Internal model principle for ellipsoidal unfalsified control

Steennis, A.R.

Published: 01/01/2006

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 author's 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

Citation for published version (APA):
Internal Model Principle for Ellipsoidal Unfalsified Control

A.R. Steennis
DCT 2006.039

Traineeship report

Coach(es):  ir. J. van Helvoort
Supervisor:  prof.dr.ir. M. Steinbuch
            dr.ir. B. de Jager

Technische Universiteit Eindhoven
Department Mechanical Engineering
Dynamics and Control Technology Group

Eindhoven, July, 2006
## Contents

1 Introduction 4

2 Ellipsoidal Unfalsified Control 5  
2.1 Concept of Ellipsoidal Unfalsified Control, EUC 5  
2.2 Control Structure 5  
2.3 Controller from EUC 8  

3 Internal Model Principle 10  
3.1 Concept of Internal Model Principle, IMP 10  
  3.1.1 Reference signal, \( r(t) \) 10  
  3.1.2 Disturbance, \( w(t) \) 11  
3.2 IMP in State Space 12  
3.3 Controller transfer function from State Space 14  
3.4 Controller from inverse notch with lead-lag 15  

4 Plant I 17  
4.1 Plant, \( P_1 \) 17  
4.2 Comparison of controller from IMP with EUC 18  
  4.2.1 Controller from IMP 18  
  4.2.2 Controller from EUC 19  
4.3 Controller from IMP inside or outside the unfalsified area 22  
4.4 Beginning notch as initial conditions in the algorithm 24  

5 Plant II 26  
5.1 Plant, \( P_2 \) 26  
  5.1.1 General Plant 26  
  5.1.2 Plant parameters 27  
5.2 Possible solutions to solve the numerical error 28  
  5.2.1 A high frequent adaption of the Plant, \( P_{2b} \) 28  
  5.2.2 A low frequent adaption of the Plant, \( P_{2c} \) 29  
  5.2.3 Lead-lag controller in front of the Plant 30  
5.3 Comparison of controller from IMP with EUC 31  
  5.3.1 Controller from IMP 31  
  5.3.2 Controller from EUC 32  
5.4 Controller from IMP inside or outside the unfalsified area 33  
5.5 Beginning notch as initial conditions in the algorithm 35  

6 Conclusion 37  

7 References 38
8 Appendix

8.1 Calculations ................................................................. 39
  8.1.1 Model formulation of controller ..................................... 39
  8.1.2 From State Space to transfer function .......................... 39
  8.1.3 $\Sigma_{tk1} \text{ and } \theta_{c1}$ ........................................... 41
  8.1.4 $\Sigma_{tk2} \text{ and } \theta_{c2}$ ........................................... 41
Chapter 1

Introduction

In this research two major theories will be compared with each other: Internal Model Principle with Ellipsoidal Unfalsified Control.

Internal Model Principle, IMP, implies that, a disturbance behind the plant with a specific frequency can be neutralized by implementing in the controller the oscillator with that specific frequency. Ellipsoidal Unfalsified Control, EUC, is a theory to determine which control parameter sets, for a specified control structure with arbitrary plant, satisfy a given performance specification. It is investigated to what extent the Internal Model Principle is present in controllers from Ellipsoidal Unfalsified Control. And, if the IMP is not present in a controller from EUC, it is researched whether a controller with IMP lays in the unfalsified area of the EUC algorithm or not, and, in what way adaptations in the initial conditions of the EUC program influences the controller.

In chapter 2 the theory of Ellipsoidal Unfalsified Control is explained and how this theory can be used in a simulation model. Eventually a general controller from EUC is determined and used in this research.

In chapter 3 the theory of Internal Model Principle is explained. A disturbance behind an arbitrary plant in a control structure is introduced and, after some considerations, two general controllers with IMP are determined. One deviated from a state space formulation and one deviated from an inverse notch with a lead-lag function. Both controllers are used in this research.

In chapter 4 both theories, EUC and IMP, are compared with each other. For this, plant $P_1$ is introduced and used do determine controllers from EUC and IMP. These controllers are compared and as a result of this comparison a new plant $P_2$ is designed.

In chapter 5 the new plant $P_2$ is introduced, but, when running the EUC algorithm with $P_2$, a numeric error arises. This numeric error will be solved and both controllers from EUC and IMP can be compared.

In chapter 6 a conclusion is drawn about the comparison of controllers from EUC with IMP.
Chapter 2

Ellipsoidal Unfalsified Control

In this chapter, first the concept of Ellipsoidal Unfalsified Control is introduced and how this theory can be used in a simulation model, section 2.1. In section 2.2, Ellipsoidal Unfalsified Control is explained more specific, with several equations and a control structure model. Finally, in section 2.3, a controller from Ellipsoidal Unfalsified Control is determined with some considerations been made.

2.1 Concept of Ellipsoidal Unfalsified Control, EUC

Ellipsoidal Unfalsified Control, EUC, (Van Helvoort, J., De Jager, B. & Steinbuch, M. (2005)), is a theory to determine which control parameter sets, for a specified control structure with arbitrary plant, satisfy a given performance specification, see section 2.2. That given performance specification is expressed as the maximum allowed tracking error of the control structure. This theory is implemented into a simulation model and, when running the program, a better performance is achieved by letting the algorithm discard control parameter sets.

Initially, the algorithm starts with parameter sets which satisfy the performance specification easily because of the high maximum allowed tracking error at the starting point. But gradually, the allowed tracking error reduces and fewer parameter sets satisfy the performance specification. So, when running the program, the area with satisfied parameter sets reduces, or, the algorithm discards parameter sets which do not satisfy the performance specification. These parameter sets are falsified and therefore discarded. The parameter sets within the area are unfalsified and therefore retained. The area with unfalsified parameter sets is of an ellipsoidal shape which now totally explains the name: Ellipsoidal Unfalsified Control.

2.2 Control Structure

To determine whether parameter sets are within the unfalsified area, the ellipsoid, or not, and to determine which new parameter set has to be implemented in the controller, the general reference unfalsified control setup is used, see Fig. 2.1. Normally, the plant P is unknown, but its input \( u(t) \) and output \( y(t) \) can be measured. The simulation model works discretely and therefore the controller is modelled in z-domain, which gives a better insight of what the algorithm is doing every sample time:

\[
\frac{1}{z} t_k = t_{k-1}
\] (2.1)

From the measured values \( u(t) \) and \( y(t) \) the program calculates \( w(u(t), y(t)) \) as the following vector:

\[
w(u(t), y(t)) = \begin{bmatrix} u & \frac{1}{z} u u_{45} & \frac{1}{z} u u & \frac{1}{z} u y & \frac{1}{z} y y & \frac{1}{z} y y \end{bmatrix}
\] (2.2)
Figure 2.1: Blockscheme of a general model reference unfalsified control setup

From \( w(u(t), y(t)) \) a set of virtual references, \( R_v(t, \theta) \), can be derived by the following equation:

\[
R_v(t, \theta) = \left\{ r_v(t) = w(u(t), y(t))^T \theta | \theta \in \mathbb{R}^p \right\}
\]  

(2.3)

Where \( \theta \) symbolizes:

\[
\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \ldots \ \theta_p]^T
\]  

(2.4)

So, \( \theta \) symbolizes a vector with \( p \) components. In theory \( R_v(t, \theta) \) symbolizes a set of infinite possible signals, and can be calculated with infinite different combinations of \( \theta \). It says that if these control parameter sets \( \theta \) where implemented in the controller \( C \), with \( R_v(t, \theta) \) as reference, the same values for \( u(t) \) and \( y(t) \) would be calculated as have been measured.

Subsequently, \( R_v(t, \theta) \) will be the reference signal of a stable reference model \( G_m(z) \), which is a model for the desired closed loop behavior. This model calculates an output \( Y_m(t) \), which is defined as

\[
Y_m(t) = \left\{ y_m(t) = G_m(z)w(u(t), y(t))^T \theta | \theta \in \mathbb{R}^p \right\}
\]  

(2.5)

which can be compared with the measured value \( y(t) \). The difference of these two values represents an error which can be compared with the maximum allowed tracking error as mentioned in section 2.1. The maximum allowed tracking error, called the threshold value, can be expressed as \( \Delta(t) \), which gives the following equation

\[
|L(z)(y(t) - y_m(t))| \leq \Delta(t)
\]  

(2.6)

with \( L(z) \) some stable filter. When combining (2.5) and (2.6), the new region of control parameter sets, \( \mathcal{F}(t) \) with a maximum error of \( \Delta(t) \) that is unfalsified, can be derived:

\[
\mathcal{F}(t) = \{ \theta | -\Delta(t) \leq L(z) \cdot (y(t) - G_m(z)w(u(t), y(t))^T \theta) \leq \Delta(t) \}
\]  

(2.7)

which defines two half-spaces, like Fig. 2.2(a). However, in this research, \( \theta \) symbolizes more than three parameters \( (p = 8) \), which can not be visualized in a figure anymore.

All these steps mentioned above are taking place at time \( t = t_k \). Keeping in mind the existence of an ellipsoid formed at time \( t = t_k - 1 \), an intersection of these areas is presumable, like Fig. 2.2(b).

As a result of that intersection, a new ellipsoid \( \varepsilon(t_k) \) can be formed which is defined by

\[
\varepsilon(t_k) = \{ \theta | (\theta - \theta_c(t_k))^T \Sigma^{-1}(t_k) \cdot (\theta - \theta_c(t_k)) \leq 1 \}
\]  

(2.8)
with its center defined by the vector $\theta_c$ and its shape by the the matrix $\Sigma(t_k)$. There are different cases distinguished in the determination of $\theta_c$ and $\Sigma(t_k)$, (Van Helvoort, J., De Jager, B. & Steinbuch, M. (2005)), which is beyond this report.

With the obtainment of the new ellipsoid, $\varepsilon(t_k)$, a new controller parameter set $\theta^*$ can be chosen and implemented in the controller C, see Fig. 2.1. Once the new parameter set is implemented, there is the point from where everything is starting over again and when computing this routine for a longer period, a better performance is achieved, see Fig. 2.3.

![Figure 2.2: Two half-spaces and ellipsoid with intersection](image)

![Figure 2.3: Tracking error and threshold value. Red line: tracking error; blue line: $\Delta(t)$; green line: $-\Delta(t)$](image)
2.3 Controller from EUC

To obtain a transfer function of the controller $C$, a better understanding of the scheme is necessary, see Fig. 2.4. In this figure the dotted line represents the controller $C$ from Fig. 2.1 with $C_1(z)$ and $C_2(z)$ transfer functions containing the parameter set $\theta$, (2.4), with $p = 8$.

$C_1(z)$ symbolizes:

$$C_1(z) = \frac{1}{\theta_1 + \frac{1}{z-0.99}\theta_2 + \frac{1}{z}\theta_3 + \frac{1}{z^2}\theta_4}$$  \hspace{1cm} (2.9)

And $C_2(z)$:

$$C_2(z) = \frac{1}{\theta_5 + \frac{1}{z}\theta_6 + \frac{1}{z^2}\theta_7 + \frac{1}{z^3}\theta_8}$$  \hspace{1cm} (2.10)

The interesting part of this structure is the transfer function between $y(t)$ and $u(t)$ which can be calculated by $\frac{U(z)}{Y(z)}$. When taking into account $r(t)$ as reference, the following relation between $y(t)$ and $u(t)$ can be derived:

$$-\frac{C_1(z)}{C_2(z)}Y(z) + C_1(z)R(z) = U(z)$$  \hspace{1cm} (2.11)

From here can be seen that the division $\frac{U(z)}{Y(z)}$ will be complicated to calculate when $r(t) \neq 0$. However, eventually, see section 3.1.2, a disturbance behind the Plant has to be described into the control parameters, and, because of that, not necessarily the reference signal. The control structure is of a linear form, so, the superposition principle holds, (H.A. van Essen & J.J.M. Rijpkema (2003/2004)). Keeping this in mind, if $r(t) = 0$, it is a good solution to let $r(t) = 0$.

Now the Controller from EUC can be written as:

$$C_{EUC}(z) = \frac{-U(z)}{Y(z)} = \frac{C_1(z)}{C_2(z)} = \frac{\theta_5 + \frac{1}{z}\theta_6 + \frac{1}{z^2}\theta_7 + \frac{1}{z^3}\theta_8}{\theta_1 + \frac{1}{z-0.99}\theta_2 + \frac{1}{z}\theta_3 + \frac{1}{z^2}\theta_4}$$  \hspace{1cm} (2.12)

After multiplication of numerator and denominator with $z-0.99$ and rewriting, $C_{EUC}(z)$ will be like:

$$C_{EUC}(z) = \frac{\theta_5 z^4 + (\theta_6 - 0.99\theta_4)z^3 + (\theta_7 - 0.99\theta_6)z^2 + (\theta_8 - 0.99\theta_7)z - 0.99\theta_8}{\theta_1 z^4 + (\theta_3 - 0.99\theta_1 + \theta_2)z^3 + (\theta_4 - 0.99\theta_3)z^2 - 0.99\theta_4 z}$$  \hspace{1cm} (2.13)

However, in this equation the parameters $\theta_5$ to $\theta_8$, in the numerator, are dependent on each other. So it will not always be possible to write the controller found from Internal Model Principle into this equation, because, there are more equations than parameters to be solved. Therefore,
returning to the basic controller $C$, $C_1(z)$ of (2.9) has to be adapted into another form which eventually gives another controller. With this adaption a new $C_1(z)$ will be defined as:

$$C_1(z) = \frac{1}{\theta_1 + \frac{1}{z}\theta_2 + \frac{1}{z^2}\theta_3 + \frac{1}{z^3}\theta_4}$$  \hspace{1cm} (2.14)$$

With this adapted $C_1(z)$ and the preserved $C_2(z)$, $C_{EUC}(z)$ will be like:

$$C_{EUC}(z) = \frac{\theta_5z^3 + \theta_6z^2 + \theta_7z + \theta_8}{\theta_1z^3 + \theta_2z^2 + \theta_3z + \theta_4}$$  \hspace{1cm} (2.15)$$

This $C_{EUC}(z)$ will be used in the rest of this research.
Chapter 3

Internal Model Principle

In this chapter, first the concept of Internal Model Principle is introduced, section 3.1. For that, different methods like the reference signal \( r(t) \) and the disturbance \( w(t) \) are used, sections 3.1.1 and 3.1.2. Secondly, the Internal Model Principle is written in state space structure, and from that state space structure a controller is calculated, section 3.3. Finally, in section 3.4, in another way, with a inverse notch function and lead-lag, a controller is determined. Both methods, (state space and notch with lead-lag) are used in this research.

3.1 Concept of Internal Model Principle, IMP

3.1.1 Reference signal, \( r(t) \)

To understand the concept of Internal Model Principle, a basic control structure is given, see Fig. 3.1. In this structure the goal is to get \( y(t) \), in steady state situation, equal to \( r(t) \), or, the tracking error equal to zero. This goal can be achieved with, for example, a PID-controller according to the properties of plant \( P \). But when \( r(t) \) is of a regular form with a specific frequency, like a sinusoid, some extra detail can be given to the controller to get \( y(t) \) equal to \( r(t) \).

To explain which detail can be given, the transfer function from \( r(t) \) to \( y(t) \), \( T_{cl}(s) \), needs to be known. Next steps are derivations to that transfer function:

\[
E(s) = R(s) - Y(s), \quad E(s)C(s)P(s) = Y(s) \tag{3.1}
\]

So,

\[
(R(s) - Y(s))C(s)P(s) = Y(s) \tag{3.2}
\]

which is equivalent to:

\[
C(s)P(s)R(s) = Y(s)(1 + C(s)P(s)) \tag{3.3}
\]

So, transfer function \( T_{cl}(s) \) will be like:

\[
T_{cl}(s) = \frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} \tag{3.4}
\]
Exact following of the reference signal will be achieved when \( y = r \). Or, when utilizing (3.4):

\[
\frac{Y(s)}{R(s)} = 1 = \frac{C(s)P(s)}{1 + C(s)P(s)}
\]  

(3.5)

This can only be reached, when \( C(s) \) or \( P(s) \) is infinite. In this research \( P(s) \) will be a transfer function of a motion system, so only \( C(s) \) can be adapted:

\[
\frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{\infty P(s)}{1 + \infty P(s)} \approx \frac{\infty P(s)}{\infty P(s)} = 1
\]

(3.6)

However, it would not be wise letting \( C(s) \) be a gain of an infinite value, which can induce instability and beside that; it is not even realizable in practice when there is no equipment to achieve that.

Though it is not possible to let \( C(s) \) be infinite for every frequency, it is possible for one specific frequency, see Fig. 3.3. This \( C(s) \) with an infinity at one frequency can be realized by implementing, in the controller, the oscillator with that specific frequency \( \omega_0 \), which is known as the internal model of the input generator. Because of this infinity at only one frequency, it is possible to stabilize the total open loop system, because the bandwidth can still be placed. (B.A. Francis & W.M. Wonham (1975)).

### 3.1.2 Disturbance, \( w(t) \)

In this research, a disturbance behind the plant \( P \), has to be neutralized, see Fig. 3.2. As argued in section 2.3, the reference signal \( r(t) \) will be zero in this situation. In this structure the goal is to neutralize the influence of the disturbance \( w(t) \) on output \( y(t) \). Therefore, the transfer function from \( w(t) \) to \( y(t) \), which is called the sensitivity, \( S(s) \), needs to be known:

\[
S(s) = \frac{Y(s)}{W(s)} = \frac{1}{1 + C(s)P(s)}
\]

(3.7)

where \( W(s) \) is the disturbance and \( Y(s) \) the output of the control structure in Laplace domain. Neutralizing the influence of \( w(t) \) on output \( y(t) \) means:

\[
\frac{Y(s)}{W(s)} = 0 = \frac{1}{1 + C(s)P(s)}
\]

(3.8)

This can only be reached, when \( C(s) \) is infinite:

\[
\frac{1}{1 + C(s)P(s)} = \frac{1}{1 + \infty P(s)} = \frac{1}{\infty P(s)} = 0
\]

(3.9)

From here the same principle is valid as in section 3.1.1, so the Internal Model Principle can be applied in the same way.

![Control structure with disturbance](image-url)
3.2 IMP in State Space

To apply the Internal Model Principle theory, explained in chapter 3.1, the following disturbance function, \( w(t) \) is used

\[
w(t) = \sin(\omega_0 t)
\]

where \( \omega_0 \) symbolizes a given frequency. When transforming (3.10) to Laplace domain, the following function, \( W(s) \) is derived:

\[
W(s) = \frac{1}{s^2 + \omega_0^2}
\]

This calculated transfer function gives an infinite value as absolute magnitude at that specific frequency which can be seen in the Bode plot of this transfer function, Fig. 3.3, where \( \omega_0 = 1 \) rad/s = \( \frac{1}{2\pi} \) Hz. Clearly can be seen a gain of infinity at only that specific frequency of \( \frac{1}{2\pi} \) Hz,

![Bode Diagram](image)

Figure 3.3: Bode plot of the transfer function \( W(s) \)

so the Internal Model Principle theory as explained in chapter 3.1.1 can be applied.

Transfer function \( W(s) \) can be written in State Space and therefore in the following matrix form

\[
\dot{x} = A_w x + B_w u
\]

\[
y = C_w x
\]

where

\[
A_w = \begin{bmatrix}
0 & -\omega_0^2 \\
1 & 0
\end{bmatrix}, \quad B_w = \begin{bmatrix}
1 \\
0
\end{bmatrix}, \quad C_w = \begin{bmatrix}
0 & 1
\end{bmatrix}
\]

(3.14)

Combining (3.12) and (3.13), the following equations can be derived:

\[
\begin{bmatrix}
0 & -\omega_0^2 \\
1 & 0
\end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}
\]

(3.15)

and

\[
\begin{bmatrix}
0 & 1
\end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = y
\]

(3.16)
which gives after a few calculations and transforming to Laplace again:

\[ X_1(s) = (-\omega_0^2 Y(s) + U(s)) \frac{1}{s}, \quad Y(s) = X_1(s) \frac{1}{s} \]  

(3.17)

These equations can be implemented in a block diagram in control form, see Fig. 3.4. This block diagram will be the internal model in the eventually obtained controller. However, when implementing this block as controller in front of the Plant P, it does not provide the assurance of a stable control structure. For that, the closed loop poles have to be placed in the left half-plane of the s-plane. The open-loop poles can be moved by implementing state feedback gains in front of the internal model and by multiplying state feedback gains with the states of the plant. This gives the following block diagram, with plant \( P = \frac{1}{s(s+1)} \) as example, see Fig. 3.5, (Gene F. Franklin, J. David Powell, Abbas Emami-Naeini, (2002)).

The overall system of this block diagram can be described in an error space. In standard state-variable form, the equations are

\[ \dot{z} = A_t z + B_t \mu \]  

(3.18)

where \( z = [e \quad \dot{e} \quad \xi^T]^T \) and

\[ A_t = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad B_t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \]  

(3.19)
When solving the equation \( \det(sI - (A_t - B_tK)) \) with \( K = [K_2 \ K_1 \ K_{01} \ K_{02}] \) to

\[ s^4 + (1 + K_{02})s^3 + (1 + K_{01})s^2 + (K_1 + 1 + K_{02})s + K_{01}K_2 = 0 \]  

the state feedback gains can be selected by pole assignment by placing these poles in the left half-plane of the s-plane. When the plant, \( P \), is of a higher order, it has more states, so simply more poles have to be placed. Therefore more state feedback gains have to be determined.

### 3.3 Controller transfer function from State Space

Eventually, the controller in Fig. 3.5, the dotted line, has to be compared with controllers from EUC which can be done by Bode plots. A Bode plot of the controller from IMP can be compared with a Bode plot of the controller from EUC. To get a Bode plot from a controller, first a transfer function has to be derived. So from the State Space diagram, the controller in Fig 3.5 has to be transformed in a transfer function. Therefore, another plant will be introduced

\[ P(s) = \frac{100s^2 + 20s + 2}{2s^3 + 2s^2} \]  

which is one of the eventual plants in which controllers from IMP have to be compared with controllers from EUC, see section 4.1. In matrix form, the plant will be like:

\[
\begin{bmatrix}
-1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
+ 
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix}
u = 
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3
\end{bmatrix}
\]  

(3.22)

and

\[
\begin{bmatrix}
50 & 10 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= y
\]  

(3.23)

When calculating (3.22) and (3.23) in several equations, the states of the plant \( x_1, x_2 \) and \( x_3 \) can be derived:

\[ X_1(s) = \frac{1}{s + 1}U(s), \quad X_2(s) = \frac{1}{s^2 + s}U(s), \quad X_3(s) = \frac{1}{s^3 + s^2}U(s) \]  

(3.24)

In Fig. 3.5, \( u \) can be calculated with (3.25). The only difference is, as a result of a higher order plant, there is one more state.

\[ U(s) = U_{IMP}(s) - K_{01}X_1(s) - K_{02}X_2(s) - K_{03}X_3(s) \]  

(3.25)

\( U_{IMP}(s) \) will be like:

\[ U_{IMP}(s) = \frac{-K_1s - K_2}{s^2 + \omega_0^2}E(s) \]  

(3.26)

When combining (3.24), (3.25) and (3.26), the following relation can be derived:

\[ (1 + K_{01}\frac{1}{s + 1} + K_{02}\frac{1}{s^2 + s} + K_{03}\frac{1}{s^3 + s^2})U(s) = \frac{-K_1s - K_2}{s^2 + \omega_0^2}E(s) \]  

(3.27)

After algebraic reshaping, the Controller \( C_{IMP}(s) \) calculated by \( -\frac{U(s)}{E(s)} \) will be expressed as

\[ C_{IMP}(s) = \frac{K_1s^4 + (K_1 + K_2)s^3 + K_2s^2}{s^5 + (1 + K_{01})s^4 + (K_{02} + \omega_0^2)s^3 + (K_{03} + \omega_0^2 + \omega_0^2K_{01})s^2 + K_{02}\omega_0^2s + K_{03}\omega_0^2} \]  

(3.28)
3.4 Controller from inverse notch with lead-lag

As written in section 3.1, a sinusoid as disturbance can be neutralized by \( C(s) \) with an infinity at one frequency, the frequency of the disturbance. To neutralize the disturbance, the model of the disturbance can be implemented in the controller, as demonstrated in section 3.2, which induces an infinity at that specific frequency. Besides a sinusoid as internal model, also an inverse notch is able to induce an infinity at one specific frequency. The general inverse Notch-equation if of the form:

\[
N_i(s) = \frac{\frac{1}{f_1^2} s^2 + \frac{2\beta_1}{f_1} s + 1}{\frac{1}{f_2^2} s^2 + \frac{2\beta_2}{f_2} s + 1}
\]  
(3.29)

with \( f_a \) and \( f_b \) the peak frequencies. Next \( \beta_1 \) and \( \beta_2 \) together determines the height of the peaks with the relation \( \frac{\beta_1}{\beta_2} \). To apply an infinite gain at one specific frequency, like \( \frac{1}{f_1} \) Hz, or, 1 rad/s, the following values for these parameters have been chosen:

\[
f_a = f_b = \omega_0 = 1 \text{ rad/s}, \quad \beta_1 = 1, \quad \beta_2 = 0
\]  
(3.30)

From this transfer function, (3.29), a Bode plot can be drawn, Fig. 3.6(a). Clearly can be seen a gain of infinity at only that specific frequency of \( \frac{1}{f_1} \) Hz.

When implementing this inverse notch as controller in front of the plant, it achieves the same goal as implementing the internal model of the disturbance into the controller: neutralizing a sinusoid of the frequency of \( \frac{1}{f_1} \) Hz as disturbance. To stabilize the total open loop system, dependent on plant \( P \), a lead-lag controller has to be added to the controller. This lead-lag controller is like:

\[
LL(s) = P \frac{\frac{1}{f_1} s + 1}{\frac{1}{f_2} s + 1}
\]  
(3.31)

with \( f_1 \) and \( f_2 \) the break points locations in the magnitude curve (Bode plot) of the lead-lag transfer function, see Fig. 3.6(b). In this figure, \( f_1, f_2 \) and \( P \) has been chosen respectively 0.001 rad/s, 1 rad/s and 0.02. This lead-lag function has to be multiplied with the inverse notch to get a transfer function which includes both abilities, the inverse notch and lead-lag function. With loopshaping, the bandwidth can be placed via the lead-lag function while the infinite peak is still at the frequency of \( \frac{1}{f_1} \) Hz. The transfer function of the lead-lag multiplied with the notch, with an infinite gain at the frequency \( \omega_0 \) and break points at \( f_1 \) and \( f_2 \) will be like:

\[
N_iLL(s) = P \frac{\frac{1}{\omega_0^2 f_1^2} s^3 + \left( \frac{1}{\omega_0^2 f_1^2} + \frac{2}{\omega_0^2 f_2} \right) s^2 + \left( \frac{2}{\omega_0^2} + \frac{1}{f_2} \right) s + 1}{\frac{1}{\omega_0^2 f_2} s^3 + \frac{\frac{1}{\omega_0^2} s^2 + \frac{1}{f_2} s + 1}{f_2}}
\]  
(3.32)
Figure 3.7: Bode plot of the transfer function of a lead-lag multiplied with a notch

From this transfer function a Bode plot can be drawn, see Fig. 3.7.
Chapter 4

Plant I

In this chapter, first the plant, \( P_1 \), will be introduced, section 4.1. With this plant both controllers from EUC and IMP will be determined and compared with each other, section 4.2. After all it is interesting whether the controller from IMP lays inside or outside the unfalsified area, section 4.3 and in section 4.4 a beginning notch as initial conditions has been implemented in the algorithm to see whether the algorithm would finish this notch or not.

4.1 Plant, \( P_1 \)

To compare controllers from IMP with controllers from EUC, the same plant has to be implemented in both controller synthesis methods. That plant is of the form:

\[
P_1(s) = \frac{100s^2 + 20s + 2}{2s^3 + 2s^2}
\]  
(4.1)

From this transfer function, \( P_1 \), a Bode plot can be drawn, see Fig. 4.1. In this Bode plot, a phase

![Bode Diagram](image)

Figure 4.1: Bode plot of the transfer function of the plant, \( P_1 \), (4.1)

of \(-90^\circ\) at the frequency of 10 Hz and higher is noticed, which implies that the EUC program could stabilize the open loop system by placing the bandwidth on at least 10 Hz. So, to achieve the performance specification, just the bandwidth could be increased. However, the EUC program could also find parameter sets with a higher gain just around that specific frequency, and, it could also find parameter sets with the Internal Model Principle included, see section 3.1.
As explained in section 2.3, in this research the reference signal \( r(t) \) is equal to zero and the influence from the disturbance \( w(t) \) behind plant \( P \) to \( y(t) \) has to be neutralized, see Fig. 4.2.

### 4.2 Comparison of controller from IMP with EUC

#### 4.2.1 Controller from IMP

![Control structure with disturbance, \( w(t) \), where \( r(t) = 0 \)](image)

In this research, the disturbance \( w(t) \) is a sinusoid with a frequency of \( \omega_0 = 1 \text{ rad/s} = \frac{1}{2\pi} \text{ Hz} \). To find a controller including the Internal Model Principle, (3.28) can be used. Therefore the state feedback gains have to be determined. So first the closed-loop poles have to placed into the left half-plane of the s-plane. Therefore, the following closed loop poles have been selected:

\[
\begin{bmatrix}
-2 + 4j \\
-2 - 4j \\
-3 + 3j \\
-3 - 3j \\
-2
\end{bmatrix}
\] (4.2)

The selection of these poles are based on the location and values of the poles. All the poles are located in the left half-plane of the s-plane and conjugated to each other. Besides that, the values are chosen in the same range of 0 - 10. From these closed loop poles, the state feedback gains can be calculated with (3.20), which function first has to be adapted, because plant \( P_1 \) is a third order transfer function with one extra state, \( x_3 \). Therefore also an extra state feedback gain has be added to the IMP model, see Fig. 3.5.

Another way to determine the state feedback gains is by the Matlab statement:

\[
>> K = \text{place}(A_t, B_t, pc)
\]

with \( B_t \) and \( A_t \) the error-state matrices according to (3.18):

\[
A_t =
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 50 & 10 & 1 \\
0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}, \quad
B_t =
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
1
\end{bmatrix}
\] (4.3)

With the Matlab statement, the state feedback gain vector will be determined which is of following form, (4.4). This feedback gain vector can be implemented in the block diagram of the controller with Internal Model Principle, Fig. 3.5.

\[
K = [K_2 \quad K_1 \quad K_{01} \quad K_{02} \quad K_{03}] = [-5.49 \quad -14.65 \quad 11.00 \quad 813.64 \quad 725.49]
\] (4.4)
These state feedback gains can be implemented in (3.28) with \( \omega_0 = 1 \text{ rad/s} \) and gives the transfer function \( C_{IMP}(s) \):

\[
C_{IMP}(s) = \frac{-14.65s^4 - 20.15s^3 - 5.49s^2}{s^5 + 12s^4 + 814.6s^3 + 737.5s^2 + 813.6s + 725.5}
\] (4.5)

From this transfer function, \( C_{IMP}(s) \), a Bode plot can be drawn, see Fig. 4.3. This Bode plot from IMP can be compared with the Bode plot from EUC, section 4.2.2.

### 4.2.2 Controller from EUC

When running the program, the algorithm discards parameter sets which do not satisfy the threshold value, \( \Delta(t) \), see (2.6). This threshold value is expressed as:

\[
\Delta(t) = Ge^{(-Lt)} + T
\] (4.6)

which shape is shown in Fig. 4.4. In this graphic, the parameters \( G \) and \( T \) are visualized. It shows that \( G + T \) is the starting value of the function and \( T \) the final value when \( t \to \infty \), which will be the final maximum allowed tracking error. In the algorithm, the right combination of \( T \) and \( L \) has to be chosen to reach the desired tracking error at the desired time. Consider \( T_a = 0.01 \), which says that, when \( t \to \infty \) (steady state), the maximum allowed tracking error will be 0.01. For this, \( G_a = 6 \) and \( L_a = 0.1 \). When running the program, until it reached the desired tracking error at \( t = 250 \text{ s} \), a last parameter set, \( \theta_{end} \), has been given:

\[
\theta_{end} = [0.5304 -0.2510 -0.0678 0.0588 0.2566 0.2144 0.1623 0.1024]
\] (4.7)

This set of parameters, \( \theta^* \) in Fig. 2.1, which is the last set, given to the controller, can be implemented in (2.15) which gives the transfer function, \( C_{EUC_{1a}}(z) \) and can be compared with \( C_{IMP}(s) \).

\[
C_{EUC_{1a}}(z) = \frac{0.2566z^3 + 0.2144z^2 + 0.1623z + 0.1024}{0.5304z^3 - 0.251z^2 - 0.0678z + 0.05883}
\] (4.8)

From the transfer function, \( C_{EUC_{1a}}(z) \), a Bode plot can be drawn, see Fig. 4.5(a). It is very clear that there is no high gain at \( \frac{1}{2\pi} \text{ Hz} \). The open-loop control system, \( C_{EUC_{1a}}(s)P_1(s) \), shows where the bandwidth has been placed. For that, first the transfer function \( C_{EUC_{1a}}(z) \) has to be transformed to \( s \)-domain. This can be done by the Matlab statement:

\[
>> \text{d2c}(C_{EUC_{1a}},’tustin’)
\]
With the Matlab statement, the transfer function in s-domain can be calculated, which will be like:

$$C_{EUC_{1a}}(s) = \frac{0.156s^3 + 2139s^2 + 3.146 \cdot 10^6s + 8.989 \cdot 10^9}{s^3 + 6373s^2 + 7.523 \cdot 10^6s + 3.303 \cdot 10^9}$$  \hspace{1cm} (4.9)$$

Now the open loop control system, $C_{EUC_{1a}}(s)P_1(s)$, can be derived. From this open-loop transfer function a Bode plot, see Fig. 4.5(b), can be drawn. This figure shows a bandwidth of 21.32 Hz. At that frequency, the phase is -104°, which denotes a phase margin of 76°. This phase margin indicates the stability margin and is the amount by which the phase exceeds the critical phase of -180°. This phase margin of 76° is enough for a stable system. To reach this bandwidth, the program did not need a very high gain, because the Plant with a controller of $C_1(s) = 1$ has a bandwidth of 7.95 Hz, see Fig. 4.1. Therefore, a gain of 8.7 dB, see Fig. 4.5(a), is enough to reach a bandwidth of 21.32 Hz. The Plant with a controller of $C_1(s) = 1$ has a phase margin of 90°, so the reason to raise the bandwidth is to decrease the tracking error. To explain this, two sensitivity functions, $Y(s)$, have to be calculated. One with a controller of $C_1(s) = 1$, $S_1(s)$ and one with the controller of EUC, $S_{EUC_{1a}}(s)$:

$$S_1(s) = \frac{1}{1 + C_1(s)P_1(s)}$$  \hspace{1cm} (4.10)$$

$$S_{EUC_{1a}}(s) = \frac{1}{1 + C_{EUC_{1a}}(s)P_1(s)}$$  \hspace{1cm} (4.11)$$

These functions gives the following bode plots, Fig. 4.6(a) and 4.6(b). From the Bode plot of $S_1(s)$, Fig. 4.6(a) shows a sensitivity of -31.08 dB at the frequency of the disturbance, 1 rad/s. The disturbance, $w(t)$, is of a sinusoid with an amplitude of 1. So, in steady state situation, the tracking error $T_a$ of (4.6) can be combined with the sensitivity function. For that, the frequency of the disturbance $w(t)$, $\omega_0 = 1$ rad/s has to be implemented:

$$|S(j\omega_0)| = \frac{|Y(j\omega_0)|}{1} \leq T = 0.01$$  \hspace{1cm} (4.12)$$

At 1 rad/s, $S_1(s)$ has a magnitude of -31.08 dB. So, $|Y(j\omega_0)| = 10^{\frac{-31.08}{20}} = 0.03$. This magnitude of 0.03 can be implemented in (4.12), which gives: $\frac{0.03}{0.03} = 0.03 > 0.01$. 

Figure 4.4: Threshold value, $\Delta(t)$
until it reached the desired error, the controller
1
the disturbance,

\[ T = 0 \]

Next, consider

\[ T = 6 \]

and

\[ L = 0.15 \]

see (4.6). When running the program,

until it reached the desired error, the controller \( C_{EUC1a}(z) \) will be derived as:

\[
C_{EUC1a}(z) = \frac{0.6274z^3 + 0.3112z^2 + 0.07947z - 0.0516}{0.07178z^3 - 0.0606z^2 + 0.03948z - 0.01639}
\]

(4.13)

From this transfer function, \( C_{EUC1a}(z) \), a Bode plot can be drawn, see Fig. 4.7(a). In this Bode plot, Fig. 4.7(a) shows a gain of 29 dB up to about 10 Hz. In comparison with the Bode plot
of $C_{EUC_{1b}}(z)$, Fig. 4.5(a), there is more static gain. This implies that, the bandwidth has been placed at a higher frequency, namely 174.75 Hz, which is far more higher than the bandwidth with controller, $C_{EUC_{1b}}(z)$, (21.32 Hz). So, the maximum allowed tracking error, $T_b$, has been achieved by raising the bandwidth with 166.79 Hz and not by adding an inverse notch at that specific frequency of 1 rad/s and neither by implementing the internal model in the controller like Fig. 3.6.

Also with this controller, $C_{EUC_{1b}}(z)$, the sensitivity, $S_{EUC_{1b}}(s)$, of the control structure can be derived with (4.11), with the use of $C_{EUC_{1b}}(s)$ instead of $C_{EUC_{1b}}(s)$. The Bode plot of $S_{EUC_{1b}}(s)$ will be like Fig. 4.7(b). In this figure, at the frequency of $\frac{1}{2\pi}$ Hz, a magnitude of -60 dB can be noticed. So, with the use of (4.12), with $T_b=0.001$, $|S(j\omega_0)|$ can be calculated. For that, $|Y(j\omega_0)| = 10^{(-60)} = 0.001$ which is equal to the maximum allowed tracking error $T_b$.

### 4.3 Controller from IMP inside or outside the unfalsified area

In section 4.2.2, is showed that the EUC algorithm did not need to implement a control parameter set with Internal Model Principle or an inverse notch at that specific frequency of 1 rad/s with $T_a = 0.01$ and $T_b = 0.001$ in the controller. It could be, the algorithm discarded parameter sets with too low bandwidths, and retained the others. So, there could be a chance that the parameter set with the Internal Model Principle has been falsified and therefore discarded. In that case, the decrease of the maximum allowed tracking error $T$, to let the algorithm implement the controller with Internal Model Principle in the controller, would be nonsense, as there will be no controller like that in the unfalsified ellipsoidal area anymore.

To determine whether the controller from IMP, $C_{IMP_1}(s)$, lays within the ellipsoid or not, (4.15) can be used, (2.8) in section 2.2):

$$\varepsilon(t_k) = \{\theta|\gamma \leq 1\}$$

(4.14)

with

$$\gamma = (\theta - \theta_c(t_k))^T\Sigma^{-1}(t_k) \cdot (\theta - \theta_c(t_k))$$

(4.15)

For this, the controller $C_{IMP_1}(s)$, (4.5), has to be transformed to z-domain and subsequently written in the form of (2.15). However, $C_{IMP_1}(s)$ is a fifth order transfer function and can not be written into (2.15) which is a third order transfer function. So, first a third order, or lower,
controller with Internal Model Principle has to be calculated, which can be done with (3.32).
With that equation the following controller has been calculated:

\[
C_{IMP_{N,LL}}(s) = P \frac{\frac{1}{\omega_0} s^3 + \left(\frac{1}{\omega_0^2} + \frac{2}{\omega_0 f_2}\right) s^2 + \left(\frac{2}{\omega_0^3} + \frac{1}{f_1}\right) s + 1}{\frac{1}{\omega_0^2 f_2} s^3 + \frac{1}{\omega_0^2} s^2 + \frac{1}{f_2} s + 1} = \frac{0.05 s^3 + 0.15 s^2 + 0.15 s + 0.05}{0.25 s^3 + s^2 + 0.25 s + 1}
\]

(4.16)

with \(\omega_0 = 1\) rad/s, \(f_1\) and \(f_2\) respectively 1 rad/s and 4 rad/s and \(P = 0.05\). From this controller,

\[
\gamma = 2.1721 > 1
\]

(4.15)

\(C_{IMP_{N,LL}}(s)\), a Bode plot can be drawn which will be like Fig. 4.8(a). Also the open-loop control system, \(C_{IMP_{N,LL}}(s)P_I(s)\), declares about where the bandwidth has been placed, Fig. 4.8(b). In Fig. 4.8(b), the bandwidth lays at 1.49 Hz, which is lower than the bandwidth of the open-loop control system, 21.32 Hz, of \(C_{EUCI}(s)P_I(s)\), see section 4.2.2. So, the controller \(C_{IMP_{N,LL}}(s)\) without the notch, \(C_{IMP_{LL}}(s)\), lays not within the ellipsoidal area with a calculated value of 2.1721, (4.15). Since \(\gamma = 2.1721 > 1\), \(C_{IMP_{LL}}(s)\) lays not within the ellipsoidal unfalsified area. So, this lead-lag function without the inverse notch lays not inside the unfalsified area. In that case, if this lead-lag function, with the existence of an inverse notch, lays inside the unfalsified area it is because of the existence of the inverse notch in the function.

First \(C_{IMP_{N,LL}}(s)\) has to be transformed to z-domain and subsequently written in the form of (2.15):

\[
C_{IMP_{N,LL}}(z) = \frac{0.1999 z^3 - 0.5991 z^2 + 0.5985 z - 0.1993}{z^3 - 2.996 z^2 + 2.992 z - 0.996} = \frac{\theta_5 z^3 + \theta_6 z^2 + \theta_7 z + \theta_8}{\theta_1 z^3 + \theta_2 z^2 + \theta_3 z + \theta_4}
\]

(4.17)

So, \(\theta\), in (4.15), will be like:

\[
\theta = \begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4 \\
\theta_5 \\
\theta_6 \\
\theta_7 \\
\theta_8
\end{bmatrix} = \begin{bmatrix}
1.000000000000000 \\
-2.996000669370000 \\
2.99201497139196 \\
-0.996007983699993 \\
0.199900299342999 \\
-0.59910149678161 \\
0.598502694636000 \\
-0.19930149699778
\end{bmatrix}
\]

(4.18)

The program, with maximum allowed tracking error \(T_a = 0.01\), delivers the \((8 \times 8)\) matrix \(\Sigma_{t_{K1}}\) and the vector \(\theta_{e1}\), (eight components). The matrix \(\Sigma_{t_{K1}}\) defines the shape and \(\theta_{e1}\) the center of the
last determined ellipsoid. For the correct values of the components of $\Sigma_{t_k}$ and $\theta_{c_1}$, see appendix 8.1.3.

These parameters can be implemented in (4.15). Since $\gamma = 0.9182 < 1$, see (4.15), it has been proven that $C_{IMP_{N,LL}}(z)$ still lays within the unfalsified ellipsoidal area, while the lead-lag function, without the inverse notch, lays not within the unfalsified area.

So, the algorithm did not discard all parameter sets with lower bandwidths than the bandwidth of the parameter set implemented in the controller, and, the algorithm did not discard the parameter set with the Internal Model Principle, $C_{IMP_{N,LL}}(s)$. So, when running the program with a lower maximum allowed tracking error, $T$, it could still bring the result of a parameter set with Internal Model Principle contained implemented in the control-structure. However, as experienced in section 4.2.2, when decreasing the desired tracking error, just the controller-gain of the EUC controller will be raised. To avoid that raising of the controller-gain, a new plant has to be designed where raising the bandwidth in such a way is impossible, see chapter 5.1.

4.4 Beginning notch as initial conditions in the algorithm

Totally from itself, the EUC program did not need to implement a controller including the Internal Model Principle. But, maybe the algorithm will not adapt the parameters extremely when implementing the parameters of an inverse notch as initial conditions in the algorithm. For that, notch parameters have to be found for a tracking error of $T = 0.09$. When running the program with a maximum tracking error of $T_a = 0.02$, it could retain the basic notch concept and adapt the parameters to a greater peak.

First, an inverse notch with lead-lag which gives a sensitivity of 0.09 at 1 rad/s has to be formulated. $(N_{i,LL})_{IC}(s)$ is of the following form:

$$
(N_{i,LL})_{IC}(s) = \frac{0.05s^3 + 0.06432s^2 + 0.06432s + 0.05}{0.25s^4 + 1.016s^2 + 0.3137s + 1}
$$

$(N_{i,LL})_{IC}(s)$ is formulated with the use of (3.32), with $\omega_0 = 1$ rad/s, $f_1$ and $f_2$ respectively 1 rad/s and 4 rad/s and $P = 0.05$, however, $\beta_1 = 0.9$ and $\beta_2 = 0.2$ instead of 1 and 0, see section 3.4. Fig. 4.9(a) is the Bode plot of $(N_{i,LL})_{IC}(s)$. In Fig. 4.9(b) the sensitivity is drawn, $S_{IC}(s)$.

![Bode plots](image)

Figure 4.9: Bode plots of $(N_{i,LL})_{IC}(s)$ and $S_{IC}(s)$

$S_{IC}(s)$ will be like:

$$
S_{IC}(s) = \frac{1}{1 + (N_{i,LL})_{IC}(s)P_k(s)}
$$

(4.20)
In Fig. 4.9(b), a sensitivity of -21 dB is noticed at the disturbance frequency of 1 rad/s. So, according to (4.12), 
\[ |S(j\omega_0)| = |Y(j\omega_0)|, \]
with \( \omega_0 \) the disturbance frequency. \( |S(j\omega)| = 10\left(\frac{-21}{20}\right) = |Y(j\omega_0)| = 0.089 \approx 0.09. \)

To write \((N_{iLL})_{IC}(s)\) in the parameter set, \(\theta\), of (2.15), the function has to be transformed to \(z\)-domain:

\[
(N_{iLL})_{IC}(z) = \frac{0.1997z^3 - 0.5989z^2 + 0.5987z - 0.1995}{z^3 - 2.996z^2 + 2.992z - 0.9959} \tag{4.21}
\]

So, \(\theta\) will be like:

\[
\theta = \begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4 \\
\theta_5 \\
\theta_6 \\
\theta_7 \\
\theta_8 \\
\end{bmatrix} = \begin{bmatrix}
1.00000000000000 \\
-2.99594332412905 \\
2.99188790634939 \\
-0.99594457822845 \\
0.19972284642488 \\
-0.59891150837343 \\
0.59865473454561 \\
-0.19946607239746 \\
\end{bmatrix} \tag{4.22}
\]

This parameter set \(\theta\) symbolizes the initial conditions in the EUC algorithm. So, when running the program, the starting controller will be \((N_{iLL})_{IC}(z)\). With these initial conditions, \(\theta\), the program runs, with \(L_c = 0.01\), \(T_a = 0.01\) and \(G_b = 3\). However, when the tracking error reaches the threshold, \(\Delta(t)\), at \(t = 371.716\) s, value, a numerical error appears.
Chapter 5

Plant II

In this chapter, first the plant, \( P_2 \), will be introduced, section 5.1. But when running the EUC algorithm with \( P_2 \) as plant, a numerical error arises and first has to be solved, section 5.2. When solved the error, a comparison between both controllers, from EUC and IMP, can be made, section 5.3. After all it is interesting whether the controller from IMP lays inside or outside the unfalsified area, section 5.4 and in section 5.5 a beginning notch as initial conditions has been implemented in the algorithm to see whether the algorithm would finish this notch or not.

5.1 Plant, \( P_2 \)

5.1.1 General Plant

This plant is determined from a motion system, see Fig. 5.1(a). From this motion system, the following equations can be determined:

\[
F - a - b = m_1 \ddot{x}_1, \quad a + b = m_2 \ddot{x}_2 \tag{5.1}
\]

Since \( a = k(x_1 - x_2) \) and \( b = d(\dot{x}_1 - \dot{x}_2) \), (5.1) can be written as:

\[
F - k(x_1 - x_2) - d(\dot{x}_1 - \dot{x}_2) = m_1 \ddot{x}_1, \quad k(x_1 - x_2) + d(\dot{x}_1 - \dot{x}_2) = m_2 \ddot{x}_2 \tag{5.2}
\]

which gives after a combining and transforming to Laplace:

\[
F = \left( m_1 s^2 + k + ds \right) \frac{m_2 s^2 + k + ds}{k + ds} x_2 - (k + ds)x_2 \tag{5.3}
\]

Consider \( F \) as input and \( x_2 \) as output of the motion system, Fig. 5.1(a). With \( F \) and \( x_2 \) as input and output, the transfer function of the motion system can be calculated by the division \( \frac{x_2}{F} \):

\[
\frac{x_2}{F} = \frac{ds + k}{m_1 m_2 s^3 + d(m_1 + m_2)s^3 + k(m_1 + m_2)s^2} \tag{5.4}
\]

This function \( \frac{x_2}{F} \) is the general transfer function of the motion system, Fig. 5.1(a).
5.1.2 Plant parameters

In this research the following parameters has been chosen for (5.4):

\[
m_1 = 3 \text{ kg} \\
m_2 = 5 \text{ kg} \\
k = 10000 \text{ N/m} \\
d = 0.9 \text{ Ns/m}
\]

When implementing these parameters in (5.4), the following transfer function, plant \( P_{2a}(s) \) will be obtained

\[
P_{2a}(s) = \frac{0.9s + 10000}{15s^4 + 7.2s^3 + 80000s^2}
\]

(5.5)

where Fig. 5.2 is a Bode plot of this plant. In this Bode plot a peak is noticed at 72 rad/s, which is the resonant frequency of the motion system.

Also is noticed in the figure a phase of \(-270^\circ\), or \(+90^\circ\), at the frequency of \(2 \times 10^5\) rad/s and higher. As the result of this phase of \(+90^\circ\), the EUC program can not stabilize the open loop system by just increasing the bandwidth like the program did with plant \( P_1(s) \), see section 4.3. The resonant peak lays at 72 rad/s, which is a normal situation, because most motion systems has a resonant peak at such a high frequency.

However, there is another reason why the resonant peak has been chosen this frequency. The disturbance frequency of 1 rad/s lays lower than the resonant frequency. In this situation, the EUC program still can find a controller with eight parameters. When the disturbance frequency lays behind the resonant frequency, the EUC program could neutralize the influence of the resonant peak, and, if the program does, no Internal Model Principle can be applied in the controller. The problem becomes clear when looking at the Notch-equation, (3.29), in section 3.4. This equation is of a second order. So implementing this notch twice in the controller, (one for the neutralization of the resonant peak and one for the Internal Model Principle), already a fourth order equation is determined. Since there are eight parameters, which gives a fourth order controller, there will be no parameter left to stabilize the open loop system. However, \( P_{2a}(s) \) needs to be stabilized. So, a plant with a resonant peak at a lower frequency than the disturbance frequency gives not the
possibility to let EUC find a controller with Internal Model Principle.

5.2 Possible solutions to solve the numerical error

When running the EUC program, with \( T_a = 0.01 \), \( G_a = 6 \) and \( L_a = 0.1 \), see 4.6 in section 4.2.2, a numerical error arises at \( t = 20.0785 \) s. At that time, the eigenvalues of the matrix \( \Sigma_{eig} \), which defines the shape of the last determined ellipsoid, are not positive definite:

\[
\Sigma_{eig} = 1.10^8 \begin{bmatrix}
-1.5054186249544 \\
-0.0000000000000 \\
-0.0000000000000 \\
0.0000000000000 \\
0.0000000000000 \\
0.0000000000000 \\
0.01008535214963 \\
\end{bmatrix}
\]

When some of the eigenvalues are negative, it says that one of the axis of the ellipsoid is negative shaped, which is impossible and logically a numerical error arises. At the time of the numeric error, the last given control parameter set is like:

\[
\theta = \begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4 \\
\theta_5 \\
\theta_6 \\
\theta_7 \\
\theta_8 \\
\end{bmatrix} = \begin{bmatrix}
0.0000000000000 \\
-0.00000000000153 \\
-0.0000000000025 \\
-0.00000000000501 \\
1.00372719734779 \\
-0.01064707824301 \\
0.01003078325866 \\
-0.00309916359851 \\
\end{bmatrix}
\]

In the control structure, the controller will be derived by the division of the controller set by \( \theta_1 \), see appendix 8.1.1. At the time of the numerical error, \( \theta_1 \) turns zero, see (5.7). So, the controller set will be divided by zero and turns to be infinite at the time of the numerical error.

This error can not be solved by decreasing the value of \( L_a \), to let the program go less fast to the desired tracking error. Besides this problem, which has to be solved eventually, some possible solutions can be applied to the plant itself, or by a controller in front of the plant. In the next sections, sections 5.2.1 to 5.2.3, different options of solutions will be discussed.

5.2.1 A high frequent adaption of the Plant, \( P_{2b} \)

With the knowledge of the succes to determine a controller with EUC to plant \( P_1(s) \), chapter 4, \( P_{2a}(s) \) can be adapted. Because, \( P_1(s) \) has a phase of \(-90^\circ\) at the frequency of 10 Hz and higher, see Fig. 4.1, the EUC program stabilizes the open loop system by increasing the bandwidth. So, to see if the program can not deal with the high frequent phase of \(+90^\circ\), see Fig. 5.5, \( P_{2a}(s) \) has to be adapted. So, at least from \( 2.10^5 \) rad/s an extra phase of 180° has to be given to \( P_{2a}(s) \). This goal can be achieved by the multiplication of the numerator of \( P_{2a}(s) \) with \((s + 1.10^3)(s + 1.10^3)\). In that case, the magnitude of \( P_{2a}(s) \) changes from a slope of +1 to a slope of -2 at the break point of \( 1.10^5 \) rad/s in \( P_{2b}(s) \):

\[
P_{2b}(s) = \frac{(0.9s + 10000)(s + 1.10^3)(s + 1.10^3)}{15s^4 + 7.2s^3 + 80000s^2} = \frac{0.9s^3 + 11800s^2 + 20900000s + 1.10^{10}}{15s^4 + 7.2s^3 + 80000s^2} \tag{5.8}
\]

where Fig. 5.3 is a Bode plot of this plant. Now, when running the program with \( P_{2b} \), with \( T_a = 0.01 \), \( G_a = 6 \) and \( L_a = 0.1 \), still a numerical error arises at \( t = 19.9015 \) s. Also this error can not be solved by decreasing the value of \( L_a \). So, the numerical error is still present with \( P_{2b} \) as plant and therefore another adaption to \( P_{2a} \) has to be applied to solve the error.
5.2.2 A low frequent adaption of the Plant, $P_{2c}$

Notable in the Bode plot of $P_{2a}(s)$, Fig. 5.2, is the slope of -2 at 10 rad/s and lower frequencies. This implies that, $P_{2a}(s)$ has a break point at 0 rad/s since the slope of -2 is maintained up to 0 rad/s. So, poles of $P_{2a}$ lays at $s = 0$, which implies instability of the plant itself. This says that, in the open loop model, a step function $u(t)$ as input, gives an infinite output $y$ in steady state situation. To determine all poles, the denominator of $P_{2a}$ will be equated to zero:

$$15s^4 + 7.2s^3 + 80000s^2 = 0$$ (5.9)

which will be:

$$s^2(15s^2 + 7.2s + 80000) = 0$$ (5.10)

So, the poles of $P_{2a}$ lays at:

- $s = 0$
- $s = 0$
- $s = -0.2400 + 73.0293j$
- $s = -0.2400 - 73.0293j$

So, these poles of $s = 0$ has to be replaced to $s < 0$, which stabilizes the plant. For that, an extra break point at a frequency higher than 0 rad/s has to be implemented. Consider a break point at 0.2 rad/s. The denominator of $P_{2a}(s)$ then will be adapted to $(s + 0.2)^2(15s^2 + 7.2s + 80000)$. $P_{2c}(s)$ is:

$$P_{2c}(s) = \frac{0.9s + 10000}{15s^4 + 13.2s^3 + 80003s^2 + 32000s + 32000}$$ (5.11)

With this adaption, $P_{2c}$ is still a fourth order transfer function, like $P_{2a}$ is. The poles of $P_{2c}$ lays at:

- $s = -0.2$
- $s = -0.2$
- $s = -0.2400 + 73.0293j$
- $s = -0.2400 - 73.0293j$

Figure 5.3: Bode plot of the transfer function of the plant $P_{2b}(s)$, (5.8)
The Bode plot of $P_{2c}(s)$, Fig. 5.4(b), shows that, low frequent (frequencies of 0.2 rad/s and lower), the magnitude of $P_{2a}(s)$ is adapted from a slope of -2 to 0, and the phase from $-180^\circ$ to $0^\circ$. When running the program with $P_{2c}$, with $T_a = 0.01$, $G_a = 6$ and $L_a = 0.1$, still a numerical error arises at $t = 20.0845$ s. Also this error can not be solved by decreasing the value of $L_a$. So, the numerical error is still present with $P_{2c}$ as plant and therefore another adaption has to be applied to solve the error.

5.2.3 Lead-lag controller in front of the Plant

Another option is to stabilize the open loop system, so the EUC program is not afford to do. Therefore, a controller has to be added in front of $P_{2a}(s)$, and, with loopshaping the right controller can be selected. The following controller, $C_{LL}(s)$ has been selected, which is of a lead-lag form:

$$C_{LL}(s) = \frac{20s + 0.02}{s + 1}$$

(5.12)

with Fig. 5.5(a) the Bode plot of the transfer function. In open loop, $P_{2a}(s)$ multiplied with $C_{LL}(s)$, the Bode plot will be like Fig. 5.5(b). This figure shows a bandwidth of 1.43 rad/s. At this frequency a phase of $-145^\circ$ is noticed which denotes a phase margin of $35^\circ$ and is enough for a stable system. In this open loop system, the resonant peak lays beneath the 0 dB, so, this peak will not be boosted. So, the controller, $C_{LL}(s)$ stabilizes the open loop system and can be implemented in front of the plant in the EUC, Simulink program.

From $C_{LL}(s)$ and $P_{2a}(s)$ the sensitivity, $S_{2a}(s)$, of the control structure can be derived:

$$S_{2a}(s) = \frac{1}{1 + C_{LL}(s)P_{2a}(s)}$$

(5.13)

with Fig. 5.6 the Bode plot of this sensitivity transfer function. In this figure, at the frequency of 1 rad/s, a magnitude of -1.8 dB is noticed, which is a magnitude of $10^{-1.8} = 0.8$. So, when running the program, the disturbance, $w(t)$, is already reduced with a factor 0.8. Nevertheless, this reduction will not be enough to satisfy the given performance specification, so, the EUC algorithm still has to find a suitable controller.

When running the program with $C_{LL}(s)$ in front of $P_{2a}(s)$, with $T_a = 0.01$, $G_a = 6$ and $L_a = 0.1$, still a numerical error arises at $t = 20.1635$ s. However, when decreasing the value of $L_a$, this error can be solved. So, the controller $C_{LL}(s)$ in front of $P_{2a}(s)$ solves the numerical error.
5.3 Comparison of controller from IMP with EUC

5.3.1 Controller from IMP

Nothing changed in control structure of Fig. 4.2. The disturbance $w(t)$ is again a sinusoid with a frequency of $\omega_0 = 1$ rad/s. To find a controller including the Internal Model Principle, (3.28) can be used with state feedback determination. For that, the closed loop poles have been selected like

$$pc = \begin{bmatrix} -2 + 4j \\ -2 - 4j \\ -3 + 3j \\ -3 - 3j \\ -5 + 5j \\ -5 - 5j \end{bmatrix}$$

(5.14)

The selection of these poles are based on the location and values of the poles. All the poles are located in the left half-plane of the s-plane and conjugated to each other. Besides that, the values
are chosen in the same range of 0 - 10. When follow the same procedure as done in section 4.2.1, the transfer function of the controller with Internal Model Principle, \( C_{IMP_2}(s) \), with plant \( P_2(s) \) can be determined:

\[
C_{IMP_2}(s) = \frac{17.86s^5 + 27.82s^4 + 95260s^3 + 10270s^2}{s^6 + 20s^5 + 212s^4 + 1312s^3 + 5379s^2 + 1292s + 5168}
\]  
(5.15)

with Fig. 5.7 the Bode plot of \( C_{IMP_2}(s) \).

5.3.2 Controller from EUC

When running the program with, with a lead-lag controller, (5.12), in front of the plant, \( P_2(s) \), as determined in section 5.2.3, the following parameters in the threshold value, (4.6), has been chosen: \( T_a = 0.01 \), \( G_a = 6 \) and \( L_c = 0.01 \). With these parameters of the threshold value and the lead-lag controller in front of the plant, the EUC program runs without numerical error. When running the EUC program, (until \( t = 250 \) s), the threshold value, \( \Delta(t) \), reaches the tracking error at time \( t = 182 \) s. At that moment, the program implements a new parameter set, \( \theta^* \), into the controller which eventually gives a tracking error of 0.008, Fig. 5.8(a) and 5.8(b), without adapting the parameter set again. This, while the maximum allowed tracking error did not even reached a value smaller than 0.01. This parameter set, \( \theta^* \), will not be substituted into another parameter set, because \( T_a = 0.01 \) which is bigger than 0.008. So, \( \theta^* \) implemented at time \( t = 182 \) s will be \( \theta_{end} \):

\[
\theta_{end} = \begin{bmatrix} 0.1593 & -0.2380 & 0.0397 & 0.0531 & 0.3396 & 0.2776 & 0.2156 & 0.1537 \end{bmatrix}
\]  
(5.16)

which can be implemented in (2.15) and gives the transfer function, \( C_{EUC_2a}(z) \):

\[
C_{EUC_2a}(z) = \frac{0.3396z^3 + 0.2776z^2 + 0.2156z + 0.1537}{0.1593z^3 - 0.2380z^2 + 0.0397z + 0.0531}
\]  
(5.17)

with Fig. 5.9(a) the Bode plot of \( C_{EUC_2a}(z) \). With the transfer functions, \( P_2(s) \), \( C_{EUC_2a}(s) \) and the lead-lag function \( C_{LL}(s) \), (5.12), the open loop control system can be derived, see Fig. 5.9(b). From this figure the bandwidth can be noticed which lays at 13.49 rad/s. At this frequency, the phase is \(-176^\circ\). This denotes a phase margin of 4\(^\circ\), which is enough for a stable system. However, no high peak at the disturbance frequency of 1 rad/s is noticed in the Bode plot of \( C_{EUC_2a}(z) \), so the tracking error of 0.008 has not been achieved by adding an inverse notch at the disturbance frequency and neither by implementing the internal model in the controller. Why
In section 5.3.2, it is shown that the EUC algorithm did not need to find a controller with Internal Model Principle or an inverse notch at the disturbance frequency of 1 rad/s. As done in section 5.4 Controller from IMP inside or outside the unfalsified area.

In section 5.3.2, it is shown that the EUC algorithm did not need to find a controller with Internal Model Principle or an inverse notch at the disturbance frequency of 1 rad/s. As done in section 5.4 Controller from IMP inside or outside the unfalsified area.

\[ S_{EUC_{2a}}(s) = \frac{1}{1 + C_{EUC_{2a}}(s)C_{LL}(s)P_{2a}(s)} \]  
(5.18)

with Fig. 5.10 the Bode plot of it. In this figure, a sensitivity of -42 dB is noticed at the disturbance frequency of 1 rad/s. So, according to (4.12), \(|S(j\omega_0)| = |Y(j\omega_0)|\), with \(\omega_0\) the disturbance frequency. \(|S(j\omega_0)| = 10^{\frac{-42}{20}} = |Y(j\omega_0)| = 0.0079 < 0.008\). So, indeed the controller \(C_{EUC_{2a}}(s)\) in combination with \(C_{LL}(s)\) reduces the error to 0.008, however, not with an inverse notch and neither by implementing the Internal Model Principle.
5.3.2, a controller has been found with a lead-lag controller already in front of the plant. As in section 4.3, it is important to know if the controller found from IMP still lays within the unfalsified area determined by the EUC algorithm. For that, the threshold value really has to reach the maximum allowed tracking error, \( T_a = 0.01 \), so, the EUC program has to run much longer. The controller with Internal Model Principle can be calculated with (3.32). However, as explained in section 5.2.3, a lead-lag controller has been implemented in front of the plant. So, in the situation with the lead-lag controller already in front of the plant, only the inverse notch controller is relevant to locate, without of within the ellipsoidal area. So, the formula for the inverse notch is like (3.29):

\[
N_i(s) = \frac{s^2 + 0.3183s + 1}{s^2 + 1}
\]

with \( f_a = f_b = \omega_0 = 1 \) rad/s, \( \beta_1 = 1 \), \( \beta_2 = 0 \)

This gives:

\[
N_i(s) = \frac{z^2 - 2z + 0.9998}{z^2 - 2z + 1} = \frac{z^3 - 2z^2 + 0.9998z}{z^3 - 2z^2 + z} = \frac{\theta_5z^3 + \theta_6z^2 + \theta_7z + \theta_8}{\theta_1z^3 + \theta_2z^2 + \theta_3z + \theta_4}
\]

So, \( \theta \), in (4.15), will be like:

\[
\theta = \begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4 \\
\theta_5 \\
\theta_6 \\
\theta_7 \\
\theta_8
\end{bmatrix} = \begin{bmatrix}
1.00000000000000 \\
-1.999999000000025 \\
1.00000000000000 \\
0.00000000000000 \\
1.00015915490330 \\
-1.999999000000025 \\
0.99984084509670 \\
0.00000000000000
\end{bmatrix}
\]

The program, with maximum allowed tracking error \( T_c = 0.0001 \), \( L_c = 0.01 \) and \( G_c = 6 \), see section ......., with a running time of 1200 s delivers a matrix \( \Sigma_{tk2} \) and the vector \( \theta_{c2} \), see appendix.
8.1.4. These parameters can be implemented in (4.15). When done, the result is: 0.7687. Since 0.769 < 1, see (4.15), it has been proven that \( N(z) \) still lays within the unfalsified ellipsoidal area. So, the algorithm did not discard all parameter sets including the Internal Model Principle. So, there is still the possibility for the EUC algorithm to implement a control parameter set with IMP when, for example, letting the desired tracking error be zero. However, when implementing a desired tracking error of zero into the EUC algorithm, again the numerical error arises.

5.5 Beginning notch as initial conditions in the algorithm

Just like wit plant \( P_1 \), the EUC program did not need to implement a controller including the Internal Model Principle. But, just like researched in section 4.4, maybe a beginning inverse notch delivers after adaption a controller with Internal Model Principle included. Fore that, both the inverse notch multiplied by the lead-lag function has to be implemented as initial conditions in the algorithm. The selected inverse notch is as follows:

\[
(N_{LL})_{IC}(z) = \frac{19.99z^3 - 59.97z^2 + 59.97z - 19.99}{z^3 - 2.999z^2 + 2.998z - 0.999} \tag{5.24}
\]

which gives the Bode plot, Fig. 5.11(a). \( S_{IC}(s) \) will be like:

![Bode plot of \( N_{LL} \)_{IC}(z)](image)

![Bode plot of \( S_{IC}(s) \)](image)

Figure 5.11: Bode plots of \( N_{LL} \)_{IC}(z) and \( S_{IC}(s) \)

\[
S_{IC}(s) = \frac{1}{1 + (N_{LL})_{IC}(s)P_1(s)} \tag{5.25}
\]

In Fig. 5.11(b), a sensitivity of -25 dB is noticed at the disturbance frequency of 1 rad/s. So, according to (4.12), \( |S(j\omega_0)| = |Y(j\omega_0)| \), with \( \omega_0 \) the disturbance frequency. \( |S(j\omega_0)| = 10^{(-25/20)} = |Y(j\omega_0)| = 0.056 \approx 0.06 \). So, \( \theta \) will be like:

\[
\theta = \begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4 \\
\theta_5 \\
\theta_6 \\
\theta_7 \\
\theta_8
\end{bmatrix} = \begin{bmatrix}
1.00000000000000 \\
-2.99896766929223 \\
2.99793637138234 \\
-0.09896870109063 \\
19.99287830494557 \\
-59.9723201573917 \\
59.96584911271638 \\
-19.98649540190279
\end{bmatrix} \tag{5.26}
\]
This parameter set $\theta$ symbolizes the initial conditions in the EUC algorithm. So, when running the program, the starting controller will be $(N_i LL)_{IC}(z)$. With these initial conditions, $\theta$, the program runs, with $L_c = 0.01$, $T_a = 0.01$ and $G_a = 3$. However, when the tracking error reaches the threshold, $\Delta(t)$, at $t = 413.93$ s, value, a numerical error appears.
Chapter 6

Conclusion

It is investigated to what extent the Internal Model Principle is present in controllers from Ellipsoidal Unfalsified Control. Two Plants has been researched, $P_1$ and $P_2$. With Plant $P_1$, the EUC algorithm did not need to implement a controller with Internal Model Principle or inverse notch into the control structure. Because of the possibility with Plant $P_1$ to increase the bandwidth infinitely, the EUC algorithm just raised the controller-gain until it reached the desired tracking error. For this reason Plant $P_2$ had been introduced. Also with Plant $P_2$ the EUC algorithm did not need to implement a controller with Internal Model Principle or inverse notch into the control structure.

However, for both Plants, the control parameter sets with Internal Model Principle still lays in the unfalsified area of the EUC algorithm. In that case, only for Plant $P_2$, there might be the possibility to let the EUC algorithm implement a controller with Internal Model Principle into the control structure, which could be reached by choosing a desired tracking error of zero. However, in that situation, a numerical error arises. Also adaptation of the initial parameter sets of the EUC algorithm to a starting controller with inverse notch did not worked out. Instead of adapting the starting controller to a greater notch, a numerical error arises.

For further research of this subject, first the numerical error has to be solved by adaptations of the EUC algorithm. When this problem has been solved, the desired tracking error can be chosen smaller, or can be chosen zero for these Plants and, for many Plants, the implementation of Internal Model Principle will be inevitable.
Chapter 7

References

Van Helvoort, J., De Jager, B. & Steinbuch, M. (2005), Unfalsified Control using an ellipsoidal unfalsified region applied to a motion system, in Proc. IFAC World Congress, Prague, Czech Republic.


Chapter 8

Appendix

8.1 Calculations

8.1.1 Model formulation of controller

\[ R_v(t, \theta) = w(u(t), y(t))^T \theta \]  \hspace{1cm} (8.1)

and in the model:

\[ U(z) = \frac{1}{\theta_1} R \theta - \theta_2 \frac{1}{z} U(z) - \theta_3 \frac{1}{z^2} U(z) - \theta_4 \frac{1}{z^3} U(z) - \theta_5 Y(z) - \theta_6 \frac{1}{z} Y(z) - \theta_7 \frac{1}{z^2} Y(z) - \theta_8 \frac{1}{z^3} Y(z) \]  \hspace{1cm} (8.2)

and, when \( R \) chosen zero:

\[ \frac{Y(z)}{U(z)} = C_{EVC}(z) = \frac{\theta_5 z^3 + \theta_6 z^2 + \theta_7 z + \theta_8}{\theta_1 z^3 + \theta_2 z^2 + \theta_3 z + \theta_4} \]  \hspace{1cm} (8.3)

8.1.2 From State Space to transfer function

\( P_1 \) written in state space:

\[ \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} \]  \hspace{1cm} (8.4)

and

\[ \begin{bmatrix} 50 & 10 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = y \]  \hspace{1cm} (8.5)

So,

\[ X_1(s) = \frac{1}{s + 1} U(s), \quad X_2(s) = \frac{1}{s^2 + s} U(s), \quad X_3(s) = \frac{1}{s^3 + s^2} U(s) \]  \hspace{1cm} (8.6)

and

\[ Y(s) = 50X_1(s) + 10X_2(s) + X_3(s) \]  \hspace{1cm} (8.7)

So,

\[ Y(s) = 50(\frac{1}{1 + s}) U(s) + 10(\frac{1}{s^2 + s}) U(s) + (\frac{1}{s^3 + s^2}) U(s) \]  \hspace{1cm} (8.8)
So,

\[
\frac{50s^2 + 10s + 1}{s^3 + s^2} U(s) = Y(s)
\]  \hspace{2cm} (8.9)

So, \( P_1(s) \) in transfer function form can be calculated by \( Y(s)/U(s) \):

\[
\frac{Y(s)}{U(s)} = P_1(s) = \frac{100s^2 + 20s + 2}{2s^3 + 2s^2}
\]  \hspace{2cm} (8.10)

\( P_2 \) written in state space:

\[
\begin{bmatrix}
-0.5 & -5333.3 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ x_4
\end{bmatrix}
+ \begin{bmatrix}
1 \\ 0 \\ 0 \\ 0
\end{bmatrix} u =
\begin{bmatrix}
\dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4}
\end{bmatrix}
\]  \hspace{2cm} (8.11)

and

\[
\begin{bmatrix}
0 & 0 & 0.06 & 666.67
\end{bmatrix}
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ x_4
\end{bmatrix} = y
\]  \hspace{2cm} (8.12)

So,

\[
X_3(s) = \frac{1}{s^3 + 0.5s^2 + 5333.3s} U(s), \quad X_4(s) = \frac{1}{s^4 + 0.5s^3 + 5333.3s^2} U(s)
\]  \hspace{2cm} (8.13)

and

\[
Y(s) = 0.06X_3(s) + 666.67X_4(s)
\]  \hspace{2cm} (8.14)

So,

\[
Y(s) = 0.06\frac{1}{s^3 + 0.5s^2 + 5333.3s} U(s) + 666.67\frac{1}{s^4 + 0.5s^3 + 5333.3s^2} U(s)
\]  \hspace{2cm} (8.15)

So,

\[
\frac{0.06s + 666.67}{s^4 + 0.5s^3 + 5333.3s^2} U(s) = Y(s)
\]  \hspace{2cm} (8.16)

So, \( P_2(s) \) in transfer function form can be calculated by \( Y(s)/U(s) \):

\[
\frac{Y(s)}{U(s)} = P_2(s) = \frac{0.9s + 10000}{15s^4 + 7.2s^3 + 80000s^2}
\]  \hspace{2cm} (8.17)
8.1.3 \( \Sigma_{tk1} \text{ and } \theta_{c1} \)

\[
\Sigma_{tk1} = 1.10^5
\begin{bmatrix}
1.1871 & -1.3066 & -0.3296 & 0.4493 & 0.1360 & 0.0770 & -0.0309 & -0.1806 \\
-1.3066 & 2.5144 & -0.7556 & -0.4519 & -0.0299 & -0.0011 & 0.0149 & 0.0168 \\
-0.3296 & -0.7556 & 2.4508 & -1.3656 & -0.0884 & -0.0414 & 0.0245 & 0.1051 \\
0.4493 & -0.4519 & -1.3656 & 1.3677 & -0.0175 & -0.0345 & -0.0086 & 0.0586 \\
0.1360 & -0.0299 & -0.0884 & -0.0175 & 2.4083 & -0.9832 & -0.8025 & -0.6210 \\
0.0770 & -0.0011 & -0.0414 & -0.0345 & -0.9832 & 2.6418 & -0.8610 & -0.7971 \\
-0.0309 & 0.0149 & 0.0245 & -0.0086 & -0.8025 & -0.8610 & 2.6443 & -0.9813 \\
-0.1806 & 0.0168 & 0.1051 & 0.0586 & -0.6210 & -0.7971 & -0.9813 & 2.3978
\end{bmatrix}
\]

\( \theta_{c1} = [0.2948 \ -0.3581 \ -0.0676 \ 0.1309 \ 0.3627 \ 0.2973 \ 0.2164 \ 0.1235] \) (8.19)

8.1.4 \( \Sigma_{tk2} \text{ and } \theta_{c2} \)

\[
\Sigma_{tk2} = 1.10^7
\begin{bmatrix}
0.0060 & -0.0174 & 0.0169 & -0.0055 & 0.0066 & -0.0044 & -0.0056 & 0.0035 \\
-0.0174 & 0.0506 & -0.0492 & 0.0160 & -0.0177 & 0.0115 & 0.0149 & -0.0088 \\
0.0169 & -0.0492 & 0.0481 & -0.0157 & 0.0159 & -0.0100 & -0.0133 & 0.0075 \\
0.0055 & 0.0160 & -0.0157 & 0.0052 & -0.0049 & 0.0029 & 0.0040 & -0.0021 \\
0.0066 & -0.0177 & 0.0159 & -0.0049 & 1.8890 & -2.5461 & -0.5646 & 1.2222 \\
-0.0044 & 0.0115 & -0.0100 & 0.0029 & -2.5461 & 4.5289 & -1.4217 & -0.5617 \\
-0.0056 & 0.0149 & -0.0133 & 0.0040 & -0.5646 & -1.4217 & 4.5312 & -2.5455 \\
0.0035 & -0.0088 & 0.0075 & -0.0021 & 1.2222 & -0.5617 & -2.5455 & 1.8854
\end{bmatrix}
\]

\( \theta_{c2} = [-0.0001 \ \ 0.0000 \ \ 0.0001 \ \ -0.0001 \ \ 0.7153 \ \ 0.3844 \ \ 0.0845 \ \ -0.1843] \) (8.21)