include the main advantages
of both methods in a new controller while disadvantages are still present. These disadvantages will be of minor importance in our setting.

3.1 Objective

After a literature study, focussed on the robust multiple model adaptive control and multiple controller adaptive control, an assessment has been made of these control design methods, both in theory and application. A new control scheme containing RMMAC Unfalsified Control elements should be designed, simulated and tested on an experimental set-up, where the beneficial advantages of both method are tried merged together.
Chapter 4

Supervisory control

In this chapter the control principle of the supervisory control is explained. Based on chapter 2, a combination between the Robust Multiple Model Adaptive Control (RMMAC) and Unfalsified Control (UC) control methods is designed. An adaptive multiple model multiple controller control scheme is created which maintains plant-assumption-free and only the input and output need to be measurable. The new controller is designed to be robust for certain multi-dimensional uncertainties and uses a well defined controller design method. The multiple controllers are constructed by using the robust control theory, applied on multiple models, in the next chapter. A supervisory control concept is used to switch one of these controllers in the control loop and is able to achieve a predefined performance. The UC principle is used to evaluate and switch the best, or at least a sufficient, controller in the loop.

4.1 Unfalsified Control

Like discussed in chapter 2, Unfalsified Control is a relative new control approach. It is a multiple controller adaptive control method where a set of controllers $\mathcal{C}$, with candidate controllers $C_i$, is tested and evaluated without inserting them in the loop. Based on a performance index that quantifies the specifications, the candidate controllers are either falsified or unfalsified. Only the unfalsified candidate controllers can be chosen to control the system. The only assumption made is feasibility, at least one of the candidate controllers can satisfy the unfalsification criterium, i.e. the performance index. Therefore the controllers are constructed by using the Robust Control principle, then feasibility is guaranteed.
Different criteria can be used to determine the falsification of a controller. Here a controller is constructed in frequency domain with certain performance specifications while the performance index is a time domain criterium. This will be discussed in section 4.3. The control scheme is presented in Fig. 4.1, where the basic blocks of the Unfalsified Control principle are drawn.

**4.2 Fictitious Reference**

One of the advantages of the Unfalsified Control principle is the possibility to test the controllers with respect to a defined performance index without inserting them in the loop, by creating a fictitious reference signal. A fictitious reference signal is a hypothetical signal \( \tilde{r}_i \) that would have produced exactly the same measured data \((u_p, y_p)\) had the controller, \( C_i \), been in the feedback loop with the unknown plant during the entire time period over which the measured data \((u_p, y_p)\) were collected Paul (2005). To collect the measured data one candidate controller \( C_{ic} \in C \) is already inserted in the closed loop, hence this fictitious reference should equal the actual reference signal. This reference can be computed by using Eq. 4.1 and a block scheme is shown in Fig. 4.2.

\[
\tilde{r}_i = C_i^{-1}u_p + y_p \tag{4.1}
\]

For every controller \( i \), this reference signal is computed by using the measured input \( u_p \) and output \( y_p \).

A candidate controller, \( C_i \), needs to be "causally left-invertible" to guarantee that the fictitious reference \( \tilde{r}_i \) exists and is unique, Stefanovic (2005). The candidate controllers need to suffice
4.3. PERFORMANCE INDEX

Experimental data

Figure 4.2: Fictitious reference creation.

this criterium besides the assumed feasibility. This reference will and can only use the past data of the input and output signals and it is used to construct the fictitious error signal, $\tilde{e}_i$.

These fictitious signals are commonly used in a performance index, where the performance of the system with the candidate controllers is evaluated. This will be discussed in the following section, 4.3.

An incorrect fictitious reference signal is computed when the plant is highly time invariant, with respect to the sampling time. In most systems this will not be a problem. The possibility of pole-zero cancelation in the fictitious reference signal could be a more troublesome property. Pole-zeros cancelation in the fictitious reference can lead to instability if the actual reference is slightly different from the fictitious signal. Then the cost-detectability, see section 4.4, property is not fulfilled when the sensitivity is used in the cost-function $\frac{\tilde{e}}{\tilde{r}}$ is stable due to the pole-zero cancelation while $\frac{e}{r}$ is actual instable, this example is given in Paul (2005). An other problem of the fictive reference is that it depends on the used reference during the measurement, which can result in unfalsifying the incorrect controller, this can be seen in chapter 6.

4.3 Performance Index

The criteria on which switching is based is set in the performance index. This index is computed by using the fictitious signals online. This results in a time domain specification on which the
falsification takes place. If the performance index of a candidate controller exceeds a fixed value, \( \Delta_{\text{p1}} \), the candidate controller is falsified.

In most motion systems proper tracking is desired. This specification is used to derive the performance index of Eq. 4.2.

\[
J = \frac{\int_0^t |\tilde{e}_i| \, dt}{a + \int_0^t |\tilde{r}_i| \, dt}
\]  

(4.2)

Where \( \tilde{e} \) and \( \tilde{r} \) are respectively the fictitious error and reference, computed in section 4.2. A constant, small value (a) is added to the denominator because the initial value of the denominator would have been zero, resulting in an infinite performance index, this is clearly not wanted. Only the value at which the candidate controller is falsified needs to be adjusted. A falsification bound is used to falsify controller based on this performance index, in this thesis a constant value of the falsification is chosen while falsification bound that converges to a final constant value takes the transient more into account. A forgetting factor can be added in J to weight the past error, input and reference signals as done in Stefanovic et al. (2004). When dealing with varying systems including this forgetting factor will result in a more accurate performance index and thus proper falsification and control.

A second criterium for switching can be added to overcome large leaps in the input signal. The difference between the actual performance index, i.e. the performance index of the controller in the control loop, and the lowest performance index may not exceed a predefined constant value of \( \Delta_{\text{p2}} \). This criterium can be added to achieve a better and more smooth response, it ensures that the switching between the best and actual controller at a certain time does not result in an exorbitant large input signal \( u_p \), which would occur when a controller with higher gain is switched into the loop in combination with a relative large error. A different method to overcome this problem is a switching algorithm that switches to a better controller based on the performance index that is still "close" to the actual controller. This will usually result in more switches and a transient which is less desirable.
4.4 Cost-Detectability

The function $J$ can only distinguish a good from a bad controller when it is cost-detectable. It can detect instability while the plant is still unknown and therefore model-mismatch instability problems are circumvented. The function $J$ and the candidate controller set $C$ is cost-detectable with respect to input $r$ when $J$ is bounded and the stability of the system is unfalsified by the previous collected measurement data, see Paul (2005) and Wang (2005) for a more detailed description of this cost-detectability. From the above stated condition, it can be seen that the cost-detectability property is independent of the plant. It depends on the cost function and the candidate controller set, $J$ and $C$.

4.5 Stability

Van Helvoort et al. (2007) proved that bounded-input bounded-output (BIBO) stability is guaranteed if the adaptive control problem is feasible, the candidate controllers are SCLI and evaluated with a cost-detectable cost-function and if the number of controller switches is limited. The control design of chapter 5 has dealt with the feasibility and SCLI property of the controllers and assured the cost-detectability property of previous defined function $J$. Limited number of switches is achieved by creating the candidate controllers based on performance specifications and switching is needed when the cost-function, that represents these specifications, exceeds the fixed value $\Delta_{\mu_1}$, then the candidate controller with the lowest cost-function value is switched in the loop. Because a limited number of candidate controllers have been used and once a candidate controller is falsified it can not become unfalsified again, a limited number of switches is guaranteed.
Chapter 5

Controller Design

To apply the previous explained supervisory control, see chapter 4, multiple controllers are designed. The used design principle is model based and therefore first identification is used to obtain a nominal plant model with uncertainty, which will be discussed in the following chapter. Only the input and output signals are needed to apply the supervisory control principle, these are the only signals used to obtain the nominal plant model with uncertainty region. An infinite number of controllers can be constructed for each individual system without uncertainty, this results in controlled systems with high performance since there is no uncertainty, an upper bound is achieved. The lower bound for the performance is achieved by synthesizing a controller for the full uncertainty region by using the robust control theory. When the uncertain model is divided in multiple models with each a smaller uncertainty region a controller with performance between the upper and lower bound can be synthesized.

Starting with the general control configuration, see Fig. 5.1, the different blocks are observed and discussed. Then a controller for the full uncertainty region is created. Thirdly, a large number of controllers is synthesized for a large number of plants without uncertainty. Then, the multiple controllers are designed with a performance between the upper and lower bound.
5.1 Generalized plant

The generalized plant is based on measurements of multiple systems of the same type, see section 6.1. From these measurements, a plant with uncertain parameters is derived. The nominal plant combined with the weight functions, result into the generalized plant. Performance specifications are set by including the input and output weight functions from and to each input and output, see Fig. 5.2. Here two inputs and two outputs are used to synthesize the controllers by using the standard algorithm for DK-iteration. The four weights are designed to bound the sensitivity (S), process sensitivity (PS), control sensitivity (KS) and the complementary sensitivity (T).

Figure 5.2: Control scheme for $\mu$-design.
5.1. GENERALIZED PLANT

The desired performance depends on the system to be controlled, because the resulting controller is used in a motion system proper tracking of the reference, disturbance rejection, small measurement noise influence and stabilization is wanted. Typically, the desired sensitivity is small at low frequency and equals 1 at high frequency. While the complementary sensitivity equals 1 at low frequency and is small at high frequency. The used algorithm tries to minimize all four functions 5.1 to 5.4 combined with the weight functions to be smaller than 1. Hence the weight functions need to be designed properly to achieve the desired behavior of the controlled system. Having two inputs and two outputs, each desired behavior is influenced by two weight functions, one input and one output function.

\[
S = \frac{e}{r} = (1 + KP)^{-1} \quad (5.1)
\]
\[
PS = \frac{e}{d} = P(1 + KP)^{-1} \quad (5.2)
\]
\[
KS = \frac{u}{r} = K(1 + KP)^{-1} \quad (5.3)
\]
\[
T = \frac{u}{d} = KP(1 + KP)^{-1} \quad (5.4)
\]

The output weight on the error is used to describe the desired sensitivity behavior while the output weight on the plant input reflects the complementary sensitivity, \(W_e\) and \(W_u\) respectively. Two two input weight functions are then used to shape the four bounds further. The process

![Graphs showing Sensitivity, Process Sensitivity, Control Sensitivity, and Complementary Sensitivity](image)

Figure 5.3: Inverse of the weights to shape the desired behavior.
sensitivity consists at low frequent of high amplitude, therefore the disturbance filter is shaped to allow the process sensitivity (and the complementary sensitivity as a result of that) to get an amplitude larger than 1 at low frequencies. To achieve the desired performance, the sensitivity bound, the input weight on the reference is maintained at a gain of 1. In Fig. 5.3, the bounds resulting from the weight functions can be seen.

In Fig. 5.4, the four used weight functions can be seen. These are used to synthesize the controller, whereafter the performance weight $W_e$ is reshaped to meet the mixed-$\mu$ criteria, see section 5.3.1. In Eq. 5.5 to Eq. 5.8, the weight functions are set.

\[
W_e = \frac{s + 0.5s + \omega_S}{s + 0.001\omega_S} 
\]
(5.5)

\[
W_u = \frac{s + 0.5s}{0.001s + \omega_T} 
\]
(5.6)

\[
V_r = 1 
\]
(5.7)

\[
V_d = 0.001 
\]
(5.8)

The bandwidth is adjusted by changing the cut off frequency of the output filters, $\omega_S$ and $\omega_T$. Because the functions $S$, $PS$, $KS$ and $T$ are shaded by each an input and an output weight function, it is not possible to shape these independently. Here, it is tried to maintain the performance bound.

Figure 5.4: The four used weight functions.
(\(W_e\) and \(V_r\)) on the sensitivity while designing the weight functions. One of the advantages of this model based control strategy is the possibility to predict and evaluate the performance before including the controller into the loop. The performance is measured by using the bandwidth of the weight filter of \(S, W_e\).

5.2 Uncertainty

The uncertainty block, \(\Delta_r\), is used to cover the difference between the nominal plant and all other possible plants. Here, this uncertainty block describes real parameter variations of the transfer functions of multiple observed plants. This leads to a structured real-valued uncertainty block, in Eq. 5.10 a typical uncertainty block for \(n\) uncertain parameters can be seen, with \(\delta_i \in [-1, 1], \forall i\).

\[
\Delta_r = \begin{pmatrix} 
\delta_1 & 0 & 0 & \cdots & 0 \\
0 & \delta_2 & 0 & \cdots & 0 \\
0 & 0 & \delta_3 & \cdots & 0 \\
\vdots & \vdots & 0 & \ddots & 0 \\
0 & 0 & 0 & 0 & \delta_n 
\end{pmatrix} \quad (5.9)
\]

This uncertainty block is diagonal, i.e. the uncertainty of a single parameter does not influence other parameters. Secondly, it can be seen that the single uncertain parameters are non-repeated which results from the uncertain description, each parameter combination can occur since the parameter uncertainties are not linked.

When the earlier described controllers for the multiple model and the lower bound controller are evaluated, the system and its uncertainty is rearranged. The performance specification \(\Delta_p\), that is used to analyze the controller(s) for robust performance, see Fig. 5.5, is included in the uncertainty description, resulting in \(\Delta_m\).

\[
\Delta_m = \begin{pmatrix} 
\Delta_r & 0 \\
0 & \Delta_p 
\end{pmatrix} \quad (5.10)
\]
The structure of $\Delta_p$ depends normally on the performance and stability requirements of the resulting controller. However, because it is chosen to use complex-$\mu$ to synthesize the controllers whereafter a mixed-$\mu$ criterium is used to analyze the performance, $\Delta_m$ is a full complex square block when synthesizing the controllers.

Complex-$\mu$ algorithms use a DK-iteration, where first a controller is synthesized for an initial scaling matrix $D$ by minimizing the peak value over the frequency ($H_{\text{init}}$). Secondly the $D$-matrix is used to minimize the maximum singular value of the controlled plant ($N$) at each frequency. The computed $D$-matrix is used to synthesize a new controller whereafter the iteration is carried out until the desired performance is achieved or the peak value does not longer decreases.

Mixed-$\mu$ is used to adopt the real values and structure of the uncertainty in the controller synthesis like real-$\mu$ but still incorporates a complex valued uncertainty. This is typically done by applying a DGK-iteration where G-matrices include the real values. However this algorithm is not used because of the computational complexity and therefore not discussed any further. In 
, the DGK-procedure is discussed in more detail.
5.3 Synthesizing the controllers

It should be noted that the resulting controllers, except the upper bound, are not optimal according to the real valued uncertainty but the controllers do suffice the specified performance criteria. This is done by analyzing the controllers with a mixed uncertainty. The following theory and method of design is discussed by using the fourth order system that will be used for simulation and experiments as well. This system will be discussed in more detail in section 6.1 where the experimental set-up is identified and an uncertain model has been made. Here this uncertain model is assumed to be known and consists of a transferfunction with multiple uncertain, real, parameters (Eq. 5.11). This leads to an order of 3 in the previous discussed uncertainty block \( \Delta_r \).

\[
H = \frac{b_0}{s^4 + a_3 s^3 + a_2 s^2}
\]  

(5.11)

Three different types of controllers are constructed, a single controller for the nominal plant and the full uncertain domain, an infinite number of controllers for each system in the uncertain set and finally for a certain number of uncertain models that fill up the total uncertain domain an equally number of controllers is designed.

5.3.1 Full uncertainty controller, lower bound

To compute a lower bound on the achievable performance, with this non optimal method, the complete parameter range of the three uncertain parameters is used. Then, weight filters are designed to perform the optimization of the complex-\( \mu \) algorithm, see Fig. 5.4. This algorithm tries to find a controller such that the maximum singular value of the controlled system \( N(K) \) is smaller than 1 regarding complex perturbations of \( \Delta_m \leq 1 \). Robust performance, including robust stability, is assured, see Eq. 5.12.

\[
\mu_{\Delta_m}(N(j\omega)) < 1, \forall \omega
\]  

(5.12)

Applying this theory with the standard algorithm in Matlab results a controller for the complex-\( \mu \) case. With a possible small shift in weight function cut off, \( \omega_S \) and \( \omega_T \), and applying a mixed-\( \mu \) analysis a final \( \mu \)-value of 0.9707 which suffices the criterium and robust performance is achieved. The \( \mu \)-bound is plotted in Fig. 5.6, where the peak value does not exceed this value of 0.9707. The peak around 45 Hz is due to the resonance of the system while the first peak is due to the phase crossing the \(-180^\circ\) in the closed loop.
First the resulting controller is discussed, what can be seen is that the uncertainty is too large to achieve a high bandwidth near the resonance. The controller, see Fig. 5.7 uses a lead part near the bandwidth to gain phase, combined with a decrease in amplitude and a wide notch at the resonance frequency to decrease the possible resonance, suggesting that the resonance is still
5.3. SYNTHESIZING THE CONTROLLERS

Critical for the bandwidth, it is however only used to decrease the overall gain in that frequency region. At high frequency the gain is decreased rapidly to decrease the effect of high frequent noise.

In Fig. 5.8, it can be seen that the anti resonance part in the controller is not sufficient to cancel the complete resonance, which can not be done by a controller because the resonance peak is uncertain in amplitude and frequency. This figure shows the complete set of possible plants. The bandwidth of the open-loop system is in a range of 0.3 to 1 Hz. The bandwidth can probably be increased by changing the weight functions to allow more control effort.

According to the open-loop and the nyquist diagram of Fig. 5.9, the controlled system is stable and will satisfy the performance criterium, maximum of 6 dB overshoot in sensitivity, for all possible plants. There is still room for improvement since the 6 dB line is not touched, this is probably due to other limiting transferfunctions, as the control sensitivity or the process sensitivity. The sensitivity and complementary sensitivity can be seen in Fig. 5.10 and Fig. 5.11, where it is shown that both criteria satisfy the specified performance according to the weight...
filters and therefore the $\mu$-value is indeed smaller than 1. Like mentioned earlier, the lowest value of the bandwidth is 0.5 Hz, resulting in a lower bound value of .5. These last figures are plotted with a smaller grid of models in the uncertain domain to enhance the clearness.

![Nyquist diagram for the controlled system.](image1)

Figure 5.9: Nyquist diagram for the controlled system.

![Sensitivity and weight.](image2)

Figure 5.10: Sensitivity and weight.
5.3. SYNTHESIZING THE CONTROLLERS

5.3.2 Single plant controllers, upper bound

The single plant controllers are designed to obtain the performance upper bound by synthesizing a controller for a single plant in the set of all possible plants, this results in a regular control problem for systems without uncertainty. This is done by designing $H_\infty$ controllers with similar criteria as the previously discussed $\mu$-controllers. By controlling each plant individually, this would lead to an infinite number of controllers, the uncertainty of one single plant becomes zero and Fig. 5.1 can be replaced by Fig. 5.12.

Figure 5.11: Complementary sensitivity and weight.

Figure 5.12: General control configuration.
By incorporating crucial weight filters in $P$ the desired controller will be synthesized. Since the system is a fourth order system and will look like a -2 slope, resonance, -4 slope in a bode plot, with perfect cancelation of the resonance the maximum bandwidth can, theoretically, be placed at each frequency. However, a frequency response function will almost never result in a perfect representation of the system near the resonance and this resonance might slightly change when using (in frequency and/or amplitude), therefore the bandwidth of the controlled system is placed before the resonance of the system. The final controllers will still try to cancel the resonance, but a misfit will not have disastrous consequences.

An $H_{\infty}$-controller will try to minimize the maximum singular value at each frequency. Which is incorporated in Eq. 5.13, see chapter 8 of Skogestad & Postlethwaite (2005).

$$\|F_l(P, K)\|_\infty = \max_{\omega} \bar{\sigma}(F_l(P, K)(j\omega))$$

(5.13)

Instead of calculating infinite number of controllers to obtain the upper bound, a grid is constructed and a large number of plants in the total uncertain model are controlled by using $H_{\infty}$-control. For each parameter 9 values are evaluated, equally divided among the range of the parameter, resulting in $9^3$ number of plants and matching controllers. All the plants are controlled by using the same weight filters and performance criteria (bandwidth at 25 Hz), only the resulting value of the maximum singular, gamma, value differs a bit between 0.98 and 1. The influence of each of the three parameters is evaluated by comparing the differences in $\gamma$ value. The idea of, eventually dividing the parameters with the largest $\gamma$ value deviation into multiple smaller parameter domains, is performed although the difference in deviation between the three real valued parameters is very small.

A typical controller synthesized for one of the $9^3$ plants, or one PATO system, can be seen in Fig. 5.13. The controller will add some phase lead at the bandwidth while the resonance is completely canceled out by using a skewed notch. This leads to a nice and smooth controlled system which open-loop is plotted in Fig. 5.14.
5.3. SYNTHESIZING THE CONTROLLERS

Figure 5.13: Controller synthesized for a single plant.

Figure 5.14: Open-loop for single plant controlled system.

5.3.3 Multiple models, multiple controllers

To increase the performance of the total uncertain system, it is divided into multiple smaller models, representing smaller sets of possible plants. From the identification of multiple PATO realizations a uncertain model is created, with three uncertain real structured parameters. By assuming that, for example, the stiffness of a PATO realization can occur in combination with the resonance, frequency and amplitude, of an other PATO realization. This leads to a conservative model which enlarges the set of plants for which this control scheme will work properly. Then,
two out of three uncertain parameters are divided into multiple regions which leads to, in this case, 8 models with a smaller uncertainty. Here, the parameters are divided in equal parts (see Fig. 5.16 without regarding the possible performance achievements like the Baram Proximity Measure of Fekri (2005).

The controller synthesizes for these multiple models is equivalent to the previous described controller synthesizes of the controller for the full uncertain region. The only difference is the smaller set of plants in a single uncertain model because of the multiple models and corresponding multiple controllers. The resulting $\mu$-values for the eight plants and eight corresponding controllers are displayed in table 5.1, with M representing the models and C the controllers.

<table>
<thead>
<tr>
<th></th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
<th>M8</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0.9872</td>
<td>0.9869</td>
<td>1.0286</td>
<td>1.0241</td>
<td>1.0781</td>
<td>1.0733</td>
<td>1.1449</td>
<td>1.1397</td>
</tr>
<tr>
<td>C2</td>
<td>0.9876</td>
<td>0.9873</td>
<td>1.0346</td>
<td>1.0301</td>
<td>1.0856</td>
<td>1.0807</td>
<td>1.1540</td>
<td>1.1488</td>
</tr>
<tr>
<td>C3</td>
<td>1.1014</td>
<td>1.1013</td>
<td>0.9777</td>
<td>0.9774</td>
<td>1.0015</td>
<td>0.9987</td>
<td>1.0509</td>
<td>1.0471</td>
</tr>
<tr>
<td>C4</td>
<td>1.1016</td>
<td>1.1015</td>
<td>0.9780</td>
<td>0.9778</td>
<td>1.0062</td>
<td>1.0033</td>
<td>1.0565</td>
<td>1.0526</td>
</tr>
<tr>
<td>C5</td>
<td>1.2209</td>
<td>1.2208</td>
<td>1.0736</td>
<td>1.0734</td>
<td>0.9642</td>
<td>0.9639</td>
<td>0.9896</td>
<td>0.9865</td>
</tr>
<tr>
<td>C6</td>
<td>1.2212</td>
<td>1.2210</td>
<td>1.0737</td>
<td>1.0736</td>
<td>0.9645</td>
<td>0.9642</td>
<td>0.9934</td>
<td>0.9904</td>
</tr>
<tr>
<td>C7</td>
<td>1.3397</td>
<td>1.3394</td>
<td>1.1752</td>
<td>1.1750</td>
<td>1.0459</td>
<td>1.0458</td>
<td>0.9563</td>
<td>0.9561</td>
</tr>
<tr>
<td>C8</td>
<td>1.3399</td>
<td>1.3397</td>
<td>1.1754</td>
<td>1.1752</td>
<td>1.0460</td>
<td>1.0459</td>
<td>0.9566</td>
<td>0.9564</td>
</tr>
</tbody>
</table>

What can be seen is that the diagonal terms are all under and close to 1, which is desired. The problem can be seen when the off-diagonal terms are evaluated. There are values smaller than the diagonal terms, this is clearly due to the design method, i.e. synthesizing controllers with full unstructured complex uncertainty (complex-$\mu$) and analyzing the controllers for a structured mixed uncertainty (mixed-$\mu$). In Fig. 5.15 the performance of the three control types are evaluated by calculating the, lowest, 0 dB crossing of the sensitivity. What can be seen is that the multiple model controllers performance is, like predicted, between the other two methods. Still, there is still a lot of room for improvement until the upper bound is reached, primarily by using a mixed-$\mu$ synthesizes.
5.3. SYNTHESIZING THE CONTROLLERS

Like mentioned above, the non-optimal design method results are troublesome results, certain controllers are better for a plant than the corresponding controller. This is also caused by the conservatisme of the $\mu$ synthesizes and analysis. The only guarantee is that the matching controller, w.r.t. one of the multiple models, is stabilizing and suffices the defined performance criteria. However, when controlling a single plant, it is possible that more than just one controller can control the plant while achieving the desired performance, or is even better than the controller for that specific model set containing the single plant. This can be seen in Fig. 5.16 as well, where the eight controllers encircle the domain for which they do suffice the performance criteria. The matching model is totally encircled by its controller but a controller can control other plants properly as well. It should be noted that this figure is used to clarify and visualize this observed behavior and is not measured or based on the calculated $\mu$-values.

This behavior leads to problems when simulating and experimenting with the full control scheme, including the supervisory control. From table 5.1 it can be seen that there are "better" controllers than the actual matching controller designed for the model, not to mention a single PATO realization. Secondly, the difference between the $\mu$-values is not large which becomes a problem because it can be the case that some plants in the complete set of plants could be controlled by all the controllers without failing the performance specifications because the $\mu$-value is an upper bound. The line between a controller that does and a controller that fails to suffice the criteria is too small to be noticed in simulations and experiments. Therefore,
the supervisory control scheme is tested in simulation and experiment by using only the $H_\infty$ controllers for the measured PATO realizations and the controller for the full uncertain domain. This leads to a control scheme that should be able to find the suitable $H_\infty$ controller for a PATO realization. This can be seen in the following chapter 6.
Chapter 6

Simulations

In order to test the previous described control method, first simulations are carried out, secondly this multiple model multiple control adaptive control method is tested on an experimental set-up. In order to correspond the simulation results with the experimental results, measurement data is used to construct the simulations. This includes the identification of multiple fourth order plants to achieve a nominal plant with an uncertainty region that is large enough to desire multiple controllers to achieve increased performance. The identification is discussed in section 6.1, where 10 models are constructed from 10 realizations of the PATO system, see Fig. 6.1. For each model a controller is synthesized in section 6.2, these 10 controllers are used in the complete supervisory control system in section 6.5. After a single PATO realization is controlled by its matching controller in 6.3 and the performance index is evaluated in section 6.4.

Figure 6.1: PATO experimental setup.
6.1 Identification

Ten different realizations of the PATO system are identified by using a standard method of obtaining the frequency response function of the observed system. Random white noise is added to the control output of a low bandwidth controller, both signals and the output of the system are measured. Then the frequency response function can be identified with the measured signals by computing the sensitivity and process sensitivity. By dividing the process sensitivity over the sensitivity, the frf of the plant is created. The resulting frequency response function of the 10 measured PATO systems can be seen in Fig. 6.2. A more detailed description of the identification method is given in the appendix A.

![Frequency response measurements](image)

Figure 6.2: Frequency response measurements.

It can be seen that the different systems differ quite a lot in gain, peak of the resonance and resonance frequency. The low frequent dynamics, around 1 Hz is caused by the reference signal during the measurements, a ramp that will rotated at a certain (this) frequency. Above 100 Hz, there are be more dynamics which are not taken into account in this measurement. These dynamics are left out of the model since these dynamics have a low gain and will not influence the controlled system with a desired bandwidth around 20 Hz. To decrease the high frequent noise and these high-frequent dynamics a low pass filter can be added to the controller.
Transfer functions are fitted to the frequency response functions, resulting in 10 different models with the same structure. Fig. 6.3 shows the 10 fitted transfer functions while in Fig. 6.4 shows the frequency response and fitted model.
6.4, a single measurement and its fit can be seen as an example. The phase of the system is plotted from 180 to 0, actually it is $-180$ to $-360$ that belongs to a fourth order system. The fitted system will represent the system from 1 to 100 Hz quite well, however high frequent behavior is left out of the model since the fit does not represent the measurements and does not represent the experimental set-up at those high frequencies. A time delay can be added to the model the overcome the difference in phase starting at 60 Hz.

### 6.2 Controller design for single plant

For each identified system, a $H_{\infty}$ controller is designed by using a single weight on the sensitivity that will fix the bandwidth of the controlled system like previous explained in section 5.3.2. This is done for each of the models to achieve a $\gamma$ of nearly 1. A typical resulting controller can be seen in Fig. 6.5. The controller consists of a lead part until a notch filter compensates for the resonance perfectly. This results in a complete cancelation of the resonance in the experimental set-up if the fit matches this experimental set-up. This can be seen in Fig. 6.6, where the open-loop of a single controlled system is plotted.

![Figure 6.5: Typical controller for a single plant.](image-url)
Besides the canceled resonance, enough phase lead is created around the bandwidth of 25 Hz. The only concern is the accuracy of the frequency response function with respect to the real system, a small change in resonance frequency can result in an unstable, controlled system.

In Fig. 6.7, the sensitivity (solid line) with the corresponding weight function (dashed line) can
be seen. The $\gamma$-value of .99 has resulted in a very small difference between the two functions. The desired behavior of the controller is achieved. This is done for all the 10 different measured experimental set-ups, resulting in multiple controllers that will only control one plant as desired. When a controller and a system are mismatched, the resulting system will probably be unstable. This can be seen in Fig. 6.8, where the nyquist diagram of the open-loop of a single plant (the plant that has been used as an example here) and some of the controllers are plotted. Only four miss-matched controllers are used in this nyquist diagram because of better visualization. In this figure the solid bold black line represents the open-loop of the correct matched controller and plant, the other four open-loop diagrams represent mismatched controlled systems. What can be seen from this figure is that indeed the matching controlled system, while the other four, mismatched controlled systems are unstable. There can be multiple controllers that result in a stable system for the same plant. When evaluating all the possible controlled systems, 10 controllers and 10 plants result in 100 possible controlled systems from which 10 are matched and 90 miss-matched, 28 miss-matched controlled systems still result in a stable system.

Before the supervisory control is used, some straight forward simulations are carried out to check the controllers and predict the outcome of the total, supervisory controlled system.

Figure 6.8: Nyquist diagram of the controlled system and mismatched controllers.
6.3 Single plant - Single controller

First each controller is simulated with the matching plant for a step and a repeating signal. A step at time 0 is simulated for a single controlled system as presented above. This can be seen in Fig. 6.9(a), with the dashed line as reference and the solid line the response.

Figure 6.9: Simulation: (a) step response; (b) third order reference signal.

Figure 6.10: Simulated error: (a) Corresponding controller; (b) Mismatched controller.
Secondly a third order reference signal is constructed and simulated, see Fig. 6.9(b). Here the reference and the response match quite well, with only an repeating error in the order of $10^{-3}$ rad, see Fig. 6.10(a). Better performance could be achieved by using feedforward or repeatative control, however minimum error is not the goal of this thesis and therefore these techniques will not be included. In Fig. 6.10(b), a mismatched pair of controller and plant can be seen. Like predicted with the nyquist diagram of Fig. 6.8, this system is unstable with increasing error to infinity.

Now it is important that the supervisory control part can detect these differences in response, which seems to be trivial but this has to be done off line when only one controller is switched into the control loop, a stabilizing or destabilizing one. This will be discussed in the following part of this thesis.

### 6.4 Performance index in simulation

A key element in the supervisory controller is to detect whether a controller is falsified or not. This is done by calculating a performance index of the fictive signals, see Eq. 4.2. For this performance index the fictive error and fictive reference are used. These signals are computed by using the inverse of the controllers, which requires Stable-Causa-left-Invertible (SCLI) controllers, as described in 4.5.

In Fig. 6.5, one of the controllers is shown for the measured frequency domain, however a high frequent pole is present to decrease high frequent noise. This pole, at a frequency in the order of $10^4$ is not wanted because the the dynamics of the fit of the experimental set-up is only valid until 100 Hz and by inverting this controller the inverse is improper which is not desired. Therefore this pole is canceled and a proper controller is achieved which can be inverted, resulting in stable inverse controllers, see Fig. 6.11.

When the performance index of a matching controlled system is computed it can be seen that the performance index of an other, unstable controller results in the lowest, best performance index. This is shown in Fig. 6.12, where the performance index is computed with the previous discussed controlled system and the third order reference. The fat black line (the highest
6.4. PERFORMANCE INDEX IN SIMULATION

Figure 6.11: Inverse controller.

The performance index (performance index) represents the performance index of the only stable controller. Nonetheless, other controllers are detected as superior with respect to the performance index although these controllers will result in an unstable controlled system when switched into the loop. In other words, instability of the controllers can only be detected if the controllers are in the loop with

Figure 6.12: Performance index of a controlled system.
this type of reference signal and controllers. This behavior of the performance index is clearly not wanted but can be explained by evaluating the inverse of the different controllers in the frequency domain, see Fig. 6.13 where the bold black line represents the matching controller.

It can be seen that the matching controller contains the highest gain in low frequencies (until $20 - 25$ Hz). The fictive reference principle is reconsidered with these resulting inverse controllers in mind. Like mentioned before, only the input and output signals of the system are used to apply the supervisory control. These signals are fixed when the initial controller is inserted to the loop. Then for each controller $(i)$ the fictive reference, the reference that would result in the same, already measured input and output signals with each controller in the loop, see Eq. 6.1. Here it can be seen that only the previous explained inverse of the controllers contributes to the differences in fictive signals and thus in performance index.

$$\tilde{r}_i = C_i^{-1}u_p + y_p$$

Because the (real) reference signal is predominantly designed containing signals with frequencies until the bandwidth, the fictive reference and fictive error will be larger when the gain of the inverse of the controller is larger than others with a smaller gains. This does not lead to
6.5. COMPLETE CONTROLLED SIMULATION

problems when the simulated system is stable and the performance index of the stable (and best in design) controller does not exceed the falsification bound. In that case the stabilizing controller maintains in the loop and the desired performance is achieved.

When the simulated controlled system is unstable, the input and output of the system \((u_p, y_p)\) are increased rapidly. The performance index exceeds the falsification bound and the controller belonging to the smallest performance index is switched into the loop. Or a switch is needed if the controller (see section 5.3.1) designed for the whole uncertain domain is the initial controller and does not meet the performance specifications. The wrong controller could be switched into the loop, the system becomes unstable and soon all the controllers are falsified. This is definitely not wanted, but it can occur with this set of controllers and this sort of reference signal that the only proper (stabilizing and satisfying the performance index) controller is falsified due to the use of the controller with the lowest magnitude when inverted.

A special, lucky case can occur when the system corresponding to the lowest inverse magnitude is controlled. Then the correct matching controller is found and remains unfalsified during the whole simulation because the desired performance is achieved. In the following part, the complete supervisory control scheme is simulated and the above behavior can be seen. Whereafter in section 6.6 some recommendations to improve the falsification bound and performance index are given.

6.5 Complete controlled simulation

In this section the full supervisory control is simulated for different situations by using the third order reference. First a random plant is controlled, the full set of controllers contains the 10 different controllers for the 10 identified plants and the low performance controller for the full uncertain domain. This low performance controllers stabilizes all the possible systems and will be used to start with, see Fig. 6.15(a) where controller 0 represents this full uncertain domain controller. In Fig. 6.14(a) the resulting error can be seen when a random plant is tried to control. Apparently the system is unstable, like predicted in previous section, and after a certain amount of time every controller is falsified and the simulation is aborted. In Fig. 6.14(b)
the performance index can be seen where the dashed line represents the controller for the full uncertain domain with low performance. This controller stabilizes the plant but due to the low performance, this controller is falsified when the performance index exceeds the value of 1 and a new controller is switched into the loop. However, the destabilizing controllers are not evaluated to be unstable while the stable full uncertain domain controller is used. For this type of controllers and reference signals, this phenomena can occur and is explained in previous sections.
The unfalsification process can be seen in Fig. 6.15(b), here at a time of .51 seconds the performance index exceeds the value of 1 (Fig. 6.14(b)) and is unfalsified. Then the controller with the smallest performance index is switched into the loop, see Fig. 6.15(a) and the performance index is reset. This reset is used to allow the new switched controller to settle and overcome the problem of fast unfalsification of every controller. This controller number 7 is apparently a mismatched controller, resulting in an unstable system. At 0.65 seconds the performance index of this controller is falsified and controller number 4 is switched into the loop, this controller leads to an unstable system as well and shortly after this last switch all the controllers are falsified and the simulation is aborted. Although there is a controller available which can control the system as desired, this controller is not identified as such.

This feasibility is demonstrated in the following simulation where, initially, the matching controller is switched into the loop. The performance index value, see Fig. 6.16(b), of 1 is not exceeded by the 10 single plant controllers, resulting in unfalsification of all the controllers except the stabilizing controller for the whole uncertain domain, see Fig. 6.17(b). The current, matching controller is kept in the loop (Fig. 6.17(a)) which results in a satisfying error in Fig. 6.16(a). This error is in the same order as the error previously simulated in direct controlled system without the supervisory part. The error oscillated in the beginning of the simulation this

![Image](a)

![Image](b)

Figure 6.16: Simulation results: (a) Error; (b) Performance index.
is due to the initial conditions, after 2.5 seconds the system produces repeating errors which could be decreased with feedforward techniques.

Different results can occur when, coincidental the plant corresponding to the smallest performance index is controlled. Depending on the initial controller, two different results are observed. First, the initial controller is a stabilizing controller that does not satisfy the performance criterium.

Figure 6.18: Simulation results: (a) Error; (b) Performance index.
This can be seen in Fig. 6.18(a), 6.18(b), 6.19(a) and 6.19(b). Here the first controller, 0 is used in the loop until falsified, whereafter the correct controller is switched in the loop because by coincidence this controller matches the smallest performance index. The error is increased in the beginning due to the low performance controller but is decreased when the correct controller is switched in the loop (number 7). All the other single plant controllers remain unfalsified.

Secondly, an unstable controller is initially switched in the loop which results in increasing error and increasing performance index, Fig. 6.20(a) and Fig. 6.20(b). From the unfalsification
Figure 6.21: Controllers in simulation: (a) Switched into loop; (b) Unfalsified.

diagram it can be seen that the low performance controller is falsified very fast whereafter the single plant controllers are falsified in succession until all are falsified and the simulation is aborted. A falsification bound which accounts for the transient behavior of the system might be the solution, however if the stabilizing controller does not result in the lowest performance index, the system remains unstable. In this case, the controller that needs to be unfalsified is falsified. Here at time 0.21 seconds every controller is falsified (Fig. 6.21(b), the current controller is falsified as last and maintains in the loop during the whole simulation time, Fig. 6.21(a). After a while the simulation is ended because all the controllers are falsified. This might not happen when the initial mismatch appears to be stable or if the performance index of the matching controller is the lowest as shown in the simulation demonstrated with Fig. Fig. 6.16(a).

6.6 Conclusion and recommendations

Although in theory feasibility is the only criterium desired to control the system, the simulations showed that while the feasibility criterium is satisfied, a wrong controller can still be switched in the loop. This is due to the specific reference and inverse controllers that do not guarantee the performance index of the best performing controller to have the smallest value. Then the (possibly only) feasible controller is falsified, the controller is mismatched and the system becomes unstable. This is definitely not the desired behavior of the supervisory control method and should not be used for this type of systems when the plant can be identified pretty easily and
the controllers for each single plant are used along with the low performance controller for the full uncertain domain.

Several situations have been simulated and discussed, these are summed up in the following table, 6.1. The situation where the initial controller is stabilizing and suffice the performance index will remain in the loop. Hence there is no reason to switch and the initial controller is the final controller. When starting with a stabilizing controller with insufficient performance (2nd column), previous simulations showed that the next controller results in a stable, well performing controlled system or an unstable system. Since there is only one stable controller with insufficient performance designed, the third option, switching to an other stable controller with insufficient performance is not possible. The last option as initial controller is a destabilizing controller. With one of the destabilizing controllers, it is only shown that the system remains unstable while the desired behavior is a switch to a stabilizing controller. The stabilizing controller with insufficient performance is less likely to be switched into the loop because the performance criterion is not matched.

<table>
<thead>
<tr>
<th>Controller K</th>
<th>Stable, Performance OK</th>
<th>Stable, Performance not OK</th>
<th>Destabilizing</th>
</tr>
</thead>
<tbody>
<tr>
<td>K + 1</td>
<td>X¹</td>
<td>✓²</td>
<td>?³</td>
</tr>
<tr>
<td>K + 1</td>
<td>X¹</td>
<td>?³</td>
<td>?³</td>
</tr>
<tr>
<td>Destabilizing</td>
<td>X¹</td>
<td>✓²</td>
<td>✓²</td>
</tr>
</tbody>
</table>

The desired result is to achieve a control method where for every initial controller type, the final, resulting controller is stable and performs well. In other words the final controller need

¹No reason to switch, initial controller remains in loop.
²This is shown in this chapter.
³Not shown in simulations.
to be on row 1. This can only occur when the correct, or at least a feasible controller based on the measured data, is directly observed as best controller and switched into the loop.

Because of the transient behavior, the falsification bound should start with a high value before decreasing to the final desired value. Secondly the performance index should represent the behavior of the controlled system if that controller actually was in the loop (like desired for the fictive reference signals). Then, one might construct a different falsification bound for each controller since after a switch each controller has a different transient behavior to get to its steady state, if the system is stable. This includes a new calculation of the falsification bound at each time, or at a certain time interval.
Chapter 7

Experiments

The designed control scheme is tested on the experimental PATO set-up. With the presence of possibly noise, friction and non-linear dynamics the control error will be larger and the control scheme needs to be fast enough to perform the real time experiment. First a PATO realization is controlled with a single, matching controller to quantify the error that can be achieved when the correct controller is switched into the control loop and indicate a falsification bound. Secondly the complete supervisory control scheme is tested with different initial controllers. The case where the experiment is started with an unstable controller is not tested because of the simulation results where an initial destabilizing controller resulted in unstable behavior.

7.1 Single Plant - Single Controller

First the single controllers are tested on the corresponding realizations. The supervisory control method is omitted during this experiment to test the controllers with respect to the PATO set-ups. The reference is of the same shape as in simulation, only scaled with a gain.

In Fig. 7.1(a) and 7.2(a) the reference and the resulting, measured output can be seen for two different controlled PATO set-ups with matching controllers. It can be seen that in the first experiment (Fig. 7.1(a) and Fig. 7.1(b)) the error is smaller and less noise is present when comparing to the second experiment, which can be seen in Fig. 7.2(a) and Fig. 7.2(b). The high amount of noise is a result of the large, high frequent, gain of the controller, this can be
decreased by adding a low-pass filter around 200 Hz. This is not necessarily needed when testing the supervisory control scheme because the main focus is to switch a proper controller into the loop with respect to the performance. As shown in simulation there is a difference between a proper controller with respect to the performance index and with respect to the achieved actual performance, which should not be the case.

Figure 7.1: Experiment: (a) reference and measured output; (b) reference and error.

Figure 7.2: Experiment: (a) reference and measured output; (b) reference and error.
7.2 Complete controlled system

When using the supervisory control scheme, initially, a stable controller is used to start the experiment. Depending on the performance index of the used controller switching occurs. First the corresponding controller is used as initial controller is tested, the system should remain at this corresponding controller or, as can be seen from the simulations, a mismatched controller will be switched in the loop resulting in an unstable system which is clearly not desired. In Fig. 7.3(a) and 7.3(b) the reference, measured output and the error can be seen. At around .3 seconds the system becomes unstable, the measured output increases rapidly and thus the error grows as well, while in the beginning the error is satisfying.

In Fig. 7.4(a) and 7.4(b) the falsification of the controllers and the performance index can be seen respectively. A value of −1 represents the falsification of the controller, initially the corresponding controller number 9, solid line, is controlling the system. However at .3 seconds the performance index of this controller (see Fig. 7.4(b)) becomes larger than 1 (fat solid line). This results in a switch of controllers, from 9 to 8 (fat dashed line). At this switch, the experiment stops, a constant control output is applied to the experimental set-up without saturating and the measurement is manually aborted at 1.3 seconds.

This phenomena is not seen in the simulations, switching in simulation is achieved without any problems. The new controller could be destabilizing, resulting in an unstable system
and saturation of the control input. A second explanation of this phenomena is that the computational burden at a switch might be too large to follow the real time experiment, resulting in a failing experiment. The first explanation does not occur since the control input is fixed at the last given control output before the experiment failed. Some other values are still computed, as can be seen in the falsification and performance index figure. However, there is no controller actually switched in the loop, as will be seen from Fig. 7.7.

Figure 7.5: Complete experiment: (a) reference and measured output; (b) reference and error.
A new experiment is performed, where in the first 3 seconds the controller is forced to stay at the desired, corresponding controller whereafter the complete supervisory control part is used to control the system further. This experiment is done an other PATO system, where controller number 7 should perform well. The 3 seconds of forced control by the corresponding controller is used to overcome the transient behavior of the performance index, this can only be done when the corresponding controller is known which the user normally does not know. From Fig. 7.5(a) and 7.5(b), it can be seen that the first 3 seconds the system behaves well although the error becomes more high frequent and increases a bit after .5 seconds. Right after the 3 seconds

![Figure 7.6: Complete experiment: (a) falsification; (b) performance index.](image)

![Figure 7.7: Controller to be switched into the loop.](image)
of forced control the system becomes unstable and the the error grows fast. This is the same phenomena seen in the previous experiment.

In Fig. 7.6(a) the falsification can be seen, where after .25 all the controllers are falsified due to the transient behavior of the performance index in Fig. 7.6(b). During this experiment, the controllers are able to become unfalsified again, as can be seen from the figure. During the first 3 seconds, controller number 7, the solid black line, is controlling the system, but this controller is falsified and controller number 8 is switched into the loop according to the performance index. Again, at this switch the system becomes unstable due to the computational burden where the minimum controller according to the performance index is failed to calculate (see Fig. 7.7). Starting from the beginning of the experiment, the best controller, according to the performance index are controller number 9, 7, 4 and finally 7. At time of .25 seconds number 8 should be the “best” controller, however this is never computed. This number is only desired at a switch but not available at that time, resulting in the behavior of Fig. 7.5(a) and a failing experiment.

7.3 Conclusion and recommendations

From the experimental results it can be seen that besides the problem already discussed in chapter 6, computation of the minimal controller needs to be maintained throughout the whole experiment. Then, with the identification of a stabilizing and well performing controller corresponding to the lowest performance index, a satisfying error can be achieved for these PATO set-ups. The main focus should be on the performance index, the "best" controller in theory (or at least a satisfying controller) should have the lowest performance index including the initial and switching transients.

Finally the computational burden should be decreased by implementing the supervisory control in a efficient manner.
Chapter 8

Conclusion

A Multiple Model Multiple Controller Adaptive control system has been developed. Initially, the controllers were based on $\mu$ analysis, synthesized by using complex-$\mu$ and evaluated with mixed-$\mu$ for the complex and real valued uncertainties of the multiple models. These models are created by identifying multiple realizations of the PATO system. These multiple controllers did not differ enough to notice differences when used on a single plant in the total uncertain model set, due to the non-optimal, conservative synthesized controllers. Therefore single $H_\infty$ controllers are designed for each identified PATO realization.

The $H_\infty$ controllers are used in combination with the supervisory, unfalsified control. Simulations showed that the unstable behavior of mismatched controllers could not be seen off-line, like desired, by using the fictive references and fictive errors. And, in some cases, the proper controller was falsified, resulting in an unstable system. This behavior was as well seen in experiments. This is inherent to the fictive reference, it does not guarantee that a well performing controller is not falsified, however the unstable behavior of some mismatched controllers should be noticed.

During experiment, controlling the plant with a single controller did result in a satisfying controlled system. But when switching is desired, the computational burden prevents computing the number of the new controller, resulting in aborting the experiment.
Chapter 9

Recommendations

Regarding the design of the robust controllers, these are analyzed by using a mixed-$\mu$ method for real and complex uncertainty. However, the controllers are synthesized with complex-$\mu$. This results in non-optimal controllers, with a performance that possibly can be increased. By using mixed-$\mu$ synthesis, the designed controllers are optimal and a larger difference between the multiple controllers should be the result which makes the supervisory part easier. The conservatism of the $\mu$ controllers is beneficial to the falsification principle since there are more stable well performing controllers for a single plant. For this, mixed-$\mu$ design in crucial instead of complex-$\mu$ design.

As can be seen in simulation, the best performing controller could not be found by using a fictive reference. A different performance index should be used, for this type of system, that is able to distinguish the different controllers better and does not only dependent on the inverse of the different controllers because the magnitude of the inverse of these controllers do not differ much for frequencies up till the bandwidth.

The falsification bound should be adapted to overcome the transient behavior of the system and falsifying controllers that should have been unfalsified after the transient of the system. A solution is an, exponentially, decreasing falsification bound towards the fixed value. The transient of a candidate controller might be taken into account by this falsification bound as well. This results in a different falsification bound for each candidate controller which enhances the control scheme but may result in unfalsifying well performing controllers.
Finally, switching to an other controller seemed to be troublesome during the real time experimental set-up. The number of the new controller was incorrect calculated after a certain amount of experimental time, resulting in a switch to an erroneous number for the next controller. The computational burden of the control scheme could be the cause for this behavior. Therefore it should be implemented in a more efficient manner to be able to switch after a certain amount of experiment time.
Bibliography


Appendix A

FRF Measurements

The used frequency response functions are obtained, as explained in section 6.1, by using a standard frf measurement method which only uses the plant input and output. This method is explained in more detail in this chapter. First, a weak controller with low bandwidth is designed (lead filter) to stabilize the system and to be able perform a measurement. A ramp is used as reference and band limited white noise is added to the plant input as can be seen in Fig. A.1.

![Identification scheme](image)

Figure A.1: Identification scheme.

The three measured signals are the disturbance (w), the plant input (u) and the plant output (y). As a reference signal a ramp is used which is detrended before usage to decrease the effect of this simple input signal. By diving the plant input over the disturbance, the sensitivity can be computed while dividing the plant output over the disturbance the process sensitivity is achieved. These can be seen in Fig. A.2 and Fig. A.3. The experiment is performed at a sampling frequency of 2 kHz, this leads to a highest measurable frequency of 1 kHz. The dynamics of the PATO system beyond that frequency is not measured, not modeled and therefore not taken into account by the controllers.
By diving the process sensitivity (PS) over the sensitivity (S), the resulting data represents the plant. The resulting plant from the below visualized measurement can be seen in the following figure, Fig. A.4. A discontinuity from the $-2$ slope at a frequency of 2 Hz can be seen, this is due to the reference signal, this discontinuity is not modeled and therefore it does not influence the
controller, simulations or experiments any further. The plant starts with a $-2$ slope whereafter the resonance a $-4$ slope can be seen, as desired. The high frequency behavior (0 slope, large amount of noise) is due to the encoder accuracy, the proper position cannot be measured accurate enough. This type of identification is done for each of the $10$ PATO realizations, whereafter a fit has been made as explained in 6.1.