

# Ellipsoidal unfalsified nonlinear controller design of a linear motion system

**Citation for published version (APA):**

Kok, T. (2005). *Ellipsoidal unfalsified nonlinear controller design of a linear motion system*. (DCT rapporten; Vol. 2005.142). Technische Universiteit Eindhoven.

**Document status and date:**

Published: 01/01/2005

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

Ellipsoidal unfalsified nonlinear controller  
design of a linear motion system

T. Kok

DCT 2005.142

Traineeship report

**Coach(es):** W.H.T.M. Aangenent  
J.J.M. van Helvoort  
M.J.G. van de Molengraft

Supervisor: M. Steinbuch

Technische Universiteit Eindhoven  
Department Mechanical Engineering  
Dynamics and Control Technology Group

Eindhoven, november, 2005

**Abstract**

Ellipsoidal Unfalsified Control (a data-driven model free control theory) is applied to determine a nonlinear controller for a linear motion system that ideally outperforms a linear controller.

In this report a number of candidates for nonlinear controller parts are discussed.

Simulations are done in order to find out which nonlinear parts could outperform linear ones. These parts are tested on a dual rotary 4<sup>th</sup> order motion system.

Conclusions are made with respect to the possibilities to use Ellipsoidal Unfalsified Control to determine a nonlinear controller.

## Nomenclature

<b>Symbol</b>	<b>Quantity</b>
$\theta$	Control parameters
$\theta^*$	Current implemented control parameters
$\varepsilon(t_{k-1})$	Old region of unfalsified control parameters
$\varepsilon(t_k)$	New region of unfalsified control parameters
$F(t_k)$	Region of control parameter sets that is unfalsified by the current data
$r(t)$	Reference
$u(t)$	Plant input
$y(t)$	Plant output
$y_m(t)$	Desired trajectory
$w(u(t), y(t))$	Vector field defining the controller structure
$R_v(t)$	Virtual references
$G_m(s)$	Reference model
$Y_m(t)$	Set of reference outputs
$L(s)$	Low-pass filter
$\Delta(t)$	Performance specification
$\Sigma(0)$	Matrix defining the initial ellipsoid
$e(t)$	Error between reference model- and plant output.

Abstract.....	2
Nomenclature.....	3
Introduction.....	5
1.1    Project goal .....	5
1.2    Report overview.....	5
2    Theory.....	6
2.1    Unfalsified Control .....	6
2.2    Ellipsoidal Unfalsified Control.....	7
3    Candidates for nonlinear controller parts.....	8
3.1    SPAN filter.....	8
3.2    Nonlinear gain.....	8
3.3    Arbitrary controller .....	9
4    Simulation.....	10
4.1    Linear controller.....	10
4.2    SPAN filter.....	10
4.3    Nonlinear gain.....	11
4.4    Arbitrary controller .....	12
5    Experiment.....	13
5.1    Linear controller.....	13
5.2    SPAN filter.....	14
5.3    Nonlinear gain.....	14
5.4    Arbitrary controller .....	15
6    Conclusions and recommendations.....	17
6.1    Conclusions.....	17
6.2    Recommendations.....	17
References.....	18
Appendices.....	19
A.    Matlab/Simulink scheme of the EUC algorithm.....	20

## **Introduction**

### **1.1 Project goal**

In the field of control, the fact that there are fundamental limitations on the performance has been the subject of research for many years (H. Bode, 1945 and more recent M.M. Seron, 1997). One of these limitations is the bode-sensitivity integral which implies that good performance at one frequency means less performance at another frequency. This, however, only applies to linear controlled systems.

There are indications that, if a nonlinear controller is used, these fundamental limitations can be overcome. At this moment there are no usable analytical tools that enable us to design a nonlinear controller with a better performance than a linear version.

Ellipsoidal Unfalsified Control (EUC) (J. van Helvoort, B. de Jager, and M. Steinbuch, 2005) is a data driven design method which determines its control parameters in such a way that it satisfies certain performance specifications.

In this report EUC will be used to investigate the possibility to design a nonlinear controller that ideally outperforms the linear version. Special attention will be paid to the resulting controller and understanding why the controller is better.

### **1.2 Report overview**

This report firstly describes the theory behind ellipsoidal unfalsified control (EUC). In chapter 3 a number of candidates for nonlinear controller parts are discussed. Chapter 5 shows the results of a number of simulations done with these controller parts added to a good performing controller structure. The same controller structures are used for experiments on a dual rotary 4th order motion system. Finally in chapter 6 some conclusions and recommendations are made.

## 2 Theory

In this research the Ellipsoidal Unfalsified Control (EUC) concept is used to design a nonlinear controller. EUC is a data driven control method which determines control parameters for a given controller structure in such a way that it can satisfy a given performance criterion. EUC is a special case of Unfalsified Control, the theory behind Unfalsified Control shall therefore be explained first.

### 2.1 Unfalsified Control

Unfalsified Control, in contrast with classical methods, doesn't use a model of the plant. Only the structure of the controller is chosen. The parameter space together with this controller structure defines the controller pool. This controller pool contains all the controller candidates. The only assumption that has to be made is that at least one controller from the original controller pool satisfies a certain chosen performance specification at all times.

In figure 2.3 a block scheme is shown of a model reference unfalsified control model. The controller has the structure:

$$r(t) = w(u(t), y(t))^T \theta^* (t) \quad (1)$$

with  $r(t)$  a chosen reference signal,  $\theta^*(t)$  the set of  $p$  current control parameters and  $w(u(t), y(t))$  a  $p$ -dimensional vector field with arguments depending on current and past values of  $u(t)$ , the input of the plant and  $y(t)$ , the output of the plant.

A set of virtual references  $R_v(t)$  is defined:

$$R_v(t) = \left\{ r_v(t, \theta) = w(u(t), y(t))^T \theta \mid \theta \in \mathbb{R}^p \right\} \quad (2)$$

(1) and (2) imply that for every  $\theta \in \mathbb{R}^p$ , the corresponding virtual reference  $r_v(t, \theta) \in R_v(t)$  results exactly in the measured  $u(t)$  and  $y(t)$ , if that control parameter set  $\theta$  were in the feedback loop at the time of the measurement.

$G_m(s)$  defines a stable reference model for the desired closed loop behavior. The set of reference outputs  $Y_m(t)$  is defined as

$$\begin{aligned} Y_m(t) &= \left\{ y_m(t, \theta) = G_m(s) r_v(t, \theta) \mid r_v(t, \theta) \in R_v(t) \right\} \\ &= \left\{ y_m(t, \theta) = G_m(s) w(u(t), y(t))^T \theta \mid \theta \in \mathbb{R}^p \right\} \end{aligned} \quad (3)$$

The unfalsification algorithm uses  $Y_m(t)$  to determine the new controller pool with only unfalsified controllers.

In early works (Michael G. Safonov, and Tung-Ching Tsao, 1994) the unfalsification was done by gridding the parameter-space (resulting in a finite set of candidate controllers). This means that a constant trade off has to be made between the number

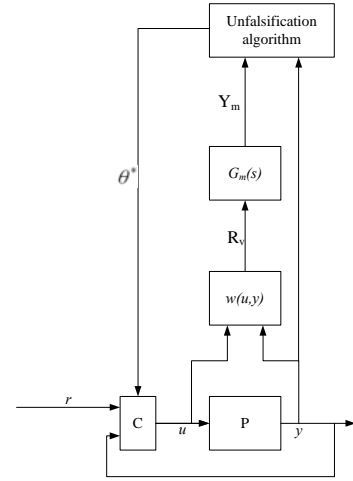


Figure 2.1: General unfalsified control model.

of controllers that is taken in to account and the computational load. This need for gridding was overcome by ([ Fabricio B. Cabral, and Michael G. Safonov, 2003) by using an ellipsoid to describe the region containing the unfalsified control parameters. The algorithm used here has however a slow convergence. Therefore the algorithm used in (van Helvoort, 2005) which has a fast enough convergence to be implemented in real time will be explained and used here.

## 2.2 Ellipsoidal Unfalsified Control

The need to use gridding is overcome by applying an ellipsoidal description of the region containing all unfalsified control parameter sets. Hereto the region of unfalsified control parameter sets is specified by an ellipsoid  $\varepsilon(t_{k-1})$ .

Consider the bound on the maximum tracking error

$$|L(s)(y(t) - y_m(t))| \leq \Delta(t) \quad \forall t \quad (4)$$

with  $L(s)$  some low-pass filter to reject noise and  $\Delta(t) > 0$  a time dependent threshold value. The region of control parameter sets that is unfalsified by the current data,  $F(t)$ , will then be given by

$$F(t) = \left\{ \theta \mid -\Delta(t) \leq L(s) \left( y(t) - G_m(s) w(u(t), y(t))^T \theta \right) \leq \Delta(t) \right\} \quad (5)$$

this defines two half-spaces which contain the region of controller parameters unfalsified by the current measurement data. This region can also be seen as a degenerate ellipsoid:

$$F(t) = \left\{ \theta \mid L(s) \left( y(t) - G_m(s) w(u(t), y(t))^T \theta \right)^2 \leq \Delta^2(t) \right\} \quad (6)$$

The new region of unfalsified control parameters is now the intersection of the two regions

$$\varepsilon(t_k) = \varepsilon(t_{k-1}) \cap F(t) \quad (7)$$

This will in general not be an ellipsoid. The intersection will therefore be approximated by an outer-bounding minimal-volume ellipsoid, so that no parameter sets will be falsified incorrectly.

In (van Helvoort, 2005) an algorithm is given for updating controller parameters using a minimum-volume outer-bounding ellipsoid intersection algorithm. This algorithm is implemented in a Matlab/Simulink model. Simulations and experiments will be done using this model.

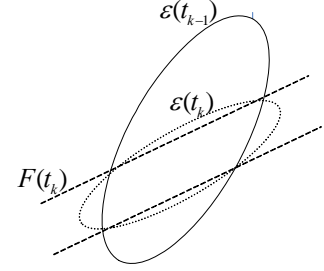


Figure 2.2: Minimal volume outer-bounding ellipsoid approximation of the intersection.



### 3 Candidates for nonlinear controller parts

When EUC is used, limitations are imposed on the controller structure that can be used. As stated above this has to be of structure  $r(t) = w(u(t), y(t))^T \theta^*(t)$ . Vector field  $w(u(t), y(t))$  should be constructed such that  $u(t)$  can be determined uniquely when  $r(t)$ ,  $y(t)$ ,  $\theta^*(t)$  and past values of  $u(t)$  and  $y(t)$  are known. As a starting point the vector field as mentioned in (van Helvoort, 2005, section 5):

$$w(u(t), y(t)) = \begin{bmatrix} u(t) \\ \frac{1}{s+10\pi} u(t) \\ y(t) \\ \frac{1}{s+10\pi} y(t) \end{bmatrix} \Rightarrow r(t) = u(t) \cdot \theta_1 + \frac{1}{s+10\pi} u(t) \cdot \theta_2 + y(t) \cdot \theta_3 + \frac{1}{s+10\pi} y(t) \cdot \theta_4$$

is used. In every experiment a nonlinear term will be added which in theory could improve performance. These nonlinear terms must have a structure that enables them to be added to the vector field mentioned above. The following controllers can be used in this structure.

#### 3.1 SPAN filter

SPAN stands for split-path nonlinear. The block scheme of a SPAN filter is given in figure 3.1. As can be seen the filter uses the input in two parallel paths of which the gain respectively the sign information is destroyed. With this filter the sign and the magnitude can be independently chosen.

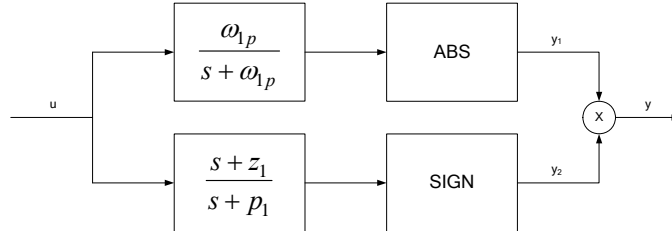


Figure 3.1: Block scheme of a SPAN filter.

As shown in (Aangenent, 2005) this filter is able to improve performance with respect to overshoot and settling time of a step response. Because of the structural limitations it is not possible to obtain values of  $\omega_{1p}$ ,  $z_1$  and  $p_1$  using EUC so these values should be known. Although it is not possible to design this filter using EUC it could be interesting to see if, when this filter is added to the vector field, a high weighting factor is assigned to it.

#### 3.2 Nonlinear gain

A nonlinear gain element multiplies the current controller with a nonlinear gain. The gain can be a function of any loop variable. As described in (Aangenent, 2005) this kind of filter could be used to reduce the overshoot of a step response without increasing the settling time. Again the problem arises that the nonlinearity itself, here the nonlinear gain, cannot be determined by EUC. It is possible to determine a multiplication factor when the shape of the nonlinear gain is known, e.g. as shown in figure 3.2.

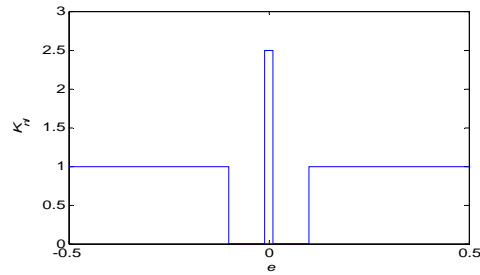


Figure 3.2:  $K_{nl}$  as a function of the error.

### 3.3 Arbitrary controller

A large variety of nonlinear controller parts can be added to the linear one. Some examples are:

- A saturation function of one of, or a combination of the variables.
- A sign function
- A relay function
- A deadzone
- Etc.

EUC should in theory only use the parts that improve the performance. This could lead to a controller of which in advance it could not have been expected that it would improve the performance. It is of course not possible to use every possible nonlinear controller. Therefore a number of randomly chosen nonlinear parts is added.

#### 4 Simulation

Simulations will be performed using Matlab/Simulink. The simulations will be done using the same algorithm as used in (van Helvoort, 2005). The settings for the simulation are:

$$G_m(s) = \frac{100\pi^2}{s^2 + 16\pi s + 100\pi^2} \quad (7)$$

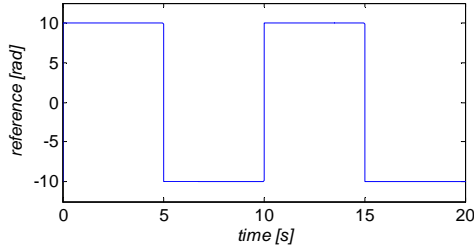
$$\begin{cases} r(t) = 400t - \frac{20}{2\pi} \sin\left(\frac{2\pi}{0.05}t\right) & 0 \leq t \leq 0.05 \\ r(t) = 20 & t > 0.05 \end{cases} \quad (8)$$

$$\Delta(t) = 0.002 + e^{(-0.1t)} \quad (9)$$

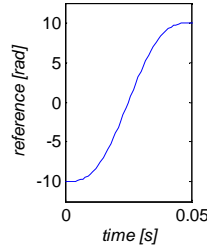
$$\Sigma(0) = 10^4 I_n \quad (10)$$

$$L(s) = 1 \quad (11)$$

$\Sigma(0)$  defines the shape of the initial ellipsoid or initial controller pool (van Helvoort, 2005) with  $n$  the number of controller parts. The reference ( $r(t)$ ) is shown in figure 4.1, it is a skew-sinus which is repeated every 5 seconds. A skew-sinus excites the system more than a normal sinus this should force the algorithm to falsify more controller which could lead to the fact that the system can only be controlled with a nonlinear controller part active.



(a) Reference used in the simulations



(b) First 0.05 seconds of the reference to illustrate the shape of a skew-sinus

Figure 4.1: Reference as a function of time.

To allow some transient error,  $\Delta(t)$  will be multiplied with 30 for 0.1 second every time a step occurs.

The plant is described in state-space form as:

$$\begin{cases} \dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -5.5e4 & -5.8 & 5.5e4 & 5.8 \\ 0 & 0 & 0 & 1 \\ 4.4e4 & 4.6 & -4.4e4 & -4.6 \end{bmatrix} x + \begin{bmatrix} 0 \\ 6.4e3 \\ 0 \\ 0 \end{bmatrix} u \\ y = [0 \ 0 \ 1 \ 0]x \end{cases} \quad (12)$$

Which represents a dual rotary 4<sup>th</sup> order motion system (PATO model). A sample frequency of 1.0 kHz is used.

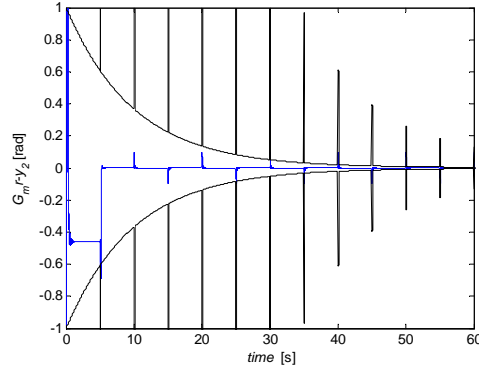
## 4.1 Linear controller

To be able to compare performance of the nonlinear controllers with the linear one a simulation with a linear controller will be done first.

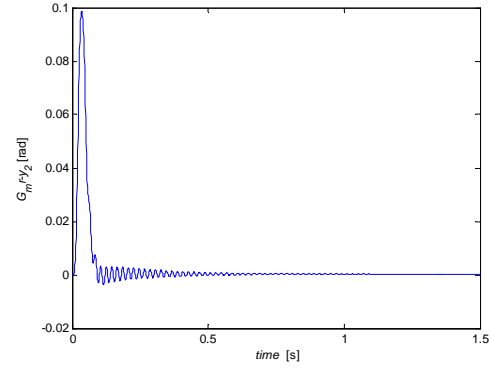
The simulation is performed using the settings as mentioned above with a vector field

$$w(u(t), y(t)) = \left[ u(t), \frac{1}{s+1} u(t), y(t), \frac{1}{s+1} y(t) \right]^T$$

which leads to a controller of structure  $r(t) = \theta_1 u(t) + \theta_2 \frac{1}{s+1} u(t) + \theta_3 y(t) + \theta_4 \frac{1}{s+1} y(t)$ . The simulation is initiated with controller parameters  $\theta^*(0) = [100 \ 0 \ 0 \ 0]^T$ .



(a) Tracking error during the experiment, along with the bounds the error has to stay within.



(b) Tracking error when the last known good  $\theta$  is inserted in the controller.

Figure 4.2: Tracking error as a function of the time.

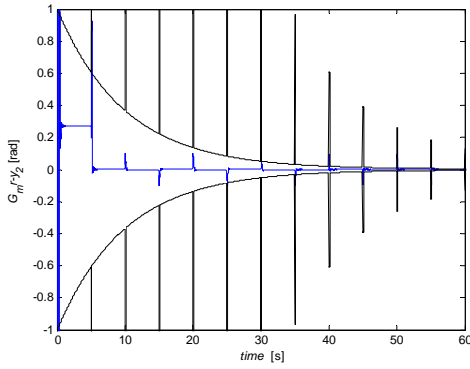
In figure 4.2(a), the tracking error  $G_m r - y$  is shown, together with the bounds  $+\Delta(t)$  and  $-\Delta(t)$ . As can be seen the bounds are multiplied with 30 for 0.1 sec every time a position change is initiated. When the error intends to pass the bounds the controller parameters are updated to force the error to stay within the bounds. Figure 4.2(b) shows that the maximum tracking error is 0.098 rad. Within 0.1 sec the tracking error is less than 0.005 rad. These values will be used as performance criteria.

## 4.2 SPAN filter

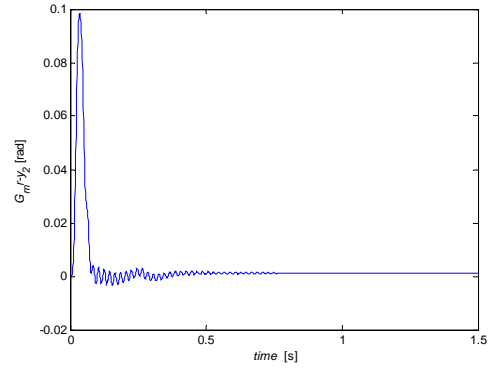
In this section a SPAN filter will be added as a nonlinear part. The simulation is performed using the settings as mentioned above with a vector field

$$w(u(t), y(t)) = \left[ u(t), \frac{1}{s+1} u(t), y(t), \frac{1}{s+1} y(t), \left| \frac{30}{s+30} u(t) \right| \text{sign}\left(\frac{10/3}{s+60} u(t)\right) \right]^T$$

The simulation is initiated with a controller with  $\theta^*(0) = [100 \ 0 \ 0 \ 0 \ 0]^T$ .



(a) Tracking error during the experiment, along with the bounds the error has to stay within.



(b) Tracking error when the last known good  $\theta$  is inserted in the controller.

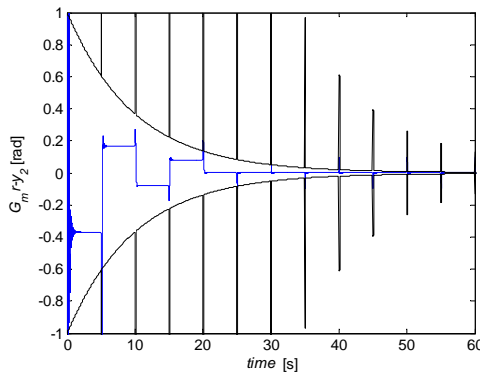
Figure 4.3: Tracking error as a function of the time with a SPAN filter implemented.

In figure 4.3(a), the tracking error  $G_m r - y$  together with the bounds  $+\Delta(t)$  and  $-\Delta(t)$ , is shown again. In figure 4.3(b) the maximum tracking error of 0.098 rad is shown. Within 0.08 sec the tracking error is less than 0.005 rad. The tracking error seems to approach 0 a little faster.

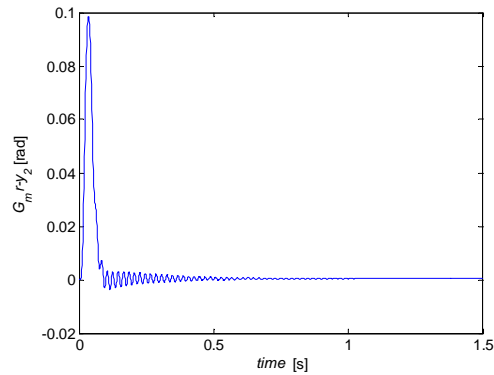
### 4.3 Nonlinear gain

Another simulation is performed with a vector field

$w(u(t), y(t)) = \left[ u(t), \frac{1}{s+1} u(t), y(t), \frac{1}{s+1} y(t), K_{nl}(e)u(t) \right]$ . Here  $K_{nl}(e)$  is a function of the error as plotted in figure 3.2. The simulation is initiated with a controller with  $\theta^*(0) = [100 \ 0 \ 0 \ 0 \ 0]^T$ .



(a) Tracking error during the experiment, along with the bounds the error has to stay within.



(b) Tracking error when the last known good  $\theta$  is inserted in the controller.

Figure 4.4: Tracking error as a function of the time with a nonlinear gain controller implemented.

Figure 4.4(b) shows almost the same plots as figure 4.2(b). This is explained when  $\theta_5$ , the  $\theta$  representing the nonlinear gain, is examined.  $\theta_5$  only changes a neglectable amount. This means that the nonlinear gain part is not used.

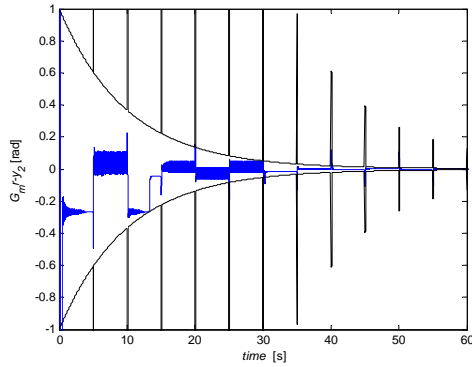
#### 4.4 Arbitrary controller

The last simulation that is performed will have a vector field

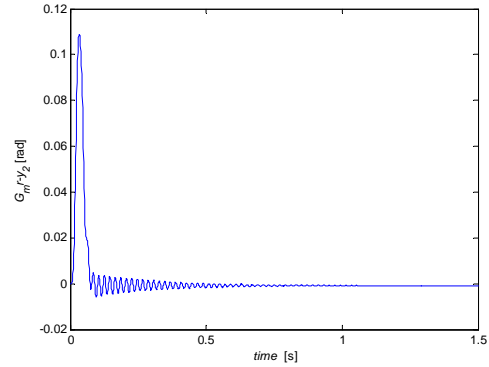
$$w(u(t), y(t)) = \begin{bmatrix} u(t), \frac{1}{s+1}u(t), y(t), \frac{1}{s+1}y(t), |e(t)|, \text{sign}(e(t)), \text{sat}(e(t)/0.01), \\ \text{dead zone}(e(t)/0.01), \text{relay}(-0.01 : e(t) : 0.01), \text{backslash}(e(t) : 0.02) \end{bmatrix}.$$

The simulation is initiated with a controller with

$$\theta^*(0) = [100 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T.$$



(a) Tracking error during the experiment, along with the bounds the error has to stay within.



(b) Tracking error when the last known good  $\theta$  is inserted in the controller.

Figure 4.5: Tracking error as a function of the time with an arbitrary controller.

Figure 4.5 shows that when a arbitrary controller is used it is possible to obtain a good performance. Figure 4.5(b) shows that although the maximum tracking error is a little bigger it is within the 0.005 band faster than with the linear case.

## 5 Experiment

The same algorithm as used in chapter 5 is implemented using a TUE/DACS with a sample frequency of 0.5 kHz. The setup is a dual rotary 4<sup>th</sup> order motion system. Again the settings are:

$$G_m(s) = \frac{100\pi^2}{s^2 + 16\pi s + 100\pi^2} \quad (13)$$

$$r(t) = 400t - \frac{20}{2\pi} \sin\left(\frac{2\pi}{0.05}t\right) \quad 0 \leq t \leq 0.05 \quad (14)$$

$$r(t) = 20 \quad t > 0.05$$

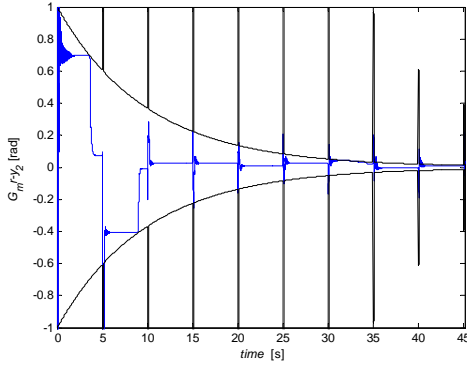
$$\Delta(t) = 0.002 + e^{(-0.1t)} \quad (15)$$

$$\Sigma(0) = 10^4 I_n \quad (16)$$

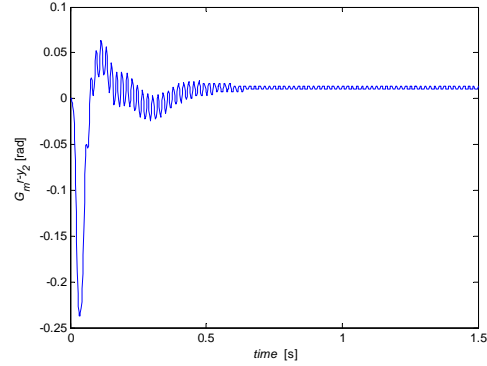
Again the controller candidates will be discussed as mentioned in chapter 4.

### 5.1 Linear controller

The first experiment is done using a linear controller with a vector field  $w(u(t), y(t)) = \left[ u(t), \frac{1}{s+1}u(t), y(t), \frac{1}{s+1}y(t) \right]$ . The experiment is initiated with a controller with  $\theta^*(0) = [3.5004 \ 16.3702 \ 2.6884 \ -54.7038]^T$ , these parameters were chosen because after some trial and error they proved to be able to keep the system stable enough for the algorithm to continue.



(a) Tracking error during the experiment, along with the bounds the error has to stay within.



(b) Tracking error when the last known good  $\theta$  is inserted in the controller.

Figure 5.1: Tracking error as a function of the time with a linear controller implemented.

Figure 5.1(a) shows the tracking error of the experiment when the EUC algorithm is running. Figure 5.1(b) shows the response to a skew-sinus when the last good controller parameter set is implemented. The maximum tracking error is 0.24 rad and the tracking error is within the 0.05 rad band after 0.18 sec.

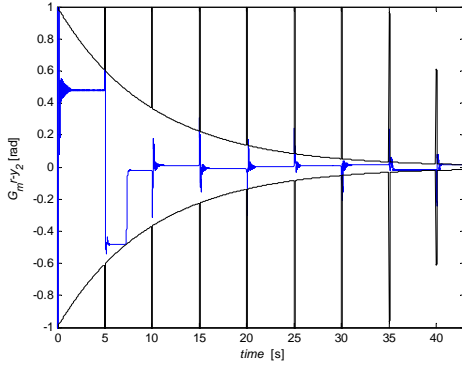
## 5.2 SPAN filter

An experiment is done with a vector field

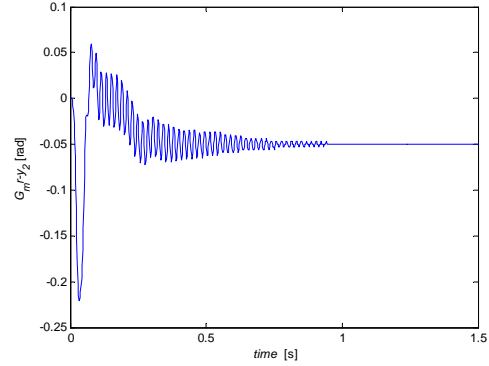
$$w(u(t), y(t)) = \left[ u(t), \frac{1}{s+1} u(t), y(t), \frac{1}{s+1} y(t), \left| \frac{30}{s+30} u(t) \right| \text{sign}\left(\frac{10/3}{s+60} u(t)\right) \right].$$

The simulation is initiated with a controller with parameter set

$$\theta^*(0) = [3.5004 \quad 16.3702 \quad 2.6884 \quad -54.7038 \quad 0]^T.$$



(a) Tracking error during the experiment, along with the bounds the error has to stay within.



(b) Tracking error when the last known good  $\theta$  is inserted in the controller.

Figure 5.2: Tracking error as a function of the time with SPAN filter implemented.

Figure 5.2 shows the tracking error during the experiment. Figure 5.2(b) shows that the performance does not outperform a linear controller. The maximum tracking error seems to be a little smaller but the final error is almost 0,05 rad. This is caused by the fact that the experiment runs for only 45 sec. EUC is not able to find a controller parameter set that forces the final tracking error to go to 0.

## 5.3 Nonlinear gain

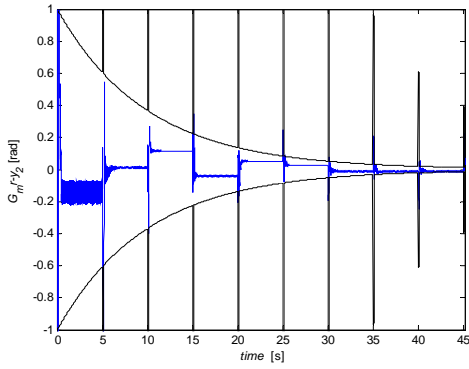
Another simulation is performed again with the same settings with a vector field

$$w(u(t), y(t)) = \left[ u(t), \frac{1}{s+1} u(t), y(t), \frac{1}{s+1} y(t), K_{nl}(e) u(t) \right].$$

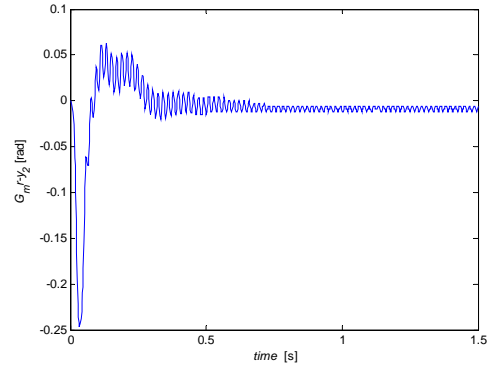
Here  $K_{nl}(e)$  is a function of the error as plotted in figure 4.2, this function is symmetrical with respect to the y-axis. The simulation is initiated with controller

$$\theta^*(0) = [3.5004 \quad 16.3702 \quad 2.6884 \quad -54.7038 \quad 0]^T.$$





(a) Tracking error during the experiment, along with the bounds the error has to stay within.



(b) Tracking error when the last known good  $\theta$  is inserted in the controller.

Figure 5.3: Tracking error as a function of the time with a nonlinear gain controller implemented.

Figure 5.3 shows the tracking error of the EUC experiment when a nonlinear gain part is added to the linear controller. The tracking error of the response to a skew-sinus when the last known good controller parameter set is used is shown in figure 5.3(b). The maximum tracking error and the time to get within the 0.05 rad band are bigger than when only a linear controller is used.

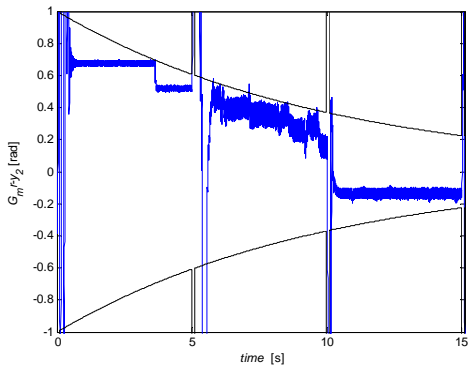
#### 5.4 Arbitrary controller

Another simulation is performed again with the same settings with a vector field

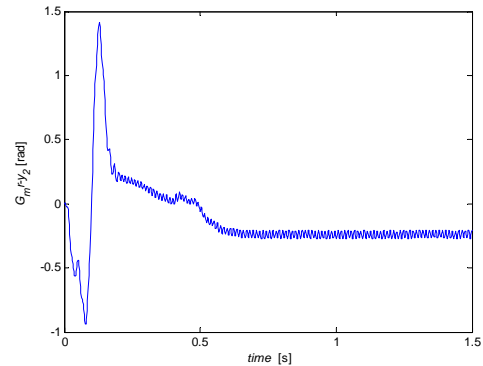
$$w(u(t), y(t)) = \begin{bmatrix} u(t), \frac{1}{s+1}u(t), y(t), \frac{1}{s+1}y(t), |e(t)|, \text{sign}(e(t)), \text{sat}(e(t)/0.01), \\ \text{dead zone}(e(t)/0.01), \text{relay}(-0.01 : e(t) : 0.01), \text{backslash}(e(t) : 0.02) \end{bmatrix}. \text{ The}$$

simulation is initiated with a controller parameter set

$$\theta^*(0) = [3.5004 \quad 16.3702 \quad 2.6884 \quad -54.7038 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T.$$



(a) Tracking error during the experiment, along with the bounds the error has to stay within.



(b) Tracking error when the last known good  $\theta$  is inserted in the controller.

Figure 5.4: Tracking error as a function of the time with a nonlinear gain controller implemented.

Figure 5.4(a) shows that the experiment only runs for 15 sec. After 15 sec the system becomes to unstable for the EUC algorithm to continue. This of course leads to a very bad performance as shown in figure 5.4(b).

## 6 Conclusions and recommendations

A number of nonlinear candidates were implemented in a EUC algorithm.

### 6.1 Conclusions

Because of the limitations with respect to the controller structure it is not possible to design a nonlinear controller using solely EUC.

- The requirement that  $u(t)$  can be determined uniquely from the other data leads to the fact that it is not possible to use only a nonlinear controller. Therefore the nonlinear controllers used in the paper are always implemented parallel with a linear controller.
- The requirement that the controller has to be of structure  $r(t) = w(u(t), y(t))^T \theta^*(t)$  with  $\theta^*(t)$  the set of controller parameters means that when a nonlinear controller is implemented only the best multiplication factor can be determined using EUC. The parameters within the nonlinear part have to be tuned by hand.
- Every time the controller parameters are changed the system has to endure a shock. This can lead to falsely falsified controller parameters. As stated in (van Helvoort, 2005, section 2.4) several update scenarios are possible. Of these possibilities the least aggressive one was used to minimize this shock.
- EUC starts from the assumption that the performance specification is given by  $|L(s)(y(t) - y_m(t))| \leq \Delta(t) \quad \forall t$  which in other words leads to good performance with respect to tracking. Thus it is not possible to give performance specifications with respect to robustness.

### 6.2 Recommendations

During the research, a point of further investigation was noted:

- The normal Unfalsified Control concept has fewer limitations on the controller structure and the performance specification. This might lead to the ability to determine nonlinear controller parameters.

## References

- [1] H. Bode, *Network analysis and feedback amplifier design*, D. van Nostrand, New York, 1945
- [2] Michael G. Safonov, and Tung-Ching Tsao, *The Unfalsified Control Concept and Learning*, in *Proc. 33th Conf. on Decision and Control*, 1994
- [3] M.M. Seron, J.H. Braslavsky, and G.C. Goodwin, *Fundamental Limitations in Filtering and Control*. Springer-verlag, London, 1997.
- [4] Michael G. Safonov, and Tung-Ching Tsao, *The Unfalsified Control Concept and Learning*, To appear *IEEE transactions on automatic control*, 1997
- [5] Fabricio B. Cabral, and Michael G. Safonov, *Unfalsified model reference adaptive control using the ellipsoid algorithm*, in *Proc. 42nd IEEE Conference on Decision and Control*, 2003
- [6] J. van Helvoort, B. de Jager, and M. Steinbuch, “Unfalsified control using an ellipsoidal unfalsified region applied to a motion system,” in *Proc. 16th IFAC World Congress*, 2005

## **Appendices**

### A. Matlab/Simulink scheme of the EUC algorithm

In this appendix the Matlab/Simulink model as used for the simulations and experiments is given.

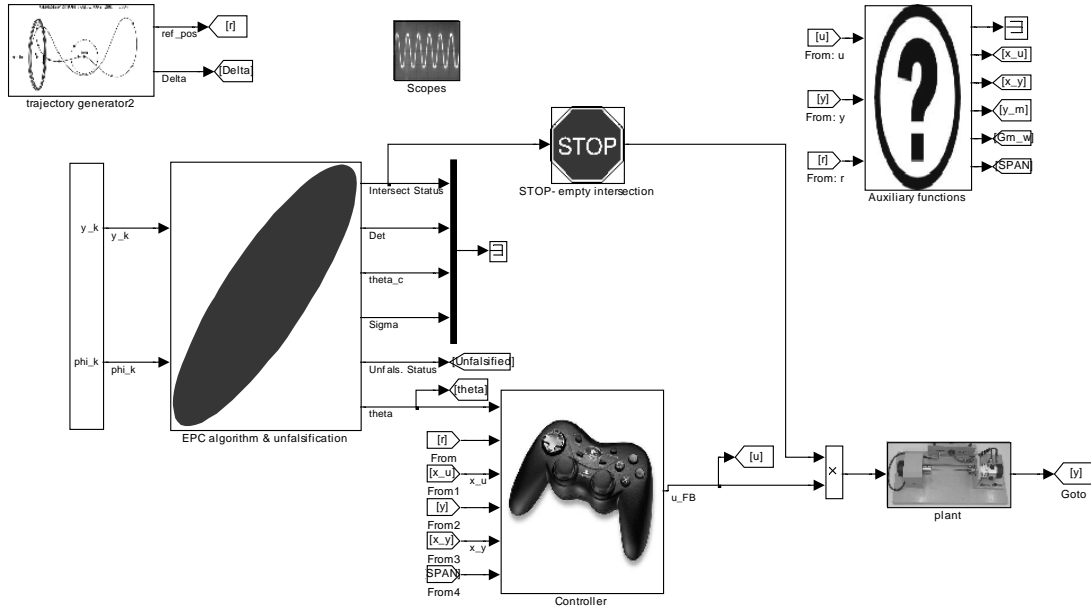


Figure A.1 Matlab/Simulink block scheme.

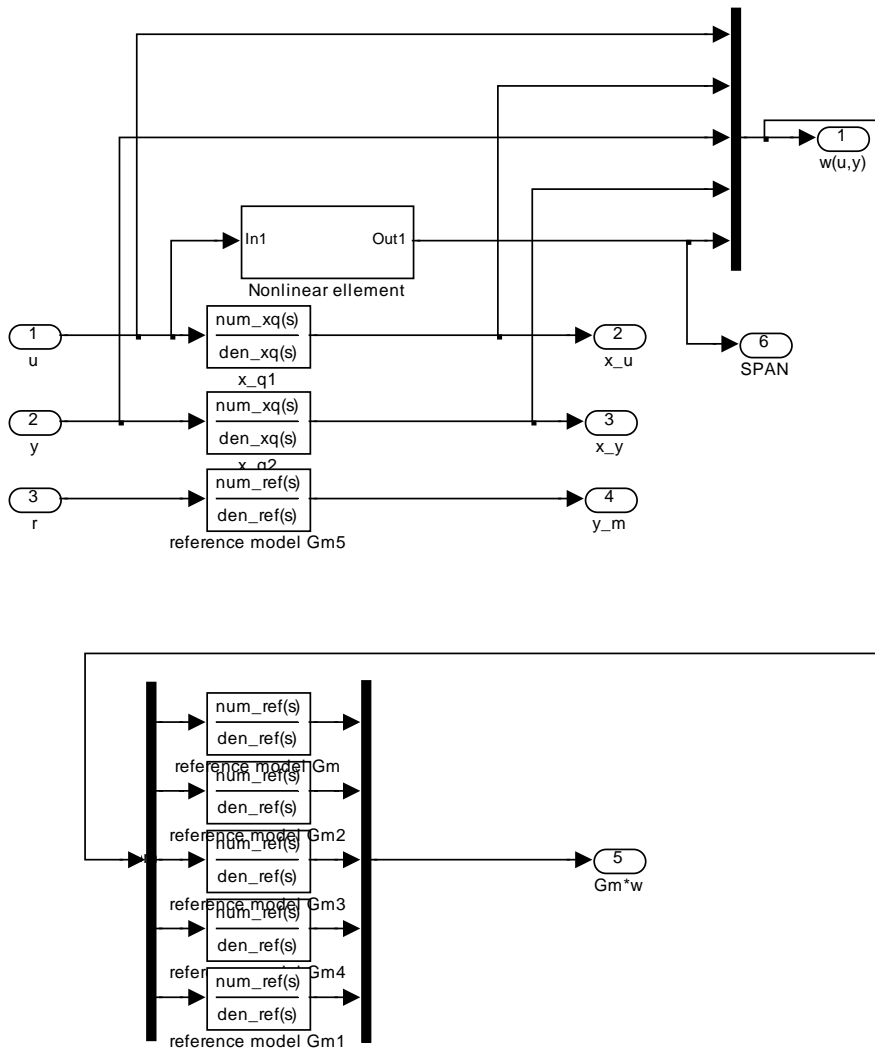


Figure A.2 Auxiliary function used in the Matlab/Simulink model

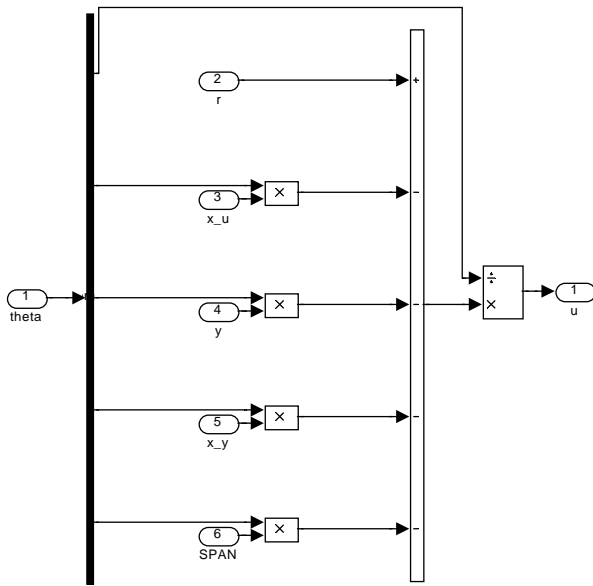


Figure A.3 Controller used in the Matlab/Simulink model

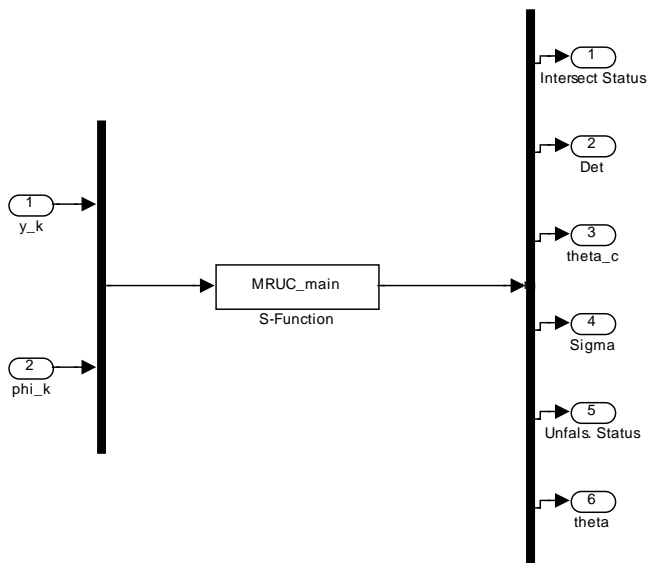


Figure A.4 Unfalsification algorithm used in the Matlab/Simulink model