POLITECNICO DI TORINO

Corso di Laurea in Mechatronic Engineering

Tesi di Laurea

Robust identification and control for sea wave energy conversion

Relatori
Prof. Vito Cerone
Prof. Diego Regruto Tomalino
Prof. Massimo Canale

Candidato
Giuseppe Casulli

A.A. 2018/2019

To my Family

Sommario

La tesi si occupa nella progettazione di un modello orientato al controllo per sistema Inertial Sea Wave Energy Converter (ISWEC). Il sistema ISWEC consiste in un dispostivo galleggiante composto da uno scafo che ospita un sistema di giroscopi il quale il sistema è indotto da oscillazioni dello scafo galleggiante sul mare. L'estrazione di energia è ottenuta dallo smorzamento del moto di precessione del giroscopio attraverso un unita elettrico di potenza (PTO), implementata da un motore capace di esercitare una coppia indotta nello smorzamento del moto. L'albero, al quale opera il moto di precessione, è meccanicamente conneso all'albero del PTO. In questa tesi, una procedura basata sulla tecnica dell'identificazione del set-membership è proposta per la progettazione di un modello multi-input di un sistema con dati input-output sperimentali. Dal momento che la tecnica del Set-membership permette di quantificare l'incertezza di un modello identificato, la descrizione del sistema ottenuto può essere utilizzato per progettare un controllo robusto per massimizzare la conversione dell'energia. Basata su un modello indetificato orientato al controllo, uno studio preliminare è stato condotto per ricercare la possibilità di una formulazione del problema di massimizzazione della conversione di energia nei termini di un controllo robusto su anello chiuso.

Abstract

This thesis aims at designing a control-oriented model for the Inertial Sea Wave Energy Converter (ISWEC) system. The ISWEC system consists of a floating device composed by a hull which hosts a gyroscope system whose motion is induced by the pitching oscillations of the hull floating on the sea surface. Energy extraction is obtained by damping the gyroscope precession motion through an electrical Power Take Off (PTO) unit, which is implemented by an electric motor able to exert the torque intended for damping such a motion. The shaft, about which the precession motion takes place, is mechanically connected to the PTO shaft. In this thesis, a procedure based on set-membership identification techniques is proposed for building a multi-inputs linear model of the system from input-output experimental data. Since set-membership techniques allow the user to quantify the uncertainty of the identified model, the obtained system description can be profitably exploited to design a robust controller aimed at maximizing the energy conversion. Based on the identified control-oriented model, a preliminary study has been conducted in order to investigate the possibility of formulating the problem of maximizing the energy conversion in terms of robust closed loop control.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ocean wave energy is one of the most promising renewable energy source and realizes a significant contribution in the world energy mix, as discussed in [1]. The ocean owns different energy sources such as wave motion, current tide, salinity gradient, temperature differential and offshore wind. The wave motion is the ocean energy source topic of this thesis work, as it realizes the excitation input of the considered system to control. Moreover, wave energy is one of the densest since it allows yearly extraction of an average power density up to 80kW per meter of shoreline. Energy harvesting from wave motion has been topic of many research activities in the world since the oil crisis in the seventies. In the last twenty years the main purpose of such activities has been developing suitable devices capable to effectively produce energy facing issues such as reliability and efficiency, in order to achieve effective performance in terms of energy production cost. Nowadays different Wave Energy Converter (WEC) devices have been developed and tested in the sea with promising results, where it is presented a thorough review of existing WEC systems. In this thesis work the Inertial Sea Wave Energy Converter (ISWEC) developed in Politecnico di Torino is considered. A picture of the ISWEC in Pantelleria sea is reported in figure 1.1.

Figure 1.1 – ISWEC device in Pantelleria Sea

The ISWEC project started in 2006 at Politecnico di Torino and involved several modelling and experimental activities at different scale levels. In August 2015, a full scale 100 kW prototype has been constructed and located near the shore of Pantelleria island (Sicily, Italy), which is one of the most effective energy sites in the Mediterranean sea. The ISWEC system consists of a floating device composed by a hull which hosts inside a gyroscope system whose motion is induced by the pitching oscillations of the hull floating on the sea surface. Energy extraction is obtained by damping the gyroscope precession motion through an electrical Power Take-Off (PTO) unit, which is implemented by an electric motor able to exert the torque intended for damping such a motion. The shaft about which the precession motion takes place is mechanically connected to the PTO shaft. The use of this application device is born for the Ocean application, but the quality of sea waves are best in Mediterranean sea, because the waves are short and frequent. So there should be a best energy conversion and development of electric energy.

As already mentioned the energy harvesting is accomplished by the action of damping the precession motion of the gyroscope by imposing a suitable torque to the PTO shaft. The damping action is computed by a suitable automatic control strategy, which in this thesis work it is based on the robust technique. Since the starting of project many control design are done, as for example a quite effective PD controller has been employed to compute the PTO command torque, and a MPC controller, that allows to handle efficiently different requirements by tuning

a customized quadratic cost function, which allows to find an optimal tradeoff among the different requirements such as energy production, command effort and PTO shaft speed limitation.

In this thesis the proposal is the design of a control-oriented model able to generate a simulated output with minor error from the real output. The design is computed with the Set-membership technique that allows at the user to quantify the uncertainty of the identified model through chosen constraints. Thanks to this building, a new project for a control will be simplest respect at the past control design, reducing the complexity of the system. At this model ,in the thesis, there is a preliminary robust control design with H-infinity technique, that allows to improve the output power, and so to improve the efficiency of ISWEC system.

Chapter 2

ISWEC system

The Inertial Sea Wave Energy Converter (ISWEC) is a system whose purpose is harvesting electric energy by converting the energy owned by the sea motions into electric energy. The following sections give a brief description of the ISWEC system and its main components relevant for understanding and addressing the control problem, as well as and its working principle and the dynamic equations which describe its behaviour.

2.1 System Description

The ISWEC consists of a hull floating on the sea surface which hosts in the inner environment different elements intended for producing and harvesting electrical energy. The hull inner environment is sealed such that the contained elements are safe from the outer environment, which could result otherwise dangerous. The main units hosted in the inner environment of the hull are the gyroscope system, which implements a spinning flywheel, and the Power Take-Off unit, which is the element intended for producing electric power. The ISWEC hull and the layout configuration of its main elements are shown in Figure 2.1

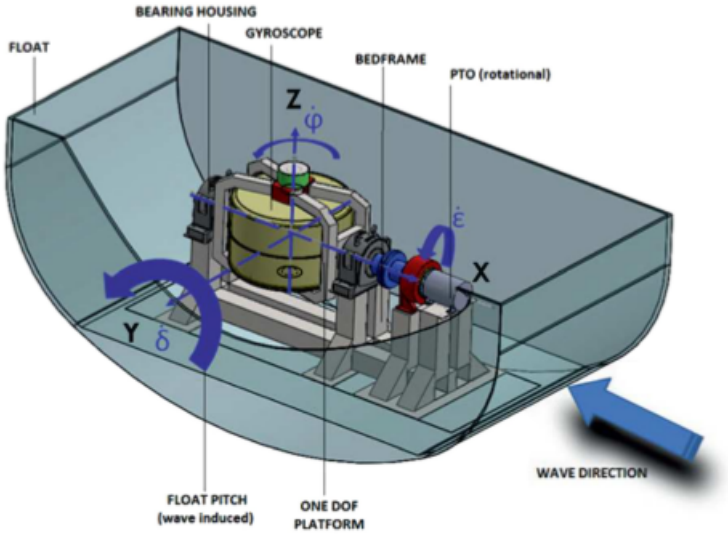


Figure 2.1 – ISWEC device

The figure also show the reference frame XYZ of the gyroscope system fixed with the hull. The axes X-axis, Y-axis and Z-axis can be equivalently referred to as -axis, -axis and -axis respectively.

The size of ISWEC device is described in this table 2.1:

Parameter	Value
Total Mass (of which sand ballast)	316 ton (200 ton)
Floater length	15 m
Floater width	8 m
Floater height	5 m
Water depth	32 m

Table 2.1 – ISWEC device

The gyroscope system is composed of a spinning flywheel which is enclosed in a case kept at low pressure for minimizing the drag resistance on the flywheel during the rotational motion. An electric motor is responsible for spinning the flywheel about the Z-axis. The flywheel case is mounted on a structure which allows the rotation about the X-axis. The physical phenomenon known as gyroscopic effect is responsible for rotating the spinning flywheel about the X-axis. More in details, as the gyroscope structure is rigidly connected to the hull floor,

whenever the hull undergoes a rotation about the Y-axis due to the excitation of the incoming wave, then the spinning flywheel of the gyroscope system undergoes the same pitch motion, and therefore an induced torque occurs on the spinning flywheel given by gyroscopic effect law which induces the flywheel to rotate about the X-axis. This phenomenon is also known as precession, and the induced rotational motion in consequently known as precession motion.

The energy production takes place by exploiting precession phenomenon. In particular the energy is obtained by damping of the precession motion through the Power Take-Off (PTO) unit, in fact the PTO is composed by an electric motor able to impose a suitable torque T induced for damping. The gyroscope shaft, which the precession motion takes place, is mechanically connected to the PTO shaft, and therefore the imposed torque by the PTO is able to damp the precession motion of the gyroscope system.

The ISWEC device in Pantelleria is composed by two gyropscopes in parallel. This application is imposed because of increasing of power of the system. Mechanically the device has the rated capacity as shown in the table 2.2 :

Parameter	Value
Rated Power	250 kW
Rated torque	200 kNm
Rated speed	25 rpm

Table 2.2 – Rated parameters of ISWEC device

2.2 System Dynamics

It's possible to provide a brief description on the dynamics of the system. It's not relevant in the computation of identification of model, but needs to understand what is the topic of the problem.

The introduced dynamics can be modelled by writing differential equations based on Newton torque equilibrium approach. Following this approach the

resulting dynamic equations with respect to the reference frame XYZ presented in figure 2.1 are the following:

$$T_\varepsilon(t) = I_g\,\ddot{\varepsilon}(t) + J\,\dot{\varphi}(t)\,\dot{\delta}(t)\cos\big(\varepsilon(t)\big)$$
$$T_\delta(t) = I_g\,\ddot{\delta}(t) - J\,\dot{\varphi}(t)\,\dot{\varepsilon}(t)\cos\big(\varepsilon(t)\big)$$
$$T_\varphi(t) = I_g\,\ddot{\delta}(t)\cos\big(\varepsilon(t)\big) - J\,\dot{\varepsilon}(t)\,\dot{\delta}(t)$$

The variables *Tϵ(t), Tδ(t)* and Tφ(t) represent the external torque exerted on the gyroscopic system respectively about *X-axis,Y-axis* and *Z-axis.Ig* is the total moment of inertia of the gyroscope system with respect to the *ϵ-axis* (or equivalently to the *δ-axis* due to the symmetry of the disc shape of the flywheel). *J* is the gyroscope axis-symmetric moment of inertia and $\dot{\varphi}$ is the flywheel spinning speed. $\dot{\epsilon}$ is the precession angular velocity and $\dot{\delta}$ the hull pitching speed, whereas $\ddot{\epsilon}$ and $\ddot{\delta}$ are respectively their angular acceleration.

The ISWEC device is self-orientating with respect to the incoming wave, thus the device interaction with waves can be formulated to a planar problem in the plane defined by the vertical Z-axis. Therefore the external dynamics can be expressed through only one degree of freedom which represent the ISWEC pitch motion about the -axis:

$$\tau_w(t) = (I_h + \mu_\infty)\ddot{\delta}(t) + \beta|\dot{\delta}(t)|\dot{\delta}(t) + K_w\,\delta(t) - J\,\dot{\varphi}(t)\,\dot{\varepsilon}(t)\,\cos\big(\varepsilon(t)\big) + \int_0^t \dot{\delta}(\tau)\,h(t-\tau)\,d\tau \tag{2.1}$$

So the system can be evaluate second the system equation, composed by (2.1) and the precession motion description:

$$T_\varepsilon(t) = I_g\,\ddot{\varepsilon}(t) + J\,\dot{\varphi}(t)\,\dot{\delta}(t)\cos\big(\varepsilon(t)\big) \tag{2.2}$$

Only this equation from the three degree equations, because this is the unique torque that acts on the direction of wave propagation, and that acts on the mechanical shaft, and so as consequence, on the PTO shaft. This is the torque which

it will be evaluated and analyzed to compute the control-oriented model.

It can be evaluate the hydrodynamics effect of the wave on gyroscope oscillation effect:

$$\mu(t) = \int_0^t \dot{\delta}(\tau)\, h(t-\tau)\, d\tau \tag{2.3}$$

2.3 Linearized model

For control purposes a linear model of the ISWEC system is required. Starting from equations (2.1) and (2.2) it is possible to apply a linearization in the neighborhood of the equilibrium position corresponding to $\epsilon = 0$ and $\delta = 0$. Moreover, the flywheel is supposed to spin at a constant speed $\dot{\varphi}$ =*const*.

The linearization of the first dynamic equation (2.1) gives:

$$T_\varepsilon(t) = I_g\, \ddot{\varepsilon}(t) + J\, \overline{\dot{\varphi}}\, \dot{\delta}(t) \tag{2.4}$$

Considering equation (2.2), it's possible to define a LTI system equation that describe the system:

$$\mu(t) = \int_0^t \dot{\delta}(\tau)\, h(t-\tau)\, d\tau \cong \begin{cases} \dot{\rho}_{rv}(t) = \boldsymbol{A}_\rho\, \boldsymbol{\rho}_{rv}(t) + \boldsymbol{B}_\rho\, \dot{\delta}(t) \\ \mu(t) = \boldsymbol{C}_\rho\, \boldsymbol{\rho}_{rv}(t) \end{cases} \tag{2.5}$$

where

$$\boldsymbol{\rho}_{rv}(t) = \left[\rho_{rv,1}(t), \quad \dots, \quad \rho_{rv,\nu}(t)\right]^T \in \mathbb{R}^\nu$$

This LTI system equation are useful to the design of a control in LTI technique, as PD controller or MPC controller, used in past project topics. This linearization is useful for these, because they base their technique on the mechanical theory

knowledge. For identification technique is useful to know the numbers of the states, expressed as follows, second the formulated LTI matrix:

$$\boldsymbol{A}_\rho = \begin{pmatrix} a_1 & a_2 & a_3 & \dots & a_\nu \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix}, \quad \boldsymbol{B}_\rho = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\boldsymbol{C}_\rho = \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_\nu \end{bmatrix}$$

A suitable choice for the value of v which results in valid approximation is v=4. According to the approximation introduced in (2.5) the linearization of the second dynamic equation (2.2) gives:

$$\tau_w = (I_h + \mu_\infty)\ddot{\delta}(t) + K_w\,\delta(t) - J\,\overline{\dot{\varphi}}\,\dot{\varepsilon}(t) + \boldsymbol{C}_\rho\,\boldsymbol{\rho}_{rv}(t) \tag{2.6}$$

The term $\beta\,|\,\dot{\delta}(t)\,|\,\dot{\delta}(t)\,|$ has been neglected as it assumes negligible values with respect to the other terms. This is due to the fact that the pitching angular speed of the hull $\dot{\delta}$ assume very small values.

And so the state vector has defined as:

$$\boldsymbol{x}(t) = \begin{bmatrix} \dot{\varepsilon}(t) & \varepsilon(t) & \dot{\delta}(t) & \delta(t) & \rho_{rv,1}(t) & \rho_{rv,2}(t) & \rho_{rv,3}(t) & \rho_{rv,4}(t) \end{bmatrix}^T \tag{2.7}$$

Appearing in the LTI dynamical system equation:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\,\boldsymbol{x}(t) + \boldsymbol{B}\,T_\varepsilon(t) + \boldsymbol{B}_d\,\tau_w(t) \tag{2.8}$$

where the matrix are:

$$
\boldsymbol{A} = \begin{pmatrix}
0 & 0 & -\frac{J\overline{\dot{\varphi}}}{I_g} & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{J\overline{\dot{\varphi}}}{I_{eq}} & 0 & 0 & -\frac{k_w}{I_{eq}} & -\frac{c_1}{I_{eq}} & -\frac{c_2}{I_{eq}} & -\frac{c_3}{I_{eq}} & -\frac{c_4}{I_{eq}} \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & a_{11} & a_{12} & a_{13} & a_{14} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
$$

$$
\boldsymbol{B} = \begin{bmatrix} \frac{1}{I_g} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T
$$

$$
\boldsymbol{B}_d = \begin{bmatrix} 0 & 0 & \frac{1}{I_{eq}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{2.9}
$$

Ieq is expressed as:

$$I_{eq} = I_h + \mu_\infty.$$

Futhermore it The linearized ISWEC model (2.8) represents the state-space model of the ISWEC which can be used for control purposes. The variable *Te(t)* is the manipulable input, usually identified by the letter *u(t)*. The variable *τw(t)* is the input, consisting in the applied torque by the incoming wave on the hull about the pitch axis δ, and it is considered as a disturbance as it is a not manipulable input.

For the computation of identification the identified state are 8. In fact remember that the number of states/variables is important for identification because define the complexity of order of model. More variables define a more complex computation, but there is more accuracy. As discuss in chapter 3, the choice of number of variables in identification has been based on compromise between complexity and accuracy. Complexity has done by used processor (Intel Core i7-7700HQ Quad Core, 2.8 GHz frequency).

2.4 Power extraction

The importance of control-oriented model is to design a controller that improve the efficiency of the system, i.e. increase the output power and reduce the power absorbed by the mechanical system. The PTO power has defined as:

$$P_{PTO}(t) = \dot{\varepsilon}(t) \cdot T_{\varepsilon}(t)$$

So the quantity *Ppto(t)* is a power absorbed by the system provided by the external environment. Therefore a power produced by the system extracted from the external environment is characterized to be a negative quantity. This implies that the quantity *Ppto(t)* has to be minimized by the robust controller in order to be as negative as possible during time.

A negative quantity of the power *Ppto(t)* is obtained by damping the gyroscope shaft precession motion through a given command action $T\epsilon(t)$ provided by a suitable feedback control law. Moreover, the computed command action Tϵ(t) is supposed to let the gyroscope system assume an oscillating behaviour around ϵ=0, thus avoiding full rotations, despite mechanically feasible. In addition the values of the PTO shaft angular speed $\dot{\epsilon}(t)$ should be minimized to reduce wear and solicitations on the PTO gearbox and power drive line. Furthermore, for optimizing the energy production, the control energy effort spent by the PTO torque action must be minimized.

The desired behaviour of controller is to reduce the PTO power and to make the control robust, i.e. stable at any disturbance.

Chapter 3

Set-Membership Identification

3.1 Introduction to Set-Membership theory

The problem of model identification for linear systems is considered, using a finite set of sampled data affected by a bounded measurement noise, with unknown bound. The objective is to identify models and their accuracy in terms of worst-case simulation error bounds. To do so, the Set Membership identification framework is exploited. Theoretical results are derived, allowing one to estimate the noise bound and system decay rate. Then, these quantities and the data are employed to define the Feasible Parameter Set (FPS), which contains all possible models compatible with the available information. Here, the estimated decay rate is used to refine the standard FPS formulation, by adding constraints that enforce the desired converging behavior of the model's impulse response. Moreover, guaranteed simulation error bounds for an infinite future horizon are derived, improving over recent results pertaining to finite simulation horizon only. These bounds are the basis for a result and method to guarantee asymptotic stability of the identified model. Finally, the desired model is identified by means of numerical optimization, and the related simulation error bounds are evaluated. Both input-output and state-space model structures are addressed.

3.1.1 Set-Membership (Bound error) Approach

It analyze a model in Time-Invariant, i.e. the model function doesn't depend on time. Otherwise it's said to be Time-Variant. To introduce at the bound error approach of a LTI model, there are two assumptions to know:

1) A-Priori assumption on the Plant: as seeing in figure 3.1 the plant can be a function that belongs to the a class function F. In this class belong non-linear function that have characteristics of continuos, differentiable, polynomial, linear comibination... The model plant is said to be Parametric when the class F is described by means of a finite number of parameters. Instead the Non-Parametric model is a model when F can not be described by means of a finite number of parameters.

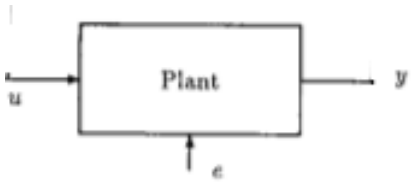


Figure 3.1 – Plant model

2) A-Priori assumption on the Noise: the errors $\epsilon(t)$ and $\eta(t)$ belong to the class Be, i.e. the class set that determines the set of errors (constraints).

$$e^N \in \bar{\mathcal{B}}_e = \left\{ e^N \in \Re^N : \|e^N\| \leq \varepsilon \right\}$$

As described before, the error has to bounded. Much important to derive the parameters of model is to get a bound in the errors. Bound set knowing the system disturbances. Remember that I have to not define the distribution of error in the set. It's not to known the distribution probability of error. The important qualities are the extremes of set, as example in figure 3.2.

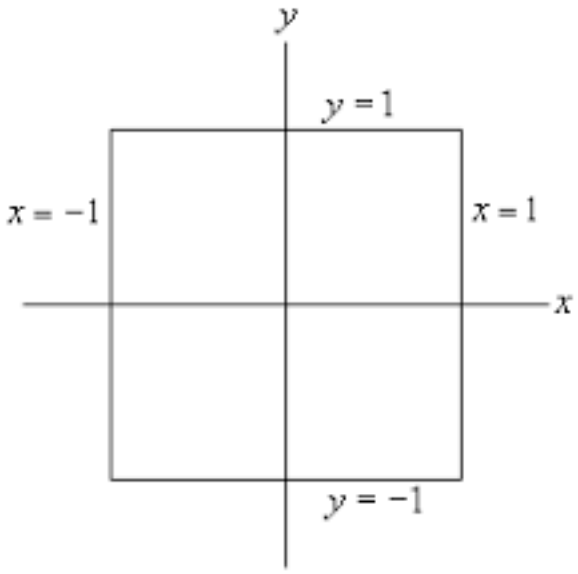


Figure 3.2 – Example of set in 2D

So the A-priori assumptions are traduced as:

The linearized equation of plant, defined in the example as EE(Equation error):

$$\begin{aligned} y(t) = &-\alpha_1 y(t-1) - \alpha_2 y(t-2) - ... - \alpha_n y(t-n) + \\ &+ \beta_0 u(t) + \beta_1 u(t-1) + ... + \beta_n u(t-n) + e(t) \end{aligned} \tag{3.1}$$

The set equation of bounded error:

$$\begin{aligned} S = \{e(t), t = 1, .., N :, \\ |e(t)| \leq \Delta e\} \end{aligned} \tag{3.2}$$

These equations are essential to define a parameter set, which are consistent with a-priori information on the plant and the noise for all the collected data. This set is called Feasible Parameter Set (FPS):

$$\begin{aligned} D_\theta = \{&\theta = [\theta_1, \theta_2, ..., \alpha_N, \beta_0, \beta_1, ..., \beta_N] \in \mathbb{R}^{2n+1} :, \\ &y(t) = -\alpha_1 y(t-1) + ... - \alpha_n y(t-n) + \beta_0 u(t) + ... + \beta_n u(t-n) + e(t) \\ &|e(t)| \leq \Delta e, t = n+1, ..., N\} \end{aligned} \tag{3.3}$$

If it defines the error by the plant equation 3.1 as:

$$\begin{aligned} e(t) = y(t) + \alpha_1 y(t-1) + \alpha_2 y(t-2) - ... + \alpha_n y(t-n) + \\ - \beta_0 u(t) - \beta_1 u(t-1) + ... - \beta_n u(t-n) \end{aligned} \tag{3.4}$$

And substituting in 3.3:

$$\begin{aligned} D_\theta = \{\theta \in \mathbb{R}^{2n+1}, \\ |y(t) + \alpha_1 y(t-1) + \alpha_2 y(t-2) - ... + \alpha_n y(t-n) + \\ - \beta_0 u(t) - \beta_1 u(t-1) + ... - \beta_n u(t-n)| \leq \Delta e, t = n+1, ..., N\} \end{aligned} \tag{3.5}$$

Analyzing the absolute value it's possible to define two equations depending on the bounded error, one for lower value, and the other one for higher value:

$$\begin{aligned} D_\theta = \{\theta \in \mathbb{R}^{2n+1}, \\ y(t) + \alpha_1 y(t-1) - \alpha_2 y(t-2) - ... \geqslant -\Delta e, \\ - y(t) - \alpha_1 y(t-1) - \alpha_2 y(t-2) - ... \geqslant -\Delta e, t = n+1, ..., N\} \end{aligned} \tag{3.6}$$

With FPS, it's defined first constraints on the system for Set-Membership Identification. The two equations 3.6 give constraints on the set of parameters of plant model.

Look an example with only two parameters:

$$\begin{aligned} D_\theta = \{\theta = [\theta_1, \theta_2] : |y(t) - \theta_1 y(t-1) - \theta_2 u(t)| \geqslant -\Delta e \\ t = n+1, ..., N\} \end{aligned} \tag{3.7}$$

$$\begin{aligned} D_\theta = \{\theta = [\theta_1, \theta_2] : y(t) - \theta_1 y(t-1) - \theta_2 u(t) \geqslant -\Delta e, \\ - y(t) + \theta_1 y(t-1) + \theta_2 u(t) \geqslant -\Delta e, t = n+1, ..., N\} \end{aligned} \tag{3.8}$$

By highlighting the parameters, it can see that they depend on:

$$D_\theta = \{\theta = [\theta_1, \theta_2] : \theta_2 \geqslant \frac{y(t-1)}{-u(t)}\theta_1 + \frac{\Delta e - y(t)}{-u(t)}, \\ \theta_2 \leq \frac{y(t-1)}{-u(t)}\theta_1 + \frac{-\Delta e - y(t)}{-u(t)}, t = n+1, ..., N\} \tag{3.9}$$

In this way, it analyzes the dependence of the parameters on each other, and observe the set in the graph 3.3:

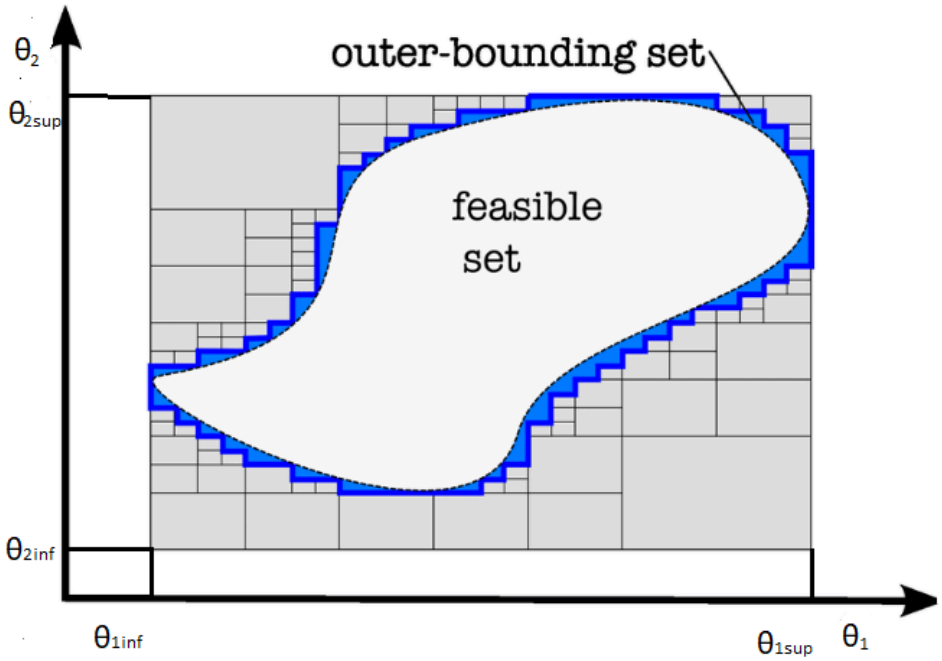


Figure 3.3 – Example of a Feasible Parameter Set

From the FPS it is possible to define parameter ranges given by the system uncertainty, Parameter Uncertainty Intervals (PUI), as can be seen from the figure above 3.3. From this it has the ability to give an objective function 3.10 to the problem, i.e. to minimize a linear function in order to obtain the desired parameter interval.

$$\theta_1 inf = min\theta_1 \tag{3.10}$$

When it has a minimization of a linear function of variables, subject to a set of a equalities and/or inequalities constraints which are linear in optimization

problem, it solves with Linear Programming problem (LP), that is a convex optimization problem,i.e. have not local minimum, but only global one.

Before talking about Linear programming, it should anticipate an alternative analysis method for the Feasible Parameter Set. Sometimes the output error can be expressed as the difference between real and simulated output as in 3.11. This type of approach is called EIV (error invariable), because the error enters the system according to the error invariable structure.

$$e(t) = y(t) - w(t) \tag{3.11}$$

,where $e(t)$ is the error, $y(t)$ is the real output, and $w(t)$ is the simulated output.

The Feasible Parameter Set is integrated with the output dependent error constraint. In this way the inequalities that make up our constraints are enriched by an additional constraint..:

$$\begin{aligned} D_\theta = \{&\theta = [\alpha_1, \alpha_2, ..., \alpha_N, \beta_0, \beta_1, ..., \beta_N] \in \mathbb{R}^{2n+1} :, \\ &y(t) = -\alpha_1 y(t-1) + ... - \alpha_n y(t-n) + \beta_0 u(t) + ... + \beta_n u(t-n), \\ &y(t) = e(t) + w(t), |e(t)| \leq \Delta e, t = n+1, ..., N\} \end{aligned} \tag{3.12}$$

By replacing the output the main equations also depend on the error, and simulated output as see in equation 3.13.

$$\begin{aligned} D_\theta = \{&\theta = [\alpha_1, \alpha_2, ..., \alpha_N, \beta_0, \beta_1, ..., \beta_N] \in \mathbb{R}^{2n+1} : \\ &e(t) + w(t) = -\alpha_1(e(t-1) + w(t-1)) ... - \alpha_n(e(t-n) + w(t-n)) \\ &+ \beta_0 u(t) + ... + \beta_n u(t-n), |e(t)| \leq \Delta e, t = n+1, ..., N\} \end{aligned} \tag{3.13}$$

Now this type of Feasible Parameter Set strictly depends on the error that enters with an EIV approach, and where the variables are the system parameters. But in order to make the set solvable you need to integrate the errors that enter as variables as well. So the FPS will be called Extended Feasible Parameter Set (EFPS). A much more complex set that no longer needs a solution in Linear

Programming(LP), but that needs a Bilinear Programming Optimization Problem, since there is a direct interaction between several variables (θ*e). So the problem results: not convex (presence of more local minima); it is defined as a special case of a Polynomial Optimization Problems; at least theoritically bilinear optimization problems are not easier than generical polynomial optimization problems.

3.1.2 Bilinear Optimization Problem

In mathematics, a bilinear program is a nonlinear optimization problem whose objective or constraint functions are bilinear. A NonLinear Programming (NLP) is the process of solving an optimization problem where some of the constraints or the objective function are nonlinear. An optimization problem is one of calculation of the extrema (maxima, minima or stationary points) of an objective function over a set of unknown real variables and conditional to the satisfaction of a system of equalities and inequalities, collectively termed constraints.
A nonlinear minimization problem is an optimization problem of the form:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } g_i(x) \leq 0 \text{ for each } i \in \{1, \ldots, m\} \\ &\qquad\qquad h_j(x) = 0 \text{ for each } j \in \{1, \ldots, p\} \\ &\qquad\qquad x \in X. \end{aligned}$$

Figure 3.4 – Example of form of NLP

The difficulty of a non-convex problem can be solved with Polynomial Optimization Problem (POP), i.e. problem is optimized through polynomials (equality or inequality) imposed to solve the minimization of the variability(s).

3.1.3 SparsePOP

A tool used for the POP problem is SparsePoP.
SparsePOP is a Matlab package for finding global optimal solutions of polynomial optimization problems (POPs). The package is an implementation of a sparse semidefinite programming (SDP) relaxation method for POPs, proposed to improve the efficiency of Lasserre's hierarchy of LMI relaxations of increasing

dimensions. SparsePOP exploits the sparsity of POPs so that it can handle POPs of larger dimensions.The package accepts a POP as input, and outputs solution information and statistics. The main part constructs a sparse SDP relaxation of the POP and uses SeDuMi to obtain an approximate global optimal solution.

The objective and constraint polynomials of a POP can be described in two different ways, namely, the GAMS scalar format and the SparsePOP format, to be read by SparsePOP.

A polynomial class is defined for this purpose as follows:

poly.typeCone	=	1	if $f(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ is used as an objective function,
	=	1	if $f(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ is used as an inequality constraint $f(\boldsymbol{x}) \geq 0$,
	=	-1	if $f(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ is used as an equality constraint $f(\boldsymbol{x}) = 0$.
poly.degree	=		the degree of $f(\boldsymbol{x})$.
poly.dimVar	=		the dimension of the variable vector $\boldsymbol{x}$.
poly.noTerms	=		the number of terms of $f(\boldsymbol{x})$.
poly.supports	=		a set of supports of $f(\boldsymbol{x})$,
			a poly.noTerms × poly.dimVar matrix.
poly.coef	=		coefficients,
			a column vector of poly.noTerms dimension.

The name **objPoly** is for the objective polynomial function $f_0(x)$ and $ineqPolySysj(j = 1,2,...,m)$ for the polynomials $f_j(x)(j = 1,2,...,m)$ of the constraints. The problem is described using the polynomial class as follows.

```
function [objPoly,ineqPolySys,lbd,ubd] = example1;
%%%%%%%%%%%%%%
% example1.m
%%%%%%%%%%%%%%
%
% The SparsePOP format data for the example1:
%
```

```
% minimize -2*x1 +3*x2 -2*x3
% subject to
%       x1^2 + 3*x2^2 -2*x2*x3 +3*x3^2 -17*x1 +8*x2 -14*x3 >= -19,
%       x1 + 2*x2 + x3 <= 5,
%       5*x2 + 2*x3 = 7,
%       0 <= x1 <= 2, 0 <= x2 <= 1.
%
% To solve the problem by sparsePOP.m:
% >> param.relaxOrder = 3;
% >> sparsePOP('example1',param);
%
% This problem is also described in terms of the GAMS scalar format in the
% file example1.gms.  See Section 3 of the manual.
%
%'example1'
% objPoly
% -2*x1 +3*x2 -2*x3
        objPoly.typeCone = 1;
        objPoly.dimVar   = 3;
        objPoly.degree   = 1;
        objPoly.noTerms  = 3;
        objPoly.supports = [1,0,0; 0,1,0; 0,0,1];
        objPoly.coef     = [-2; 3; -2];
% ineqPolySys
% 19 -17*x1 +8*x2 -14*x3 +6*x1^2 +3*x2^2 -2*x2*x3 +3*x3^2 >= 0,
        ineqPolySys{1}.typeCone = 1;
        ineqPolySys{1}.dimVar   = 3;
        ineqPolySys{1}.degree   = 2;
        ineqPolySys{1}.noTerms  = 8;
        ineqPolySys{1}.supports = [0,0,0; 1,0,0; 0,1,0; 0,0,1; ...
                                   2,0,0; 0,2,0; 0,1,1; 0,0,2];
        ineqPolySys{1}.coef     = [19; -17; 8; -14; 6; 3; -2; 3];
%
```

```
   % 5 -x1 -2*x2 -x3  >= 0.
           ineqPolySys{2}.typeCone = 1;
           ineqPolySys{2}.dimVar   = 3;
           ineqPolySys{2}.degree   = 1;
           ineqPolySys{2}.noTerms  = 4;
           ineqPolySys{2}.supports = [0,0,0; 1,0,0; 0,1,0; 0,0,1];
           ineqPolySys{2}.coef     = [5; -1; -2; -1];
   %
   % 7 -5*x2 -2*x3 = 0.
           ineqPolySys{3}.typeCone = -1;
           ineqPolySys{3}.dimVar   = 3;
           ineqPolySys{3}.degree   = 1;
           ineqPolySys{3}.noTerms  = 3;
           ineqPolySys{3}.supports = [0,0,0; 0,1,0; 0,0,1];
           ineqPolySys{3}.coef     = [7; -5; -2];
   % lower bounds for variables x1, x2 and x3.
   % 0 <= x1, 0 <= x2, -infinity < x3:
           lbd = [0,0,-1.0e10];
   % upper bounds for variables x1, x2 and x3



% x1 <= 2, x2 <= 1, x3 < infinity:
        ubd = [2,1,1.0e10];
return
% end of example1.m
```

We note that -1.0e10 in lbd and 1.0e10 in ubd mean -∞ and ∞, respectively, indicating x3 can take any value in the above example.

Since example1.m contains the description of the POP in the SparsePOP format, the POP can be solved by SparsePOP as follows:

```
>> [objPoly,ineqPolySys,lbd,ubd] = example1;
>> param.relaxOrder = 3;
>> sparsePOP(objPoly,ineqPolySys,lbd,ubd,param);
```

3.1.4 Optimization toolbox for SparsePOP

When Optimization Toolbox is available in Matlab, it can help refine an approximation to an optimal solution of a POP obtained from its sparse SDP relaxation by setting the parameter. That is, set param.POPsolver to 'active-set', 'trust-region-reflective' or 'interiorpoint' to apply the Matlab function *fmincom.m* from the toolbox. For example, if we issue the commands.

```
>> param.relaxOrder = 3;
>> param.POPsolver = 'active-set';
>> sparsePOP('example1.gms',param);
```

Then Matlab computes the parameters of objective value and defined it as:

```
SparsePOP 2.20
by H.Waki, S.Kim, M.Kojima, M.Muramatsu,
   H.Sugimoto and M.Yamashita, July 2009

  . . . . . .

## Computational Results by sparsePOP.m with SeDuMi ##
## Printed by printSolution.m ##

  . . . . . .

# Approximate optimal value information:
  SDPobjValue         = -7.9550950e+00
  POP.objValue        = -7.9550951e+00
  relative obj error  = +1.086e-09
  POP.absError        = -1.295e-06
  POP.scaledError     = -2.642e-08
```

```
# Approximate optimal solution information:
  POP.xVect =
     1:+4.7754773e-01    2:+5.1730095e-08    3:+3.4999999e+00

## fmincon or fminunc in Optimization Toolbox

Optimization terminated: first-order optimality measure less
 than options.TolFun and maximum constraint violation is less
 than options.TolCon.

## Computational Results by medium-scale: SQP, Quasi-Newton, line-search method
   with the initial solution obtained by the sparse SDP relaxation ##



  exitflag            =      1
  iterations          =      2
  elapsed time        =      0.76
# Approximate optimal value information:
  POP.objValueL       = -7.9550954e+00
  relative obj error  = +6.771e-07
  POP.absErrorL       = +0.000e+00
  POP.scaledErrorL    = +0.000e+00
# Approximate optimal solution information:
  POP.xVectL =
     1:+4.7754769e-01    2:+0.0000000e+00    3:+3.5000000e+00
```

It is noted that the parameters calculated by Optimization tool are under the POPxVectL field unlike POPxVect which are the parameters without optimization calculated by SparsePop. In addition, the optimized parameters are better because they assume precise values. Observe how the second value from 10-8 has passed to 0, since the number 10-8 is very small and therefore negligible.

3.2 Identification for ISWEC model

This section will show the identification procedure adopted in the ISWEC model. Procedure described in the previous sections, but that undergoes modifications because the objective function does not result the parameters, as described in the chapter 3.1.1, but the error between real and simulated output. This kind of procedure is done because the goal is to minimize the error to find the parameters referred to the minimum calculated error.

3.2.1 Collected Data

First of all, the collected data used to identify the multi input-output model (Double Input Single Output) was taken from old projects, where a MPC controller was used. It collected the inputs, such as the wave signal, where we remember there are 9 different types of wave signals, and the torque generated by the drive shaft, which then continues towards the PTO shaft. Instead as output, the angular speed of the PTO shaft, i.e. the first derivative the angular displacement ϵ, is taken.

So the model that will be analyzed will be the one that will try to simulate the behavior of a model in which there is a MPC controller. So the output is not just the mechanical system, but a controlled output.The input of the torque is also subject to the action of the controller.

This approach is useful to try to carry out a control that improves the efficiency of the model, since it is already based on a first controller.

It shows the Simulink block where the data is taken from:

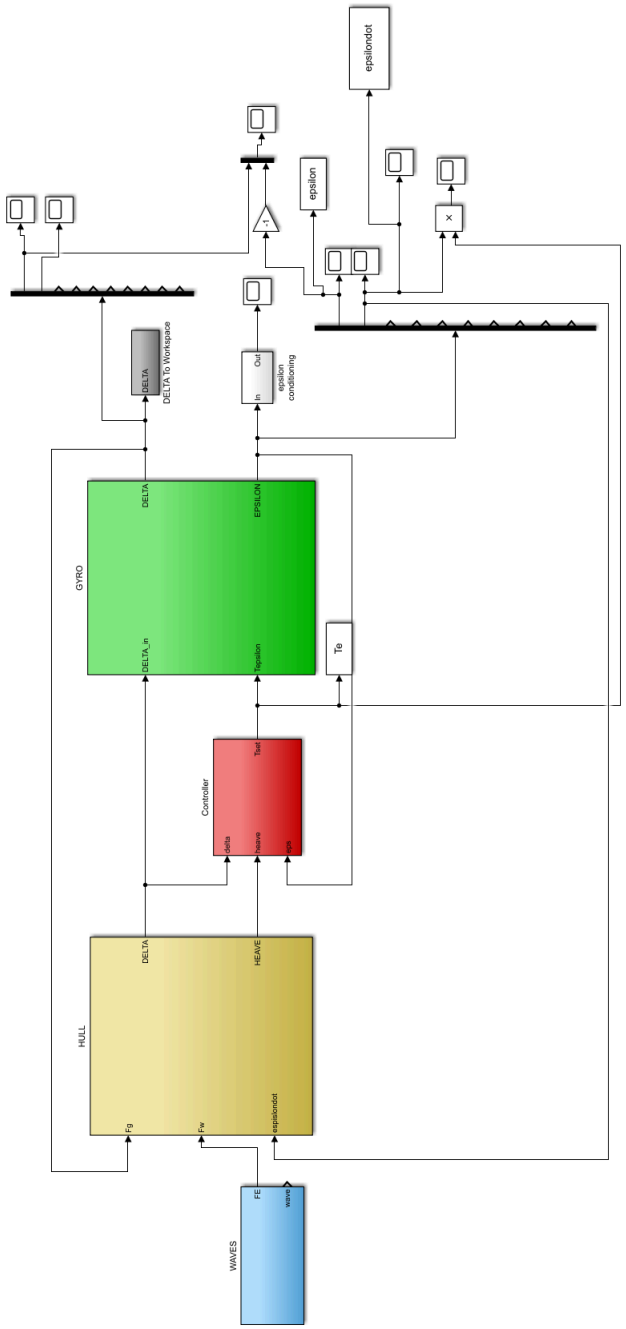


Figure 3.5 – MPC controller

3.2.2 SparsePOP Identification

Once the data has been collected, the next step is to define a SparsePOP construct to identify the model. To do this, it need to build and define equalities and inequalities adhering to the system.
Before imposing the transfer function equation, it is necessary to impose and decide how many states the model must be in order to understand how many parameters must be identified.
And above all, the parameters are variables in the system and therefore more parameters there are, more complex the system becomes. So the decision is taken according to the states of the mechanical model, which are 8, as observable in 2.7, and the desired complexity that one wants to give to the model. Obviously, the more complex the system is, the more accurate it will be, but there is a limit depending on the processor used. So it was decided to use a model that had 9 states.

Therefore the plant model will be of this structure:

$$G_1(q^{-1}) = \frac{\dot{\epsilon_1}(z)}{T_\epsilon(z)} = \frac{\beta_{10} + \beta_{11}q^{-1} + \beta_{12}q^{-2} + ... + \beta_{19}q^{-9}}{1 + \alpha_{11}q^{-1} + \alpha_{12}q^{-2} + ... + \alpha_{19}q^{-9}} \tag{3.14}$$

This is the model that relates the PTO torque input T_ϵ with simulated output $\dot{\epsilon_1}$. This is the plant model of the system.

But as mentioned earlier, this is a system with two inputs and one output. The other input is the wave signal, which in our system we will evaluate as disturbance. A disturbance, however, which, as you can see in the control chapter, will have the input functionality in the system such as to generate output power, just like in the mechanical operation of the system.

Therefore the disturbance model will be of this structure:

$$G_2(q^{-1}) = \frac{\dot{\epsilon_2}(z)}{Fry(z)} = \frac{\beta_{20} + \beta_{21}q^{-1} + \beta_{22}q^{-2} + ... + \beta_{29}q^{-9}}{1 + \alpha_{21}q^{-1} + \alpha_{22}q^{-2} + ... + \alpha_{29}q^{-9}} \tag{3.15}$$

This is the model that relates the signal wave input Fry with simulated output

$\dot{\epsilon_2}$ .this is the disturbance model

Note that in the two transfer functions there are the $\dot{\epsilon_1}$ and $\dot{\epsilon_2}$ outputs, which are the contributions of the angular speed of the PTO shaft, of each input. Where their sum makes up the system's output $\dot{\epsilon}$:

$$\dot{\epsilon}(t) = \dot{\epsilon_1}(t) + \dot{\epsilon_2}(t) \tag{3.16}$$

From these three equations 3.14, 3.15, 3.16 are the constraints of SparsePop. Together with these, we have to add the equation that links the real output with the simulated one, then we show the link with the error, which its minimization will be the objective function.

$$y(t) = \dot{\epsilon}(t) + \eta(t) \tag{3.17}$$

And also because the Set Membership approach is a bounded error. You will have to put the inequality that says that the absolute value of the error must be less than or equal to a value, in order to make it bounded.

$$|\eta(t)| \leq \gamma \tag{3.18}$$

From these simple equality and inequalities it will be possible to define a SparsePop structure. First of all you have to work on the transfer functions in order to make them dependent on input and output and build an equation that affirms this.

Starting from the transfer function G_1 3.14, an equation is defined:

$$\frac{\dot{\epsilon_1}(t)}{T_\epsilon(t)} = \frac{\beta_{10} + \beta_{11}q^{-1} + \beta_{12}q^{-2} + ... + \beta_{19}q^{-9}}{1 + \alpha_{11}q^{-1} + \alpha_{12}q^{-2} + ... + \alpha_{19}q^{-9}} \tag{3.19}$$

$$\begin{aligned} &\dot{\epsilon_1}(t)(1 + \alpha_{11}q^{-1} + \alpha_{12}q^{-2} + ... + \alpha_{19}q^{-9}) = \\ &T_\epsilon(t)(\beta_{10} + \beta_{11}q^{-1} + \beta_{12}q^{-2} + ... + \beta_{19}q^{-9}) \end{aligned} \tag{3.20}$$

Applying the rules of discretization defines the equation as:

$$\begin{aligned}&\dot{\epsilon_1}(t) + \alpha_{11}\dot{\epsilon_1}(t-1) + \alpha_{12}\dot{\epsilon_1}(t-2) + ... + \alpha_{19}\dot{\epsilon_1}(t-9)) = \\ &\beta_{10}T_\epsilon(t) + \beta_{11}T_\epsilon(t-1) + \beta_{12}T_\epsilon(t-2) + ... + \beta_{19}T_\epsilon(t-9))\end{aligned} \tag{3.21}$$

And so the equation is found:

$$\begin{aligned}&\dot{\epsilon_1}(t) + \alpha_{11}\dot{\epsilon_1}(t-1) + \alpha_{12}\dot{\epsilon_1}(t-2) + ... + \alpha_{19}\dot{\epsilon_1}(t-9)) - \\ &- \beta_{10}T_\epsilon(t) - \beta_{11}T_\epsilon(t-1) - \beta_{12}T_\epsilon(t-2) + ... - \beta_{19}T_\epsilon(t-9)) = 0\end{aligned} \tag{3.22}$$

An equation is found that determines the relationship between the torque of the PTO shaft and the angular speed of the shaft in this equation constraint.

Same thing it has to do for the other transfer function, where instead it determines the relationship between the wave signal and the speed.

$$\frac{\dot{\epsilon_2}(t)}{Fry(t)} = \frac{\beta_{20} + \beta_{21}q^{-1} + \beta_{22}q^{-2} + ... + \beta_{29}q^{-9}}{1 + \alpha_{21}q^{-1} + \alpha_{22}q^{-2} + ... + \alpha_{29}q^{-9}} \tag{3.23}$$

$$\begin{aligned}&\dot{\epsilon_2}(t)(1 + \alpha_{21}q^{-1} + \alpha_{22}q^{-2} + ... + \alpha_{29}q^{-9}) = \\ &Fry(t)(\beta_{20} + \beta_{21}q^{-1} + \beta_{22}q^{-2} + ... + \beta_{19}q^{-9})\end{aligned} \tag{3.24}$$

$$\begin{aligned}&\dot{\epsilon_2}(t) + \alpha_{21}\dot{\epsilon_2}(t-1) + \alpha_{22}\dot{\epsilon_2}(t-2) + ... + \alpha_{29}\dot{\epsilon_2}(t-9)) = \\ &\beta_{20}Fry(t) + \beta_{21}Fry(t-1) + \beta_{22}Fry(t-2) + ... + \beta_{29}Fry(t-9))\end{aligned} \tag{3.25}$$

$$\begin{aligned}&\dot{\epsilon_2}(t) + \alpha_{21}\dot{\epsilon_2}(t-1) + \alpha_{22}\dot{\epsilon_2}(t-2) + ... + \alpha_{29}\dot{\epsilon_2}(t-9)) - \\ &- \beta_{20}Fry(t) - \beta_{21}Fry(t-1) - \beta_{22}Fry(t-2) + ... - \beta_{29}Fry(t-9)) = 0\end{aligned} \tag{3.26}$$

In the equation 3.22 there are $\dot{\epsilon_1}$ and the parameters α_{1i} and β_{1i} that turn out to be variables, i.e. unknowns of the system. Instead in the 3.26 as variables there are $\dot{\epsilon_2}$ and the parameters α_{2i} and β_{2i}. This shows that the resolution is bilinear where only T_ϵ and Fry are known values.

The relationship that links the two outputs of the transfer functions is the equation 3.16:

$$\dot{\epsilon}(t) = \dot{\epsilon_1}(t) + \dot{\epsilon_2}(t) \tag{3.27}$$

,where $\dot{\epsilon}$ is known, and ties the two unknown output $\dot{\epsilon_1}$ and $\dot{\epsilon_2}$ with the sum of the two.

Now moving on to the equation that links the real output with the simulated one, it can define an equation in which the error is present. So from the equations 3.17 and 3.18 the equation is formed:

$$\eta(t) = y(t) - \dot{epsilon}(t) \tag{3.28}$$

$$|y(t) - \dot{epsilon}(t)| \leq \gamma \tag{3.29}$$

From this equation it is possible to define two constraints equalities, negative and positive value of the absolute value:

$$y(t) - \dot{epsilon}(t) \leq \gamma \tag{3.30}$$

$$-y(t) + \dot{epsilon}(t) \leq \gamma \tag{3.31}$$

To put them as constraint inequalities in SparsePop they must be represented in this way:

$$-\dot{\epsilon(t)} \leq \gamma - y(t) \tag{3.32}$$

$$\dot{\epsilon(t)} \leq \gamma + y(t) \tag{3.33}$$

From these 3.33, 3.32, substituing in 3.16, inequalities are formed to put inside in SparsePop structure:

$$\dot{\epsilon_1}(t) + \dot{\epsilon_2}(t) - y(t) + \gamma \geqslant 0 \tag{3.34}$$

$$-\dot{\epsilon_1}(t) - \dot{\epsilon_2}(t) + y(t) + \gamma \geqslant 0 \tag{3.35}$$

From the equations 3.22, 3.26, 3.27, 3.32 e 3.33 it can form the Extended Feasible Parameter Set (EFPS), so formed:

$$\begin{aligned} D_{\theta,\dot{\epsilon_1},\dot{\epsilon_2},y,\gamma} = \{\theta = [&\alpha_{11}, \alpha_{12}, ..., \alpha_{19}, \beta_{10}, \beta_{11}, ..., \beta_{1N}, \\ &\alpha_{21}, \alpha_{22}, ..., \alpha_{29}, \beta_{20}, \beta_{21}, ..., \beta_{2N}] \in \mathbb{R}^{2n+1} : \\ &\dot{\epsilon_1}(t) + \alpha_{11}\dot{\epsilon_1}(t-1) + \alpha_{12}\dot{\epsilon_1}(t-2) + ... + \alpha_{19}\dot{\epsilon_1}(t-9)) - \\ &- \beta_{10}T_\epsilon(t) - \beta_{11}T_\epsilon(t-1) - \beta_{12}T_\epsilon(t-2) + ... - \beta_{19}T_\epsilon(t-9)) = 0, \\ &\dot{\epsilon_2}(t) + \alpha_{21}\dot{\epsilon_2}(t-1) + \alpha_{22}\dot{\epsilon_2}(t-2) + ... + \alpha_{29}\dot{\epsilon_2}(t-9)) - \\ &- \beta_{20}Fry(t) - \beta_{21}Fry(t-1) - \beta_{22}Fry(t-2) + ... - \beta_{29}Fry(t-9)) = 0, \\ &\dot{\epsilon_1}(t) + \dot{\epsilon_2}(t) - y(t) + \gamma \geqslant 0, \\ &- \dot{\epsilon_1}(t) - \dot{\epsilon_2}(t) + y(t) + \gamma \geqslant 0, t = n+1, ..., N\} \end{aligned} \tag{3.36}$$

Where the objective function is:

$$argmin\gamma \tag{3.37}$$

It will minimize the error to find the parameters corresponding to its minimization.

When talking about the constraints,it will discuss the choice of the values to be placed within the SparsePop structure, which values will reflect the translation of the chosen constraints.

The first things to do are to define the objective function structure, **objPoly** , where it enter global information about the whole structure to be given on polynomial constraints:

```
% objective function, minimize deps to find theta
        objPoly.typeCone = 1;
        objPoly.dimVar   = 39+2*N;
        objPoly.degree   = 1;
        objPoly.noTerms  = 1;
        objPoly.supports = [zeros(1, 39+2*N )];
        objPoly.coef     = 1;
        objPoly.supports(39) = 1;   %posizione 39 che corrisponde a deps
```

It is observed that the values *objPoly.typeCone=1* because it is used for the objective function, *objPoly.degree=1* for a single varable which is the error γ in the objective function, *objPoly.dimVar=39+2*N*. which defines the size of the variables and therefore 38 parameters to identify θ, 1 error variables γ , and 2*N variables of the outputs of the two transfer functions which are unknown to us. Other parameters to set are *objPoly.noTerms=1*. , which is the number of terms placed in the objective function, only one since the function contains only one term. *objPoly.coef=1* which defines that the term has coefficient 1, and *objPoly.support(39)=1* which defines that at position 39 of the support matrix (matrix that defines whether the parameters are unknown or not, where 1 is placed if the variable is unknown) there is an unknown variable (the error bounded γ).

The equalities in the SparsePop structure enter as:

```
        %e2 equation e2(t) +th1*e2(t-1)+
        %th2*e2(t-2)+...+th9*e2(t-9)-th10*Fry(t) -th11*Fry(t-1)
        %- th12*Fry(t-2)-... -th19*Fry(t-9)=0;

for t=10:N
            ineqPolySys{t-9}.typeCone = -1;
            ineqPolySys{t-9}.dimVar   = 39+2*N;
            ineqPolySys{t-9}.degree   = 2;
            ineqPolySys{t-9}.noTerms  = 20;
            ineqPolySys{t-9}.supports = [zeros(20,39+2*N)];
            ineqPolySys{t-9}.supports(1,t+39)=1;
```

This is the SparsePop structure for the equation 3.25, disturbance transfer function.

The values *ineqPolySys.typeCone=-1* because it is used for the equalities function, *ineqPolySys.degree=2* for a bilinear function which is the product between the parameters θ and output $_2$, *ineqPolySys.dimVar=39+2*N*. which defines always the size of the variables. Other parameters to set are *ineqPolySys.noTerms=20*, inequality composed by 20 terms. *ineqPolySys.coef=1* which defines that the term has coefficient 1, and *ineqPolySys.supports= [(zeros(20,39+2*N)]* which defines the structure of support matrix (size of matrix is the product between number of terms and number of dimVar).

```
ineqPolySys{t-9}.supports(1,t+39)=1;
ineqPolySys{t-9}.supports(2,[1,t-1+39]) = 1;
ineqPolySys{t-9}.supports(3,[2,t-2+39]) = 1;
ineqPolySys{t-9}.supports(4,[3,t-3+39]) = 1;
ineqPolySys{t-9}.supports(5,[4,t-4+39]) = 1;
ineqPolySys{t-9}.supports(6,[5,t-5+39]) = 1;
ineqPolySys{t-9}.supports(7,[6,t-6+39]) = 1;
ineqPolySys{t-9}.supports(8,[7,t-7+39]) = 1;
ineqPolySys{t-9}.supports(9,[8,t-8+39]) = 1;
ineqPolySys{t-9}.supports(10,[9,t-9+39]) = 1;
ineqPolySys{t-9}.supports(11,10) = 1;

ineqPolySys{t-9}.supports(12,11) = 1;
ineqPolySys{t-9}.supports(13,12) = 1;
ineqPolySys{t-9}.supports(14,13) = 1;
ineqPolySys{t-9}.supports(15,14) = 1;
ineqPolySys{t-9}.supports(16,15) = 1;
ineqPolySys{t-9}.supports(17,16) = 1;
ineqPolySys{t-9}.supports(18,17) = 1;
ineqPolySys{t-9}.supports(19,18) = 1;
ineqPolySys{t-9}.supports(20,19) = 1;

ineqPolySys{t-9}.coef= [ones(1,10),-Fry(t),-Fry(t-1),....
    -Fry(t-2), -Fry(t-3),-Fry(t-4),-Fry(t-5),...
    -Fry(t-6),-Fry(t-7),-Fry(t-8),-Fry(t-9)]';
end
```

ineqPolySys.coef defines the coefficients of terms, where is *=1* for the term with only variables, and *=Fry(t-i)* for term with direct dependence with signal wave input.*ineqoPolySys.support(i,j)=1* defines the positions in the support matrix of the variables.

Now it see the analysis for the SparsePop structure for the equation 3.22:

```
%e1 equation
   %e1(t) +th20*e1(t-1)+ th21*e1(t-2)+...+th28*e1(t-9)-th29*Te(t)
   %-th30*Te(t-1) - th31*Te(t-2)-.... -th38*Te(t-9)=0;

for t=10:N
        ineqPolySys{t-18+N}.typeCone = -1;
        ineqPolySys{t-18+N}.dimVar   = 39+2*N;
        ineqPolySys{t-18+N}.degree   = 2;
        ineqPolySys{t-18+N}.noTerms  = 20;
        ineqPolySys{t-18+N}.supports = [zeros(20,39+2*N)];
```

Same structure for this equality.

```
        ineqPolySys{t-18+N}.supports(1,t+39+N) = 1;
        ineqPolySys{t-18+N}.supports(2,[20,t-1+39+N]) = 1;
        ineqPolySys{t-18+N}.supports(3,[21,t-2+39+N]) = 1;
        ineqPolySys{t-18+N}.supports(4,[22,t-3+39+N]) = 1;
        ineqPolySys{t-18+N}.supports(5,[23,t-4+39+N]) = 1;
        ineqPolySys{t-18+N}.supports(6,[24,t-5+39+N]) = 1;
        ineqPolySys{t-18+N}.supports(7,[25,t-6+39+N]) = 1;
        ineqPolySys{t-18+N}.supports(8,[26,t-7+39+N]) = 1;
        ineqPolySys{t-18+N}.supports(9,[27,t-8+39+N]) = 1;
        ineqPolySys{t-18+N}.supports(10,[28,t-9+39+N]) = 1;
        ineqPolySys{t-18+N}.supports(11,29) = 1;
        ineqPolySys{t-18+N}.supports(12,30) = 1;
        ineqPolySys{t-18+N}.supports(13,31) = 1;
```

```
            ineqPolySys{t-18+N}.supports(14,32) = 1;
            ineqPolySys{t-18+N}.supports(15,33) = 1;
            ineqPolySys{t-18+N}.supports(16,34) = 1;
            ineqPolySys{t-18+N}.supports(17,35) = 1;
            ineqPolySys{t-18+N}.supports(18,36) = 1;
            ineqPolySys{t-18+N}.supports(19,37) = 1;
            ineqPolySys{t-18+N}.supports(20,38) = 1;

            ineqPolySys{t-18+N}.coef= [ones(1,10),-Te(t),-Te(t-1),...
                -Te(t-2),-Te(t-3),-Te(t-4),-Te(t-5),...
                -Te(t-6),-Te(t-7),-Te(t-8),-Te(t-9)]';
end
```

ineqPolySys.coef defines the coefficients of terms, where is *=1* for the term with only variables, and *=Tϵ(t-i)* for term with direct dependence with torque input.*ineqPolySys.support(i,j)=1* defines the positions in the support matrix of the variables.
Note that some of them are marked with *t-i+39+N*, where *N* is present, which identifies the sample number of the data collection. So it marks the position for the variable $\dot{\epsilon}_1$, present *N* times, referring to the collected data, because it defines that it is after *N+39* position of the matrix, i.e. present after *38* parameters, the bounded error, and *N* values of $\dot{\epsilon}_2$, placed in the previous equality.

Now it observes the expression in the SparsePop structure of the constraint inequality 3.36:

```
%output inequality     e1(t)+e2(t)-y(t)+deps>=0;
for t=10:N
            ineqPolySys{t-27+2*N}.typeCone = 1;
            ineqPolySys{t-27+2*N}.dimVar   = 39+2*N;
            ineqPolySys{t-27+2*N}.degree   = 1;
            ineqPolySys{t-27+2*N}.noTerms  = 4;
            ineqPolySys{t-27+2*N}.supports = [zeros(4,39+2*N)];
            ineqPolySys{t-27+2*N}.supports(1,t+39) = 1;
            ineqPolySys{t-27+2*N}.supports(2,t+39+N) = 1;
            ineqPolySys{t-27+2*N}.supports(4,39) = 1;

            ineqPolySys{t-27+2*N}.coef= [1,1,-P(t),1]';
end
```

The structure analyzes an inequality, so the value *ineqPolySys.typeCone=1*, as in the objective function. Moreover, the inequality shows the relationship between real and simulated output, in particular the simulated output of the two transfer functions. For this reason, in the *ineqPolySys.coef* it reports the output collected $\dot{e}$ to the position of the real outptut of the SparsePop structure.
The other parameters of the SparsePop structure are identical to the structures of the equations described above.

The expression in the SparsePop structure of the constraint inequality 3.37:

```
%output inequality     -e1(t)-e2(t)+y(t)+deps>=0;
for t=10:N
            ineqPolySys{t-36+3*N}.typeCone = 1;
            ineqPolySys{t-36+3*N}.dimVar   = 39+2*N;
            ineqPolySys{t-36+3*N}.degree   = 1;
            ineqPolySys{t-36+3*N}.noTerms  = 4;
            ineqPolySys{t-36+3*N}.supports = [zeros(4,39+2*N)];
            ineqPolySys{t-36+3*N}.supports(1,t+39) = 1;
            ineqPolySys{t-36+3*N}.supports(2,t+39+N) = 1;
            ineqPolySys{t-36+3*N}.supports(4,39) = 1;

            ineqPolySys{t-36+3*N}.coef= [-1,-1,edot(t),1]';
end
```

Identical to the previous structure, only the coefficients and their signs change in the parameter *ineqPolySys.coef*.

The last step to give in the SparsePop structure is to define the boundary of each single variable of the structure. As expressed below:

```
    lbd=-1e10*ones(1,39+2*N);
    ubd=-lbd;
end
```

The choice of lower and upper limits was arbitrary, because there was no constraint on that. That's why it was chosen to put a very large value (*1e10*), which simulates the value of infinity ∞ ($+\infty$ for upper bound (ubd) and $-\infty$ for

lower bound (lbd)).

Once the SparsePop structure is defined, it runs the script used to launch the SparsePop function created, to calculate the desired parameters of the two analyzed models:

```
load('d1_dot.mat')

N=101;
% for t=1:N
%     dep(t)=2*deps*rand-deps; %range scelto da me a priori
% end

    e_rid=edot(3001-N+1:3001);
    Te_rid=Te(3001-N+1:3001);
    Fry_rid=Fry(3001-N+1:3001);



   [objPoly,ineqPolySys,lbd,ubd] =min_th_data1P(N,e_rid,Fry_rid,Te_rid);
    param.relaxOrder = 1;
    param.POPsolver = 'active-set';
    [param,SDPobjValue,POP,elapsedTime,SDPsolverInfo,SDPinfo] = ...
        sparsePOP(objPoly,ineqPolySys,lbd,ubd,param);
    dep=POP.xVect(39);
    for j=1:38
    th(j)=POP.xVect(j);
    end

    %OPTIMIZATION

    depL=POP.xVectL(39);
    for j=1:38
    thL(j)=POP.xVectL(j);
    end
```

From this script it can see that only 100 data from the 3000 available from the collected data sample are taken for identification, to facilitate the calculation difficulty.

It also uses the SparsePop calculation optimization, it can see it from the parameter on the script *param.POPsolver='activeset'*.It was also used as a parameter *param.relaxorder=1*. This defines the relaxation order of the non-convex problem with which it is solved. Obviously the larger the order, the more complex the calculation will be, so it wanted to use *1* to define a low degree of complexity to best perform the calculations on the processor in use. By using this optimization it has to impose that the values of the desired variables are those of the *POP.xVectL*, obvious the values ranging from the position *1* to *38*.

3.2.3 Identification Results

By identifying the models for all 9 types of wave signals, and then for all 9 types of collected data, it can study and analyze them to understand how much error they individually have. To do this we define the error between real and simulated output. Real output is the one provided by collected data from previous projects, as analyzed in previous chapters. The simulated output is the one obtained by stimulating the model found with the input collected.
To define the simulated output, there is a need to build models. Models that depend on the parameters found. The parameters, as explained above, are optimized, but it is necessary to check between models with optimized and non-optimized parameters (use of 'active-set' tool), in order to understand if the identification has been successful.

The output is defined as the sum of the two outputs described by the two transfer functions G_1 and G_2. It is easier to understand if a simple Simulink model is shown where the model simulation is represented:

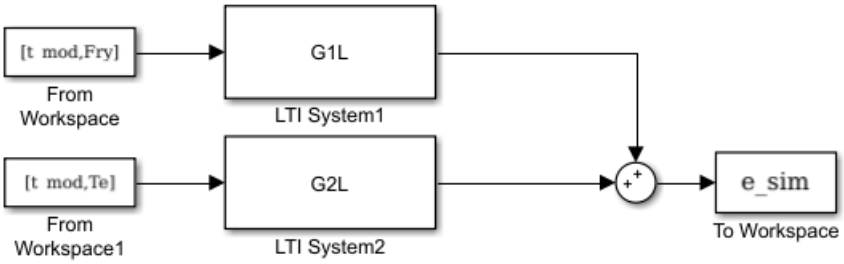


Figure 3.6 – Simulink simulation identificated model

A check is made between optimized and non-optimized model output. And to do this, take the first data set as reference:

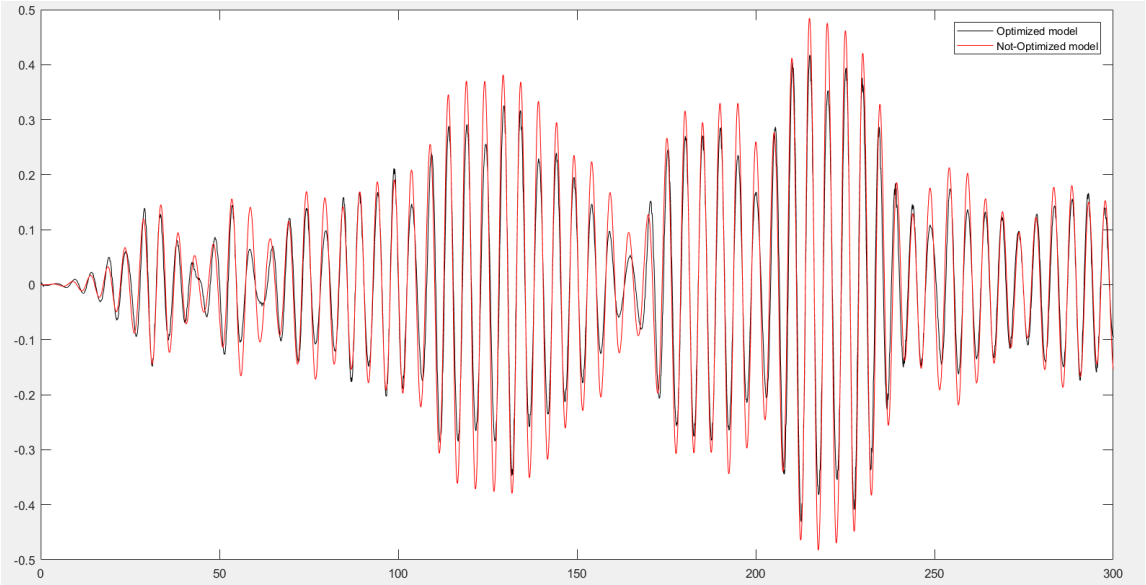


Figure 3.7 – Comparison between optimized and non-optimized models

Observable in more detail in the figure 3.7:

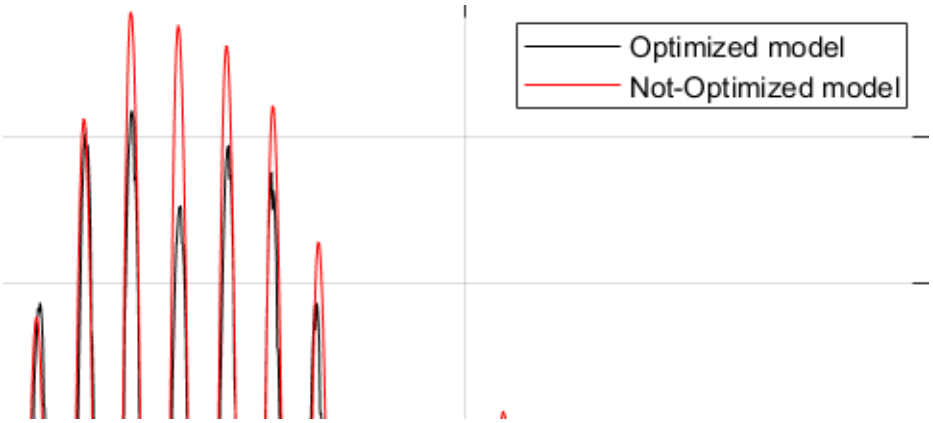


Figure 3.8 – Detail of comparison between optimized and non-optimized models

It is noted that there is a huge difference between the two outputs, the optimized and the non-optimized. This helps to understand that the use of identification optimization has led to a huge improvement.
To understand the actual improvement an analysis is made with the real output:

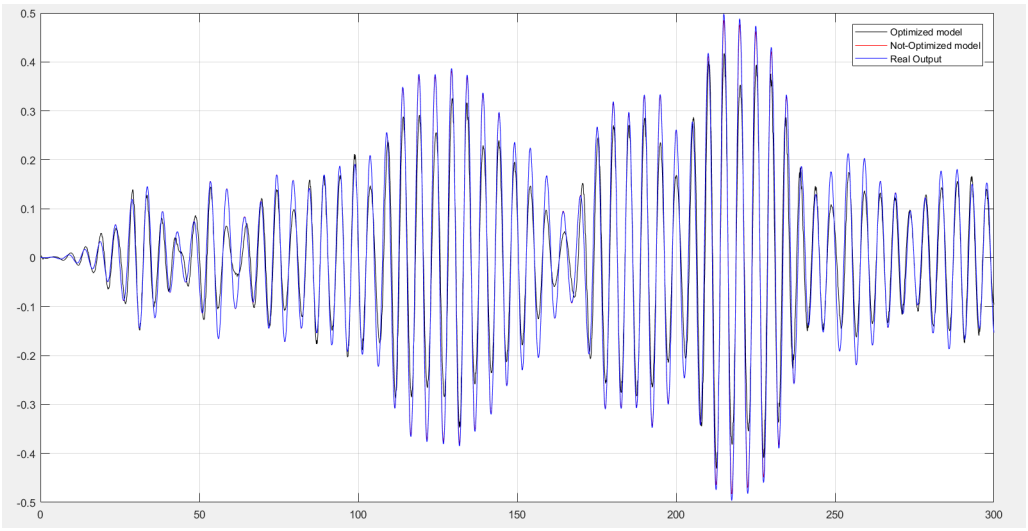


Figure 3.9 – Comparison between optimized, non-optimized models and real output

Observable in more detail in the figure 3.9:

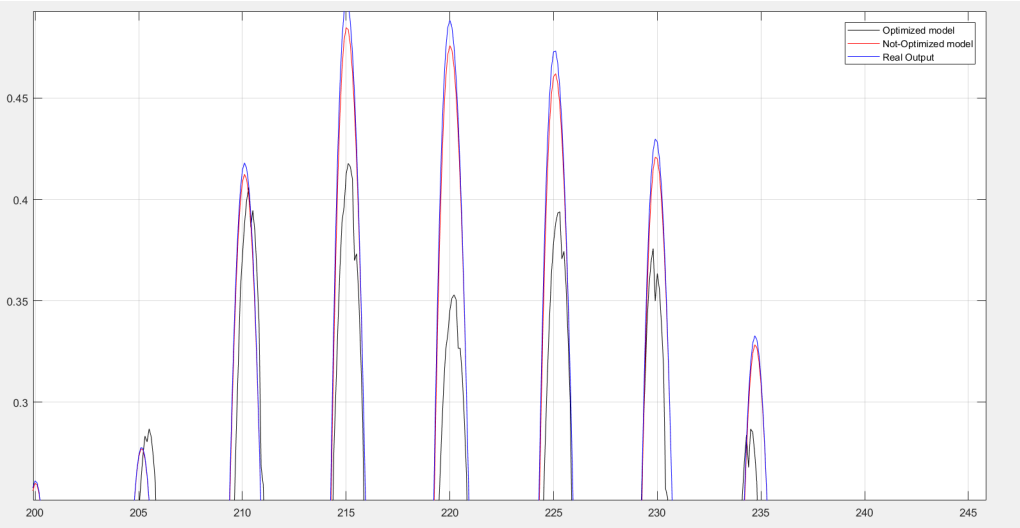


Figure 3.10 – Detail of comparison between optimized,non-optimized models and simulated output

The real output is highlighted with blue, and it can see that it similarly retraces the trend of the optimized output, unlike the non-optimized one which has a trend that differs from the real one. This means that the optimization has served, and that the models that will be analyzed will be based on the optimized identification (remember that it was done thanks to the 'activeset' function of SparsePop).

So it can see the effectiveness of the identification. The figure 3.9 represents the identification for the first set of data. It was necessary to make an identification for each of the 9 sets, to understand if the optimization was good for all of them.

Once an identification is made for each of the 9 sets, a table showing the effectiveness is represented, highlighting the RMSE (Root-mean-square deviation) error and the maximum error in the output peaks. The error that is represented is not defined on the angular velocity output, but on the power, since our goal is to create a model to simulate the power in order to perform a control that improves the efficiency.

Remember that the power calculation, as described in the chapter 2.4, depends

on the torque, collected data, input to the plant model, and the angular speed of the PTO shaft, a value analyzed and found as the real output of the model identified.

A comparison is made between real power, taken from real output, and simulated power, taken from simulated output from the model. And it can see it in the following figure:

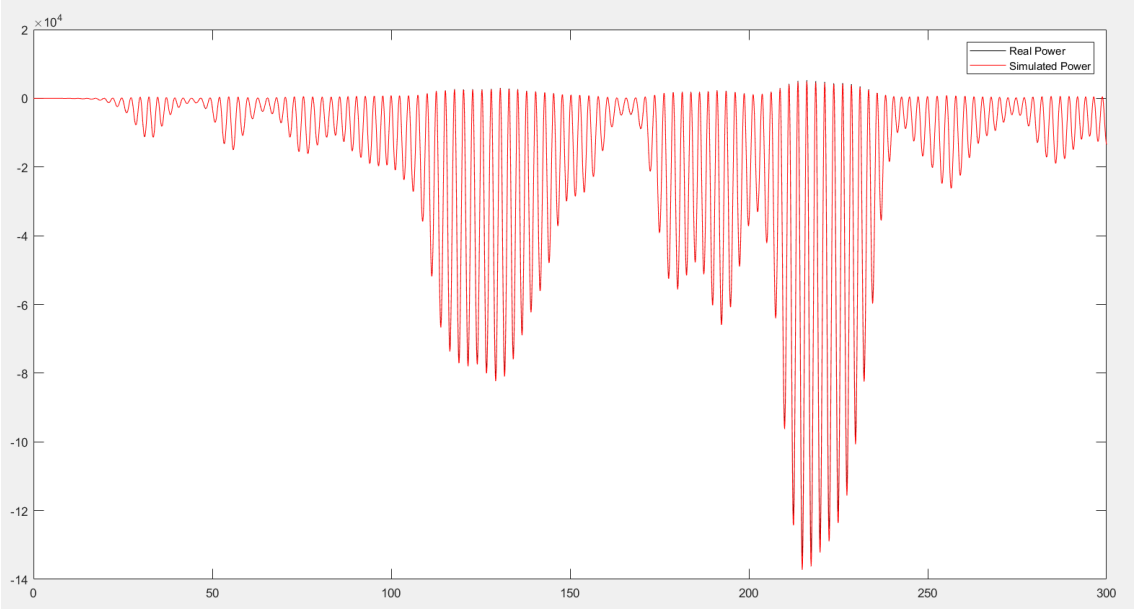


Figure 3.11 – Comparison between Real and Simulated Power

Observable in more detail in the figure 3.12:

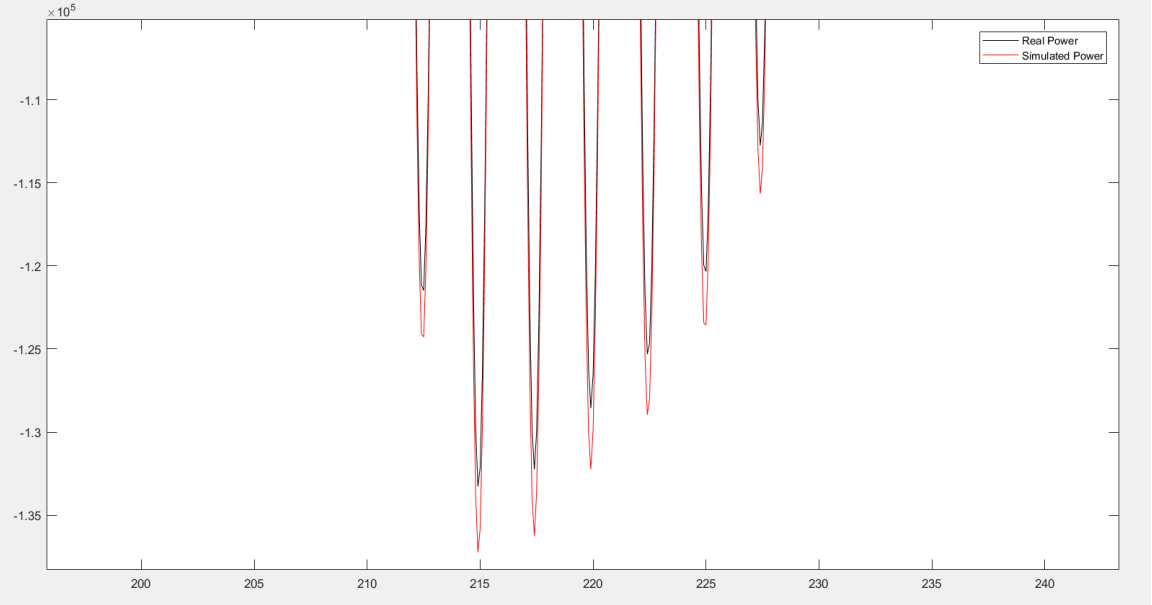


Figure 3.12 – Detail of peak of comparison between Real and Simulated Power

It can see from the picture that the error is minimal in the peaks. One more detail to say that the identification was successful, through the parameters found. This comparison, as already mentioned, was based on the first data set. A comparison is made on each of the 9 available data sets. A comparison based on the maximum error by observing the peaks, i.e. the maximum values of the power curve. But the analysis of this type of error is not very influential for a correct estimation. This is why an analysis of the RMSE error must be carried out.
This type of error is made considering the trend of the whole curve, and therefore considering the error on each single value of the real curve compared to the simulated one.
The RMSE error is calculated as follows:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}.$$

The RMSE error has been calculated for each of the 9 available data sets.
The comparison is shown first by analyzing the maximum peak error and then,

the RMSE error. The comparison that will be made shows the iteration of a model identified with a given data set with other data sets, to understand if it is possible to define a unique model that simulates each type of data set and therefore each type of wave signal entering the model.

The table that defines the maximum errors of the detected peaks for each of the 9 models and iteration with each data set is shown:

Application of identified models on data sets (maximum error of the Power)

Data Id.	1	2	3	4	5	6	7	8	9
1	4,22E+03	1,81E+04	42,790	43,3223	2,50E+03	3,25E+03	200,5619	3,99E+03	45,9446
2	4,34E+03	1,71E+04	128,009	101,2746	2,66E+03	2,62E+03	232,7325	4,10E+03	297,3074
3	5,76E+03	2,31E+04	104,9264	57,6591	3,24E+03	4,13E+03	373,433	5,05E+03	132,71
4	5,19E+03	2,11E+04	59,429	30,2543	2,93E+03	3,79E+03	312,5594	4,51E+03	81,8104
5	1,74E+03	9,85E+03	302,6183	205,1751	847,1296	992,839	582,0143	1,44E+03	478,5361
6	9,63E+03	3,95E+04	109,1173	53,4635	4,50E+03	3,49E+03	333,9577	5,68E+03	192,3616
7	6,10E+03	2,49E+04	100,0473	47,4802	3,32E+03	3,89E+03	343,4648	4,90E+03	127,0613
8	5,48E+03	1,25E+04	550,636	481,2786	3,33E+03	4,58E+03	805,5572	4,04E+03	1,13E+03
9	6,02E+03	2,41E+04	103,4762	59,2343	3,26E+03	4,20E+03	392,996	4,96E+03	121,2293
max(P abs)	1,37E+05	2,64E+05	1,76E+04	1,43E+04	1,01E+05	1,00E+05	3,25E+04	1,21E+05	2,27E+04

Red — Identified model

Yellow — best model for the data set

Table 3.1 – Table of maximum error

As said, the comparison of each individual model with each data set is made to identify a unique model that can best simulate each data set.
For each data set a better model has been found (note the yellow highlighting for the best one). Instead, the model calculated for that particular data set is written in red color. In order to understand how in some cases another model can cause a minor error and therefore a better identification.
This analysis is also valid for RMSE error.
Error to give more attention, as said before, for the fact that it analyzes every single value of the simulated output curve compared to the real one.

It shows the table 3.1, with a percentage expression:

Data	1	2	3	4	5	6	7	8	9
Id.									
1	3,1%	6,8%	0,2%	0,3%	2,5%	3,2%	0,6%	3,3%	0,2%
2	3,2%	6,5%	0,7%	0,7%	2,6%	2,6%	0,7%	3,4%	1,3%
3	4,2%	8,7%	0,6%	0,4%	3,2%	4,1%	1,2%	4,2%	0,6%
4	3,8%	8,0%	0,3%	0,2%	2,9%	3,8%	1,0%	3,7%	0,4%
5	1,3%	3,7%	1,7%	1,4%	0,8%	1,0%	1,8%	1,2%	2,1%
6	7,0%	14,9%	0,6%	0,4%	4,5%	3,5%	1,0%	4,7%	0,8%
7	4,4%	9,4%	0,6%	0,3%	3,3%	3,9%	1,1%	4,0%	0,6%
8	4,0%	4,7%	3,1%	3,4%	3,3%	4,6%	2,5%	3,3%	5,0%
9	4,4%	9,1%	0,6%	0,4%	3,2%	4,2%	1,2%	4,1%	0,5%
max(P abs)	1,37E+05	2,64E+05	1,76E+04	1,43E+04	1,01E+05	1,00E+05	3,25E+04	1,21E+05	2,27E+04

Table 3.2 – Table of maximum error in percentage

A further analysis carried out is to understand which model is the best. It is easy to understand that there is not a single model capable of being better for each data set, but there are two models (the first data set, and the fifth one) that are the best for more than two data sets. To understand which is the best of the two it need to understand what are the results of the RMSE error.

RMSE error									
Data	1	2	3	4	5	6	7	8	9
Id.									
1	6,11E+02	3,35E+03	9,17E+00	1,16E+01	2,87E+02	3,55E+02	2,01E+01	3,80E+02	1,09E+01
2	6,73E+02	3,10E+03	1,82E+01	2,24E+01	3,71E+02	3,77E+02	4,29E+01	4,23E+02	5,53E+01
3	7,40E+02	3,89E+03	7,15E+00	5,82E+00	3,81E+02	4,57E+02	3,60E+01	4,84E+02	1,86E+01
4	6,86E+02	3,70E+03	5,56E+00	4,87E+00	3,52E+02	4,31E+02	3,13E+01	4,54E+02	1,28E+01
5	3,94E+02	1,68E+03	4,55E+01	4,89E+01	1,98E+02	2,05E+02	8,99E+01	2,47E+02	9,40E+01
6	1,01E+03	4,77E+03	7,75E+00	6,63E+00	4,11E+02	4,37E+02	3,42E+01	4,87E+02	2,33E+01
7	7,13E+02	3,81E+03	7,10E+00	5,88E+00	3,64E+02	4,34E+02	3,44E+01	4,65E+02	1,70E+01
8	1,21E+03	2,32E+03	7,54E+01	1,08E+02	6,99E+02	7,48E+02	1,56E+02	7,86E+02	2,57E+02
9	7,36E+02	3,86E+03	7,43E+00	5,79E+00	3,77E+02	4,64E+02	3,80E+01	4,87E+02	1,79E+01
max(P abs)	1,37E+05	2,64E+05	1,76E+04	1,43E+04	1,01E+05	1,00E+05	3,25E+04	1,21E+05	2,27E+04

Table 3.3 – Table of RMSE error

It shows the table 3.3, with a percentage expression:

RMSE error									
Data	1	2	3	4	5	6	7	8	9
Id.									
1	0,44%	1,27%	0,05%	0,08%	0,28%	0,35%	0,06%	0,31%	0,05%
2	0,49%	1,17%	0,10%	0,16%	0,37%	0,38%	0,13%	0,35%	0,24%
3	0,54%	1,47%	0,04%	0,04%	0,38%	0,46%	0,11%	0,40%	0,08%
4	0,50%	1,40%	0,03%	0,03%	0,35%	0,43%	0,10%	0,37%	0,06%
5	0,29%	0,64%	0,26%	0,34%	0,20%	0,20%	0,28%	0,20%	0,41%
6	0,73%	1,80%	0,04%	0,05%	0,41%	0,44%	0,11%	0,40%	0,10%
7	0,52%	1,44%	0,04%	0,04%	0,36%	0,43%	0,11%	0,38%	0,07%
8	0,88%	0,88%	0,43%	0,75%	0,69%	0,75%	0,48%	0,65%	1,13%
9	0,54%	1,46%	0,04%	0,04%	0,37%	0,46%	0,12%	0,40%	0,08%
max(P abs)	1,37E+05	2,64E+05	1,76E+04	1,43E+04	1,01E+05	1,00E+05	3,25E+04	1,21E+05	2,27E+04

Table 3.4 – Table of RMSE error in percentage

It analyzes the model of the first data set and the fifth, and we find that although the first model is able to define minimum errors for some data sets, which the second model is not able to do (from the identification of the fifth data set). But this feature is not constant for all data sets, for example for the second data set it assumes an error of 1.27%, too big.
Instead, for the same data set, the fifth data set assumes a smaller error of 0.64%. Also it has to understand that the data sets are different from each other, and for example the fifth model assumes much larger values of power, in the order of $10^5 W$. And it find that the fifth model has the ability, as for the second data set, to define minimum errors for data sets capable of producing large power in the order of $10^5 W (= 100 kW)$, a feature not found for the first model. In fact, it is only able to define a minimum error only for data sets that generate power smaller than the order of $10^4 W (= 10 kW)$.

So it can say that the fifth model, even if assuming a bit bigger errors, is the best because it has a constant error in all 9 data sets, and unlike the first one it assumes smaller error values for data sets with more power.
For this reason the fifth model is evaluated as the best, since it can give even more priority to the datasets with more power, since the goal is to improve the efficiency of the system, and thus improve the power generated compared to

the power absorbed. So it is right to focus on datasets with more power generated.

Below it shows the transfer function of the fifth model (remember that the system model includes two, one for the disturbance, and one for the plant):

$$G_1(z) = \frac{-1.6462 \cdot 10^{-05}(z+0.8825)(z^2-1.958z+0.9999)(z^2-1.253z+0.5234)}{(z+0.7451)(z^2-1.638z+0.7177)(z^2-0.1205z+0.295)} \\ \frac{(z^2-0.949z+0.7326)(z^2+1.18z+0.9467)}{(z^2+1.172z+0.6663)(z^2-0.3242z+0.5984)} \tag{3.38}$$

,where $G_1(z)$ is the transfer function of the plant, expressed in discrete, Z-Domain.

It shows its expression 3.40 in continuous , in Time-Domain, so that the transfer function can be used in the control design:

$$G_1(s) = \frac{-1.6462 \cdot 10^{-05}(s-43.26)(s-0.798)(s^2+2.904s+11.95)(s^2+7.642s+127)}{(s^2+3.317s+9.49)(s^2+5.134s+191.5)(s^2+12.21s+250.3)} \\ \frac{(s^2-1.738s+431.2)(s^2+1.053s+999)}{(s^2+4.06s+566.4)(s^2+5.885s+995.6)} \tag{3.39}$$

The second transfer function is shown, that of the disturbance model:

$$G_2(z) = \frac{5.8069 \cdot 10^{-08}(z-0.9619)(z-1.01)(z-1.051)(z^2+1.996z+1.126)}{(z-0.7525)(z^2-1.969z+0.9837)(z^2+1.516z+0.6354)} \\ \frac{(z^2-1.061z+0.9271)(z^2+1.2z+1.067)}{(z^2+0.8057z+0.6857)(z^2-0.3268z+0.6004)} \tag{3.40}$$

It shows its expression 3.38 in continuous , in Time-Domain, so that the transfer

function can be used in the control design:

$$G_2(s) = \frac{5.8069 \cdot 10^{-08}(s-0.9926)(s+0.5759)(s-0.04332)(s^2-3.135s+116.6)}{(s+2.843)(s^2+0.1639s+1.456)(s^2+5.101s+191)} \\ \frac{(s^2+1.468s+549.3)(s^2-21.12s+1123)}{(s^2+3.773s+435.7)(s^2+4.534s+804.1)} \tag{3.41}$$

These are the models $G_1(s)$ and $G_2(s)$ identified to simulate the ISWEC system, which will be used in the next chapter for the design of the robust control to improve efficiency.

Chapter 4

Robust Control for ISWEC

The identified models have the function of simplifying the ISWEC system model. But, as mentioned in the previous chapters, the aim is to build a model designed to design a control to improve the power of the system in a simple way. To do this we try to design a preliminary Robust Control, since it is a simple computational mode, which compared to past designs, there is no need for any prediction that makes the calculation more complex. The Robust Control is designed considering the angular speed as a reference to follow, since the power generation mainly depends on it. And it is also designed to mitigate the disturbance, from which the model was also built.
This is the block tracking error, where shows that output power is generate by output of model (angular speed) and PTO torque (that is the input of model G_{pn}):

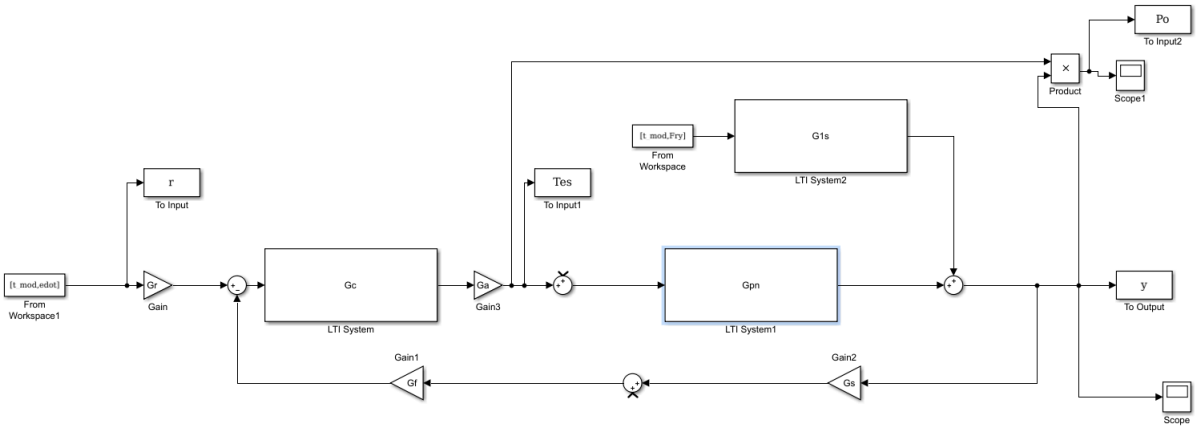


Figure 4.1 – Tracking error control

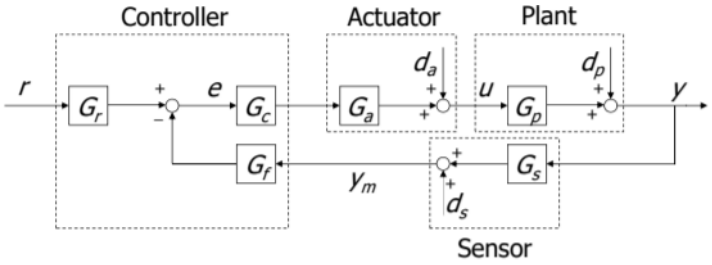
4.1 Introduction to Robust Control theory

In this chapter robustness for single-input single-output (SISO) linear time invariant systems is considered.

Irrespective of their complexity, mathematical models cannot exactly describe a real physical process. Sometimes we may prefer simplified approximate models, as the model identified.
Thus, model uncertainty has to be taken into account when a mathematical model is used to analyze the behavior of a system or to design a feedback control system.

Model uncertainty is essentially due to:
- physical parameters not exactly known;
- unmodeled (linear or nonlinear) dynamic;

Uncertainty due to approximate knowledge of some parameter values is called **parametric uncertainty**, instead a uncertainty due to unmodeled dynamics is called **dynamic uncertainty**.
The basic approach to take uncertainty into account is to describe the plant under study as a member of a set of systems (also called model set).
A LTI System is considered as in this figure:



Without loss of generality we assume that
$G_r = 1$
G_a = constant (large bandwidth because of fast dynamics)
G_f = constant (to handle the control system steady-state gain)
G_s = constant (large bandwidth because of fast dynamics)

Figure 4.2 – LTI System

Model sets for LTI uncertain systems can be classified as:

- **structured uncertainty model set**: when the set is parameterized by a finite number of parameters;
- **unstructured uncertainty model set**: when complete ignorance regarding the order and the phase behavior of the system is assumed;

Parametric uncertainty can also be described (with some conservativeness) by means of unstructured model sets. And there are different unstructured model sets to be considered:
- additive uncertainty;
- multiplicative uncertainty;
- inverse additive uncertainty;
- inverse multiplicative uncertainty;

The additive uncertainty model set is defined by:

$$M_a = \{G_p(s) : G_p(s) = G_{pn}(s) + W_u(s)\Delta(s)\,,\, \|\Delta(s)\|_\infty \leq 1\}$$

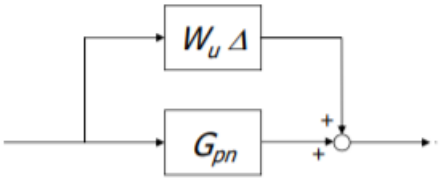


Figure 4.3 – Additive uncertainty

where $Gp(s)$ is the transfer function of the generic member of the uncertainty set, $Gpn(s)$ is the transfer function of the nominal model, $\Delta(s)$ can be any possible transfer function whose $H\infty$ norm is less than 1, $Wu(s)$ is a weighting function which accounts for the size of the uncertainty, and it is assumed that all systems belonging to Ma must have the same number of unstable poles.

By construction, the weighting function Wu must satisfy the following condition:

$$\left\| \frac{G_p(s) - G_{pn}(s)}{W_u(s)} \right\|_\infty = \|\Delta(s)\|_\infty \leq 1 \tag{4.1}$$

which is equivalent to:

$$\left| G_p(j\omega) - G_{pn}(j\omega) \right| \leq \left| W_u(j\omega) \right| \quad \forall \omega \tag{4.2}$$

The weighting function W_u has this slope, which is greater than the error between nominal and generic model:

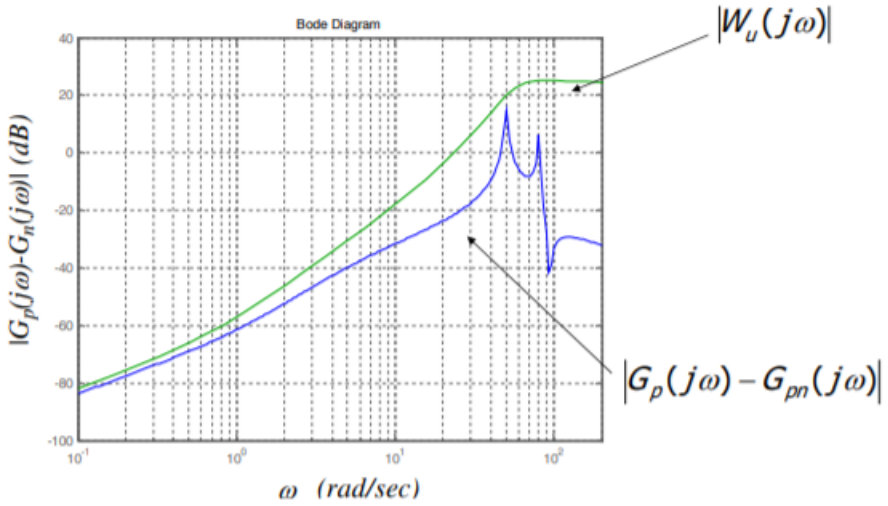


Figure 4.4 – Example of W_u of additive uncertainty

The multiplicative uncertainty model set is defined by:

$$M_m = \left\{ G_p(s) : G_p(s) = G_{pn}(s)[1 + W_u(s)\Delta(s)] \,,\, \|\Delta(s)\|_\infty \leq 1 \right\}$$

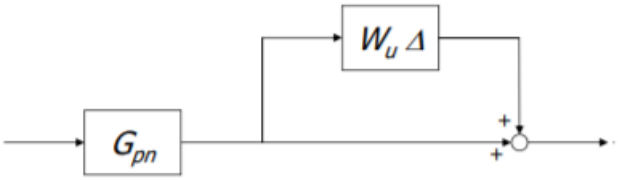


Figure 4.5 – Multiplicative uncertainty

,where, by construction, the weighting function W_u must satisfy the following condition:

$$\left\| \left(\frac{G_p(s)}{G_{pn}(s)} - 1 \right) \frac{1}{W_u(s)} \right\|_\infty = \|\Delta(s)\|_\infty \leq 1 \tag{4.3}$$

which is equivalent to:

$$\left| \frac{G_p(j\omega)}{G_{pn}(j\omega)} - 1 \right| \leq |W_u(j\omega)| \quad \forall \omega \tag{4.4}$$

The weighting function W_u has this slope, which is greater than the error between nominal and generic model:

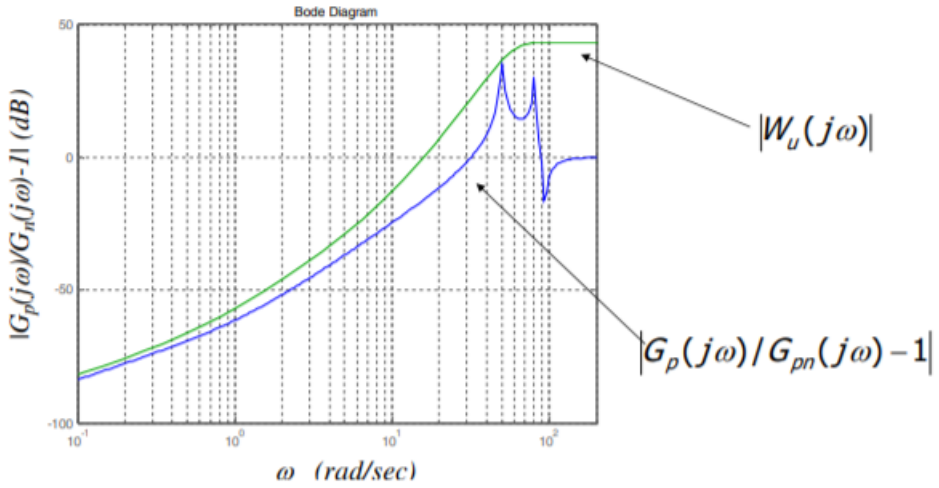


Figure 4.6 – Example of W_u of multiplicative uncertainty

The robustness of the system and analysis in an unstructured uncertainty is now considered. The robustness of the system and analysis in an unstructured uncertainty is now considered. This is called **Robust Stability**:
The feedback system in figure is robustly stable if and only if it is internally stable for each *Gp* which belongs to *Mm*.

Assume that *Gp* belongs to *Mm*. Assume that the feedback control system is stable when the nominal model *Gpn* for the plant is considered. The feedback system is robustly stable if and only if the following condition is satisfied:

$$\|W_u T_n\|_\infty < 1$$

Where *Tn* is the nominal complementary sensitivity function:

$$T_n = \frac{G_c G_{pn}}{1 + G_c G_{pn}}$$

Now it has recalled the **nominal performance** conditions (i.e.performance conditions in the uncertainty-free case) derived previously. Performance requirements affecting the sensitivity function leads to the following condition:

$$\|W_S S_n\|_\infty < 1 \Leftrightarrow |1 + L_n(j\omega)| > |W_S(j\omega)| \quad \forall \omega$$

,while performance requirements affecting the complementary sensitivity function are translated into:

$$\|W_T T_n\|_\infty < 1$$

From nominal performances it has defined the **Robust performances**. His definition is:
The feedback system guarantees robust performance if and only if performance requirement are satisfied for each *Gp* which belongs to the given uncertainty model set. So the robust performance if and only if the following condition is satisfied:

$$\| |W_s S_n| + |W_u T_n| \|_\infty < 1$$

This is the result that by definition, the feedback system guarantees robust performance if and only if:

$$\|W_s S\|_\infty < 1$$

where

$$S = \frac{1}{1 + L_n(1 + W_u\Delta)}$$

thus, it can write the following robust performance condition:

$$\left\| \frac{W_s}{1 + L_n(1 + W_u\Delta)} \right\|_\infty < 1 \tag{4.5}$$

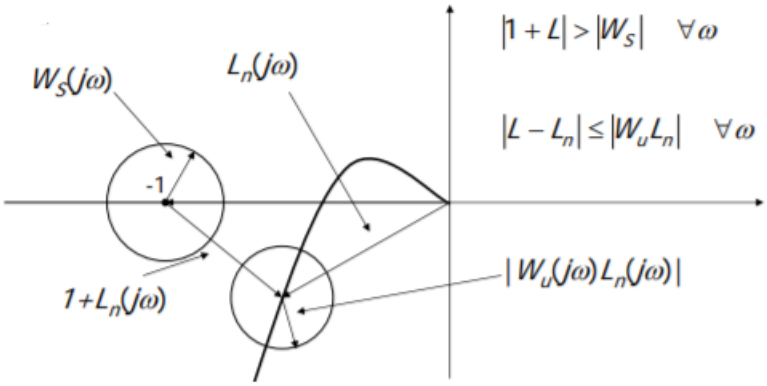
which, being $L = Ln(1 + W_u\Delta)$, can be equivalently written as:

$$|1 + L(j\omega)| > |W_S(j\omega)| \quad \forall\omega \tag{4.6}$$

Now it's considered the following relation straightforwardly derived from the definition of multiplicative uncertainty model set on the loop transfer function:

$$|L(j\omega) - L_n(j\omega)| \le |W_u(j\omega)L_n(j\omega)| \quad \forall\omega \tag{4.7}$$

From the following figure it is clear that the feedback system guarantees robust performance if and only if the two disks do not overlap:



the condition to avoid overlapping of the two disks can be formally written as:

$$
\begin{aligned}
&|W_S| + |W_u L_n| < |1 + L_n| \quad \forall \omega \\
\Leftrightarrow &\left|\frac{W_S}{1 + L_n}\right| + \left|\frac{W_u L_n}{1 + L_n}\right| < 1 \quad \forall \omega \\
\Leftrightarrow &\left\| |W_s S_n| + |W_u T_n| \right\|_\infty < 1
\end{aligned}
$$

Before talking about the $H\infty$ approach, it is useful to write where the weighting funcion on the sensitivy and complementary sensitivity comes from.
A weighting function is a weight function that keeps track of a function. It defines the low and high frequency amplitude limits of a function. Taking for example the sensitivity function, it can define a weight function on it.

$$
\left|W_S^{-1}(j\omega)\right| \le M_S^{LF} \quad \forall \omega_p \le \omega_p^+ \quad , \quad \max_\omega \left|W_S^{-1}(\infty)\right| \le S_{po}
$$

And it may take this form according to such limitations:

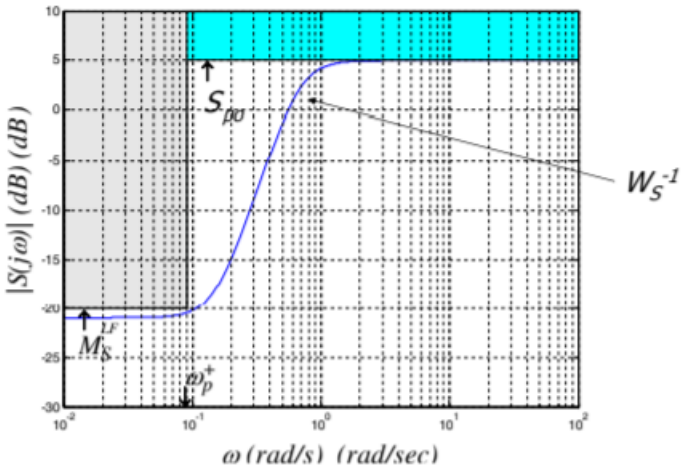


Figure 4.7 – Weighting function on the sensitivity example

The same analysis is carried out with regard to complementary sensitivity:

$$
\left|W_T^{-1}(j\omega)\right| \le M_T^{HF} \quad \forall \omega_s \ge \omega_s^- \quad , \quad \max_\omega \left|W_T^{-1}(j\omega)\right| \le T_{po}
$$

Its figure:

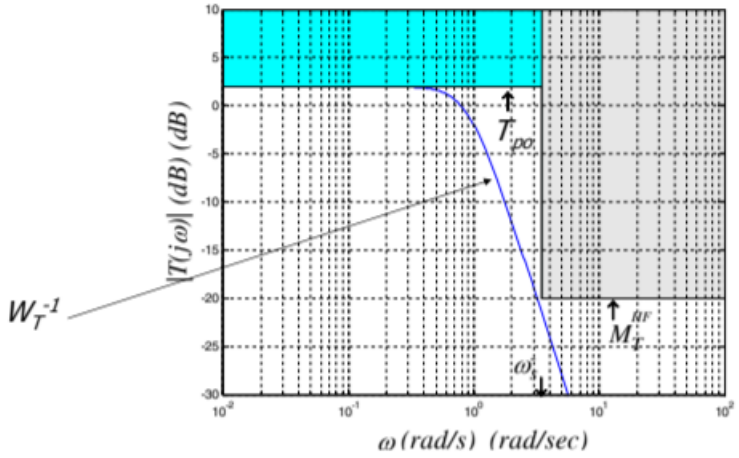


Figure 4.8 – Weighting function on the complementary sensitivity example

4.1.1 Introduction to $H\infty$ design for robust control

First of all, let us recall the definition of the $H\infty$ norm of a SISO LTI system with transfer function $H(s)$:

$$\|H(s)\|_\infty = \max_\omega |H(j\omega)| \tag{4.8}$$

The H norm minimization approach, called H control, refers to a general formulation of the control problem which is based on the following block diagram representation of a general feedback system.

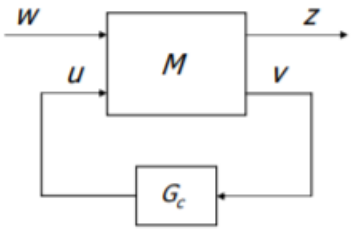


Figure 4.9 – Robust control H_∞ representation

,where M is the *generalized plant*, G_c is the controller, u are the control inputs, v are the controller inputs, w are the external inputs and z are the external outputs. The external input and output signals of the generalized plant are not necessarily physical variables of the control system. The external input and output signals of the generalized plant must be carefully selected in order to take into account the stability/performance requirements of the considered control problem.

In the H design, the controller is obtained by solving the following optimization problem:

$$G_c(s) = \arg \min_{G_c \in G_c^{stab}} \|T_{wz}(s)\|_\infty \tag{4.9}$$

,where T_{wz} is the closed loop transfer function between the input w and the output z.

It consider the problem of designing a controller G_c to robustly stabilize an uncertain system described by means of the unstructured multiplicative model set (look equation of figure 3.3) and $||W_u T_n||_\infty < 1$. The problem can be solved chosen:

$$T_{wz}(s) = W_2 T_n$$

where

$$W_2(s) = W_u(s)$$

Assume, without loss of generality, that: $Gf = Gs = Ga = 1$.

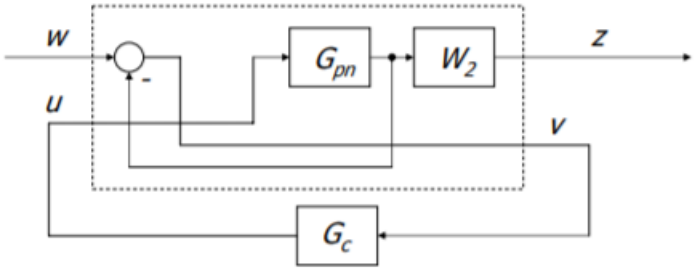


Figure 4.10 – Generalized Plant *M*

Now it consider the problem of designing a controller *Gc* to satisfy the nominal performance conditions:

$$\|W_S S_n\|_\infty < 1 \quad , \quad \|W_T T_n\|_\infty < 1$$

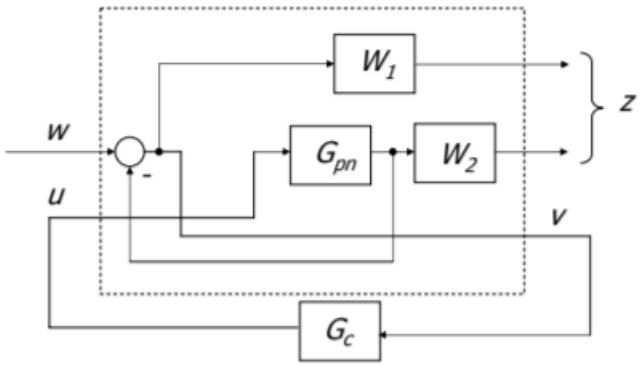
can be solved by choosing the following transfer function T_{wz}:

$$T_{wz}(s) = \begin{bmatrix} W_1 S_n \\ W_2 T_n \end{bmatrix}$$

where

$$W_1(s) = W_S(s)$$
$$W_2(s) = W_T(s)$$

So, considering these conditions, the generalized plant is:



The generalized plant M can be internally stabilized by an LTI controller Gc having input v and output u if and only if W_1 and W_2 are stable transfer functions. However, it knows that some common performance requirements on the steady-state response to polynomial reference signals and disturbances lead to an unstable weighting function W1 (due to the presence of one or more poles at s=0).
In order to satisfy the assumption of stability it replaces W_1 in the generalized plant M with a new weighting function W_1* obtained as follows:

$$W_1^*(s) = W_1 \frac{s^{v+p}}{(s+\lambda^*)^{v+p}}$$

where $\lambda* > 0$ is a low frequency pole (a reasonable choice for $\lambda*$ is $\lambda* \leq 0.01\omega_c$). The controller obtained with such a modified weighting function will have, if any, at most +p poles at about $s = -\lambda*$. Each pole at about $s = -\lambda*$ in the controller must be replaced with a pole at $s = 0$.

For W_2 it has to make a choice, in fact as explained above, it depends on W_t and W_u.The choice is based on choosing W_2 as the maximum between W_t and W_u, as can be seen in the following figure:

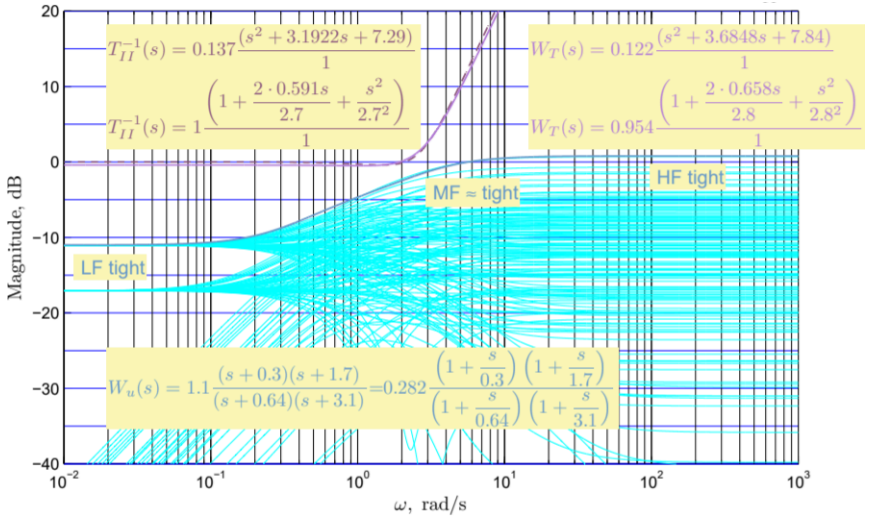


Figure 4.11 – W_t e W_u weighting function example

The maximum choice between the two weighting functions is W_2.

The problem is that it has to make the generalized plant M stable, and to do this the weighting function W_2 must be stable too, but it can see from the previous figure that it is an improper function. So it makes a choice. In the generalized plant you put only its dc gain(W_{2mod}), and forcefully add the poles. Poles that will be the cut-off frequency of the W_2, (added with the function *sderiv*).

4.2 $H\infty$ approach for ISWEC model

The first step for the $H\infty$ approach is to define the weighting functions, both uncertainty and sensitivity and complementary, such as to build control.
The uncertainty weighting function W_u is calculated based on the 9 models calculated by identification. In fact, as mentioned in the previous chapter, the W_u is based on the error between the nominal model and the other general models of the system. The fifth model is identified as the nominal model, and the other 8 models as general models. The multiplicative uncertainty model set is used as uncertainty calculation, so the error will be based on the ratio between nominal model and general models, as written in equation 4.4.
Then the relative error is calculated eight times, one for error with any 8 general models, then eight possible weighting functions are calculated, and the W_u is chosen as the maximum. It can see the superpostion of the 8 weighting functions on the following graph:

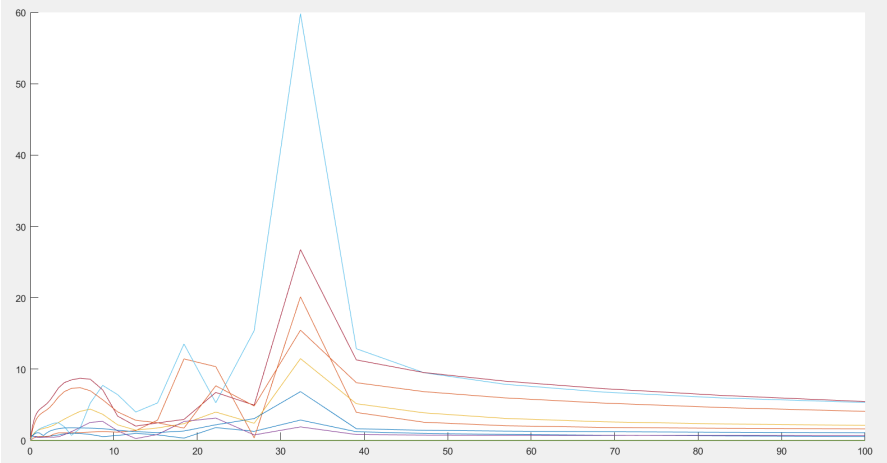


Figure 4.12 – W_u weighting function superposition

Through the *ginput* Matlab function it takes several points on the graph where it takes the maximum at each point on the function overlap. In this way it builds the W_u, defined:

$$W_u = 5.5721 \frac{s^2 + 28.86s + 987.2}{s^2 + 2.942s + 1017} \tag{4.1}$$

Once it defines the W_u construction, it has to build the W_t and W_s to define the W_1 and W_2, for the $H\infty$.
The W_t and W_s are built based on the information it has from the system and that we try to impose. For example, we try to define an error to infinity of 0.02 ($err_{inf} = 0.02$). And a sinusoidal reference that it imposes, is 0.5, which corresponds to the maximum angular velocity reference of the 9 data sets ($R_0 = 0.5001$).

These parameters are used to evaluate the a parameter of the Butterworth function, which is used to build the W_s weighting function:

$$W_s^{-1} = \frac{as(1 + \frac{s}{w_1})}{1 + 1.414\frac{s}{w_2} + (\frac{s}{w_2})^2} \tag{4.2}$$

This a parameter is defined by resolving the limit:

$$\begin{aligned}
&\left|e_r^{\infty}\right| = \lim_{t \to +\infty} \left|e_r(t)\right| = \lim_{s \to 0} s\left|e_r(s)\right| = \lim_{s \to 0} s \mid K_d r(s) - y_r(s) \mid = \\
&= \lim_{s \to 0} s\left|G_{re}(s) r(s)\right| = \lim_{s \to 0} s\left|S(s) K_d r(s)\right| = \\
&= \lim_{s \to 0} s\left|s^{\nu+p} S^*(s) K_d \frac{R_0}{s^{h+1}}\right| = \begin{cases} 0 \leftarrow (\nu + p > h) \\ \left|S^*(0) K_d R_0\right| \leftarrow (\nu + p = h) \end{cases}
\end{aligned}$$

,where the a is calculated as S_{star0}. From this limit it can see that:

$$S_{star0} = \frac{er_{\infty}}{K_d R_0} \tag{4.3}$$

,where K_d is the gain imposed on the reference (constant scaling factor), and *=1* is imposed. So a is given by the ratio of e_{rinf} and R_0. Value equal to *=0.04*.

Other parameters to be defined are the frequencies w_1 and w_2 . The w_2 depends strictly on w_1 and a, so let's focus on the choice of w_1. It depends on the cut-off

frequency that it wants to give on the open loop function L_n. To calculate it, it relies on the settling time dependent calculation:

$$t_{s,\alpha\%} = -\frac{\ln \alpha}{\omega_n \zeta} \qquad \omega_c = \omega_n \sqrt{\sqrt{1 + 4\zeta^4} - 2\zeta^2}$$

The inverse calculation of settling time equation defines w_n and thus the calculation of w_{cts}. After defining this parameter, it is useful to understand that w_1 is empirically less than half of w_{cts}. w_{cts} for the calculation is equal to *0.1468*. So this defines that the minimum cutting frequency is *0.1468*. In fact, remember that this calculation does not exactly define the w_c, but only a minimum of the range.

w_2, as mentioned above, depends on w_1 and a, as:

$$w_2 = \sqrt{\frac{S_{p0} w1}{a}} \tag{4.4}$$

S_{p0} is the maximum peak in sensitivity amplitude that is imposed per project on *1.4*. T_{p0} is the peak for the complementary sensitivity that is imposed on *1.05*. So for the calculation w_2 is equal to *1.5652*. This builds the W_s, which we will be modified for optimal control calculation.

Now the focus shifts to the W_t calculation, which is calculated as:

$$W_t^{-1} = \frac{T_{p0}}{1 + 1.414\frac{s}{w_t} + (\frac{s}{w_t})^2} \tag{4.5}$$

The calculation of the w_t (complementary sensitivity cut-off frequency) is very empirical. In fact, since it has no information on complementary sensitivity constraints. It will be considered a one decade than the w_s cut-off frequency.

After building the main weighting functions, you have to build the weighting functions for the generalized plant. W_1 as in the previous paragraph, is based on the W_s and the deletion of the pole at 0 and the replacement of a fairly small pole. The choice of the *(s+0.01w$_c$)* pole, where w_c is the system's cut-off frequency, is chosen as the average of the previously calculated w_{cts} and w_t.

The construction of the W_s depends on the choice of a and w_1. Now a is imposed by the choices made previously (*0.04*). But that doesn't mean that a is a fixed value, it can be modified to build a W_s appropriate to the control design. We empirically noted two types of weighting function W_s, one capable of tracking angular velocity output but not capable of generating and improving power, another capable of improving power but not tracking output. We analyze the first one, the one capable only of chasing the output:

$$W_s^{-1} = \frac{1.4s(s + 7e - 05)}{(s^2 + 0.05715s + 0.001633)} \tag{4.6}$$

Here there is the choice of $a = 0.06$ and $w_1 = 0.00007 rad/s$.

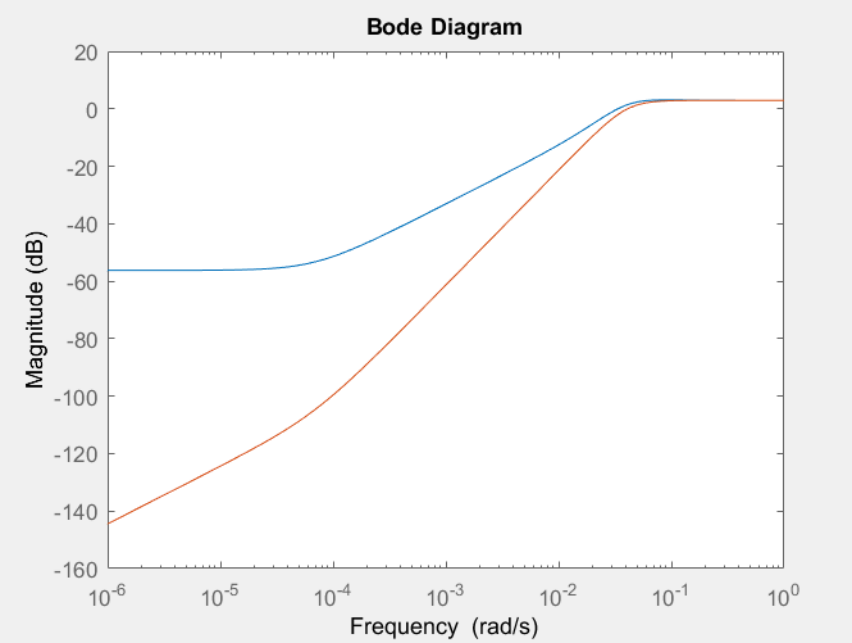


Figure 4.13 – W_s and W_1 weighting function

The other, the one capable to improve the power is:

$$W_s = \frac{1.4s(s + 0.7)}{(s^2 + 9.898s + 49)} \tag{4.7}$$

Here there is the choice of $a = 0.02$ and $w_1 = 0.7 rad/s$.

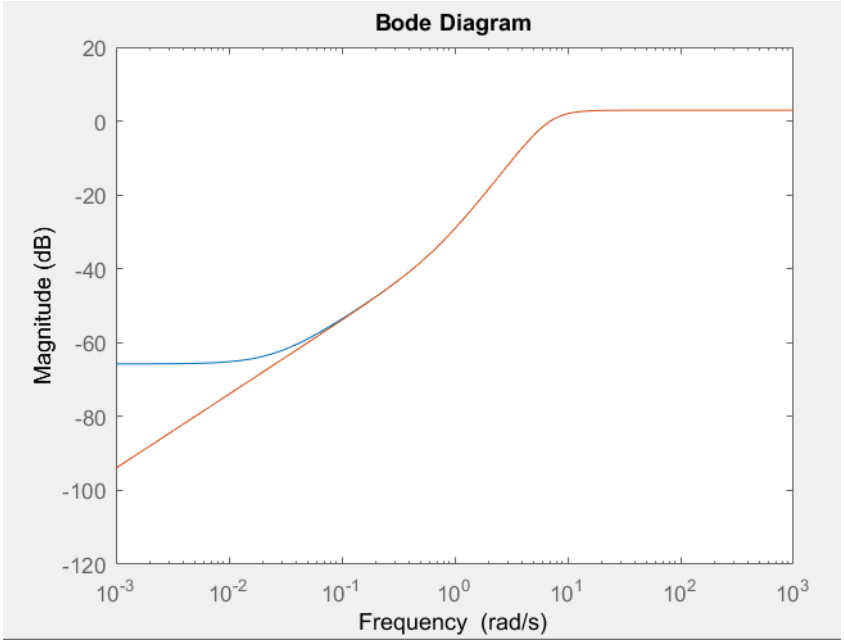


Figure 4.14 – W_s and W_1 weighting function

For both it chooses a w_c that is the average between w_{cts} and w_t, where w_t will be a chosen value of *5 rad/s*, a choice that will be motivated later with the superposition of W_u.

The construction of W_2, as mentioned in the previous paragraph, is calculated as the maximum between W_t and W_u. Through this superposition it is also possible to define the w_t cut-off frequency of the complementary. Because if there is an overlay of W_u on W_t, the cut-off frequency w_t to be imposed in the *sderiv* for the control calculation changes, since it must be adjusted to W_2. To practically understand it, it has to take the W_u and see what the compare with W_t. Now see an example, bearing in mind the construction conditions of the W_t, said on the previous page.

Considering the W_u calculated before, and defined the W_t with the frequency w_t of *10 rad/s* we define the W_2 considering the maximum superposition of the two weighting functions:

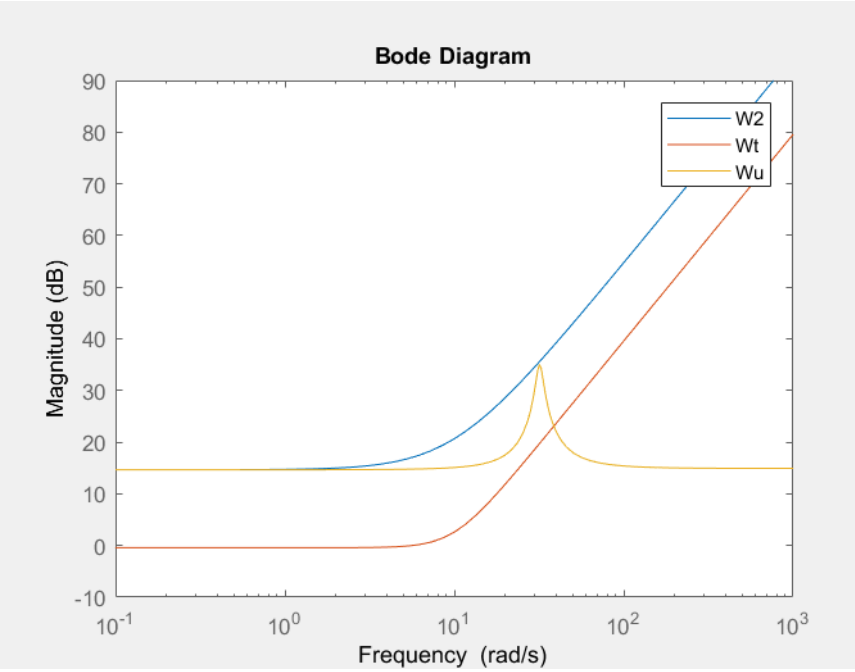


Figure 4.15 – W_t and W_u weighting function

Note that the choice of the w_t of *10 rad/s* was correct, since the superposition with the W_u, except for a small overlap, is not a problem except for the dcgain part. In fact you can see that the W_u cuts W_t in the dcgain part (constant part of the function). The dcgain of W_t is *0.9524*, that of W_u is *5.4091*. This is very important because it defines the W_{2mod} (see previous paragraph).

So with the calculation of W_1 and W_2 it can define the control. Obviously two W_s were defined first, and then two W_1. Then you will have two different controls. They will be:

-For the first weighting function W_s 4.1:

$$G_c = \frac{10.516(s+1.099e05)(s+0.05091)(s^2+3.317s+9.491)}{(s+135.5)(s+8.511)(s+3.715)(s+0.05073)(s+7e-05)(s^2+2.922s+11.95)} \\ \frac{(s^2+5.134s+191.5)(s^2+12.21s+250.3)(s^2+4.06s+566.4)(s^2+5.885s+995.6)}{(s^2+7.641s+126.9)(s^2+4.474s+432.8)(s^2+1.053s+999)} \tag{4.8}$$

-For the second weighting function W_s 4.2:

$$G_c = \frac{-1.4325e06(s+1e05)(s+1.071)(s^2+3.317s+9.49)}{(s+2.798e04)(s+0.7)(s+0.05073)(s^2+2.904s+11.95)(s^2+7.642s+127)} \frac{(s^2+5.134s+191.5)(s^2+12.21s+250.3)(s^2+4.06s+566.4)(s^2+5.885s+995.6)}{(s^2-1.61s+409.6)(s^2+1.053s+999)(s^2-24.77s+4538)} \tag{4.9}$$

4.3 Robust Control Results

As mentioned in the previous paragraph, the first control (4.8) has the function of following the reference signal, i.e. the angular speed, but is not capable of generating a high power, so much so as to improve or even emulate the initial power. For this case it is right to entrust only an analysis for a given set, since it does not improve the power.

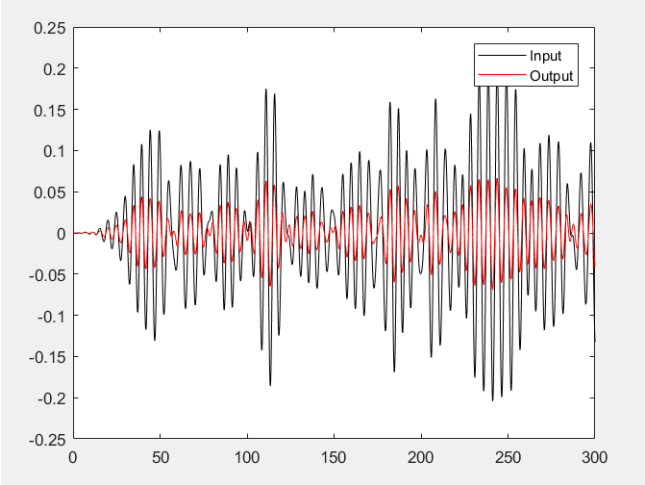


Figure 4.16 – Output and Input reference of tracking error control

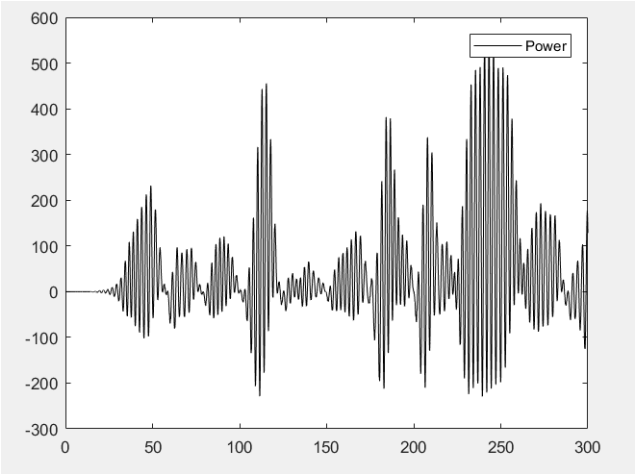


Figure 4.17 – Output power of control

It is possible to notice that the power is generated very far from our goals, but the control is able to follow the reference input even if it is very attenuated. A result that we have not been able to improve. These two graphs were made on the ninth data set, where the power is in the order of $10^4 W$.

Let's shift our attention completely to the second control, which is capable of generating emulating the initial power for each data set.
An analysis is always performed on the ninth data set. These are the graphs that show the input trend that is not followed by the output and the power generated by it:

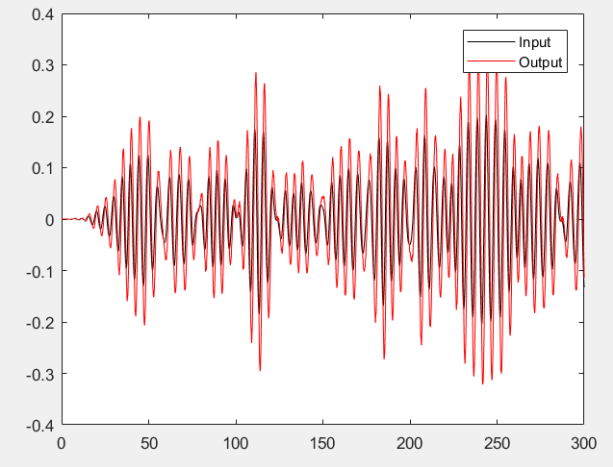


Figure 4.18 – Output and Input reference of tracking error control

It can see in the ninth data set that the control is not able to follow the reference input. This means that the control enters a phase of delay with the reference, even if it is able to follow and understand the trend of it. But there is a non-negligible delay, which however it can notice defines greater powers than the previous control. it can now see the graph of the generated power:

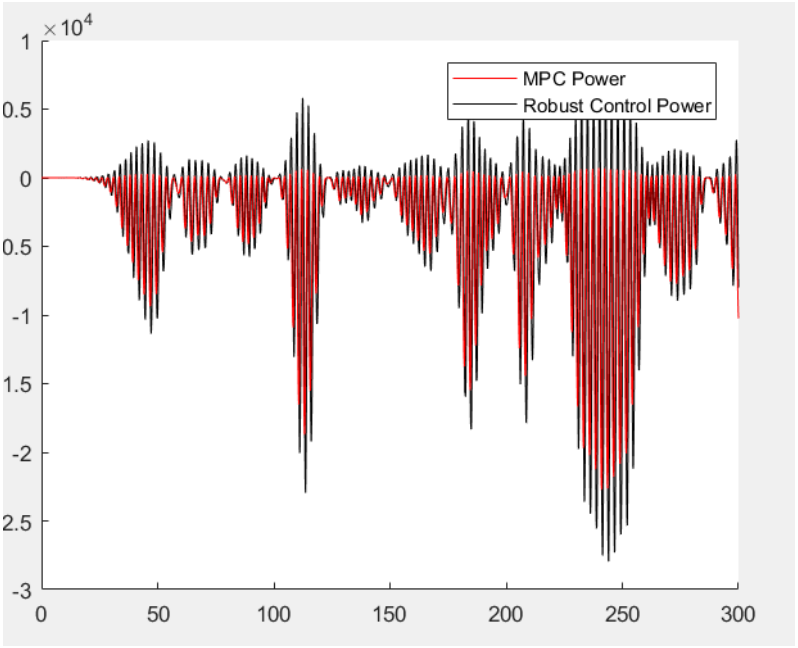


Figure 4.19 – Output power of control

Reproduced powers in the same order of the initial reference power of the ninth data set. This means that the system is able to generate higher powers but without being able to follow the reference. It test the controller for all nine data sets.

Data	1	2	3	4	5	6	7	8	9
MPC Pow. (max)	1,37E+05	2,64E+05	1,76E+04	1,43E+04	1,01E+05	1,00E+05	3,25E+04	1,21E+05	2,27E+04
Rob. Pow. (max)	1,82E+05	4,07E+05	2,13E+04	1,75E+04	1,34E+05	1,26E+05	3,80E+04	1,60E+05	2,79E+04

Table 4.1 – Table of maximum of power

As can be seen from the table 4.2 larger powers are generated, but by making them less stable, in fact, positive powers are also generated that are not negligible.

In fact you can see from the following table that the average power is lower than the reference power of the MPC project.
Power generated according to this formula:

$$\overline{P} = -\frac{1}{n_s}\sum_{k=1}^{n_s} P(k)$$

The table showing the average power is as follows:

Average Produced Power									
Data	1	2	3	4	5	6	7	8	9
MPC Pow. (avg)	1,57E+04	3,84E+04	1,24E+03	1,61E+03	1,15E+04	1,12E+04	2,81E+03	1,30E+04	-2,83E+03
Rob. Pow. (avg)	1,46E+04	3,67E+04	1,20E+03	1,56E+03	1,12E+04	1,09E+04	2,66E+03	1,25E+04	-2,66E+03
Gap(%)	-7%	-4%	-3%	-3%	-2%	-2%	-5%	-4%	-6%

Table 4.2 – Table of average of power

It can be noted that compared to the MPC reference design, the average power generated by Robust Control is minimal compared to the reference power. You can see that you reach minimum errors of 2% as in data sets 5 and 6, and maximum errors of 7% as in data set 1. This means that the robust control did not improve the average power generated, because it generates positive power which, as it can see, even if it generates higher maximum power, lowers the average power.
Losses of power not too high, but not negligible.

What it has analyzed is the result of a Robust control, and as noted in section 4.1, a control to be considered robust must meet certain conditions, such as Robust Stability, Nominal Performances and Robust Performances. These conditions shall be analysed according to paragraph 4.1.

The Robust Stability:

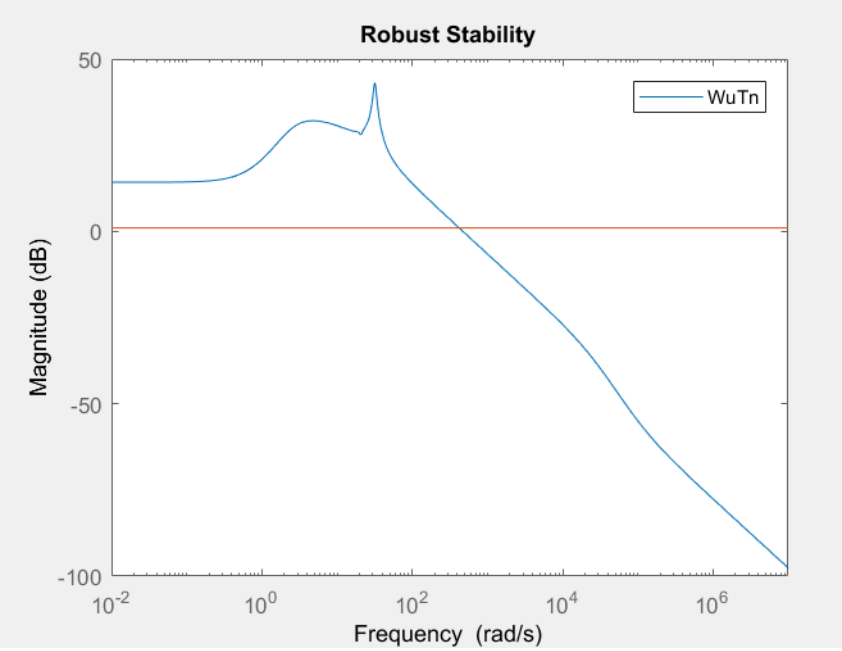


Figure 4.20 – Robust Stability

The Nominal Performance:

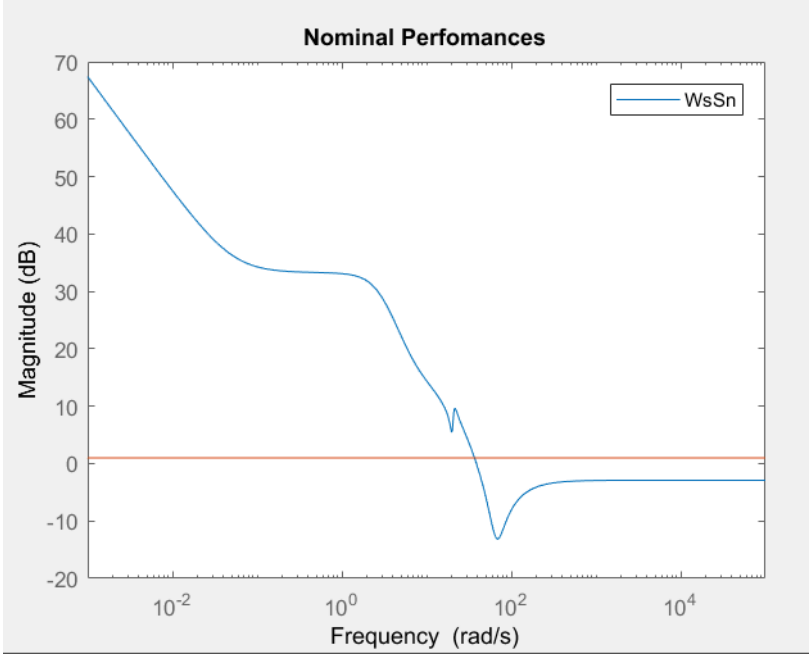


Figure 4.21 – Nominal Performance

The Robust Performance:

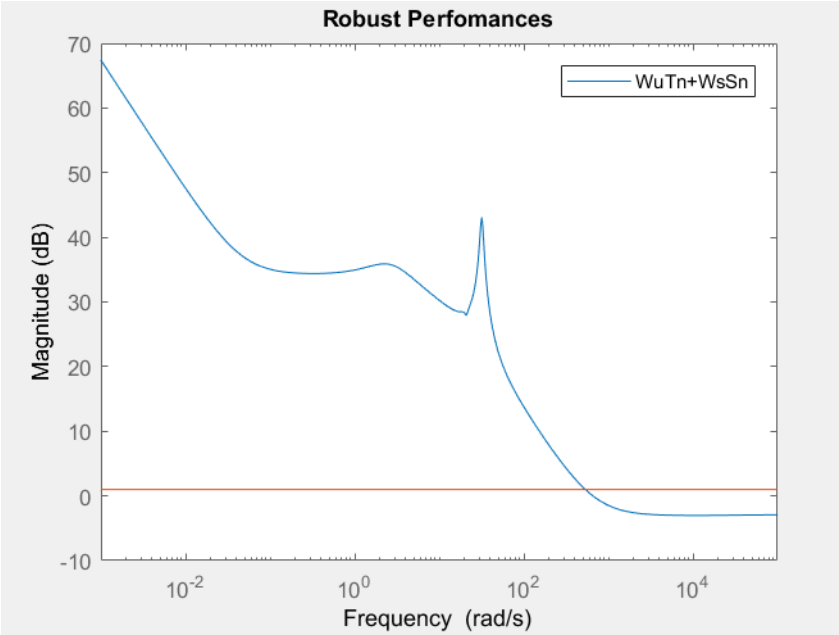


Figure 4.22 – Robust Performance

It can be observed that all the robustness conditions are not respected, in fact all the values are over 1. And they are not respected so much, in fact it comes to maximum values of 65.9 for Robust and Nominal Performances, and 43 for Robust Stability. This non-respect of conditions confirms that the control cannot be defined as robust, even if it almost succeeds in reproducing the power of old designs such as MPC. A next goal for a next application is to improve the control by making it robust (i.e. respecting the main conditions of robustness).

Chapter 5

Conclusions

This thesis showed the effects of identification on an ISWEC (Inertia Sea Wave Energy Converter) device. An identification that simplifies the most complex mechanical model for the design of a Control, for example a Robust Control.

The identification made using Set-Membership theory, with the SparsePop technique used for its bilinear complexity, shows positive results. Even more promising if it uses optimization tools, able to reduce the error. In fact, the models identified for each data set have small errors compared to the mechanical model.

Each model defines a minimum error for its own data set, but this is also positively guaranteed for other data sets. In fact, when trying to use the models for each data set, there are always minimum errors.

This means that the identification is derived from the goal of computing models for each data set, capable of being applied to different inputs, in addition to those used as reference in the identification.

By checking all the models identified on each data set, it has the possibility to define a model capable of defining an output with an error below 1% for each available data set.

This model is applied in a Robust Control. Control used for its feature of

using less computational work on the controller, unlike the MPC controller used in older projects.

The Robust Control was able to replicate the average power generated by the old controls, but with a small margin error. But it has not been possible to define a Control that respects the conditions of robustness, that is to say that it is able to attenuate any kind of disturbance and uncertainty that is affected a system.

Future Development

The model identified was believed to be capable of bearing any type of data set, and therefore wave signals. But in the preliminary study of a Robust Control it was not possible to improve the power generated in the conversion of mechanical energy into electrical energy.

For the next applications it would like to define a control starting from this simplified model able to improve the power, ensuring a robust control so that it is an effective control for any kind of disturbance and system uncertainty.

It is always advisable to use a Robust Control to continue with the simplification of the system. After simplifying the model it is correct to use a Robust Control that simplifies the computational effort of the control respect than the predictive control of the MPC controller.

Bibliography

[1]D. Magagna and A. Uihlein, "Ocean energy development in europe: Current status and future perspectives," *International Journal of Marine Energy*, vol. 11, pp. 84–104, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214166915000181.

[2]J. V. Ringwood, G. Bacelli and F. Fusco, "Energy-maximizing control of wave-energy converters," *IEEE Control Systems Magazine*, vol. 34, no. 5, pp. 30–55, 2014.

[3]T. Aderinto and H. Li, "Ocean wave energy converters: Status and challenges," *Energies*, vol. 11, p. 1250, 05 2018.

[4]S. Salter, "The use of gyros as a reference frame in wave energy converters," *Proceedings of the 2nd International Symposium on Wave Energy utilization*, pp. 99–115, 06 1982.

[5]F. Salcedo, P. R. Minguela, R. R. Arias, P. Ricci, and M. Santos-Mugica, "Oceantec: Sea trials of a quarter scale prototype," 01 2009.

[6]H. Kanki, S. A. ans T. Furusawa, and T. Otoyo, "Development of advanced wave power generation system by applying gyroscopic moment," 7 2009, pp. 280–283.

[7]G. Bracco, E. Giorcelli, and G. Mattiazzo, "Iswec: A gyroscopic mechanism for wave power exploitation," Mechanism and Machine Theory, vol. 46, no. 10, pp. 1411–1424, 2011, cited By 41.

[8]M. Raffero, M. Martini, B. Passione, G. Mattiazzo, E. Giorcelli, and G. Bracco, "Stochastic control of inertial sea wave energy converter," *The Scientific World Journal*, vol. 2015, pp. 1–14, 2015.

[9]G. Bracco, A. Cagninei, E. Giorcelli, G. Mattiazzo, D. Poggi, and M. Raffero, "Experimental validation of the iswec wave to pto model," *Ocean Engineering*, vol. 120, pp. 40–51, 2016.