COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

Robust tuning of turbine-governor models

Autor: Ignacio Loureiro Orejas

Director: Lukas Sigrist

Madrid

Julio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Robust tuning of turbine-governor models

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/2025 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Ignacio

Fdo.: Ignacio Loureiro Fecha: 23/07/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Lukas Sigrist Fecha: ....../ ....../ ......

ROBUST TUNING OF TURBINE-GOVERNOR MODELS

Autor: Loureiro Orejas Ignacio.
Director: Loureiro, Ignacio.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

La herramienta de ajuste de modelos de reguladores (AMOR) diseñada en el entorno Matlab/Simulink permite ajustar modelos de reguladores de velocidad mediante el método de mínimos cuadrados, gracias a la lectura incorporada de archivos COMTRADE y diagramas de bloques de los sistemas completos en Simulink.

Palabras clave: sistemas turbina y regulador, ajuste de modelos parametrizados, rechazo de carga, variación de frecuencia.

1. Introducción

Los sistemas de energía insulares, a diferencia de aquellos terrestres, carecen de interconexiones que les aporten inercia, por lo que son mucho más susceptibles de caer ante cualquier tipo de incidencia en sus generadores. Precisan, en mayor medida que sus homólogos terrestres, de un método fiable para prepararse ante estas incidencias, además de las protecciones eléctricas y un exhaustivo mantenimiento, un correcto modelaje de sus sistemas de turbina-regulador puede resultar realmente útil en esta tarea.

Con este objetivo en mente, se realizan en estos sistemas ensayos de rechazo de carga y variación de frecuencia. En los ensayos de rechazo de carga, se desconecta súbitamente la carga asociada al sistema en cuestión, observándose no solo la respuesta transitoria de la frecuencia, sino también la oscilación final en estado permanente. Contrariamente, en la variación de frecuencia, se modifica la entrada de frecuencia del sistema a estudiar y se observa el cambio en la potencia, una vez más, no siendo solo el transitorio importante sino también el valor final de la misma.

2. Definición del Proyecto

Los modelos paramétricos permiten, descritos mediante funciones de transferencia o sistemas de ecuaciones, crear modelos de sistemas reales, como es el caso de los sistemas de turbina-regulador. Gracias a la existencia de parámetros en estos modelos, tienen la capacidad de ser ajustados hasta cierto punto, dependiendo de la naturaleza de la función de transferencia o el número de incógnitas del sistema de ecuaciones.

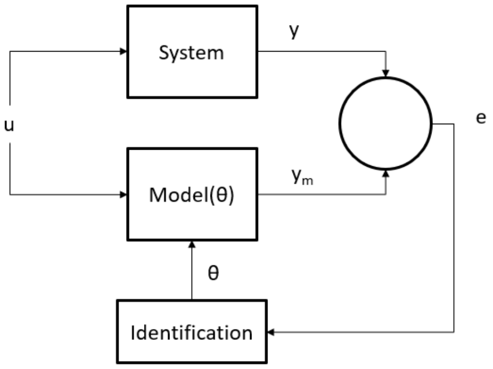


Figure 1: Esquema de identificación de parámetros

Como muestra la Figure 1, el método de ajuste de nuestro programa consistirá en minimizar la diferencia entre los datos proporcionados por los ensayos y la salida proporcionada por nuestro modelo parametrizado mediante el conjunto de parámetros θ.

3. Descripción del modelo/sistema/herramienta

La herramienta AMOR, perteneciente al entorno Matlab/Simulink utiliza dos módulos principales para llevar a cabo la lectura de datos y el ajuste de modelos como ilustra la Figure 2.

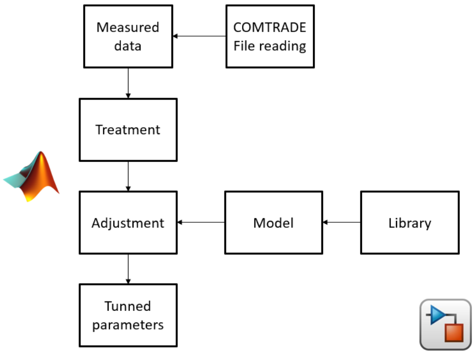


Figure 2: Diagrama de funcionamiento de AMOR

El primer módulo se encarga de leer los archivos COMTRADE que contienen los datos de los ensayos con sus respectivos vectores de tiempo, además, les aplicará a estos datos una serie de filtros y desfases para adaptarlo a las necesidades del algoritmo; para mejorar la experiencia del usuario, posee también una detección automática de escalón. El segundo módulo, consiste en una librería en la que se almacenan todos los modelos de los sistemas de turbina-regulador que se hayan desarrollado, así como sus parámetros iniciales desde los que partirá el algoritmo a la hora de realizar el ajuste. Una vez ambos módulos han realizado su trabajo, se obtienen unos parámetros óptimos, que aproximarán lo máximo posible a la realidad la simulación de nuestro modelo.

4. Resultados

Gracias al ajuste simultáneo de varios ensayos tanto de rechazo de carga como de variación de frecuencia, se ha conseguido exitosamente simular respuestas a incidencias de forma prácticamente idéntica a las obtenidas mediante la experimentación.

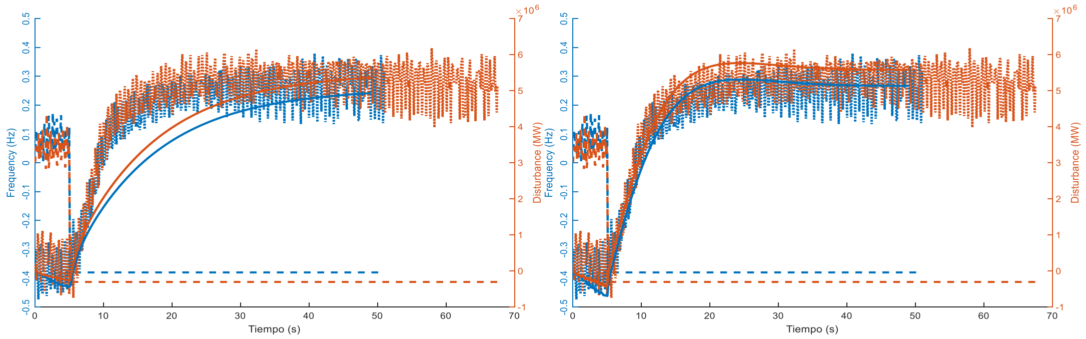


Figure 3: Simulación con parámetros iniciales y ajustados

Como se puede observar en la parte izquierda de la Figure 3, la simulación de un ensayo de variación de frecuencia con unos parámetros genéricos no es del todo útil, mientras que, si se usan parámetros ajustados como en la parte derecha, la simulación es prácticamente perfecta.

5. Conclusiones

La herramienta AMOR resulta ser realmente efectiva a la hora de ajustar y obtener los mejores parámetros para ciertos modelos de sistemas de turbina-regulador, especialmente de aquellos en los que se dispone de una gran cantidad de ensayos. Esto permite el uso del ajuste simultáneo, que mejora la generalidad de los resultados, disminuye la probabilidad de ajuste excesivo y aumenta el aprovechamiento de cada muestra.

Los resultados obtenidos resultan de gran utilidad, teniendo además en cuenta que solo se necesita el diagrama de bloques de dicho sistema y un ensayo de rechazo de carga o de variación de frecuencia. Además, este diseño le permite ser realmente versátil, pudiendo adaptarse a sistemas que aún no se han fabricado, por lo que su uso puede resultar útil no solo en la actualidad sino también en los próximos años.

ROBUST TUNING OF TURBINE-GOVERNOR MODELS

Author: Loureiro Orejas Ignacio.
Supervisor: Loureiro, Ignacio.
Collaborating entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

The model tuning tool AMOR, developed in the Matlab/Simulink environment, allows for the adjustment of speed governor models using the least squares method. This is made possible thanks to its built-in capability to read COMTRADE files and work with complete system block diagrams in Simulink.

Keywords: turbine-governor systems, parameterized model tuning, load rejection, frequency variation.

1. Introduction

Unlike mainland power systems, islanded grids lack interconnections that provide additional inertia, making them much more vulnerable to any type of disturbance affecting their generators. These systems require, even more than their mainland counterparts, a reliable method to prepare for such events. In addition to protective schemes and extensive maintenance routines, accurate modelling of their turbine-governor systems can be extremely useful in this task.

With this objective in mind, load rejection and frequency variation tests are carried out in these systems. In load rejection tests, the load connected to the system under study is suddenly disconnected. This allows for the observation of both the transient frequency response and the final steady-state oscillation. In contrast, during frequency variation tests, the input frequency to the system is deliberately modified, and the resulting change in power output is monitored. Once again, both the transient response and the final steady-state value are of interest.

2. Project definition

Parametric models, described through transfer functions or systems of equations, allow for the representation of real-world systems such as turbine-governor systems. Thanks to the presence of adjustable parameters, these models can be tuned to a certain extent, depending on the structure of the transfer function or the number of unknowns in the system of equations.

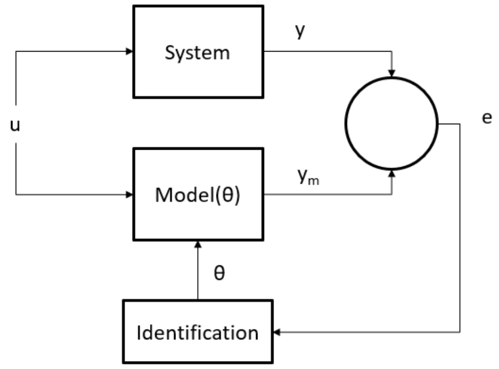


Figure 4: Parameter identification diagram

As shown in Figure 4, the tuning method implemented in our tool consists of minimizing the difference between the data obtained from the tests and the output provided by the parameterized model, using the parameter set θ.

3. Tool description

The AMOR tool, developed within the Matlab/Simulink environment, relies on two main modules to perform data reading and model tuning, as illustrated in Figure 5.

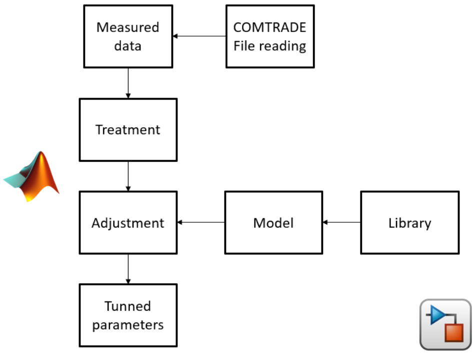


Figure 5: AMOR working flowchart

The first module is responsible for reading the COMTRADE files, which contain the test data along with their corresponding time vectors. It also applies a series of filters and offsets to these signals to adapt them to the needs of the algorithm. To improve user experience, it also includes an automatic step detection feature.

The second module consists of a library where all the turbine-governor system models developed so far are stored, along with their initial parameter sets, which serve as the starting point for the tuning algorithm. Once both modules have completed their tasks, the result is a set of optimal parameters that allows the simulation to closely approximate the real system behavior.

4. Results

Thanks to the simultaneous tuning of multiple samples, including both load rejection and frequency variation experiments, it has been possible to successfully simulate system responses that closely match those observed experimentally.

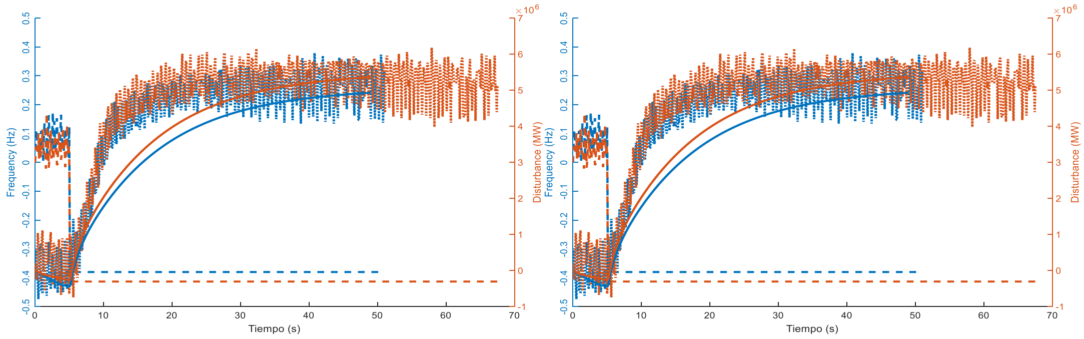


Figure 6: Initial and adjusted parameters simulation

As shown on the left side of Figure 6, simulating a frequency variation test using generic parameters is not particularly useful, as the results do not accurately reflect the system's behavior. However, when tuned parameters are used, as seen on the right side of the figure, the simulation becomes nearly perfect.

5. Conclusion

The AMOR tool has proven to be highly effective in tuning and obtaining optimal parameters for certain turbine-governor system models, especially those for which a large number of test data is available. This enables the use of simultaneous adjustment, which improves the generality of the results, reduces the likelihood of overfitting, and maximizes the value extracted from each data set.

The results obtained are highly valuable, particularly considering that the tool only requires the system's block diagram and a single load rejection or frequency variation test. In addition, its flexible design allows it to adapt to systems that have not yet been built, making it a useful solution not only for current applications but also for future developments.

Index

Figure Index

Equation Index

1. Introduction

1.1 Problem definition

Insular electrical systems, such as those found on the Canary Islands, face different challenges in maintaining grid stability, particularly in response to fault events. Unlike common continent power systems, which benefit from the interconnected support of larger networks, most insular grids operate in isolation, only relying on the limited local generation sources. This isolation results in significantly reduced system inertia, a critical factor that stabilizes frequency and minimizes the impact of disturbances. As a result, insular systems are far more susceptible to rapid frequency changes and instability when disturbances, such as generator or transmission lines faults, take place.

To effectively manage and predict the behavior of these systems under these conditions, it is essential to develop accurate models that can simulate generator responses. In this context, turbine-governor system models, are essential for the whole system's working analysis. To make sure these models reflect reality as closely as possible, their parameters need to be adjusted. Since frequency dynamics, which is key for frequency stability, depend mainly on system inertia and on the response of the turbine and speed control systems (turbine-governor systems), this project focuses specially on those models. Through these simulations, we can predict how different factors such as load variations or certain faults affect system stability. Matlab, with its powerful computational capabilities and Simulink with its modeling solutions, enable us to implement these models in a controlled environment where we can test, analyze, and optimize grid performance.

However, the accuracy of these simulations depends heavily on setting appropriate parameters within the models. Parameters such as governor time constants, droop settings, and turbine response times must be tuned to reflect the unique dynamics of each system and generator type. This tuning process is not easy, the values assigned to these parameters must be derived from real-world measurements and past fault data specific to the system. Consequently, models must realistically capture both general turbine and governor behavior.

Accurate parameter selection is critical because it allows the models to realistically emulate system dynamics, enabling us to predict faults and system responses with high precision. Improperly tuned parameters could lead to inaccurate simulations, potentially resulting in suboptimal or even hazardous control strategies when applied in real-world conditions. This makes the task of model calibration both technically complex and vital to ensure that simulations reflect reality. Our challenge, then, is to adjust these parameters through testing and optimization processes to develop accurate models.

Through this project, we aim to develop a robust, Matlab based simulation tool that facilitates precise parameter tuning for turbine and governor models. By doing so, we enable operators and engineers to anticipate system behaviors, identify potential instability risks, and design effective response strategies to mitigate the impact of faults. This work not only advances the reliability of insular electrical systems but also provides a foundational approach to modeling in environments where system inertia is limited, and grid stability is a constant priority.

1.2 State of the Art

Accurate parameter identification of generator models is essential for power system analysis to make sure that simulations and predictive models reflect real operational behavior. Several methods have been developed to address this challenge, using different mathematical approaches, system identification techniques, and optimization algorithms. These methods allow engineers to extract dynamic characteristics from experimental data, making sure that models can reliably predict generator responses to disturbances such as load rejection and frequency variations.

Traditionally, parameter identification is done using time-domain and frequency-domain methods. In the time domain, system responses obtained from controlled tests such as load rejection tests and frequency variation tests are analyzed to estimate key parameters. One widely used technique is least-squares optimization, characterized by the expression in Equation 1 for one sample and in Equation 2 for various samples, where parameters are adjusted to minimize the difference between simulated and real responses, as explored in [1]. This method has been particularly effective for estimating the governor dynamics and improving accuracy in transient stability studies. Additionally, output-error models have been widely used to minimize the mismatch between model outputs and real measurements, improving the accuracy of excitation system models, as detailed in [2].

$$J(\theta) = \sum_{k=1}^{N} \left(y(k) - \widehat{y_m}(k,\theta)\right)^2$$

Equation 1: Least mean squares parametric equation.

Where:

y(k) is the measured output of sample k

$\hat{y}$ (k,θ) is the predicted output using estimated parameters θ.

N is the number of data points.

For multiple samples the equation would be:

$$J(\theta) = \sum_{j=1}^{K} \sum_{i=1}^{N_j} \left(y_{ij} - \widehat{y_{ij}}(\theta)\right)^2$$

Equation 2: Parametric model error sum function for multiple samples

Where:

K is the number of test events.

y_{ij} is the measured output at sample i for event j.

$\widehat{y_{ij}}$ is the corresponding modeled output.

In contrast, frequency domain-based methods analyze system responses to sinusoidal perturbations, allowing for a more detailed characterization of generator dynamics. Frequency Response Analysis is a commonly used technique that extracts transfer functions from experimental data, enabling accurate estimation of governor and excitation system parameters, as described in *[3]*. Similarly, Fourier transform-based methods have been applied to extract relevant system characteristics in cases where inverter-based resources affect system inertia, a key issue highlighted in [4].

Beyond direct system identification, optimization algorithms play a key role in refining parameter estimations, especially for models with nonlinear dependencies. Metaheuristic optimization techniques have gained popularity due to their ability to explore complex parameter spaces. One of the most effective methods is Genetic Algorithms, which iteratively adjust parameter values based on their fitness to match observed system behavior, ideal for tuning tasks like ours.

This approach has been widely applied in [5]. Another widely used method is Particle Swarm Optimization (PSO), which iteratively improves a set of candidate solutions based on their performance compared to their neighbors. PSO has proven particularly useful in fine-tuning turbine-governor parameters, as shown in [6]**,** by iteratively adjusting parameter values based on swarm intelligence, PSO enhances voltage regulation and system stability, minimizing oscillations and improving dynamic performance.

Another interesting method for adjusting parameters are Particle Filters (PF) which have been successfully applied to estimate probability distributions of generator parameters, dynamically updating them as new data becomes available, as explained in [5]**.** In contrast, other approaches rely on iterative simulation-based fitting using physically grounded models, such as GAST and SEXS, combined with load rejection tests to extract generator dynamics. This methodology, as demonstrated in [7], provides accurate parameter estimation even under limited or noisy data conditions.

Several case studies have demonstrated the effectiveness of these techniques in real-world applications. In the power systems operated by ERCOT (Electric Reliability Council of Texas) and Svenska Kraftnät (Svk), frequency response models have been parameterized using historical event data, allowing for precise real-time assessment of system inertia and frequency control capabilities. Similarly, time-response identification methods have been applied to excitation systems, improving stability studies and enabling better tuning of automatic voltage regulators, as analyzed in [8]. Another relevant example comes from Lithuanian power plants, where operational data has been used to refine steam turbine models, ensuring an accurate representation of transient behaviors, as detailed in [9].

Despite these advancements, several challenges remain. Measurement noise and data quality continue to be limiting factors, as parameter estimation depends heavily on high-resolution and well-calibrated data, a challenge we will try to mitigate by using mean approximations when possible. Computational complexity is another issue, particularly for advanced optimization methods that require significant processing power to explore large parameter spaces efficiently, as seen in [10]. Additionally, the dynamic nature of modern power systems, where renewable energy integration and grid decentralization introduces new uncertainties, calls for adaptive identification methods capable of real-time parameter tuning, as discussed in [11].

The field of generator model parameterization is continuously evolving, with ongoing research focused on improving robustness and adaptability. Future developments are expected to integrate machine learning techniques, enabling dynamic parameter estimation based on continuously updated operational data, an area of study highlighted in [5]. As power systems become more complex and decentralized, the ability to accurately identify and adapt generator parameters will be key to ensuring stability and performance optimization in an increasingly dynamic grid environment.

1.3 Objectives

Under this project's objectives, we aim to develop a Matlab GUI tool that serves as a robust interface for analyzing and tuning turbine-governor system models based on real operational tests. Specifically, our tool processes test data from power generation systems, adjusting the parameters of simulation models to minimize the discrepancy between simulated outcomes and actual test results. Achieving close alignment between simulations and measurements is essential to accurately represent system behaviors under different conditions.

To determine the best possible fit for the turbine-governor system, the tool utilizes a least-squares method to optimize the selected parameters. This mathematical approach identifies the parameter set that minimizes the overall error between simulated and real test results, depending on which parameters are targeted for tuning. Through this method, the tool works to ensure the simulation is as close as possible to actual system responses, allowing us to more effectively analyze performance and predict outcomes under varied conditions.

The tool's primary function is to optimize parameter adjustments by evaluating multiple real test datasets simultaneously, rather than focusing on a single test result. This approach allows us to obtain a global fit that reflects the performance across various samples, helping create models that better approximate real responses. To achieve this, the program reads and processes multiple test files, automatically identifying the start point of each trial, which simplifies and standardizes data processing.

Beyond adjusting parameters, the tool will generate graphical plots of the samples, allowing for quick visual comparisons between simulated and given data. By ensuring that each test's unique starting point is correctly identified within the file, it allows for more consistent and reliable data usage for simulation.

Evaluating multiple samples collectively allows the tool to minimize the total error, ensuring the final model configuration will be optimized for all test data, not just a single trial. This goal of minimizing total error through the least-squares method ensures that simulation models are flexible enough to accurately represent a range of real-world scenarios, making them highly useful for forecasting, system planning, and responding to faults in isolated power systems with low inertia.

Ultimately, the tool provides a practical solution to parameter tuning by automating much of the process, making it more accessible for engineers to refine generation system models and align them closely with actual system behaviors.

1.4 Alignment with SDG

This project contributes to several Sustainable Development Goals (SDGs), primarily focusing on affordable and clean energy (SDG 7) and industry, innovation, and infrastructure (SDG 9).

By developing a tool that optimizes the parameters of models used for generating systems, we aim to enhance the efficiency of energy production, particularly in isolated systems like island power grids. These systems are often vulnerable to fluctuations, making reliable energy generation crucial. The adjustment of excitation systems and speed governing through accurate modeling and parameter tuning ensures that the energy produced meets demand while minimizing waste.

Additionally, the project promotes sustainable industry practices by applying robust methods to improve system performance and reliability. This is vital for promoting innovation within the energy sector and ensuring that energy infrastructure can withstand future challenges, including climate change.

Through this work, we not only address immediate energy needs but also contribute to a more sustainable and resilient energy future. Our efforts aim to support the transition towards renewable energy sources and more efficient energy management practices.

2. Tool for model adjustment

AMOR was developed to serve as an effective tool for model adjustment. AMOR is a Matlab & Simulink tool used to optimize the simulation-based equivalent models of real-world turbine-governor systems. As previously mentioned, accurate equivalent models are crucial for anticipating and managing incidents in the power grid, especially in isolated systems such as those in the Canary Islands.

2.1 Methodology

In this section, we present the method for tuning the parameters of the models used in our study. A system can be represented by either a parametric model or a non-parametric model. Parametric models are those described using transfer functions or systems of differential and algebraic equations. The models found in simulation software package libraries are typically parametric models. While parametric models require specifying the model structure a priori, they offer the advantage that when there are more measurements than unknowns (parameters to adjust), there is redundancy in the data.

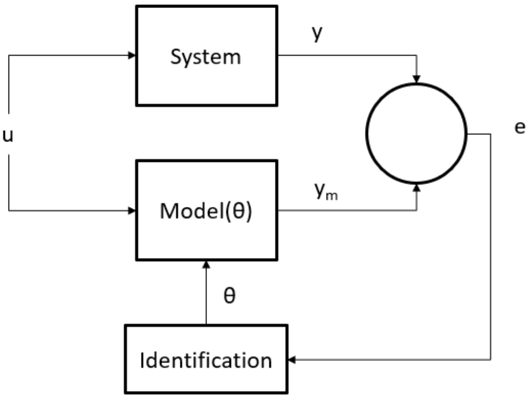


Figure 7: Parameter identifying schematic

Figure 7 illustrates a schematic for identifying the parameters of a given system model. The model is a parametric model, characterized by a vector of adjustable parameters θ. The tuning process involves minimizing the error, the difference between the measured output of the system and the model output in response to a disturbance, mathematically, this process would translate into Equation 3.

$$\min_{\theta} J(\theta) = \sum_{k=1}^{N} e(k,\theta)^2 = \sum_{k=1}^{N} \left(y(k) - y_m(k,\theta) \right)$$

Equation 3: Minimization of parametric simulation error.

Assuming that the noise is uncorrelated with the system input u, one can choose a model structure without a noise model, known as the output error structure. This structure is suitable for identifying a model for simulating the system without the need to model the noise. However, the objective function J(θ) is non-linear, requiring an iterative algorithm for minimization.

A linear model with constant coefficients can be described by a transfer function G(s,θ), which relates the input and output of the model in the Laplace domain, as shown generically in Equation 4.

$$Y_m\left(\mathrm{s},\theta\right) = G(s,\theta)\cdot \mathrm{U(s)}$$

Equation 4: Generic model description in Laplace domain.

If the measurement is sampled at a certain sampling time Ts, the linear model in the Laplace domain can be replaced by a discrete linear model like the one shown in Equation 5.

$$G(z,\theta) = \frac{Y_m(z,\theta)}{U(z)} = \frac{Z\left\{L^{-1}\left(G(s)\cdot U(s)\right)\Big|_{kTs}\right\}}{Z\left\{L^{-1}\left(U(s)\right)\Big|_{kTs}\right\}}$$

Equation 5: Model equation example.

The transfer function is given by:

$$G(z,\theta) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1\cdot z^{-1} + b_m\cdot z^{-m}}{a_0 + a_1\cdot z^{-1} + a_n\cdot z^{-n}}$$

Equation 6: Generic transfer function.

The output of the model in discrete time is:

$$y_m\left(k\right) + \sum_{i=1}^{n} a_i\cdot y_m\left(k-i\right) = \sum_{j=1}^{m} b_j\cdot u\left(k-j\right)$$

Equation 7: Discrete time output equation.

Given that, in Equation 7, both y_m (k) and y_m (k-i) depend on the parameters a_i and b_j, y_m (k) is a non-linear function of these parameters. Thus, to minimize the objective function, which aims to reduce the output error, as previously stated, an iterative algorithm must be employed.

This need for precise parameter estimation is especially relevant in turbine-governor systems, which play a key role in maintaining system stability during disturbances. A lack of understanding of their dynamic behavior under fault conditions can seriously compromise the grid's ability to respond effectively. For that reason, these systems are tested under scenarios such as load rejection and frequency variation. Dynamic experiments are essential for evaluating the system's stability and performance under realistic disturbances.

A load rejection test is performed by disconnecting a fraction or the totality of the load from a generator while it is operating at steady-state conditions. This simulates situations like faults or emergency shutdowns of large consumers. It provides insight into how the generator behaves in terms of frequency and voltage following a disturbance and allows for the assessment and fine tuning of excitation and governor control systems.

In a frequency variation test, the system's ability to maintain synchronization and return to steady-state conditions is evaluated. The input frequency is deliberately varied, either increased or decreased from its nominal value, to simulate grid instability caused by load imbalances, generation losses, or external perturbations. Frequency variation tests are particularly important in systems with low inertia, such as islanded or renewable-heavy networks, where frequency fluctuations are more frequent and severe.

Depending on the different test type selected, load rejection or speed variation, AMOR will be able to perform 3 different types of adjustment: single, mean or simultaneous.

In the single adjustment, only one sample is used, and the final model parameters are directly extracted from the least mean square (LMS) algorithm applied to that individual dataset.

The other two types of adjustments need 2 or more tests:

- In the **mean adjustment**, the LMS algorithm is applied separately to each sample, obtaining a set of parameters for each one. The final parameters are then computed as the **average** of all individual solutions.
- In the **simultaneous adjustment**, all samples are given together to the LMS algorithm, which generates a single set of parameters that minimizes the **total error** across all samples at once, rather than optimizing each one individually.

How this adjustment is performed will be explained further in this document.

AMOR is designed in the GUI environment, made up of two parts, the code and the user interface, in this part of the document we will go through the user interface in the normal order the tool would be used.

2.2 Toolbox Overview

Once we have explained the purpose of this tool and the methodology behind it, we will illustrate and review how the entire program functions, as well as the inputs it requires to achieve its purpose.

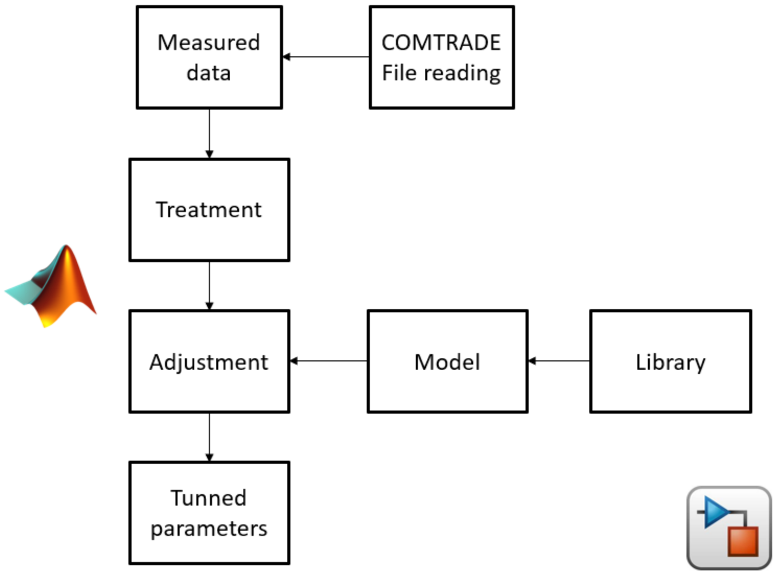


Figure 8: AMOR working flowchart

Figure 8 illustrates the main parts and inputs AMOR needs to work. It all starts with the turbine-governor system we want to model and simulate. We must build a Simulink model of said system with all its key parameters. This is done through Simulink and its block diagrams, depending on the system the model will vary a lot in number of parameters and relationships between blocks, which will also influence the difficulty of the adjustment. These diagrams are sometimes available online or in the system's manufacturer database. In our case, we will work with DEGOV1 and GAST2A systems, which have been carefully modeled as illustrated in Figure 9 and Figure 10.

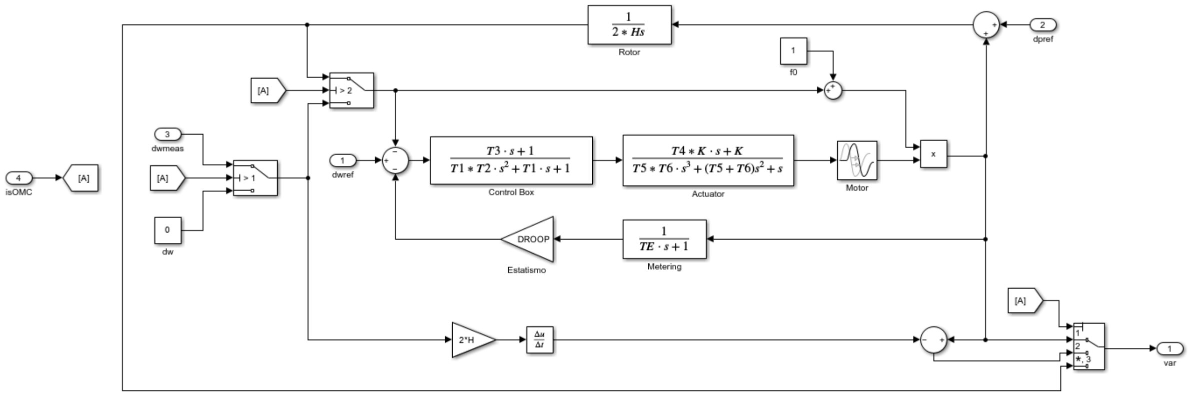


Figure 9: DEGOV1 turbine and speed regulator model

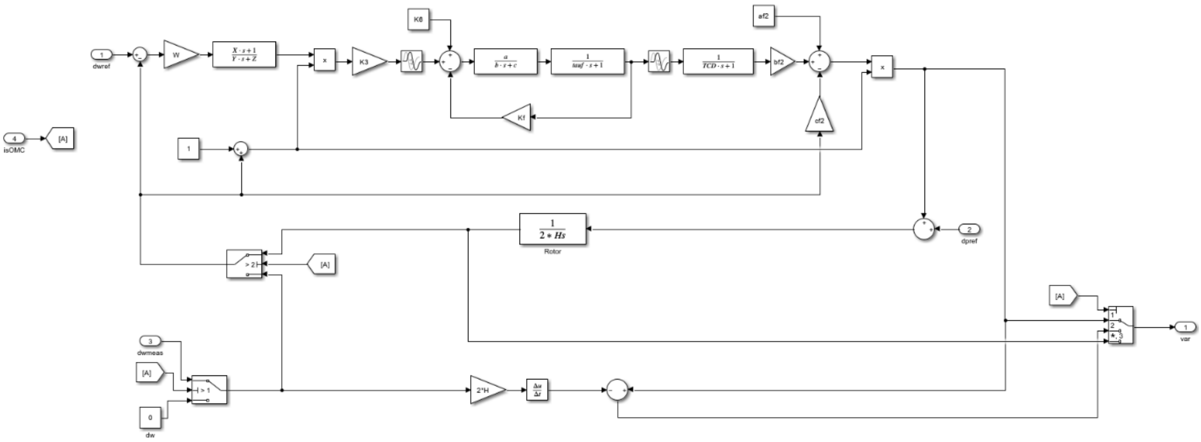
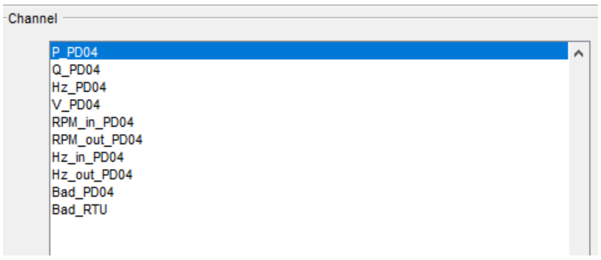


Figure 10: GAST2A turbine and speed regulator model

Once the model setup is finalized, we can go through the data branch of the chart. Like any simulation, data is needed, in our case, the data comes in COMTRADE format, a standardized file format used to store time stamped waveform data recorded normally by power monitoring equipment. In our case these files will always contain frequency and power records, as those are the magnitudes measured in both types of tests, but they may include other magnitudes also measured during the experiments as illustrated in Figure 11.


Channel

P_PD04
Q_PD04
Hz_PD04
V_PD04
RPM_in_PD04
RPM_out_PD04
Hz_in_PD04
Hz_out_PD04
Bad_PD04
Bad_RTU

Figure 11: COMTRADE file contents

The data in the COMTRADE file will undergo some treatment, such as time trimming or certain offset to compensate for some transit fluctuations. When the power and frequency of the desired test are selected, some input values will be needed, these may vary depending on the type of simulation or adjustment, but some like frequency base, system's power base or initial and final time of the simulation will always be needed; these will be explained more thoroughly in the next section.

When both the model and the data are selected and treated according to the test and turbine-governor system, we can proceed to the adjustment procedure. This part relies heavily on the parameters, as there is a limit as to how many can be adjusted, not only due to time complexity or computational limitations, but also due to the systems nature, for example, in a second order system response, only four parameters can be adjusted, two poles and two zeros.

If all the data and parameters have been inputted correctly, the program will return the best fitting values for the parameters selected; the aim of these parameters is to be the best but also most general parameters possible. We don't want parameters that can perfectly replicate one sample's output but at the cost of being more imprecise in other samples of the same system. To avoid this overfitting possibility, we have developed the **simultaneous adjustment**, which feeds the algorithm several samples at a time to find the parameters that best replicate the general system's response, how this is done will be explained more in detail later in the document.

2.3 User guide

2.3.1 Reading Interface Description

To start the program, we can either press the Run button on the top of the MATLAB editor or by writing AMOR on the console. Once AMOR has loaded all the necessary components for its correct operation the launch screen will appear as depicted in Figure 12.

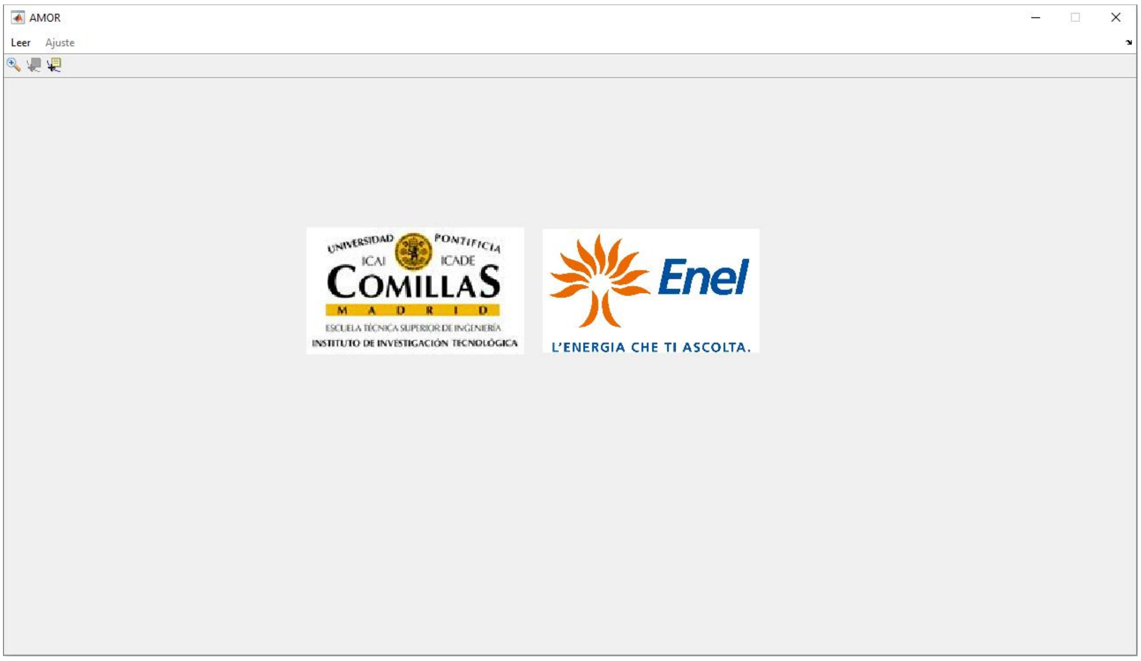
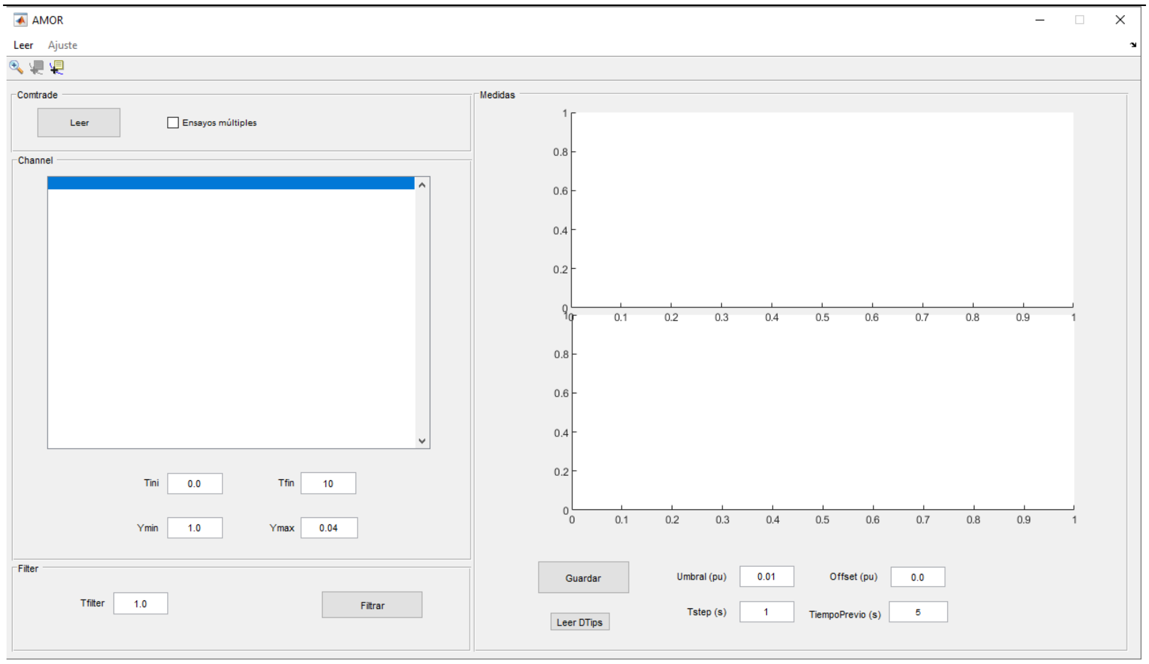

AMOR
Leer
Ajuste
UNIVERSIDAD PONTIFICIA
ICAI ICADE
COMILLAS
MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INSTITUTO DE INVESTIGACIÓN TECNOLÓGICA
Enel
L'ENERGIA CHE TI ASCOLTA.

Figure 12: AMOR launch screen

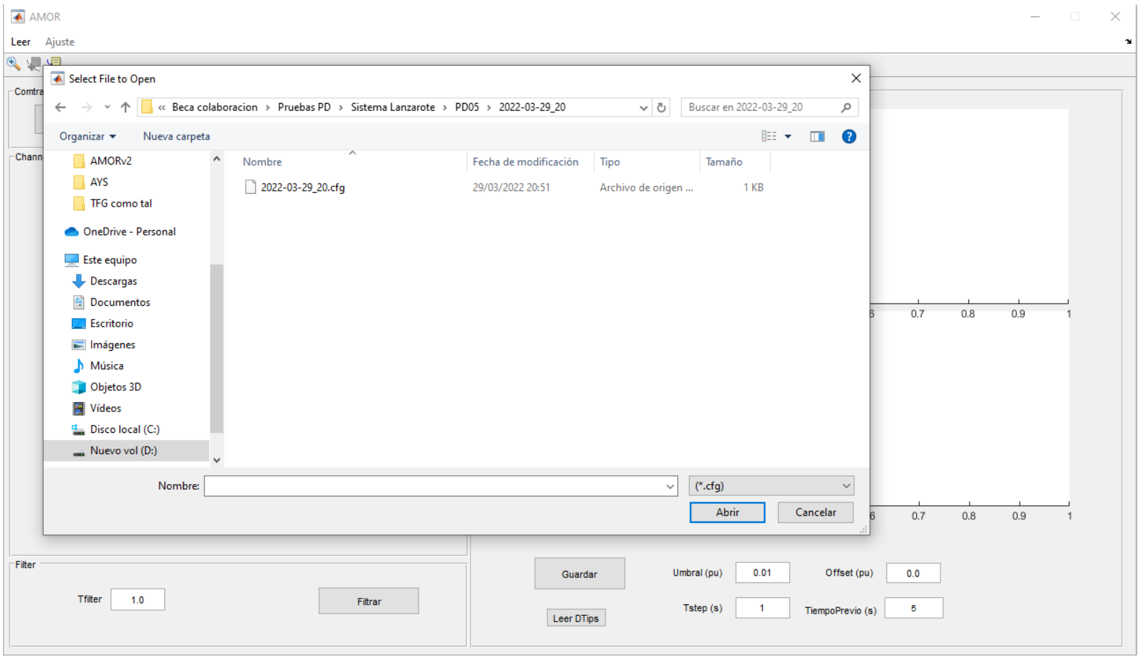
We begin by clicking the only available button, labeled "Leer". Once pressed, the interface switches to the data reading section of the program, displaying a new set of buttons and checkboxes, as shown in Figure 13.



Tini 0.0
Tfin 10
Ymin 1.0
Ymax 0.04
Tfilter 1.0
Umbral (pu) 0.01
Offset (pu) 0.0
Tstep (s) 1
TiempoPrevio (s) 5

Figure 13: AMOR reading interface.

Then, by clicking the "Leer" button located under the "COMTRADE" section, the file explorer will open, allowing us to select the dataset to be used for the adjustment, as shown in Figure 14.



« Beca colaboracion › Pruebas PD › Sistema Lanzarote › PD05 › 2022-03-29_20

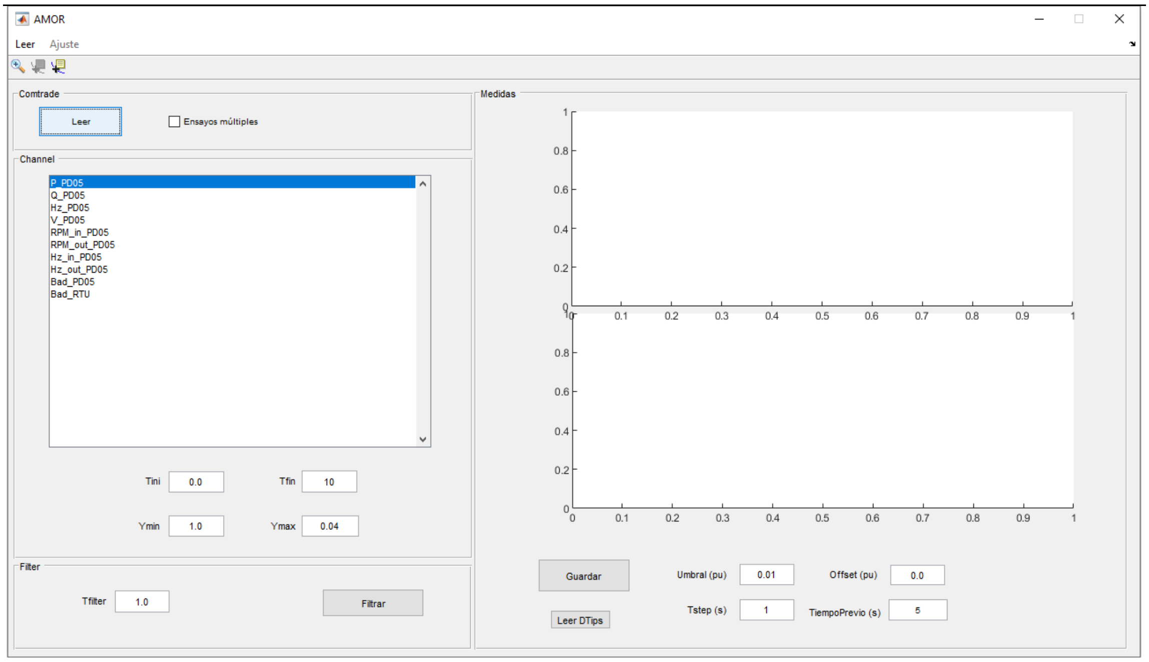
Nombre	Fecha de modificación	Tipo	Tamaño
2022-03-29_20.cfg	29/03/2022 20:51	Archivo de origen ...	1 KB

Nombre:
(*.cfg)

Tfilter 1.0
Umbral (pu) 0.01
Offset (pu) 0.0
Tstep (s) 1
TiempoPrevio (s) 5

Figure 14: Comtrade file selection process.

After selecting the corresponding Comtrade file, If the reading process is successful, the program will display the signals available as shown in Figure 15.



□ Ensayos múltiples

Channel

P_PD05
Q_PD05
Hz_PD05
V_PD05
RPM_in_PD05
RPM_out_PD05
Hz_in_PD05
Hz_out_PD05
Bad_PD05
Bad_RTU

Tini 0.0 Tfin 10

Ymin 1.0 Ymax 0.04

Filter

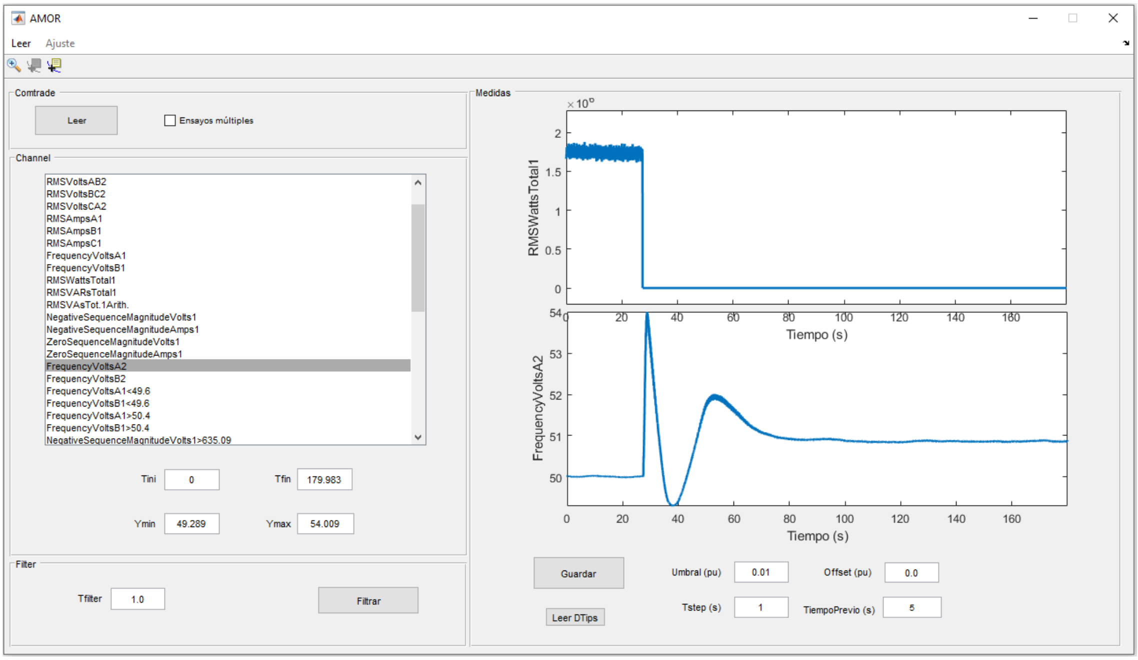
Tfilter 1.0

Umbral (pu) 0.01 Offset (pu) 0.0

Tstep (s) 1 TiempoPrevio (s) 5

Figure 15: Comtrade contents organizer.

Once the data is available, we can select the specific signals we wish to use for the adjustment. In both cases, we will choose power and frequency, clicking once to select frequency and twice to select power. The interface at this stage will resemble what is shown in Figure 16 and Figure 17.



□ Ensayos múltiples

Channel

RMSVoltsAB2
RMSVoltsBC2
RMSVoltsCA2
RMSAmpsA1
RMSAmpsB1
RMSAmpsC1
FrequencyVoltsA1
FrequencyVoltsB1
RMSWattsTotal1
RMSVARsTotal1
RMSVAsTot.1Arith.
NegativeSequenceMagnitudeVolts1
NegativeSequenceMagnitudeAmps1
ZeroSequenceMagnitudeVolts1
ZeroSequenceMagnitudeAmps1
FrequencyVoltsA2
FrequencyVoltsB2
FrequencyVoltsA1<49.6
FrequencyVoltsB1<49.6
FrequencyVoltsA1>50.4
FrequencyVoltsB1>50.4
NegativeSequenceMagnitudeVolts1>635.09

Tini 0 Tfin 179.983

Ymin 49.289 Ymax 54.009

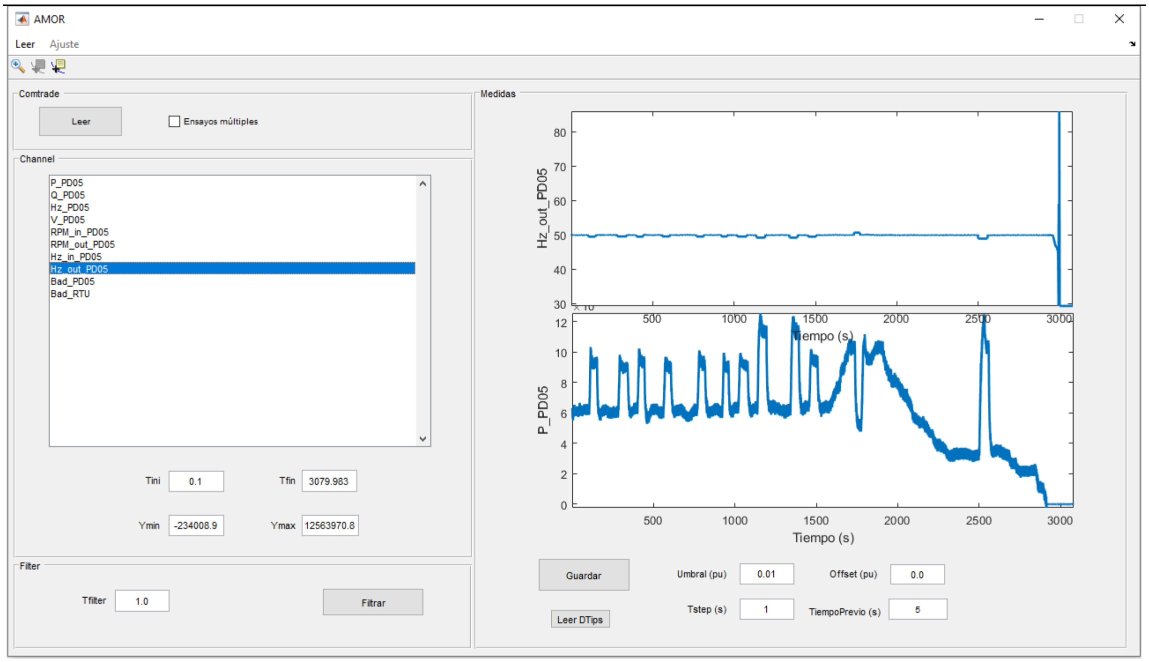
Filter

Tfilter 1.0

Umbral (pu) 0.01 Offset (pu) 0.0

Tstep (s) 1 TiempoPrevio (s) 5

Figure 16: Generic load rejection test data.



□ Ensayos múltiples

Tini 0.1

Tfin 3079.983

Ymin -234008.9

Ymax 12563970.8

Tfilter 1.0

Umbral (pu) 0.01

Offset (pu) 0.0

Tstep (s) 1

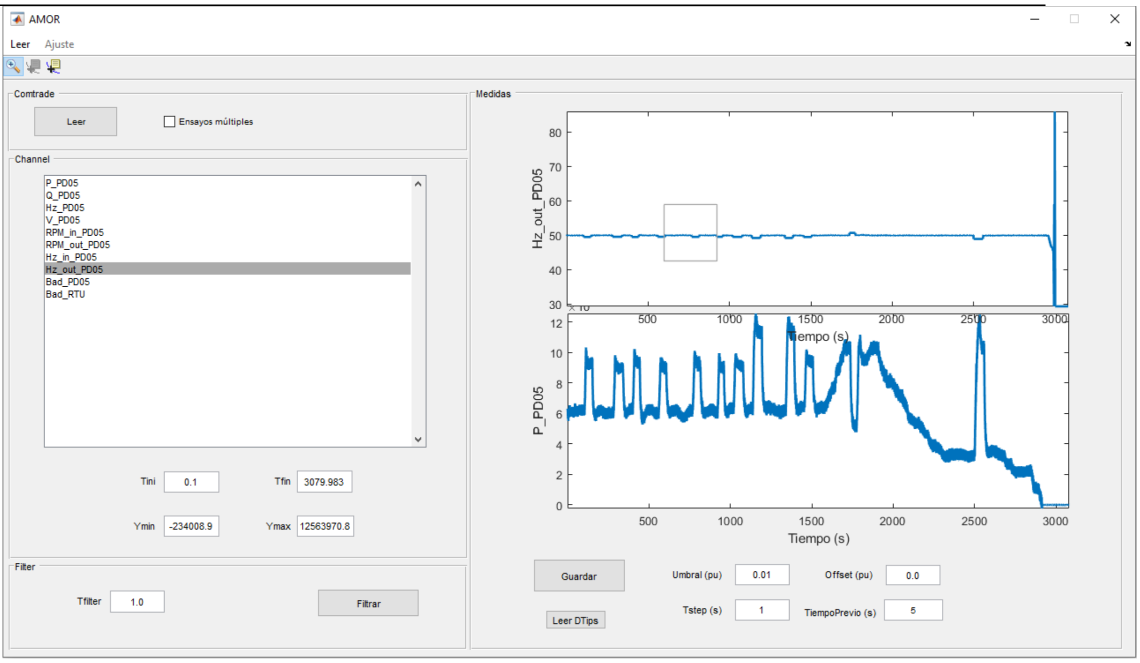
TiempoPrevio (s) 5

Figure 17: Generic speed variation test data.

When performing a load rejection test, only one sample is handled at a time. Therefore, neither the mean nor the simultaneous adjustment modes are available. As a result, no additional selection process is required; only the time range needs to be specified.

On the other hand, frequency variation tests typically consist of multiple examples that can be selected individually or collectively. To choose a specific example, the following steps should be followed:

1. Click the magnifying glass icon located at the top left of the interface, and use it on either of the two selected graphs to zoom into the desired example, as illustrated in Figure 18, Once the zoom is applied, the data should appear as shown in Figure 19.



□ Ensayos múltiples

P_PD05
Q_PD05
Hz_PD05
V_PD05
RPM_in_PD05
RPM_out_PD05
Hz_in_PD05
Hz_out_PD05
Bad_PD05
Bad_RTU

Tini 0.1 Tfin 3079.983

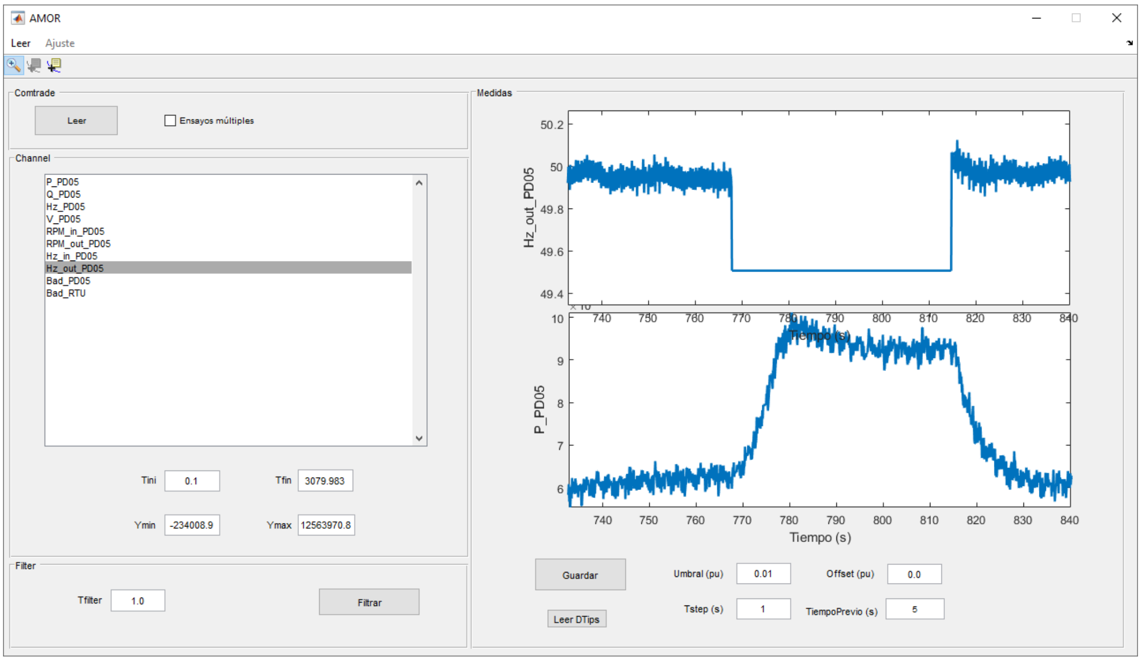
Ymin -234008.9 Ymax 12563970.8

Tfilter 1.0

Umbral (pu) 0.01 Offset (pu) 0.0

Tstep (s) 1 TiempoPrevio (s) 5

Figure 18: Magnifying glass usage guide.



□ Ensayos múltiples

P_PD05
Q_PD05
Hz_PD05
V_PD05
RPM_in_PD05
RPM_out_PD05
Hz_in_PD05
Hz_out_PD05
Bad_PD05
Bad_RTU

Tini 0.1 Tfin 3079.983

Ymin -234008.9 Ymax 12563970.8

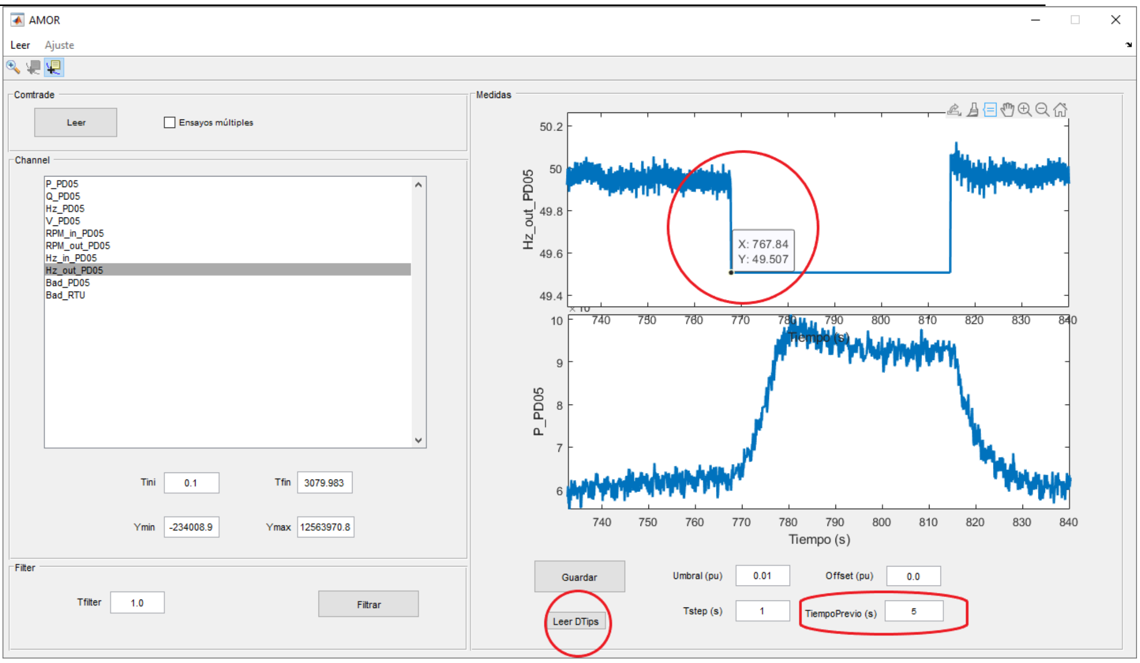
Tfilter 1.0

Umbral (pu) 0.01 Offset (pu) 0.0

Tstep (s) 1 TiempoPrevio (s) 5

Figure 19: magnified data example.

2. Once only one example is visible on the screen, define how much time you want to display before the step occurs by entering the desired value in the "Tiempo Previo (s)" box. Then, click the "Leer DTips" button located at the bottom of the interface. Once everything is set, place the tooltip on the appropriate point of the graph to mark the step moment. All these features are located as shown in Figure 20.



☐ Ensayos múltiples

P_PD05
Q_PD05
Hz_PD05
V_PD05
RPM_in_PD05
RPM_out_PD05
Hz_in_PD05
Hz_out_PD05
Bad_PD05
Bad_RTU

Tini 0.1 Tfin 3079.983

Ymin -234008.9 Ymax 12563970.8

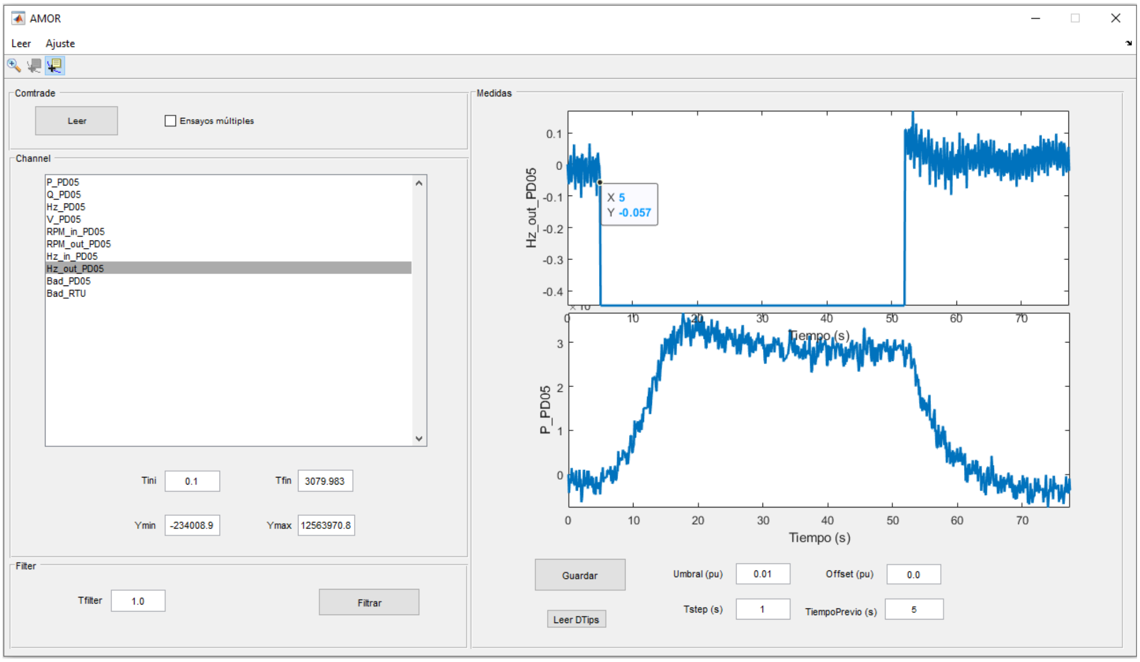
Tfilter 1.0

Umbral (pu) 0.01 Offset (pu) 0.0

Tstep (s) 1 TiempoPrevio (s) 5

Figure 20: Leer DTips mode of operation.

3. After completing the previous steps, the program will automatically adjust the data to the correct time and value scales, in accordance with Figure 21.



☐ Ensayos múltiples

P_PD05
Q_PD05
Hz_PD05
V_PD05
RPM_in_PD05
RPM_out_PD05
Hz_in_PD05
Hz_out_PD05
Bad_PD05
Bad_RTU

Tini 0.1 Tfin 3079.983

Ymin -234008.9 Ymax 12563970.8

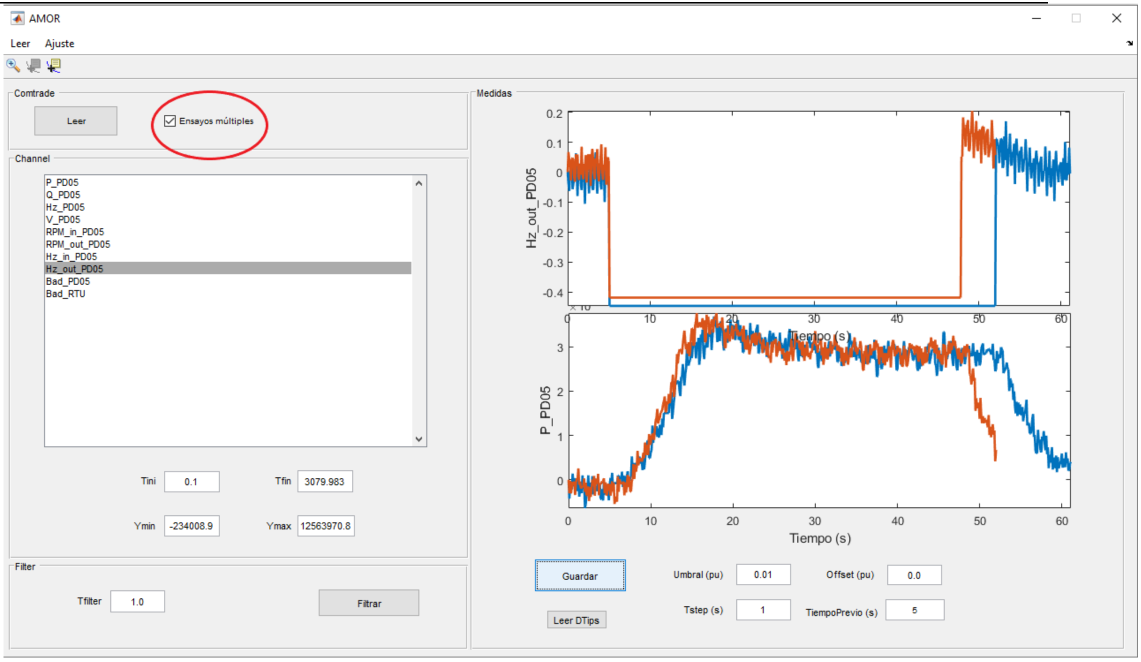
Tfilter 1.0

Umbral (pu) 0.01 Offset (pu) 0.0

Tstep (s) 1 TiempoPrevio (s) 5

Figure 21: Leer Dtips detection point example.

4. If the user wants to select two or more examples, the process is straightforward: simply check the "Ensayos múltiples" box, marked with a red circle in Figure 22, and repeat the steps described above. Once all the desired examples have been selected and saved, the program will display them in different colors.



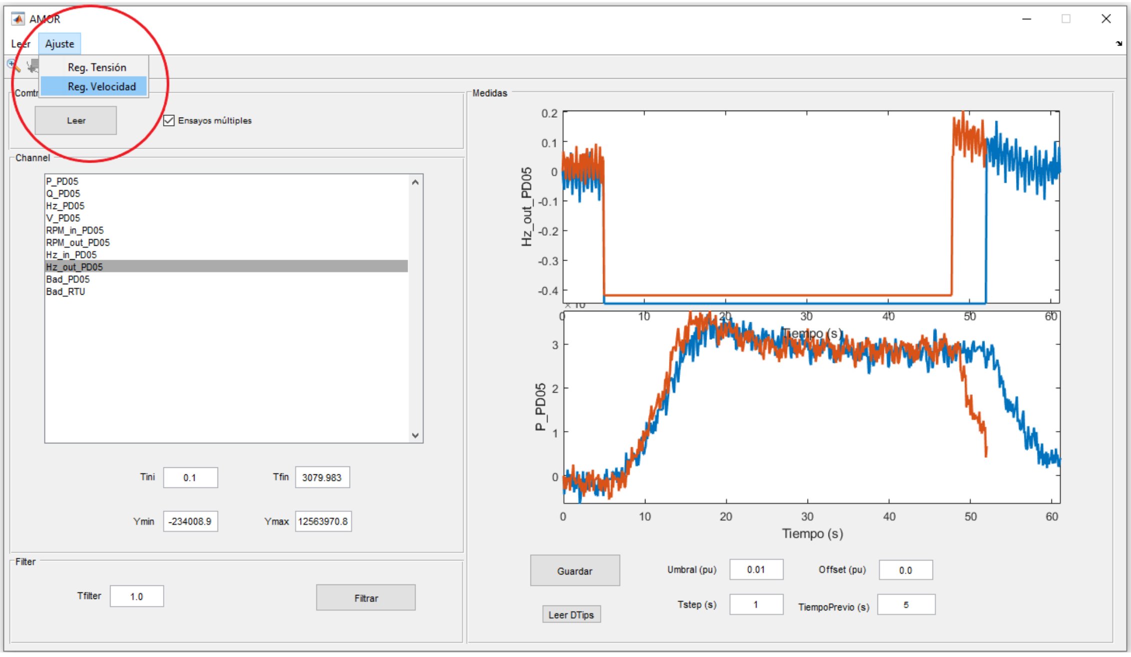
☑ Ensayos múltiples

P_PD05
Q_PD05
Hz_PD05
V_PD05
RPM_in_PD05
RPM_out_PD05
Hz_in_PD05
Hz_out_PD05
Bad_PD05
Bad_RTU

Tini 0.1 Tfin 3079.983
Ymin -234008.9 Ymax 12563970.8
Tfilter 1.0
Umbral (pu) 0.01 Offset (pu) 0.0
Tstep (s) 1 TiempoPrevio (s) 5

Figure 22: Multiple sample mode example.

After saving and configuring all the examples correctly, we can proceed with the model adjustment using the selected data. To do this, click on the “Ajuste” tab located at the top left of the interface, and then select “Reg. velocidad”, once again highlighted with a red circle in the Figure 23.



☑ Ensayos múltiples

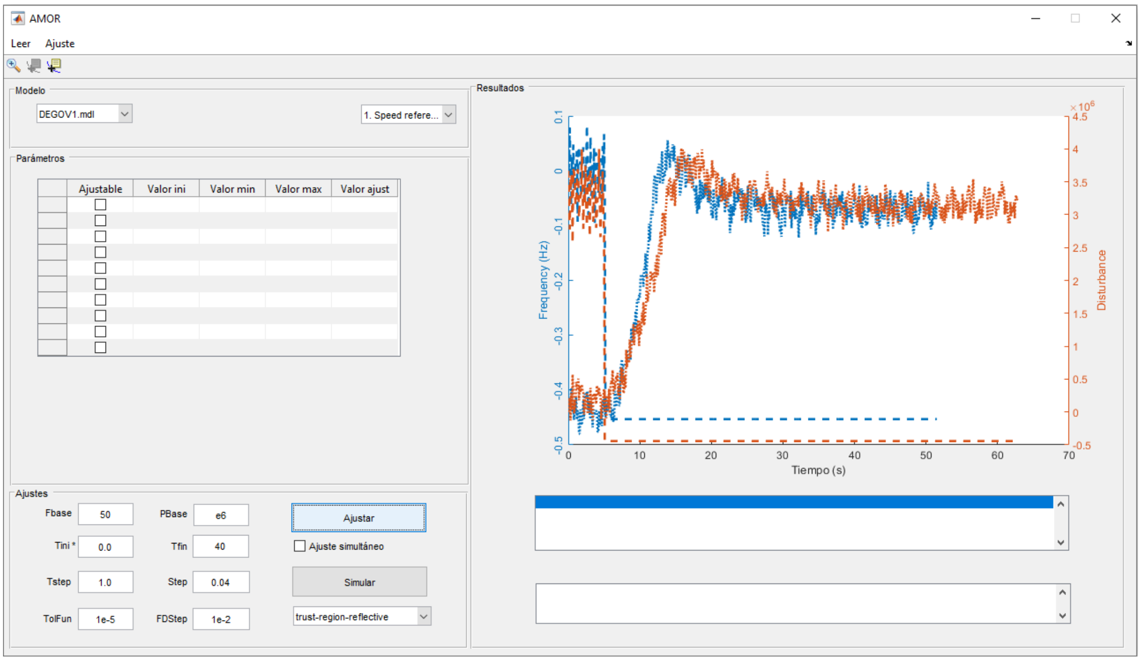
P_PD05
Q_PD05
Hz_PD05
V_PD05
RPM_in_PD05
RPM_out_PD05
Hz_in_PD05
Hz_out_PD05
Bad_PD05
Bad_RTU

Tini 0.1 Tfin 3079.983
Ymin -234008.9 Ymax 12563970.8
Tfilter 1.0
Umbral (pu) 0.01 Offset (pu) 0.0
Tstep (s) 1 TiempoPrevio (s) 5

Figure 23: Simulation and adjustment interface selection menu.

2.2 Adjustment and simulation Interface description

When all the previous steps have been completed, the adjustment and simulation interface, shown in Figure 24, will appear, displaying the tuning section of the program. This section allows us to configure all the necessary parameters for the proper adjustment of the different system models.



Modelo: DEGOV1.mdl
1. Speed refere...

	Ajustable	Valor ini	Valor min	Valor max	Valor ajust
	□				
	□				
	□				
	□				
	□				
	□				
	□				
	□				
	□				
	□				

Fbase 50
PBase e6
Tini * 0.0
Tfin 40
Tstep 1.0
Step 0.04
TolFun 1e-5
FDStep 1e-2
□ Ajuste simultáneo
trust-region-reflective

Figure 24: Simulation and adjustment interface.

A variety of buttons, menus, and input fields are displayed on the screen, all of which become easy to understand when explored one at a time.

2.3.1 Test types

In the top middle area of the screen there is a pop-up menu which will allow us to choose the type of example we want to use: Speed reference, Power and Frequency or Load rejection, as Figure 25 illustrates.

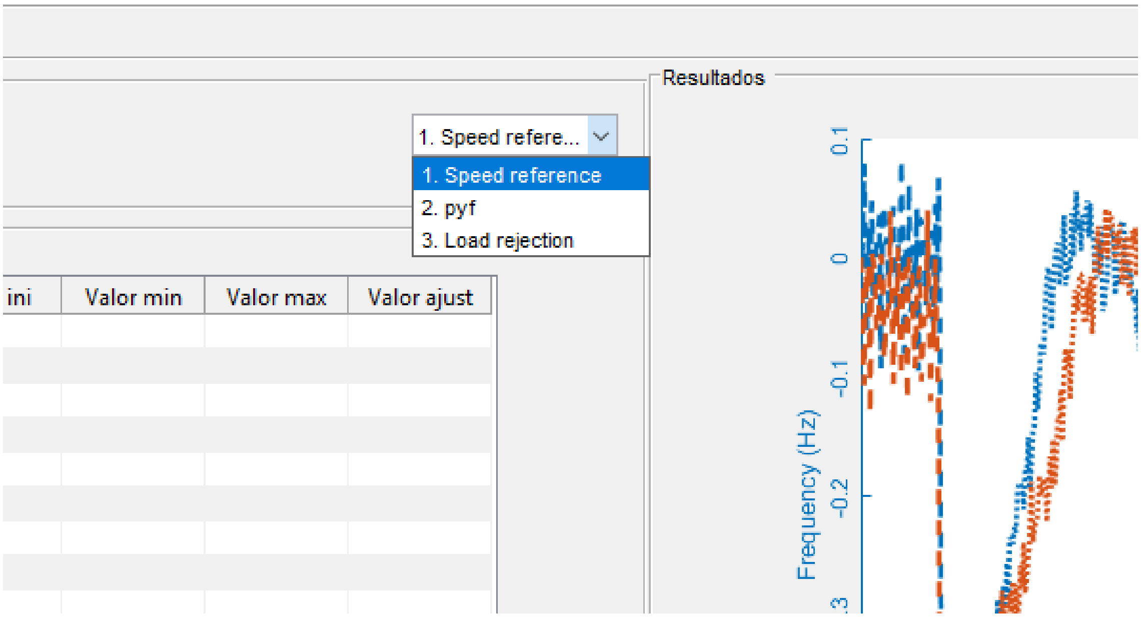


Figure 25: Simulation mode selection menu.

Although the pop-up menu displays three selectable options, only two types of tests are currently supported: load rejection and speed reference, as previously discussed. The purpose of the 'Power and Frequency' option lies in its relationship with the speed reference tests. Only, when working with this type of test, the user can either provide the magnitude of the step variation, resulting in a synthetic simulation, or input real measured data to generate a more accurate response. Both approaches produce satisfactory results; however, the 'Power and Frequency' mode minimizes the risk of human error and yields a more realistic simulation, as it incorporates actual system measurements.

2.3.2 Model selection

Looking at Figure 26 we can see a pop-up menu located in the top left area of the screen, which allows us to select the model that corresponds to our dataset. Each turbine-governor system has its own structural characteristics, physical limitations, and particularities, often influenced by factors such as the type of energy source or the manufacturer. This is why several Simulink models are available in the list: the more accurately the chosen model reflects the real system, the better we can predict its behavior in response to fault events.

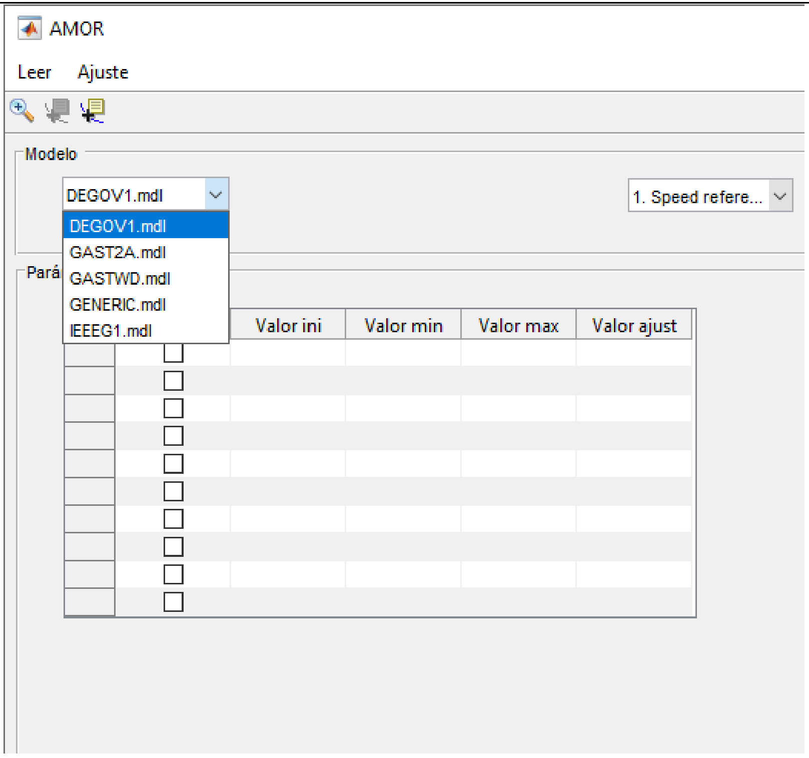
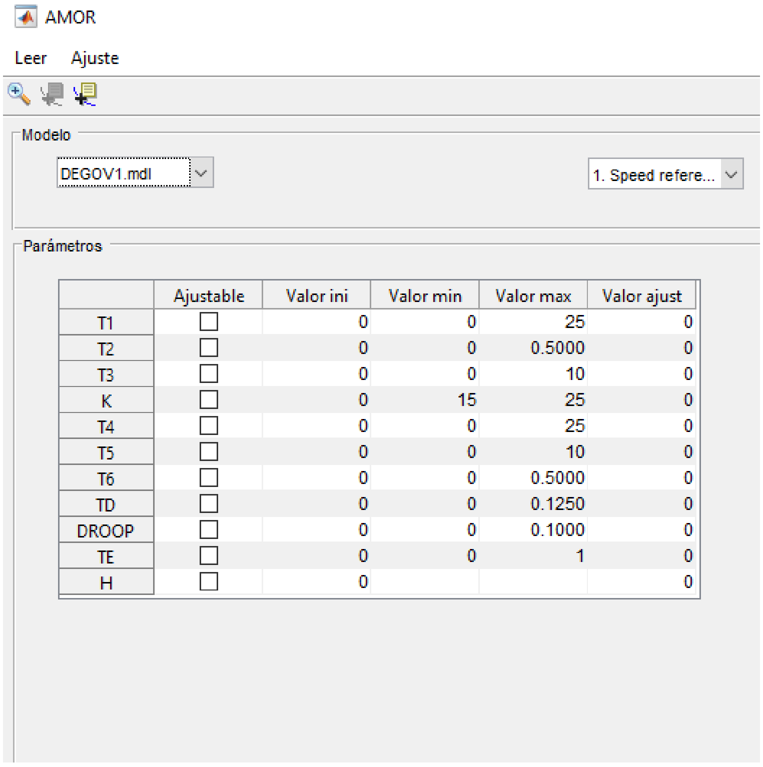


Figure 26: Model selection menu.

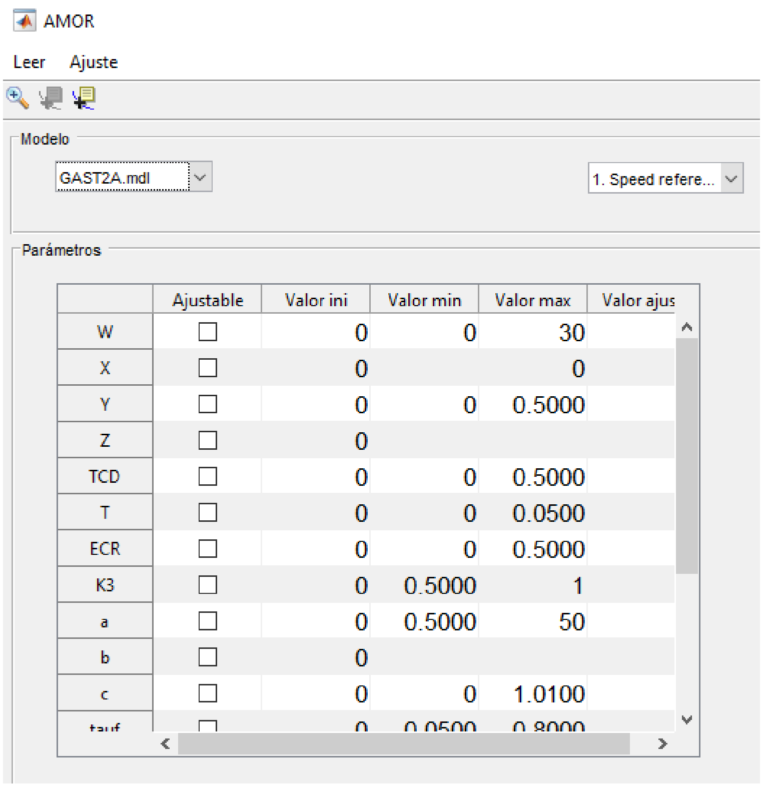
To select the model, we simply need to click on it. The program will then automatically adjust the size of the parameter table to fit the number of parameters associated with the selected model. For example, DEGOV1 model has, like displayed in Figure 27, fewer parameters than the GAST2A model, which is displayed in Figure 28.



Modelo: DEGOV1.mdl
1. Speed refere...

	Ajustable	Valor ini	Valor min	Valor max	Valor ajust
T1	□	0	0	25	0
T2	□	0	0	0.5000	0
T3	□	0	0	10	0
K	□	0	15	25	0
T4	□	0	0	25	0
T5	□	0	0	10	0
T6	□	0	0	0.5000	0
TD	□	0	0	0.1250	0
DROOP	□	0	0	0.1000	0
TE	□	0	0	1	0
H	□	0			0

Figure 27: DEGOV1 parameter list.



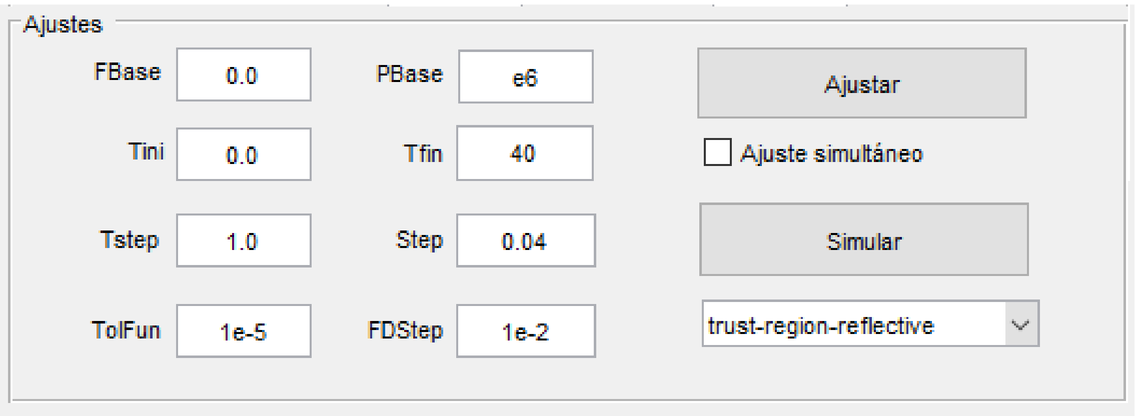
Modelo: GAST2A.mdl
1. Speed refere...

	Ajustable	Valor ini	Valor min	Valor max	Valor ajus
W	□	0	0	30	
X	□	0		0	
Y	□	0	0	0.5000	
Z	□	0			
TCD	□	0	0	0.5000	
T	□	0	0	0.0500	
ECR	□	0	0	0.5000	
K3	□	0	0.5000	1	
a	□	0	0.5000	50	
b	□	0			
c	□	0	0	1.0100	
tauf	□	0	0.0500	0.8000	

Figure 28: GAST2A parameter list.

2.3.3 Data boxes

In the bottom left corner of the interface, we find a set of input fields that are required to run any simulation or adjustment procedure, as shown in Figure 29.



Ajustes

FBase 0.0

PBase e6

Tini 0.0

Tfin 40

Tstep 1.0

Step 0.04

TolFun 1e-5

FDStep 1e-2

☐ Ajuste simultáneo

trust-region-reflective

Figure 29: Input data boxes.

Although these settings may appear unusual at first, they are very straightforward once their role is understood. In the following paragraphs, each parameter will be explained in detail, following the order shown in the interface:

1. “FBase” and “PBase” are used to define the frequency and power bases of the system. The frequency is almost always 50 Hz or 60 Hz, while the power base tends to vary more significantly, but is typically in the megavolt-ampere range. For this reason, the “PBase” field is initialized with “e6”, allowing the user to input values quicker.

2. “Tini” and “Tfin” are used to define the initial and final times for the simulation or adjustment. These values can be applied commonly across all samples or individually for each one. If the program detects a single value, it will apply it to every sample. However, if multiple values are entered and separated by commas, each will be assigned to its corresponding sample.

3. “Tstep” and “Step” are needed only for Load rejection and Speed variation modes only, not for Power and Frequency, as this mode takes the step information directly from the frequency data. “Tstep” data box must only be used when “LeerDTIps” hasn’t been activated to save the sample. If power and frequency mode is not being used, the program will simulate a step of the input size given in the “Step” box at the time “TSteep” counting from the “Tini” moment, which is usually zero.
   In both speed reference and load rejection tests, the step size must be specified individually for each sample, separated by commas. For speed

reference tests, the values are given in hertz (Hz), while for load rejection ones, the values are in megawatts (MW).

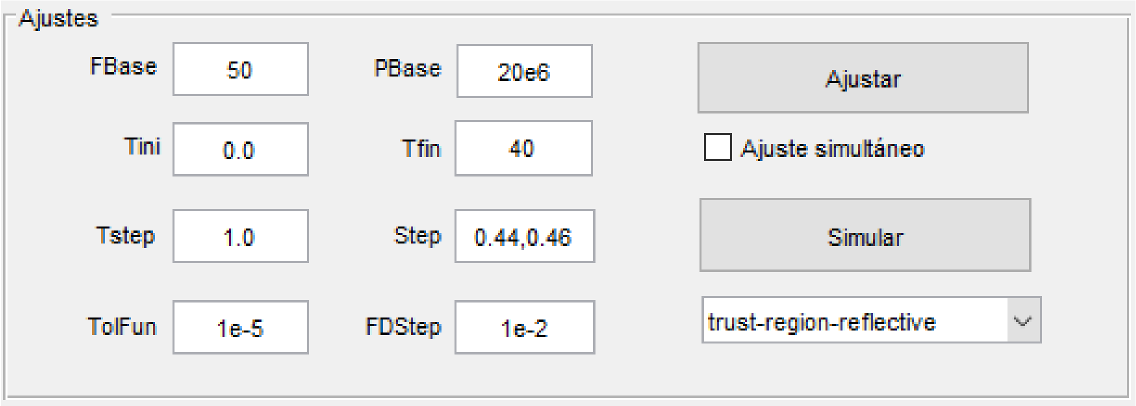
4. "TolFun" and "FDStep" are tuning parameters used to define the desired error tolerance for the adjustment. These values are typically not modified by the user, as the default settings are sufficient for most cases.

2.4 Simulation and adjustment of model parameters

To simulate the different responses of different turbine-governor systems, different models must be used, as previously mentioned. Depending on the building characteristics of the different systems, the models will use certain types of equations and thus will have a different number of parameters to adjust and simulate.

In our study we have focused on DEGOV1 and GAST2A turbine governor system models, whose diagrams were shown in Figure 9 and Figure 10. These models have been extracted from the manufacturers and known online block diagram sources for these types of systems.

Once familiar with the models, information boxes and menus of the program, we can proceed to the simulation of different tests. This process is very straight forward, firstly, the data is collected by the saving method mentioned earlier, then the correct type of test, model and values for the boxes are selected and introduced. For example, when working with two samples, obtained using the "Leer Dtips" method, the input fields must be filled as illustrated in Figure 30.



Ajustes

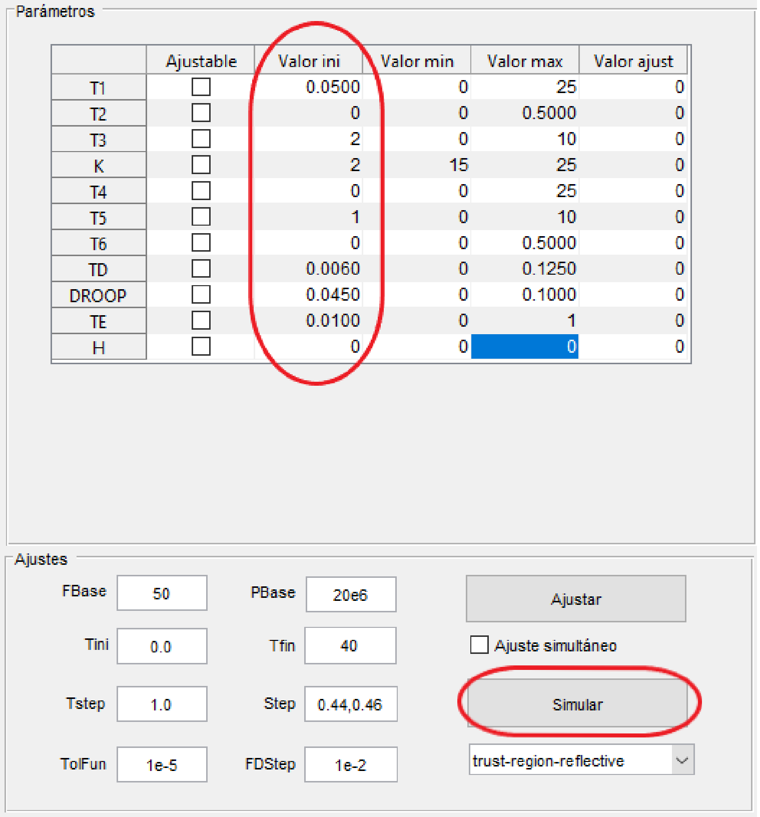
FBase	50	PBase	20e6
Tini	0.0	Tfin	40
Tstep	1.0	Step	0.44,0.46
TolFun	1e-5	FDStep	1e-2

□ Ajuste simultáneo

trust-region-reflective

Figure 30: Example of data needed for a double sample simulation.

And finally, we only need to write initial values for each of the parameters used in the given model, these values should be written in the "Valor ini" column of the model table (seen in Figure 31) and press "Simular button".



Parámetros

	Ajustable	Valor ini	Valor min	Valor max	Valor ajust
T1	□	0.0500	0	25	0
T2	□	0	0	0.5000	0
T3	□	2	0	10	0
K	□	2	15	25	0
T4	□	0	0	25	0
T5	□	1	0	10	0
T6	□	0	0	0.5000	0
TD	□	0.0060	0	0.1250	0
DROOP	□	0.0450	0	0.1000	0
TE	□	0.0100	0	1	0
H	□	0	0	0	0

Ajustes

FBase 50
PBase 20e6
Tini 0.0
Tfin 40
Tstep 1.0
Step 0.44,0.46
TolFun 1e-5
FDStep 1e-2
□ Ajuste simultáneo
trust-region-reflective

Figure 31: simulation final step.

2.4.1 Simulation

If all the required information is provided, the program will proceed to simulate the model's response to the different step sizes. Once the simulation is complete, the results will be displayed using different colors for each case.

To run the simulation, the program takes the input values provided through the GUI and feeds them into the Simulink model to generate the desired response. This process is straightforward thanks to the simplicity of the interface, while the data transfer is handled using the implemented Matlab and Simulink commands.

If this is the first simulation since the program was launched, the model may take some time to load, this is completely normal. Once the model is loaded, any changes made to the simulation parameters will be applied almost instantly, and the plot will update within a few seconds. If everything runs correctly, we can expect the following results for the different types of available simulations:

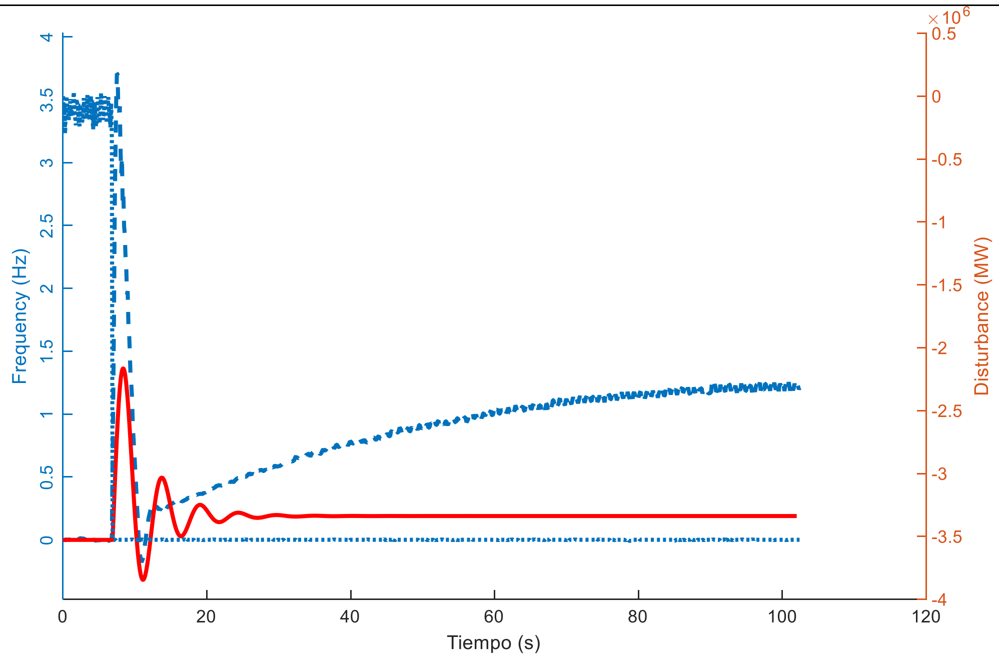


Figure 32: Simulation example for load rejection test.

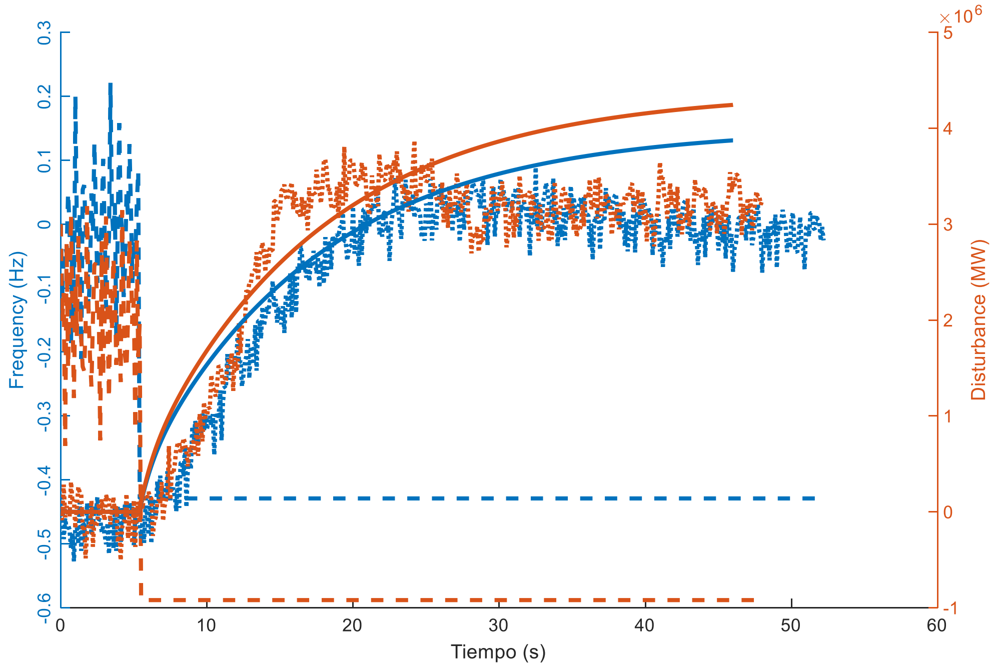


Figure 33: Simulation example for Speed Reference test with manual step size

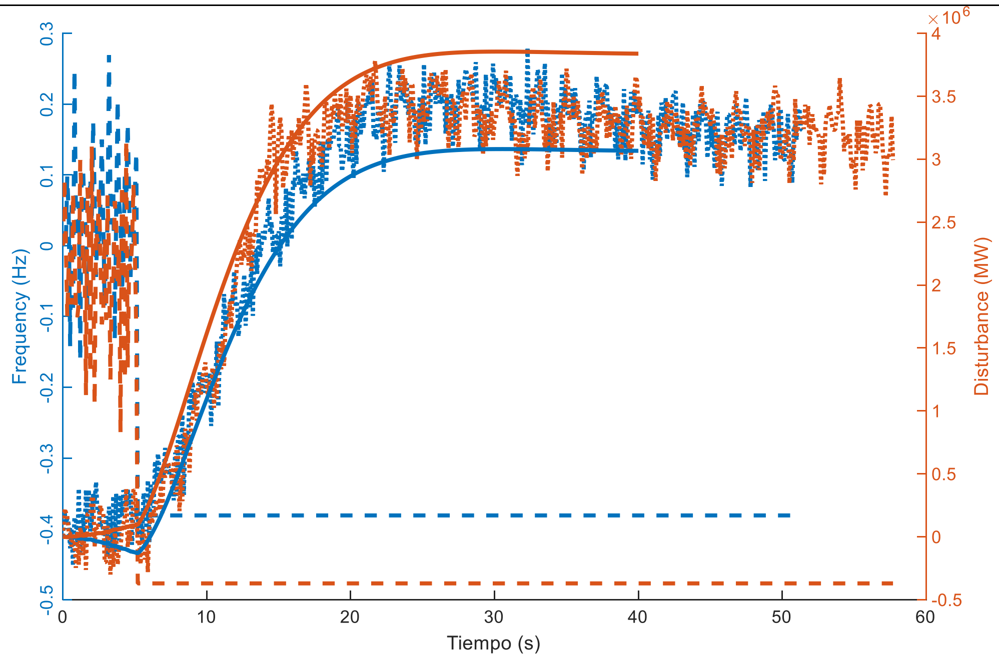


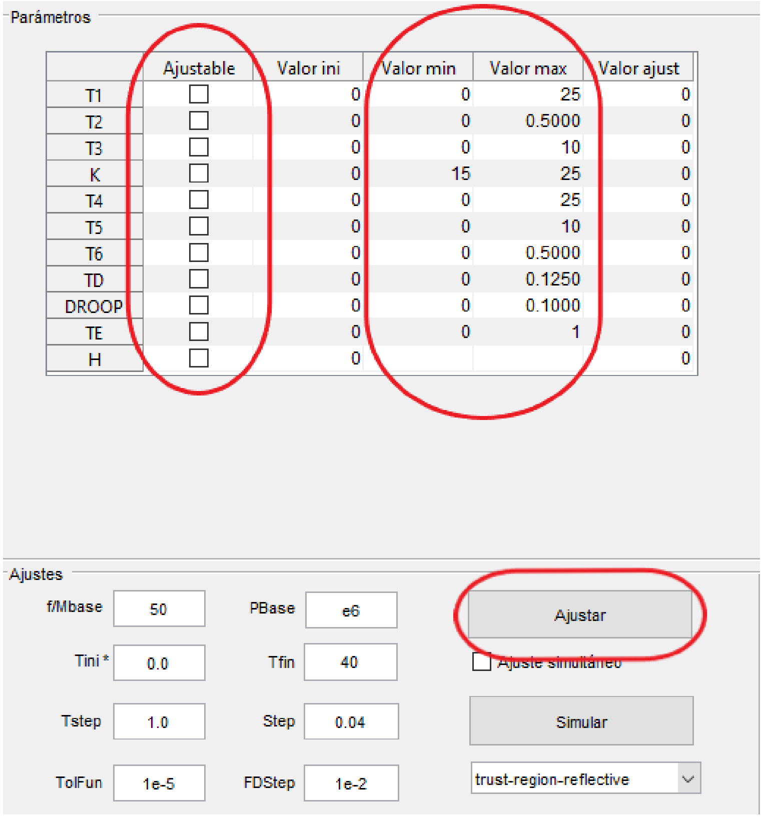
Figure 34: Simulation example for automatic Power and Frequency test

As illustrated in Figure 32, Figure 33 and Figure 34, all three simulation methods produce relatively good results when using the initial parameter configuration. However, the power and frequency-based simulation provides a slightly better fit to the initial part of the experimental data, suggesting a higher accuracy in capturing the system's actual behavior.

2.4.2 Adjustment procedure

The adjustment is performed thanks to the tuning of the given data to make it suitable for the least squares algorithm of MATLAB, this process is done by two different functions, "main tuning" and "tuning_model_Simulink", in charge of interpolating certain samples when necessary and making the sum of the output error for the measurements.

The adjustment procedure follows a similar methodology to the simulation. In addition to the information required for running the simulation, two extra columns must be completed to define the parameter boundaries for the adjustment. Not all parameters share the same range, while some may vary in value over twenty units, others may only vary between negative and positive one. Defining appropriate ranges for each parameter helps the algorithm converge more quickly and efficiently. Once the limits are set, the parameters to be adjusted are selected using the checkboxes in the first column of the table. After completing all previous steps, the adjustment process can be initiated by clicking the "Ajustar" button, situated on the lower part of the interface, as illustrated in Figure 35.



Parámetros

	Ajustable	Valor ini	Valor min	Valor max	Valor ajust
T1	□	0	0	25	0
T2	□	0	0	0.5000	0
T3	□	0	0	10	0
K	□	0	15	25	0
T4	□	0	0	25	0
T5	□	0	0	10	0
T6	□	0	0	0.5000	0
TD	□	0	0	0.1250	0
DROOP	□	0	0	0.1000	0
TE	□	0	0	1	0
H	□	0			0

Ajustes

f/Mbase: 50
PBase: e6
Tini *: 0.0
Tfin: 40
Tstep: 1.0
Step: 0.04
TolFun: 1e-5
FDStep: 1e-2
□ Ajuste simultáneo
trust-region-reflective

Figure 35: Adjustment final step.

The different samples, each with its own step size, produce distinct responses that are not perfectly proportional to one another. If they were, a single adjustment based on just one sample would be sufficient for all cases. This is precisely why the best results are obtained by using multiple samples simultaneously. Although the outcome may appear less accurate when compared to the result of a single sample adjustment, the overall error across all samples is minimized when the adjustment is done globally, as it reduces overfitting.

To perform a simultaneous adjustment, we must click the “ajuste simultaneo” checkbox between “ajustar” and “similar” buttons, otherwise, the results will be just the mean value of the individual adjustments and not the single best for all of them as a group.

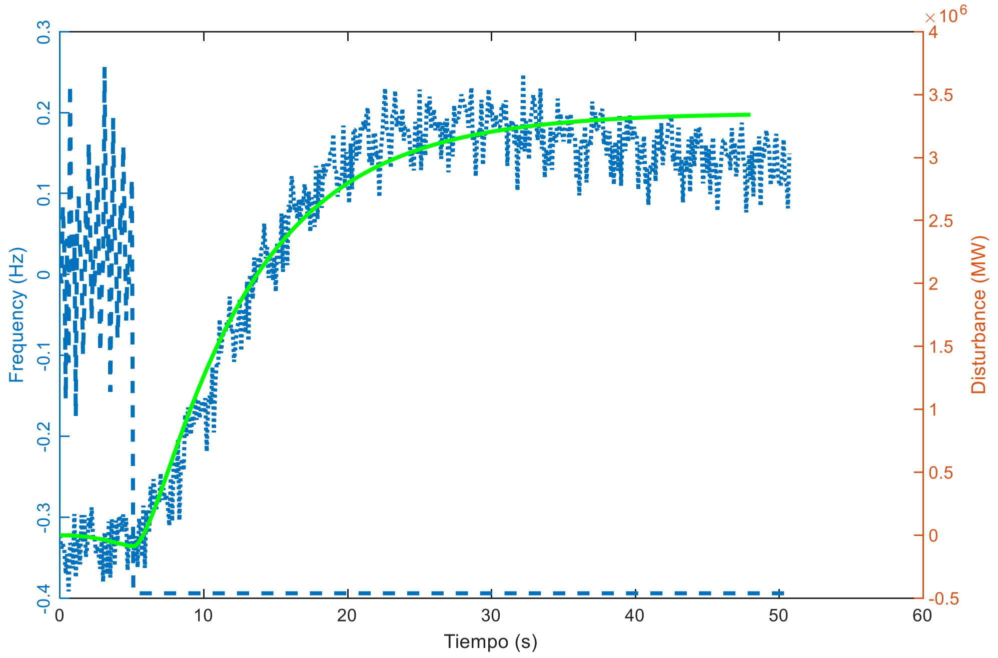


Figure 36: Example of single test adjustment.

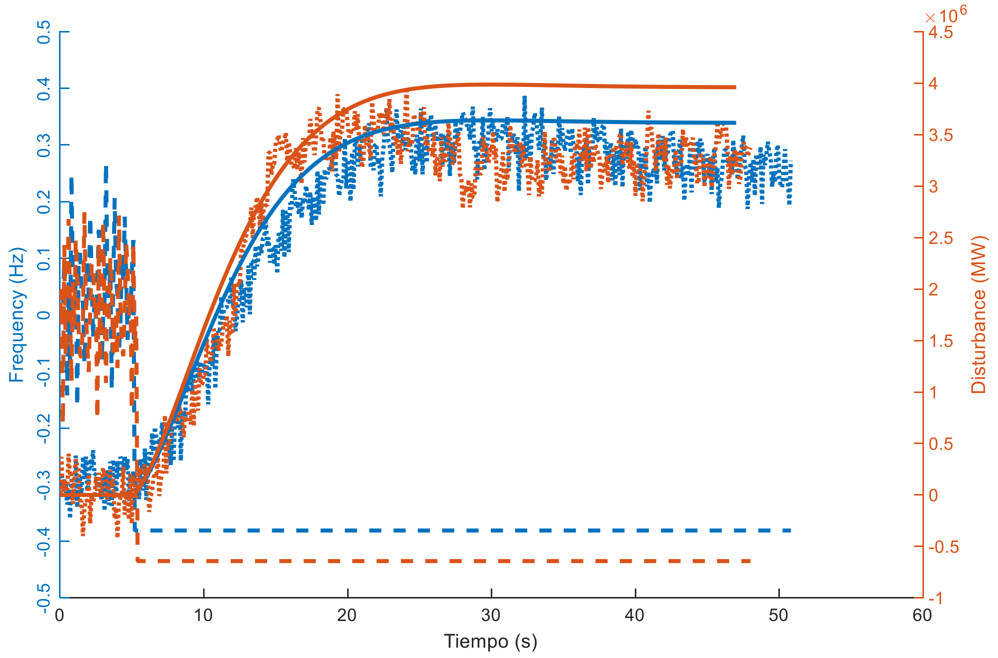


Figure 37: Example of adjustment with mean values.

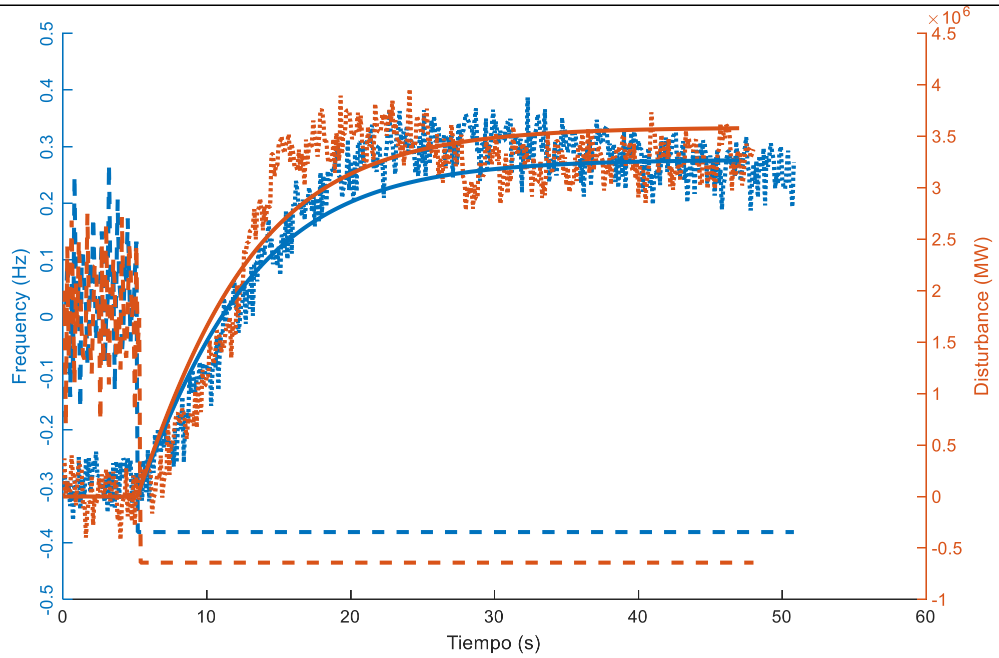
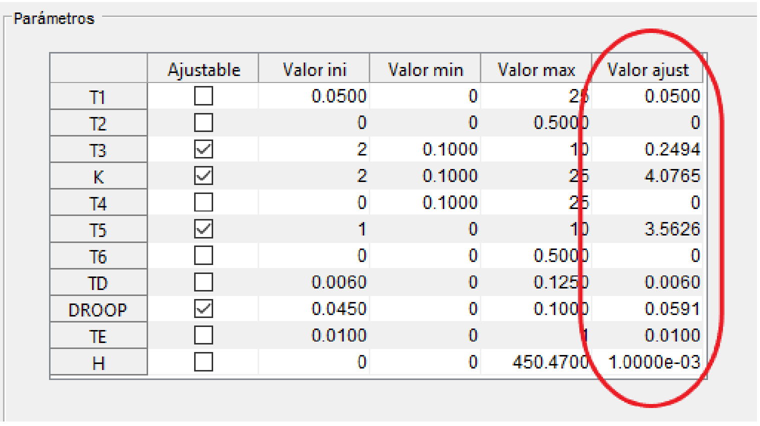


Figure 38: Example of simultaneous adjustment.

If we analyze Figure 36, Figure 37 and Figure 38, we can observe that the single sample adjustment yields highly accurate results. However, this level of accuracy is reached precisely because the parameters are tuned specifically for that individual sample, rather than for the general behavior of the turbine-governor system. This limitation is partially addressed by the mean value adjustment, which considers multiple samples. However, because of its simplistic averaging approach, it is still worse than the simultaneous adjustment method, which proves to be a more precise and reliable tuning strategy.

When a successful adjustment is performed, a new column, highlighted in Figure 39, will be filled with the values obtained by the algorithm so that they can be easily checked and noted.


Parámetros

	Ajustable	Valor ini	Valor min	Valor max	Valor ajust
T1	□	0.0500	0	25	0.0500
T2	□	0	0	0.5000	0
T3	☑	2	0.1000	10	0.2494
K	☑	2	0.1000	25	4.0765
T4	□	0	0.1000	25	0
T5	☑	1	0	10	3.5626
T6	□	0	0	0.5000	0
TD	□	0.0060	0	0.1250	0.0060
DROOP	☑	0.0450	0	0.1000	0.0591
TE	□	0.0100	0	1	0.0100
H	□	0	0	450.4700	1.0000e-03

Figure 39: Adjusted values column filled with results.

2.4.3 Simple Adjustment

Until now, we have focused on how users interact with the tool to obtain results from their models and tests. Before presenting some tunned parameters and outcomes for a generic set of turbine-governor systems, we will now shift perspective and explain how the tool internally processes the data to produce those results.

In both the single and mean adjustment modes, the underlying method is the same. First, the simulation inputs are retrieved from the GUI menus and information boxes and stored in several vector fields. This data is then passed to a separate script called "main_tuning". This script is responsible for preparing the measurements for tuning: it trims the time series according to the user's selection and normalizes the step size using the appropriate power or frequency base.

Once all measurements have been processed, each sample, along with the selected adjustable parameters, is individually sent to the LMS algorithm. The best fitting parameters for each individual sample are then computed. When multiple samples are selected at once, in the mean adjustment mode, the program simply calculates the average of all individually obtained results. However, this approach is not ideal and may lead to suboptimal tuning.

2.4.3 Simultaneous Adjustment

To overcome the limitations inherent to the mean adjustment mode, a more advanced strategy known as simultaneous adjustment was developed. While it involves a higher level of complexity, this method significantly increases the accuracy of the parameter fitting process. The motivation behind this approach is that users are generally not interested in finding the best fitting parameters for a single, isolated sample. Instead, the objective is to obtain a model that can accurately replicate the system's behavior across a wide range of test conditions.

To achieve this, the internal functioning of the tool differs from the simpler methods. After collecting all the relevant input data from the GUI, the program uses the duration of the first selected sample as a reference. It then interpolates the rest of the signals so that all vectors are aligned and share the same length. This standardization step is essential, as it allows the algorithm to process all the information in a consistent format.

Once the data has been normalized, the program writes it in a single structure that includes the information from all selected samples. This combined dataset is then passed to the LMS algorithm, which searches for a unique set of parameters that minimizes the total fitting error across all samples simultaneously. By treating the adjustment as a global optimization problem rather than a collection of independent ones, in this way, the program can obtain more robust and generalizable results.

The simultaneous adjustment offers several advantages over simpler approaches. Firstly, it **improves generalization**: the resulting parameters are not tailored to a specific scenario but instead reflect the system's average behavior, increasing their reliability under different conditions. Secondly, it **helps prevent overfitting**, as it avoids adapting the model too closely to any individual dataset, which may contain noise or non-representative features. Lastly, it allows for **better use of available data**, especially in cases where the number of real measurements is limited, by extracting the maximum amount of information from all available samples at once.

3. Results obtained

Now that the program's interface and methodology has been thoroughly defined and explained, we can proceed to present the results obtained by tuning both the DEGOV1 and GAST2A models. These results are based on the two types of experiments performed, load rejection and speed variation, and include, when applicable, the three available tuning methods: single adjustment, multiple adjustment with mean value, and simultaneous adjustment.

3.1 Adjustment of DEGOV1 turbine and speed regulator model

These groups will represent most of our test cases, as we have four different configurations with distinct base power levels and operational characteristics. This variety allows us to better illustrate and compare the impact of the different adjustment methods. The main parameters selected for tuning are **K, T3, T4, T5,** and the **droop**, as they provide the most significant improvement relative to computational effort and have a greater influence on the model's accuracy compared to other parameters.

3.2 DEGOV1 results comparison

3.2.1 PD01

In this example a 9.4 MVA system undergoes a load rejection test in which 3.47 MW are suddenly disconnected, this results in a final variation of 1.27 Hz and the transitory response seen in blue.

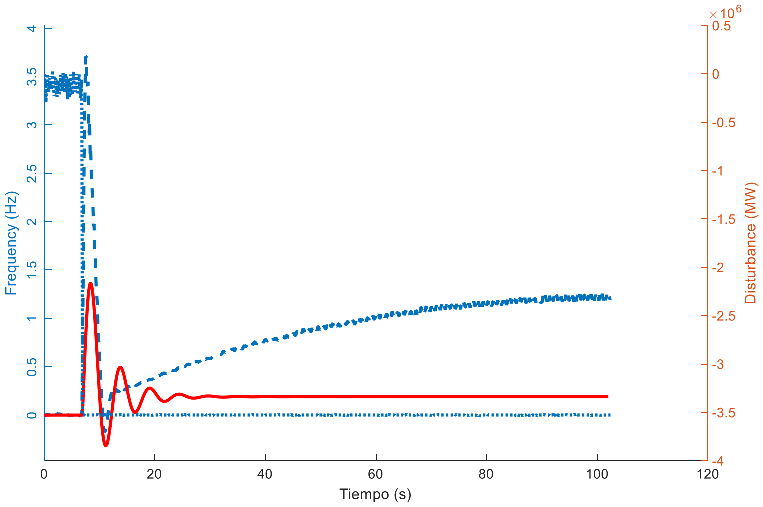


Figure 40: PD01 initial parameters simulation

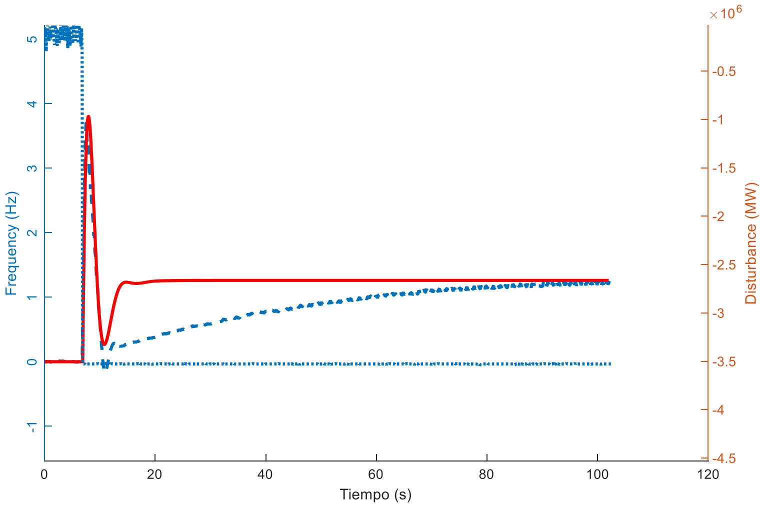


Figure 41: PD01 adjusted parameters simulation

3.2.2 PD02

In this example another 9.4 MVA system undergoes a load rejection test in which 2.941 MW are suddenly disconnected, this results in a final variation of 0.78 Hz and the transitory response seen in blue. Although the simulation with initial values may seem relatively precise, the adjusted values improve the fitment for the first 15 seconds, which is the main objective of our adjustment.

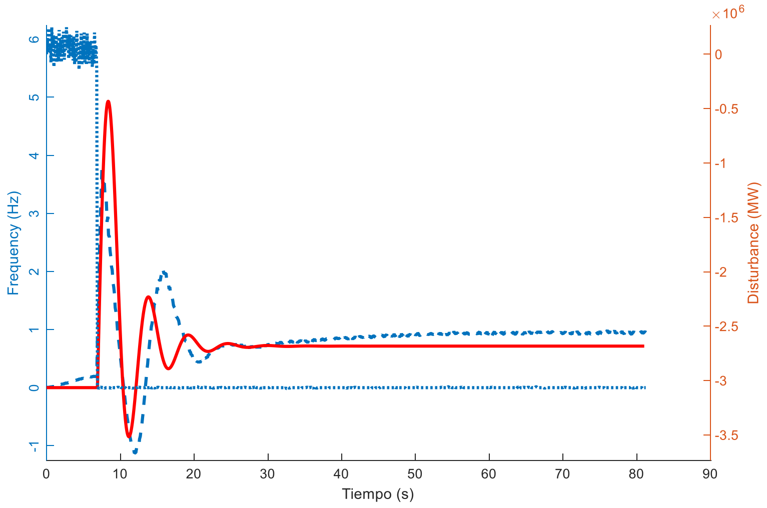


Figure 42: PD02 initial parameters simulation

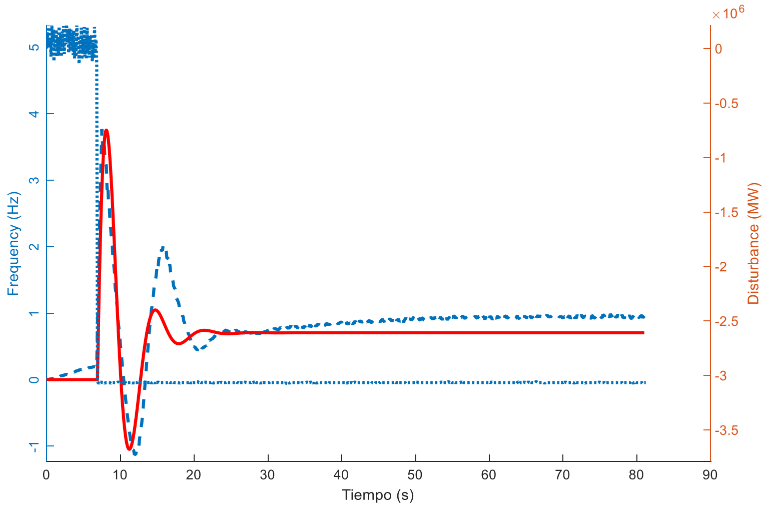


Figure 43: PD02 adjusted parameters simulation

3.2.3 PD04 with simultaneously adjusted parameters

In this case we study a 20 MVA turbine-governor system that has been tested using a speed variation method, yielding these characteristic curves that we will see often in the following cases. Once again default parameters achieve a good approximation, as seen in Figure 44, but it is not until the parameters are adjusted that we get a more precise simulation. Although the mean value method of adjustment in Figure 45 is a good fit, the Simultaneous adjustment somewhat more precise, as shown in Figure 46.

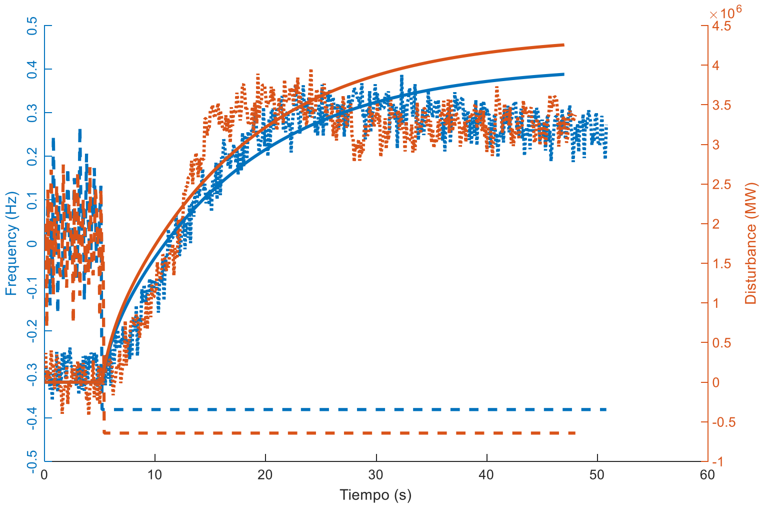


Figure 44: PD04 default parameter's simulation

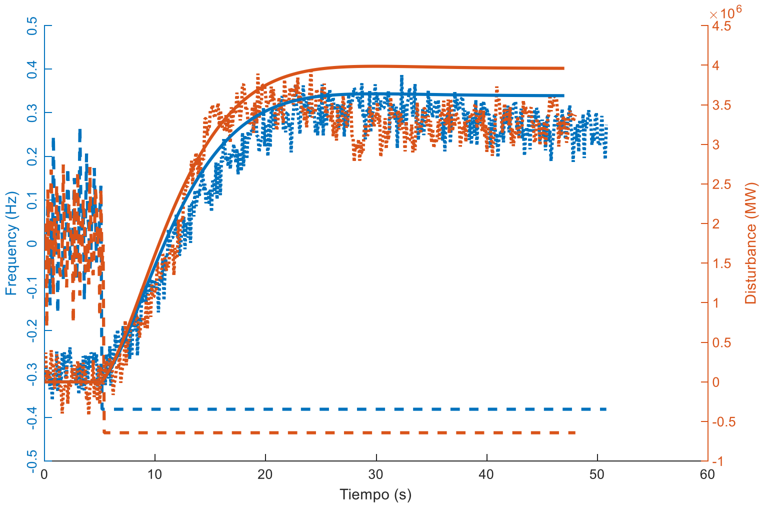


Figure 45: PD04 means adjusted parameters simulation

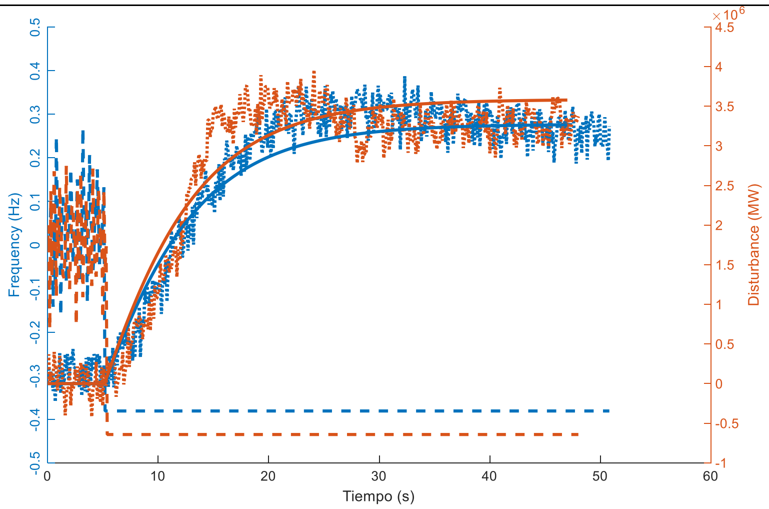


Figure 46: PD04 simultaneously adjusted parameters simulation

3.2.4 PD06

This case addresses a 30 MVA turbine-governor system that has been tested using a speed variation method, with a very similar result as the PD04 case. Here the default parameters fail to adjust to the initial slope of the power as seen in Figure 47, and both the mean and simultaneous adjustment methods are quite similar, with the simultaneous in Figure 49 being a bit superior than the mean in Figure 48.

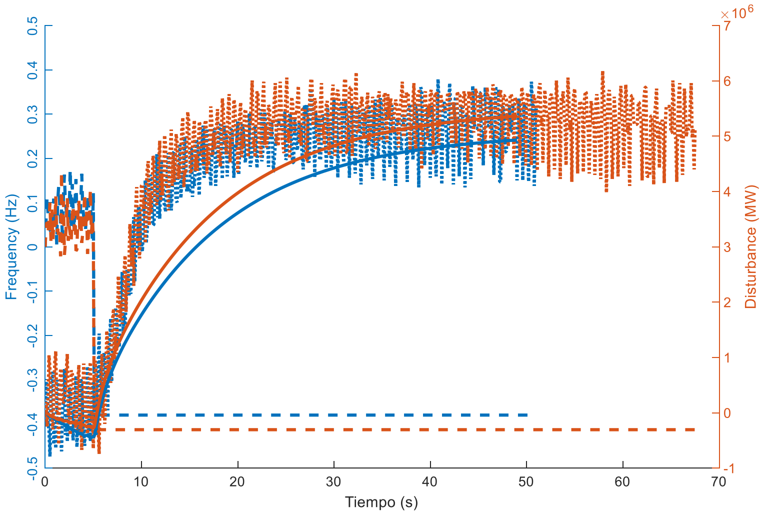


Figure 47: PD06 default parameters simulation

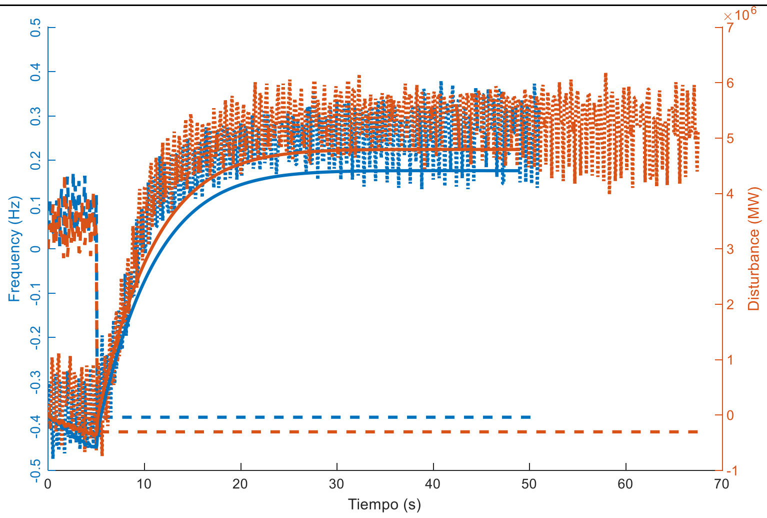


Figure 48: PD04 means adjusted parameters simulation

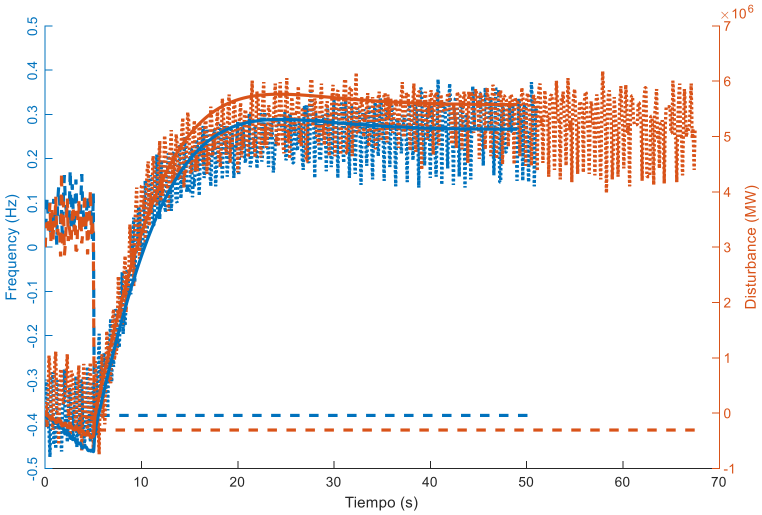


Figure 49: PD04 simultaneously adjusted parameters simulation

3.2.5 PD10

The following case is very similar to PD06, even if it does not have the same nominal power, being this one 22.5 MVA. As we can see the default parameters simulation in Figure 50 are a very poor approximation, and the simultaneous adjustment in Figure 52 is once again more precise than the means adjustment in Figure 51.

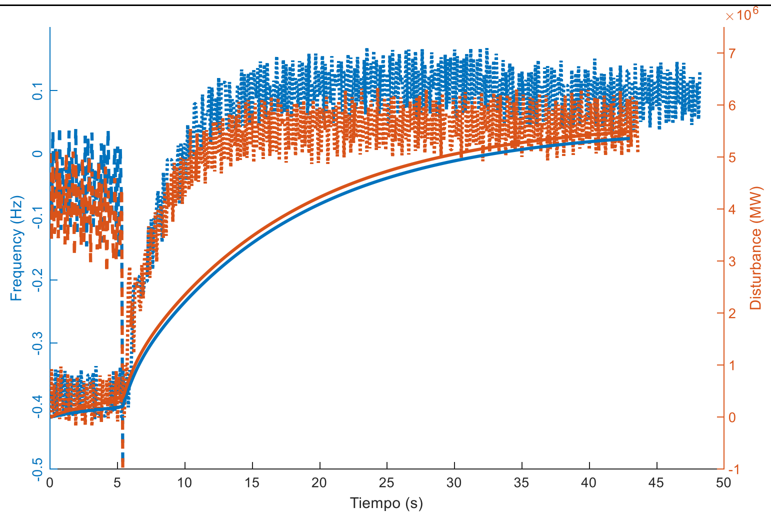


Figure 50: PD10 default parameters simulation

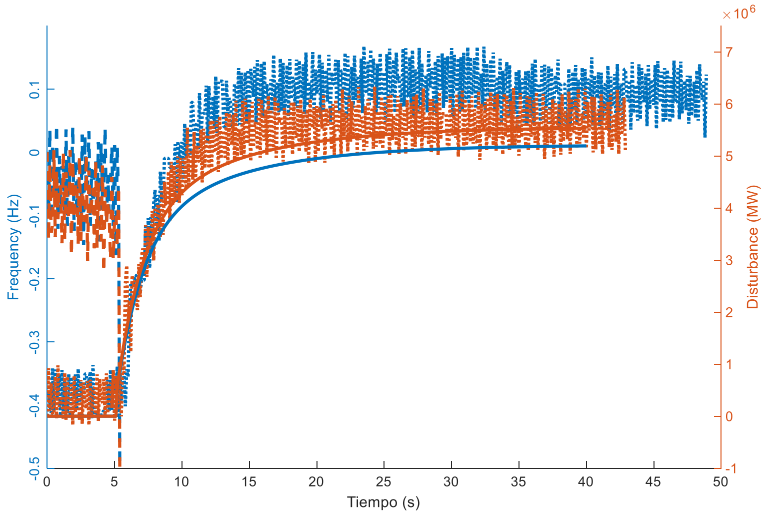


Figure 51: PD10 means adjusted parameters simulation

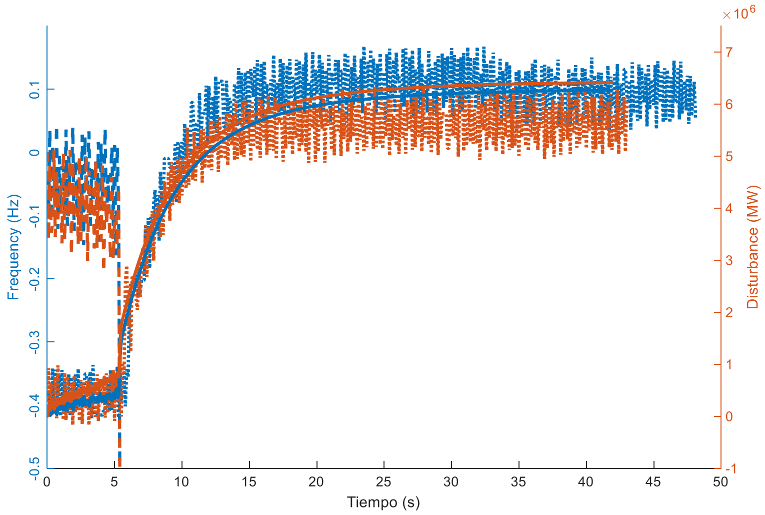


Figure 52: PD10 simultaneously adjusted parameters simulation

3.3 Adjustment of GAST2A turbine and speed regulator model

In the case of the GAST2A turbine-governor system, only two samples of speed variation tests have been adjusted. Although it may not seem as a significant amount of data, it still proves the versatility and adaptation capacities of the program. It proves the program only needs a block diagram to adjust a whole system. The main parameters selected for the GAST2A turbine-governor system tuning are **W, X, Y** and the **TCD**, as they provide the most significant improvement relative to the computational effort and have a greater influence on the model's accuracy compared to other parameters.

3.4 GAST2A results comparison

3.4.1 PG01

In this particular case, the nominal power is 25.94 MVA, and as we can appreciate, the data given represents a speed variation type of test in which the default parameters do a good job as seen in Figure 53, and due to the oscillating behavior observed after the initial transient response, the adjustment is not really capable of improving the simulation in a significant way as seen in Figure 54.

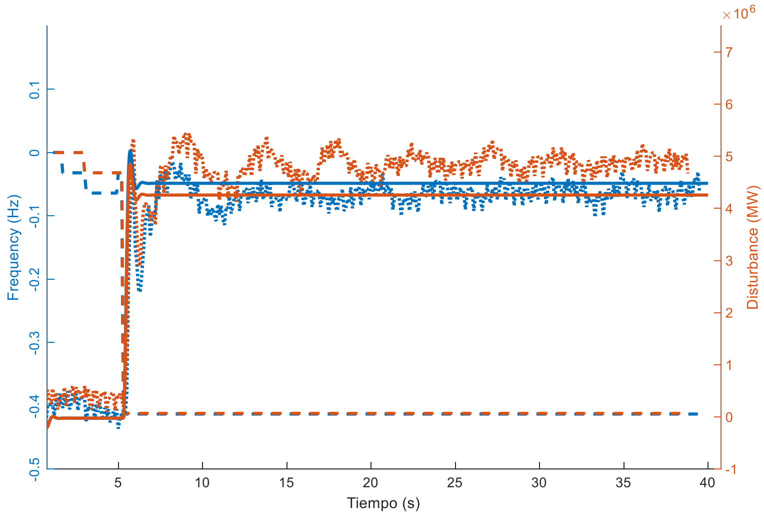


Figure 53: PG01 default parameters simulation

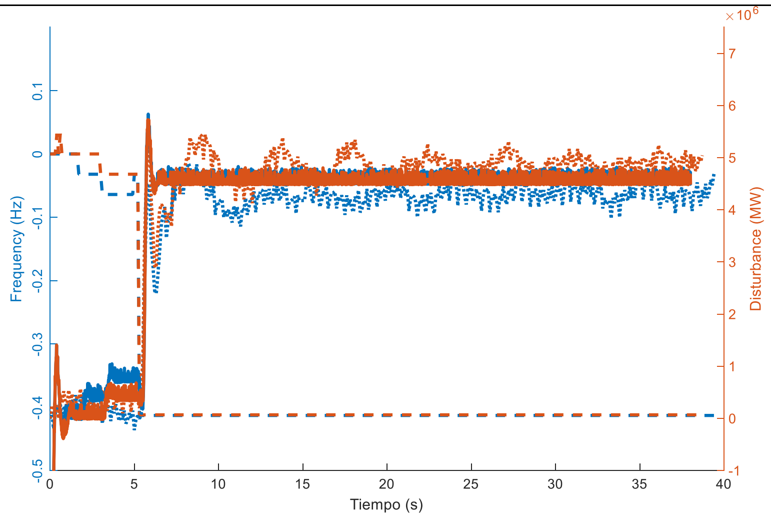


Figure 54: PG01 simultaneously adjusted parameters simulation

3.4.2 PG02

As with the previous case, the nominal power is 25.94 MVA, and the data given represents a speed variation type of test in which the default parameters do not work properly as seen in Figure 55, in contrast, the adjusted parameters, mimic almost perfectly the real behavior of the system as seen in Figure 56.

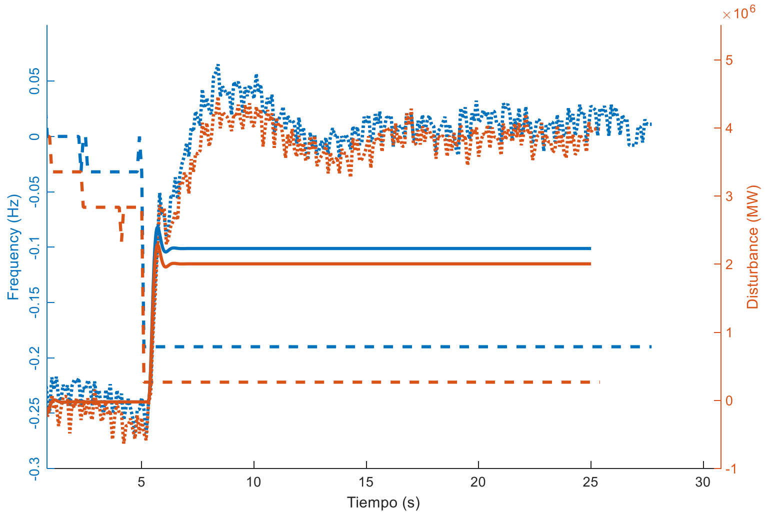


Figure 55: PG02 default parameters simulation

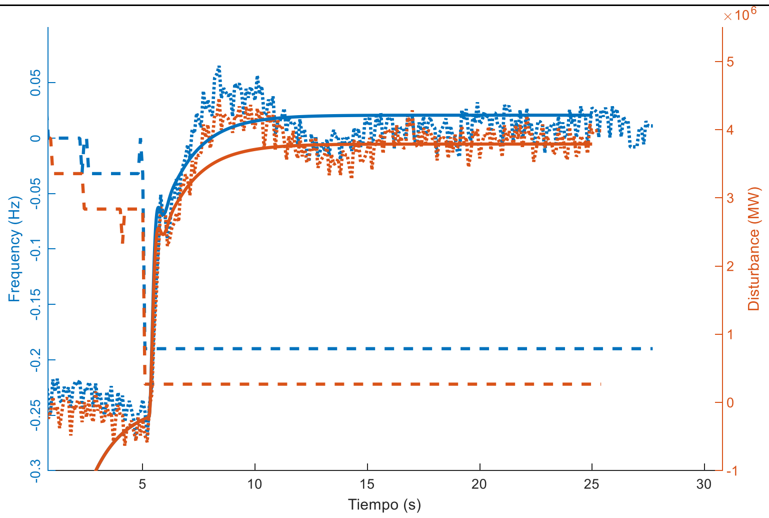


Figure 56: PG02 simultaneously adjusted parameters simulation

4. Conclusion

4.1 Main findings

The regulator model adjustment tool or AMOR allows automatic adjustment of speed regulator models, for accurate system predictions. The tool, based in the Matlab and Simulink relationship, performs a least mean squares adjustment by minimizing the difference between the models simulated output and the real measured response of a fault event.

For this purpose, various interfaces and functionalities have been added to the GUI interface of Matlab. Being the predominant, a COMTRADE reading module, with functionalities like automatic step detection; and a model adjustment module, that allows the adjustment of the parameters found in the block diagram models. These models are built by translating the real turbine-governor systems into Simulink's block diagrams.

This functionality and methodology allow the user to adjust any desired turbine-governor system, requiring only its block diagram and at least one load rejection or speed variation test. As a result, the tool proves to be highly flexible and adaptable to a wide range of systems, even in scenarios where available data is limited.

As seen in the comparison of the load rejection test like the one performed in Figure 43, AMOR greatly increases the precision of the simulation by adjusting the parameters of the DEGOV1 model implemented in the model library. This adjustment allows for a better understanding of the transient part of the incident and can positively influence fault prediction in the future.

When it comes to speed variation tests, the program is also really precise, reaching parameters that can fit most samples in a very realistic way, like demonstrated in Figure 36. However, AMOR's full potential can only be extracted from speed variation tests, as they enable the program to perform a global adjustment of the turbine-governor system. By using several samples of a test at once, the algorithm is able to adjust for the total error of all the simulations at the same time rather than finding the best fitting value for each individual sample and averaging them, like the mean method does in Figure 45 or Figure 48. With this simultaneous adjustment, the simulations were more precise in all cases. It improves generalization, helps the overfitting problem and improves the use of available data by extracting the maximum out of the samples.

This adjusting method has proven more effective for systems with both poor and rich datasets, by yielding parameters that adjust better not only to the samples used for the adjustment but for the rest of samples from a certain system.

Considering the results and functionalities discussed, AMOR proves to be a robust, versatile, and highly effective tool for the modeling and parameter adjustment of turbine-governor systems. Its ability to deliver accurate model calibration with minimal external input, requiring only the system's block diagram and a small number of real-world test samples, makes it especially useful for real-case applications in which the need for modeling is extreme or the data availability limited. Its adjusting methodology, which includes both individual and simultaneous adjustment strategies, ensures high accuracy, better generalization, and improved performance

Beyond its applicability to existing systems, AMOR's flexible design enables its use in the early stages of system development, providing meaningful insights and reliable parameter estimation. This easily adaptable capability makes the tool not only worth as a model optimizer but also as a strategic program for planning and future deployment of generation systems. Overall, AMOR contributes significantly to advancing the current work in dynamic modeling and simulation, greatly improving the traditionally slow process of parameter tuning, and reducing the possibility of human error thanks to its thoughtful framework.

4.2 Limitations and future work

Even if our results are quite satisfactory, there are still some things that can be improved. The main issue found when working with several samples simultaneously was the large amount of time it takes to adjust when selecting several parameters. While this could be a problem given by the complexity of the adjustment procedure, it is also possible that our adjusting ranges are far too wide, which would waste computational resources in vain. To solve this, a thorough study of each parameter would be needed, to minimize their possible ranges and thus maximize efficiency.

Looking into the future, our goal should be to migrate the current program, built in the GUI Matlab interface, to the new app designer, which would take the programs interface and performance a step further, not only making in more appealing but also making sure AMOR's remains as useful as it is now.

5. References

[1] M. Ažubalis, V. Ažubalis∗ , A. Jonaitis, R. Ponelis “identification of model parameters of steam turbine and governor “oil Shale, 2009, Vol. 26, No. 3 Special, pp. 254–268.

[2] Zazo, J.L. Zamora, L. Rouco y F.L. Pagola, “Identification of Excitation Systems from Time Response Tests”. International Conference Control'94 , 21 - 24 Marzo 1994, University of Warwick, Coventry (Reino Unido). IEE Conference Publication No. 389, pp. 839-844.

[3] C. -C. Tsai, W. -J. Lee, E. Nashawati, C. -C. Wu and H. -W. Lan, "PMU based generator parameter identification to improve the system planning and operation," *2012 IEEE Power and Energy Society General Meeting*, San Diego, CA, USA, 2012, pp. 1-8.

[4] Kittiwattanaphon, Sapanan & Wangdee, Wijarn & Katithummarugs, Siriporn. (2019). Generator Excitation System Parameter Identification and Tuning by Using PSO. 1-4.

[5] Tetiana Bogodorova, “Modeling, Model Validation and Uncertainty Identification for Power System Analys”, Doctoral Thesis Stockholm, Sweden 2017.

[6] S. Kittiwattanaphon, W. Wangdee and S. Katithummarugs, "Generator Excitation System Parameter Identification and Tuning by Using PSO," 2019 7th International Electrical Engineering Congress (iEECON), Hua Hin, Thailand, 2019, pp. 1-4, doi: 10.1109/iEECON45304.2019.8939048.

[7] Mugarra, Asier, José M. Guerrero, Kumar Mahtani, and Carlos A. Platero. 2023. "Synchronous Generator Stability Characterization for Gas Power Plants Using Load Rejection Tests" *Applied Sciences* 13, no. 20: 11168.

[8] L. Rouco, J.L. Zamora, A. Zazo, M.A. Sanz, F.L. Pagola, “A Comprehensive Tool for Identification of Excitation and Speed-governing Systems for Power System Stability Studies”. Proceedings of the 13th Power Systems Computation Conference, Trondheim, Noruega, 28 Junio - 2 Julio, 1999, pp. 966-970.

[9] Azubalis, Mindaugas & Azubalis, Vaclovas & Jonaitis, Audrius & PONELIS, R. (2009). IDENTIFICATION OF MODEL PARAMETERS OF STEAM TURBINE AND GOVERNOR. Oil Shale. 26. 10.3176/oil.2009.3S.08.

[10] M. Sajjadi and H. Seifi, "Governor Parameter Estimation Considering Upper/Lower Production Limits," *2019 IEEE Milan PowerTech*, Milan, Italy, 2019, pp. 1-6, doi: 10.1109/PTC.2019.8810572.

[11] Zografos, D., Matevosyan, J., Eriksson, R., Baldick, R., & Ghandhari, M. (2020). *Frequency Response Assessment: Parameter Identification of Simplified Governor Response Models Using Historic Event Data*. *Cigre Science & Engineering*, **17**, 150-167.