

**UNIVERSITY OF SÃO PAULO
SÃO CARLOS SCHOOL OF ENGINEERING**

Gabriel José Negrelli Gomes

**Development of Software for Parameter Estimation and
its Application on Wind Power Plant Equivalent Model**

São Carlos

2020

Gabriel José Negrelli Gomes

**Development of Software for Parameter Estimation and
its Application on Wind Power Plant Equivalent Model**

Dissertation presented to the São Carlos
School of Engineering at the University of
São Paulo in fulfillment of the requirements
for the degree of Master of Science in Electrical
Engineering.

Area: Electrical Power Systems

Supervisor: Prof. Dr. Elmer Pablo Tito Cari

FINAL VERSION

São Carlos

2020

I HEREBY AUTHORIZE THE TOTAL OR PARTIAL REPRODUCTION OF THIS WORK BY ANY CONVENTIONAL OR ELECTRONIC MEANS, FOR RESEARCH PURPOSES, PROVIDED THE SOURCE IS PROPERLY ACKNOWLEDGED.

G633d Gomes, Gabriel José Negrelli
Development of Software for Parameter Estimation and its Application on Wind Power Plant Equivalent Model / Gabriel José Negrelli Gomes ; advisor Elmer Pablo Tito Cari. – São Carlos, 2020.
82 p.

Dissertation (Masters) - Post-Graduation Program in Electrical Engineering, Electrical Power Systems Area – São Carlos School of Engineering, University of São Paulo, 2020.

1. Parameter estimation. 2. Wind power plants. 3. MVMO. 4. Trajectory sensibility. 5. Python. I. Cari, Elmer Pablo Tito, advisor. II. Title.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **GABRIEL JOSÉ NEGRELLI GOMES**.

Título da dissertação: "Desenvolvimento de software para estimação de parâmetros e sua aplicação em modelo equivalente de parques eólicos".

Data da defesa: 07/12/2020

Comissão Julgadora

Resultado

Prof. Dr. **Elmer Pablo Tito Cari**
(Orientador)
(Escola de Engenharia de São Carlos – EESC/USP)

APROVADO

Prof. Dr. **Edson Aparecido Rozas Theodoro**
(Universidade Tecnológica Federal do Paraná/UTFPR)

APROVADO

Prof. Dr. **Eduardo Coelho Marques da Costa**
(Escola Politécnica/EP-USP)

APROVADO

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Titular **Ivan Nunes da Silva**

Presidente da Comissão de Pós-Graduação:
Prof. Titular **Murilo Araujo Romero**

To J  ssica, my love.

ACKNOWLEDGEMENTS

I would like to thank everyone that helped me during this project.

First, to all my family members for the support. Special thanks to my wife, Jéssica, for her help and patience.

Second, to all my friends and colleagues at LACOSEP for the enriching discussions and moments of joy.

Finally, to the São Carlos School of Engineering at the University of São Paulo and its staff, in special to Prof. Dr. Elmer Cari, for the guidance during this project.

*“You miss 100% of the shots you
don’t take. -Wayne Gretzky”
Michael Scott*

ABSTRACT

GOMES, G. J. N. **Development of Software for Parameter Estimation and its Application on Wind Power Plant Equivalent Model.** 2020. 82p. Dissertation (Masters) - São Carlos School of Engineering, University of São Paulo, São Carlos, 2020.

This project focused on the development of a software for parameter estimation of non-linear dynamic models and its application on wind power plant equivalent model. To achieve this goal, a parameter estimation package was developed in Python 3 containing the estimation methods applied alongside the models used during the study. To represent wind power plants, a generic model well-known in the literature was chosen, based on its ability of representing said plants during transients. The method applied for estimation of the model parameters is composed of two optimization algorithms. At first, Mean-Variance Mapping Optimization, an heuristic approach, is used in order to reduce the search region around a feasible solution. Afterwards, a non-linear algorithm based on Trajectory Sensitivity is used to determine the local minimum, thus estimating the parameters of the model. The method validation was made using data from simulated systems. Also, a guided user interface was developed for this application, aiding new users and improving its usability. Both the package and interface projects are hosted on the author's [GitHub page](#).

Keywords: Parameter Estimation. Wind Power Plants. MVMO. Trajectory Sensitivity. Python.

RESUMO

GOMES, G. J. N. **Desenvolvimento de Software para Estimação de Parâmetros e sua aplicação em Modelo Equivalente de Parques Eólicos**. 2020. 82p. Dissertation (Masters) - São Carlos School of Engineering, University of São Paulo, São Carlos, 2020.

O presente trabalho tem como objetivo o desenvolvimento de um *software* para estimação de parâmetros de modelos dinâmicos não-lineares e sua aplicação em um modelo equivalente de parques eólicos. Para este objetivo, um módulo de estimação de parâmetros foi desenvolvido em Python 3 contendo os métodos de estimação estudados, além dos modelos utilizados. Foi escolhido um modelo genérico da literatura para representar plantas eólicas, baseado na sua capacidade em simular de forma adequada o comportamento destes parques durante transitórios. O método utilizado para a estimação dos parâmetros do modelo é constituído por dois algoritmos de otimização. Primeiramente, o algoritmo de Otimização por Mapeamento de Média-Variância, uma abordagem heurística, é empregado a fim de reduzir a região de busca dos parâmetros em torno de uma solução factível. Em seguida, lança-se mão de um algoritmo não-linear, baseado no Método de Sensibilidade de Trajetória, para determinar o mínimo local, estimando os valores dos parâmetros de forma mais precisa. A validação do método foi realizada utilizando dados de sistemas simulados. Com o intuito de facilitar a experiência do usuário com o programa, também foi desenvolvida uma interface gráfica amigável. Os códigos desenvolvidos durante este projeto estão disponibilizados na página do [GitHub](#) do autor.

Palavras-chave: Estimação de Parâmetros. Plantas Eólicas. MVMO. Sensibilidade de Trajetória. Python.

LIST OF FIGURES

Figure 1 – Share of electricity demand in the EU covered by wind energy during 2019	26
Figure 2 – Electricity generation in Brazil by source during 2019	27
Figure 3 – Wind and water regime in the Brazilian Northeast Region	27
Figure 4 – Example of Wind Power Plant	28
Figure 5 – Representation of Type-1 Wind Turbine Generator	31
Figure 6 – Representation of Type-2 Wind Turbine Generator	32
Figure 7 – Torque-speed curve	33
Figure 8 – Representation of Type-3 Wind Turbine Generator	33
Figure 9 – Representation of Type-4 Wind Turbine Generator	34
Figure 10 – Share of installed capacity for each wind turbine generator type	35
Figure 11 – WPP model	36
Figure 12 – Current priority operation	36
Figure 13 – Measurement of $\phi_v(t)$ with two PMUs installed on the grid	40
Figure 14 – WPP model proposed in this research	41
Figure 15 – Exemplification of MVMO process	45
Figure 16 – Example of MVMO mapping function	46
Figure 17 – Effect of different mean and shape factor values on mapping function	47
Figure 18 – Comparison between symmetrical and asymmetrical shape factors	47
Figure 19 – Flowchart of hybrid estimation method	50
Figure 20 – Spring-mass system	50
Figure 21 – Error evolution of TSM with $m_0 = 7$ kg and $k_0 = 7$ N/m	52
Figure 22 – Convergence region of Trajectory Sensitivity Method	52
Figure 23 – Error evolution of TSM with $m_0 = 8$ kg and $k_0 = 10$ N/m	53
Figure 24 – Search region chosen for MVMO	54
Figure 25 – Error evolution of MVMO	55
Figure 26 – Error evolution with of hybrid approach for spring-mass system	55
Figure 27 – <i>identPy</i> initial page	58
Figure 28 – Model selection page	59
Figure 29 – Method selection page	59
Figure 30 – MVMO configuration page	60
Figure 31 – TSM configuration page	60
Figure 32 – Results page	61
Figure 33 – Test system simulated	63
Figure 34 – WPP bus voltage	64
Figure 35 – Active power of real system (measured) and mathematical model (simulated) after parameter estimation	65

Figure 36 – Reactive power of real system (measured) and mathematical model (simulated) after parameter estimation	66
Figure 37 – Active power from original and proposed models	67
Figure 38 – Reactive power from original and proposed models	67
Figure 39 – Active power of proposed model after estimation	69
Figure 40 – Reactive power of proposed model after estimation	69
Figure 41 – Schematic of Z-IM Load Model.	81
Figure 42 – Result of parameter estimation for Z-IM Load Model	82

LIST OF TABLES

Table 1 – Parameters estimated by Trajectory Sensitivity Method	52
Table 2 – MVMO parameter boundaries	53
Table 3 – MVMO parameter boundaries	54
Table 4 – Comparison of approaches	56
Table 5 – MVMO search region	64
Table 6 – Estimated values of parameters	65
Table 7 – Influence of population size in MVMO	66
Table 8 – MVMO search region used on estimation of proposed model	68
Table 9 – Estimated parameters for proposed model	68

LIST OF ABBREVIATIONS AND ACRONYMS

ABEEólica	Brazilian Wind Energy Association
ANEEL	Brazilian Electricity Regulatory Agency
DFIG	Doubly Fed Induction Generator
EESG	Electrical Excited Synchronous Generator
EU	European Union
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
LSC	Line-Side Converter
MVMO	Mean-Variance Mapping Optimization
PMU	Phasor Measurement Unit
PMSG	Permanent Magnet Synchronous Generator
PROINFA	Program of Incentive to Alternative Electric Energy Sources
RoW	Rest of World
RSC	Rotor-Side Converter
SCIG	Squirrel Cage Induction Generator
TSM	Trajectory Sensitivity Method
US	United States of America
UK	United Kingdom
WECC	Western Electricity Coordinating Council
WPP	Wind Power Plant
WRIG	Wound Rotor Induction Generator
WTG	Wind Turbine Generator

LIST OF SYMBOLS

\mathbf{e}	Error vector
f_s	MVMO scaling factor
G	Error sensibility
h	MVMO transformation function
i_{max}	Maximum generator current
J	l^2 -norm of error vector
k_I	Gain of PI controller
k_{VC}	Gain of proportional block
\mathbf{p}	Parameter vector
p_i	i-th element of the parameter vector
\bar{p}_i	Mean value of i-th element of the parameter vector
P	Active power generated
Q	Reactive power generated
R	Thevenin equivalent resistance
s_i	MVMO shape factor of i-th parameter
T_I	Time constant of PI controller
T_V	Time constant of delay block
\mathbf{u}	Input vector
v^*	Bus voltage threshold
v_d	Direct-axis voltage component
v_i	Variance of i-th parameter
v_q	Quadrature-axis voltage component
v_T	Bus voltage magnitude
\mathbf{x}	State vector

X	Thevenin equivalent reactance
\mathbf{y}	Output vector
\mathbf{y}_r	Output measured on real system
ϕ_v	Bus voltage angle
Γ	Jacobian matrix of error sensibility

CONTENTS

1	INTRODUCTION	25
1.1	Wind Energy	25
1.2	Difficulties in Representing Wind Power Plants	28
1.3	Research Goals	29
1.4	Work Organization	29
2	MODELLING OF WIND TURBINE GENERATORS	31
2.1	Generic Models of Wind Turbine Generators	31
2.1.1	Type-1: Squirrel Cage Induction Generator (SCIG)	31
2.1.2	Type-2: Wound Rotor Induction Generator (WRIG)	32
2.1.3	Type-3: Doubly Fed Induction Generator (DFIG)	32
2.1.4	Type-4: Full-Converter Generator	34
2.2	Wind Power Plant Model Selected for Parameter Estimation	34
2.2.1	Proposed Wind Power Plant Model for Further Studies	40
3	PARAMETER ESTIMATION PROCESS	43
3.1	Mean-Variance Mapping Optimization Method	44
3.2	Trajectory Sensitivity Method	48
3.3	Hybrid Estimation Method	49
3.4	Parameter Estimation of Spring-Mass Model	50
3.4.1	Trajectory Sensitivity Estimation	51
3.4.2	MVMO Estimation	53
3.4.3	Hybrid Estimation	54
4	IDENTPY: A SOFTWARE FOR PARAMETER ESTIMATION	57
4.1	<i>identPy</i> Library	57
4.2	<i>identPy</i> GUI	58
5	ESTIMATION RESULTS AND DISCUSSIONS	63
5.1	Parameter Estimation of Original WPP Model	63
5.1.1	Influence of MVMO Population Size	65
5.2	Parameter Estimation of Proposed WPP Model	66
6	CONCLUSION	71
7	ONGOING RESEARCH	73
7.1	Regarding the WPP proposed model	73

7.2	Regarding the Hybrid Method	73
8	LIST OF PUBLICATIONS	75
	BIBLIOGRAPHY	77
	APPENDIX	79
	APPENDIX A – PARAMETER ESTIMATION OF LINEARIZED Z-IM LOAD MODEL	81

1 INTRODUCTION

During the last two decades, the world has seen an increasing participation of renewable sources in power generation, led mainly by wind and solar energy. These technologies provide an alternative to sources based on fossil fuel, such as oil, gas and coal, lowering pollution levels and reducing greenhouse gas emissions. On the other hand, the power output from these sources strongly relies on weather conditions and cannot be fully controlled.

This increase is seen worldwide, as part of policies to reduce the human impact on climate and the environment. This ‘renewable wave’ is led mainly by the European Union (EU), United States (US) and China. In particular, the EU has set in 2010 a strategy plan to reduce its greenhouse emissions by at least 20% compared to 1990 levels and increase the share of renewable sources to at least 20% by 2020 ([European Commission, 2010](#)).

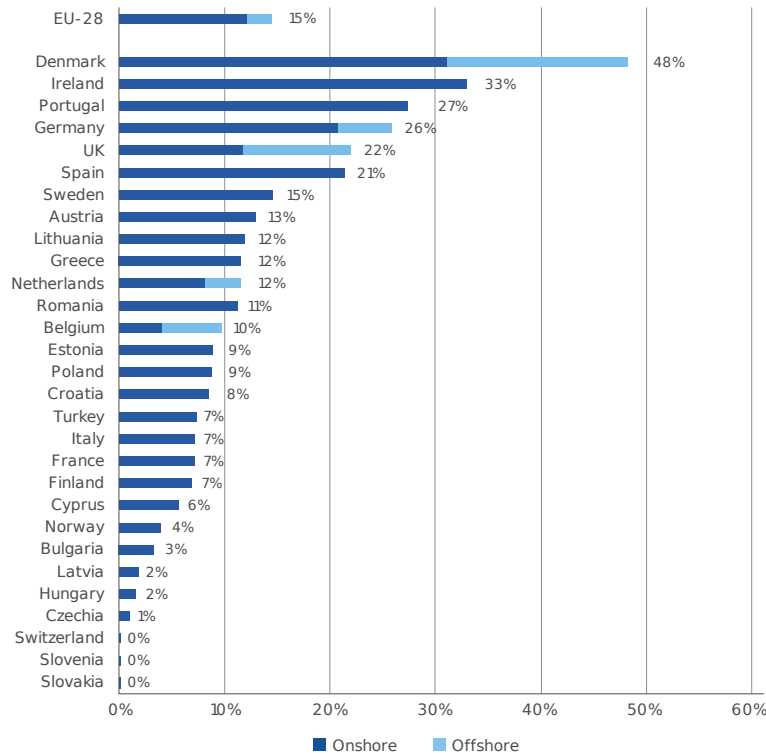
Brazil does not lag far behind the EU regarding renewable sources policies. In 2002, the country passed a bill that, among other actions, creates the Program of Incentive to Alternative Electric Energy Sources (PROINFA). This program aims to increase the share of wind, solar, small hydro and biomass energy production. The final goal is to have these energy sources corresponding to 10% of Brazil’s annual energy consumption by 2024 ([Federative Republic of Brazil, 2002](#)).

1.1 Wind Energy

Those policies have favored the increase of wind energy participation, reaching a scenario where it is one of the main energy sources of some countries, such as Denmark and Ireland. In the EU, wind energy alone generated 417 TWh in 2019, covering 15% of the electricity demand, a share 1% higher than 2018, with wind turbine generators (WTGs) installed both onshore (within the countries) and offshore (in the ocean). Among the EU countries, Denmark leads in this sector, with 48% of its demand supplied by wind power plants (WPPs), followed by Ireland (33%), Portugal (27%) and Germany (26%). The total installed capacity across the 28 EU countries (UK included) is 192 GW, with Germany in first position, with a total installed capacity of 61 GW, followed by Spain and the United Kingdom (UK), with 26 and 24 GW installed, respectively ([Wind Europe, 2020](#)). Figure 1 displays the detailed percentage of electricity demand covered by wind in the EU.

In Brazil, wind energy contributed to the electricity matrix with 48.5 TWh during 2018, resulting in a participation share of 8.1%. For comparison, Itaipu, the largest power

Figure 1 – Share of electricity demand in the EU covered by wind energy during 2019



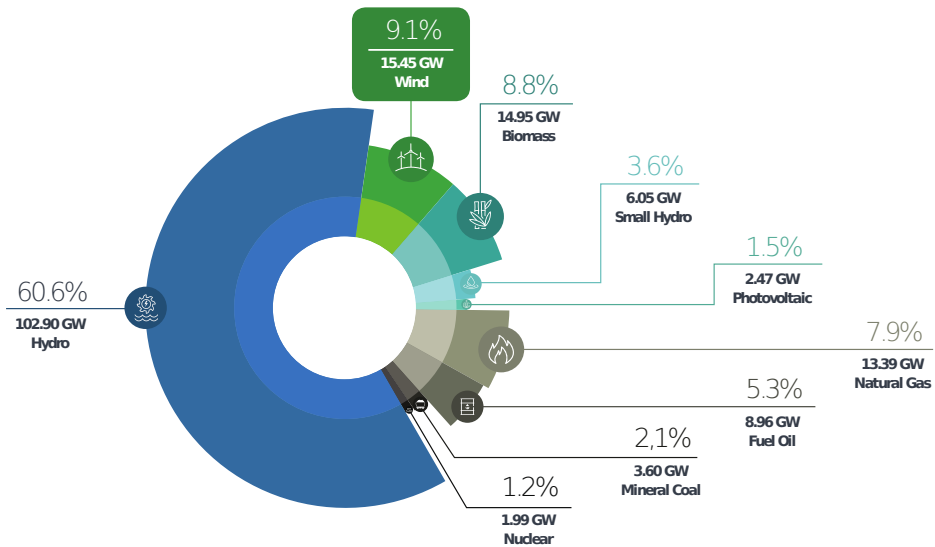
Source: Wind Europe, 2020

plant in Brazil, has produced 96.5 TWh during the same period. But, while other sources, such as gas and coal, had their share lowered, wind energy had the highest increase among sources comparing to 2017, increasing its contribution by 14.4% (EPE, 2019).

Regarding the installed capacity, wind power plants appear in 2nd place, with 15.5 GW installed, only behind hydro power plants (ABEEOLICA, 2020), as shown in Figure 2. However, there is still plenty of energy yield for this source to be explored. In (AMARANTE et al., 2001) is shown that Brazil has potential to generate 272.2 TWh per year, with an installed capacity of 143.5 GW. The Northeast Region has the higher potential, with an annual energy yield of 144.3 TWh and potential to host up to 75.0 GW. Also, the wind regime in the Northeast Region is complimentary to the water regime of the main river responsible to power generation in the region, as presented by Figure 3. This characteristic would help controlling reservoir water level during dry season, an important resource not only for power generation, but also irrigation of crops and water supply (ANEEL, 2005).

Therefore, it is expected that wind energy will increase its participation in electricity generation in the near future. However, some aspects of the wind energy must be considered prior to the implementation of wind power plants on large scale.

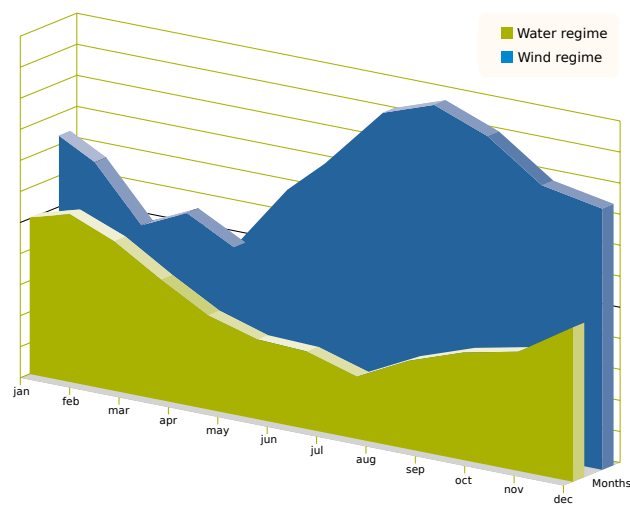
Figure 2 – Electricity generation in Brazil by source during 2019



Source: ABEEólica, 2020

The main difficulties are due to the nature of the energy source and characteristics of the generator. The wind regime is not constant and evenly distributed across the country, depending on the region geography and vegetation. This results in a energy source that is concentrated in certain areas and not entirely reliable. Moreover, wind turbine generators are usually partially or entirely decoupled from the grid via power electronic converters, resulting in machines with low inertia. Thus, systems with high penetration of WTGs may experience stability issues (XIONG et al., 2019).

Figure 3 – Wind and water regime in the Brazilian Northeast Region



Source: ANEEL, 2005

In order to guarantee the electrical power system reliability, studies simulating various conditions must be performed beforehand, so operators can be prepared for different fault scenarios. Therefore, robust mathematical models, capable of adequately simulate the behaviour of every component on the grid, are vital to these studies. Otherwise, operators may not be able to find a solution to the fault problem or, even worse, provide one that will aggravate the fault.

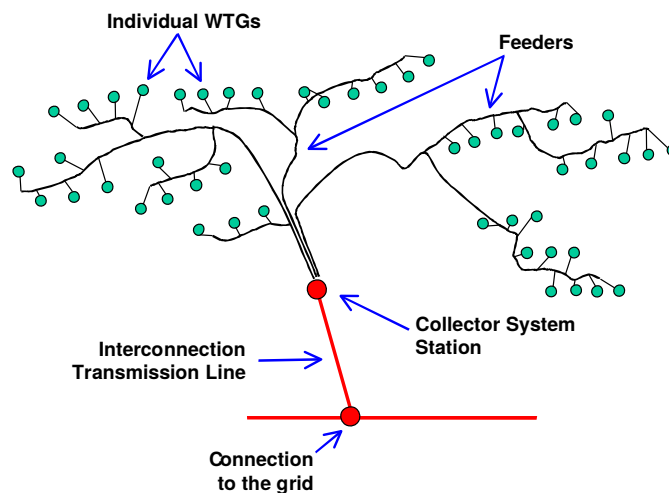
1.2 Difficulties in Representing Wind Power Plants

With a growing share of energy covered by wind, system operators must consider how wind turbines affect the system stability during faults and maneuvers. To reach this goal, mathematical models capable of describing the behaviour of these machines are crucial.

Obtaining these models, on the other hand, is not an easy task. Due to confidentiality, most manufacturers provide little or no information about the functioning of their wind turbine generators. In addition, there is a great amount of WTGs available, with different manufacturers, technologies, sizes and characteristics. Thus, a model that well describes a particular machine will not necessarily work for others.

Modelling entire wind power plants is even harder, since these facilities contain a large variety of WTGs spread over a wide area. Besides, line impedance of each generator is different, since their distances to the substation is not the same, as depicted in Figure 4. Hence, having one model for every wind turbine within a power plant would result in a mathematical model with high computational cost and extremely complex (ERLICH et al., 2012).

Figure 4 – Example of Wind Power Plant



Source: Adapted from (MULJADI; ELLIS, 2008)

In order to avoid using such complex models, many studies addressed the application of generic equivalent models to simulate the behaviour of WPPs. In particular, (ELLIS et al., 2011) shows that most wind power plants can be represented by a single-machine equivalent model. The benefits of using such equivalent model include reduction on model complexity and the fact it can be adjusted to match any given wind power plant. On the downside, these models cannot reproduce how particularities of each generator, such as wind speed and low voltage ride through capability, affect the WPP behaviour.

1.3 Research Goals

The main goal of this research was to develop a software for parameter estimation of nonlinear model and apply it to a wind power plant equivalent model used in transient stability studies. In order to achieve this goal, a hybrid estimation approach was applied combining two methods: Mean-Variance Mapping Optimization, a population-based meta-heuristic approach, and Trajectory Sensitivity Method, a nonlinear approach. Also, as a secondary goal, a graphical user interface (GUI) was developed, so other users can easily apply the tool created. This research is an extension of (CARI; ALBERTO; ERLICH, 2015), that focused on the parameter estimation using only Trajectory Sensitivity Method.

1.4 Work Organization

This section summarizes how the remainder of the text is organized. Chapter 2 will focus on the generic models for wind turbine generators and power plants and the selected mathematical model for parameter estimation purposes will be presented. The hybrid estimation process proposed and its methods will be subject of chapter 3. The python package and GUI created in this research will be detailed on chapter 4. On chapter 5, the results obtained on this research will be discussed. Finally, chapters 6 and 7 cover the conclusions and ongoing work of this study.

2 MODELLING OF WIND TURBINE GENERATORS

Due to the huge variety of wind turbine generators and their different characteristics, modelling each machine of a wind power plant separately would be a long and exhaustive work. To address this problem, studies such as (MULJADI; ELLIS, 2008), (ELLIS et al., 2011), (COUNCIL, 2008) and (ASMINE et al., 2011), conducted specially by the Institute of Electrical and Electronics Engineers (IEEE) and the Western Electricity Coordinating Council (WECC), developed generic models able to predict the behaviour of entire wind power plants. Such models reduced the problem complexity, since they are composed of a single equivalent generator. A two-machine model is needed only in rare cases, such as when the wind power plant is composed of two or more types of wind turbines (ELLIS et al., 2011).

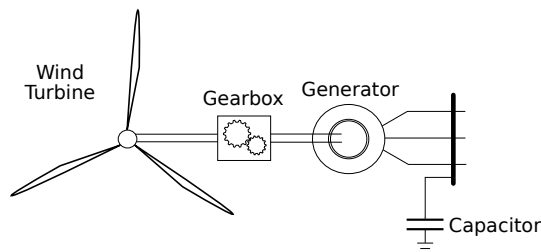
2.1 Generic Models of Wind Turbine Generators

The studies mentioned above have concluded that commercial wind turbine generators could be sorted into four basic types, according to their characteristics and technologies (ELLIS et al., 2011). These types are described in the following subsections.

2.1.1 Type-1: Squirrel Cage Induction Generator (SCIG)

The first type of wind turbine generator is composed of a Squirrel Cage Induction Generator (SCIG) connected to a wind turbine through a controlled gearbox, as displayed in Figure 5.

Figure 5 – Representation of Type-1 Wind Turbine Generator



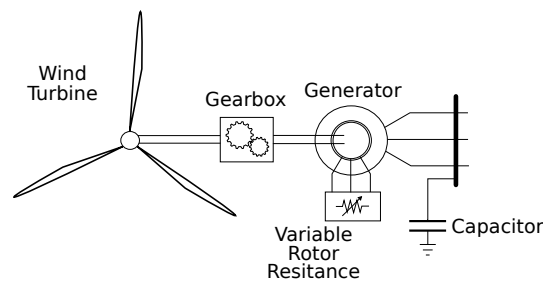
Due to its torque-speed characteristics, generators of this type operate at constant rotor speed, requiring robust controllers on gearbox and blade. Besides, as usual to any induction generator, the SCIG absorbs reactive power during operation. Thus, capacitors are often employed for power factor correction purposes. Moreover, type-1 generators limit aerodynamic power by varying the pitch angle of their blades, imposing great mechanical

stress on blades, shafts and gears, demanding a robust mechanical design and preventing these generators to operate above certain wind speed (ELLIS et al., 2011).

2.1.2 Type-2: Wound Rotor Induction Generator (WRIG)

Similarly to Type-1 WTG, Type-2 Wind Turbine Generators are composed of an asynchronous machine connected to a wind turbine via gearbox. However, instead of SCIG, Wound Rotor Induction Generator (WRIG) are used to convert kinetic energy into electricity. Generators of this type grant access to the rotor windings, allowing variances on the rotor resistance. As a direct consequence, this machine can operate in different wind speeds by adjusting its torque-speed curve as needed (ELLIS et al., 2011). Therefore, Type-2 WTG have a WRIG with a variable resistance connected to its rotor terminals, as shown in Figure 6.

Figure 6 – Representation of Type-2 Wind Turbine Generator



This type of generator has three speed control systems, with rotor resistance control responding to rapid changes in speed, gearbox control for medium variations and pitch control for slow changes. These control systems work together to maintain power output at the desired level and reduce mechanical stress on components. The effects on the torque-speed curve caused by different rotor resistances are shown in Figure 7.

For a fixed power, increasing rotor resistance increases the speed needed on the shaft, allowing the wind turbine to operate above rated wind speed. However, the speed range is only $\pm 10\%$ of rated slip. Also, this machine still needs a reactive compensation circuit on its terminals, since it naturally drains reactive power from the grid (MULJADI et al., 2010).

2.1.3 Type-3: Doubly Fed Induction Generator (DFIG)

A Type-3 Wind Turbine Generator, often called Doubly Fed Induction Generator (DFIG), is also composed of a wound rotor machine connected to a wind turbine. But, instead of having a variable resistance connected to its rotor terminals, as the previous WTG type, DFIGs have their rotor windings supplied with AC voltage by a back-to-back frequency converter, as displayed in Figure 8.

Figure 7 – Torque-speed curve

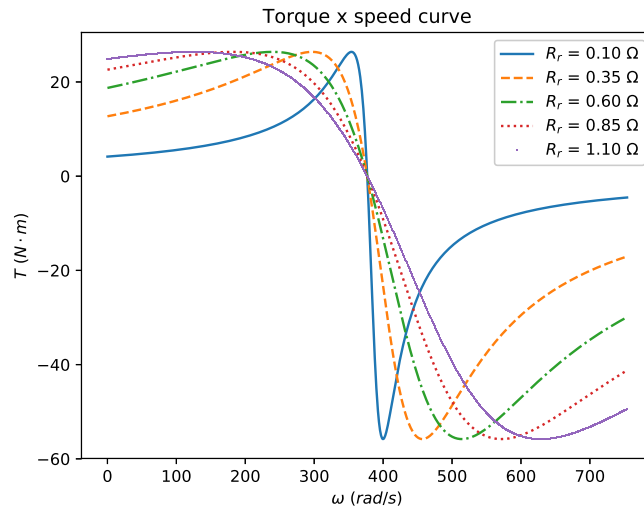
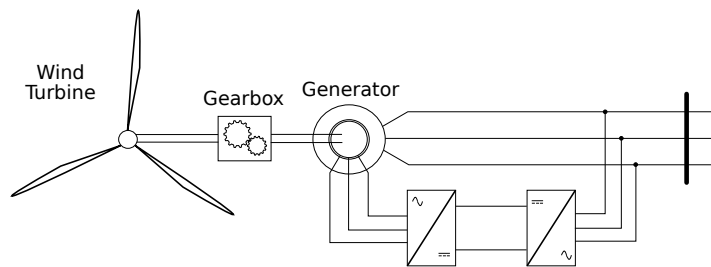


Figure 8 – Representation of Type-3 Wind Turbine Generator



The line-side converter (LSC) main purpose is to maintain voltage level on the DC link and provide reactive current during fault, while the rotor-side converter (RSC) controls active and reactive power generated by the machine by emulating different voltage frequencies on the rotor windings. By varying voltage frequency on the rotor circuit, this type of generator is able to supply power to the grid in a wider range of wind speed, reaching $\pm 30\%$ of rated slip. In addition, as result of the independent control of active and reactive power, reactive compensator circuits are no longer needed (MULJADI et al., 2010).

Since approximately 30% of rated power flows through the rotor windings, power electronics components have lower specifications and do not have great impact on overall costs. On the other hand, these generators need regular maintenance due to slip rings, brushes and gearbox, preventing its broad use in offshore applications (YARAMASU et al., 2015).

2.1.4 Type-4: Full-Converter Generator

The last type of wind turbine generator, also called Full-Converter Generator, is composed of an electrical machine connected to the grid through a back-to-back frequency converter. The converters operate by controlling the frequency of the generated voltage so it matches the grid standard frequency. In this configuration, WTGs are able to produce energy in a large range of wind speed (up to almost 100% of rated slip). Also, the connection between rotor and wind turbine shaft can be made directly or via gearbox. Likewise, it allows the use of both synchronous and asynchronous electrical machines as generator, with Permanent Magnet Synchronous Generator (PMSG), Electrical Excited Synchronous Generator (EESG) and SCIG being the most common, due to cost and maintenance purposes. Similar to DFIG, full-converter generators are able to independently control real and reactive power injected into or drained from the grid. However, since the entire generated power must flow through the power electronic devices, the overall cost of these generators is usually higher than other WTG types (YARAMASU et al., 2015). Figure 9 depicts a typical Type-4 Wind Turbine Generator.

Figure 9 – Representation of Type-4 Wind Turbine Generator

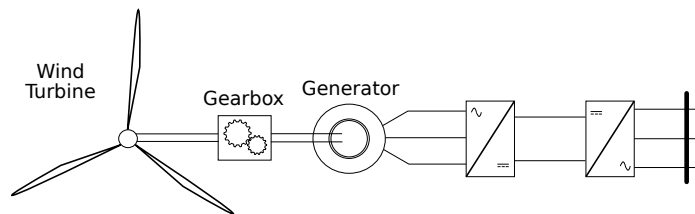


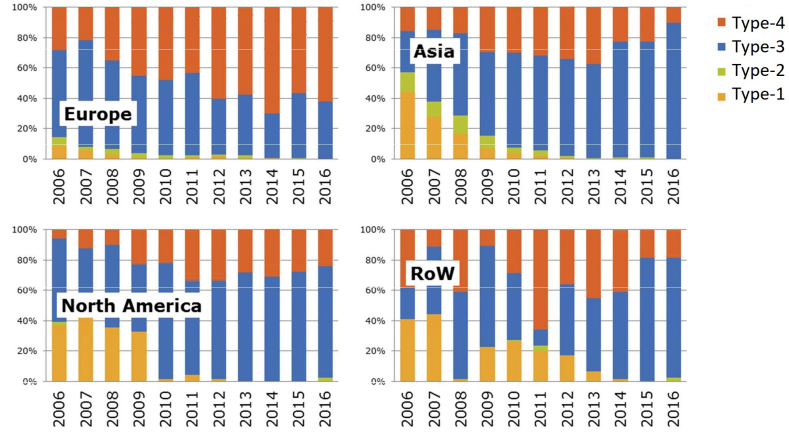
Figure 10 shows the evolution of share in installed capacity onshore for each generator type. The data shows how SCIG and WRIG lost space in the segment and how participation of DFIG and Full-Converter Generators rose, dominating the global market (MAGAGNA et al., 2017).

2.2 Wind Power Plant Model Selected for Parameter Estimation

Many mathematical models were developed during the last years that are able to represent wind turbine generators of all types. All those models have in common the fact that they are based on the generic models proposed by the studies made by WECC and IEEE and presented in the last section. The mathematical model selected for this work is presented in this section.

Initially proposed by (ERLICH et al., 2012), this mathematical model is able to represent the dynamic behaviour of both Type-3 and -4 WTGs and can be applied on simulation of entire wind power plants. Since this model is used for transient stability

Figure 10 – Share of installed capacity for each wind turbine generator type



Source: Adapted from (MAGAGNA et al., 2017)

studies, it is assumed that wind speed and, consequently, mechanical power are constant during that interval.

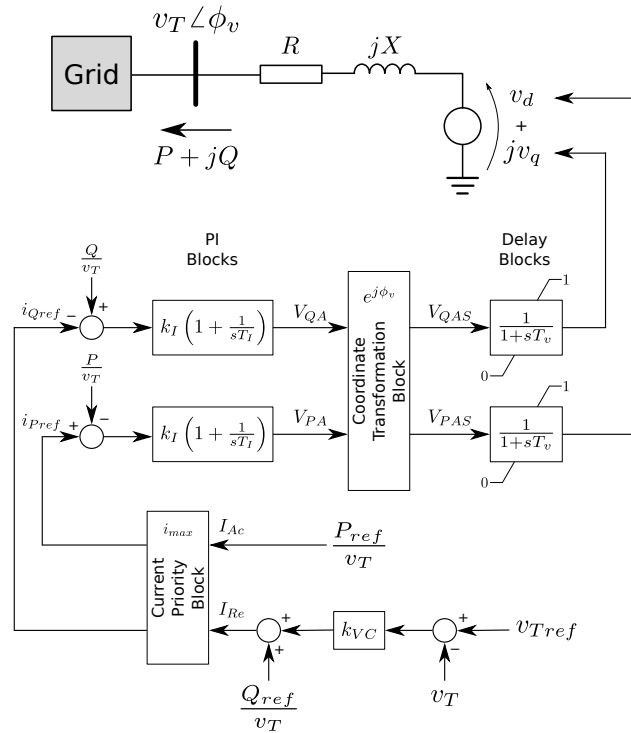
In this model, WPPs are represented by their Thevenin equivalent, composed of a voltage source connected to the grid by an impedance. The direct and quadrature components of the equivalent voltage source are controlled by a series of blocks that simulate the controllers of WTGs. The block diagram of this model is shown in Figure 11, where v_T and ϕ_v correspond to the voltage magnitude and angle at substation bus, P and Q are the active and reactive power generated by the WPP, v_d and v_q stand for the direct and quadrature-axis components of the generated voltage, R and X correspond to the Thevenin equivalent resistance and reactance, k_I and T_I are the PI block gain and time constant, T_V is the delay block time constant and k_{VC} is the voltage block gain.

The reference values of bus voltage (v_{Tref}), active (P_{ref}) and reactive power (Q_{ref}) are obtained during regime and vary according to wind speed. In this application, these reference values will be set as equal to the values of v_T , P and Q right before the disturbance. The real and imaginary current components that WPP must provide in order to maintain voltage and power at the reference level can be easily obtained by the equations below.

$$\begin{cases} I_{Ac} = \frac{P_{ref}}{v_T} \\ I_{Re} = k_{VC}(v_{Tref} - v_T) + \frac{Q_{ref}}{v_T} \end{cases} \quad (2.1)$$

However, under certain conditions, those current components must be prioritized. System conditions are analysed by a current priority block, which limits the current previously calculated by checking if it surpasses the limits allowed and if terminal voltage is below a threshold value v^* , characterizing voltage sag. Figure 12 exemplifies the block

Figure 11 – WPP model

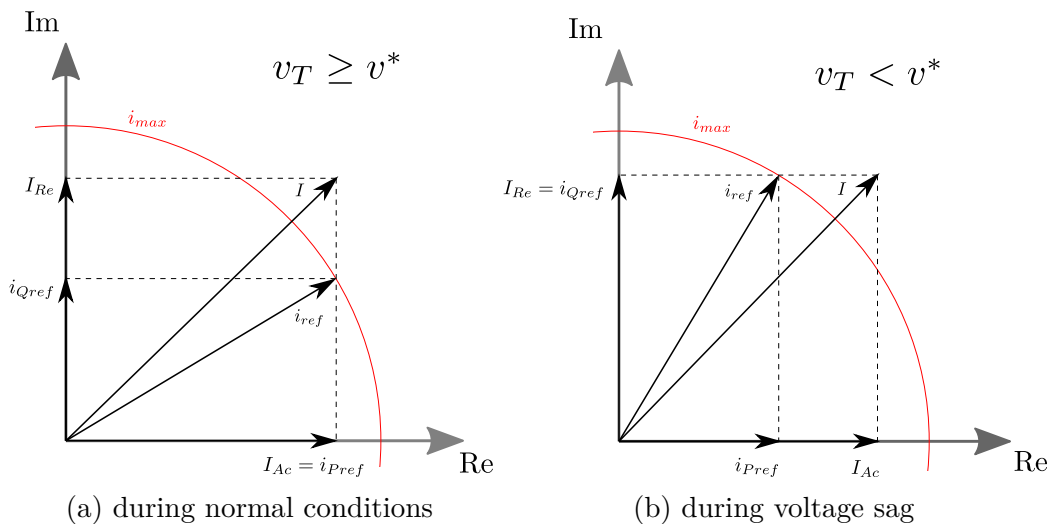


Source: Adapted from (ERLICH et al., 2012)

operation.

Under normal conditions, the WPP must focus on delivering active power to the grid, as shown in Figure 12a. On the other hand, in the event of voltage sag, the WPP must act on the voltage regulation, providing as much reactive power as possible, as displayed

Figure 12 – Current priority operation



in Figure 12b. Also, under no circumstances the current limitations of the components should be violated.

The current priority block outputs the reference values of active and reactive current that will be used to control the power generated by the WPP. The following algorithm describes the current priority block operation.

```

if  $\sqrt{I_{Ac}^2 + I_{Re}^2} \leq i_{max}$  then
   $i_{Pref} = I_{Ac}$ 
   $i_{Qref} = I_{Re}$ 
else
  if  $v_T \geq v^*$  then
     $i_{Pref} = \min(i_{max}, I_{Ac})$ 
     $i_{Qref} = \sqrt{i_{max}^2 - i_{Pref}^2}$ 
  else
     $i_{Qref} = \min(i_{max}, I_{Re})$ 
     $i_{Pref} = \sqrt{i_{max}^2 - i_{Qref}^2}$ 
  end if
end if

```

At the next stage, proportional-integral blocks are used to represent the controllers of WTGs, specially the RSC controller. Their outputs follow the equations presented below. The values V_{PA} and V_{QA} are, respectively, the active and reactive voltage components (on grid coordinates) that the Thevenin voltage source must supply so the reference values of bus voltage and power are satisfied.

$$\begin{cases} V_{PA} = k_I[(i_{Pref} - \frac{P}{v_T}) + \frac{1}{T_I} \int_0^t (i_{Pref} - \frac{P}{v_T}) dt] \\ V_{QA} = k_I[(\frac{Q}{v_T} - i_{Qref}) + \frac{1}{T_I} \int_0^t (\frac{Q}{v_T} - i_{Qref}) dt] \end{cases} \quad (2.2)$$

By analysing (2.2), one can notice that active and reactive components are completely decoupled. Normally, small (but not insignificant) cross-coupling signals should contribute on both equations. However, as presented in (ERLICH; SHEWAREGA, 2006), the effects of the cross-coupling can be neglected without major issues by binding the PI controllers using the same gain and time constant.

At this point, the controllers operate on grid coordinates, so a coordinate transformation block is needed to adequate to $d - q$ coordinates. This block operates according to equation (2.3). It is important to notice that these equations rely on the knowledge of ϕ_v and, in order to obtain that measurement, the WPP must have a phasor measurement

unit (PMU) installed on its substation.

$$\begin{cases} V_{PAS} = -V_{PACOS}(\phi_v) - V_{QASIN}(\phi_v) \\ V_{QAS} = V_{PASIN}(\phi_v) - V_{QACOS}(\phi_v) \end{cases} \quad (2.3)$$

Finally, limited lag blocks, simulating the delays caused by the electrical machine and converters present on WTGs, supply the voltage source. The lag block constraints limit the values of those voltage components to remain in a safety margin, between 0 and 1 pu. Equation (2.4) describes how these blocks affect v_d and v_q .

$$\begin{cases} \dot{v}_d = \frac{1}{T_V}(v_d - V_{PAS}) \\ \dot{v}_q = \frac{1}{T_V}(v_q - V_{QAS}) \end{cases} \quad (2.4)$$

With direct and quadrature-axis voltage components obtained, the power generated by the WPP can be easily calculated, as shown below.

$$\begin{aligned} S^* &= v_T^* \cdot I = v_T^* \cdot \frac{[(v_d + jv_q) - v_T]}{R + jX} \\ &= (v_{Td} - jv_{Tq}) \cdot \left[\frac{v_d + jv_q - (v_{Td} + jv_{Tq})}{R + jX} \right] \cdot \frac{R - jX}{R - jX} \\ &= \frac{v_{Td}v_d + v_{Tq}v_q - v_T^2 + j(v_{Td}v_q - v_{Tq}v_d)}{R^2 + X^2} \cdot (R - jX) \\ &= \frac{R(v_{Td}v_d + v_{Tq}v_q - v_T^2) + X(v_{Td}v_q - v_{Tq}v_d)}{R^2 + X^2} - j \frac{X(v_{Td}v_d + v_{Tq}v_q - v_T^2) - R(v_{Td}v_q - v_{Tq}v_d)}{R^2 + X^2} \end{aligned}$$

Thus,

$$\begin{cases} P = \frac{R(v_{Td}v_d + v_{Tq}v_q - v_T^2) + X(v_{Td}v_q - v_{Tq}v_d)}{R^2 + X^2} \\ Q = \frac{X(v_{Td}v_d + v_{Tq}v_q - v_T^2) - R(v_{Td}v_q - v_{Tq}v_d)}{R^2 + X^2} \end{cases} \quad (2.5)$$

The initial conditions of the states $\mathbf{x}_0 = [v_{d0}, v_{q0}]^T$ can be obtained via equation (2.6), where v_{T0} and v_{T0}^* are the terminal bus voltage and its conjugate at the initial time. Likewise, P_0 and Q_0 stand for the active and reactive power measured at the same instant.

$$v_{d0} + jv_{q0} = v_{T0} + (R + jX) \cdot \frac{P_0 - jQ_0}{v_{T0}^*} \quad (2.6)$$

The summary of the original model is shown below.

$$\left\{ \begin{array}{l} \dot{v}_d = \frac{1}{T_V}(v_d - V_{PAS}) \\ \dot{v}_q = \frac{1}{T_V}(v_q - V_{QAS}) \\ P = \frac{R(v_{Td}v_d + v_{Tq}v_q - v_T^2) + X(v_{Td}v_q - v_{Tq}v_d)}{R^2 + X^2} \\ Q = \frac{X(v_{Td}v_d + v_{Tq}v_q - v_T^2) - R(v_{Td}v_q - v_{Tq}v_d)}{R^2 + X^2} \\ V_{PAS} = -V_{PA}\cos(\phi_v) - V_{QA}\sin(\phi_v) \\ V_{QAS} = V_{PA}\sin(\phi_v) - V_{QA}\cos(\phi_v) \\ V_{PA} = k_I[(i_{Pref} - \frac{P}{v_T}) + \frac{1}{T_I} \int_0^t (i_{Pref} - \frac{P}{v_T}) dt] \\ V_{QA} = k_I[(\frac{Q}{v_T} - i_{Qref}) + \frac{1}{T_I} \int_0^t (\frac{Q}{v_T} - i_{Qref}) dt] \end{array} \right. \quad (2.7)$$

if $\sqrt{I_{Ac}^2 + I_{Re}^2} \leq i_{max}$ **then**

$$i_{Pref} = I_{Ac}$$

$$i_{Qref} = I_{Re}$$

else

if $v_T \geq v^*$ **then**

$$i_{Pref} = \min(i_{max}, I_{Ac})$$

$$i_{Qref} = \sqrt{i_{max}^2 - i_{Pref}^2}$$

else

$$i_{Qref} = \min(i_{max}, I_{Re})$$

$$i_{Pref} = \sqrt{i_{max}^2 - i_{Qref}^2}$$

end if

end if

$$\left\{ \begin{array}{l} I_{Ac} = \frac{P_{ref}}{v_T} \\ I_{Re} = k_{VC}(v_{Tref} - v_T) + \frac{Q_{ref}}{v_T} \end{array} \right.$$

Thus, this model can be interpreted as presented in (2.8), where the input (\mathbf{u}), state (\mathbf{x}), parameter (\mathbf{p}) and output (\mathbf{y}) vectors are described by (2.9), (2.10), (2.11) and (2.12), respectively.

$$\left\{ \begin{array}{l} \dot{\mathbf{x}}(t) = f(t, \mathbf{u}, \mathbf{x}, \mathbf{p}) \\ \mathbf{y}(t) = g(t, \mathbf{u}, \mathbf{x}, \mathbf{p}) \end{array} \right. \quad (2.8)$$

$$\mathbf{u} = [v_T, \phi_v, P, Q]^T \quad (2.9)$$

$$\mathbf{x} = [v_d, v_q]^T \quad (2.10)$$

$$\mathbf{p} = [R, X, k_I, T_I, T_V, k_{VC}, i_{max}]^T \quad (2.11)$$

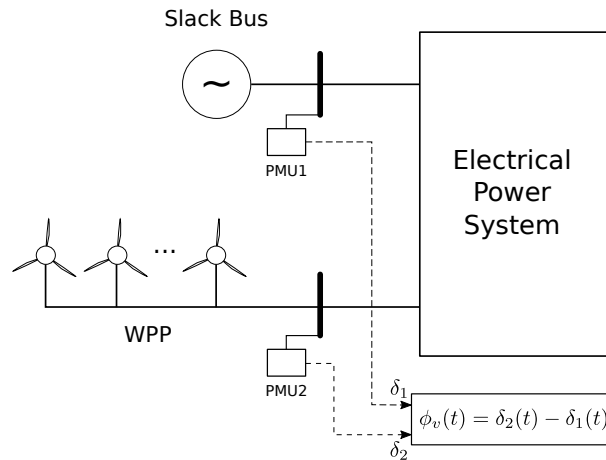
$$\mathbf{y} = [P, Q]^T \quad (2.12)$$

The parameters presented in (2.11) are the subject of the estimation process presented on the next chapter. Those values define the behaviour of specific wind power plants during disturbances.

2.2.1 Proposed Wind Power Plant Model for Further Studies

One of the disadvantages of the model presented above is the need of information about voltage angle ϕ_v . In order to acquire the voltage angle measurements, the grid must have special equipment, such as PMUs, installed in two of its buses, as shown in Figure 13. One of the PMUs must be located at the WPP terminal bus while the other must be installed at the slack bus¹ of the electrical power system. By processing the angle information provided by both equipment, the angle ϕ_v used on the estimation process is obtained.

Figure 13 – Measurement of $\phi_v(t)$ with two PMUs installed on the grid

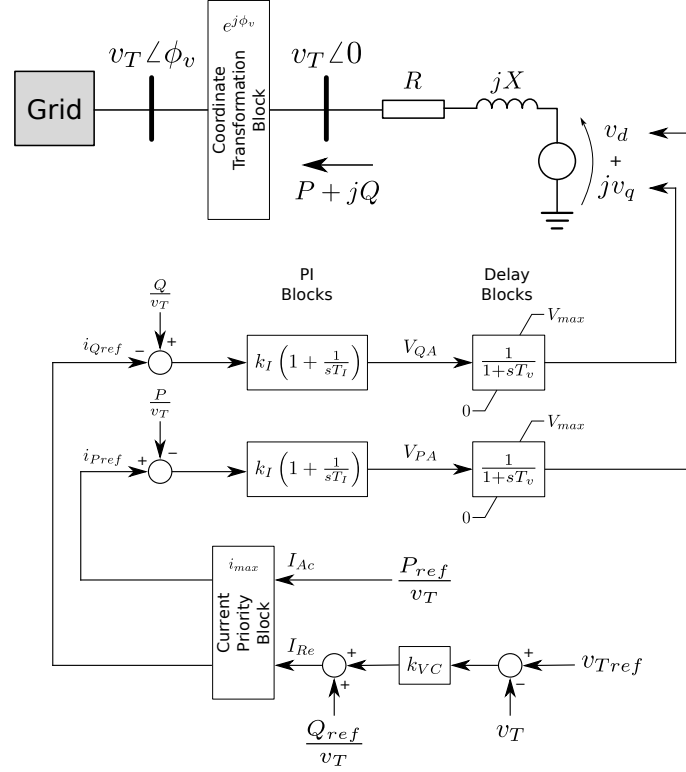


However, the requirement of PMUs may limit the application of the original model (2.7) when these equipment are not available. One alternative to avoid that problem,

¹ The slack bus is the bus used as angle reference in load flow studies

proposed in this research, is to exclude the angle ϕ_v from the model equations. This can be obtained by dislocating the coordinate transformation block to the WPP terminal bus, as depicted in Figure 14.

Figure 14 – WPP model proposed in this research



The model equations are slightly altered by this modification, as shown below by the summary of the proposed model. Moreover, as observed during simulations, the limits V_{max} applied to the delay blocks must also change. Thus, these new limits will become parameters of the estimation process as well.

$$\left\{ \begin{array}{l} \dot{v}_d = \frac{1}{T_V}(v_d - V_{PA}) \\ \dot{v}_q = \frac{1}{T_V}(v_q - V_{QA}) \\ P = \frac{R(v_T v_d - v_T^2) + X v_T v_q}{R^2 + X^2} \\ Q = \frac{X(v_T v_d - v_T^2) - R v_T v_q}{R^2 + X^2} \\ V_{PA} = k_I \left[\left(i_{Pref} - \frac{P}{v_T} \right) + \frac{1}{T_I} \int_0^t \left(i_{Pref} - \frac{P}{v_T} \right) dt \right] \\ V_{QA} = k_I \left[\left(\frac{Q}{v_T} - i_{Qref} \right) + \frac{1}{T_I} \int_0^t \left(\frac{Q}{v_T} - i_{Qref} \right) dt \right] \end{array} \right. \quad (2.13)$$

if $\sqrt{I_{Ac}^2 + I_{Re}^2} \leq i_{max}$ then

$$i_{Pref} = I_{Ac}$$

$$i_{Qref} = I_{Re}$$

else

if $v_T \geq v^*$ **then**

$$i_{Pref} = \min(i_{max}, I_{Ac})$$

$$i_{Qref} = \sqrt{i_{max}^2 - i_{Pref}^2}$$

else

$$i_{Qref} = \min(i_{max}, I_{Re})$$

$$i_{Pref} = \sqrt{i_{max}^2 - i_{Qref}^2}$$

end if

end if

$$\begin{cases} I_{Ac} = \frac{P_{ref}}{v_T} \\ I_{Re} = k_{VC}(v_{Tref} - v_T) + \frac{Q_{ref}}{v_T} \end{cases}$$

The new input, state, parameter and output vectors are given by:

$$\mathbf{u} = [v_T, P, Q]^T \quad (2.14)$$

$$\mathbf{x} = [v_d, v_q]^T \quad (2.15)$$

$$\mathbf{p} = [R, X, k_I, T_I, T_V, k_{VC}, i_{max}, V_{max}]^T \quad (2.16)$$

$$\mathbf{y} = [P, Q]^T \quad (2.17)$$

3 PARAMETER ESTIMATION PROCESS

Given a dynamic system described by (2.8), it is possible to adjust the output \mathbf{y} by altering the parameters \mathbf{p} . Parameter estimation problems rely on finding a set of parameters such as \mathbf{y} is close enough to the behaviour observed in a given system.

Thus, parameter estimation problems can be interpreted as optimization problems, where one must find the optimal values of parameters in order to reduce error between real system and model when the same disturbance is applied to it. The error vector \mathbf{e} at any instant t is calculated as the difference between the output measured on the real system \mathbf{y}_r and the one obtained from the model equations \mathbf{y} , as presented below.

$$\mathbf{e}(t) = \mathbf{y}_r(t) - \mathbf{y}(t) \quad (3.1)$$

The l^2 -norm of error vector, denoted by J , is applied to evaluate how well the model describes the real system behaviour. The norm is obtained through equation (3.2). It is important to notice that, since the model output \mathbf{y} varies with the parameter vector \mathbf{p} , so does J . The constant $\frac{1}{2}$ is only used for further simplification.

$$J(\mathbf{p}) = \frac{1}{2} \int_0^{T_0} \|\mathbf{e}(t)\|_2^2 dt = \frac{1}{2} \int_0^{T_0} \mathbf{e}(t)^T \cdot \mathbf{e}(t) dt \quad (3.2)$$

Many methods were developed for solving optimization problems, but two approaches have been largely employed during the last years. The first approach applies metaheuristics to obtain a sufficiently good solution. These methods are used in a variety of fields, ranging from biology to engineering, due to the fact that they are not developed for a specific type of problem.

Metaheuristics employ a stochastic search to find near-optimal solutions inside a given region. However, those methods often take a great amount of time to converge to a solution (BLUM; ROLI, 2003). Examples of metaheuristics are Ant Colony Optimization, Differential Evolution, Particle Swarm Optimization and Genetic Algorithm. Applications of this approach in electrical power system cases can be found in (TODOROVSKI; RAJIČIĆ, 2006) and (YOSHIDA et al., 2000).

The second approach applies analytical methods to find the local optimum solution from equations derived from the problem. Thus, they are problem specific and must be adapted from one case to another. Analytical methods often converge in few iterations, reducing processing time, but they are sensitive to initial conditions. Some examples of analytical methods are Newton's Method, Kalman Filter, Unscented Kalman Filter, etc.

By combining both approaches, the resulting method is expected to reduce the effects of their disadvantages and improve overall convergence. Mean-Variance Mapping Optimization (MVMO) was the metaheuristic chosen for this problem, alongside Trajectory Sensitivity Method (TSM) as analytical method. Both methods will be discussed in the following sections.

3.1 Mean-Variance Mapping Optimization Method

Presented in (ERLICH; VENAYAGAMOORTHY; WORAWAT, 2010), this metaheuristic based in evolution of populations shares characteristics with other evolutionary algorithms, but differ from them on how to induce mutations on the offspring in order to diversify the population. By considering statistical data of population during mutation process, MVMO introduces a memory factor to it, enhancing its search mechanism. Due to this factor, MVMO performs better than similar metaheuristics when population size is relatively small (NAKAWIRO; ERLICH; RUEDA, 2011). For the sake of analogy, in this section the terms ‘gene’, ‘individuals’ and ‘population’ will be used to refer to ‘parameter’, ‘parameter vector’ and ‘set of parameter vectors’, respectively.

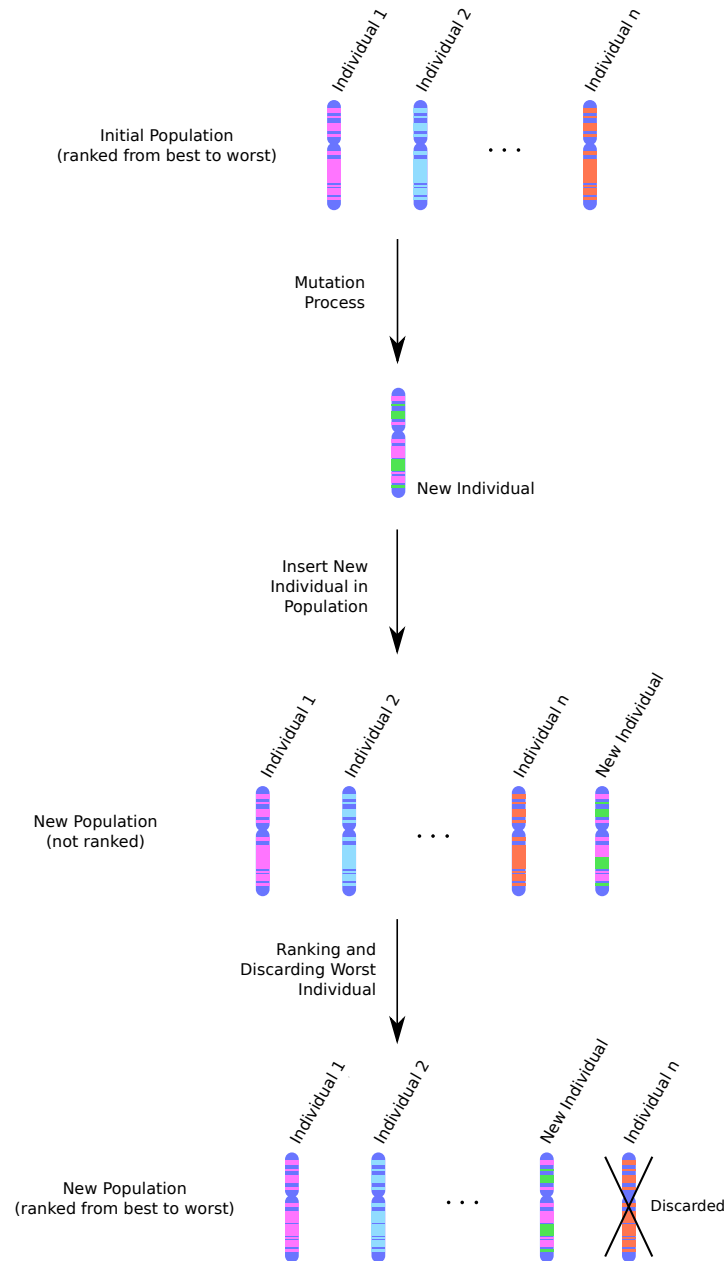
Two other relevant concepts largely used in metaheuristics are exploration and exploitation. The first one refers to a broad search carried through the region of interest. On the other hand, exploitation means the search on a small neighbourhood close to the best solutions.

Before starting the parameter search process, a few settings must be defined, such as population size, number of offspring, number of genes selected for mutation and selection method. Also, the search region is defined by setting the range within genes can vary. This constrains the parameters values within a feasible region, preventing divergence. The search region is later normalized for all genes by rescaling the values into the interval $[0, 1]$, where the lower and upper boundaries correspond to 0 and 1, respectively. Termination criteria is also set in this step. In this work, both number of generations and error will be used as stop criteria.

After that, a randomly-distributed population is generated, evaluated and sorted. Moreover, the mean and variance of every gene in the population are calculated. These values will later be used on the mutation process. The individual with lower error is selected as parent for a new generated individual. The offspring is then created following three steps common in evolutionary algorithms: gene selection, mutation and crossover. After creation, the offspring is introduced to the population and the worst individual is discarded, as depicted in Figure 15.

Gene selection can be done in many ways and even vary throughout the estimation process, with its efficiency depending on the problem. However, three strategies are

Figure 15 – Exemplification of MVMO process



commonly used in this step. The first one is comprised of randomly selecting which genes will suffer mutation and which will be directly inherited from the parent. Genes can also be selected by a moving window approach, where a fixed-length window informs which consecutive genes will suffer mutation. At each iteration, this window moves a predefined number of steps, resampling the genes. Furthermore, a combination of both strategies, with part of the genes selected at random and other through a moving window can be applied for selecting the genes.

Mutation step takes place after gene selection. At first, each selected gene receives a random value \tilde{p} in the interval $[0,1]$ that will be used as input to a mapping function

based on the mean and variance values of each particular gene on the population. Variance v_i will directly influence the shape factor s_i , given by (3.3), where f_s is the scaling factor responsible for focusing on exploration or exploitation. In the event of zero variance, the last non-null value of v_i is used.

$$s_i = -f_s \ln(v_i) \quad (3.3)$$

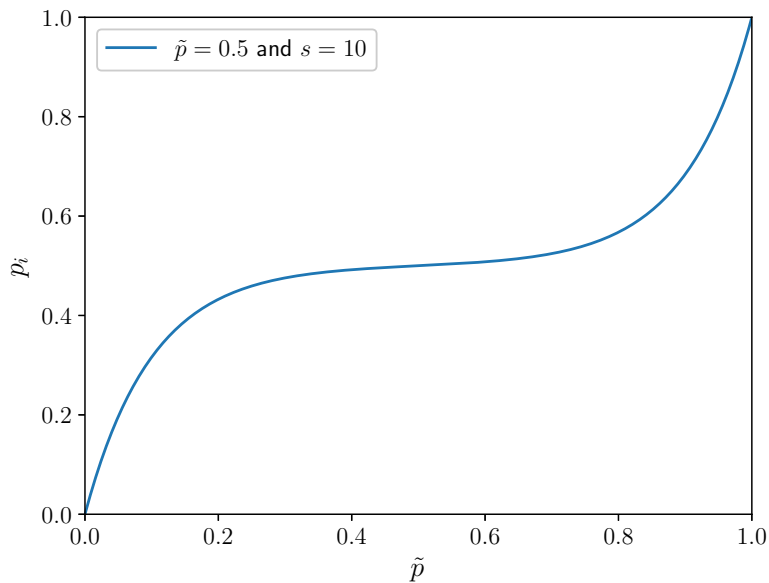
Shape factor s_i and mean value \bar{p}_i of genes are used as inputs to the transformation function h , detailed in (3.4). The final value of the gene is obtained through the mapping function described by equation (3.5), where $h_p = h(z_i = \tilde{p})$, $h_1 = h(z_i = 1)$ and $h_0 = h(z_i = 0)$. It is important to notice that the mapping function will always provide a result in the interval $[0,1]$, not violating the normalization made at the beginning.

$$h(\bar{p}_i, s_{i1}, s_{i2}, z_i) = \bar{p}_i(1 - e^{-z_i s_{i1}}) + (1 + \bar{p}_i)e^{-(1-z_i)s_{i2}} \quad (3.4)$$

$$p_i = h_p + (1 - h_1 + h_0)\tilde{p} - h_0 \quad (3.5)$$

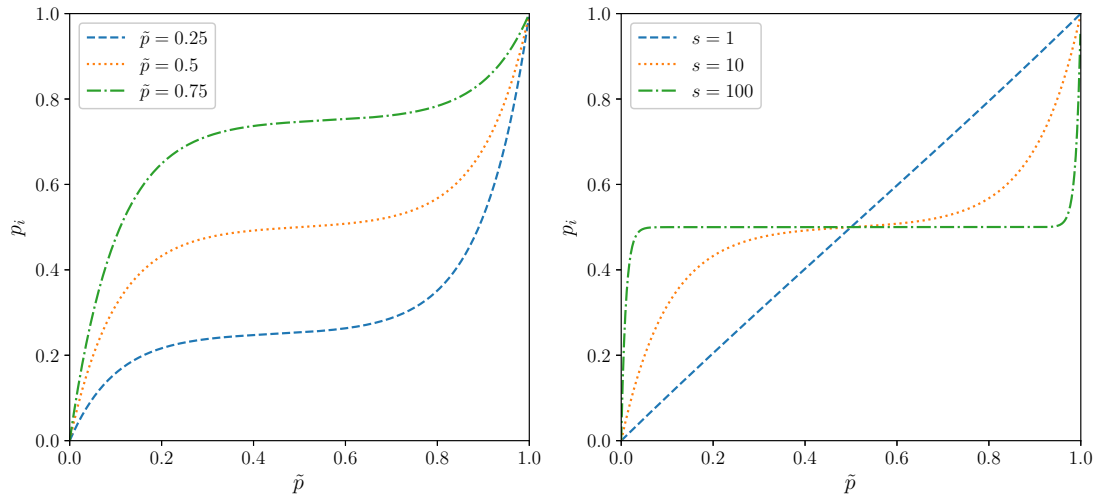
The resulting mapping function is depicted in Figure 16. The effects of different mean and shape factor values can be observed on Figure 17.

Figure 16 – Example of MVMO mapping function



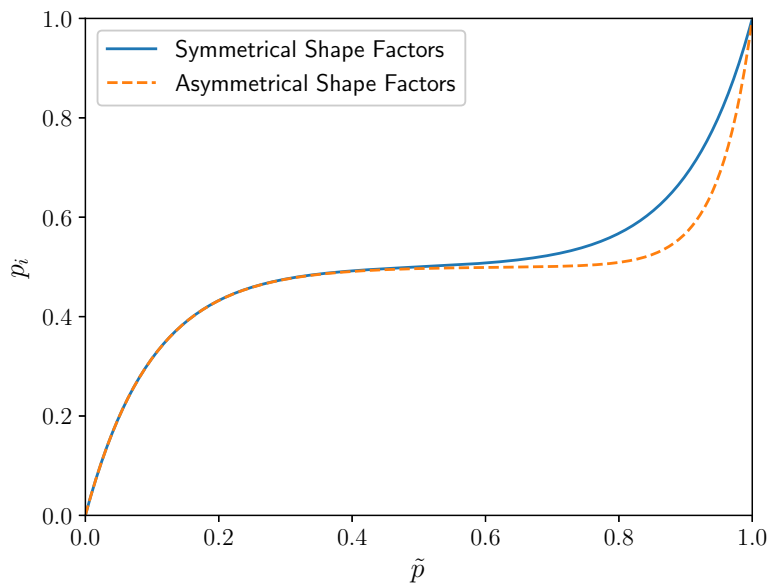
As shown in (3.4), two shape factors (s_{i1} and s_{i2}) are used to evaluate the function. Different values of shape factors can be applied to focus the search on the regions below

Figure 17 – Effect of different mean and shape factor values on mapping function



or above the mean value. Thus, by controlling the values s_{i1} and s_{i2} , the method can prioritize exploration (global search) or exploitation (refining local solution) on a given region. Figure 18 depicts how asymmetrical shape factors affect the mapping function.

Figure 18 – Comparison between symmetrical and asymmetrical shape factors



The final step during offspring generation is crossover. During this phase, the mutated genes are united with the remaining genes inherited from parent, forming the new individual. This new individual is evaluated and included to the population if it is

better than, at least, the population's worst individual. This process goes on until at least one stop criterion has been fulfilled.

The main advantages of this method are low computational cost, good performance using small populations, constrained search region, preventing divergence, and the fact that it is non-specific. On the other hand, this method, as other metaheuristics, takes a great amount of time to converge when its error approaches zero.

3.2 Trajectory Sensitivity Method

Considering a nonlinear system described by (2.8), in order to minimize the error between model and real system, given by (3.2), one must discover a parameter vector \mathbf{p}^* such as:

$$G(\mathbf{p}^*) = \frac{\partial J(\mathbf{p}^*)}{\partial \mathbf{p}} = 0 \quad (3.6)$$

This derivative can be written as:

$$G(\mathbf{p}) = - \int_0^{T_0} \left(\frac{d\mathbf{y}}{d\mathbf{p}} \right)^T (\mathbf{y}_r - \mathbf{y}) dt \quad (3.7)$$

Truncating the Taylor series for $G(\mathbf{p})$ on the first-order term results in (3.8). The matrix Γ is described in (3.9).

$$G(\mathbf{p}^*) = G(\mathbf{p}) + \Gamma(\mathbf{p})(\mathbf{p}^* - \mathbf{p}) \quad (3.8)$$

$$\Gamma(\mathbf{p}) = \frac{\partial G(\mathbf{p})}{\partial \mathbf{p}} \approx \int_0^{T_0} \left(\frac{d\mathbf{y}}{d\mathbf{p}} \right)^T \left(\frac{d\mathbf{y}}{d\mathbf{p}} \right) dt \quad (3.9)$$

By rearranging the terms on (3.8), the following equation is obtained. It describes how the parameters are updated after the n^{th} iteration.

$$\mathbf{p}^{n+1} = \mathbf{p}^n + \Gamma^{-1}(\mathbf{p}^n) \cdot G(\mathbf{p}^n) \quad (3.10)$$

Obtaining the Jacobian matrix (also called trajectory sensitivity matrix) $\frac{d\mathbf{y}}{d\mathbf{p}}$, used in (3.7) and (3.9), analytically is a hard task. However, by applying the definition of derivative, given by (3.11), the sensibilities can be approximated without any analytical derivation of the equations.

$$\frac{df(x)}{dx} = \lim_{\delta \rightarrow 0} \frac{f(x + \delta) - f(x)}{\delta} \quad (3.11)$$

Consider two parameter vectors \mathbf{p} and \mathbf{p}^ϵ , where \mathbf{p}^ϵ is obtained by adding a small perturbation ϵp_i to the i -th element of \mathbf{p} , as shown in (3.12).

$$\mathbf{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_i \\ \vdots \\ p_n \end{bmatrix}; \quad \mathbf{p}^\epsilon = \begin{bmatrix} p_1 \\ \vdots \\ p_i + \epsilon p_i \\ \vdots \\ p_n \end{bmatrix} \quad (3.12)$$

With ϵ sufficiently small, the partial derivative with respect to the parameter p_i can be approximated by the difference shown in equation (3.13). The value of $\epsilon = 0.1 \times 10^{-3}$ have shown great results for most cases. Using the approximation of the partial derivatives allows Trajectory Sensitivity method to be applied on both differentiable and non-differentiable systems (BENCHLUCH; CHOW, 1993), (CARI; ALBERTO; BRETAS, 2006).

$$\frac{\partial \mathbf{y}}{\partial p_i} = \frac{\partial g(t, \mathbf{u}, \mathbf{x}, \mathbf{p})}{\partial p_i} \approx \frac{g(t, \mathbf{u}, \mathbf{x}, \mathbf{p}^\epsilon) - g(t, \mathbf{u}, \mathbf{x}, \mathbf{p})}{\epsilon p_i} \quad (3.13)$$

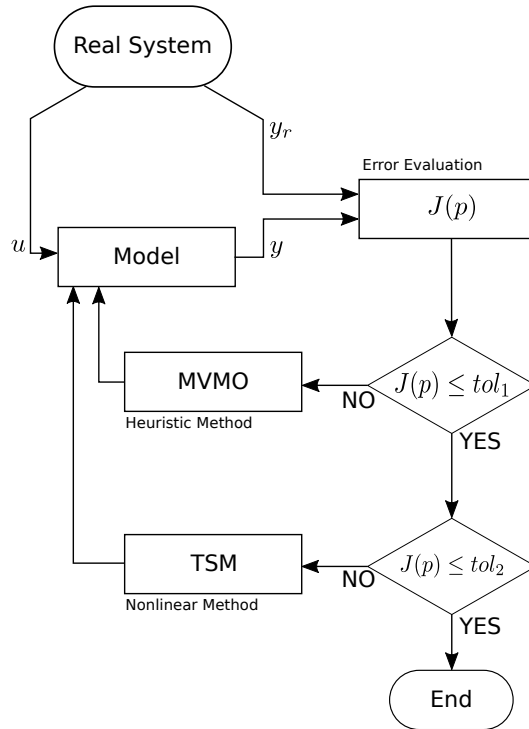
The Trajectory Sensitivity Method has fast convergence characteristics and can be applied directly to nonlinear problems, not requiring linearization. Also, by analyzing the sensitivities, the method provides information about identifiability of parameters. However, TSM is very sensitive to initial value of parameter chosen as starting point. Thus, if the initial values are too far from real, the method can either diverge or converge to wrong values. Besides, the information provided to the method must contain the effects of the parameters, otherwise they won't be observable (BENCHLUCH; CHOW, 1993).

3.3 Hybrid Estimation Method

By associating MVMO and TSM, the hybrid estimation approach applied in this work combines most benefits of both methods whilst mitigating their disadvantages. The resulting method has smaller convergence time when compared to MVMO alone and is less sensitive to initial values of parameters than TSM. The flowchart depicted in Figure 19 illustrates how this method works.

At first, a set of measurement data are recorded in the "Real System" block. These measurements must contain information of the dynamic response of the system. After that, part of the data is used as input to the block "Model", which is composed of equations of the mathematical model. The outputs of the "Real System" and "Model" are compared to evaluate the error function $J(\mathbf{p})$. While $J(\mathbf{p})$ is greater than a given tolerance tol_1 , MVMO algorithm will look for possible optimal solutions across the search region. Afterwards, the error will eventually drop to a value lower than tol_1 , and TSM will be used to refine the search to an optimal solution, with error level below tolerance tol_2 .

Figure 19 – Flowchart of hybrid estimation method

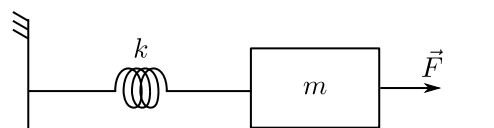


In order to assess the parameter estimation methods (TSM, MVMO and hybrid method), the parameter estimation of spring-mass model was conducted and its results are shown in the following section. Spring-mass is a nonlinear dynamic system and was chosen for this assessment due to its simplicity and reduced order.

3.4 Parameter Estimation of Spring-Mass Model

The spring-mass is a simple mathematical model often used as example of dynamic systems. It is composed of an object of mass m connected to a fixed point in space by a spring of stiffness constant k . The mass is on a surface with no friction and oscillates when disturbed by an external force \vec{F} . The system is depicted in Figure 20.

Figure 20 – Spring-mass system



By applying Newton's laws of motion on the system shown above, one can easily deduct that the movement of the mass is described by (3.14). The mass position and speed

are represented by x and v , respectively.

$$\begin{cases} \dot{x} = v \\ \dot{v} = \frac{k}{m}x - \frac{F}{m} \end{cases} \quad (3.14)$$

This system can be interpreted as of (2.8), with the state, input, output and parameter vectors given by (3.15), (3.16), (3.17) and (3.18), respectively.

$$\mathbf{x} = [x, v]^T \quad (3.15)$$

$$\mathbf{u} = [F] \quad (3.16)$$

$$\mathbf{y} = [x, v]^T \quad (3.17)$$

$$\mathbf{p} = [m, k]^T \quad (3.18)$$

The initial values of x and v are needed in order to solve the ordinary differential equation given by (3.14). Those values, as well as the magnitude of \vec{F} are given by:

$$\begin{cases} x(t = 0) = 0 \text{ m} \\ v(t = 0) = 0 \text{ m/s} \end{cases} \quad (3.19)$$

$$F(t \geq 0) = 4 \text{ N} \quad (3.20)$$

The system was simulated with parameters set at $m = 3 \text{ kg}$ and $k = 6 \text{ N/m}$ and the output behaviour obtained was used as reference for the estimation. Three different estimation processes were executed: TSM, MVMO and the hybrid approach with MVMO and TSM combined. The final tolerance for all three estimations was set at $tol = 0.1$.

3.4.1 Trajectory Sensitivity Estimation

It took 7 iterations (10 seconds on a PC) for Trajectory Sensitivity Method to estimate the parameters when the initial values were $m_0 = 7 \text{ kg}$ and $k_0 = 7 \text{ N/m}$. The final error of the estimation process was 1.90×10^{-3} . This shows how fast and precise this method is, as long as the initial values chosen are in the neighbourhood of the real parameters. Figure 21 shows the error evolution during estimation using TSM while Table 1 displays the values estimated for each parameter.

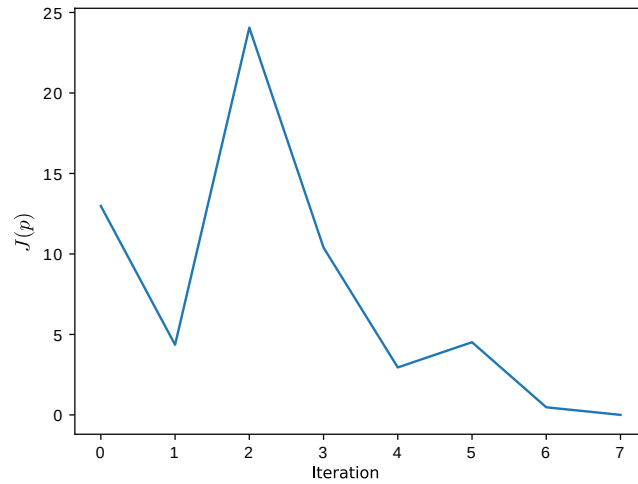
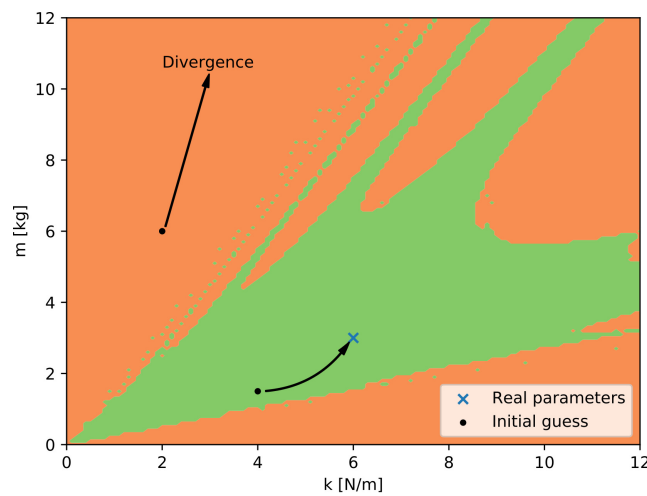
Figure 21 – Error evolution of TSM with $m_0 = 7$ kg and $k_0 = 7$ N/m

Table 1 – Parameters estimated by Trajectory Sensitivity Method

Parameter	Estimated value	Real value
m	3.080	3
k	6.145	6

However, the convergence region¹ of TSM is limited, diverging for initial points far enough from the real values. The convergence region for this simulation is shown as the green region in Figure 22. For comparison, the MVMO and the hybrid approach converge for the entire search region displayed on the graph.

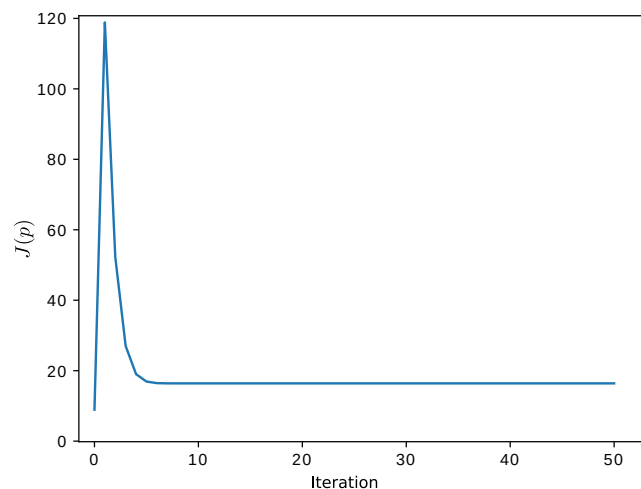
Figure 22 – Convergence region of Trajectory Sensitivity Method



¹ Convergence region is a region defined in the parameter space where, if the initial guess for parameter values is inside it, the convergence to true values is guaranteed.

To illustrate the importance of good initial values, the parameters were reestimated using TSM, but now it was selected initial values $m_0 = 8$ kg and $k_0 = 10$ N/m outside the convergence region. Notice that these values are not too far from the ones used in the previous estimation. The results, on the other hand, were extremely different. The method was not able to lower $J(\mathbf{p})$ below 16.4 and the parameters converged to different values $m_f = 2.7$ kg and $k_f = 118.6$ N/m. The error evolution for this estimation is depicted in Figure 23.

Figure 23 – Error evolution of TSM with $m_0 = 8$ kg and $k_0 = 10$ N/m



3.4.2 MVMO Estimation

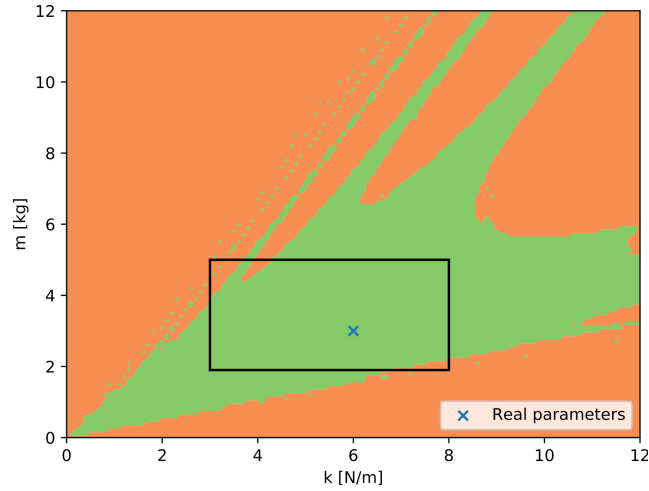
In order to estimate the parameter using MVMO, a search region in parameter space must be defined first. By the fact that MVMO will be compared to TSM, the parameter boundaries obtained from TSM convergence region displayed in Figure 22. The limits chosen define a region where TSM convergence is almost always guaranteed for any starting point inside of it. Figure 24 displays the search region chosen for MVMO estimation and Table 2 presents the parameter boundaries selected.

Table 2 – MVMO parameter boundaries

Parameter	Lower boundary	Upper boundary
m	1.9	5
k	3	8

The effects of population size were evaluated by running estimations for 4 different values: 5, 10, 20 and 50 individuals. For each population size, 35 estimation process were conducted with tolerance of $tol = 0.1$, maximum number of generation equals to $n_{gen} = 1000$ and one new individual generated at each generation. The average number of

Figure 24 – Search region chosen for MVMO



generations, duration and estimated parameter values (\bar{m} and \bar{k}) for each population size evaluated is shown in Table 3.

Table 3 – MVMO parameter boundaries

Population size	Average # of generations	Average duration (s)	\bar{m}	\bar{k}
5	71	32.23	3.23	6.20
10	27	15.43	3.38	6.05
20	5	10.22	3.45	5.96
50	1	20.72	3.66	5.76

From these results, one can infer that the optimal population size for estimating the parameter of the spring-mass model is around 20 individuals. It can also be noticed that, despite converging on the first generation, populations of size 50 usually take more time than others due to the large amount of individuals that must be generated and evaluated at the start of the estimation.

Figure 25 depicts the error evolution for one of the 35 estimations conducted for a population of 20 individuals. The heuristic method converged after 21 generations, with estimated values of $m = 2.6$ kg and $k = 5.2$ N/m.

3.4.3 Hybrid Estimation

By combining both methods, the hybrid approach benefits from the quick error reduction provided by MVMO and the fast convergence from TSM when inside convergence region. The tolerances for this approach were set at $tol_1 = 0.5$ (MVMO) and $tol_2 = 0.1$ (TSM). The population size was set at 20 individuals according to the results obtained in the previous section. Parameters of the spring-mass system were estimated using this method for 35 times and the average estimation time was 11.65 s.

Figure 25 – Error evolution of MVMO

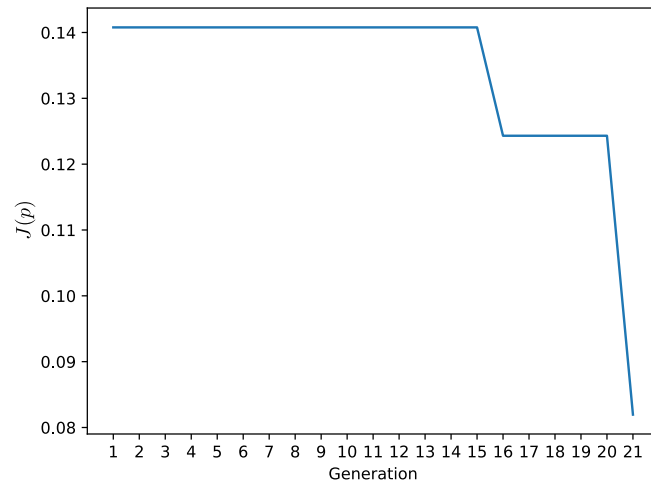
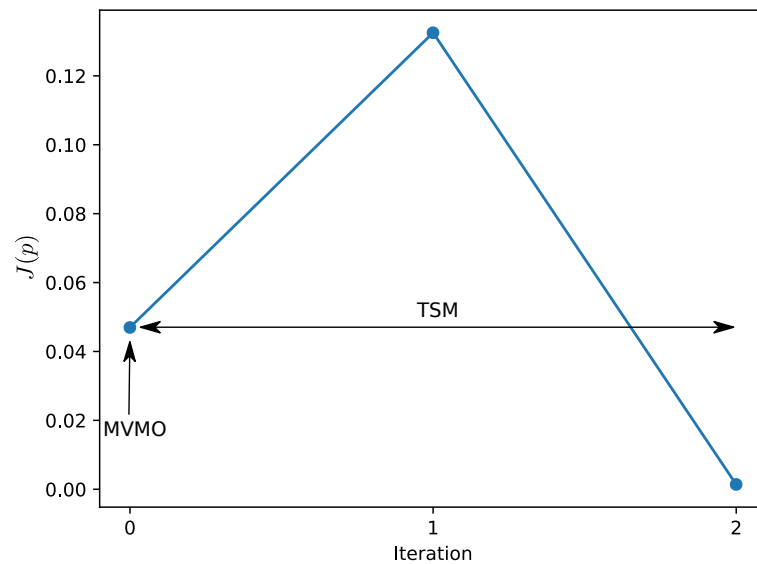


Figure 26 depicts the error evolution for one of the 35 estimations conducted. The hybrid estimation method converged after 3 iterations, with estimated values of $m = 3.063$ kg and $k = 6.134$ N/m.

Figure 26 – Error evolution with of hybrid approach for spring-mass system



When compared to each method, the hybrid approach converges faster than MVMO but slower than TSM, as shown in Table 4. Although all methods converged to parameter values that resulted in an error level below the tolerance, TSM and the Hybrid approach provided final results with error level below the values obtained by MVMO.

Table 4 – Comparison of approaches

Approach	Tolerance	Processing Time (s)	Final Error $J(\mathbf{p})$	\bar{m}	\bar{k}
TSM	0.1	7	2.76×10^{-3}	3.08	6.15
MVMO	0.1	10	52.32×10^{-3}	3.45	5.96
Hybrid	$tol_1 = 0.5$ $tol_2 = 0.1$	12	24.46×10^{-3}	3.08	6.15

The great advantage of the hybrid method is the increment of success in the convergence of parameters to the true value. This way, the MVMO approach is only used as an "intelligent initial guess".

4 IDENTPY: A SOFTWARE FOR PARAMETER ESTIMATION

In order to estimate the parameters of mathematical models, such as the one presented on chapter 2, a python library called *identPy* was developed by the author. *identPy* provides a customizable framework for parameter estimation with built-in mathematical models and estimation methods.

With this tool, comparison between performance of methods and precision of models can be easily done. Also, users are allowed to customize or create new features to match their needs without having to rewrite the entire framework, reducing time spent on coding. A Graphical User Interface (GUI) was developed as well, providing a simple environment where users can apply the estimation tool without any contact with the code.

The entire library and GUI were written in Python 3, a powerful, simple and fast high-level programming language that has gained large space in various sectors of industry and academy. Its rise is due mainly to the enormous number of libraries and forums developed and maintained by the users. Some examples of libraries used in this project are *numpy* (for scientific computing), *matplotlib* (graph plotting) and *PySide2* (GUI toolkit). Also, python is open-source, not requiring a paid software to code and most of its applications are free.

The following sections describe how the library is organized and illustrate the estimation process using *identPy* GUI.

4.1 *identPy* Library

identPy was developed as a python library so users can simply install the package and apply its features as needed. The library, as well as related scripts and files, is hosted on a [GitHub repository](#) and maintained by the author.

The package contains four submodules named **error**, **method**, **model** and **objects**. Each one of those submodules initialize core python classes developed for this framework. The entire library architecture was conceived in order to allow future contributors to easily work with it and help expand its applications.

The **error** module comprises the methods used to evaluate the functional error $J(\mathbf{p})$. Currently, only weighted l^2 -norm is implemented.

Inside **method** module, users will find the estimation methods implemented by the author and used on this study, as well as auxiliary scripts for parameter classification. Also, script containing a Particle Swarm Optimization algorithm implementation can be found in this module.

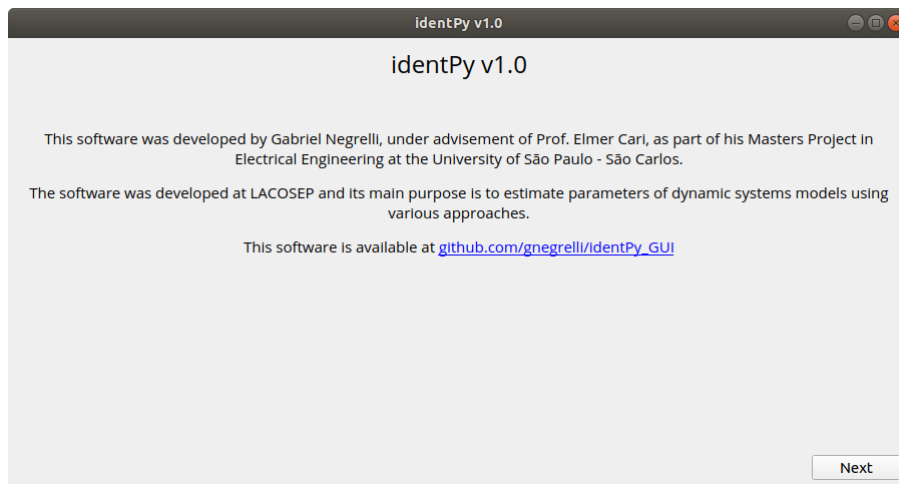
All classes with mathematical models are organized inside the **model** module. Besides the WPP equivalent model presented before, this module contains simple models used for testing, such as Spring-Mass and Pendulum systems, and other complex models used in previous studies, such as Linearized Z-IM Load Model. It also has a submodule within that implements implicit methods, such as the Runge-Kutta method, applied to obtain the behaviour of these models.

Finally, the estimator object and all abstract classes developed are organized inside **objects**. The Estimator object is responsible for building the estimation framework, by gathering the measured data, model and estimation methods, and executing it. The abstract classes located here constitute base objects from where the other classes can inherit.

4.2 *identPy* GUI

The *identPy* GUI was created with *Qt*, a software for design of user interfaces and applications. The views and scripts created specifically for the GUI application are hosted on a different [repository](#). When the application is launched, a starting window with information about the software and its development are displayed, as shown in Figure 27.

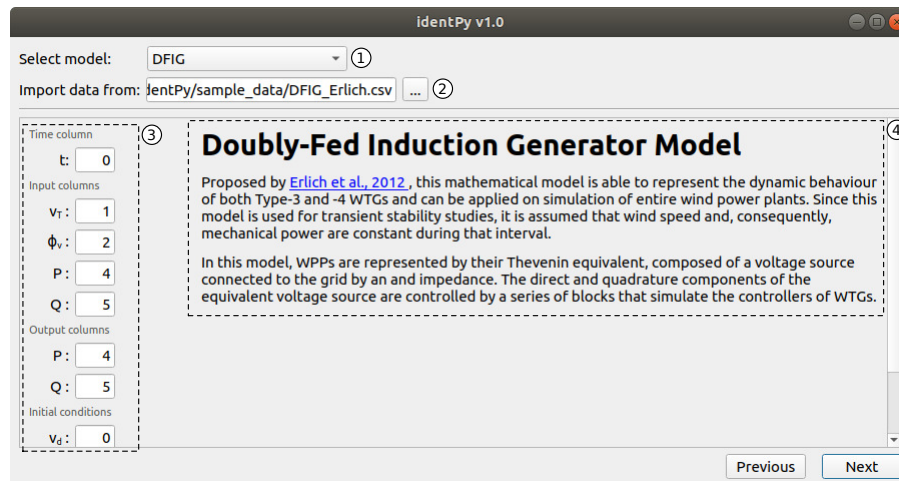
Figure 27 – *identPy* initial page



Next, a page for model selection is shown. The page layout is depicted in Figure 28. In this page, the user will choose, from a list indicated by "1", which mathematical model will be estimated. In "2", the user will be able to select the file containing the measured data used in the estimation process. The measured data must be in .csv, .dat or .txt format in order to be read correctly by the software. In the field indicated by "3", the user must identify which column of the data file corresponds to the inputs and outputs and what are the initial conditions of states. Notice that the column containing the time

measurement must be the first one on the file (indicated by the column index 0). The area indicated by "4" displays a short text with information about the model selected.

Figure 28 – Model selection page



On the next page, the user will be able to select two estimation methods from the list of available methods, as displayed in Figure 29. The drop-down list, indicated by "5", contains all methods available for estimation. The method selected in this field will be the first applied by the estimation tool. In "6", the user can indicate that the estimation will be conducted by two methods in cascade, as the hybrid method explained in the previous chapter. The second method to be applied can be selected in "7".

Figure 29 – Method selection page



The configuration of the chosen methods is done on the following windows. Each method has its own custom configuration page based on a template.

MVMO configuration page is depicted in Figure 30. For this method, the user must set general configurations, such as population size, number of offspring and tolerance. These general configurations are set on the fields indicated by "8". The user must also define what are the limits of the search region, which are set on the fields identified by "9". These fields change according to the choice made by the user on the model selection page.

Figure 30 – MVMO configuration page

IdentPy v1.0

MVMO Settings

Population Size 15 ⑧

Offspring 1

Max. Generation 5000

Tolerance 0.5

R : from 0.027 to 0.040 ⑨

X : from 0.16 to 0.24

k_i : from 5.58 to 8.37

T_i : from 0.028 to 0.043

T_v : from 0.22 to 0.32

k_{vc} : from 1.6 to 2.4

l_{max} : from 0.88 to 1.32

Previous Next

The configuration page of TSM, shown in Figure 31, is relatively similar to the MVMO page. General configurations (number of iterations and tolerance) are set in "10". The initial values of the parameters used as starting point by TSM are set in "11". If this is the second method of a hybrid approach, the initial values will be the ones found by the previous method. In this case, the values entered in "11" will be discarded by the software.

Figure 31 – TSM configuration page

IdentPy v1.0

Trajectory Sensitivity Settings

Max. Iteration 5 ⑩

Tolerance 5e-4

R : 0.03 ⑪

X : 0.2

k_i : 6

T_i : 0.04

T_v : 0.25

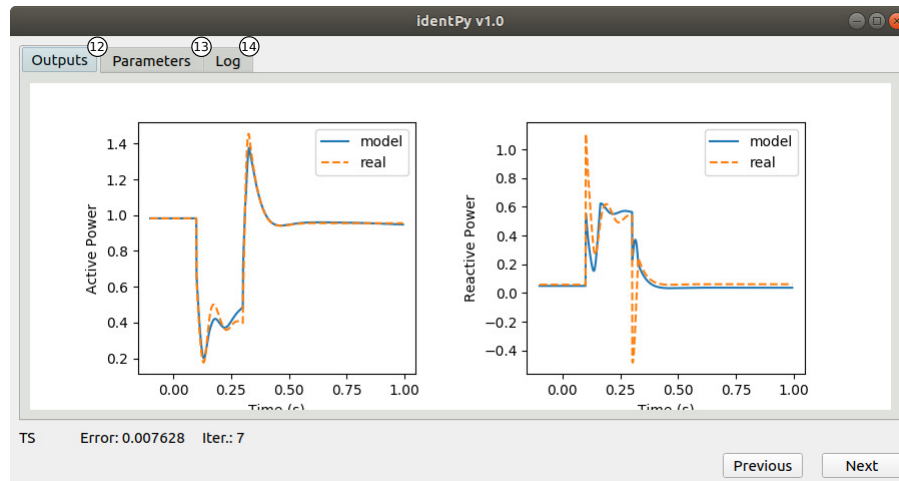
k_{vc} : 2

l_{max} : 1.1

Previous Estimate

With all set, the estimation process can now start. The page depicted in Figure 32 is displayed to the user while the estimation runs on background.

Figure 32 – Results page



On the tab indicated by "12", graphs depicting the outputs from the real system and the current solution are shown. The tab identified by "13" presents the current value of the parameters estimated by the software. A log with information of how the estimation process evolved is displayed in "14". The data shown in those three tabs are updated as the estimation process runs, with all information representing the current solution found.

As the estimation package evolves, other pages may be included in order to improve the software performance. For instance, as the methods list grows, configuration pages will be designed to set these new methods. Also, a page to analyze model identifiability may be included to enhance the estimation software.

5 ESTIMATION RESULTS AND DISCUSSIONS

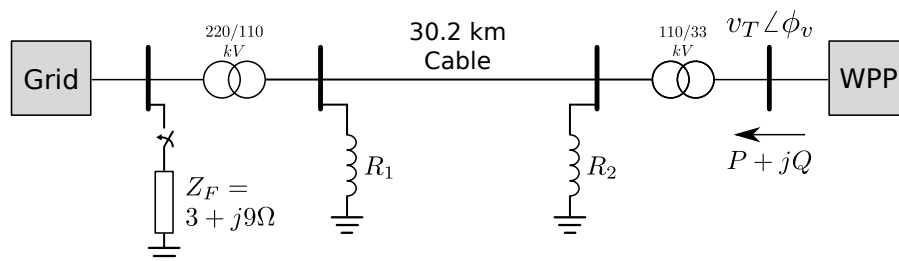
The estimation software was applied to the original mathematical model (equation 2.7) presented in Section 2.2, developed to represent wind power plants. The hybrid estimation method presented in Section 3.3 was applied, with MVMO providing a smart initial solution that was later be refined by TSM. The results of the estimation process are shown in the following section of this chapter. In addition, studies on the effects of population size on MVMO convergence time were also conducted and its results are presented. In the end, the application of the estimation software to the proposed mathematical model (equation 2.13) is also presented in this chapter.

Besides, in order to evaluate the package support on different electrical equipment models, it was used to estimate the parameters of a Linearized Z-IM Load Model. The estimation of this model is presented in the appendix A.

5.1 Parameter Estimation of Original WPP Model

The original model equation (2.7) developed in Section 2.2 was used in this case. The estimation was conducted using the disturbance data collected in (CARI; ALBERTO; ERLICH, 2015) as the real system output. In such study, a fault was simulated on a test system using *PowerFactory 14* and the data was used to estimate the parameters of the WPP model using only TSM. The system was simulated during 1 s with measurements taken every 0.001 s. The fault occurred at $t = 0.1$ s and cleared out by the protection devices at $t = 0.3$ s. Figure 33 displays the test system used to collect data and Figure 34 depicts the WPP bus voltage measured during simulation.

Figure 33 – Test system simulated



MVMO search region was defined as depicted in Table 5. These boundaries were decided based on the values previously found in (CARI; ALBERTO; ERLICH, 2015).

At first, 15 individuals (set of parameters) were randomly generated inside the given region, evaluated and ranked according to their error $J(\mathbf{p})$. At every generation,

Figure 34 – WPP bus voltage

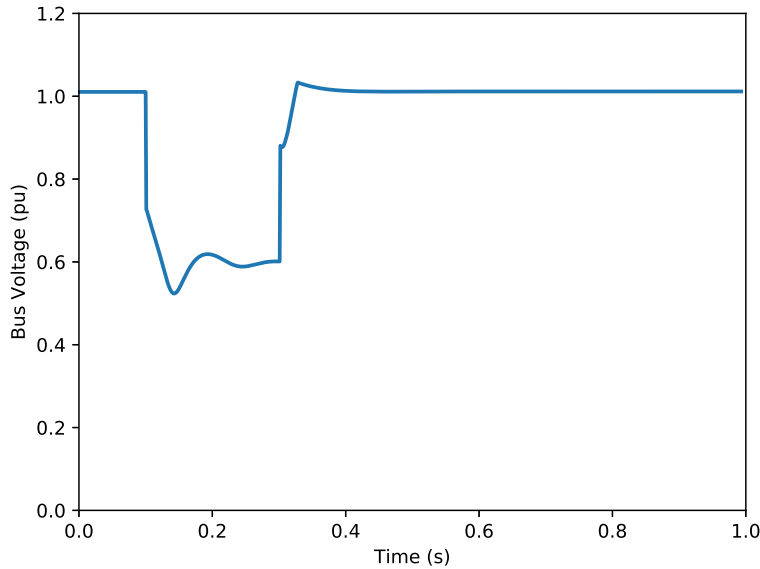


Table 5 – MVMO search region

Parameter	Lower boundary	Upper boundary
R	0.027	0.040
X	0.16	0.24
k_I	5.58	8.37
T_I	0.028	0.043
T_V	0.22	0.32
k_{VC}	1.60	2.40
i_{max}	0.88	1.32

three parameters of the fittest individual suffered mutation in order to generate a new individual. This process continued until the fittest individual reached an error level below a tolerance of $tol_1 = 0.5$. If this criterion were not met, MVMO would halt when the 5000th generation was reached.

The fittest individual found by MVMO was then used as the starting point to TSM. Since this method converges quickly when close to the optimal solution, it was configured to run for only 7 iterations. Its goal was to find a set of parameters with $J(\mathbf{p})$ below $tol_2 = 5 \times 10^{-4}$.

The parameters were estimated in 8.11 seconds in total, with MVMO and TSM taking 4.19 and 3.92 seconds, respectively, resulting in an error $J(\mathbf{p}) = 1.5 \times 10^{-4}$. The value estimated for each parameter is displayed in the Table 6.

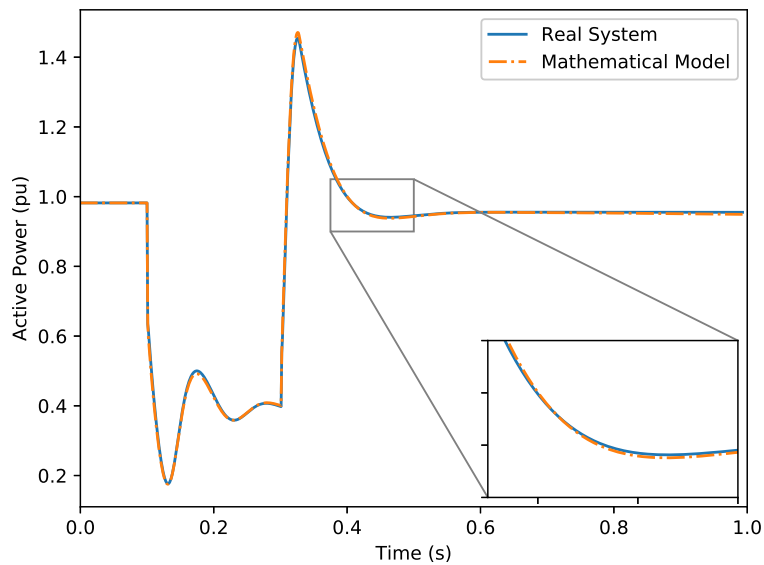
Figures 35 and 36 depict, respectively, real and active power measured from the simulated system and calculated using the WPP model with estimated parameters. For

Table 6 – Estimated values of parameters

Parameter	Estimated value
R	0.034
X	0.198
k_I	6.333
T_I	0.0348
T_V	0.246
k_{VC}	1.999
i_{max}	1.100

both components, the model output matches almost exactly with the curve expected. Therefore, the WPP model adjusted with the parameter vector found is able to reproduce the behaviour of the same wind power plant in similar conditions.

Figure 35 – Active power of real system (measured) and mathematical model (simulated) after parameter estimation



5.1.1 Influence of MVMO Population Size

In order to evaluate how the population size affects MVMO, a series of estimations using only this method were executed varying only the number of individuals. Five population sizes were chosen for this experiment (5, 10, 20, 50 and 100 individuals) and, for each one of them, 35 estimations were executed.

Apart from population size, all method configurations were fixed, so the results could be directly compared. The error tolerance was set at $tol_1 = 0.5$ and the search region was the same as presented in Table 5. The mean duration of estimation for each population size is shown below.

Figure 36 – Reactive power of real system (measured) and mathematical model (simulated) after parameter estimation

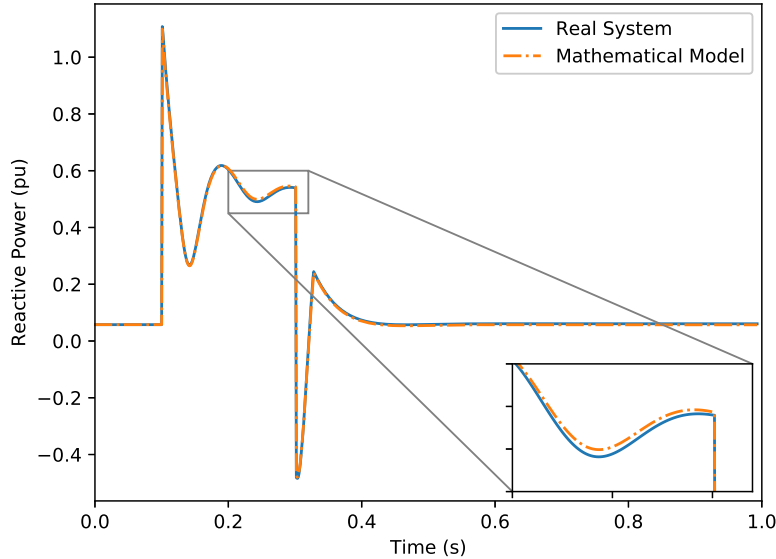


Table 7 – Influence of population size in MVMO

# of individuals	Average duration (s)
5	15.62
10	11.08
20	13.02
50	17.84
100	29.00

For considerably small populations (less than 5 individuals), the algorithm has to further explore the search region, due to reduced number of candidates. On the other extreme, large populations (more than 50 individuals) usually present good candidates in their initial population, only requiring a refinement of the solution. However, generating and evaluating all initial candidates has a huge cost, impacting on the estimation time. For these reasons, populations of 10 to 20 individuals present better convergence times than others, as depicted in Table 7.

5.2 Parameter Estimation of Proposed WPP Model

The proposed model equation (2.13) developed in Section 2.2 was used in this case. Given the similarities between original and proposed models, the simulations of the latter were conducted using the same parameter values found for the former. It was observed that, in order to adjust the pre-fault behaviour of the proposed model, the upper limit of the delay blocks should be increased to at least $V_{max} = 1.1$ pu. With these changes

made, the outputs of the proposed model (with the parameters presented in Table 6) were compared to the original WPP model, as displayed in Figures 37 and 38.

Figure 37 – Active power from original and proposed models

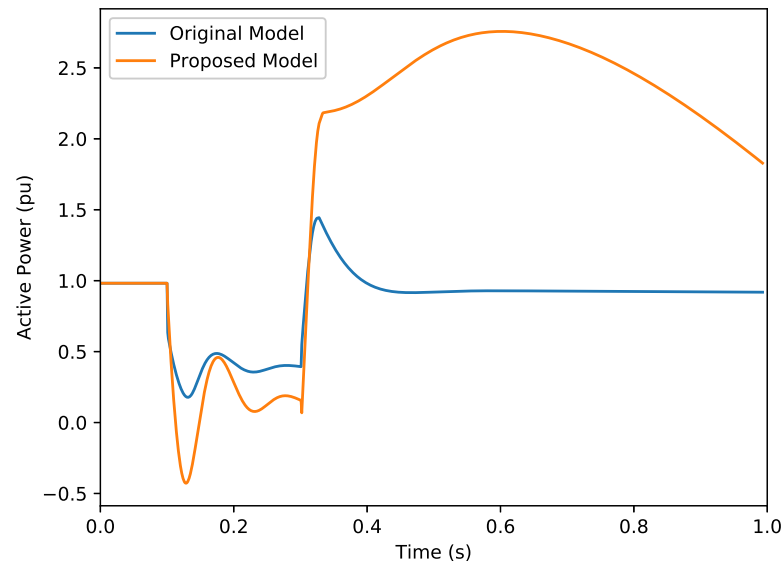
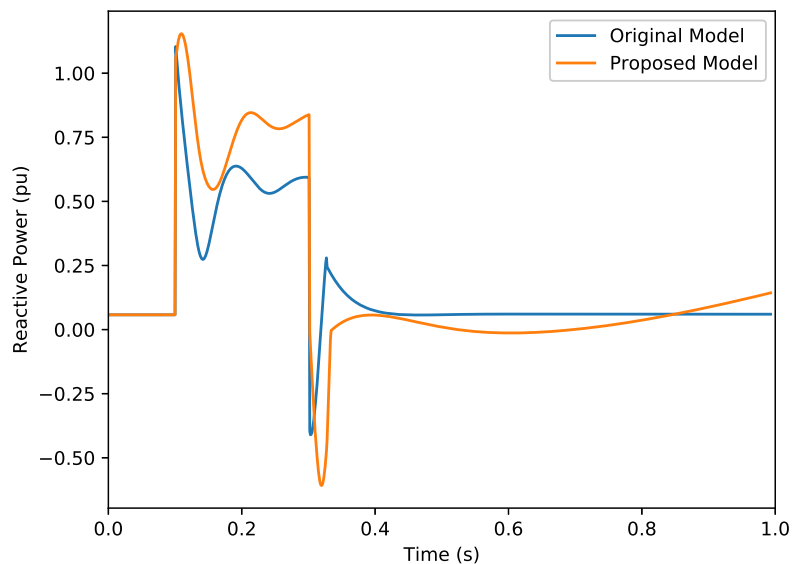


Figure 38 – Reactive power from original and proposed models



It can be noticed that reactive power obtained with the proposed model is not distant from the behaviour obtained with the original WPP model. Active power, on the other hand, diverged from the expected, specially during post-fault period.

Afterwards, the estimation of the parameters of the proposed model was conducted applying only MVMO. The tolerance was set at $tol = 1 \times 10^{-3}$ and the search region

defined as presented in Table 8. The parameter V_{max} was not included in the estimation and its value was set to 1.1.

Table 8 – MVMO search region used on estimation of proposed model

Parameter	Lower boundary	Upper boundary
R	0.01	0.1
X	0.05	0.8
k_I	3	15
T_I	0.01	0.2
T_V	0.1	1
k_{VC}	1	5
i_{max}	0.7	1.6

The resulting parameter vector is depicted in Table 9. Behaviours of active and reactive power for the proposed model are shown in Figures 39 and 40, respectively. The estimation process took 5000 generations (1133 seconds on i3 desktop) and the final error was $J(\mathbf{p}) = 15.84 \times 10^{-3}$.

Table 9 – Estimated parameters for proposed model

Parameter	Estimated value
R	0.030
X	0.646
k_I	3.246
T_I	0.200
T_V	0.971
k_{VC}	2.483
i_{max}	0.952
V_{max}	1.1

Despite the similarities, there are still differences between the behaviours of the proposed WPP model and the real system. These results indicate that the proposed WPP model is not entirely able to represent wind power plants yet, requiring further studies for improvement.

Figure 39 – Active power of proposed model after estimation

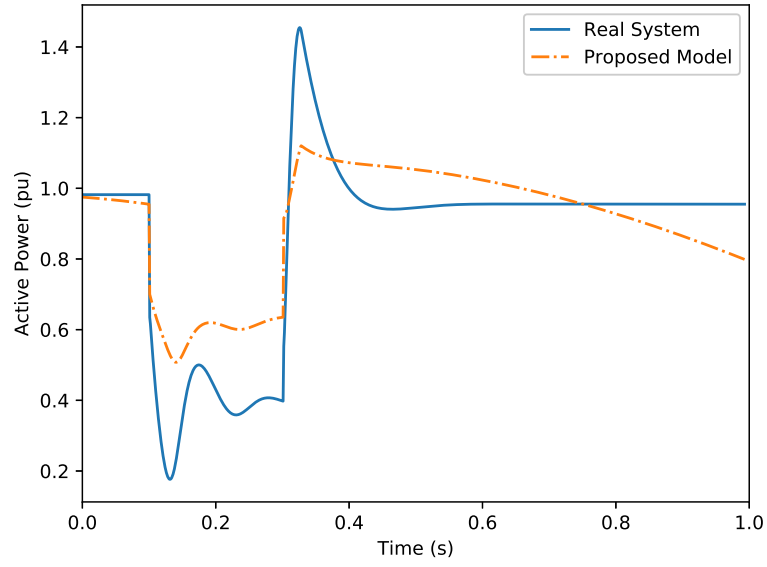
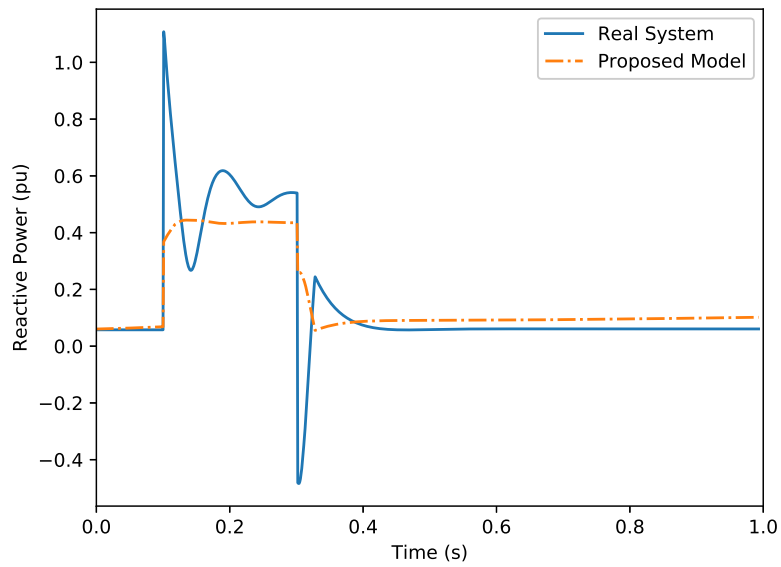


Figure 40 – Reactive power of proposed model after estimation



6 CONCLUSION

In this work, a software for parameter estimation of mathematical models was developed and applied on a wind power plant model. The software created consists of framework and GUI, both developed in Python 3, where users can configure an estimation tool and select which models and methods it will be apply. For this application, two estimation methods were developed: MVMO, a population-based metaheuristic, and TSM, a non-linear method based on Newton-Raphson. However, new methods can be easily added to the framework as they are required.

Initial studies have shown that, by combining both MVMO and TSM, the estimation method obtained is able to perform better than both methods separately. This hybrid method initially applies MVMO in order to sweep the search region and provide a solution candidate close to the optimal values. The solution candidate provided by MVMO is then refined using TSM, which quickly reduces the error between model output and measurements. The combination of both approaches provides a robust and quick estimation method.

The software was then applied to estimate the parameters of a WPP model using the hybrid method mentioned above. For this application, the WPP model parameters can be estimated using measurement data of voltage angle and magnitude (ϕ_v and v_T) and active and reactive power (P and Q) taken from the power plant terminal bus. Since measurements of voltage angle are needed by this model, the grid must contain special equipment, such as PMUs, installed on.

To avoid the requirement of voltage angle data and, thus, the installation of PMUs, a new model was proposed. This model, based on the one forementioned, would not need voltage angle measurements as input, allowing its application on a larger set of cases. However, initial tests on the proposed model were not successful and will be topic of ongoing research.

The software developed during the course of this work is available for download at https://github.com/gnegrelli/identPy_GUI.

7 ONGOING RESEARCH

7.1 Regarding the WPP proposed model

The mathematical model proposed in subsection 2.2.1 is promising since it does not require voltage angle measurements in order to represent the behaviour of WPPs. However, initial studies have shown that the model as proposed was not successful on simulating the wind power plant output. Possible reasons of this issue are:

- V_{max} is an unknown parameter and must be estimated;
- The value of V_{max} is not constant during fault;
- Reference values v_{Tref} , P_{ref} and Q_{ref} must be estimated with other parameters; and
- Missing blocks must be included to the model.

These and other possible reasons will be subject of further studies. The results of these studies will allow the application of the proposed model on cases where PMU measurements are not provided.

7.2 Regarding the Hybrid Method

The estimation framework and GUI developed can be further improved in the future. In order to improve the robustness of the hybrid method, the classical MVMO can be further replaced by its multiple swarm variant, called MVMO-SH.

In addition, new estimation methods, such as Kalman Filters, Monte Carlo, Particle Swarm Optimization and Differential Evolution, may be implemented to enlarge user options. Moreover, additional estimation steps, such as identifiability analysis, can be applied to validate and provide more information about the process. Improvements on GUI may surface as other features are created and new users start applying the tool.

8 LIST OF PUBLICATIONS

Below is presented a list of the publications generated during the course of this research.

- XXIII Congresso Brasileiro de Automática (2020) - "Parameter Estimation of Wind Power Plant Equivalent Model through a Hybrid Method"
- XIV Conferência Brasileira de Dinâmica, Controle e Aplicações (2019) - "Hybrid parameter estimation method for load model disturbed by OLTC"
- 2019 IEEE Canadian Conference of Electrical and Computer Engineering (2019) - "Load Model Identification Through a Hybrid Approach"

BIBLIOGRAPHY

- ABEEOLICA. **Annual Wind Energy Report**. [S.l.], 2020. 80 p.
- AMARANTE, O. A. C. do et al. **Atlas do Potencial Eólico Brasileiro**. Brasilia, 2001.
- ANEEL. **Atlas de Energia Elétrica do Brasil**. Brasilia, 2005. Disponível em: http://www.aneel.gov.br/livros/-/asset{_}publisher/eZ674TKh9oF0/content/atlas-de-energia-eletrica-do-brasil/656.
- ASMINE, M. et al. Model Validation for Wind Turbine Generator Models. **Power Systems, IEEE Transactions on**, v. 26, n. 3, p. 1769–1782, 2011. ISSN 0885-8950.
- BENCHLUCH, S. M.; CHOW, J. H. A Trajectory Sensitivity Method for the Identification of Nonlinear Excitation System Models. **IEEE Transactions on Energy Conversion**, v. 8, n. 2, p. 159–164, jun 1993. ISSN 15580059. Disponível em: <http://ieeexplore.ieee.org/document/222699/>.
- BLUM, C.; ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. **ACM Computing Surveys**, v. 35, n. 3, p. 268–308, sep 2003. ISSN 0360-0300. Disponível em: <http://portal.acm.org/citation.cfm?doid=937503.937505>.
- CARI, E. P.; ALBERTO, L. F.; BRETAS, N. G. A methodology for parameter estimation of synchronous generators based on trajectory sensitivity and synchronization technique. In: **2006 IEEE Power Engineering Society General Meeting, PES**. IEEE, 2006. p. 6 pp. ISBN 1424404932. Disponível em: <http://ieeexplore.ieee.org/document/1709492/>.
- CARI, E. P. T.; ALBERTO, L. F. C.; ERLICH, I. Assessment of model parameters to identify an equivalent wind power plant. In: **2015 IEEE Eindhoven PowerTech, PowerTech 2015**. IEEE, 2015. p. 1–5. ISBN 9781479976935. Disponível em: <http://ieeexplore.ieee.org/document/7232472/>.
- CHOI, B. K. et al. Measurement-based dynamic load models: Derivation, comparison, and validation. **IEEE Transactions on Power Systems**, 2006. ISSN 08858950.
- COUNCIL, W. E. C. Wecc wind power plant power flow modeling guide. **WECC Wind Generator Modeling Group, Tech. Rep**, 2008.
- ELLIS, A. et al. Generic models for simulation of wind power plants in bulk system planning studies. **IEEE Power and Energy Society General Meeting**, p. 1–8, 2011. ISSN 19449925.
- EPE. **Anuário Estatístico de Energia Elétrica 2019. Ano base 2018**. [S.l.], 2019. 254 p. Disponível em: http://epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-160/topico-168/Anu{á}rio_2019_W.
- ERLICH, I.; SHEWAREGA, F. Modeling of wind turbines equipped with doubly-fed induction machines for power system stability studies. p. 978–985, 2006.
- ERLICH, I. et al. Determination of Dynamic Wind Farm Equivalents using Heuristic Optimization. **Power and Energy Society General Meeting, IEEE**, p. 1–8, 2012.

ERLICH, I.; VENAYAGAMOORTHY, G. K.; WORAWAT, N. A Mean-Variance Optimization algorithm. **2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010**, n. February, p. 1–6, 2010.

European Commission. **A strategy for smart, sustainable and inclusive growth**. Brussels, 2010. 1–37 p. Disponível em: <<https://www.eea.europa.eu/policy-documents/com-2010-2020-europe-2020>>.

Federative Republic of Brazil. **Lei 10438/2002**. Brasilia: [s.n.], 2002. 1–21 p. Disponível em: <http://www.planalto.gov.br/ccivil/_03/leis/2002/L10438.htmhttp://www.planalto.gov.br/ccivil/_03/LEIS/2002/L104>.

GOMES, G. J. N. et al. Load Model Identification Through a Hybrid Approach. In: **2019 IEEE Canadian Conference on Electrical and Computer Engineering**. Edmonton: [s.n.], 2019.

MAGAGNA, D. et al. **Supply chain of renewable energy technologies in Europe: An analysis for wind, geothermal and ocean energy**. [S.l.], 2017. Disponível em: <<https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/supply-chain-renewable-energy-technologies-europe-analysis-wind-geothermal-and-ocean-energy>>.

MULJADI, E.; ELLIS, A. Validation of wind power plant models. **IEEE Power and Energy Society 2008 General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century, PES**, p. 1–7, 2008. ISSN 1932-5517.

MULJADI, E. et al. Short circuit current contribution for different wind turbine generator types. In: **IEEE PES General Meeting, PES 2010**. IEEE, 2010. p. 1–8. ISBN 9781424483570. Disponível em: <<http://ieeexplore.ieee.org/document/5589677/>>.

NAKAWIRO, W.; ERLICH, I.; RUEDA, J. L. A novel optimization algorithm for optimal reactive power dispatch: A comparative study. **2011 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)**, n. 1, p. 1555–1561, 2011. ISSN 0278-0046. Disponível em: <<http://ieeexplore.ieee.org/document/5994144/>>.

TODOROVSKI, M.; RAJIČIĆ, D. An initialization procedure in solving optimal power flow by genetic algorithm. **IEEE Transactions on Power Systems**, v. 21, n. 2, p. 480–487, 2006. ISSN 08858950.

Wind Europe. **Wind energy in Europe in 2019: Trends and statistics**. [S.l.], 2020.

XIONG, L. et al. Stability Enhancement of Power Systems With High DFIG-Wind Turbine Penetration via Virtual Inertia Planning. **IEEE Transactions on Power Systems**, v. 34, n. 2, p. 1352–1361, mar 2019. ISSN 0885-8950. Disponível em: <<https://ieeexplore.ieee.org/document/8463592/>>.

YARAMASU, V. et al. High-power wind energy conversion systems: State-of-the-art and emerging technologies. **Proceedings of the IEEE**, v. 103, n. 5, p. 740–788, may 2015. ISSN 00189219. Disponível em: <<http://ieeexplore.ieee.org/document/7109820/>>.

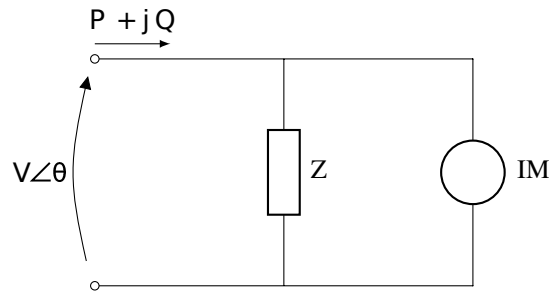
YOSHIDA, H. et al. A Particle swarm optimization for reactive power and voltage control considering voltage security assessment. **IEEE Transactions on Power Systems**, 2000. ISSN 08858950.

Appendix

APPENDIX A – PARAMETER ESTIMATION OF LINEARIZED Z-IM LOAD MODEL

The hybrid approach was also employed on parameter estimation of the linearized Z-IM load model. This model is able to predict the behaviour of electrical loads during faults in the grid. It is composed of an impedance in parallel with a third-order induction motor (IM), as shown in Figure 41.

Figure 41 – Schematic of Z-IM Load Model.



According to (CHOI et al., 2006), this model results in low error levels for both active and reactive power, alongside having a smaller parameter vector when compared with other models. It is described by equations (A.1), (A.2) and (A.3).

$$\begin{cases} \Delta \dot{E}' = \frac{-1}{T_o X'} [X \Delta E' + (X - X') V_0 \sin \delta_0 \Delta \delta] + \frac{(X - X') \cos \delta_0}{T_o X'} \Delta V \\ \Delta \dot{\delta} = \frac{(X - X') V_0}{T_o X' E'_0} \cdot \left(\frac{\sin \delta_0 \Delta E'}{E'_0} - \cos \delta_0 \Delta \delta \right) + \Delta \omega - \frac{(X - X') \sin \delta_0}{T_o X' E'_0} \Delta V \\ \Delta \dot{\omega} = \frac{-V_0}{M X'} (\sin \delta_0 \Delta E' + E'_0 \cos \delta_0 \Delta \delta) - \frac{E'_0 \sin \delta_0}{M X'} \Delta V \end{cases} \quad (\text{A.1})$$

$$\begin{cases} \Delta P = \frac{-V_0}{X'} (\sin \delta_0 \Delta E' + E'_0 \cos \delta_0 \Delta \delta) + \left(2G_s V_0 - \frac{E'_0 \sin \delta_0}{X'} \right) \Delta V \\ \Delta Q = \frac{-V_0}{X'} (\cos \delta_0 \Delta E' + E'_0 \sin \delta_0 \Delta \delta) + \left(2B_s V_0 + \frac{2V_0 - E'_0 \cos \delta_0}{X'} \right) \Delta V \end{cases} \quad (\text{A.2})$$

$$\begin{cases} X = X_s + X_m \\ X' = X_s + \frac{X_m X_r}{X_m + X_s} \\ T_o = \frac{X_r + X_m}{\omega_s R_r} \end{cases} \quad (\text{A.3})$$

where the terms $\Delta E'$ and $\Delta \delta$ represent the variation on voltage magnitude and angle at the motor terminals, $\Delta \omega$ is the variation on stator speed, in rad/s. X_m , X_s and X_r are the

magnetizing, stator and rotor reactances, respectively, R_r stands for the rotor resistance, ω_s is the synchronous speed, T_o represents the open-circuit transient time constant, M is the motor inertia and V_0 is the voltage on the load terminals before the disturbance. The input, state, parameter and output vectors are presented in (A.4), (A.5), (A.6) and (A.7), respectively.

$$\mathbf{u} = [\Delta V] \quad (\text{A.4})$$

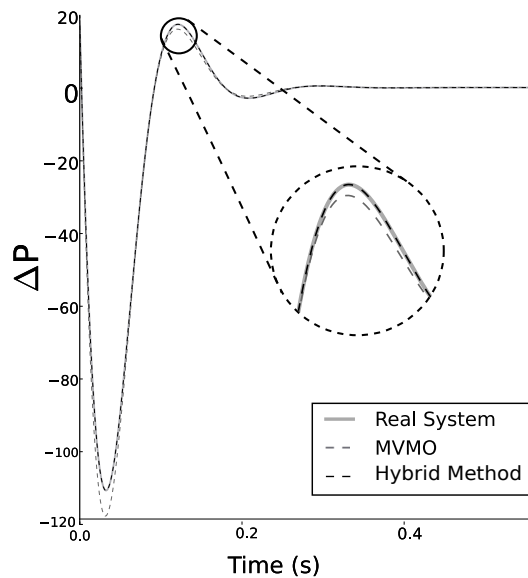
$$\mathbf{x} = [\Delta E', \Delta \delta, \Delta \omega]^T \quad (\text{A.5})$$

$$\mathbf{p} = [X, X', T_0, M, G_s, B_s, E'_0, \delta_0]^T \quad (\text{A.6})$$

$$\mathbf{y} = [\Delta P, \Delta Q]^T \quad (\text{A.7})$$

The hybrid approach proposed was able to estimate the parameters of this system and the comparison between real and modeled behaviour with the parameters obtained can be seen in the Figure 42.

Figure 42 – Result of parameter estimation for Z-IM Load Model



Source: (GOMES et al., 2019)

The application of the hybrid approach on Linearized Z-IM Load Model is subject of a paper presented by the author on the 2019 IEEE Canadian Conference on Electrical and Computer Engineering entitled “*Load Model Identification Through a Hybrid Approach*”.