Optimal Tuning of PWPF Modulator for Attitude Control

Master Thesis
Spring 2005

by

Trond Dagfinn Krøvel

Department of Engineering Cybernetics
Norwegian University of Science and Technology

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk
Institutt for teknisk kybernetikk

MASTEROPPGAVE

Kandidatens navn: Trond Dagfinn Krøvel

Fag: Teknisk Kybernetikk

Oppgavens tittel (norsk): Optimal tuning av PWPF-modulator for styring av orientering av romskip.

Oppgavens tittel (engelsk): *Optimal tuning of PWPF-modulator for attitude control of spacecraft.*

Oppgavens tekst:

ESMO er en mikrosatelitt under ESAs SSETI-prosjekt. Satelitten er tenkt å gå inn i bane rundt månen. Orienteringen til ESMO styres av thrustere som er av/på av natur, men pådrag gjerne designes som kontinuerlige signaler.
Utsetting av pådrag på thrustere, både for ulineære og lineære regulatorer gjøres derfor ved hjemp av en eller annen form for modulator. Puls-bredde-puls-frekvens modulatoren (PWPF) har vist seg å være godt egnet for dette. Modulatoren inneholder fem uavhengige parametere som påvirker ytelsen. Det er et problem å finne gode verdier for disse parameterene. Følgende skal utføres:

1. Presenter metoder for å sette ut et kontinuerlig pådrag på av/på-thrustere.
2. Gi en beskrivelse av hva slags effekt parameterene i PWPF algoritmen har på ytelsen. Utfør en statisk og dynamisk analyse av modulatoren.
3. I systemet besående av attitude-dynamikken til ESMO, regulator og PWPF-modulator skal optimale parametere for modulatoren finnes. Det skal optimeres med hensyn på drivstofforbruk.

Resultatene skal illustreres med simuleringer utført i Simulink.

Oppgaven gitt: 12/1-05

Besvarelsen leveres: 8/7-05

Besvarelsen levert:

Utført ved Institutt for teknisk kybernetikk

Trondheim, den

Jan Tommy Gravdahl
Faglærer

Preface

This work is submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering Cybernetics, from the Norwegian University of Science and Technology, Department of Engineering Cybernetics.

I would like to thank my supervisor, Associate Professor Jan Tommy Gravdahl, for the help and support he has given me the last year. He of all knows that I have had enough to fill my days with the last year, and the goodwill he has shown me has been invaluable.

I would also like to thank my fellow students at G238 for help, support, many good discussions and a lots of fun throughout the last year!

Trond Dagfinn Krøvel
Trondheim, July 8, 2005

Abstract

After a spacecraft is launched, the main factor deciding the lifetime of the spacecraft is fuel. If the spacecraft can reduce its fuel consumption, the lifetime can be increased or the launch weight can be similarly reduced. Both options have economical advantages for the operator. This calls for fuel optimal attitude control solutions, and in this work the focus is on the unit translating the continuous desired torque signal to a on/off signal to the spacecraft thrusters.

A presentation of different ways to translate a continuous commanded torque signal to an on/off signal is done. Out of the presented translators a PWPF modulator is chosen, and a thorough analysis of its characteristics and behavior is performed.

Varying ranges of PWPF modulator parameters are then analysed through system simulations, and a general conclusion is drawn based upon the simulation results. The conclusion apply for all systems using PWPF modulators, but will have to be specially tailored for each specific PWPF system, based on controller choices and system dynamics. All simulations are thoroughly documented, and are easy to recreate.

The results from the general PWPF analysis are then applied to a model of ESMO. First a range of close to optimal parameters are found for three different controllers. The parameters are primarily tuned for lowest possible fuel consumption, but the number of thruster firings, which also is important for the lifetime of a spacecraft, are also taken into consideration.

Then optimal PWPF parameters for ESMO are found through simulations. The parameters that give an accuracy equal to the desired accuracy, with the lowest fuel usage and the minimum number of thruster firings are chosen.

Keywords: Pulse-Width Pulse-Frequency Modulation; PWPF; PWPF Parameter Optimisation; Fuel Optimal PWPF Parameters; ESMO;

Table of Contents

List of Figures

Chapter 1

Introduction

Even though it is almost half a century since the human space age started, it is still very expensive to design, build and launch a spacecraft. Because of the relative high cost to enter space, it is imperative for any company, organisation or agency to have the highest possible life expectancy for their spacecrafts. This is partially achieved through using high quality components, but when the spacecraft has entered space, the amount of fuel carried along is the major limiting factor for the life expectancy, since a spacecraft without fuel normally will have problems meeting its mission requirements. Therefore, in order to expand the life span of spacecrafts, optimal methods of control to minimise fuel consumption are needed.

1.1 Purpose

The purpose of this thesis is to find an optimal region of operation, with respect to fuel consumption, for the attitude control system on the micro spacecraft European Student Moon Orbiter (ESMO), using a pulse-width pulse-frequency (PWPF) modulator for thruster control.

1.2 Background

The thesis is written in collaboration with the European Space Agency (ESA) and the Student Space Exploration and Technology Initiative (SSETI). It is a continuation of a work done by the author during autumn 2004 (Krøvel 2004). SSETI is responsible for the design and construction of ESMO, in collaboration with ESA. In the following is a short introduction to SSETI and its projects.

1.2.1 ESA

The European Space Agency is an European organisation, which cooperates closely with most of the European countries in the fields of space research, -technology and -industry. Its mission is to shape the development of Europe's space capability and ensure that investment in space continues to deliver benefits to the citizens of Europe. ESA has 16 Member States. By coordinating the financial and intellectual resources of its members, it can undertake programmes and activities far beyond the scope of any single European country (ESA 2005).

ESA Education Department

The Education department of ESA was created in 2000, to stimulate the interest for space related issues among children, youth, students and young professionals. They have several projects going on to meet their goal, and SSETI is one of them. Other projects are Young Engineers' Satellite 2 (YES2), parabolic flight campaigns and student sponsoring for major space related events.

1.2.2 SSETI

SSETI is a organization founded after an initiative from ESA's Office for Educational Projects Outreach Activities in April 2000. The SSETI organization allows groups of students to participate at their satellite projects to gain hands on knowledge in building and managing satellite projects.

SSETI currently have three satellite projects going on, namely the SSETI Express, European Student Earth Orbiter (ESEO) and ESMO. The European Student Moon Rover (ESMR) which is the last satellite in the SSETI moon programme, does not yet have a management or student participation since it relies heavily on the success of it's predecessors.

SSETI Express

SSETI Express is the first satellite to be launched by SSETI. This is a micro-satellite weighing approximately 70 kg. It will be launched the 25th of August 2005 with the Cosmos-3M rocket from the Plesetsk Cosmodrome. The satellite will enter a low, sun synchronous earth orbit with an altitude of approximately 700 km to perform its mission.

The goals of the satellite as stated by the SSETI Association are the following: "The SSETI Express mission is an educational mission that shall deploy CUBESAT pico-satellites developed by universities, take pictures of Earth, act as a test-bed and technology demonstration for hardware of the complementary project: the European Student Earth Orbiter, and function as a radio transponder for the rest of the mission duration." That is, the satellite is supposed to be an experimental satellite to see if the developed hardware is capable of withstanding a space flight and works as it should, to see if the developed software works as it should, take a picture of the earth and possibly the moon, and to launch three pico-satellites.

The three pico-satellites are XI-V from University of Tokyo, Japan, UWE-1 from University of Würzburg, Germany and Ncube-2 which is a cooperation between Narvik University College (NUC), Norwegian University of Science and Technology (NTNU), the University of Oslo (UiO), Norwegian University of Life Sciences (UMB) and the Norwegian Space Center, all located in Norway.

Express is now finished, and will be transported from it current location at European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands to the launch site at Plesetsk, Russia in July 2005.

ESEO

ESEO is scheduled for launch late 2007. It will be based on the SSETI Express, but weigh 120 kg and contain more sophisticated hardware and software, based on the experiences made from the SSETI Express. It will be launched by an Ariane 5 launcher from Kourou, French Guiana, and will enter directly into an geostationary transfer orbit (GTO). Here it will conduct its scheduled experiments for a period of two months. The successful deployment and operation of ESEO and/or SSETI Express is imperative for the launch of the next micro satellite in the programme, ESMO.

ESMO

ESMO will be the first student spacecraft to orbit the moon, and is to be launched in 2009. The payload will consist of several surface mapping systems and possibly a pico-spacecraft. The surface mapping will be done with a high resolution, narrow angle camera and a syntectic aperture radar (SAR). These will complement each other in the creation of a high resolution moon surface map. The camera can also be used to identify objects on the surface of the moon. In light of the recent attempts to question the validity of the American moon landing claims, it would be interesting to present images of the remains of Tranquility Base on the 40th anniversary for the first human landing on the moon. But, regardless of the payload, the ESMO project is without doubt the most ambitions student spacecraft project in the world - it competes in complexity with many other comercial/professional projects.

This thesis is a contribution to the work on the Attitude Determination and Control System (ADCS) subsystem on ESMO. The subsystem is under the responsibility of a joint team of students from NTNU and NUC.

ESMR

ESMR will be the final spacecraft in the SSETI Moon programme. Since this project depends so much upon the success of the previous satellites in the programme, this project has no official status in the SSETI Association yet. But a ESMR-team has already started the design of Moon Rover prototypes. The Moon Rover is to land at the surface of the moon and perform scientific experiments there. The orbiting spacecraft will be used as a relay station to send the scientific data back to earth for processing.

Future SSETI Missions

SSETI will not seize to exist after the Moon Rover project. There are already plans to expand the SSETI Association into a support organisation for universities around Europe that want to start educational satellite projects. These projects can be everything from pico-satellites to micro-satellites, and SSETI will provide advice and help based on the experience gained in previous satellite missions. It is also possible that SSETI will continue the student exploration of the solar system - then Mars could be a possible target for a future SSETI spacecraft.

1.3 Presentations

The main results of (Krøvel 2004), which is a preliminary work to this thesis, was presented at the Space Technology and Education Conference (STEC) 2005 in Aalborg, Denmark. The Presentation held there, is included in Appendix E, together with the abstract submitted prior to the conference.

1.4 Outline of the Master-Thesis

The structure of the thesis:

Chapter 2 Basic thruster theory
Several methods to translate a continuous desired torque signal to an on/off signal is presented.

Chapter 3 General PWPF theory
The mathematics behind the PWPF modulator is presented.

Chapter 4 General optimisation of PWPF modulators
The PWPF modulator is analysed for different types of input signals, and conclusions on parameter regions that give low fuel usage is drawn.

Chapter 5 Optimal PWPF-parameters for ESMO
The parameter regions found in Chapter 4 are tested on the model of ESMO, to find the region where the lowest fuel consumption occurs.

Chapter 6 Results
The results of the simulations are presented

Chapter 7 Conclusion and recommendations for future work.

Chapter 2

Basic Thruster Control System Theory

On a modern spacecraft, the ADCS is one of the most important house-holding subsystems. This system senses the attitude of the spacecraft and issues control commands to the attitude control actuators to maintain the desired orientation, so that the payloads can perform their duties. On most spacecrafts a type of attitude actuators called thrusters are used. These thrusters produce a torque in the desired direction upon the reception of a command signal from the ADCS. They do so by ejecting mass at high speed through a nozzle. Such thrusters are on/off by nature, but the ADCS produces a continuous desired torque signal. Therefore it is necessary to translate the continuous desired torque signal from the controller to an on/off signal that can control the thruster system. A simple model of a spacecraft with a thruster control system is shown in Figure 2.1.

In this chapter an outline of commonly used thruster control systems is given.

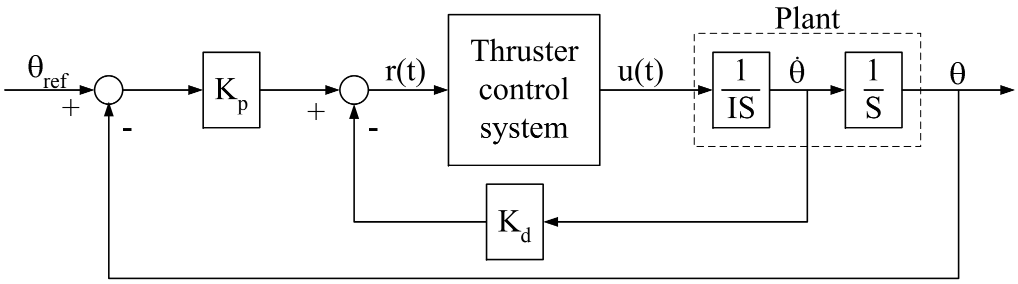


Figure 2.1: A simple spacecraft model

2.1 Bang-Bang Control Systems

Three main types of bang-bang controllers exist: Simple bang-bang controllers, bang-bang controllers with deadzone and time-optimal bang-bang controllers.

2.1.1 Bang-Bang Controller

This is the simplest of all bang-bang control systems. It consist only of a on/off switch and can be expressed mathematically like this

$$u(t) = \begin{cases} Usgn(r(t)) & \text{if} \quad |r(t)| > 0 \\ 0 & \text{if} \quad r(t) = 0 \end{cases} \tag{2.1}$$

U is the magnitude of the thruster on-signal, $u(t)$ is the bang-bang controller output signal and $r(t)$ is the reference signal. As one can see from equation (2.1), this controller will have a zero output only if the ADCS system has a zero output. Else it will fire the appropriate thruster. This makes the system very vulnerable to noise. Such systems are not employed in practice, but shows an important principle about on/off thruster control. A simple graphical representation of the controller is found in Figure 2.2.

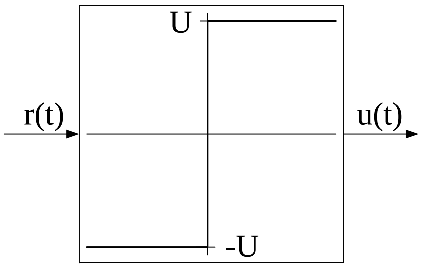


Figure 2.2: Bang-bang controller

2.1.2 Bang-Bang Controller with Deadzone

To avoid unnecessary thruster chattering due to noise, vibrations, disturbances, etc, a deadzone is implemented in the bang-bang controller. Now the controller avoids giving commands before the input reaches a certain threshold.

The mathematical expression for the controller with a deadzone is

$$u(t) = \begin{cases} Usgn(r(t)) & \text{if} \quad |r(t)| \geq \alpha \\ 0 & \text{if} \quad |r(t)| < \alpha \end{cases} \tag{2.2}$$

where α is half the deadzone of the controller. Figure 2.3 show the basic layout for a deadzone controller.

The reduced thruster activity obviously give a reduced fuel consumption compared to the simple bang-bang controller. The pointing accuracy depends directly upon the size of the deadzone, where a larger deadzone implies less accuracy. On the other hand, the system will become more robust to noise with a larger deadzone. It is obviously important to choose the size of the deadzone with great care, to maximise both accuracy and robustness towards noise and disturbance.

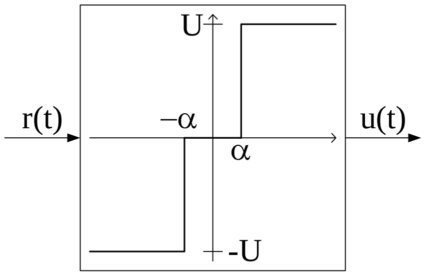


Figure 2.3: Bang-bang controller with deadzone

2.1.3 Time-Optimal Bang-Bang Control

To minimise the time a spacecraft uses on reorienting itself, time-optimal controllers are used. A time-optimal bang-bang controller is optimal for a one-axis reorientation (Elgerd 1967). The mathematical expression for a time-optimal controller is

$$u(t) = -Usgn\left(\theta - \theta_{ref} + \frac{I\dot{\theta}|\dot{\theta}|}{2U}\right) \tag{2.3}$$

Time optimal behavior obviously implies a very bad fuel economy.

2.1.4 Schmitt Trigger

The Schmitt trigger is a deadzone/hysteresis device, which is very similar to the bang-bang with deadzone controller. The only difference lies in the hysteresis functionality of the Schmitt trigger. Instead of a single on/off threshold, the trigger has one on- and one off value. Figure 2.4 shows the basic layout of a Schmitt trigger.

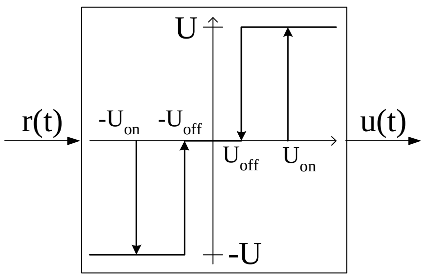


Figure 2.4: Schmitt trigger

2.2 Pulse Modulators

A range of different pulse modulators are discussed in the following chapter. Pulse modulators are commonly employed due to their advantages of reduced propellant consumption and near-linear duty cycle. In general, pulse modulators produce a pulse command sequence to the thruster valves by adjusting pulse width and/or pulse frequency (Song & Agrawal 1999).

2.2.1 Pulse-Width Modulator

The pulse-width modulator (PWM) modulates the width of its output pulses proportionally to the level of torque commanded. Furthermore, the output is not a thruster firing state, but a thruster pulse width. A zero-order-hold filter transmits that signal to the thruster (Wie 1998). More information about PWM's can be found in (Sidi 1997, Wie 1998) Figure 2.5 show the schematic of a PW modulator.

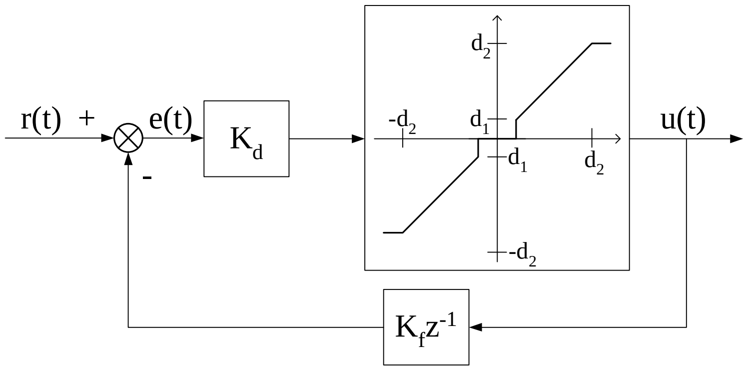


Figure 2.5: Pulse-width modulator

Multi-pulse-width modulators is a derivation from the pulse-width modulator. It allows several pulses to appear in each sampling interval. This system can be interpreted as a PWPF system, but it is based on different assumptions and design techniques (Bernelli-Zazzera, Mantegazza & Nurzia 1998)

2.2.2 Pulse-Frequency Modulator

In pulse-frequency modulators (PFM), a continuous signal is converted into a pulse stream with a frequency that is proportional to the level of torque commanded. The PFM has many derivations, but the most commonly use device is the integral pulse-frequency modulator (IPFM). The IPFM employ a integrator in front of a bang-bang controller, which resets the integrator output whenever a thruster firing command is given. The IPFM schematics is shown in Figure 2.6.

2.2.3 Pseudo Rate Modulator

The pseudo-rate modulator (PRM), or derived-rate modulator (DRM) as it also is known as, consists of a Schmitt trigger and a first order filter in a feedback-loop. Millar & Vigneron (1979) gives a thorough walkthrough of PRM's in their article, and supplementing info can be found in (Wie 1998) and (Sidi 1997).

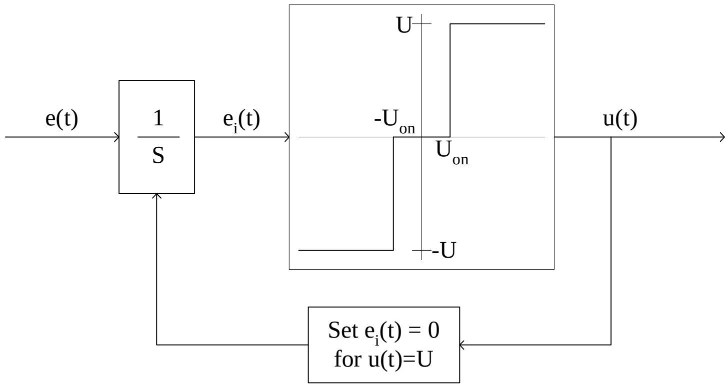


Figure 2.6: Integral pulse-frequency modulator

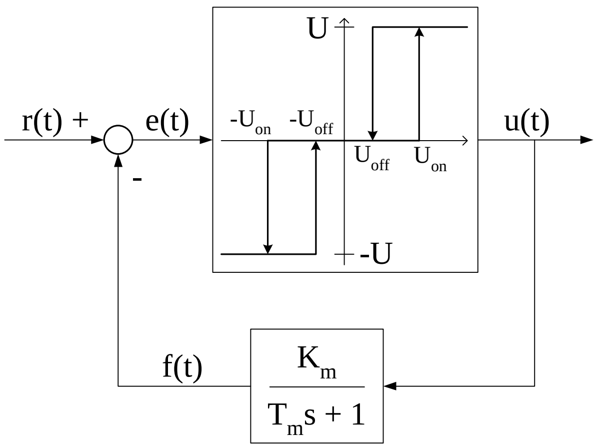


Figure 2.7: Pseudo-rate modulator

2.2.4 Pulse-Width Pulse-Frequency Modulator

The pulse-width pulse-frequency (PWPF) modulator consist of a Schmitt trigger, a first order filter and a feedback loop, just like the PRM. But in the PWPF modulator, the filter is placed in front of the Schmitt trigger and not in the feedback loop. The schematics is shown in Figure 2.8. Relevant literature claims that the PWPF modulator holds advantages over other types of pulse modulators (Wie 1998, Song & Agrawal 2001, Hu & Ma 2005), and it has also been demonstrated that it holds considerable advantages over bang-bang control systems (Krøvel 2004). PWPF systems are predominant in ADCS systems for space use (Sidi 1997). It has the following advantages:

- Close to linear (pseudo-linear) operation (Wie 1998, Sidi 1997).
- A wide range of parameters to tune give the designer freedom with respect to system-specific considerations.

- Tunability gives flexibility under operation as well, the parameters can be tuned to meet different requirements through different phases of operation(McClelland 1994, Buck 1996).

- Low fuel consumption and good pointing accuracy especially when vibrations are present (Buck 1996).

Some of the negative sides are the contribution to the system phase lag (Wie & Plesica 1984), which can cause instability, and the nonlinear characteristic makes it really hard to analyse with respect to stability margins (Buck 1996).

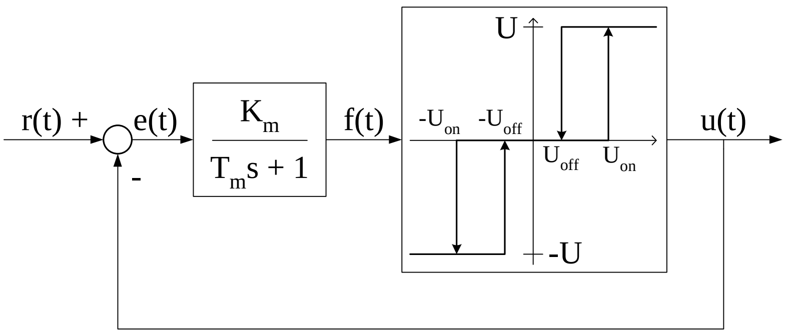


Figure 2.8: PWPF Modulator

Given the advantages the PWPF modulator holds over the other thruster control systems, this modulator is chosen as a candidate for use on ESMO, and it will be thoroughly analysed in the following chapters.

CHAPTER 3

Analysis of PWPF Modulator Behavior

In this chapter, a thorough investigation of the mathematical background PWPF modulator behave is done. Focus will be on the static characteristics, as they are sufficient to determine rigid body system behavior (Wie & Plesica 1984). ESMO will most likely be built in such a manner that it can be considered a rigid body (no flexible appendages). The static characteristics are found by feeding the modulator with a step input of constant magnitude C. A method to determine the modulator parameter behavior when a varying input signal is fed to the modulator, is also explained.

3.1 PWPF Modulator parameters

Before the modulator characteristics are determined, it is necessary to identify the PWPF modulator parameters. From Figure 2.8 in Chapter 2.2.4, one easily recognises four of the system parameters: K_m, T_m, U_{on} and U_{off}. K_m is the filter gain and T_m is the filter time constant, while U_{on} is the Schmitt trigger on-value and U_{off} is the Schmitt trigger off-value. Instead of using U_{off}, h can be used. h is the hysteresis width, and is derived from $U_{on} - U_{off} = h$. There is one additional system parameter, K_{pm}, that does not appear on the figure. This is the pre-modulator gain, which is used to amplify the input signal to a proper magnitude. A thorough explanation of the impact of each of these five parameters is given in Chapter 4.

3.2 Static Behavior Analysis

To determine the basic behavior and response pattern of a given PWPF modulator, a constant input signal is fed to the modulator. This gives the possibility to perform a static analysis of the modulator behavior. The static analysis gives a good indication on how a rigid body spacecraft, such as ESMO, will behave with a PWPF modulator implemented in the control system.

3.2.1 System Equations

With reference to Figure 2.8 , it can be seen that the filter output equation is given by

$$f(s) = e(s)\frac{K_m}{(T_m s + 1)} + \frac{T_m f(0)}{T_m s + 1} \tag{3.1}$$

where $f(0)$ is the initial condition of the filter, and the error signal, composed of a reference signal and the modulator output signal, is

$$e(s) = r(s) - u(s) \tag{3.2}$$

Since both the reference signal and the modulator output are step functions, they are given by

$$r(s) = \frac{C}{s} \tag{3.3}$$

$$u(s) = \frac{U}{s} \tag{3.4}$$

Then equation (3.3) and (3.4) is inserted into equation (3.1) and the resulting equation is

$$f(s) = \frac{K_m(C-U)}{s(T_m s + 1)} + \frac{T_m f(0)}{T_m s + 1} \tag{3.5}$$

Equation (3.5) can then be transformed to the time-domain equation for the system

$$f(t) = K_m (C-U) \left(1 - e^{-t/T_m}\right) + f(0)e^{-t/T_m} \tag{3.6}$$

Equation (3.6) is then used in the following section to determine important system variables.

3.2.2 System Variables

In this section, some of the system variables are identified and derived for later use.

In Figure 3.1 the PWPF modulator response pattern for a constant input is shown. It shows that the filter output starts increasing when the step input occurs. Given the filter time constant (T_m), the filter output gradually increases towards the Schmitt trigger on-value (U_{on}). When the on-value is reached, the Schmitt trigger sets its output to the prescribed magnitude U. This immediately leads to a decrease in the filter input value due to the negative feedback loop, and the filter output then decreases until the Schmitt trigger off-value (U_{off}) is reached. This is the hysteresis effect (h) of the trigger. When the Schmitt trigger off-value is reached, the modulator output is set to zero. The period of time in which the modulator has a non-zero output is called the thruster on-time and is denoted T_{on}. The thruster off-time is denoted T_{off} and is the time when the modulator has a zero output.

On- and Off-time

Based on the calculations from section 3.2.1, the equations for the values discussed above can be derived. Equation (3.6) can be reformulated as

$$f(t) = f(0) + (K_m(C-U) - f(0)) \left(1 - e^{-t/T_m}\right) \tag{3.7}$$

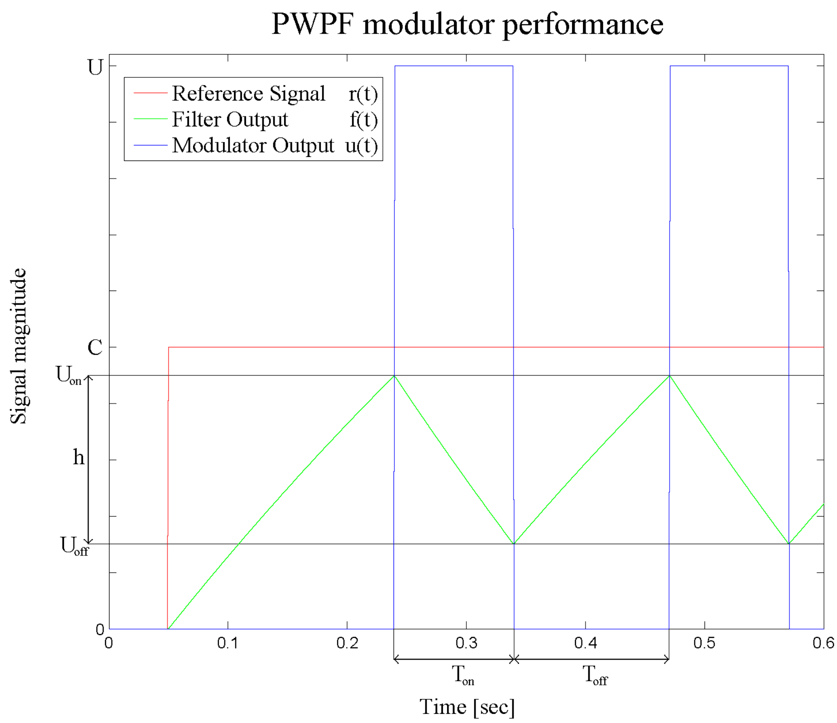


Figure 3.1: PWPF modulator filter, time response

From Figure 3.1 it can be seen that during the thruster on-time, the filter output is decreasing asymptotically. Furthermore, $f(0) = U_{on}$ and $u(t) = U$ in this period. And the decrease will stop at U_{off}. So equation (3.7) can be rewritten as

$$U_{on} + (K_m(C - U) - U_{on})\left(1 - e^{-T_{on}/T_m}\right) = U_{off} \tag{3.8}$$

and is solved for T_{on} to find the thruster on-time

$$T_{on} = -T_m \ln\left(1 - \frac{h}{U_{on} - K_m(C - U)}\right) \tag{3.9}$$

where $h = U_{on} - U_{off}$.

The same process i repeated to find T_{off}, but this time $f(0) = U_{off}$ and $f(t)$ tends towards U_{on}

$$U_{off} + (K_m(C - U) - U_{on})\left(1 - e^{-T_{off}/T_m}\right) = U_{on} \tag{3.10}$$

which leads to

$$T_{off} = -T_m \ln\left(1 - \frac{h}{K_m C - U_{off}}\right) \tag{3.11}$$

Similar results are found in (Buck 1996) and (Sidi 1997).

Duty cycle

Now the duty cycle can be calculated, using equation (3.9) and (3.11)

$$DC = \frac{T_{on}}{T_{on} + T_{off}} \tag{3.12}$$

The duty cycle is a measure on how the modulator responds to an input. It can for example be used to determine how well the modulator output follows the input.

Output frequency

The output frequency can also be determined using T_{on} and T_{off}

$$f_o = \frac{1}{T_{on} + T_{off}} \tag{3.13}$$

Internal deadzone

The input signal needs to have a certain magnitude to activate the Schmitt trigger. This signal magnitude is called internal deadzone. From Figure 2.8, one can easily see that

$$K_m C \geq U_{on} \tag{3.14}$$

to activate the Schmitt trigger. Hence, the internal deadzone C_{dz} is

$$C_{dz} = \frac{U_{on}}{K_m} \tag{3.15}$$

If a pre-modulator gain K_{pre} is used, this gain will have to be multiplied with K_m. The internal deadzone directly determines the pointing accuracy of the spacecraft, since a commanded torque signal with a magnitude smaller than C_{dz} won't activate the spacecraft thrusters.

Saturation level

When the input value gets high enough, the modulator will enter a saturation region. The saturation level is

$$C_{sat} = U + \frac{U_{off}}{K_m} \tag{3.16}$$

When the modulator has entered the saturation region, the appropriate thrusters will be on all the time, and hence no more torque can be produced along that axis. This behavior can be seen in Figure 3.3, for input values above C_{sat}.

Minimum pulse width

For a given set of system variables, it is possible to find the minimum pulse width for the modulator. Equation (3.9) and (3.15) gives

$$\Delta = -T_m \ln\left(1 - \frac{h}{K_m}\right) \tag{3.17}$$

3.3 Describing Function Analysis

By doing a static analysis, the system response to a constant input signal is found. This is sufficient for rigid body systems. But for spacecrafts with a flexible structure, it is necessary to perform an analysis of the dynamic system characteristics (Wie & Plesica 1984). This is done by using the describing function theory. Flexible structures are normally only used on larger spacecrafts, and is therefor not a serious concern for micro-satellites like ESMO. Hence, only a quick walkthrough of describing functions is done here.

The following section is based on a similar chapter in (Wie 1998) and (Anthony, Wie & Caroll 1989), as well as a chapter in (Khalil 2000), and is only a slight rewrite of these.

The describing function of a nonlinear element is the complex ratio of the fundamental harmonic component of the output to the input, when the fundamental harmonic component is significant for a sinusoidal input to that nonlinear element (Wie 1998)

$$N(X, \omega) = (Y_1/X)e^{j\phi} \tag{3.18}$$

where
N = describing function
X = sinusoidal input amplitude
ω = sinusoidal input frequency
Y_1 = fundamental harmonic amplitude
Y_1/X = describing function phase gain
ϕ = describing function phase

To calculate the describing function, a Fourier series analysis will have to be done to obtain the fundamental component of the output. Y_1 and ϕ is expressed as

$$Y_1 = \sqrt{A_1^2 + B_1^2} \tag{3.19}$$

$$\phi = \arctan(A_1/B_1) \tag{3.20}$$

where

$$A_1 = \frac{1}{\pi}\int_0^{2\pi} y(t)cos(\omega t)d(\omega t)) \tag{3.21}$$

$$B_1 = \frac{1}{\pi}\int_0^{2\pi} y(t)sin(\omega t)d(\omega t)) \tag{3.22}$$

and $y(t)$ = is the output of the nonlinear device. If the nonlinear element can be adequately characterised by the describing function $N(X, \omega)$, the loop transfer function is given by $N(X, \omega)G(j\omega)$, where $G(j\omega)$ is the frequency response function of the linear components in the loop (Anthony et al. 1989). To conduct limit cycle analysis, an investigation of the following equation is done

$$1 + N(X, \omega)G(j\omega) = 0 \tag{3.23}$$

$$G(j\omega) = -\frac{1}{N(X,\omega)} \tag{3.24}$$

Equation (3.24) is often called the harmonic balance equation (Khalil 2000). By examining a graphical representation of the loci of $-1/N$ and $G(j\omega)$, it can be determined wether or not limit cycles will occur in the system. If the two loci intersect normally, there will be limit cycling in the system, while a tangential intersection give no limit cycling (Anthony et al. 1989).

The challenge presented by the describing function method is to obtain the most realistic sinusoidal model of the pulse output. For a thruster system, the integral of the thruster torques is angular momentum. Therefore two torque profiles of equal area will impart an identical impulse to a plant, if the plant is linear (Wie 1998). A sine wave of equal area to the pulse output will then be sufficient as a describing function.

Using the area matching criteria, it is then easy to find a describing function to match the modulator output. The area of a modulator pulse, integrated from 0 to π/ω, is simply the minimum pulse width, Δ. The area of the describing function sine over the same area is $2Y_s/\omega$, where Y_s is the area matching sinusoid amplitude and ω is the frequency of both the sine and the pulse signal.

$$\Delta = \frac{2Y_s}{\omega} \Rightarrow Y_s = \frac{\omega\Delta}{2} \tag{3.25}$$

Now, the amplitude of the Fourier fundamental harmonic Y_1 can be found, by solving equation (3.21) and (3.22) and substituting into equation (3.19)

$$Y_1 = \frac{4}{\pi} sin(\frac{\omega\Delta}{2}) = \frac{4}{\pi} sin(Y_s) \tag{3.26}$$

Several other methods of finding a good describing function exists, some of them can be found in (Wie 1998) and (Anthony et al. 1989). This chapter is meant as a brief introduction to describing function analysis for PWPF modulated systems, and the reader should consult other sources of information for a more thorough walkthrough.

3.4 Limit Cycles

Another aspect of PWPF modulated systems (and any other thruster control systems), is that they will be subject to self sustained oscillations because of the nonlinearities in the system. These oscillations are called limit cycles, and in a phase plane analysis they will occur as a closed path in which the system remains for ever, given a stable limit cycle (Khalil 2000).

In a rigid body system, the limit cycles occur because of the minimum thruster on-time. In a flexible system the limit cycles also occur because of system vibrations, due to flexible appendages such as large solar arrays extending several meters from the spacecraft body. (McClelland 1994). The body of ESEO/ESMO is assumed to be rigid, so any limit cycles occurring will be from the minimum thruster on-time.

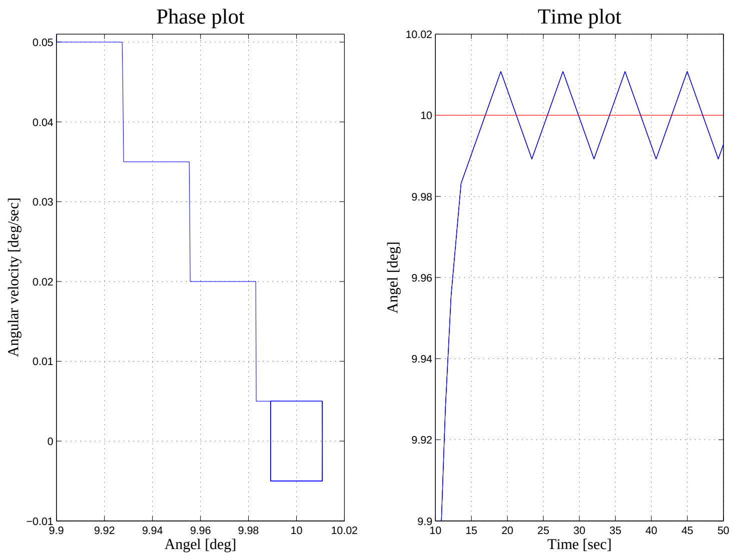


Figure 3.2: A phase plane and time plane behavior plot for a limit cycling system

In the simulation plots, the limit cycling will appear as a oscillation around the steady state. A series of simulations with a simple PD controller system were done. See Figure **??** for details about the system used in the simulation. Figure 3.2 shows the limit cycle behavior about the steady state point for both the phase plane and the time domain.

It is obvious that the oscillation is undesirable with respect to pointing accuracy and fuel consumption, as well as to physical thruster valve wear. The limit cycle problem can be avoided by including a reaction/momentum wheel system, and let this system handle the smallest changes in attitude. Such systems are indeed implemented on most commercial and scientific spacecrafts today.

3.5 Linear Behavior

The purpose of any pulse modulator is to translate the continuous command from the onboard ADCS unit to an on/off signal for the thruster system, as mentioned earlier. To get the best possible control of the spacecraft, the output of the modulator has to be close to or equal to the input, i.e. have a pseudo-linear or linear response. An idealised pseudo-linear response for a PWPF modulator is shown in Figure 3.3. For all practical purposes, the response will differ from the idealised pseudo-linear, as can be seen in Figure 4.3. But with good tuning, it is possible to get pseudo-linear modulator response .

The PWPF modulator has, when tuned correctly, a very large linear operational range. It can also be tuned to be more or less perfectly linear, but then it looses some of its other important characteristics, like noise and disturbance rejection capabilities. As a demonstration of PWPF pseudo-linearity, a system with a simple PD-controller is set up, and tested on a linear actuator and a PWPF modulator system. The test setup is found in C.1. The results, shown in Figure 3.4, indicates that the PWPF modulator is

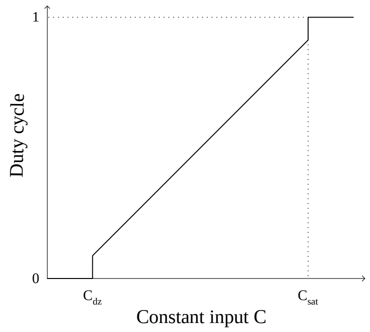


Figure 3.3: Idealised duty cycle vs constant input

unable to follow the linear actuator in the ”rising period” of the slew maneuver. This is because the linear actuator has an unlimited and scalable power output, giving it the capability to follow the regulator output perfectly. The PWPF modulator, on the other hand, has a limited power output and no power scalability (0 or $\pm U$), hence the somewhat slower convergence towards the desired attitude. A phase lag in the modulator filter will also contribute to a slower convergence.

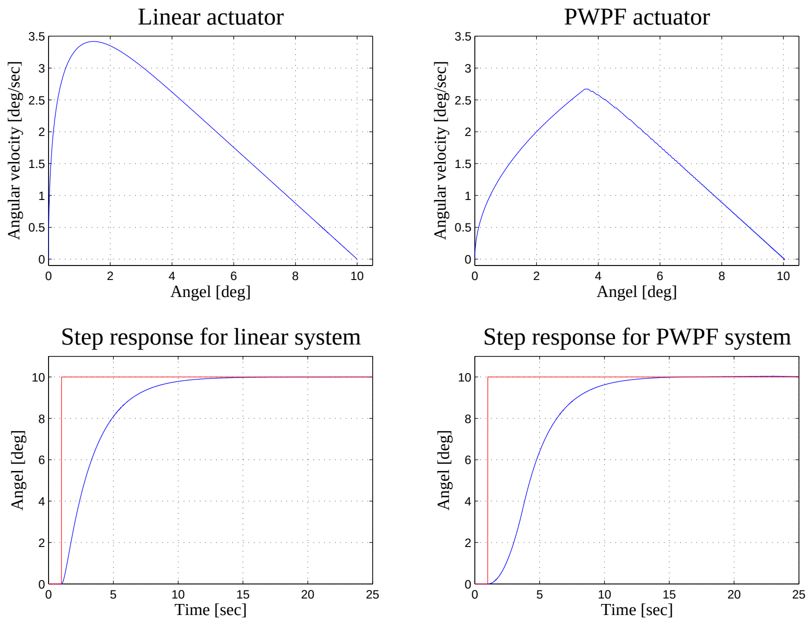


Figure 3.4: A slew maneuver comparison between a linearly actuated system and a PWPF actuated system

The mathematical expression for the modulator duty cycle is found in equation (3.12). The duty cycle is the average output of the modulator, and when compared to the input, it gives an estimate of the linearity of the modulator for a given set of modulator parameters.

CHAPTER 4

Optimisation of PWPF Parameters

In the following chapter, an investigation is done to determine the optimal region of operation for a PWPF modulator. The optimal region is here defined as the region where the modulator behavior is as close to linear as possible ($output \approx input$), and at the same time where fuel consumption and thruster activity is at the lowest. Given the extremely nonlinear nature of the modulator, an analytic approach is difficult (Velde 1983) to follow. Instead, system simulations are done, to recognise what range of parameter values that give the desired modulator behavior.

4.1 Determining Parameters from Static Analysis

4.1.1 System Simulation Setup

All simulations in chapter 4.1 are done using a MATLAB/Simulink setup. Unless otherwise stated, a standard setup with a Simulink model and a range of MATLAB .m-files are used to create the simulation plots used to determine the behavior for varying modulator parameters.

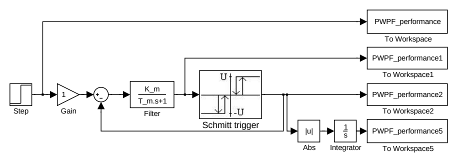


Figure 4.1: The Simulink model used to create the simulation plots

Figure 4.1 show the Simulink model used to create the plots for the static simulations. In Appendix B, an overview of the .m files used to create the plots is given.

General Comments

To maintain the functionality of the Schmitt trigger, the hysteresis value should never exceed twice the U_{on} value or $h < 2U_{on}$, alternatively $U_{off} > -U_{on}$. On the other side of the scale, the hysteresis value should always be larger than zero, or $h > 0$, alternatively $U_{on} > U_{off}$ (recall from Chapter 3.2 that $h = U_{on} - U_{off}$).

The Figures presented in the following chapters are created by varying different modulator parameters in a certain value-range. This range is kept as small as possible to maintain high resolution in the operational area, while at the same time keeping simulation time at at an acceptable level. When a parameter is not varied, it is set to the "Fixed value" in the table below. The fixed values are based on a list of recommendations made by McClelland (1994), that were found to give satisfying PWPF modulator behavior in (Krøvel 2004).

Parameter	Test range	Fixed value
K_m	$0 \rightarrow 10$	4.5
T_m	$0 \rightarrow 1$	0.15
U_{on}	$0 \rightarrow 1$	0.45
U_{off}	$-1 \rightarrow 1$	0.15
K_{pre}	$1 \rightarrow 10$	1
C	$0 \rightarrow 1$	0.75

The pre-modulator gain K_{pre} is only tested in the system tests, since both the static and dynamic test runs have a balanced input.
The constant input, C, is also included in the table. It must not be mistaken for a modulator parameter. It is included to show that the input signal range is from 0 to 1, and that all other parameters are tuned with respect to that.

Plots that are too large to include in this chapter are included in Appendix D. This include plots that show modulator behavior for three different parameters, for example K_m vs T_m for different input frequencies, f. Also, some plots are repeated or shown together to underline specific behavior in this chapter.

4.1.2 Optimising with Respect to Pseudo-Linear Behavior

In chapter 3.5, the pseudo-linear behavior properties of a PWPF modulator was mentioned. In the following, the effect of all modulator parameters on the pseudo-linearity, for a static test setup, will be determined.

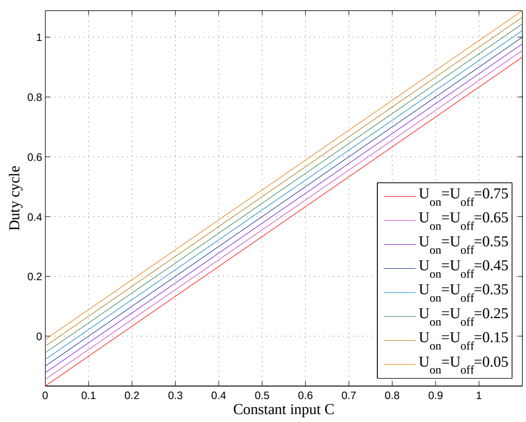


Figure 4.2: Pseudo-linear operation with $U_{on} = U_{off}$ (zero hysteresis)

First, the Schmitt trigger parameters are evaluated. To get an basic idea on how the trigger parameters affect the modulator response, two extreme cases are tested, $U_{off} = U_{on}$ (zero hysteresis) and $U_{on} = -U_{off}$. For the first case, it is discovered that the output follows the input, but with a constant offset. This offset is reduced as U_{on} gets smaller. When $U_{on} \rightarrow 0$, the input≈output, but one should notice that a small U_{on} will give bad noise/disturbance rejection capabilities, and is therefore undesirable. Figure 4.2 show the behavior for a range of U_{on} and U_{off} values.

Then, a varying set of U_{on}-values are tested against a fixed, non-zero hysteresis value. It can now be seen from figure D.1 that a lower U_{on} value will give a more linear relation between the input and the output, even when the hysteresis is kept constant.

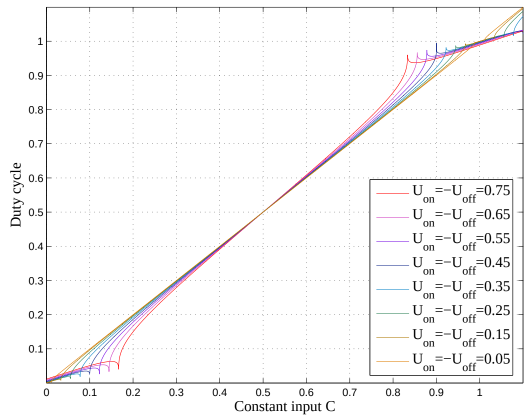


Figure 4.3: Pseudo linear operation with $U_{on} = -U_{off}$

For the second case, $U_{on} = -U_{off}$, a more nonlinear input/output relation appears. But lower values of U_{on} still give a more linear relationship between the input and the output. Figure **??** show the simulation plots for a range of U_{on} and U_{off} values.

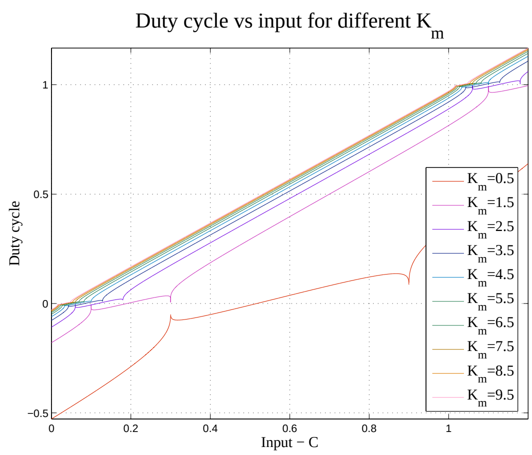


Figure 4.4: The effect on linearity for varying K_m

Next, the filter parameters are evaluated. A quick look at the equation for duty cycle, (3.12), reveals that T_m don't have any impact on the linearity of the modulator. K_m on the other hand, has a profound impact on the linearity. Figure 4.4 show a series of plots for a range of K_m values between 0 and 10. It is easily seen that low K_m give a smaller region of pseudo-linear following, while larger K_m give a more linear input/output relation. The relation is sustained for varying U_{on} and U_{off}, and it is determined that $K_m < 2.5$ will give undesirable modulator behavior. $K_m > 7.5$ give little increase in the linear operation area, and can also be avoided.

With respect to linear behavior, it can then be concluded that U_{on} should be as small as possible and K_m should be in the interval $(2.5 < K_m < 7.5)$.

4.1.3 Optimising with Respect to Thruster Activity

Thruster valves are subject to wear when they are opened and closed. To minimise the thruster valve wear, it is desirable to minimise the number of commanded thruster firings from the PWPF modulator. First, the filter parameter are tested against each other. Figure 4.5 show the number of thruster firings for a range of K_m's and T_m's.

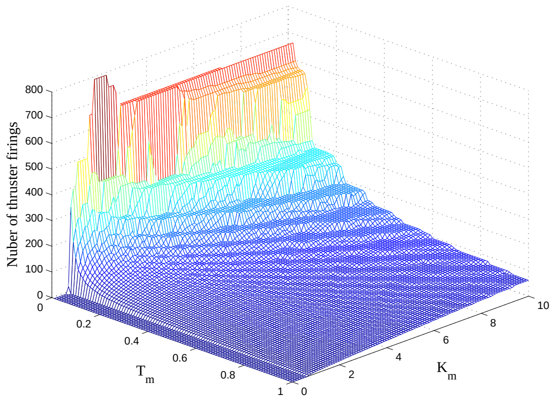


Figure 4.5: Thruster activity for varying K_m and T_m

It is clear that higher T_m's and lower K_m's give less thruster activity. For very low T_m's the thruster activity increases dramatically, since the damping effect that T_m normally give, disappears almost entirely.

Then the number of thruster firings for U_{on} vs h was examined. Figure 4.6 clearly show that low hysteresis values give very high thruster activity and should be avoided. From the figure, one can see that a $h < 0.20$ should be avoided. This means that the left triangle (as seen in Figure 4.6) of the U_{on} vs h plots are uninteresting. Furthermore, a special behavior become apparent for $h = 2U_{on}$ (or $U_{on} = -U_{off}$), as mentioned in Chapter 4.1.1. Here, U_{off} lies to the right of $-U_{on}$ (see Figure 2.8), and the trigger will go directly from a U to $-U$ trigger state, and vice versa, without the possibility to turn it self off. Hence the large number of thruster firings. Another thing to notice, is that for a increasing K_m, the region with high thruster activity increases in area as well. See Figure D.3.

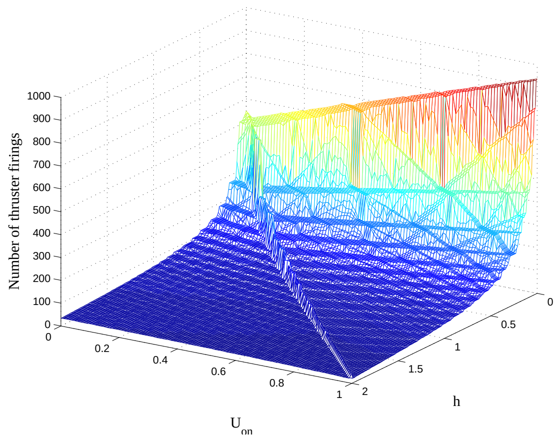


Figure 4.6: Thruster activity for varying U_{on} and h

With respect to thruster activity, it can then be concluded that T_m should be as large as possible (T_m>0.1).

4.1.4 Optimising with Respect to Fuel Consumption

It is obvious that thruster activity and fuel consumption is closely related. But two parameter configurations that give the same fuel consumption, does not necessarily turn the thrusters on and off the same amount of times. In the previous section the parameter ranges that give high thruster activity were identified, and in this section, the focus is on fuel consumption only.

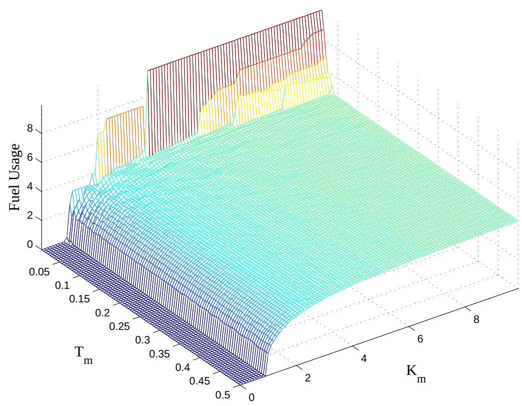


Figure 4.7: Fuel consumption for varying K_m and T_m

First, the filter parameters are evaluated. Figure 4.7 show a fuel consumption plot for K_m and T_m against each other. From the figure it can be seen that very low T_m values give a dramatic increase in fuel consumption. $T_m < 0.05$ should be avoided entirely in general tuning of PWPF modulators, but this demand has already been overridden in the thruster activity chapter.

Then the trigger parameters are evaluated. A simulation run for varying U_{on} and h is done, and the results can be seen in Figure 4.8

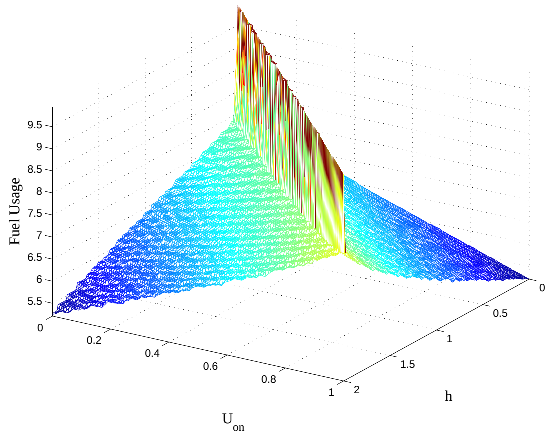


Figure 4.8: Fuel consumption for varying U_{on} and h

4.1.5 Other Optimalisation Criteria

Other factors than those mentioned above are also important for the PWPF modulator performance. A large phase lag will for example be undesirable, since it can make the system unstable. Simple filter theory then indicates that T_m should be as low as possible, since this will give a lower phase lag and better amplitude following. This gives the designer a dilemma when choosing T_m, since thruster firing considerations implies a large T_m. And a low T_m will give a worse high frequency noise rejection as well, since the filter high frequency damping decreases for lower T_m's. A compromise between these considerations give a lower limit for T_m at 0.10 (given earlier in this chapter) and an upper limit at 1.0. Though, it must be noted that the upper limit is a result of a series of bode-plot analysis of the filter only, and should be considered as a hint only.

A PWPF modulator also has a deadzone, given by equation (3.15). This deadzone needs to be as close to the accuracy requirement as possible, indicating that K_m and U_{on} should be tailored to meet this requirement. Here, K_{pre} also comes in to the picture, as a mean to fit the desired torque signal to the linear operation range of the modulator.

4.1.6 Static Optimisation Summary

The obvious parameter limits have now been determined. Some others, like the upper limit on T_m, have only vaguely been indicated, since it is difficult to set that limit only by looking at one optimalisation criteria at the time. Here follows a summary of each parameter impact, for all optimalisation criteria, with a conclusion on parameter range at the end.

Filter Gain, K_m

With respect to linear behavior, K_m needs to be larger than 2.5 and as large as possible. With respect to thruster firings, K_m should be as small as possible, especially for small T_m. With respect to fuel usage, K_m also should be as small as possible, but that is not a demand as stringent as for thruster firings. Noise and disturbance rejection call for a minimum limit on the deadzone, and hence K_m will have to meet a maximum requirement, based on the size of U_{on}.

Filter Time Constant, T_m

With respect to linear behavior, based on the equation for duty cycle, T_m does not have an impact. With respect to thruster firings, T_m should be as large as possible. With respect to fuel usage, T_m should meet a minimum requirement. When considering high frequency noise rejection, T_m should be as large as possible, while a phase lag consideration calls for a small T_m. When looking at the equation for minimum pulse width, Δ, it is easily seen that a larger T_m gives a larger Δ. This could be advantageous in a real system, where wider pulses have higher specific impulse, I_{sp} (Sidi 1997, Sutton & Biblarz 2000).

Schmitt Trigger On-Value, U_{on}

With respect to linear behavior, U_{on} should be as small as possible. With respect to thruster firings, U_{on} should be small. With respect to fuel usage, U_{on} should be large. And in order to maintain the characteristics of the Schmitt trigger, the U_on value will have to be at least half the size of the hysteresis width.

Schmitt Trigger Hysteresis, h

Since U_{off} is a function of h and U_{on}, and h more efficiently show the characteristics of the modulator, h is the parameter explain here. With respect to linear behavior, h should be as small as possible. With respect to thruster firings, h should be as large as possible. With respect to fuel usage, h should be small.

Pre-modulator gain, K_{pre}

No specific conclusion has been draw on K_{pre}, but it has been determined that it will be an effective mean to tailor the desired torque signal to fit the linear range of operation of the modulator.

It has been determined that the PWPF parameters should be in the following region to give the most linear behavior, while at the same time minimising fuel usage and thruster activity.

Parameter	Range
K_m	$2.5 \rightarrow 7.5$
T_m	$0.1 \rightarrow 1.0$
U_{on}	$0.1 \rightarrow 1.0$
h	$0.2 \rightarrow 2U_{on}$

->General comment: The slower a spacecraft can afford to move to a desired attitude, the less fuel the spacecraft will use/the fewer thruster firings well occur.

->new system parameter?: different nominal thrust value in PWPF feedback loop and system feedback loop.

4.2 Determining Parameters from Dynamic Analysis

4.2.1 System Simulation Setup

To determine the useful parameter range for a dynamic input, the constant reference input is swaped with a sine-wave reference. The simulation set up is seen in Figure C.3. A new parameter is introduced, namely the input frequency f. The input signal frequency is is in the range 1 - 150 rad/sec and the signal amplitude is set to 1.

4.2.2 Optimising with Respect to Thruster Activity

First, the new parameter f is tested against T_m. The results, shown in figure 4.9, show that there is very high thruster firing activity for the lowest values of T_m, for all input frequencies. Therefore, T_m should be larger than 0.1. It is difficult to set an upper limit on T_m based on this figure, without knowing the upper bound on the input signal frequency. This frequency will be different for all spacecraft, and no general conclusion can be drawn.

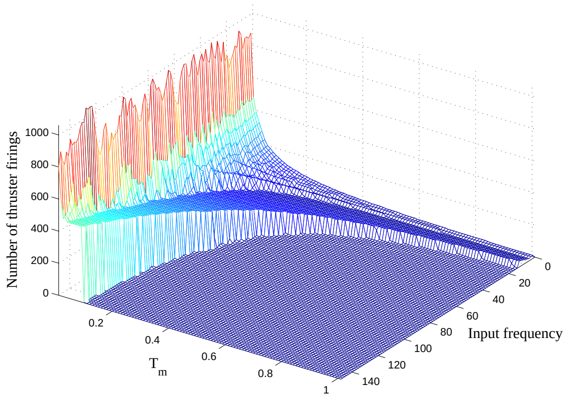


Figure 4.9: The number of thruster firings for varying T_m and f

Then T_m and K_m are tested against each other for varying input frequencies. Figure D.4 show the result, and it is clear that higher input frequency give higher thruster activity and a smaller area where the thrusters actually fires. But it is also apparent that the input frequency has such a large impact on the modulator behavior, that it is impossible to find a coherent range of K_m and T_m values, based on this set of plots. The same conclusion is drawn for U_{on} and h, based on Figure D.5.

4.2.3 Optimising with Respect to Fuel Consumption

First, the input frequency is plotted against T_m. The plot, Figure 4.10, show that the fuel consumption is high in the entire region where thruster firings occurs, but have a peak for the lowest T_m values.

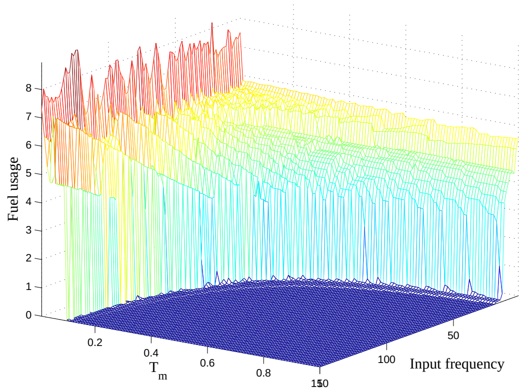


Figure 4.10: The fuel usage for varying T_m and f

4.2.4 Other Optimalisation Criteria

As discussed in a earlier chapter, large T_m values give a large phase lag. It can be seen from Figure 4.9 that there is a large region of no thruster firing. In this region, the modulator phase lag is too large for the modulator to respond to the input signal. So, to be able to follow inputs of high frequency, high T_m values will have to be avoided. For example, if it is desired to be able to follow input frequencies up to 100 rad/sec ($\approx$ 15 Hz), T_m will have to be below 0.2 for a K_m larger than 5.0. For $K_m = 10$, T_m can be as large as 0.4 to meet this requirement.

4.2.5 Dynamic Optimisation Summary

Not many conclusions about the PWPF modulator parameters can be drawn from the dynamic analysis. Through the analysis of the dynamic behavior, it becomes apparent that the input frequency have a severe impact on all modulator parameters, and that no *general* conclusions on K_m, U_{on} and h can be drawn. It is necessary to know what frequency range the system will feed the modulator with, in order to say anything specific about the modulator parameters. The only parameter which has been limited, is T_m with a lower bound of 0.1.

Parameter	Range
T_m	$0.1 \rightarrow$

4.3 Determining Parameters from a System Analysis

The static and dynamic input parameter optimisation tests have given good indications on optimal parameter ranges for the PWPF modulator. But none of these tests feed the PWPF modulator with an input signal similar to those a real satellite ADCS system would send to its PWPF modulator. Therefore, all modulator parameters are tested in a simple spacecraft model, with a PD controller.

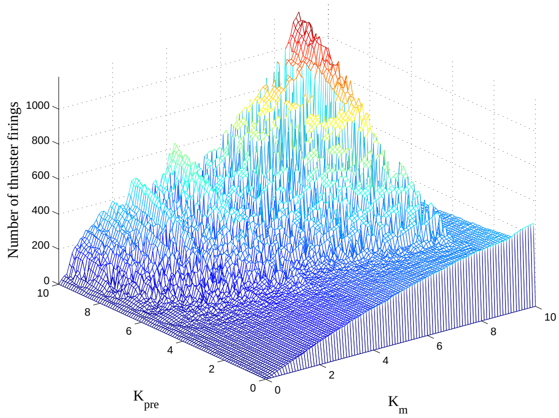


Figure 4.11: Thruster activity for varying pre-modulator gain K_{pre} and filter gain K_m

4.3.1 System Simulation Setup

The spacecraft model with a PD controller and PWPF modulator depicted in Figure C.1 is used to create the simulation plots. The standard parameter ranges are used, except for $C = 15$ which gives a slew maneuver of 15°. New parameters used in this simulation are $K_d = 7$, $K_p = 1.5$, $K_i = 0$, $I = 2$.

4.3.2 Optimising with Respect to Thruster Activity

First the pre-modulator gain, K_{pre}, is tested against K_m. The resulting plot, shown in Figure 4.11, indicates that K_{pre} can be as large as 4 and K_m should be smaller than 4 to stay clear of the high thruster firing activity. A somewhat peculiar behavior is seen for $K_m > 9$, where the thruster activity (and fuel consumption) stays at the same level for all K_{pre} values.

Next, the filter parameters are tested against each other. The results, shown in Figure 4.12, indicates that the system with a PD controller give the same type of behavior as seen both in the static and dynamic analysis chapters, except that now the peaks values are more than a decade higher. The lowest T_m values should be avoided in order to reduce thruster activity ($T_m > 0.1$).

Then the Schmitt trigger parameters U_{on} and h are tested against each other. The resulting plot, shown in Figure 4.13, show that there is an increase in thruster activity for $h < 0.2$, thus this region should be avoided.

4.3.3 Optimising with Respect to Fuel Usage

First the pre-modulator gain, K_{pre}, is tested against K_m. The resulting plot in Figure 4.14 show that K_{pre} should be smaller than 4 and that K_m can range from 0.1 to 10.

Next, the filter parameters are tested against each other. The resulting plot, see Figure 4.15, indicates that the lowest fuel consumption occur for $0.1 < T_m < 0.5$.

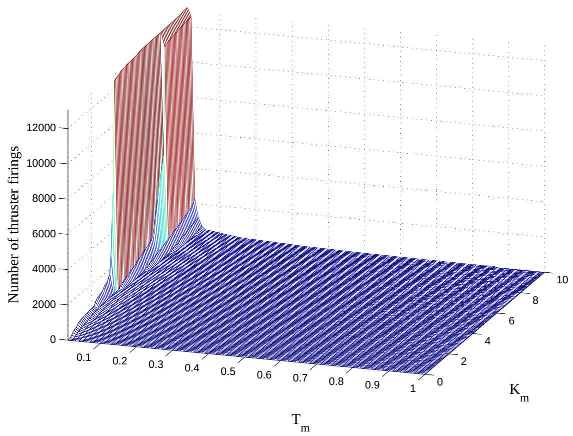


Figure 4.12: Thruster activity for varying filter parameters (K_m and T_m)

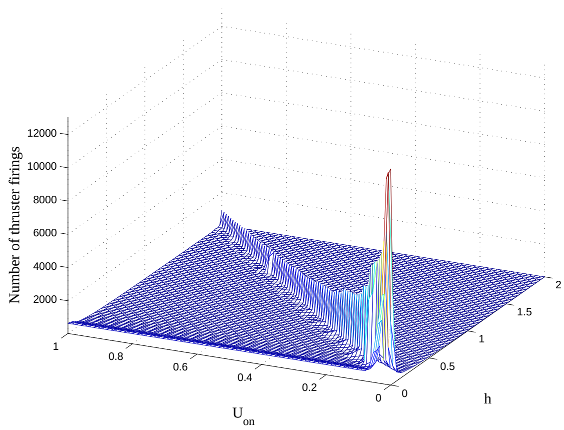


Figure 4.13: Thruster activity for varying Schmitt trigger parameters (U_{on} and h)

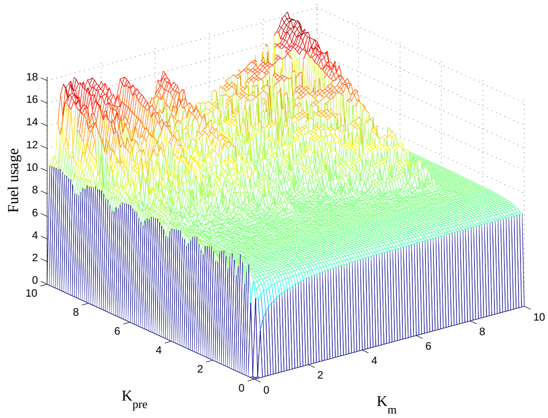


Figure 4.14: Fuel usage for varying pre-modulator gain K_{pre} and filter gain K_m

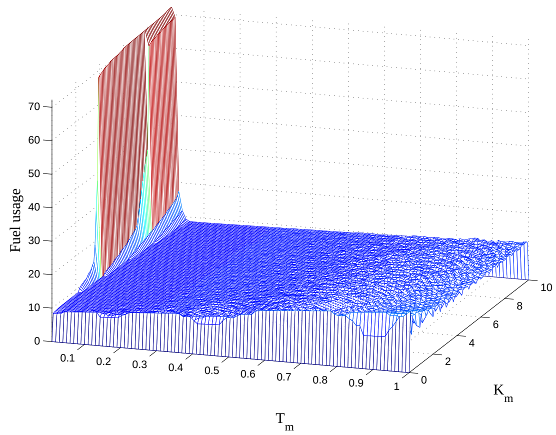


Figure 4.15: Fuel usage for varying K_m and T_m

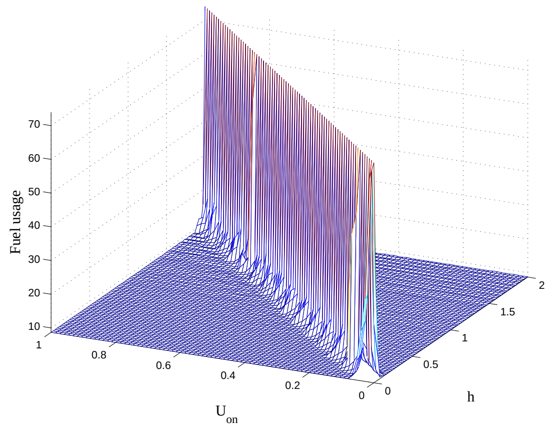


Figure 4.16: Fuel usage for varying U_{on} and h

The trigger parameters U_{on} and h are then tested against each other, as seen in Figure 4.16. Once more the behavior seen earlier repeats itself, but this time there is no drop in the fuel usage for the higher U_{on} values and lower h values, it is more or less flat for the entire region, with exception for $h = 2U_{on}$. As mentioned earlier, h should never reach or exceed $2U_{on}$.

4.3.4 System Optimisation Summary

Parameter	Range
K_m	$0.1 \rightarrow 4$
T_m	$0.1 \rightarrow 0.5$
U_{on}	$0.1 \rightarrow 1$
h	$0.2 \rightarrow 2U_{on}$
K_{pre}	$0.1 \rightarrow 4$

4.3.5 Optimal Parameter Range Summary

Several limiting factors for the PWPF modulator parameters has now been discovered. The different simulation runs have shown that it is possible to limit the parameter values to a certain extent, but that other will have to remain undecided. In the table below, a summary of the results from the previous chapters are presented, and a set of parameter range recommendations are given.

Table 4.1: PWPF parameter recommendation

Parameters	Analysis type			Recommendations
	Static	Dynamic	System	
K_m	$2.5 \rightarrow 7.5$			$2.5 \rightarrow 7.5$
T_m	$0.1 \rightarrow 1.0$	$0.1 \rightarrow 1.0$	$0.1 \rightarrow 0.5$	$0.1 \rightarrow 1.0$
U_{on}	$0.1 \rightarrow 1.0$		$0.1 \rightarrow 1$	$0.1 \rightarrow 1.0$
h	$0.2 \rightarrow 2U_{on}$		$0.2 \rightarrow 2U_{on}$	$0.2 \rightarrow 2U_{on}$

Chapter 5

Optimal PWPF Parameter Ranges for ESMO

In this chapter, optimal PWPF parameters ranges for ESMO (European Student Moon Orbiter) are found. A system model of ESEO, developed by Topland (2004), is used as a base for the simulations. ESMO is believed to be similar to ESEO in size and mass, and therefore ESEO is used, since no accurate physical data exists for ESMO yet. For simplicity, this system model will be called the ESMO model throughout the chapter.

5.1 Initial Parameter Optimisation

In this chapter the goal is to narrow down the parameter ranges found in the previous chapter. This is done through a new series of simulations with the ESMO system model for varying modulator parameters. In the subsequent chapters the results found for each of the three controllers used, are presented.

5.1.1 Simulation Setup

The model used to find the optimal range of PWPF modulator parameters is shown in Figure C.4. Optimal parameter ranges will be found for three different controllers. These controllers are shown in Figure C.5 to C.7. The PWPF modulator implementation is shown in Figure C.8. The PWPF parameter ranges found in Chapter 4, are used as a basis to refine the search for optimal ranges. All simulations are done with an ode1 fixed step solver with a time step of 0.01 seconds.

5.1.2 Initial Parameter Optimisation for a PD Controller

The first step in finding an optimal parameter range, is to look at fuel usage and thruster firings for a set of varying K_{pre} and K_m values, to determine a useful range of gains. Figure 5.1 and 5.2 show a plot for K_{pre} versus K_m, and it can be seen that there is a sharp rise in the number of thruster firings in the region where $K_m > 4.5$, thus this region should be avoided. With respect to fuel consumption, no special conclusion can be drawn.

Next, K_m is tested against T_m. The resulting plot for thruster firings (Figure 5.3)) show that T_m values lower than 0.2 must be avoided. No specific conclusion is drawn

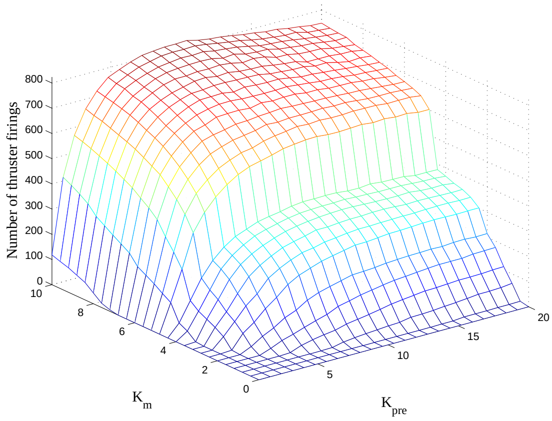


Figure 5.1: The number of thruster firings for varying K_{pre} and K_m for a PD controller.

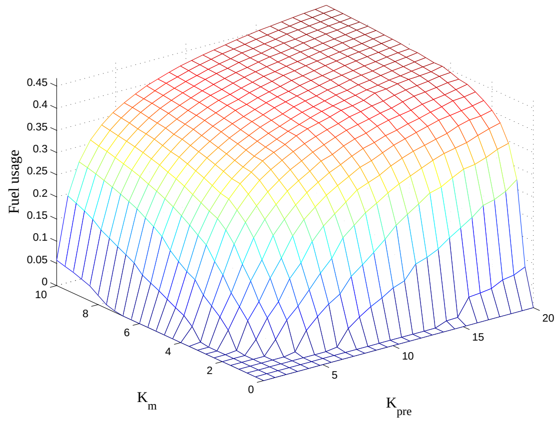


Figure 5.2: Fuel usage for varying K_{pre} and K_m for a PD controller.

on basis of the fuel usage plot.

Finally, the Schmitt trigger parameters U_{on} and h are tested against each other. Both Figure 5.5 and 5.6 indicates that h never should reach $2U_{on}$. The thruster firing plot indicates that U_on and h should be as large as they possibly can (this is difficult to see from the figure included, but can be verified by generating a MATLAB plot).

The tests presented, indicate the following:

Parameter	Range
K_{pre}	$1 \rightarrow$
K_m	$0.1 \rightarrow 4.5$
T_m	$0.1 \rightarrow$
U_{on}	$\rightarrow$
h	$\rightarrow 2U_{on}$

It should be mentioned that these parameter ranges do not consider the accuracy demands, and that it might be necessary to violate them in order to attain the demanded

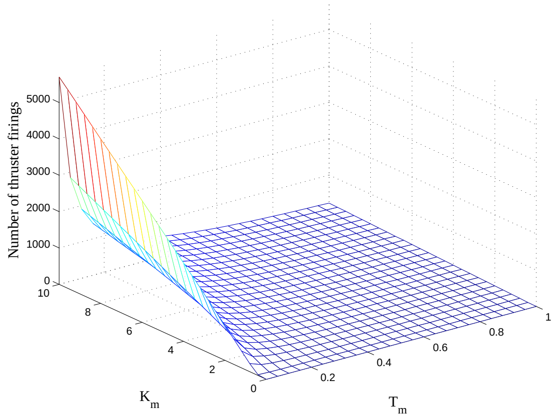


Figure 5.3: The number of thruster firings for varying K_m and T_m for a PD controller.

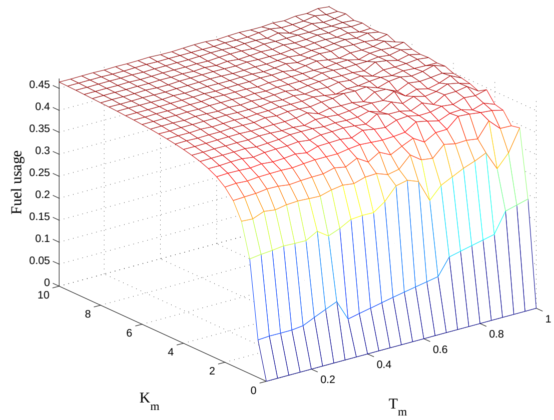


Figure 5.4: Fuel usage for varying K_m and T_m for a PD controller.

accuracy.

General Comments

There are a couple of problems by using these figure as a base to find optimal PWPF parameters. Since the parameters that not are varied have to be fixed, the plots above only are valid for these parameters. They can of course be used to look at general behavior, but it is not guaranteed that they give a correct indication. One should therefore fix the parameter values based on the previous plots, and not on general indications. Then, by using the results from the previous simulation, one could go through the simulation, creating behavior plots for varying parameters based on previous results. In the end this recursive optimisation method would provide a optimal parameter range, and even a set of optimal parameters. This is not done here, as it would require an enormous amount of computational capacity and time to complete (As an example: The six plot in this chapter each required a computation time of 14 hours). This comment also apply to the two following chapters (5.1.3 and 5.1.4).

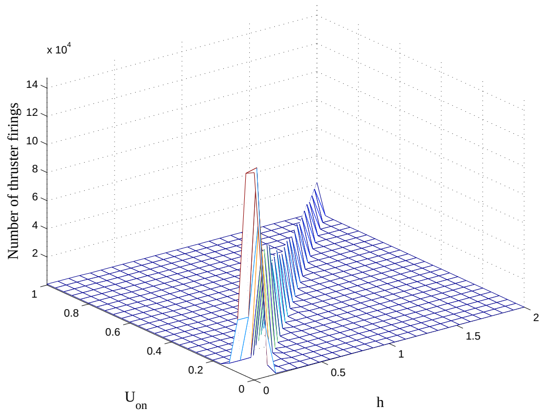


Figure 5.5: The number of thruster firings for varying U_{on} and h for a PD controller.

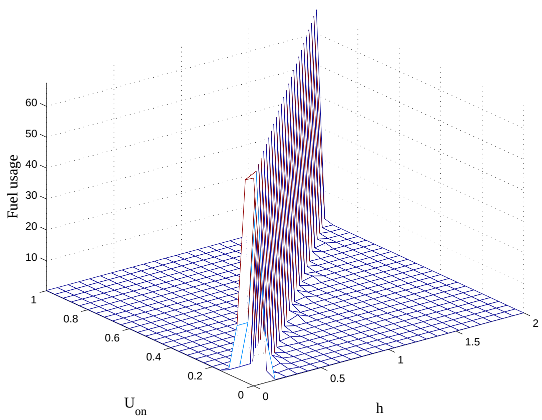


Figure 5.6: Fuel usage for varying U_{on} and h for a PD controller.

5.1.3 Initial Parameter Optimisation for a Sliding Mode Controller

With the comment from the previous chapter in mind, a the optimal parameter ranges for the sliding mode controller will not developed. But it will still be useful to look at the general behavior of the controller. K_{pre} and K_m are especially interesting. A set of varying K_{pre} and K_m values are then evaluated. Figure 5.7 and 5.8 show a plot for K_{pre} versus K_m, and like with the PD controller, it can be seen that there is a sharp rise in the number of thruster firings in the region where $K_m > 3.4$, thus this region should be avoided. This limit will of course change when varying the other modulator parameters, but it gives a good indication to which area the K_m value should lie. With respect to fuel consumption, no special conclusion can be drawn.

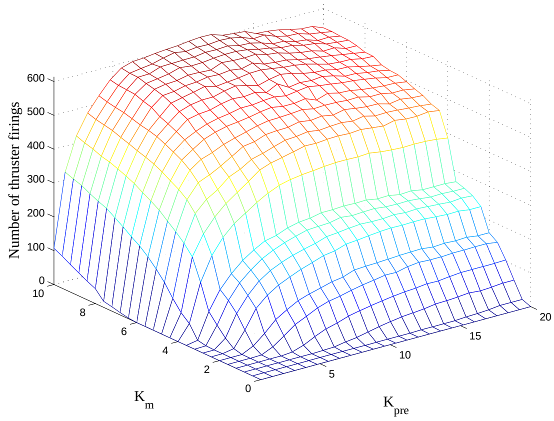


Figure 5.7: The number of thruster firings for varying K_{pre} and K_m for a Sliding Mode controller.

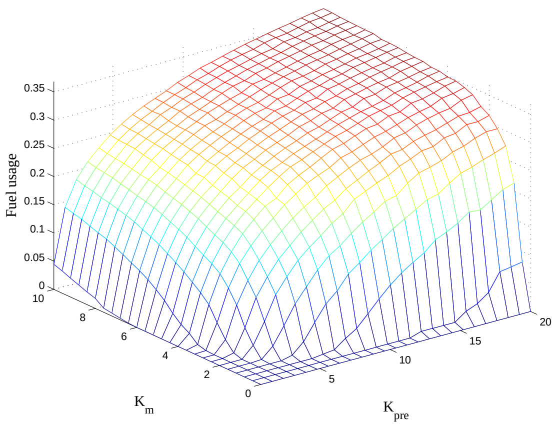


Figure 5.8: Fuel usage for varying K_{pre} and K_m for a Sliding Mode controller.

Next, K_m is tested against T_m. It can be seen from Figure 5.9 that the lowest T_m values must be avoided. With respect to fuel usage (Figure 5.10), no information can be extracted.

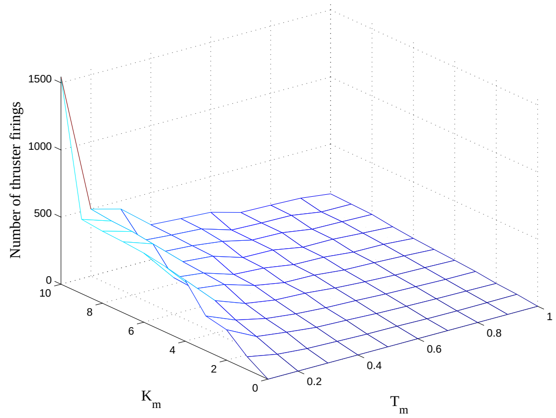


Figure 5.9: The number of thruster firings for varying K_m and T_m for a Sliding Mode controller.

5.1.4 Initial Parameter Optimisation for a Lyapunov Controller

On basis on the comment made in Chapter 5.1.2 and the similarities in behavior between the PD controller and the sliding mode controller, only the K_{pre} vs K_m plot is included here (Figure **??** and **??**). Like with the other controllers, it can be seen that there is a rise in the number of thruster firings in the region where $K_m >$. Since a low upper limit on the K_m potentially can reduce pointing accuracy, a high K_{pre} value might be necessary.

5.2 Concluding Remarks on Optimal Parameter Regions

It is difficult to find general, optimal parameter ranges for ESMO by looking at two and two parameter ranges at the time. Due to the fact that three parameters have to be fixed whenever varying two others, and that the fixed parameter values not necessary have a relevant size with respect to the optimal parameter region, behavior plots only give an approximate indication on how the system will behave. Recursive methods will help solve this problem, but the problem with initial parameter selection still remains. Furthermore, it is uncertain if an analytical approach will be possible,

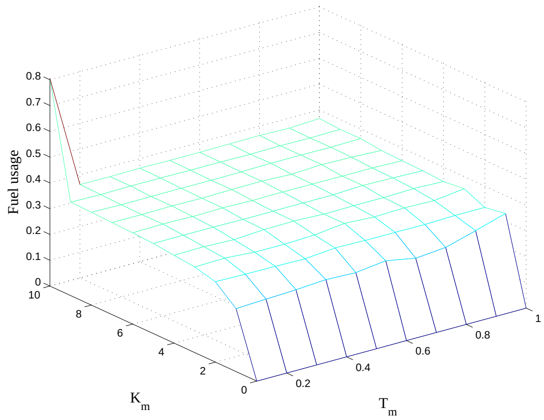


Figure 5.10: Fuel usage for varying K_m and T_m for a Sliding Mode controller.

due to the highly nonlinear nature of both the spacecraft and the PWPF modulator, as well as the presence of five parameters which all influence each other.

Chapter 6

Results

In the previous chapter, optimal PWPF parameter ranges for three different controllers for ESMO were found. In this chapter, these parameter ranges are refined, and a conclusion on a set of optimal PWPF parameters for each of the three controllers is drawn. When the term ”optimal” is used, it needs to be explained a little bit further. When dealing with a nonlinear system such as a satellite, several optimal solutions may exist, and even more sub-optimal solutions. Furthermore, a fuel optimal solution alone is useless, since it most likely will violate system demands such as pointing accuracy and reaction time. Additional requirements, like maximum number of thruster firings will also have to be taken into the account. This gives a complicated set of optimisation criteria, all of which has to be met to some extent. Therefore, it is very difficult to find the fuel optimal PWPF parameters for ESMO. But it is possible to find solutions that meets all the demands listed above, and finding the sub-optimal PWPF parameters in this parameter region. But it is impossible to guarantee that this is the optimal solution. So, when the term ”optimal” is used, it is a reference to a locally optimal solution, not necessarily a global optimal solution.

6.1 Simulation Setup

The model used to find the optimal PWPF modulator parameters is shown in Figure C.4. Optimal parameters will be found for three different controllers. These controllers are shown in Figure C.5 to C.7. The PWPF modulator implementation is shown in Figure C.8. The optimal PWPF parameter ranges for ESMO, found in Chapter 5, are used as a basis for the simulations.

In this section, the optimal parameters are found ”manually”, i.e. by tuning the parameters one by one, to achieve the desired results. First, the parameters are tuned to meet the desired pointing accuracy, then they are tuned to have the lowest fuel consumption possible, within the accuracy limits.

All simulations are done with an ode1 fixed step solver with a time step of 0.01 seconds.

6.1.1 Limiting Factors

When simulating the behavior of ESMO to find the optimal modulator parameters, it is important to have knowledge about any limitation the payload may call for. ESMO will most likely carry an optical imager, to map the surface of the moon with high resolution. It has been mentioned that it should be able to take pictures with a ground resolution of around 0.3 m from a circular orbit 100 km above the surface of the moon. It is reasonable to believe that the image sensor used will have a resolution of around $2.5 \cdot 2.5$ megapixels. To achieve a ground resolution of 0.3 m with such technology from a 100 km height, the lens must have a viewing angle of 0.43°. This calls for a pointing accuracy while performing ground mapping of at least 0.215° to always be able to see objects in the nadir point. But for an efficient mapping, accuracy should be around 90% around one axis, measured in desired coverage area versus actual coverage area. Then the pointing accuracy should be better than 0.043°. Other sub-systems, like the radio transmission line to earth, will meet similar demands. These are very strict demands, and will have to be met by simultaneous tuning of controller parameters as well as PWPF parameter tuning. Controller parameter tuning is beyond the scope of this thesis, so the pointing accuracy demands will be relaxed to 0.5°.

6.2 ESMO Simulations

In this chapter the optimal PWPF parameters are found and presented. All parameters are found by using a Simulink model of ESMO and the respective controller. A ode1 solver with a fixed time step of 0.01 seconds was used.

6.2.1 Optimal PWPF Parameters for a PD Controller

Based on the optimal range of PWPF parameters from Chapter 5.1.2, a series of simulation were performed. As accuracy largely depends upon the values of K_m and K_{pre}, these were initially set to 4.4 and 20, respectively, to ensure a highest possible accuracy, without having too much thruster activity. This lead to a result with too low pointing accuracy, and the U_{on} value was adjusted downwards to increase accuracy further. At the same time T_m was adjusted upwards from a value of 0.2. This resulted in, after some minor adjustments, the following parameters:

Parameter	Value
K_{pre}	20
K_m	4.3
T_m	0.35
U_{on}	0.075
h	0.085

An ESMO simulation run, using these parameters, is shown in Figure **??**.

6.2.2 Optimal PWPF Parameters for a Sliding Mode Controller

Another simulation run was performed, this time with a sliding mode controller. From Figure 5.7, it can be seen that the thruster activity increases drastically for $K_m > 3.6$. The same approach used for the PD controller is used again. But this time, it turned out that a K_{pre} value of 40 was advantageous. The T_m value was adjusted up from a starting value of 0.13. The optimal values was then found to be the following:

Parameter	Value
K_{pre}	40
K_m	3
T_m	0.20
U_{on}	0.19
h	0.17

6.2.3 Optimal PWPF Parameters for a Lyapunov Controller

The last simulation run is for the Lyapunov controller. Yet again the results from Chapter 5 are used as base for simulations. It was found that a very high K_{pre} was needed to get the desired accuracy, and that it was advantageous with a T_m above 0.20. The optimal parameters found are listed in the table below. The resulting system response is shown in Figure **??**.

Parameter	Value
K_{pre}	47
K_m	4
T_m	0.25
U_{on}	0.16
h	0.19

6.3 Concluding Remarks on Optimal PWPF Parameters

A set of optimal PWPF parameters for ESMO have now been found for three different controllers. All simulation runs to find the optimal parameters was based on the optimal regions found in Chapter 5. As was commented there, these ranges were only advisory, and for all controller one or more of the optimal parameters were outside the advised region. This is most likely due to the problems mentioned in Chapter 5.2.

There are some sources of error when relying on numerical simulation results. It was for example discovered that using a shorter time step in the simulations opened for higher accuracy on the cost of more thruster firings and higher fuel usage. The resolution on some of the plots are very low as well, wiping out possibly important details.

Chapter 7

Conclusion and Recommendations for Future Work

7.1 Conclusion

In this work, a brief presentation of the background for the thesis is done, together with a introduction to the Student Space Exploration and Technology Initiative (SSETI) and the SSETI projects.

Then several popular methods of translating a continuous desired torque signal to an on/off signal are presented. Through a review of available literature, the PWPF modulator is chosen as the best alternative. A thorough mathematical analysis of the PWPF modulator is done and the results are used to do an in-depth analysis of the modulator behavior

The PWPF modulator is then selected for a thorough review, and a the mathematical background for the modulator is derived. A series of PWPF modulator tests are performed, and for each of them a conclusion on viable modulator parameter ranges, with respect to fuel consumption, number of thruster firings and other important aspects, is drawn.

Then a series of simulations to find the optimal parameter range for three different controller for ESMO are done, and partially based on the conclusion from these, optimal PWPF parameters for each of the controller used are found. It was possible to tune the PWPF modulator for all the controllers used to attain the desired accuracy.

7.2 Recommendations for future work

If ESMO should become a larger spacecraft than currently planned, it might be necessary to cover a larger area with solar cells as well. It will not be possible to place more solar cells on the body of ESMO, and they will therefore be placed on panels extending from the sides of ESMO. These panels will be flexible, and under the influence of thruster firings, they will start to vibrate if the thrusters fire with the wrong frequency. The characteristics of the PWPF modulator can prevent a part of this vibration, but not all. Therefore methods like input shaping has been developed (Buck 1996), and it can be interesting to perform a study of this technology.

Even though it is claimed in the literature that the PWPF modulator holds several advantages over other pulse modulation techniques (Wie 1998, Song & Agrawal 2001, Hu & Ma 2005), it is difficult to find relevant comparisons of these systems. It would be interesting to do a comparison between the PWPF modulator and other pulse modulators, to find out exactly what advantages the PWPF modulator holds over the others, and vice versa.

All on-off triggers constitutes a nonlinearity in a spacecraft system. As the pointing accuracy demands get higher, these nonlinearities are a source of concern (Avanzini & De Matteis 2001). A further analysis of stability on PWPF modulated systems could be interesting. In conjunction with this, it could be found out whetter or not a analytic approach to find optimal solutions is possible.

Bibliography

Anthony, T. C., Wie, B. & Caroll, S. (1989), 'Pulse modulated control synthesis for a flexible spacecraft', *Journal of Guidance, Control and Dynamics* **13**(6), 1014–1022.

Avanzini, G. & De Matteis, G. (2001), 'Bifurcation analysis of attitude dynamics in rigid spacecraft with switching control logics', *Journal of Guidance, Control and Dynamics* **24**(5), 953–959.

Bernelli-Zazzera, F., Mantegazza, P. & Nurzia, V. (1998), 'Mulit pulse-width modulated control of linear systems', *Journal of Guidance, Control and Dynamics* **21**(1), 64–70.

Buck, N. (1996), Minimum vibration maneuvers using input shaping and pulse-width pulse-frequency modulated thruster control, Master's thesis, Naval Postgraduate School, California, USA.

Elgerd, O. I. (1967), *Control System Theory*, McGraw Hill Electrical Engineering Series, McGraw-Hill.

ESA (2005), 'www.esa.int/esacp/ggg4sxg3aec_index_0.html'. From the ESA web site, 24th May 2005.

Hegrenæs, y. (2004), Attitude control by means of explicit model predictive control, via multi-parametric quadratic programming, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Hu, Q. & Ma, G. (2005), Variable structure control and active vibration suppression of flexible spacecraft durung attitude maneuver. To be published in Aerospace Science and Technology.

Khalil, H. K. (2000), *Nonlinear Systems*, 3rd edn, Pearson Education International.

Krøvel, T. D. (2004), PWPF modulation of thrusters on the micro-satellite SSETI/ESMO, Technical report, Norwegian University of Science and Technology, Trondheim, Norway.

McClelland, R. S. (1994), Spacecraft attitude control system performance using pulse-width pulse-frequency modulated thrusters, Master's thesis, Naval Postgraduate School, California, USA.

Millar, R. & Vigneron, F. (1979), 'Attitude stability of a pseudorate jet-controlled flexible spacecraft', *Journal of Guidance, Control and Dynamics* **2**(2), 111–118.

Sidi, M. J. (1997), *Spacecraft Dynamics and Control - A Practival Engineering Approach*, Cambridge Aerospace Series, Cambridge University Press.

Song, G. & Agrawal, B. N. (1999), Vibration refuction for flexible spacecraft attitude control using pwpf modulator and smart structures, *in* 'Aerospace Conference, 1999. Proceedings', Vol. 2, IEEE, pp. 161–172.

Song, G. & Agrawal, B. N. (2001), 'Vibration supression of flexible spacecraft during attitude control', *Acta Astronautica* **49**(2), 73–83.

Sutton, G. P. & Biblarz, O. (2000), *Rocket Propulsion Elements*, 7th edn, John Wiley and Sons.

Topland, M. P. (2004), Nonlinear attitude control of the micro-satellite eseo, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Velde, W. V. (1983), 'Design of space structure control system using on-off thrusters', *Journal of Guidance, Control and Dynamics* **6**(1), 53–60.

Wie, B. (1998), *Space Vehicle Dynamics and Control*, AIAA Education Series, AIAA.

Wie, B. & Plesica, C. (1984), 'Attitude stabilization of flexible spacecraft during station-keeping maneuvers', *Journal of Guidance, Control and Dynamics* **7**(4), 430–436.

Appendix A

Abbreviations

Appendix B

MATLAB Code

B.1 Code for Chapter 3

In this section, the code used in chapter 3 is presented.

B.1.1 Figure 3.1

```
%
% Creates a plot of the PWPF modulator output, the filter output and the
% constant reference value. The plot show important aspects of the filter
% and Schmitt trigger behavior.
% The practical interpretation of a system receiving a constant input
% is a system that continuously increases its rotational speed around one
% axis.

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/06/23 16:20 $

C     = 0.5                            %Constant input value
K_p   = 1                              %Pre-modulator gain
K_m   = 4.5                            %Modulator gain
T_m   = 0.85                           %Modulator time constant
U_on  = 0.45                           %Schmitt trigger on-value
U_off = U_on/3                         %Schmitt trigger off-value

t_s   = 0.6;                           %Set the simulation time-span
                                       %to 0.6 seconds
simopts = simset('solver','ode5',...   %Use a ode5 solver with a step
    'FixedStep', 1e-3);                %length of 0.005
sim('PWPF_test',t_s, simopts);         %Run 'PWPF_test' for t_s second
                                       %with the parameters from simopts

figure(2);                             %
hold on;                               %
plot(PWPF_perf.time,...                %Reference
    PWPF_perf.signals.values, 'r');    %
```

```
plot(PWPF_perf2.time,...                %Filter output
    PWPF_perf2.signals.values, 'g');    %
plot(PWPF_perf3.time,...                %Modulator output
    PWPF_perf3.signals.values, 'b');    %
title('PWPF modulator performance',...  %Title the plot
    'Interpreter','tex',...             %
    'FontName','Times New Roman',...    %
    'FontSize',20);                     %
legend({'Reference Signal',...          %Create a legend
    'Filter Output',...                 %
    'Modulator Output'},...             %
    'Location', 'NorthWest',...         %
    'FontName','Times New Roman',...    %
    'FontSize',14,...                   %
    'Interpreter','tex');               %
xlabel('Time [sec]',...                 %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Signal magnitude',...           %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %

grid off;                               %
axis tight;                             %
hold off;                               %
box on;                                 %

% End of file
```

B.1.2 Figure 3.2

```
%
% Creates a plot of the limit cycle behavior in both the phase plane and
% the time domain for the PWPF modulator. The model used to get
% the plot, is a simple PD-controlled system. The limit cycles, when they occur,
% will cause unnecessary thruster activity and should be avoided.

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/06/23 20:50 $

C     = 15;                             %Constant input value
K_p   = 1.5;                            %Proportional gain
K_d   = 4;                              %Derivative gain
K_m   = 4.5;                            %Modulator gain
T_m   = 0.15;                           %Modulator time constant
U_on  = 0.45;                           %Schmitt trigger on-value
U_off = U_on/3;                         %Schmitt trigger off-value
U=1;                                    %Schmitt trigger output signal level
```

```
t_s=100;                                %Set the simulation time-span to
                                        %50 seconds
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);                 %length of 0.005
                                        %

sim('PWPF_test_with_PD',t_s, simopts);  %Run 'PWPF_test_with_PD' for t_s
                                        %second with the parameters from
                                        %simopts

figure(3);                              %
subplot(1,2,1);                         %
    plot(PWPF_perf.signals.values,...   %
        PWPF_perf1.signals.values);     %Create a phase plot of the
                                        %modulator behavior
    title('Phase plot',...              %Give the plot a title
        'Interpreter','tex',...         %
        'FontName','Times New Roman',...%
        'FontSize',20);                 %
    xlabel('Angle [deg]',...            %Label the axis
        'FontName','Times New Roman',...%
        'FontSize',14);                 %
    ylabel('Angular velocity [deg/sec]',...
        'FontName','Times New Roman',...%
        'FontSize',14);                 %
    grid on;                            %
    axis tight;                         %
    % xlim([9.9 10.02]);                %
    % ylim([-0.01 0.051]);              %
subplot(1,2,2); hold on;                %
    plot(PWPF_perf.time,...             %Create a time plot of the behavior
        PWPF_perf.signals.values, 'b'); %
    plot(PWPF_perf2.time,...            %Create a time plot of the behavior
        PWPF_perf2.signals.values, 'r');%
    title('Time plot',...
        'Interpreter','tex',...         %Give the plot a title
        'FontName','Times New Roman',...%
        'FontSize',20);                 %
    xlabel('Time [sec]',...             %Label the axis
        'FontName','Times New Roman',...%
        'FontSize',14);                 %
    ylabel('Angle [deg]',...            %
        'FontName','Times New Roman',...%
        'FontSize',14);                 %
    grid on;                            %
    axis tight;                         %
```

```
    hold off;                           %
    % xlim([10 50.01]);                 %
    % ylim([9.9 10.02]);                %

box on;                                 %

% End of file
```

B.1.3 Figure 3.4

```
%
% Creates a plot of the step response for a system with a PWPF modulator
% and a linear actuator. The model used to create the plot, is a simple PD-
% controlled system. The behavior of the PWPF system gives an idea on how
% it performs vs a linear actuator, or its "linearity".

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/06/23 21:50 $

C     = 15;                             %Constant input value
K_p   = 1.5;                            %Proportional gain
K_d   = 4;                              %Derivative gain
K_m   = 4.5;                            %Modulator gain
T_m   = 0.15;                           %Modulator time constant
U_on  = 0.35;                           %Schmitt trigger on-value
U_off = U_on/3;                         %Schmitt trigger off-value
U=1;                                    %Schmitt trigger output signal level

t_s=50;                                 %Set the simulation time-span to
                                        %50 seconds
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);                 %lenght of 0.005
sim('PWPF_test_with_PD',t_s, simopts);  %Run 'PWPF_test_with_PD' for t_s
                                        %second with the parameters from
                                        %simopts
sim('Linear_test_with_PD',t_s, simopts);%Run 'Linear_test_with_PD' for t_s
                                        %second with the parameters from
                                        %simopts
figure(1);                              %
subplot(1,2,1); hold on;                %
    plot(Linear_perf.signals.values,... %Create a phase plot
        Linear_perf1.signals.values,'b');
    plot(PWPF_perf.signals.values,...   %
        PWPF_perf1.signals.values,'g'); %
    xlabel('Angle [deg]',...            %Label the axis
        'Interpreter','tex',...         %
        'FontName','Times New Roman',...%
        'FontSize',14);                 %
```

```
    ylabel('Angular velocity [deg/sec]',...
        'Interpreter','tex',...         %
        'FontName','Times New Roman',...%
        'FontSize',14);                 %
    box on;                             %
    %axis tight;
    grid on;                            %
    %xlim([0 C+0.5]);
    %ylim([-0.1 3.5]);

subplot(1,2,2); hold on;                %
    plot(Linear_perf.time,...           %Create a time plot
        Linear_perf.signals.values,'b');%
    plot(PWPF_perf.time,...             %
        PWPF_perf.signals.values,'g');  %
    plot(Linear_perf2.time,...          %
        Linear_perf2.signals.values,'r');
    xlabel('Time [sec]',...             %Label the axis
        'Interpreter','tex',...         %
        'FontName','Times New Roman',...%
        'FontSize',14);                 %
    ylabel('Angle [deg]',...            %
        'Interpreter','tex',...         %
        'FontName','Times New Roman',...%
        'FontSize',14);                 %
    hold off;                           %
    box on;                             %
    grid on;                            %

% End of file
```

B.2 Code for Chapter 4

In this section, the code used in Chapter 4 is presented.

B.2.1 Figure 4.2

```
%
% Creates a plot of duty cycle vs constant input value for varying U_on=
% U_off (zero hysteresis).

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/06/26 21:10 $

K_pre = 1;                              %Pre-modulator gain
K_m   = 4.5;                            %Filter gain
T_m   = 0.15;                           %Filter time constant
```

```
U     = 1;                              %Modulator output value

for i=1:1:8                             %For-loop to give a range of
    U_on  = 0.1*i-0.05                  %U_on values and
    U_off = U_on-0.0001;                %U_off values, with the following
    h     = U_on-U_off;                 %hysteresis width
    for i=1:1:12000                     %For-loop to give a range of C
                                        %values
        Cv(i) =(i-1)*0.0001;            %Create a array of C values
        C     =(i-1)*0.0001;            %Set the C value
        T_on  =-T_m*log(1-h/...         %Calculate T_on, based on formula
            (U_on-K_m*(C-U)));          %(3.9)
        T_off =-T_m*log(1-h/...         %Calculate T_off, based on formula
            (K_m*C-U_off));             %(3.10)
        DC(i) =T_on/(T_on+T_off);       %Calculate the duty cycle and
                                        %create an array of the values
    end                                 %
    figure(1);                          %
    plot(Cv, DC);                       %Create a plot of the duty cycle vs
                                        %C for each K_m value
    axis tight;                         %
    hold on;                            %
    grid on;                            %
end                                     %
hold off;                               %

xlabel('Constant input C',...           %Label the axis
    'Interpreter','tex',...             %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Duty cycle',...                 %
    'Interpreter','tex',...             %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
legend({'U_{on}=U_{off}=0.05',...       %Create a legend
    'U_{on}=U_{off}=0.15',...           %
    'U_{on}=U_{off}=0.25',...           %
    'U_{on}=U_{off}=0.35',...           %
    'U_{on}=U_{off}=0.45',...           %
    'U_{on}=U_{off}=0.55',...           %
    'U_{on}=U_{off}=0.65',...           %
    'U_{on}=U_{off}=0.75'},...          %
    'FontName','Times New Roman',...    %
    'FontSize',14,...                   %
    'Location','SouthEast')             %

% End of file
```

B.2.2 Figure 4.3

```
%
% Creates a plot of duty cycle vs constant input value for a varying U_on=
% -U_off (hysteresis equal two times U_on).

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/06/26 21:20 $

for i=1:1:9                             %For-loop to give a range of
    U_on  = 0.1*i-0.05                  %U_on values and
    U_off = -U_on;                      %U_off values, with the following
    h     = U_on-U_off;                 %hysteresis width
    for j=1:1:12000                     %For-loop to give a range of C
                                        %values
        Cv(j) =(j-1)*0.0001;            %Create a array of C values
        C     =(j-1)*0.0001;            %Set the C value
        T_on  =-T_m*log(1-h/...         %Calculate T_on, based on formula
            (U_on-K_m*(C-U)));          %(3.9)
        T_off =-T_m*log(1-h/...         %Calculate T_off, based on formula
            (K_m*C-U_off));             %(3.10)
        DC(j) =T_on/(T_on+T_off);       %Calculate the duty cycle and
                                        %create an array of the values
    end                                 %
    figure(1);                          %
    plot(Cv, DC);                       %Create a plot of the duty cycle vs
                                        %C for each K_m value
    axis tight;                         %
    hold on;                            %
    grid on;                            %
end                                     %
hold off;                               %

xlabel('Constant input C',...           %Label the axis
    'Interpreter','tex',...             %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Duty cycle',...                 %
    'Interpreter','tex',...             %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
legend({'U_{on}=-U_{off}=0.05',...      %Create a legend
    'U_{on}=-U_{off}=0.15',...          %
    'U_{on}=-U_{off}=0.25',...          %
    'U_{on}=-U_{off}=0.35',...          %
    'U_{on}=-U_{off}=0.45',...          %
    'U_{on}=-U_{off}=0.55',...          %
    'U_{on}=-U_{off}=0.65',...          %
```

```
    'U_{on}=-U_{off}=0.75'},...         %
    'FontName','Times New Roman',...    %
    'FontSize',14,...                   %
    'Location','SouthEast')             %

% End of file
```

B.2.3 Figure 4.4

```
%
% Creates a plot of duty cycle vs constant input value for a varying K_m

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/06/26 20:45 $

K_pre = 1;                              %Pre-modulator gain
T_m   = 0.15;                           %Filter time constant
U_on  = 0.45;                           %Schmitt trigger on-value
U_off = 0.15;                           %Schmitt trigger off-value
h     = U_on-U_off;                     %Hysteresis value
U     = 1;                              %Modulator output value


for i=1:1:10                            %For-loop to give a range of K_m
                                        %values
    K_m=i-0.5                           %Set K_m
%     C_dz  = U_on/K_m;                 %Calculate the deadzone for the
%                                       %given K_m value
%     C_sat = 1+U_off/K_m;              %Calculate the saturation limit for
%                                       %the given K_m value
    for i=1:1:12000                     %For-loop to give a range of C
                                        %values
        Cv(i) =(i-1)*0.0001;            %Create a array of C values
        C     =(i-1)*0.0001;            %Set the C value
        T_on  =-T_m*log(1-h/...         %Calculate T_on, based on formula
            (U_on-K_m*(C-U)));          %(3.9)
        T_off =-T_m*log(1-h/...         %Calculate T_off, based on formula
            (K_m*C-U_off));             %(3.10)
        DC(i) =T_on/(T_on+T_off);       %Calculate the duty cycle and
                                        %create an array of the values
    end                                 %
    figure(1);                          %
    plot(Cv, DC);                       %Create a plot of the duty cycle vs
                                        %C for each K_m value
    axis tight;                         %
    hold on;                            %
    grid on;                            %
end                                     %
```

```
hold off;                               %

xlabel('Constant input - C',...         %Label the axis
    'Interpreter','tex',...             %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Duty cycle',...                 %
    'Interpreter','tex',...             %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
legend(...                              %Create a legend
  {'K_{m}=0.5','K_{m}=1.5',...          %
  'K_{m}=2.5','K_{m}=3.5',...           %
  'K_{m}=4.5','K_{m}=5.5',...           %
  'K_{m}=6.5','K_{m}=7.5',...           %
  'K_{m}=8.5','K_{m}=9.5'},...          %
  'FontName','Times New Roman',...      %
  'FontSize',14,...                     %
  'Location','SouthEast');              %

% End of file
```

B.2.4 Figure 4.5

```
%thruster_firings_plot_for_K_m_vs_T_m.m
%
% Creates a 3D mesh of the number of thruster firings for different values
% of K_m and T_m. All other variables are set prior to the simulation.
% N and M determines number of steps for the K_m and T_m loop, respectively.
%

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/07/01 19:30 $

clear all

C=0.75;                                 %Constant input value
K_p=1;                                  %Pre-modulator gain
U_on=0.45;                              %Trigger on-value
h=0.3;                                  %Trigger hysteresis width
U=1;                                    %Trigger output level

N=100;                                  %Number of steps in the i for-loop
M=100;                                  %Number of steps in the j for-loop

t_s=10;                                 %Set the simulation time span
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);                 %length of 0.005
```

```
for i=2:1:(N+1)                          %Start the i for-loop
    i
    K_m=(10/N)*(i-1);                    %Set the K_{m} value range
    K_m_values(i)=K_m;                   %Create an array of the K_{m} values
    for j=1:1:M                          %Start the j for-loop
        T_m=(1/M)*(j);                   %Set the T_{m} value range
        T_m_values(j)=T_m;               %Create an array of the T_{m} values
        sim('PWPF_test',t_s, simopts);   %Run 'PWPF_test' for t_s second
                                         %with the parameters from simout
        Thruter_firings(i,j)=...         %Count the number of thruster firings
            thruster_firing_counter(...  %
            PWPF_performance2...         %
            .signals.values);            %
    end                                  %
end                                      %

mesh(T_m_values,...                      %Create a 3D mesh of the number of
    K_m_values,...                       %thruster firings
    Thruter_firings);                    %
zlabel('Number of thruster firings',... %Label the axis
    'FontName','Times New Roman',...     %
    'FontSize',14);                      %
ylabel('K_{m}',...                       %
    'FontName','Times New Roman',...     %
    'FontSize',14);                      %
xlabel('T_{m}',...                       %
    'FontName','Times New Roman',...     %
    'FontSize',14);                      %
axis tight;                              %
```

B.2.5 Figure 4.6

```
%thruster_firings_plot_for_U_on_vs_h.m
%
% Creates a 3D mesh of the number of thruster firings for different values
% of U_on and h. All other variables are set prior to the simulation.
% N and M determines number of steps for the U_on and h loop, respectively.
%

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/07/01 20:30 $

C   = 0.75;                              %Constant input value
K_p = 1;                                 %Pre-modulator gain
K_m = 4.5                                %Modulator gain
T_m = 0.15                               %Modulator time constant
U   = 1;                                 %Trigger output level
```

```
N=100;                                  %Number of steps in the i for-loop
M=10=;                                  %Number of steps in the j for-loop

t_s=10;                                 %Set the simulation time span
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);                 %length of 0.005

for i=1:1:(N+1)                         %Start the i for-loop
    i
    U_on=(1/N)*(i-1);                   %Set the U_{on} value range
    U_on_values(i)=U_on;                %Create an array of the U_{on} values
    for j=1:1:(M+1)                     %Start the j for-loop
        h=(2/M)*(j-1);                  %Set the h value range
        h_values(j)=h;                  %Create an array of the h values
        sim('PWPF_test',t_s, simopts);  %Run 'PWPF_test' for t_s second
                                        %with the parameters from simout
        Thruter_firings(i,j)=...        %Count the number of thruster firings
            thruster_firing_counter(... %
            PWPF_performance2...        %
            .signals.values);           %
    end                                 %
end                                     %

figure(4)                               %
mesh(h_values,...                       %Create a 3D mesh of the number of
    U_on_values,...                     %thruster firings
    Thruter_firings);                   %
zlabel('Number of thruster firings',... %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('U_{on}',...                     %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
xlabel('h',...                          %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %
```

B.2.6 Figure 4.7

```
%Fuel_usage_plot_for_K_m_vs_T_m.m
%
% Creates a 3D mesh of the fuel usage for different values of K_m and T_m.
% All other variables are set prior to the simulation.
% N and M determines number of steps for the K_m and T_m loop, respectively.
%
```

```
% Author: Trond Dagfinn Krøvel
% $Revision: 1.2 $  $Date: 2005/07/02 22:00 $

clear all

C   = 0.75;                         %Constant input value
K_p = 1;                            %Pre-modulator gain
U_on= 0.45;                         %Trigger on-value
h   = 0.3;                          %Trigger hysteresis width
U   = 1;                            %Trigger output level

N   = 100;                          %Number of steps in the i for-loop
M   = 100;                          %Number of steps in the j for-loop

t_s = 10;                           %Set the simulation time span
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);             %length of 0.005

for i=2:1:(N+1)                     %Start the i for-loop
    i
    K_m=(10/N)*(i-1);               %Set the K_{m} value range
    K_m_values(i)=K_m;              %Create an array of the K_{m} values
    for j=1:1:M                     %Start the j for-loop
        T_m=(1/M)*(j);              %Set the T_{m} value range
        T_m_values(j)=T_m;          %Create an array of the T_{m} values
        sim('PWPF_test',t_s, simopts);  %Run 'PWPF_test' for t_s second
                                    %with the parameters from simout
        Fuel_usage(i,j)=...         %Estimate fuel usage
            PWPF_performance5.signals...%
            .values(size(...        %
            PWPF_performance5.signals...%
            .values, 1));           %
    end                             %
end                                 %

figure(1)                           %
mesh(T_m_values,...                 %Create a 3D mesh of the fuel usage
    K_m_values,...                  %
    Fuel_usage);                    %
zlabel('Fuel usage',...             %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                 %
ylabel('K_{m}',...                  %
    'FontName','Times New Roman',...    %
    'FontSize',14);                 %
xlabel('T_{m}',...                  %
    'FontName','Times New Roman',...    %
```

```
    'FontSize',14);                  %
axis tight;                          %
```

B.2.7 Figure 4.8

```
%Fuel_usage_plot_for_U_on_vs_h.m
% Creates a 3D plot and a contour plot of the fuel usage for different values
% of U_on and h. All other variables are set prior to the simulation.
% N and M determines number of steps for the U_on and h loop, respectively.
%

% Author: Trond Dagfinn Krøvel
% $Revision: 1.2 $  $Date: 2005/07/02 22:10 $

clear all

C   = 0.75;                          %Constant input value
K_p = 1;                             %Pre-modulator gain
K_m = 4.5                            %Modulator gain
T_m = 0.15                           %Modulator time constant
U   = 1;                             %Trigger output level

N=100;                               %Number of steps in the i for-loop
M=200;                               %Number of steps in the j for-loop

t_s=10;                              %Set the simulation time span
simopts = simset('solver','ode5',... %Use a ode5 solver with a step
    'FixedStep', 5e-3);              %length of 0.005

for i=1:1:(N+1)                      %Start the i for-loop
    i
    U_on=(1/N)*(i-1);                %Set the U_{on} value range
    U_on_values(i)=U_on;             %Create an array of the U_{on} values
    for j=1:1:(M+1)                  %Start the j for-loop
        h=(2/M)*(j-1);               %Set the h value range
        h_values(j)=h;               %Create an array of the h values
        sim('PWPF_test',t_s, simopts); %Run 'PWPF_test' for t_s second
                                     %with the parameters from simout
        Fuel_usage(i,j)=...          %Estimate fuel usage
            PWPF_performance5.signals...%
            .values(size(...         %
            PWPF_performance5.signals...%
            .values, 1));            %
    end                              %
end                                  %

figure(1)                            %
mesh(T_m_values,...                  %Create a 3D mesh of the fuel usage
```

```
    K_m_values,...                      %
    Fuel_usage);                        %
zlabel('Fuel usage',...                 %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('K_{m}',...                      %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
xlabel('T_{m}',...                      %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %
```

B.2.8 Figure ??

```
%Thruster_firing_K_m_vs_U_on.m
%
% Creates a 3D mesh and a contour plot of both the number of thruster
% firings and fuel usage for different values of K_m and U_on. All other
% variables are set prior to the simulation.
% N and M determines number of steps for the K_m and U_on loop, respectively
% (the plot resolution).

% Author: Trond Dagfinn Krøvel
% $Revision: 1.2 $  $Date: 2005/07/02 20:30 $

C   = 0.75;                             %Constant input value C
K_p = 1;                                %Pre-filter gain
T_m = 0.15;                             %Filter tim constant
h   = 0.3                               %Filter gain
U   = 1;                                %Modulator output value

N   = 100;                              %Number of steps in the i for-loop
M   = 100;                              %Number of steps in the j for-loop
t_s = 10;                               %Set the number of seconds the
                                        %simulation will go
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);                 %length of 0.005

for i=1:1:(N+1)                         %Start the i for-loop
    i                                   %
    K_m=(10/N)*(i-1);                   %Set the K_m value range
    K_m_values(i)=K_m;                  %Create an array of the K_m values
    for j=1:1:(M+1)                     %Start the j for-loop
        U_on=(1/M)*(j-1);               %Set the U_on value range
        U_on_values(j)=U_on;            %Create an array of the U_on values
        sim('PWPF_test',t_s, simopts);  %Simulate 'PWPF_test' for t_s seconds
                                        %with the parameters from simopts
```

```
        Thruster_firings(i,j)=...       %Create a matrix with the number of
            thruster_firing_counter(... %thruster firings
            PWPF_performance2.signals.values);
    end                                 %
end                                     %

mesh(U_on_values,...                    %Create a 3D thruster firing plot
    K_m_values,...                      %
    Thruster_firings);                  %
xlabel('U_{on}',...                     %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('K_{m}',...                      %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
zlabel('Number of thruster firings',... %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %
```

B.2.9 Figure 4.9

```
%Thruster_firing_plot_for_T_m_vs_f.m
%
% Creates a 3D mesh of the fuel usage for different values of T_m and f.
% All other variables are set prior to the simulation.
% N and M determines number of steps for the T_m and f loop, respectively.
%

% Author: Trond Dagfinn Krøvel
% $Revision: 1.2 $  $Date: 2005/07/02 22:00 $

clear all

K_p = 1;                                %Pre-modulator gain
K_m = 4.5;
U_on= 0.45;                             %Trigger on-value
h   = 0.3;                              %Trigger hysteresis width
U   = 1;                                %Trigger output level

N   = 100;                               %Number of steps in the i for-loop
M   = 100;                               %Number of steps in the j for-loop

t_s = 10;                               %Set the simulation time span
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);                 %length of 0.005

for i=1:1:N                             %Start the i for-loop
```

```
    i
    T_m=(1/N)*i;                        %Set the T_{m} value range
    T_m_values(i)=T_m;                  %Create an array of the T_{m} values
    for j=1:1:M                         %Start the j for-loop
        f=(150/M)*(j-1);                %Set the f value range
        f_values(j)=f;                  %Create an array of the f values
        sim('PWPF_test',t_s, simopts);  %Run 'PWPF_test' for t_s second
                                        %with the parameters from simout
        Thruster_firings(i,j)=...       %Count the number of thruster firings
            Thruster_firing_counter(... %
            PWPF_performance2...        %
            .signals.values);           %
    end                                 %
end                                     %

figure(1);                              %
mesh(f_values,...                       %Create a 3D thruster firing plot
    T_m_values,...                      %
    Thruster_firings);                  %
xlabel('Input frequency',...            %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('T_{m}',...                      %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
zlabel('Number of thruster firings',... %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %
```

B.2.10 Figure 4.10

```
%Fuel_usage_plot_for_T_m_vs_f.m
%
% Creates a 3D mesh of the fuel usage for different values of T_m and f.
% All other variables are set prior to the simulation.
% N and M determines number of steps for the T_m and f loop, respectively.
%

% Author: Trond Dagfinn Krøvel
% $Revision: 1.2 $  $Date: 2005/07/02 22:00 $

clear all

C   = 0.75;                             %Constant input value
K_p = 1;                                %Pre-modulator gain
K_m = 4.5;
U_on= 0.45;                             %Trigger on-value
```

```
h   = 0.3;                            %Trigger hysteresis width
U   = 1;                              %Trigger output level

N   = 100;                            %Number of steps in the i for-loop
M   = 100;                            %Number of steps in the j for-loop

t_s = 10;                             %Set the simulation time span
simopts = simset('solver','ode5',...  %Use a ode5 solver with a step
    'FixedStep', 5e-3);               %length of 0.005

for i=1:1:(N)                         %Start the i for-loop
    i
    T_m=(1/N)*(i);                    %Set the T_{m} value range
    T_m_values(i)=T_m;                %Create an array of the T_{m} values
    for j=1:1:M                       %Start the j for-loop
        f=(150/M)*(j);                %Set the f value range
        f_values(j)=f;                %Create an array of the f values
        sim('PWPF_test_fuel_usage',... %Run 'PWPF_test' for t_s second
            t_s, simopts);            %with the parameters from simout
        Fuel_usage(i,j)=...           %Estimate fuel usage
            PWPF_performance5.signals...%
            .values(size(...          %
            PWPF_performance5.signals...%
            .values, 1));             %
    end                               %
end                                   %

figure(1)                             %
mesh(f_values,...                     %Create a 3D mesh of the fuel usage
    T_m_values,...                    %
    Fuel_usage);                      %
xlabel('Input frequency',...          %Label the axis
    'FontName','Times New Roman',...  %
    'FontSize',14);                   %
ylabel('T_{m}',...                    %
    'FontName','Times New Roman',...  %
    'FontSize',14);                   %
zlabel('Fuel usage',...               %
    'FontName','Times New Roman',...  %
    'FontSize',14);                   %
axis tight;                           %
```

B.2.11 Figure 4.11 and 4.14

```
%Fuel_and_firing_plot_for_K_pre_vs_K_m.m
%
% Creates a 3D mesh of the number of thruster firings for different values
% of K_pre and K_m. All other variables are set prior to the simulation.
```

```
% N and M determines number of steps for the K_pre and K_m loop, respectively.
%

% Author: Trond Dagfinn Krøvel
% $Revision: 1.2 $  $Date: 2005/07/06 19:30 $

clear all

C       = 15;                           %Reference input value

K_p     = 1.5;                          %Proportional gain
K_d     = 7;                            %Derivative gain
K_i     = 0;                            %Integral gain

T_m     = 0.15;                         %Filter time constant
U_on    = 0.45;                         %Trigger on-value
h       = 0.3;                          %Trigger hysteresis width
U       = 1;                            %Trigger output level

I       = 2;                            %Satellite inertia

N   = 100;                              %Number of steps in the i for-loop
M   = 100;                              %Number of steps in the j for-loop

t_s = 75;                               %Set the simulation time span
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);                 %length of 0.005

for i=1:1:(N+1)                         %Start the i for-loop
    i
    K_pre=(10/N)*(i-1);                 %Set the K_{pre} value range
    K_pre_values(i)=K_pre;              %Create an array of the K_{pre} values
    for j=1:1:(M+1)                     %Start the j for-loop
        K_m=(10/M)*(j-1);               %Set the K_{m} value range
        K_m_values(j)=K_m;              %Create an array of the K_{m} values
        sim('PWPF_test_with_PD',...     %Run 'PWPF_test_whit_PD' for t_s
            t_s, simopts);              %second with the parameters from
                                        %simopts
        Thruter_firings(i,j)=...        %Count the number of thruster firings
            Thruster_firing_counter(... %
            PWPF_performance2...        %
            .signals.values);           %
        Fuel_usage(i,j)=...             %Estimate fuel usage
            PWPF_performance5.signals...%
            .values(size(...            %
            PWPF_performance5.signals...%
            .values, 1));               %
```

```
    end                                 %
end                                     %

figure(1)                               %
mesh(K_m_values,...                     %Create a 3D mesh of the number of
    K_pre_values,...                    %thruster firings
    Thruter_firings);                   %
xlabel('K_{m}',...                      %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('K_{pre}',...                    %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
zlabel('Number of thruster firings',... %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %

figure(2)                               %
mesh(K_m_values,...                     %Create a 3D mesh of the number of
    K_pre_values,...                    %thruster firings
    Fuel_usage);                        %
xlabel('K_{m}',...                      %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('K_{pre}',...                    %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
zlabel('Fuel usage',...                 %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %
```

B.2.12 Figure 4.12 and 4.15

```
%Fuel_and_firing_plot_for_K_m_vs_T_m.m
%
% Creates a 3D mesh of the number of thruster firings for different values
% of K_m and T_m. All other variables are set prior to the simulation.
% N and M determines number of steps for the K_m and T_m loop, respectively.
%

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/07/01 19:30 $

clear all

C       = 15;                           %Reference input value
```

```
K_p     = 1.5;                          %Proportional gain
K_d     = 7;                            %Derivative gain
K_i     = 0;                            %Integral gain

K_pre   = 1;                            %Pre-modulator gain
U_on    = 0.45;                         %Trigger on-value
h       = 0.3;                          %Trigger hysteresis width
U       = 1;                            %Trigger output level

I       = 2;                            %Satellite inertia

N   = 100;                              %Number of steps in the i for-loop
M   = 100;                              %Number of steps in the j for-loop

t_s = 75;                               %Set the simulation time span
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);                 %length of 0.005

for i=2:1:(N+1)                         %Start the i for-loop
    i
    K_m=(10/N)*(i-1);                   %Set the K_{m} value range
    K_m_values(i)=K_m;                  %Create an array of the K_{m} values
    for j=1:1:M                         %Start the j for-loop
        T_m=(1/M)*(j);                  %Set the T_{m} value range
        T_m_values(j)=T_m;              %Create an array of the T_{m} values
        sim('PWPF_test_with_PD',...     %Run 'PWPF_test_with_PD' for t_s
            t_s, simopts);              %seconds with the parameters from
                                        %simopts
        Thruter_firings(i,j)=...        %Count the number of thruster firings
            Thruster_firing_counter(... %
            PWPF_performance2...        %
            .signals.values);           %
        Fuel_usage(i,j)=...             %Estimate fuel usage
            PWPF_performance5.signals...%
            .values(size(...            %
            PWPF_performance5.signals...%
            .values, 1));               %
    end                                 %
end                                     %

figure(1)                               %
mesh(T_m_values,...                     %Create a 3D mesh of the number of
    K_m_values,...                      %thruster firings
    Thruter_firings);                   %
xlabel('T_{m}',...                      %Label the axis
    'FontName','Times New Roman',...    %
```

```
    'FontSize',14);                     %
ylabel('K_{m}',...                      %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
zlabel('Number of thruster firings',... %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %

figure(2)                               %
mesh(T_m_values,...                     %Create a 3D mesh of the number of
    K_m_values,...                      %thruster firings
    Fuel_usage);                        %
xlabel('T_{m}',...                      %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('K_{m}',...                      %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
zlabel('Fuel usage',...                 %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %
```

B.2.13 Figure 4.13 and 4.16

```
%Fuel_and_firing_plot_for_U_on_vs_h.m
%
% Creates a 3D mesh of the number of thruster firings for different values
% of U_on and h. All other variables are set prior to the simulation.
% N and M determines number of steps for the U_on and h loop, respectively.
%

% Author: Trond Dagfinn Krøvel
% $Revision: 1.1 $  $Date: 2005/07/01 19:30 $

clear all

C       = 15;                           %Constant input value

K_p     = 1.5;                          %Proportional gain
K_d     = 7;                            %Derivative gain
K_i     = 0;                            %Integral gain

K_pre   = 1;                            %Pre-modulator gain
K_m     = 4.5;                          %Filter gain
T_m     = 0.15;                         %Filter time constant
U       = 1;                            %Trigger output level
```

```
I       = 2;                            %Satellite inertia

N   = 100;                              %Number of steps in the i for-loop
M   = 100;                              %Number of steps in the j for-loop

t_s = 75;                               %Set the simulation time span
simopts = simset('solver','ode5',...    %Use a ode5 solver with a step
    'FixedStep', 5e-3);                 %length of 0.005

for i=1:1:(N+1)                         %Start the i for-loop
    i
    U_on=(1/N)*(i-1);                   %Set the U_{on} value range
    U_on_values(i)=U_on;                %Create an array of the U_{on} values
    for j=1:1:(M+1)                     %Start the j for-loop
        h=(2/M)*(j-1);                  %Set the h value range
        h_values(j)=h;                  %Create an array of the h values
        sim('PWPF_test_with_PD',...     %Run 'PWPF_test_with_PD' for t_s
            t_s, simopts);              %seconds with the parameters from
                                        %simopts
        Thruter_firings(i,j)=...        %Count the number of thruster firings
            Thruster_firing_counter(... %
            PWPF_performance2...        %
            .signals.values);           %
        Fuel_usage(i,j)=...             %Estimate fuel usage
            PWPF_performance5.signals...%
            .values(size(...            %
            PWPF_performance5.signals...%
            .values, 1));               %
    end                                 %
end                                     %

figure(1)                               %
mesh(h_values,...                       %Create a 3D mesh of the number of
    U_on_values,...                     %thruster firings
    Thruter_firings);                   %
xlabel('h',...                          %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('U_{on}',...                     %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
zlabel('Number of thruster firings',... %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %
```

```
figure(2)                               %
mesh(h_values,...                       %Create a 3D mesh of the fuel usage
    U_on_values,...                     %
    Fuel_usage);                        %
xlabel('h',...                          %Label the axis
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('U_{on}',...                     %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
zlabel('Fuel usage',...                 %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
axis tight;                             %
```

B.3 Code for Chapter 5

In this section, the code used in chapter 5 is presented.

B.3.1 Figure ??

B.4 Optimal PWPF Parameter simulations

B.4.1 Optimal PD Controller Simulation

```
%ESMO_PD_single_run.m
% Simulate a X degree slew maneuver with ESMO, using the optimal PWPF
% parameters.

% Author: Trond Dagfinn Krøvel
% $Revision: 1.0 $  $Date: 2005/07/07 21:00 $

clear all;                              %

init;                                   %

K_pre   = 20;                           %
K_m     = 4.3;                          %
T_m     = 0.35;                         %
U_on    = 0.075;                        %
h       = 0.085;                        %

t_s = 2000;                             %Set the simulation time span
simopts = simset('solver','ode1',...    %Use a ode5 solver with a step
    'FixedStep', 1e-2);                 %length of 0.005
```

```
sim('ESMO_sytem_model_PD_nice_clean',...%Run 'PWPF_test_whit_PD' for t_s
    t_s, simopts);                      %second with the parameters from simopts

Thruster_firings =...                   %
    Thruster_firing_counter(PWPF_performance21.signals.values)+...
    Thruster_firing_counter(PWPF_performance22.signals.values)+...
    Thruster_firing_counter(PWPF_performance23.signals.values);

Fuel_usage =...                         %
    PWPF_performance5.signals.values(size(PWPF_performance5.signals.values,1));

figure(1)                               %
subplot(3,1,1); hold on;                %
plot(simout.time,...                    %
    simout.signals.values(:,5), 'r');   %
plot(simout.time,...                    %
    simout.signals.values(:,6), 'b');   %
plot(simout.time,...                    %
    simout.signals.values(:,7), 'g');   %
plot(simout.time,...                    %
    simout.signals.values(:,8), 'k');   %
title('ESMO behavior with PD controller',...
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Angle [deg]',...                %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylim([29.5 30.5]);                      %
box on;                                 %

subplot(3,1,2); hold on;                %
title('Thruster activity',...           %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Torque [Nm]',...                %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
plot(simout.time,...                    %
    simout.signals.values(:,2), 'r');   %
plot(simout.time,...                    %
    simout.signals.values(:,3), 'b');   %
plot(simout.time,...                    %
    simout.signals.values(:,4), 'g');   %
box on;                                 %

subplot(3,1,3); hold on;                %
```

```
plot(simout.time,...                    %
    simout.signals.values(:,1), 'r');   %
title('Relative fuel consumption',...   %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
xlabel('Time [sec]',...                 %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Fuel usage',...                 %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylim([0.5 0.6]);                        %
box on;                                 %
hold off;                               %

Fuel_usage                              %
Thruster_firings                        %
```

B.4.2 Optimal SM Controller Simulation

```
%ESMO_SM_single_run.m
% Simulate a X degree slew maneuver with ESMO, using the optimal PWPF
% parameters.

% Author: Trond Dagfinn Krøvel
% $Revision: 1.0 $  $Date: 2005/07/07 21:00 $

clear all;                              %

init;                                   %

K_pre   = 40;                           %Pre-modulator gain
K_m     = 3;                            %Filter gain
T_m     = 0.2;                          %Filter time constant
U_on    = 0.19;                         %Schmitt trigger on-value
h       = 0.15;                         %Schmitt trigger hysteresis width

t_s = 400;                              %Set the simulation time span
simopts = simset('solver','ode1',...    %Use a ode5 solver with a step
    'FixedStep', 1e-2);                 %length of 0.005

sim('ESMO_sytem_model_SM_nice_clean',...%Run 'PWPF_test_whit_PD' for t_s
    t_s, simopts);                      %second with the parameters from simopts

Thruster_firings =...                   %Count the number of thruster firings
    Thruster_firing_counter(PWPF_performance21.signals.values)+...
    Thruster_firing_counter(PWPF_performance22.signals.values)+...
    Thruster_firing_counter(PWPF_performance23.signals.values);
```

```
Fuel_usage =...                         %Estimate the fuel usage
    PWPF_performance5.signals.values(size(PWPF_performance5.signals.values,1));

figure(1)                               %
subplot(3,1,1); hold on;                %
plot(simout.time,...                    %Plot the euler angles vs the
    simout.signals.values(:,5), 'r');   %reference signal
plot(simout.time,...                    %
    simout.signals.values(:,6), 'b');   %
plot(simout.time,...                    %
    simout.signals.values(:,7), 'g');   %
plot(simout.time,...                    %
    simout.signals.values(:,8), 'k');   %
title('ESMO behavior with PD controller',...
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Angle [deg]',...                %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylim([29.5 30.5]);                      %
box on;                                 %

subplot(3,1,2); hold on;                %
title('Thruster activity',...           %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Torque [Nm]',...                %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
plot(simout.time,...                    %
    simout.signals.values(:,2), 'r');   %
plot(simout.time,...                    %
    simout.signals.values(:,3), 'b');   %
plot(simout.time,...                    %
    simout.signals.values(:,4), 'g');   %
box on;                                 %

subplot(3,1,3); hold on;                %
plot(simout.time,...                    %
    simout.signals.values(:,1), 'r');   %
title('Relative fuel consumption',...   %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
xlabel('Time [sec]',...                 %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
```

```
ylabel('Fuel usage',...                %
    'FontName','Times New Roman',...   %
    'FontSize',14);                    %
ylim([0.35 0.45]);                     %
box on;                                %
hold off;                              %

Fuel_usage                             %
Thruster_firings                       %
```

B.4.3 Optimal L3 Controller Simulation

```
%ESMO_L3_single_run.m
% Simulate a X degree slew maneuver with ESMO, using the optimal PWPF
% parameters.

% Author: Trond Dagfinn Krøvel
% $Revision: 1.0 $  $Date: 2005/07/07 21:00 $

clear all;                             %

init;                                  %

K_pre   = 46                           %
K_m     = 4.0                          %
T_m     = 0.25                         %
U_on    = 0.16                         %
h       = 0.19                         %

t_s = 700;                             %Set the simulation time span
simopts = simset('solver','ode1',...   %Use a ode5 solver with a step
    'FixedStep', 1e-2);                %length of 0.005

sim('ESMO_system_model_L3_nice_ribbet',...%Run 'PWPF_test_whit_PD' for t_s
    t_s, simopts);                     %second with the parameters from simopts

Thruster_firings =...                  %
    Thruster_firing_counter(PWPF_performance21.signals.values)+...
    Thruster_firing_counter(PWPF_performance22.signals.values)+...
    Thruster_firing_counter(PWPF_performance23.signals.values);

Fuel_usage =...                        %
    PWPF_performance5.signals.values(size(PWPF_performance5.signals.values,1));

figure(1)                              %
subplot(3,1,1); hold on;               %
plot(simout.time,...                   %
    simout.signals.values(:,5), 'r');  %
```

```
plot(simout.time,...                        %
    simout.signals.values(:,6), 'b');   %
plot(simout.time,...                        %
    simout.signals.values(:,7), 'g');   %
plot(simout.time,...                        %
    simout.signals.values(:,8), 'k');   %
title('ESMO behavior with PD controller',...
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Angle [deg]',...                    %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylim([29.5 30.5]);                          %
box on;                                     %

subplot(3,1,2); hold on;                    %
title('Thruster activity',...               %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Torque [Nm]',...                    %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
plot(simout.time,...                        %
    simout.signals.values(:,2), 'r');   %
plot(simout.time,...                        %
    simout.signals.values(:,3), 'b');   %
plot(simout.time,...                        %
    simout.signals.values(:,4), 'g');   %
box on;                                     %

subplot(3,1,3); hold on;                    %
plot(simout.time,...                        %
    simout.signals.values(:,1), 'r');   %
title('Relative fuel consumption',...   %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
xlabel('Time [sec]',...                     %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylabel('Fuel usage',...                     %
    'FontName','Times New Roman',...    %
    'FontSize',14);                     %
ylim([0.5 0.6]);                            %
box on;                                     %
hold off;                                   %

Fuel_usage                                  %
```

```
Thruster_firings                              %
```

Appendix C

Simulink Figures

In the following chapter, the MATLAB/Simulink models used in the thesis is presented.

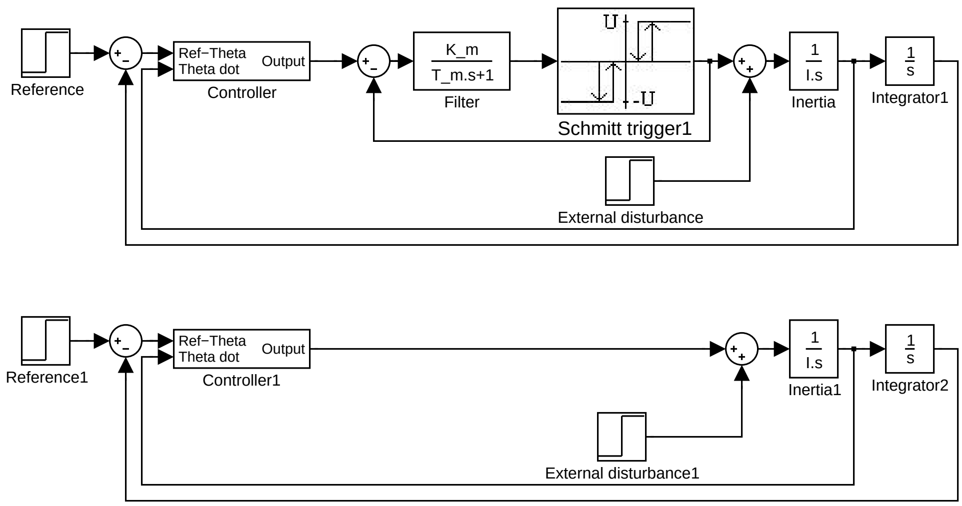


Figure C.1: The Simulink test setup used to create Figure 3.4. On top, the PWPF modulator system and below, the linear actuator system. The PWPF modulator system was also used to create Figure 4.11 to Figure 4.16.

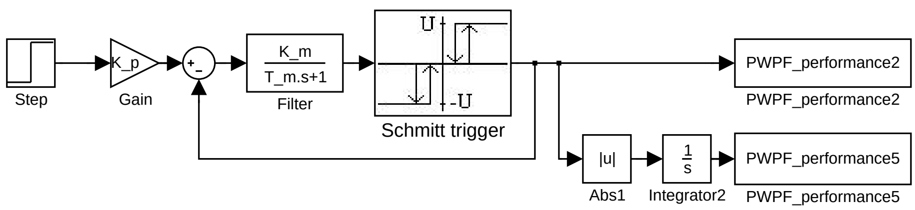


Figure C.2: The Simulink model used to create the simulation plots in Figure 4.5 to 4.8.

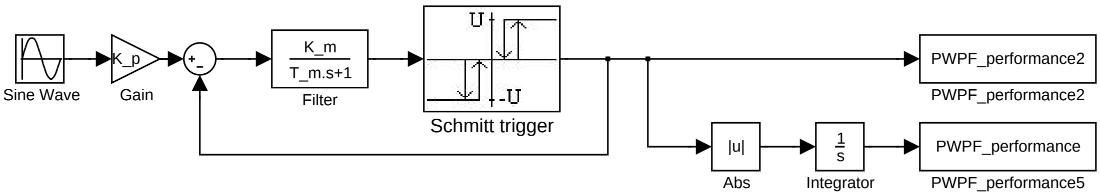


Figure C.3: The Simulink model used to create Figure 4.9 and 4.10.

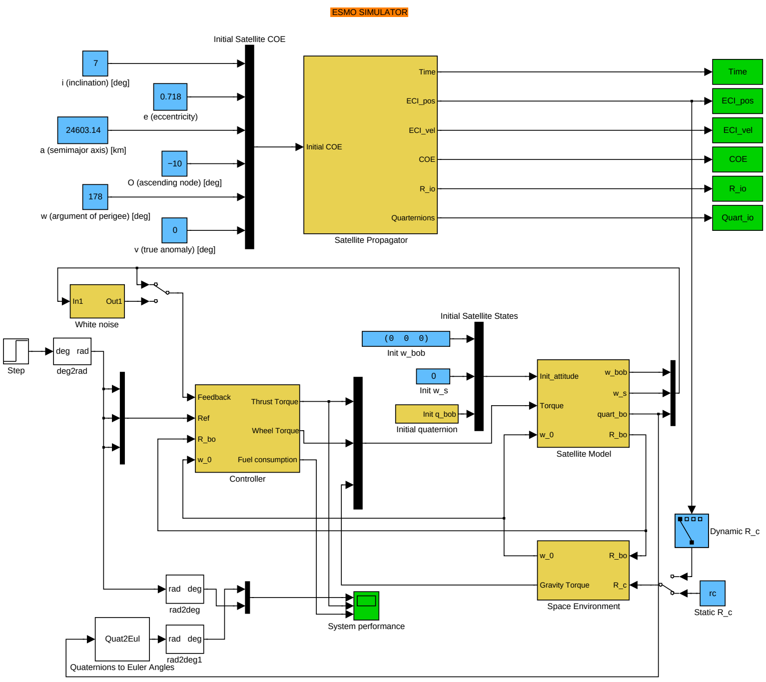


Figure C.4: The Simulink model of ESMO used in Chapter 5

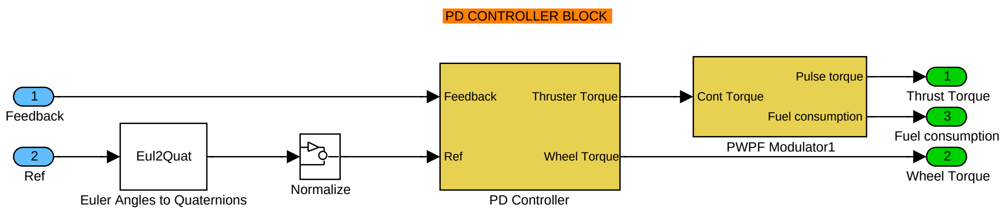


Figure C.5: The PD controller model for ESMO.

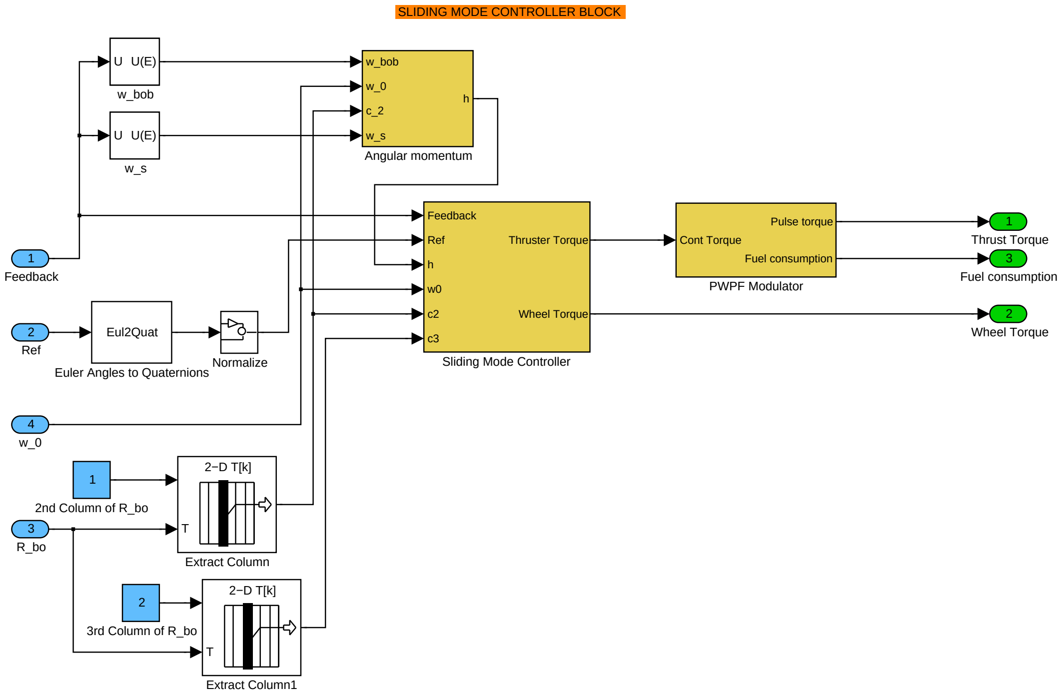


Figure C.6: The Sliding Mode controller model for ESMO.

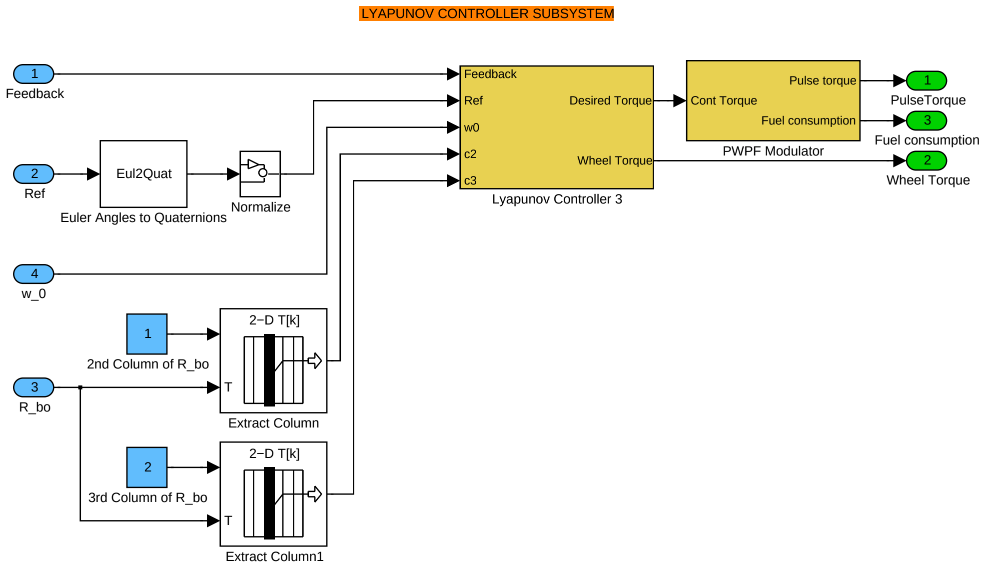


Figure C.7: The Lyapunov controller model for ESMO.

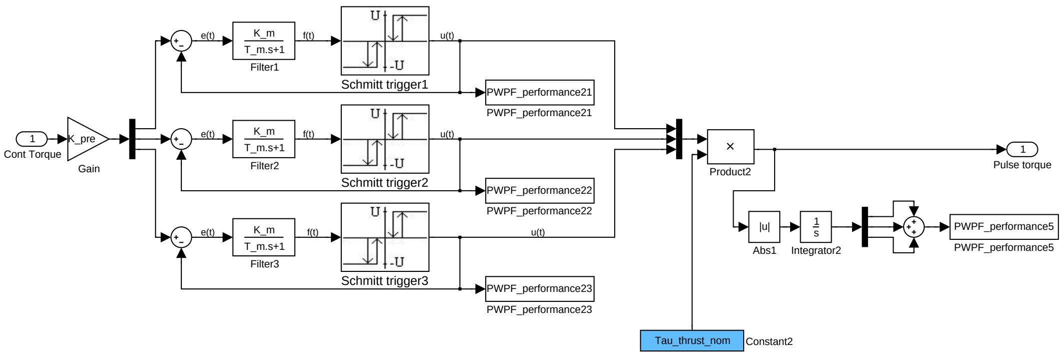


Figure C.8: The PWPF modulator used in the ESMO model, implemented in Simulink

Appendix D

Simulation Plots

In the following chapter, a series of simulation plots are presented. These plots are mainly behavior plots for three parameters tested against each other, and are included to give a better understanding of parameter choices, modulator behavior etc. Explanations for the plots are found in the caption of each figure.

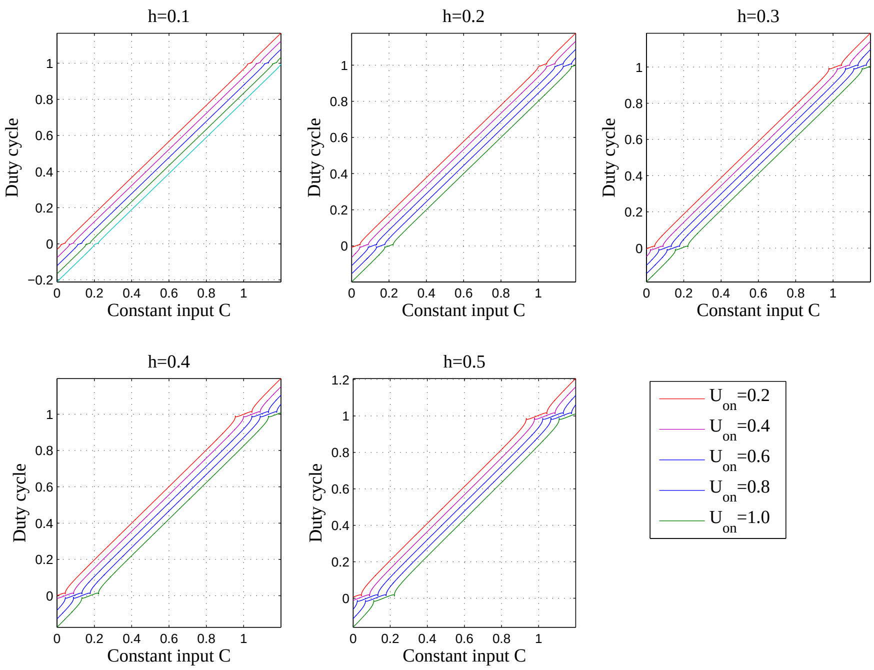


Figure D.1: The effect on linearity for a varying U_{on}, with different, fixed, non-zero h. Notice the increasing nonlinear area for duty cycle of zero and one for increasing h.

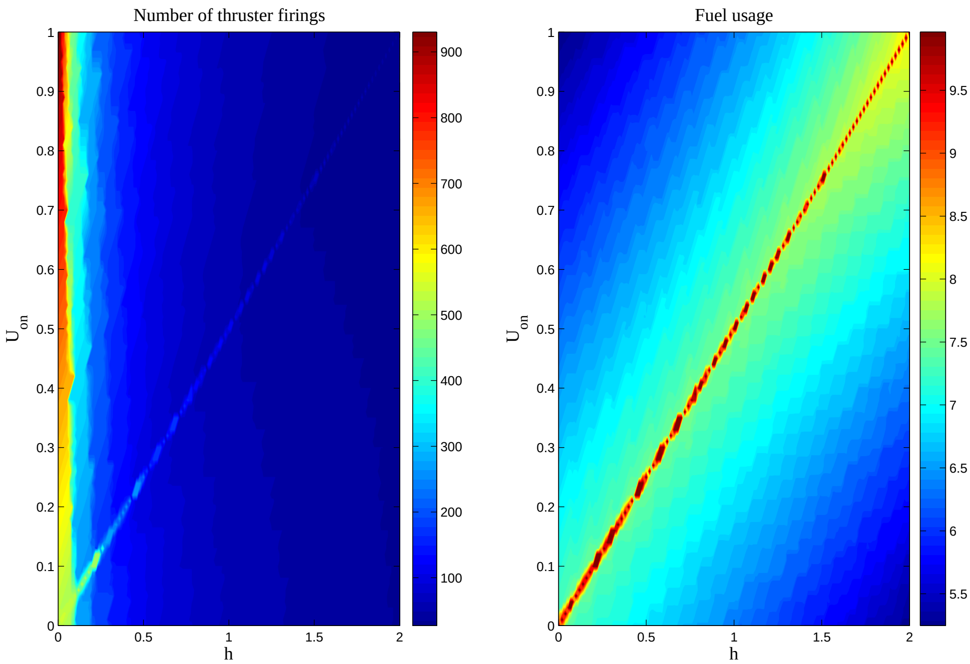


Figure D.2: A comparison between the fuel usage and number of thruster firings for varying U_{on} and h. Notice the line crossing the plot for $U_{on} = h/2$, which is due to the Schmitt trigger characteristics.

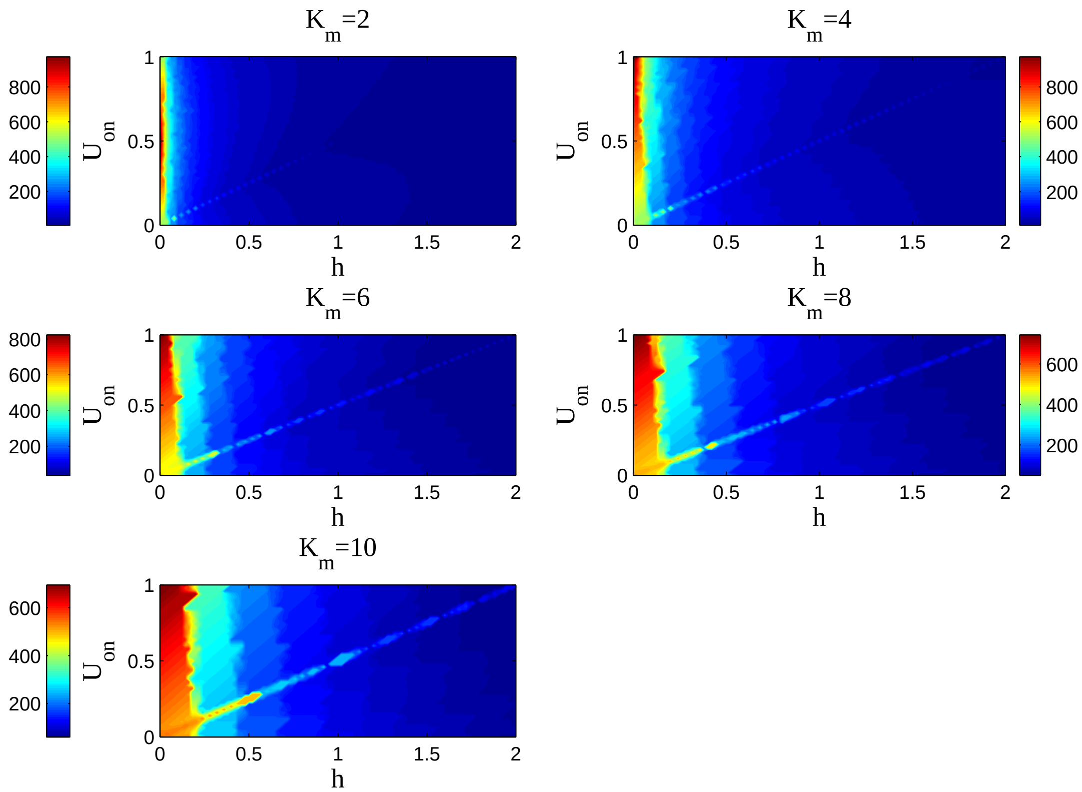


Figure D.3: A comparison between different K_m's for varying U_{on} and h. Please notice that the color scale has two different ranges.

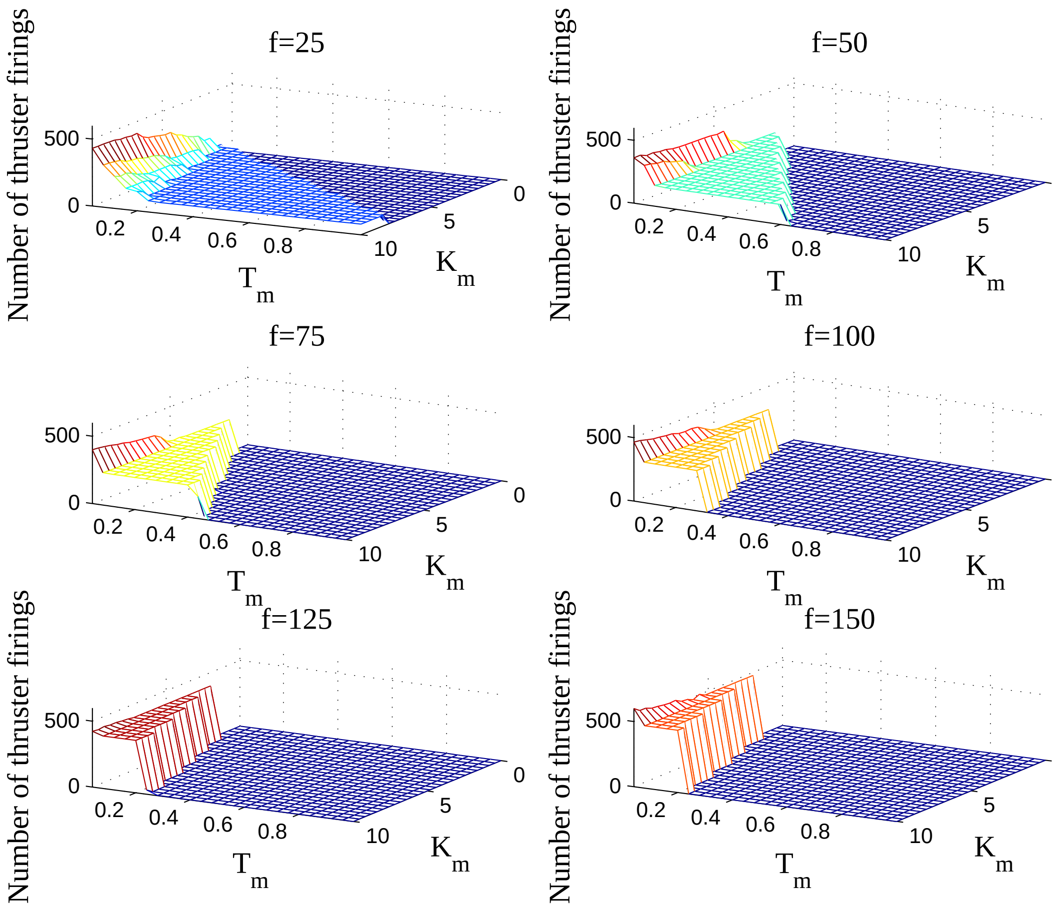


Figure D.4: A thruster firing comparison for different f's with varying K_m and T_m.

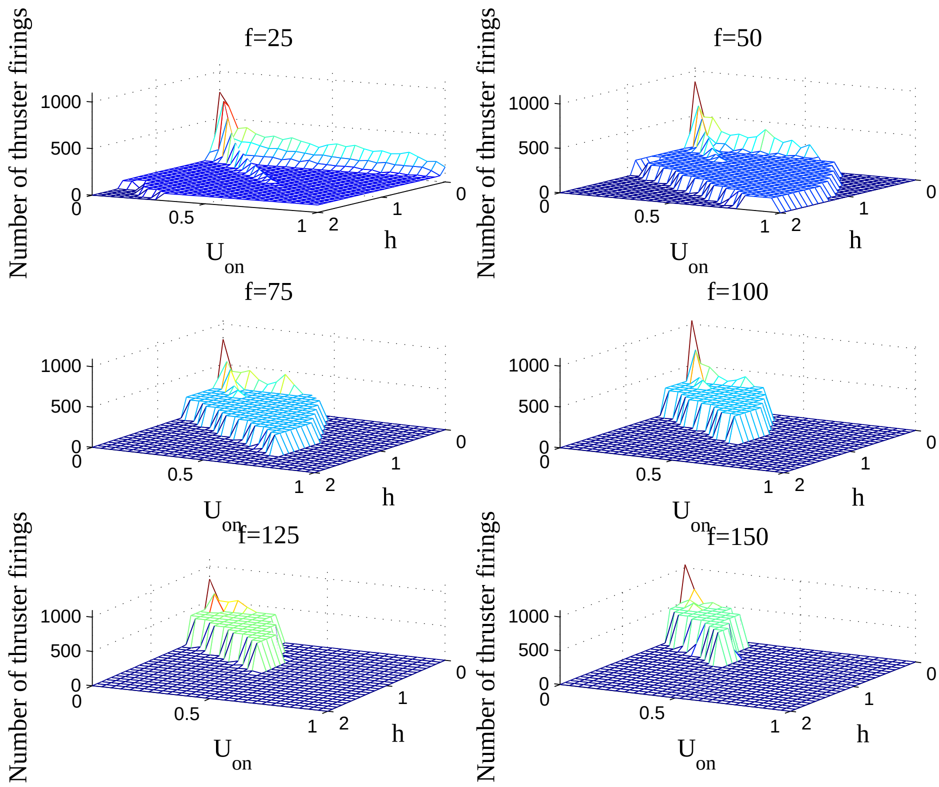


Figure D.5: A thruster firing comparison for different f's with varying U_{on} and h.

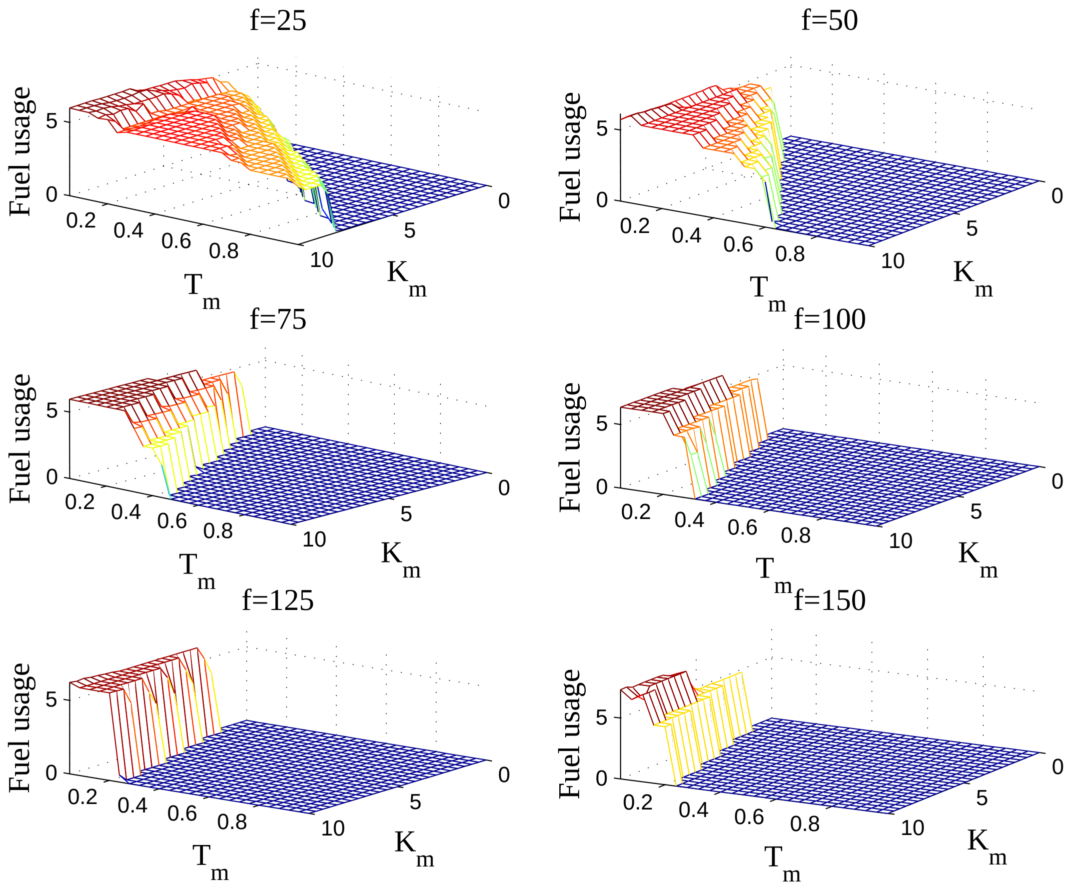


Figure D.6: A fuel usage comparison for different f's with varying T_m and K_m.

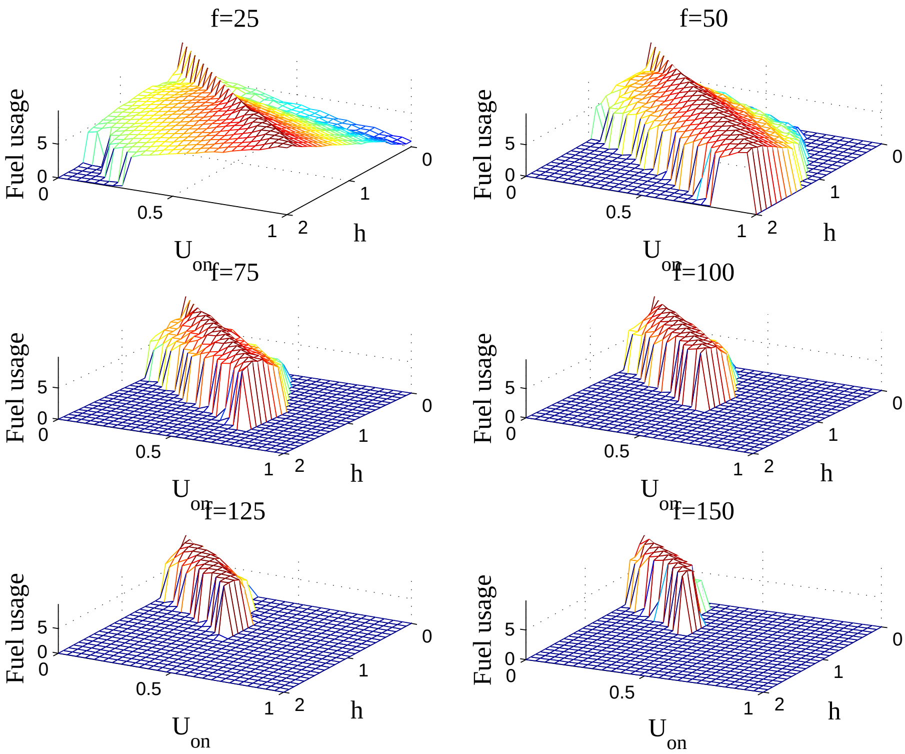


Figure D.7: A fuel usage comparison for different f's with varying U_{on} and h.

Appendix E

Presentations

Pulse Width-Pulse Frequency Modulation of Thrusters for the Micro-Satellite SSETI/ESMO

T. D. Krøvel[1], J. T. Gravdahl[2]
Department of Engineering Cybernetics, NTNU, Trondheim, Norway.
O. S. Bragstads Plass 2D, 7491 Trondheim, Norway
Phone: +47 93005368
Email: tronddag@stud.ntnu.no

ABSTRACT

In this project we have investigated modulation of controller signals for attitude control thrusters. Our main result is that pulse width-pulse frequency (PWPF) modulation holds a considerable advantage over bang-bang controller systems.

In the satellite industry, demands for lower fuel consumption as well as higher pointing accuracy, has forced researchers to develop better methods of controlling satellite thrusters ever since the first advanced space mission.

This has lead to research on several different methods to “translate” the controller output signal to an on/off signal that can be fed directly to a thruster system. The different methods have their own advantages, and can be used in different situations. But it is of course advantageous to have methods that performs close to optimal for all kinds of systems and configurations.

Pulse-width pulse-frequency (PWPF) modulation of the controller signal is the most commonly implemented thruster control scheme. This method is based on fairly old technology, but still stands strong today. A PWPF modulator consists of a first order lag filter, a Schmitt trigger and a feedback loop. This system modulates both the pulse width and the pulse frequency, and gives a quasi-linear operation of the thrusters.

A bang-bang controller system and a PWPF modulation system were compared. The PWPF modulator and the bang-bang controller were implemented in the model of the European Student Earth Orbiter (ESEO) satellite, for a simulation comparison. The ESEO is chosen because the physical parameters for the European Student Moon Orbiter (ESMO) still not are decided. The physical parameters of ESMO are believed to be similar to those of ESEO.

The simulations were divided in two parts, where one part consisted of simple simulations and the other with noisy simulations. The simple simulations were made with zero initial conditions, static simulation environment and no noise. The noisy simulations were made with fairly aggressive initial conditions, dynamic satellite environment and induced noise. The simulations were done for both linear and nonlinear controllers.

The pointing accuracy was determined by examining the Euler angels, and fuel consumption was determined by looking at the PWPF modulator and bang-bang controller output.

All simulations with PWPF modulation of the thruster control signal were compared with similar bang-bang control systems. The results showed both equal or better pointing accuracy and lower fuel consumption for PWPF modulated systems, regardless of regulator and initial conditions.

Further work on this topic will include studies where we look deeper into how the different parameters of the PWPF modulator affect performance. We will also attempt to find optimal parameters with respect to fuel consumption.

[1] Student
[2] Associate Professor

Pulse-Width Pulse-Frequency Modulation of Thrusters for the Micro-Satellite SSETI-ESMO

Trond Dagfinn Krøvel

Department of Engineering Cybernetics,
Norwegian University of Science and Technology,
N-7491 Trondheim
Norway

Outline of presentation

- Introduction to SSETI
- Background
- Modulation techniques
- Implementation and testing
- Simulations of performance

Introduction to SSETI

- The Student Space Exploration and Technology Initiative (SSETI) was started by ESA in 2000
  - To create a network of students, educational institutions and organisations (on the internet) to facilitate the distributed design, construction and launch of (micro-) satellites and potentially more complex projects such as a moon-lander
- Participation from 25 universities throughout Europe, divided on 42 teams
- All teams are working in close collaboration with the Education Department of ESA

Introduction to SSETI

- Three ongoing projects
  - Express - Launch mid 2005
  - European Student Earth Orbiter (ESEO) - Launch late 2007
  - European Student Moon Orbiter (ESMO) - Launch 2010
- Future projects
  - European Student Moon Rover (ESMR)
  - Continued collaboration with European universities is desired, new project will come

Background

- On previous ESEO MATLAB/Simulink models (Topland 2004) a bang-bang regulator was used

$$u(t) = \begin{cases} U sgn(e(t)) & \text{if} \quad |e(t)| > 0 \\ 0 & \text{if} \quad e(t) = 0 \end{cases}$$

- This causes excessive thruster firings and high fuel consumption. It also creates unnecessary vibration in flexible structures

Background

- A more advanced bang-bang controller with a deadzone was implemented later

$$u(t) = \begin{cases} U sgn(e(t)) & \text{if} \quad |e(t)| \geq \alpha \\ 0 & \text{if} \quad |e(t)| < \alpha \end{cases}$$

- This controller gave less thruster firings and hence less fuel consumption
- But the controller was still very simple, and an in depth investigation was started

Modulation techniques

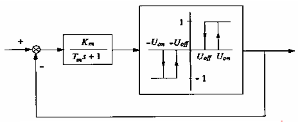
- In the literature, several modulation techniques have been proposed
  - Pulse-Width, Pulse-Width Pulse-Frequency and Derived Rate modulator
- All aims to give better performance and lower consumption.

Pulse-Width Pulse-Frequency

- With reference to previous work, an investigation of Pulse-Width Pulse-Frequency (PWPF) modulators was started
- The basics is easily available through the literature
- But little work about PWPF modulation has been published

The PWPF-modulator

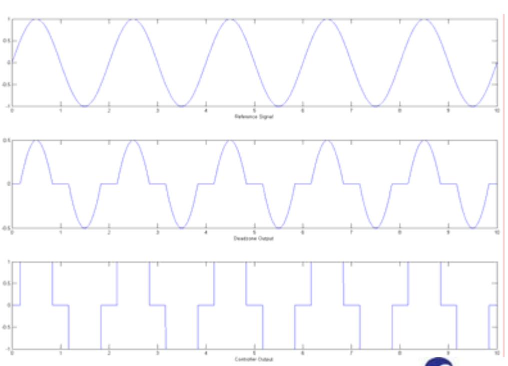
- The PWPF-modulator consists of a Schmitt-trigger, a first order lag filter and a feedback loop

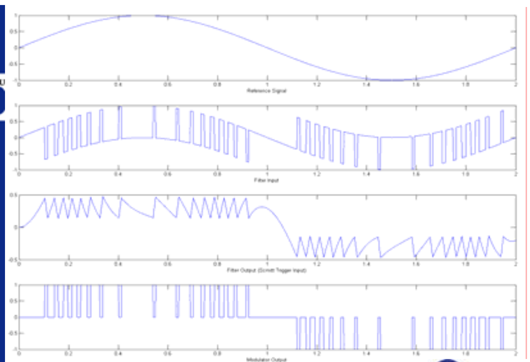


$$e_f(t) = (\theta_{cmd} - u(t))\frac{K_m}{T_m s + 1}$$

Comparison

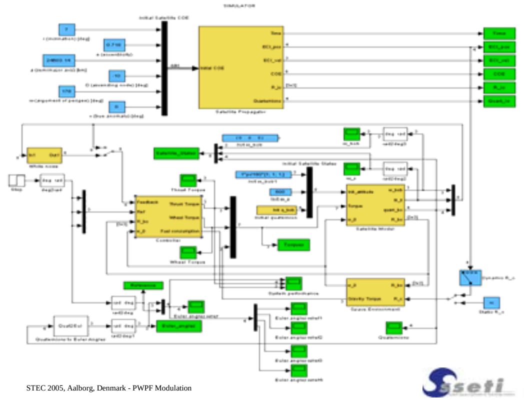
- A test system was made to see the differences between a bang-bang regulator and a PWPF modulator as well as to control the functionality of a developed PWPF-modulator code
- The system was made in MATLAB/Simulink

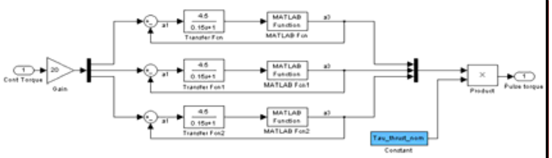




ESMO system test

- A MATLAB/Simulink model of ESEO was used, since no data for ESMO was available





Parameters

- The most difficult aspect of using PWPF modulators is parameter tuning
- There are five parameters
  - K_m: Filter gain
  - T_m: Filter time constant
  - U_on: Schmitt trigger on value
  - U_off: Schmitt trigger off value
  - K_p: Pre-modulator gain

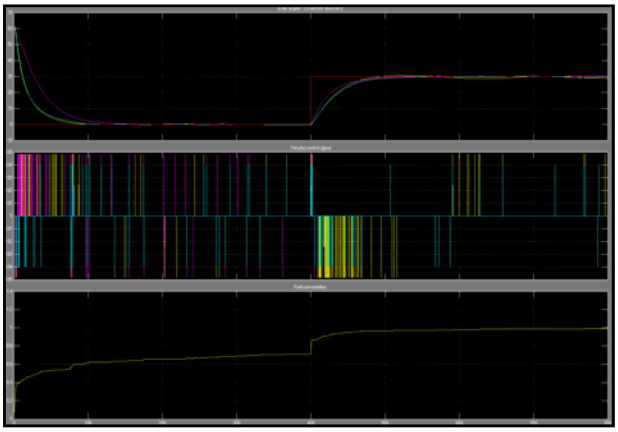
Parameters

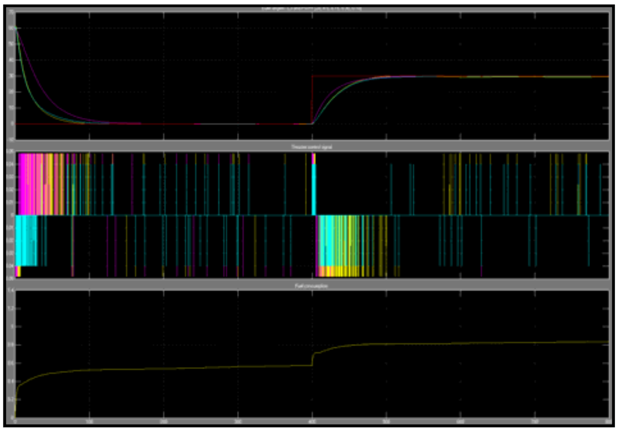
- A parameter recommendation has been given i the literature

Parameters	Analysis type Static	Dynamic	Slew	Recommendations
K_p	< 10.0	N/A	≥ 2.0	$2.0 - 10.0$
K_m	< 6.0	N/A	≥ 1.0	$1.0 - 6.0$
T_m	N/A	$0.1 - 0.2$	$0.1 - 0.2$	$0.1 - 0.2$
U_{on}	> 0.3	N/A	N/A	> 0.3
$U_{on} - U_{off}$	$< 0.8(U_{on} - U_{off})$	N/A	N/A	$< 0.8(U_{on} - U_{off})$

Simulations

- A series of simulations were performed, with different regulators and initial conditions, for both the PWPF modulator and the bang-bang controller





Results

- For all simulations made, it was possible to tune the PWPF modulator to have both higher pointing accuracy and lower fuel consumption
- This lead to a continued investigation of the properties of the PWPF modulator

Ongoing work

- A deeper study of how each parameter affects the performance of the PWPF modulator
- How to optimize the fuel consumption by manipulating the PWPF parameters
- Simulation tests

References

- Topland, M. P. (2004), Nonlinear attitude control of the micro-satellite eseo, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Literature

- McClelland, R. S. (1994), Spacecraft attitude control system performance using pulse width-pulse frequency modulated thrusters, Master's thesis, Naval Postgraduate School, California, USA.
- Buck, N. (1996), Minimum vibration manoeuvres using input shaping and pulse width-pulse frequency modulated thruster control, Master's thesis, Naval Postgraduate School, California, USA.
- Sidi, M. J. (1997), Spacecraft Dynamics and Control, Cambridge Aerospace Series, Cambridge University Press.
- Wie, B. (1998), Space Vehicle Dynamics and Control, AIAA Education Series, AIAA.