DEGREE PROJECT IN ELECTRICAL ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2017

Semantic and Physical Modeling and Simulation of Multi-Domain Energy Systems: Gas Turbines and Electrical Power Networks

MIGUEL AGUILERA CHAVES

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING

SEMANTIC AND PHYSICAL MODELING AND SIMULATION OF MULTI-DOMAIN ENERGY SYSTEMS: GAS TURBINES AND ELECTRICAL POWER NETWORKS

by

Miguel Aguilera Chaves

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science in Electric Power Engineering

KTH Royal Institute of Technology

July 2017

Supervisor: Luigi Vanfretti

Examiner: Luigi Vanfretti

Electric Power and Energy Systems Department,
KTH Royal Institute of Technology,
Sweden

ABSTRACT

The ITEA3 OpenCPS (Open Cyber-Physical System Model-Driven Certified Development) project focuses on interoperability between the Modelica/Unified Modeling Language (UML)/Functional Mock-up Interface (FMI) standards, improved (co-)simulation execution speed, and verified code generation. The project aims to develop a modeling and simulation framework for cyber-physical and multi-domain systems. One of the main use cases for the framework, is the multi-domain equation-based modeling and simulation of detailed gas turbine power plants (including the explicit equation-based modeling of turbomachinery dynamics) and the electrical power grid.

In this work, UML class diagrams based on the Common Information Model (CIM) standard are used to describe the semantics of the electrical power grid. An extension based on the standard ISO 15926 has been proposed to derive the multi-domain semantics required by the models that integrate the electrical power grid with the detailed gas turbine dynamics.

Furthermore, the multi-domain physical modeling and simulation Modelica language has been employed to create the equation-based models of the use case of this project. A comparative analysis between the Single-Domain and Multi-Domain model responses has been performed both in time and frequency. The results show some interesting differences between the turbine dynamics representation of the commonly used GGOV1 standard model and the less simplified model of a gas turbine.

Finally, the models from each domain can be exchanged between two different stakeholders by means of Functional Mock-Up Units (FMUs), defined by the FMI standard. Promising test results were obtained with different simulation tools that support the standard, which demonstrates the feasibility of exchanging unambiguous multi-domain models with a detailed gas turbine representation. This shows the potential of the FMI standard for manufacturers to exchange equation-based multi-domain models, while at the same time protecting their intellectual property.

SAMMANFATTNING

Projektet ITEA3 OpenCPS (Öppen, certifierad modelldriven utveckling för cyberfysiska system) fokuserar på interoperabilitet mellan standarderna för Modelica, Unified Modeling Language (UML) och Functional Mock-up Interface (FMI), förbättring av tidsåtgången för (sam-)simulering, samt verifierad kodgenerering. Projektet syftar till att utveckla en plattform för modellering och simulering av cyberfysiska system och system som representeras av olika sorters fysikaliska modeller (eng. multi-domain modeling). Ett av de främsta användningsområdena för plattformen är ekvationsbaserad modellering och simulering av detaljerade gasturbinmodeller i elkraftverk (inklusive explicit ekvationsbaserad modellering av dynamiken mellan termiska och roterande komponenter) som är sammankopplade med kraftsystemet.

I detta examenarbete används klassdiagram i UML baserade på standarden för Common Information Model (CIM) och komponentdiagram från SysML för att beskriva kraftsystemet. En utvidgning baserat på ISO 15926-standarden föreslås för att härleda semantiken för modeller som integrerar både kraftsystemet och detaljerad dynamik för gasturbiner.

Vidare så har Modelica, ett språk för modellering och simulering av olikartade fysikaliska system, utnyttjats för att skapa ekvationsbaserade modeller som utvecklats i detta projekt. En komparativ analys har genomförts för en detaljerad modell med termodynamik och en förenklad modell genom undersökning av både frekvenssvar och tidssimuleringar. Resultaten visar att skillnader uppstår mellan den detaljerade modellen och den i kraftsystemsammanhang allmänt använda modellen GGOV1, vilket kan förklaras med förenklingar i den snare

Slutligen kan modeller från olika fysikaliska domäner utbytas mellan intressenter med s.k. Functional Mock-Up Units (FMU-enheter), som definieras av FMI-standarden. Lovande resultat uppnåddes med simuleringsverktyg som stöder FMI-standarden vilket visar möjligheten till ett otvetydigt utbyte av detaljerade gasturbinmodeller och elkraftsmodeller. Detta visar potentialen i FMI-standarden för modellutbyte mellan olika modelleringsdomäner, vilket skulle kunna låta tillverkare att dela ekvationsbaserade modeller utan att ge upp immateriella tillgångar.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGMENTS

I would first like to thank my thesis supervisor Prof. Luigi Vanfretti. It was very encouraging to be welcomed with his detailed presentation to the SmarTS Lab research topics. I was very motivated by the challenges he proposed to me later, some of which went beyond the basic objectives of my thesis. Additionally, something that deserves to be highlighted is the freedom that he always gave me to contribute with my own ideas. I believe that under his supervision I have not only acquired theoretical tools, but also some research skills.

The last thing I mentioned was also possible thanks to all the help I received from the PhD students at the lab. I am grateful to Tetiana Bogodorova for all the feedback I received from her. I did not only get support in modeling and control theory, but also very valuable general advices. I would also like to acknowledge Francisco Gómez for his supervision in the semantic modeling of this thesis. It has been quite fun to work with him. I don't want to leave Tin Rabuzin, Maxime Baudette and Jan Lavenius out of the list. Thanks for the initial models, the training and the challenging questions.

Finally, I want to dedicate this thesis to my family and friends, from whom I always received support to continue.

This work was sponsored by ICE (Instituto Costarricense de Electricidad) and MICIT (Ministerio de Ciencia y Tecnología de Costa Rica) through a grant for post-graduate studies. Their support is gratefully acknowledged.

ACRONYMS

CGMES. Common Grid Model Exchange Specification.

CIM. Common Information Model.

CPS. Cyber-Physical Systems.

DAE. Differential Algebraic Equations.

FMI. Functional Mock-up Interface.

FMU. Functional Mock-up Unit.

GGOV1. GE General Governor/Turbine Model.

IEC. International Electrotechnical Commission.

ISO. International Organization for Standardization.

MST. Master Simulation Tool.

SGAM. Smart Grid Architecture Model.

SMIB. Single Machine Infinite Bus.

SysML. Systems Modeling Language.

TSO. Transmission System Operator.

UML. Unified Modeling Language.

Chapter I

1. INTRODUCTION

1.1 Background

Environmental concerns are driving agreements on new energy technology and business trends in different regions of the world. Several roadmaps have proposed targets for the medium and long-term respect to carbon dioxide (CO_2) emissions reduction. Regardless of the aggressiveness of the envisioned goals, there is a consensus that to achieve a more sustainable energy supply it is required to significantly speed up the integration of renewable resources.

The growth potential is higher for the category of the so-called variable energy resources (VER), respect to other conventional renewable sources like hydro and geothermal power. Variable energy resources, like wind and solar power, require a special attention due to the challenges that its intermittent nature impose on the power grid operation. As a result, a smooth transition to an overarching architecture that allows acceptable levels of reliability and security and affordable prices is required [1].

The operational flexibility of gas power plants makes them a good backup to variable renewables. It is very likely that policies will promote the increase of gas power, at least in the next decade, especially because they produce less emissions. Furthermore, new improvements including the application of carbon capture and storage (CCS) will eventually extend their use even after 2040 [2].

The variability of wind and solar power can be expressed as slow or fast fluctuations. Both, the less environment-friendly coal power plants and, combined cycle gas plants can be used to compensate slow power intermittency. On the other hand, power increase/reduction required to deal with fast power fluctuations can be achieved by means of fast ramping sources like gas natural

turbines. Manufacturers like Siemens or GE have recently introduced combined cycle technologies that are still able to deliver relatively quick start ramps while keeping the high combustion efficiencies that are expected from this kind of power plants [1].

A reliable operation of power systems with high penetration of VERs depends, among other factors, on more trustable forecasts and accurate models that can be tailored to the several kinds of off-line power system simulations. Existing gas turbine models, such as GGOV1, IEEE [3] and Rowen, have different levels of complexity and accuracy. Simplicity was a desired property for the first proposed models, primarily due to computer power and data availability limitations of the time when they were proposed, the 1980- early 1990s. Such was the case of the GAST model which was widely used in the United States, but was demonstrated to be inaccurate and thus replaced by the somewhat more complex GGOV1 model. The widely-accepted models GGOV1, IEEE and Rowen do not employ a detailed physical representation of the gas turbine dynamics; instead, they model dynamics using abstractions in the form of logic and transfer functions. In fact, it has been recently shown [4] that more detailed models are required to include the frequency dependency feature of gas turbines with the aim of undertaking power system stability studies on abnormal system frequency behavior (e.g. black start and islanded operation). Because the correctness of the more complex physical models of gas turbines relies on the availability of data from the manufacturers, it is reasonable that they create and then share such models [4].

The evolution towards a more sustainable power system is also the motivation of a smart grid. This technology and business trend of the power industry aims to improve the operation and control of the power system, leading to its increased security, reliability and efficiency. For this purpose, the smart grids bring convergence between control systems and, information and communications systems [5]. A concept that can contribute to the goal of smart grids is that of

'cyber physical systems', systems that offer a tight integration between computation, networking and physical processes.

The adoption of the "The Third Energy Package" [6] of legislative proposals by the European Commission (EC) on September 19, 2007 aimed to enable a competitive and integrated market. As a next step, the European Network of Transmission System Operators of Electricity (ENTSO-E) was established by Regulation (EC) 714/2009 [7] with the purpose of guaranteeing the reliable and secure operation of the European interconnected transmission grid. The regulation emphasized the need of coordination between TSOs, based on the definition of a common transmission model. The use of this model was intended to support "common network operation tools to ensure coordination of network in normal and emergency conditions".

The EC then issued Mandate M/490 to the European standardization bodies CEN/CENELEC and ETSI. A resulting report [5] considered that the Common Information Model (CIM) (IEC 61970, 61968 and 62325) together with IEC 61850 were the most relevant data models, required to meet the stated recommendation of establishing a "common information model that is to be used throughout many applications and systems".

The information exchange required to meet the needs of coordination of TSOs operation under any conditions, should comprise both steady-state and dynamic models that can be used for power system simulation. Although the CIM is currently addressing the requirement of dynamic information exchange through IEC-61970-302 and IEC 61970-457, it might lead to the exchange of ambiguous models. There are two main reasons to explain why this situation can happen. First, the modeling approach based on conventional block diagrams only encourage the sharing of parameters, and hence excludes an explicit mathematical representation. Second, dynamic models lack consistency among different

platforms due to simplifications that emerge from their modeling paradigm and implementations [8].

The authors of [8] and [9] have shown that *Modelica* language is able to cope with the exposed ambiguous model sharing issue while facilitating the access and/or modification of models at the "equation-level". Some additional advantages of *Modelica* are the open distribution of several libraries meant to represent physical systems, and the fact that models are independent from IDEs and solvers [9].

1.2 Problem Definition

The ITEA3 OpenCPS (Open Cyber-Physical System Model-Driven Certified Development) project aims to provide interoperability between the Modelica/Unified Modeling Language (UML)/Functional Mock-up Interface (FMI) standards, together with improved (co-)simulation execution speed, and verified code generation. The project aims to develop modeling and simulation toolchains that can be applied to cyber-physical and multi-domain systems [10]. Within its work package D5.3B, *CIM/UML and Modelica Electrical Power System Models and Tooling for MST Testing and Demonstration*, KTH SmarTS Lab agreed to develop benchmark power grid models that could be used to test the functionalities of the OpenCPS toolchains.

In the second use case of the work package, the benchmark case corresponds to multi-domain models of improved gas turbines coupled to the power grid to meet European standardization requirements for grid connection. These requirements include design space exploration and trade-off analysis, together with information exchange through the IEC-CIM-based Common Grid Modeling Exchange Standard (CGMES).

The use case is indeed a Smart Grid application involving several stakeholders and electric power domains and therefore it can be represented by means of the SGAM (Smart Grid Architecture Model) plane. Figure 1.1 shows the result of placing this use case inside of the SGAM plane. As it can be appreciated this

scenario encompasses the generation and transmission domains, since the power grid is represented by a Single-Machine Infinite Bus (SMIB) model. It does not only cover the electric power conversion chain but also the controls of the gas turbine and the generator and thus, both the process and the field level are being employed. A new envisioned player in this use case might be either a gas turbine manufacturer, or another turbo-machinery domain expert from the Academy or the Industry. Its role is to provide the physical gas turbine model that supersedes the typical simplified model used by the electric power system simulators. The final customers of the multi-domain models are the TSOs which would be using a combination of CIM/XML standards, Modelica libraries and Functional Mock-up Units (FMUs) to exchange the required information. The semantics are initially given by the IEC-CIM-UML but there is a need of an extension to embrace the non-electric power parametrization of the gas turbine.

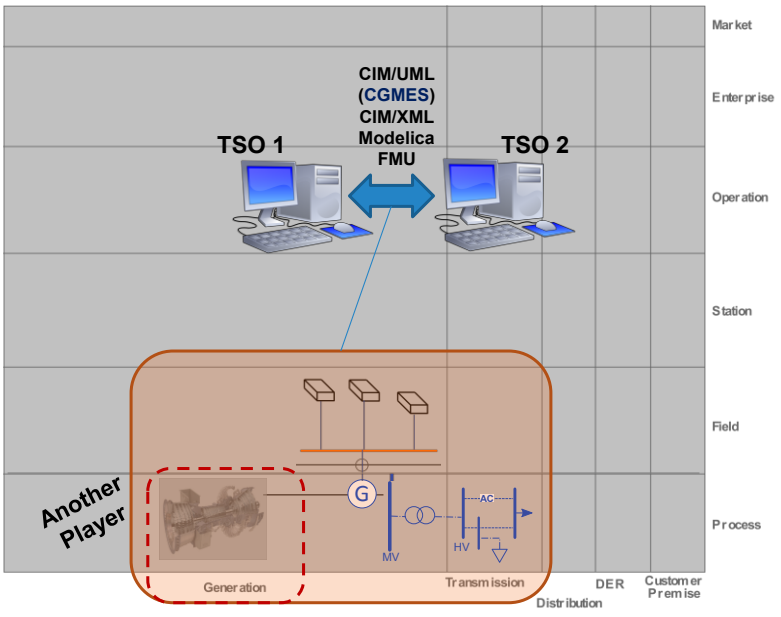


Figure 1.1 OpenCPS D5.3B Use Case 2 Application Case in the SGAM plane

The multi-domain models from Use Case 2 need to be compatible and follow the different toolchains being developed in OpenCPS. Two work streams were adopted in the form of Master Simulation Tools (MST) which are depicted on Figure 1.2.

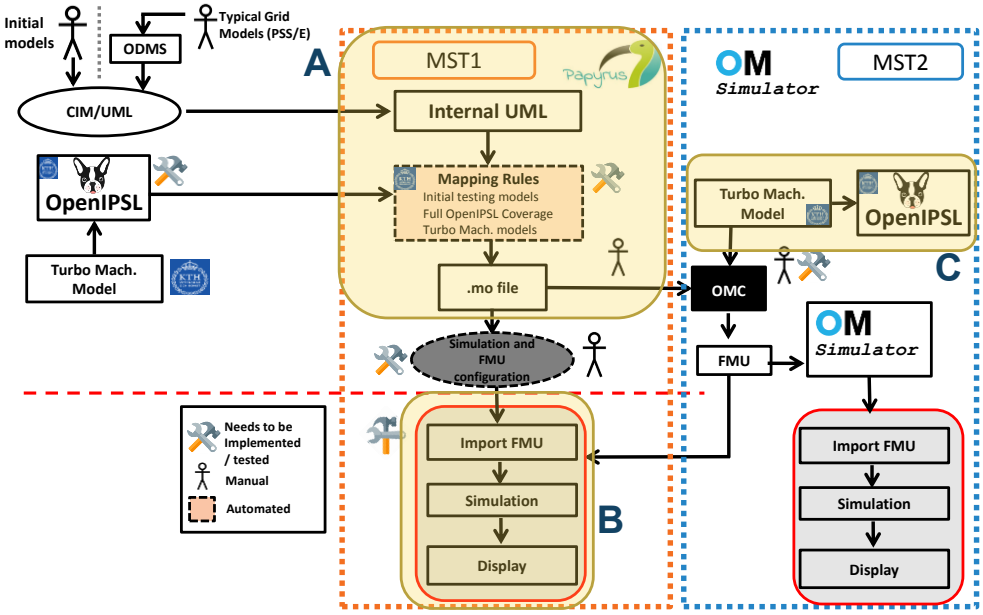


Figure 1.2 MST Workflows for OpenCPS D5.3B Use Case 2

A first MST makes use of an Eclipse-based modeling tool called Papyrus. It enables the creation of an Internal UML model that describes the data exchange format required to obtain the parameters of the multi-domain model. As it was explained before, the UML model uses the CIM standards as a basis but requires an extension to include the semantics of the gas turbine. Through mapping rules, the complete Modelica *.mo files* can be generated and then compiled as FMUs. Finally, the FMUs should be able to be imported and simulated in Papyrus.

The second MST employs *OpenModelica* instead of Papyrus as the modeling and simulation environment. The Modelica benchmark models can be edited inside and then compiled and simulated as FMUs.

OpenIPSL is an open-source Modelica library for power systems developed by SmarTS Lab [11]. It contains the required components to model the electric power grid, however it does not provide the components to represent the gas turbine in detail. An additional library should thus be combined with *OpenIPSL* to achieve the multi-domain benchmark model.

1.3 Objectives

The objective of this thesis is to develop the multi-domain gas turbine and power grid semantic and equation-based models, required to test the functionalities of the OpenCPS toolchains as explained in the previous section. In more detail, the sub-tasks to be solved in this thesis are,

a) Generate the Modelica Multi-domain model that comprises the physical model of the gas turbine, the governor, a Single Machine Infinite Bus (SMIB) network and a stochastic load model.
b) Create the semantic UML/SysML models needed to define the parameter values that will populate the Modelica multi-domain model instances before simulation.
c) Perform an analysis of the multi-domain system from (a) that includes a comparison with the GGOV1-based equivalent system.
d) Generate FMUs for the SMIB network, the gas turbine and the governor model blocks from (a) to test the FMU simulation stages of the OpenCPS MST workflows.

1.4 Overview of the report

The thesis is divided into five chapters:

Chapter 1: INTRODUCTION This chapter provides an overview of the context of the project together with the problem formulation and the list of objectives.

Chapter 2: THEORETICAL BACKGROUND This chapter offers an overview of the theoretical concepts related to the operation, modeling and control of gas turbines.

Chapter 3: EQUATION-BASED MODELS This chapter describes the equation-based models that were developed in Modelica to meet objective **a** of the project.

Chapter 4: SEMANTIC MODELS This chapter begins with a description of a methodology for multi-domain semantic modeling. Later, a concrete model has been proposed for the case of gas turbines, to fulfill the requirement of objective **b**.

Chapter 5: Studies, Simulation and Results This chapter describes the studies and simulations that were applied on the models. Later, the results are shown and discussed.

Chapter 6: Multi-Domain Modelling and Simulation with FMUs This chapter presents a user's guide that is aimed to help analysts to export FMUs in Dymola, and then import them in different tools to perform multi-domain simulation.

Chapter 7: CLOSURE This chapter offers a general summary of the work done in the thesis along with the conclusions and the future work.

The Appendices contain supporting data and results that could not be included in the chapters of this report.

Chapter II

2. THEORETICAL BACKGROUND

2.1 Gas Turbine Theory

The principles of operation of the gas turbine can be better understood if its theory is presented in a progressive way. In this section, the gas turbine will first be defined as Carnot heat engine. The thermodynamic cycle followed by the gas turbine, i.e. the Brayton Cycle, will be discussed as a next step. Then, the component operation characteristics required to represent the off-line performance of gas turbines will be described. Finally, the modeling equations of the heat addition process that takes place in the combustion chamber will be explained.

2.1.1 The Gas Turbine as a Heat Engine

A gas turbine can be described as a heat engine, a device that produces work from a heat source while operating in a thermodynamic cycle that also leads to heat rejection to a heat sink (see Figure 2.1). Based on the definition of the first law of thermodynamics, when 5 MJ of heat is supplied to the engine (Q_1) no more than 5 MJ minus the rejected heat (Q_1) can be obtained as work. Moreover, the second law of thermodynamics explains that it is not possible to get 5 MJ of work, if 5 MJ of heat is being supplied to the engine. The previous statement yields to the concept of efficiency, the ratio of the work output (W) and the heat input, which therefore will have values that can never be one. The reason that explains why the heat engine cannot reach 100% thermal efficiency is that there is an increase in entropy during the cycle that prevents the heat rejection (Q_2) from getting zero. Equations (2.1) and (2.2) can be used to define work and efficiency in the device, respectively [12].

$$W = Q_1 - Q_2 \tag{2.1}$$

$$\eta = \frac{W}{Q_1} = 1 - \frac{Q_2}{Q_1} \tag{2.2}$$

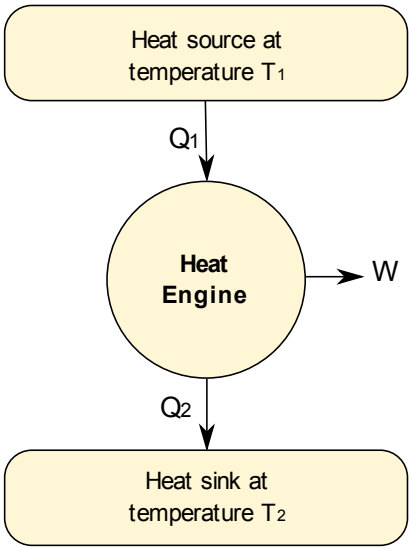


Figure 2.1 Gas Turbine as a Heat Engine

The maximum efficiency developed by a heat engine is defined by the Carnot efficiency $\eta_{th,max}$ given by Equation (2.3)

$$\eta_{th,max} = 1 - \frac{T_1}{T_2} \tag{2.3}$$

where T_1 and T_2 correspond to the temperatures of the heat source and sink respectively [13].

A gas turbine could also be interpreted as a system where a fluid (gas) is flowing at a steady rate while both work and heat are being transferred to the surroundings. This phenomenon is analytically described by equation (2.4), the so-called steady flow energy equation.

$$Q - W = h_1 - h_2 = c_p(T_1 - T_2) \tag{2.4}$$

where Q is the specific heat transfer (kJ/kg), W is the specific work transfer (kJ/kg), h_1-h_2 is the specific static enthalpy change (kJ/kg), c_p is the specific heat at constant pressure and $T_1 - T_2$ is the static temperature change. It is important to mention that equation (4) is only valid under the assumption of a perfect gas [13].

2.1.2 The Brayton Cycle

The single gas turbine operation follows a thermodynamic cycle known as the Brayton cycle. A combined cycle gas turbine configuration, which typically has a steam turbine as a second stage, also follows one process of a Rankine cycle.

The Brayton cycle is composed of four processes, two adiabatic and two isobaric. An adiabatic process is a process where work is produced with no heat transfer, and an isobaric process means constant pressure. The first process is an adiabatic compression that is performed by a compressor. It is then followed by a heat supply at constant pressure obtained by means of a heat exchanger, the combustion chamber. The next process is an adiabatic expansion produced by a turbine (expander), after which there is an isobaric heat release [12].

A thermodynamic cycle can be represented by a temperature – entropy (T-s) diagram or, by a pressure - volume (P-v) diagram. A good property of these diagrams is that they allow an easy graphical evaluation of the work and/or heat transfer given by each process, since these quantities can be determined by solving the integrals $\int p\, dv$ and $\int t\, ds$, respectively. The best way to illustrate the Brayton cycle is by employing a T-s diagram [13].

It is very common that the literature describes the ideal Brayton cycle first, and then continues with the explanation of the actual cycle. Nevertheless, for the sake of brevity both concepts will be presented together. Figure 2.2 shows the T-s diagram of the ideal and actual gas turbine cycles, while Figure 2.3 shows the components that participate in each process according to the explanation given in

the previous paragraph. In the ideal cycle, the adiabatic compression (process 1-2) and the adiabatic expansion (3-4) are said to be isentropic, which means that the entropy does not change.

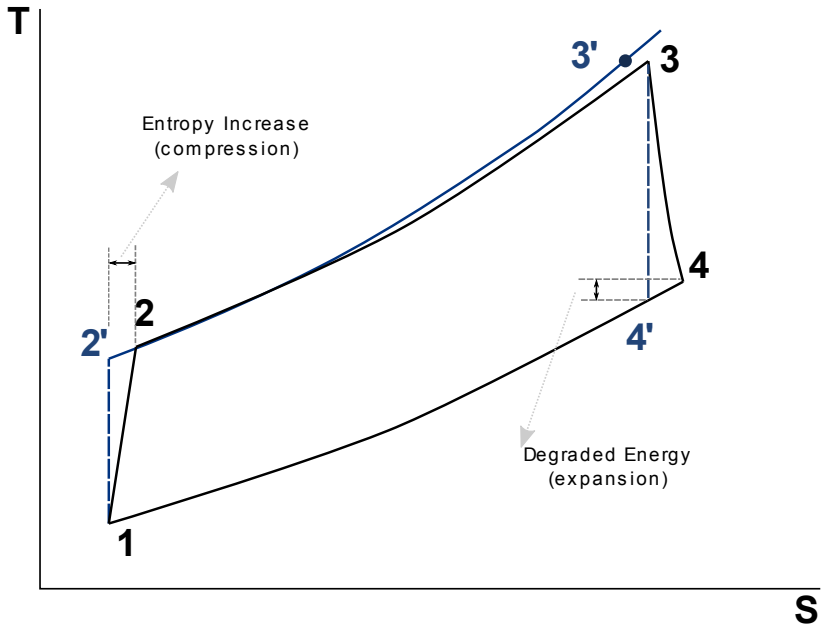


Figure 2.2 T-s diagram of the ideal and actual Brayton Cycle

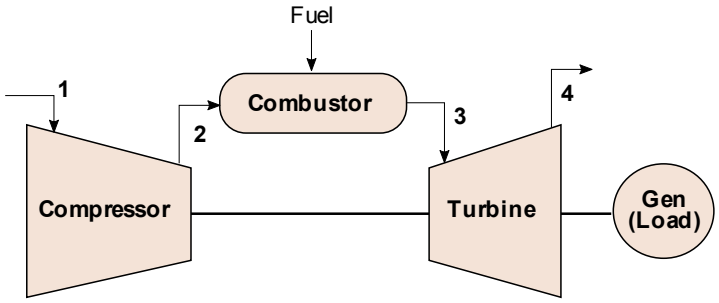


Figure 2.3 Diagram of a simple open cycle gas turbine

The steady flow energy equation can be applied to obtain expressions for the work required by the adiabatic compression (WC) and the work produced in the adiabatic expansion (WT) of the ideal cycle [13]:

$$W_C = W_{12} = m \cdot c_p \cdot (T_2 - T_1) \tag{2.5}$$

$$W_T = W_{34} = m \cdot c_p \cdot (T_4 - T_3) \tag{2.6}$$

The power required by the compressor and the work produced by the expansion turbine can also be written in terms of the difference of enthalpies and the mass flow rate of the fluid in each process [14]:

$$\dot{W}_C = \dot{m}_a \cdot (h_2 - h_1) \tag{2.7}$$

$$\dot{W}_T = (\dot{m}_a + \dot{m}_f) \cdot (h_3 - h_4) \tag{2.8}$$

Additionally, the power transmitted from and to the shaft by these components can be written as:

$$\dot{W}_C = \tau \cdot \omega \cdot \eta_{mech} \tag{2.9}$$

$$\dot{W}_T = \frac{\tau \cdot \omega}{\eta_{mech}} \tag{2.10}$$

where η_{mech} is the mechanical efficiency, τ is the torque and ω is the rotational speed [15].

In the actual cycle, the increase of enthalpy and the subsequent energy degradation effect, are considered (i.e. the processes are no longer isentropic). Figure 2.2 shows the actual compression process (1-2), the actual expansion process (3-4), as well as the isentropic compression (1-2') and the isentropic expansion (3-4')

The energy degradation of the actual cycle gives rise to the concept of isentropic adiabatic efficiencies for the compression and the expansion processes. Those efficiencies can be defined as the ratio of the ideal work to the actual work. Hence, the following equations can be derived to define the adiabatic compression and adiabatic expansion efficiencies, respectively [12], [13]:

$$\eta_C = \frac{W_{ideal}}{W_{actual}} = \frac{(h_2' - h_1)}{h_2 - h_1} \tag{2.11}$$

$$\eta_T = \frac{W_{actual}}{W_{ideal}} = \frac{(h_3 - h_4)}{h_3 - h_4'} \tag{2.12}$$

Alternatively, if constant specific heats are assumed [16]:

$$\eta_C = \frac{(T_2' - T_1)}{T_2 - T_1} \tag{2.13}$$

$$\eta_T = \frac{(T_3 - T_4)}{T_3 - T_4'} \tag{2.14}$$

In the ideal cycle, the heat input (from process 2-3) is given by [13]:

$$\boldsymbol{Q_{23}} = c_p(T_3 - T_2) \tag{2.15}$$

The final temperatures of the compression and expansion processes, T_2 and T_4 respectively, can be expressed as:

$$T_2 = T_1(PR_C)^{\frac{\gamma-1}{\gamma}} \tag{2.16}$$

$$T_4 = T_3(PR_T)^{\frac{\gamma-1}{\gamma}} \tag{2.17}$$

where γ is the isentropic index, the ratio of specific heats at constant pressure c_p and at constant volume c_v; while PR_C and PR_T are the pressure ratios of the compression and expansion processes [13]:

$$PR_C = \frac{P_2}{P_1} \tag{2.18}$$

$$PR_T = \frac{P_4}{P_3} \tag{2.19}$$

The total output work of the single cycle gas turbine is [12]:

$$W_{tot} = W_T - W_C \tag{2.20}$$

So, if constant specific heats are considered, the specific work (net work done per unit mass flow rate) is [13]:

$$\boldsymbol{W_{tot}} = c_p(T_3 - T_4) - c_p(T_2 - T_1) \tag{2.21}$$

The definition of thermal efficiency that was presented with the heat engine theory (equation 2.2) can be applied to the gas turbine [13]:

$$\eta_{th} = \frac{\boldsymbol{W_{tot}}}{\boldsymbol{Q_{23}}} \tag{2.22}$$

By substituting equations (2.16), (2.17), (2.21) and (2.22) and then re-arranging it is possible to obtain a simplified expression of the thermal efficiency of the ideal simple-cycle gas turbine:

$$\eta_{th} = 1 - \frac{T_1}{T_2} = 1 - \frac{1}{PR_C^{\frac{\gamma-1}{\gamma}}} \tag{2.23}$$

As it can be seen, the ideal cycle thermal efficiency depends only on the compressor temperature ratio and, its value is less than the theoretical maximum thermal efficiency of a heat engine, the Carnot efficiency ($\eta_{th,max} = 1 - T_1/T_3$) [13].

However, a more elaborated expression can be obtained for the actual cycle case, due to the influence of the adiabatic compression and expansion efficiencies [14]:

$$\eta_{th} = \frac{\eta_t T_3 - T_1 PR_C^{\frac{\gamma-1}{\gamma}}}{T_3 - T_1 - T_1\left(\frac{PR_C^{\frac{\gamma-1}{\gamma}} - 1}{\eta_C}\right)} \tag{2.24}$$

Additionally, if the same mass flow rate $\dot{m}$ is assumed in the compression and expansion processes, the following expression can be derived for the total work output of the actual cycle of the gas turbine [16]:

$$W_{tot} = \dot{m} c_p \left[\left(\eta_t \frac{T_3}{T_1} - \frac{PR_C^{\frac{\gamma-1}{\gamma}}}{\eta_C} \right) \left(1 - \frac{1}{PR_C^{\frac{\gamma-1}{\gamma}}} \right) \right] \quad (2.25)$$

From equations (2.24) and (2.25), it can be observed that the thermal efficiency and the output power of the actual cycle depend on the compressor pressure ratio PR_C, the maximum and the minimum cycle temperatures (T_3 and T_1). In general, the thermal efficiency and the output work increase with the maximum cycle temperature and the pressure ratio. However, after a peak value is obtained at a certain pressure ratio, these variables start to decrease [13]. Several cycle modifications can be applied to obtain higher thermal efficiencies and/or total work. Some common examples are: regeneration/recuperation, compressor intercooling, turbine reheat and, water injection. For instance, regeneration can lead to higher efficiencies but in a smaller range of low values of compressor pressure ratios [16]. Detailed description about these cycle variants can be found in [13], [14] and [16].

2.1.3 Compressor and Turbine Operation Characteristics

The operation of the compressor and the turbine over given ranges of flow rate and speed can be described by means of operation characteristics [14].

For instance, for a compressor the following expression can be used to relate the pressure ratio and the efficiency with the flow rate $\dot{m}$, the speed N, and the ambient conditions T_a and P_a:

$$PR_C, \eta_C = f\left(\frac{\dot{m}\sqrt{T_a}}{P_a}, \frac{N}{\sqrt{T_a}} \right) \quad (2.26)$$

Since the scaling of the speed and flow rate variables by a constant do not introduce any functional dependence, it is possible to re-formulate the previous expression as:

$$PR_C, \eta_C = f\left(\frac{\dot{m}\sqrt{\theta}}{\delta}, \frac{N}{\sqrt{\theta}}\right) = f(\dot{m}_c, N_c) \tag{2.27}$$

where $\dot{m}_c$ and N_c are the corrected flow rate and speed, θ and δ are scaling factors defined as follows

$$\theta = T_a/T_{a0} \tag{2.28}$$

$$\delta = P_a/P_{a0} \tag{2.29}$$

The reference temperature T_{a0} and reference pressure P_{a0} can take the values of the ISO reference conditions, i.e. 15°C and 101 kPa, respectively [14].

The operation characteristics can be represented as a map $PR = f\left(\frac{\dot{m}\sqrt{\theta}}{\delta}, \frac{N}{\sqrt{\theta}}\right)$ with contour graphs for several efficiencies η. However, this is not suitable for models of a gas turbine aimed to be used in dynamic simulations. Hence, a table or look-up table representation is used instead where PR_C, η_C and $\frac{\dot{m}\sqrt{\theta}}{\delta}$ can be defined as [17]:

$$PR_C = f\left(\beta, \frac{N}{\sqrt{\theta}}\right) \tag{2.30}$$

$$\eta_C = f\left(\beta, \frac{N}{\sqrt{\theta}}\right) \tag{2.31}$$

$$\frac{\dot{m}\sqrt{\theta}}{\delta} = f\left(\beta, \frac{N}{\sqrt{\theta}}\right) \tag{2.32}$$

with β, an auxiliary variable used as pivot in the look-up table. This variable is determined from the conversion from the graphical maps to table-format, that requires the introduction of uniformly separated parallel lines to the surge-line (β-lines) [15]. The surge line defines an unstable operational limit known as surging, characterized by a flow reversal and a breakdown of the continuous steady flow through the compressor [14].

2.1.4 Heat Addition and the Combustion Chamber

If pressure losses are ignored, the behavior of the combustion chamber can be described in terms of the mass and energy conservation equations:

$$V_{cc}\frac{d\rho}{dt} = \dot{m}_{air} + \dot{m}_{fuel} - \dot{m}_{flue\ gas} \tag{2.33}$$

$$\frac{d\left(\dot{M}_{cc}U_{cc}\right)}{dt} = \dot{m}_{air}h_{air} + \dot{m}_{fuel}LHV - \dot{m}_{flue\ gas}h_{flue\ gas} \tag{2.34}$$

where LHV corresponds to the lower heating value of the fuel composition, while M_{cc} and U_{cc} are the total mass of gases at each time step and the internal specific energy of the combustion chamber, respectively. The total mass of gases M_{cc} is the product of the density of the flue gas ρ and the volume of the combustion chamber V_{cc} [18].

A simplistic model of losses due to convection can also be included. If assumptions like thermally insulated wall from the outside, and uniform metal wall temperature and heat transfer coefficient between wall and fluid are considered, convection losses can be described by:

$$C_m\frac{dT_m}{dt} = \Gamma \cdot S \cdot \left(T_{flue\ gas} - T_m\right) \tag{2.35}$$

where C_m is the metal heat capacity (J/K), Γ is the heat transfer coefficient (W/m2K), S is the inner surface (m2) and T_m is the metal temperature [19].

To model the behavior of the combustion chamber it is also important to derive mass-balance equations for every component of the exhaust gas. If natural gas, consisting of nitrogen (N2), carbon dioxide (CO_2) and methane (CH_4) is taken as the fuel, the resulting chemical reaction is:

$$\begin{aligned} CH_4 + 2O_2 &\rightarrow CO_2 + 2H_2O \\ CO_2 &\rightarrow CO_2 \\ N_2 &\rightarrow N_2 \end{aligned} \tag{2.36}$$

Air can be assumed to be composed by the following species: oxygen (O_2), nitrogen (N_2), argon (Ar), water (H_2O) and carbon dioxide. Hence, the gas at the exhaust will be composed by molecules of those species, and a total of five mass-balance equations should be derived (one per component) [20].

For a complete combustion to be obtained, the fuel should be completely oxidized. The stoichiometric air/fuel ratio $(A/F)_s$ can be used as a reference value to determine if this is the case. For the chemical reaction of the natural gas combustion, it can be defined as:

$$(A/F)_s = \frac{m_a}{m_f} = \frac{2\frac{x_{f,CH_4}}{M_{CH_4}}}{\frac{x_{a,O_2}}{M_{O_2}}} \tag{2.37}$$

where x_{f,CH_4} and x_{a,O_2} are the methane and oxygen mass fractions of the fuel and the air, while M_{CH_4} and M_{O_2} are the mole masses of the methane and the oxygen, respectively [20].

2.2 Classification of Simple-Cycle Gas Turbines

Based on its size, simple-cycle gas turbines used for power generation can be classified in 4 main groups. A first group is composed by the large frame type heavy-duty gas turbines. Their generated power goes from 3 to 480 MW and they exhibit efficiencies that range from 30% to 48%. The second group comprises the so-called *aeroderivative gas turbines*, a variant that was originally conceived for aircraft applications. Their adaptation to the electric power generation industry required the removal of the bypass fans and the addition of a free power turbine at their exhaust. They provide output power in the range of 2.5 to 50 MW, with efficiencies ranging from 35 to 45%. The two last groups are small gas turbines and microturbines. Small turbines are in the range from 0.5 to 2.5 MW, while microturbines are in the range from 20 to 350 kW. Their efficiencies are lower than those of larger gas turbines [14].

If the arrangement of its components is the criteria of attention, then simple-cycle gas turbines can be classified as single-shaft gas turbines, two-shaft or three-shaft gas turbines with a power turbine and two-shaft gas turbines. The single-shaft gas turbines have already been discussed in the previous sections. They are specifically chosen for base-load power generation, where they operate at fixed speed. The two-shaft gas turbines with power turbine is the result of splitting the expansion process into two separate turbines. The first drives the compressor and together receive the name of gas generator. The second, called the power turbine, is employed to drive the load. In mechanical drive applications, it avoids the poor engine performance of the single-shaft layout that results from low speed operating condition. In industrial power generation applications, the power turbine is designed to operate at a fixed speed, while the gas generator speed varies with the electrical load [13].

2.3 Interaction between Gas Turbine Dynamics and Controls

As it has been discussed in the previous section, the temperature of the gases leaving the combustion chamber (T_3) is the highest temperature of the cycle. On the other hand, the ambient temperature T_1, corresponding to the temperature of the air entering the first stage of the compressor, is the lowest one. It has also been explained in the previous section that in general the output power and the efficiency of the gas turbine increase with T_3. Nevertheless, T_3 should be kept below a certain limit to reduce the stress that the turbine parts where the hot gases flow can experience.

Both T_1 and T_3 have an impact on the performance of the gas turbine. However, if T_3 is assumed to be kept at its maximum allowable value, then it can be stated that the maximum output will be a function of T_1 and the gas turbine shaft speed. An increase in the ambient temperature or a decrease in the shaft speed will result in less air mass-flow through the gas turbine components. Hence, the amount of fuel to be mixed and combusted with air must be reduced so that the maximum

limit of T_3 is not exceeded. The final consequence of this operational condition is a reduction in the output power delivered by the gas turbine [21].

Although essential for ensuring the correct operation of the gas turbine, it is in practice not possible to measure T_3. One reason is that the state of the hot gases leads to several temperature values instead of only one. Additionally, it is not feasible to place fast response time thermocouples at the inlet of the expansion turbine. Hence, the control of T_3 is achieved based on the indirect measurement of the temperature at the exhaust of the turbine T_4 [21].

A simplified thermodynamic model of the gas turbine that captures the description provided before by means of explicit equations is presented in [17] and [18].

The air mass flow rate can be expressed as a dimensionless or per-unit quantity W_a that is given by the following equation:

$$W_a = q(T_a, P_a)u(\Delta\omega_C)\frac{\sin(\theta_{IGV} - \theta_0)}{\sin(\theta_{max} - \theta_0)} \tag{2.38}$$

The parameters θ_0 and θ_{max} correspond to the initial and maximum inlet guide vane opening angles, while θ_{IGV} is the current angle. The dimensionless functions u and q are defined in terms of the per-unit speed change $\Delta\omega_C$, and the scaling factors of equations (2.28) and (2.29) as follows:

$$u(\Delta\omega_C) = 1 + A_0\Delta\omega_C + A_1\Delta\omega_C^2 + A_2\Delta\omega_C^3 \tag{2.39}$$

$$q(T_a, P_a) = \frac{\delta}{\sqrt{\theta}} \tag{2.40}$$

with the per-unit speed ω_C and speed change $\Delta\omega_C$ given by

$$\omega_C = \frac{\omega}{\sqrt{\theta}} \tag{2.41}$$

$$\Delta\omega_C = \omega_C - 1 \tag{2.42}$$

As it can be appreciated in Figure 2.4, given the air flow W_a from equation (2.38) and the fuel flow W_f, the exhaust temperature T_X, the discharge compressor ratio CPR, and the mechanical power P_m can be derived as [21]:

$$T_X = \frac{T_a + A_3\delta + A_4 W_f}{W_a} \tag{2.43}$$

$$CPR = \left(A_5 W_a + A_6 W_f\right)\frac{1}{\delta} + A_7 \tag{2.44}$$

$$P_m = \frac{1}{1 + T_{trb}s}\left(\frac{W_f - W_{f0}}{1 - W_{f0}}\right) \tag{2.45}$$

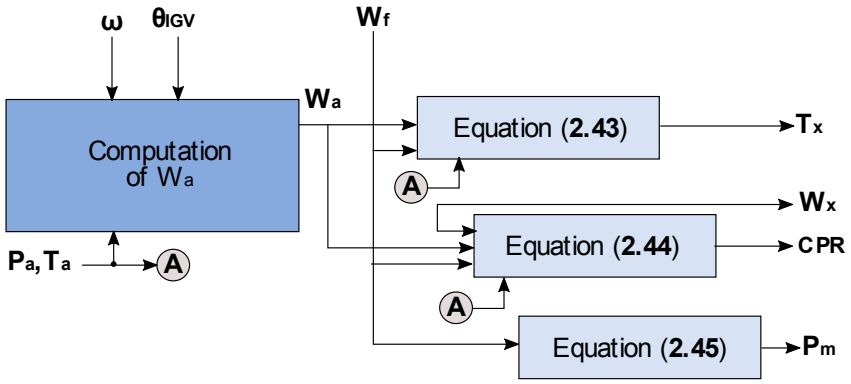


Figure 2.4 Explicit simplified gas turbine dynamics model (modified from [21])

The required gas turbine control systems and their interaction with the simplified gas turbine dynamics are shown in Figure 2.5. The vendor specific *gegt1*, available in the GE PSLF® software, utilizes the simplified explicit thermodynamic model of Figure 2.4 and implements the controls of Figure 2.5 [21][22]. However, the most commonly-used power system simulation tools employ turbine-governor models such as GGOV1 which involve further simplifications in the dynamics or the control modeling [21].

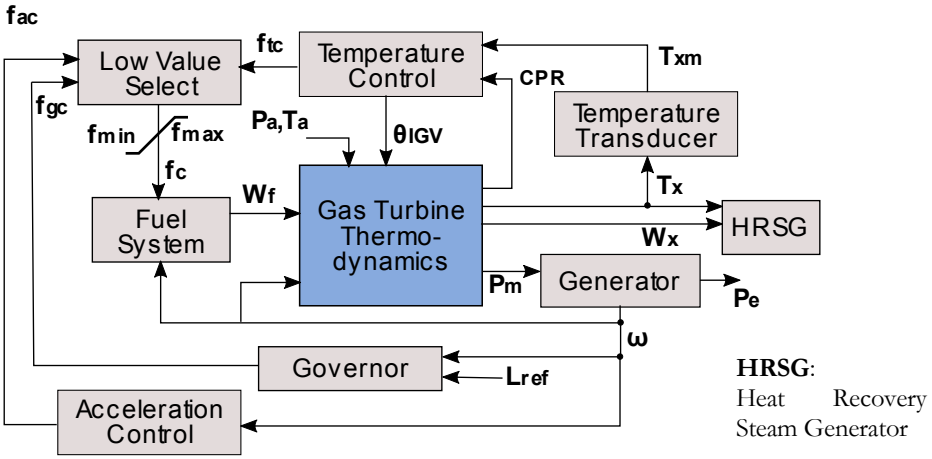


Figure 2.5 Combined Gas turbine dynamics and control model (modified from [21])

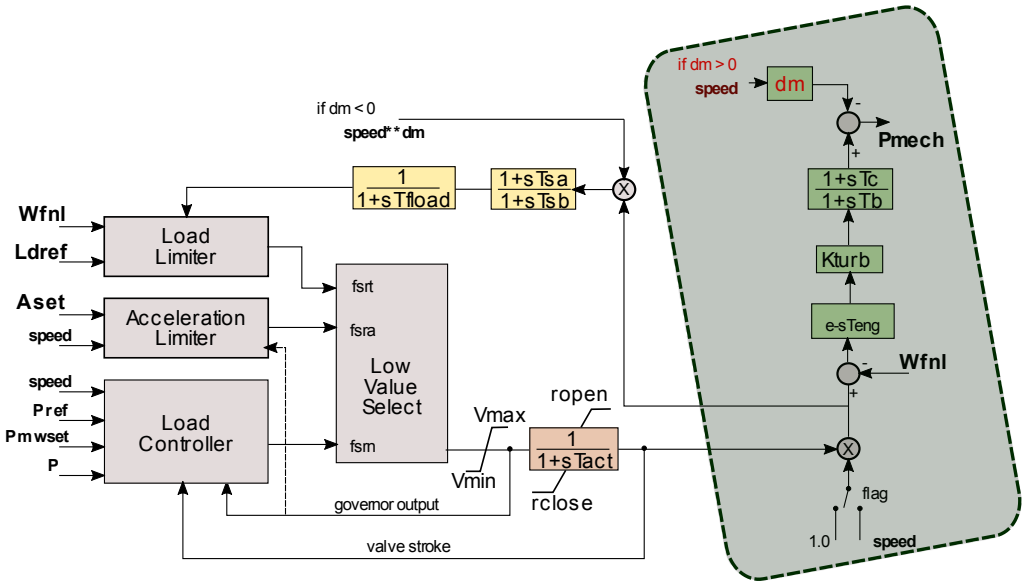


Figure 2.6 GGOV1 gas turbine – governor model (modified from [21])

The GGOV1 model is shown in Figure 2.6. Some considerations of this model are:

- A constant temperature limit (load limit) is assumed.

- There is no explicit representation of ambient effects (i.e. changes in ambient pressure and temperature).
- There is no explicit aim to include the dependence of the maximum turbine output power on the ambient conditions.
- A damping term in the speed (dm) is introduced to model the dependence of the maximum turbine output on the shaft speed (shown in yellow blocks).
- Inlet gate vane (IGV) controls and dynamics are neglected which may not give accurate results for large load variation studies (e.g. black start).

The turbine model of GGOV1 is simply composed by the blocks enclosed inside of the green rectangle of Figure 2.6: a gain K_{turb}, a transport delay with time constant T_{eng} and a lead-lag function with time constants T_b and T_c. The blocks in yellow give the exhaust temperature and its sensor, respectively. A complete description of the variables and parameters and their range can be found in [21].

Chapter III

3. EQUATION BASED MODELS

This chapter describes the equation-based models that were used or created in a Modelica environment to fulfill the objective (a) of this thesis.

3.1 Background

Modelica is an equation-based object-oriented programming language that enables the modeling and simulation of cyber-physical systems. Its mathematical textual definition allows acausal representation rather than just the commonly used assignment statements. In a process known as translation, that precedes the simulation, after which C code generation is obtained and then compiled [9][23]. Modelica delivers multi-domain modeling capabilities that will be exploited to achieve the objectives of this master thesis. In addition to the Modelica Standard Library (MSL), two libraries known as *OpenIPSL* and *ThermoPower* have been used to build the equation-based models of this project.

3.1.1 Package Structure

The equation based models were built or/and modified inside of a package structure in the Dymola F2016 Modelica IDE. The adopted package structure was conceived to classify the models in terms of the domain they belong to. As it can be seen in Figure 3.1 the first two packages, namely *TurboMachineryDomain* and *PowerSystemDomain*, contain the physical gas turbine models and the electric power system models, respectively. A third package, called *MultiDomain*, comprises the results of merging components from the two former packages to obtain the multi-domain equation based models. In the *PowerSystemDomain* only components based on the *OpenIPSL* library are included. In addition to the SMIB network models, a stochastic variable load model and the gas turbine controls based on the GGOV1 model are provided. On the other hand, the

TurboMachineryDomain package was aimed to comprise only elements from libraries specialized in gas turbines and other thermal power generation technologies such as *ThermoPower* [24], *ThermoSysPro* [25] or *ThermoFluid* [26].

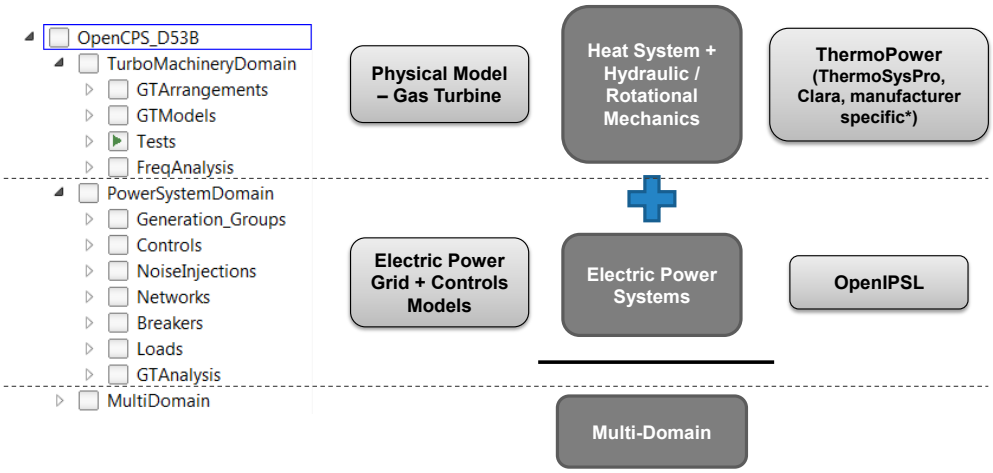


Figure 3.1 Adopted project package structure for the Modelica models

3.1.2 *OpenIPSL* Library

OpenIPSL is an open-source Modelica library that can be used to create power system networks and then perform dynamic time-domain simulation. Because it is based on the Modelica language it provides the flexibility that is not common to find in other power system modeling and simulation tools [11]. Some of the features of *OpenIPSL* are:

- It makes use of object-oriented paradigm properties like inheritance and instantiation.
- It can be supported by different Modelica compliant IDEs. The user can choose the development and environment of his preference. Additionally, it allows users such as TSOs to get equivalent simulation results if the

same integration solver (as the one used in their preferred proprietary simulation tool) is employed.

- The dynamic behavior of the models in the library have been described in terms of formal different algebraic equations (DAE) from theory. The user can have awareness of the model limitations due to its fully accessible description.
- It can be coupled with CIM/UML standards of the CGMES specification, to provide the complete required dynamic modeling details. One example of this, is the use of the Steady-State Hypothesis (SSH) and the State Variable (SV) CIM profiles to feed the *OpenIPSL* components with power flow solutions from external tools.
- It has been successfully tested with quite complex power system networks like Nordic 44-Bus system.

3.1.3 ThermoPower Library

ThermoPower is an open-source Modelica library that provides components that can be used to model thermal power plants. Some examples of the types of power plants that can be modeled are steam, gas and combined cycle plants [24] [27].

The library has 2 basic packages called Gas and Water containing models of components where the working fluid is water/steam or gas mixtures, respectively. The Gas package, for instance, contains the models of the gas turbine compressor, expansion turbine and combustion chamber. Their modeling description is based on similar equations to those presented in the first two sections of Chapter 2.

The package Media include default models of fluids which can be replaced by those compliant with the *Modelica.Media* interface. Another feature of *ThermoPower*

is the use of standard interfaces known as flanges that enable the component connectivity [27].

More information about the library can be found in the official website (see URL: https://casella.github.io/ThermoPower).

3.2 Turbo Machinery Domain Modelling

The *TurboMachineryDomain* package contains models which employ ThermoPower components. Its contents are organized in 3 sub-packages, namely *GTArrangements*, *GTModels* and *Tests*.

3.2.1 The *GTArrangements* package

As the names implies, the first package was aimed to include the elementary gas turbine topologies. The *SingleShaftGT* model represents a single shaft gas turbine and it is based on the *Plant* model of the Brayton Cycle examples of *ThermoPower*. The model excludes the boundary conditions, sensors and actuators and only focuses on the internal components of the gas turbine. The parameters of the compressor, combustion chamber and turbine are propagated and therefore, the *SingleShaftGT* can be used as a generic block in the representation of gas power plants. A second model (*TwinShaftGT*) that is available inside the *GTArrangements* package was an attempt to replicate the same design principle to model a twin shaft gas turbine.

3.2.2 The *GTModels* package

The second package has the models that result from combining the basic parametrized gas turbine arrangement with given boundary conditions, sensors and actuators. The only example included to date is the complete ThermoPower Single Shaft Gas Turbine *ThPowerSSGT* model, which can be seen in Figure 3.2.

Due to unavailability of data, the design parameters and component characteristics of the *ThPowerSSGT* gas turbine model were not modified respect to the ones of the original ThermoPower example. However, a still simple but

complete model of the fuel inlet valve that takes valve position as input instead of fuel mass flow reference was added. This change was needed to make the physical model of the turbine comparable to the simplified power system GGOV1-based turbine model.

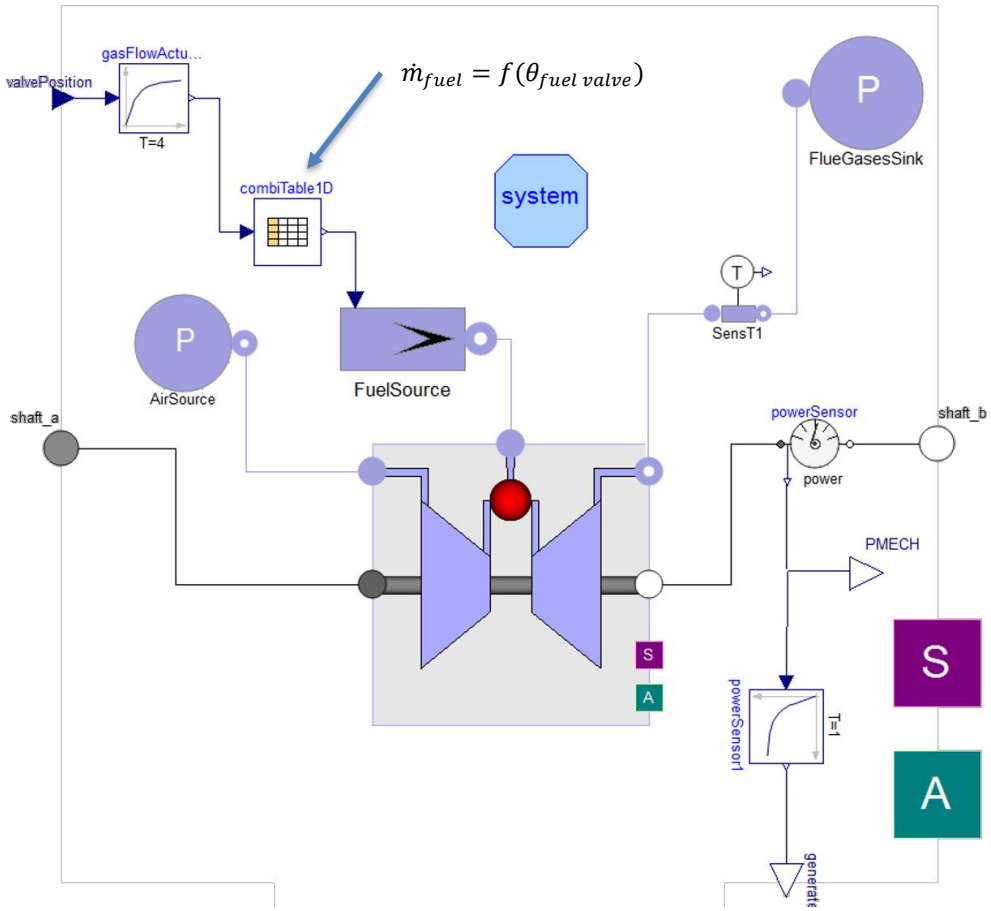


Figure 3.2 The ThermoPower Single Shaft Gas Turbine model

In order to build the valve model, the fuel mass flow rate $\dot{m}_{fuel}$ values required to obtain mechanical power from 0 to the maximum value of 10 MW were measured. Then, the gas turbine dynamics blocks from the GGOV1 model were used as a reference to obtain the following expression that relates the output mechanical power P_{mech} (in pu) with the fuel inlet valve position $\theta_{fuel\ valve}$ (also in pu):

$$\theta_{fuel\ valve} = P_{mech}/K_{turb} + W_{fnl} \tag{3.1}$$

where K_{turb} is the gas turbine gain and W_{fnl} is the fuel mass flow rate at no load conditions (in pu). Those parameters are set to the default recommended values of 1.5 and 0.15 respectively, leading to the data of Table 3.1.

Table 3.1 Fuel inlet valve model design data

P_{mech} (MW)	P_{mech} (pu)	$\theta_{fuel\ valve}$ (pu)	$\dot{m}_{fuel}$ (kg/s)
0	0	0.1500	1.8497
1	0.1	0.2167	1.9192
2	0.2	0.2833	1.9889
3	0.3	0.3500	2.0589
4	0.4	0.4167	2.1290
5	0.5	0.4833	2.1995
6	0.6	0.5500	2.2703
7	0.7	0.6167	2.3414
8	0.8	0.6833	2.4131
9	0.9	0.7500	2.4853
10	1.0	0.8167	2.5580

Finally, the curve $\dot{m}_{fuel} = f(\theta_{fuel\ valve})$ from Table 3.1 was specified in the model as the look-up table that is shown in Figure 3.2.

3.2.3 The *Tests* package

The third package integrates several open-loop and closed-loop tests that can be applied on the gas turbine models under load and no-load conditions. Because the gas turbine model makes use of a shaft flange interface from the MSL - Rotational Mechanics - library, it is possible to employ source models from this library such as *ConstantSpeed*, *Torque* or *Speed*.

3.3 Power System Domain Modelling

This section provides an overview of the grid, load and control models which are based on OpenIPSL library components.

3.3.1 Generation Groups

The use case 2 of the OpenCPS project work package which is associated with this thesis required different scenarios for the SMIB network model. Two

examples of these scenarios are a SMIB model without controls and a SMIB model with only excitation system. Table 3.2 summarizes the composition of the generation groups that were implemented to fulfill this need. They were made available inside the sub-package *Generation_Groups* of the *PowerSystemDomain* package. The last row of the table also describes the model of the infinite generator.

Table 3.2 Generation groups data

Gen. Bus Id.	Generator*	Exciter*	Turbine Governor*	Stabilizer*
GEN1	GENROU			
GEN1	GENROU	IEEE T1		
GEN1	GENROU	IEEE T1		PSS2A
GEN1	GENROU		GGOV1	
GEN1	GENROU	IEEE T1	GGOV1	PSS2A
GEN2	GENCLS			

*-detailed values of each type are given in Appendix A.

3.3.2 Controls

The GGOV1 model implementation of the OpenIPSL library was enhanced so to fit the needs of the studies of this thesis.

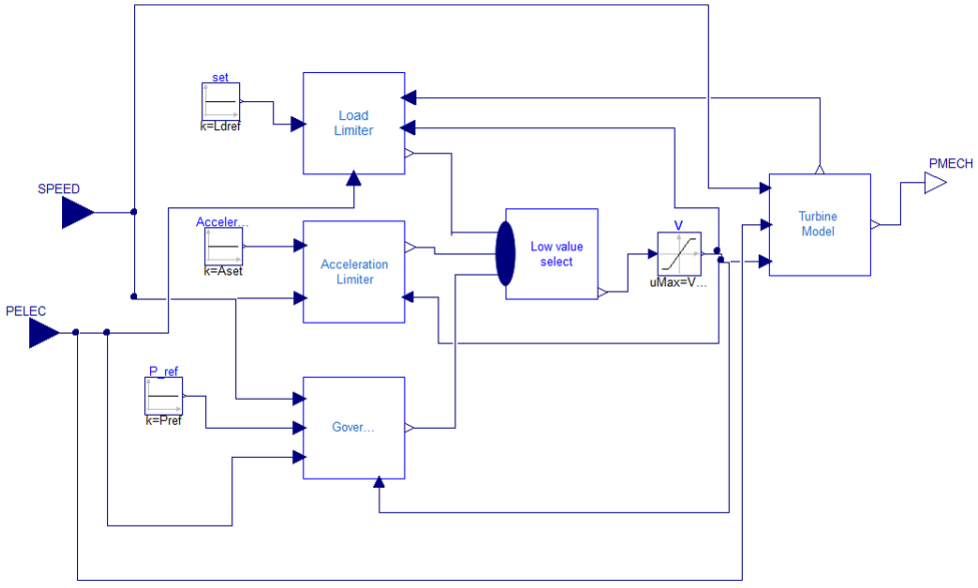


Figure 3.3 GGOV1 Turbine Governor model

As it can be seen in Figure 3.3, a re-design was applied on the GGOV1 model to explicitly show its internal composite blocks. This means that a separate model was created for each of the three controls logics that are inside of the GGOV1 model, namely the load limiter, the acceleration limiter and the main governor. Another model was developed to represent only the turbine, thus obtaining a convenient way to re-use the models when a certain study requires only the turbine or the governor instead of the complete group.

3.3.3 Load Models

In addition to the variable load model that is available in the OpenIPSL library, a variable load model with stochastic behavior was developed. The Modelica code of this model is as follows:

```
model VariableLoad "PSS/E Load with variation"
  extends OpenIPSL.Electrical.Loads.PSSE.BaseClasses.baseLoad;
  parameter Real d_P "Active Load Variation (pu)";
  parameter Modelica.SIunits.Time t1 "Time of Load Variation";
  parameter Modelica.SIunits.Time d_t "Time duration of load variation";
protected
  parameter Real PF=if q0 == 0 then 1 else p0/q0;
  parameter Real d_Q=(p0 + d_P)/PF - q0;
public
  Modelica.Blocks.Interfaces.RealInput u annotation (Placement(transformation
(extent={{48,16},{88,56}}), iconTransformation(extent={{-100,36},{-
62,74}})));
equation
  if time >= t1 and time <= t1 + d_t then
    kI*S_I.re*v + S_Y.re*v^2 + kP*(S_P.re + d_P) + u = p.vr*p.ir + p.vi*p.ii;
    kI*S_I.im*v + S_Y.im*v^2 + kP*(S_P.im + d_Q) = (-p.vr*p.ii) + p.vi*p.ir;
  else
    kI*S_I.re*v + S_Y.re*v^2 + kP*S_P.re + u = p.vr*p.ir + p.vi*p.ii;
    kI*S_I.im*v + S_Y.im*v^2 + kP*S_P.im = (-p.vr*p.ii) + p.vi*p.ir;
  end if;
end VariableLoad;
```

The code of this model is essentially the same as the original variable load model from the OpenIPSL library, with the difference that it has a real input for modulation u. Consequently, the new model has a component representing the physical load variability plus a component that allows for active power modulation. The second component is adjusted by the parameters d_P, d_t and t1, while the former relies on the definition of the modulation, namely noise injection source that is connected to this model.

As mentioned by the authors of [28], new technologies such as distributed generation lead to a more dynamic behavior of distribution networks. As a result, it is necessary to replace the traditional load deterministic models with models that show the normal stochastic behavior of the load. This is the role of the noise injection model that produces a noise-carrying signal at the input of the *VariableLoad* model discussed before. The carrier signal can be for instance a sine wave or a triangle wave, while the active power noise is defined in terms of the expectation value, the standard deviation and the sample period.

The following code fragment illustrates how the *Modelica.Noise* library is used to implement the noise injection signal model with sine wave carrier:

```
partial model BaseClass
  // Parameters to be used by the noise generators
  parameter Real active_mu=0 "Expectation value active power noise";
  parameter Real active_sigma=0.01 "Standard deviation active power noise";

  parameter Real samplePeriod=0.02 "Sample period";

  Modelica_Noise.Blocks.Noise.NormalNoise noise_gen(
    mu=active_mu,
    samplePeriod=samplePeriod,
    sigma=active_sigma)
  Modelica.Blocks.Interfaces.RealOutput y
end BaseClass;

model SineNoiseInjection
  extends BaseClass;
  parameter Real amplitude=1 "Amplitude of sine wave";
  parameter Modelica.SIunits.Frequency freqHz(start=1) "Frequency of sine wav
e";
  parameter Modelica.SIunits.Angle phase=0 "Phase of sine wave";
  parameter Real offset=0 "Offset of output signal";
  parameter Modelica.SIunits.Time startTime=0
    "Output = offset for time < startTime";

  Modelica.Blocks.Sources.Sine sine(
    amplitude=amplitude,
    freqHz=freqHz,
    phase=phase,
    offset=offset,
    startTime=startTime)
equation
  y = noise_gen.y + sine.y;
end SineNoiseInjection;
```

3.3.4 Network Models

A SMIB network model was developed for each of the generation groups described in section 3.3.1. These models follow the inheritance feature of the Modelica language to extend one of two basic partial models, where common network elements and parameters are specified. Figure 3.4 shows the SMIB network case where the load model is stochastic and the generation group has no controls.

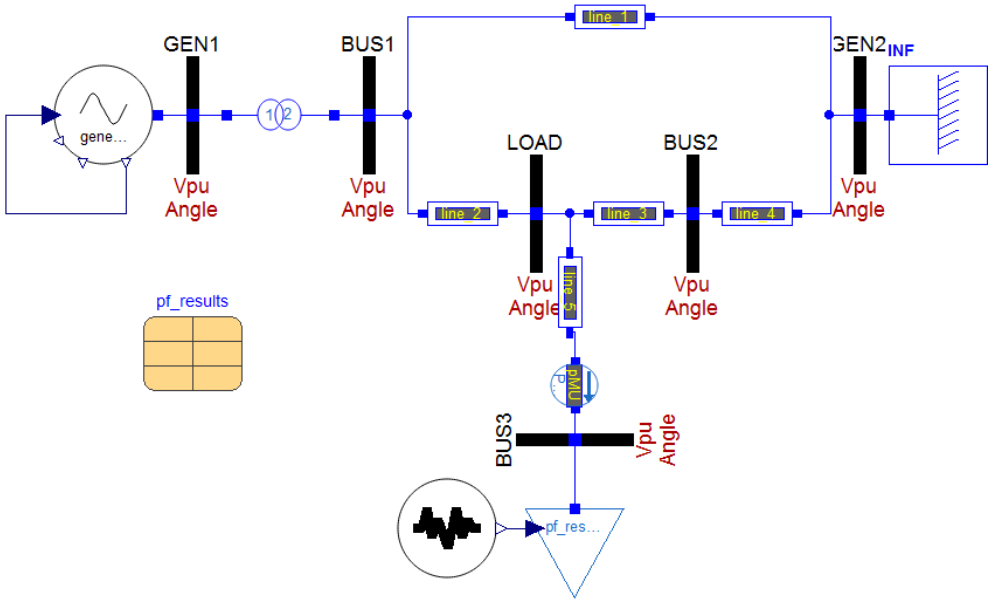


Figure 3.4 SMIB network model with stochastic load and no turbine governor model

The initial voltage magnitude, voltage angle, active power and reactive power values of the generators, the load and the buses are specified by means of the record *pf_results*.

3.4 Multi-Domain Model

A SMIB network model and a governor block model from the *PowerSystemDomain* package can be combined with the physical model of a gas turbine from the

TurboMachineryDomain package. The result of this procedure gives the so-called multi-domain model that can be appraised in Figure 3.5.

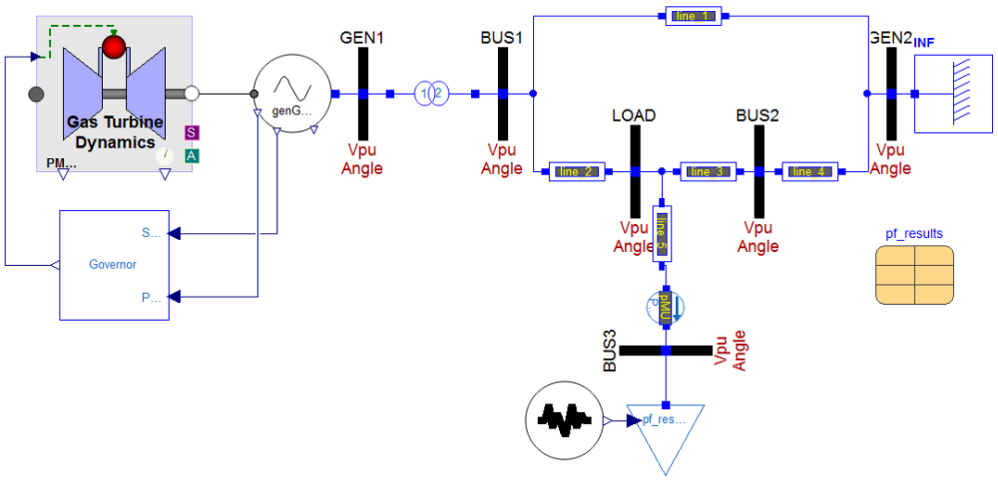


Figure 3.5 Multi-domain SMIB model with stochastic load and GGOV1-based governor model

New generation group sets have been created to allow the connection between the generator and the detailed gas turbine model. Even though they still rely on the previously defined groups of the *PowerSystemDomain* package, they also include an interface block. The function of this new element is to relate the rotational mechanics flange internal variables of the gas turbine model with the generator mechanical power and speed as shown below:

```
model TM2EPConverter
  "Interface between OpenIPSL generators and ThermoPower gas turbine models"
  import Modelica.Constants.pi;
  outer OpenIPSL.Electrical.SystemBase SysData
  parameter Integer Np=2;
  parameter OpenIPSL.Types.ApparentPowerMega S_b=SysData.S_b
    "System base power"
  Modelica.Mechanics.Rotational.Interfaces.Flange_a shaft
  Modelica.Blocks.Interfaces.RealOutput PMECH
  Modelica.Blocks.Interfaces.RealInput SPEED
  Real omega_e;
equation
  omega_e = der(shaft.phi)*Np;
  SPEED = omega_e/(100*pi) - 1;
  PMECH = der(shaft.phi)*shaft.tau/(S_b*1e6);
end TM2EPConverter;
```

Chapter IV

4. SEMANTIC MODELS

This chapter aims to describe the proposed semantic modeling approach for the multi-domain application of this thesis. The presented methodology and the resulting models are intended to meet objective b of Chapter 1.

4.1 Proposed Multi-Domain Semantic Modelling Methodology

The CIM user's group[1] has adopted a mechanism for managing a CIM-based enterprise semantic model. This methodology has been formalized by the authors of [29] and given the name of CIM Layered Architecture. The architecture comprises three layers, namely Information, Contextual and Message Syntax layers, and follows the UN/CEFACT Core Components Technical Specification (CCTS) [30]. An extended version of the layered architecture that can be seen in Figure 4.1 has been proposed in this thesis.

The information layer covers typical electrical industry concepts that are presented in the IEC 61970-301, IEC 61968-11 and IEC 62325-301 standards. Together they constitute the base model that is distributed in UML notation through *EAP* files of the *Enterprise Architect* software of *Sparx Systems*. However, when this layer was first proposed, the possibility of embracing other sources of information to create an extended version of the semantic model was also envisioned. Such a principle can be used to solve the need to develop a CIM-based multi-domain information model. Two possible ways of including the semantics required to represent non-electric power energy systems are as follows:

1. Perform a direct CIM/UML extension to include new classes or class attributes.

[1] See URL: http://cimug.ucaiug.org

2. Follow bridging rules to "merge" CIM with foreign models that are focused on other domains.

The first option makes sense when the number of concepts to incorporate is reduced. This is the case, for instance, of new control schemes that are not in the existing list of CIM standard models of exciters, stabilizers, etc. Nevertheless, when it comes to representing the details of gas turbine dynamics, this is not the most feasible solution. The second option is recommended and a somehow mature foreign standard that has the potential to describe the turbo-machinery domain is ISO 15926. Together with CIM, ISO 15926 can define a harmonized multi-domain semantic model, as indicated in Figure 4.1.

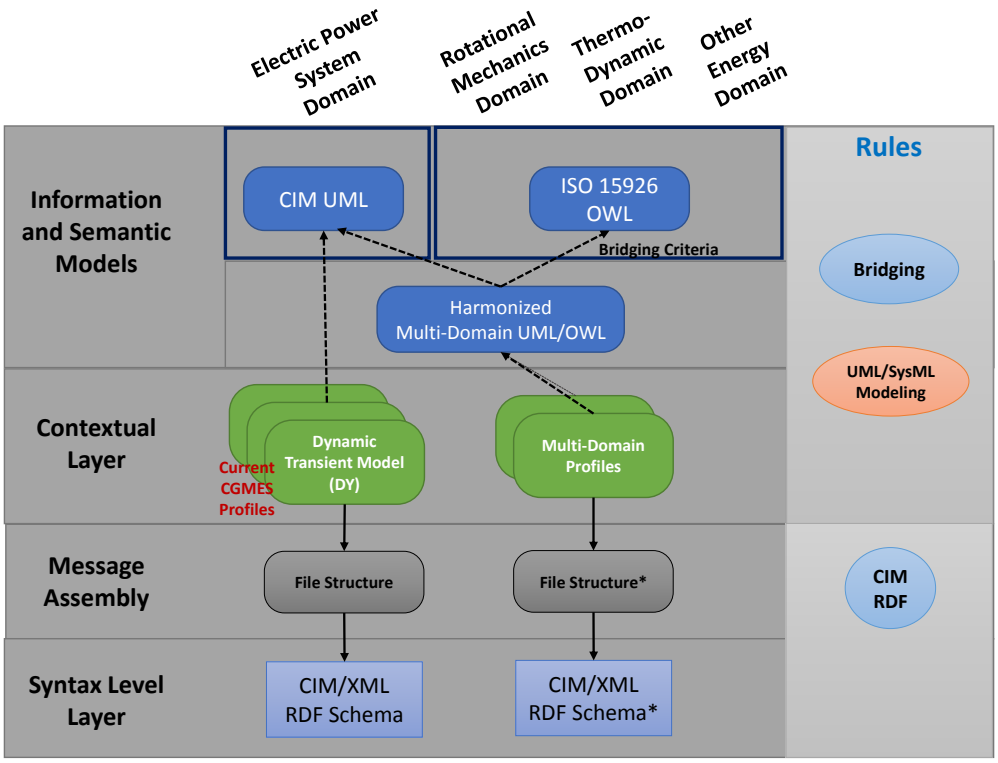


Figure 4.1 Application of the CIM Layered Architecture for Multi-Domain Modelling

There are several challenges in bridging the two models in question. While CIM describes its concepts using UML, ISO uses the OWL semantic representation

standard. In addition, ISO has different modeling rules when compared to CIM. This feature applies to aspects such as connectivity, composition and properties definition of physical objects. The ideal method would be to strictly follow both modelling criteria in parallel. However, this has the potential to become a new thesis topic by itself. A more pragmatic approach has been adopted in this project and will be discussed in section 4.3.

The Contextual Layer comprises a series of profiles that establish subsets of the combined information models contained in the Information Layer. The mission of this layer is to create a logical separation between the pure definition of the semantic model and the multiple specific applications that can be given to it. Some examples of existing CIM profiles that are being utilized in this project are Equipment Model (EQ), State Variables (SV), Steady State Hypothesis (SSH) and the Dynamics Profile (DY). Since the proposed multi-domain semantic model was driven by a specific use case, i.e. the detailed gas turbine – SMIB network data exchange, in this thesis there is no difference between profile and harmonized multi-domain model.

Concrete techniques to implement the information exchange of the information contained in the profiles are given by the rules that the Contextual Layer provides. While this topic is not discussed herein, the rules that define the CIM/XML file creation can form the basis for the formulation of schemas that comprise data from both CIM and ISO 15976.

4.2 Power System Semantic Modeling based on CIM/UML standards

As mentioned in the previous section, the full CIM model is distributed through EAP files. Since the semantic modeling must be consistent with OpenCPS MST 1 described in section 1.2, that file must be able to be imported by the Papyrus tool. Although this is possible through a slight modification in the file headers, it is not possible to retrieve all the information. This is the case of the class diagrams, so it is necessary to draw them manually.

Block A in Figure 1.2 attempts to illustrate the role of Papyrus in the semantic modeling work flow of MST 1. Section 4.2.1 shows an example of the manually drawn CIM Class Diagrams in Papyrus, while Section 4.2.2 explains how SysML is used as an intermediate step to accomplish the expected transformation from UML information to Modelica.

4.2.1 Internal UML and Class Diagrams

The CIM package structure, classes with their attributes, and class relationships are loaded successfully into Papyrus. Figure 4.2 shows a reduced view of the CIM packages together with one example of a Class Diagram.

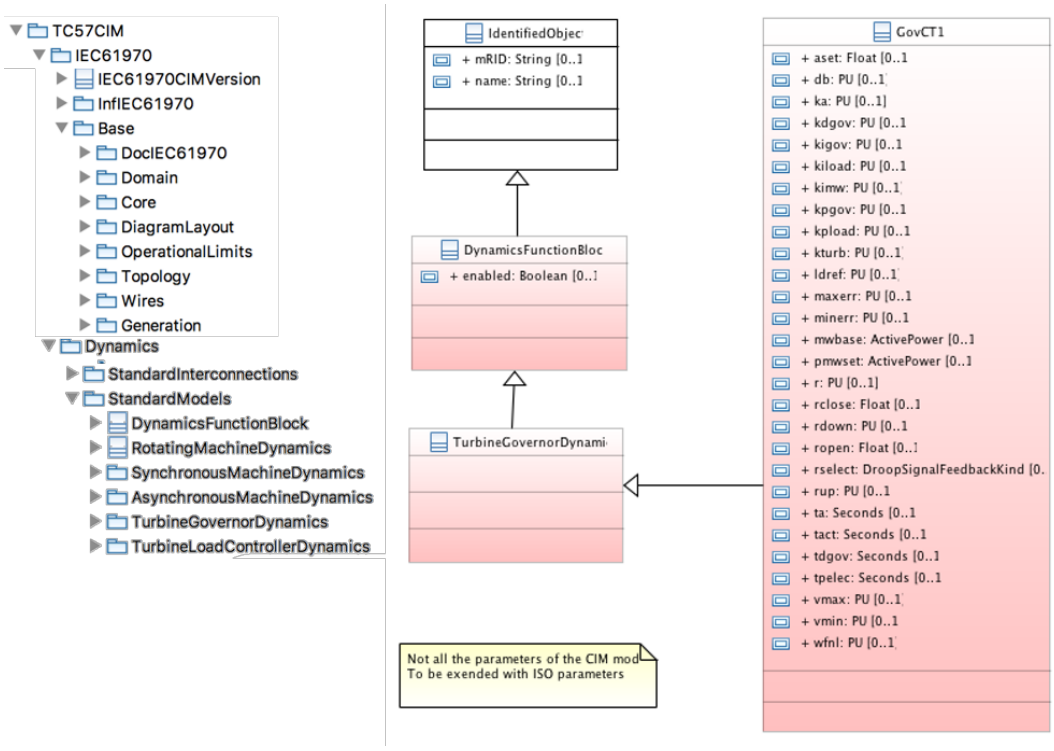


Figure 4.2 CIM UML package structure and Class Diagram example in Papyrus

The example of Figure 4.2 illustrates how 4 classes are combined to provide the semantic representation of a GGOV1-based governor. Although the GovCT1 model constitutes the entire turbine governor group GGOV1, in this case the

class attributes were restricted to only governor parameters. This allows the use of existing CIM semantics to obtain the multi-domain model.

Appendix B includes a diagram of the classes required to represent the SMIB network. Along with the classes that come from the EQ, TP, SV and DY profiles, the *StochasticLoad* class has been added to model the stochastic load behavior. The second diagram shows the classes that are being used to model the controls of the generation groups of Table 3.2.

4.2.2 SysML for Model Transformation

The conversion from CIM/UML model to Modelica can be done through a standard M2M (model-to-model transformation) procedure that was implemented by the OpenCPS coordinator in KTH, Francisco Gómez [9]. This mechanism, which is based on transformation rules, requires the representation of both the CIM classes and the respective Modelica components as SysML blocks.

Figure 4.3 shows a diagram of the SMIB network block containing additional blocks that symbolize the components from the OpenIPSL library.

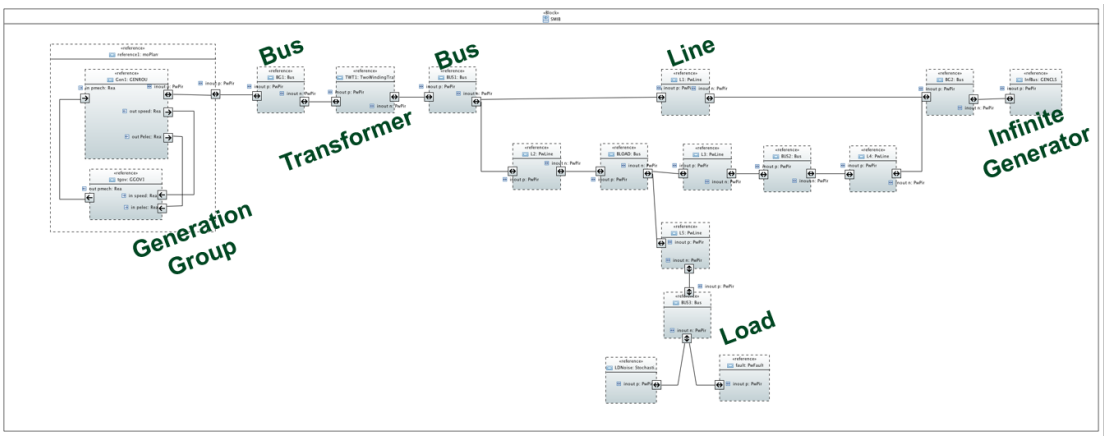


Figure 4.3 SMIB block diagram with OpenIPSL-related component blocks

The example of the figure demonstrates the use of the *connector*, a special type of attribute the SysML blocks contain. It allows the interconnection of blocks that is

required to specify how OpenIPSL components are linked together to create the Modelica SMIB models.

The CIM/XML files parsing method intended to feed the parameters of the OpenIPSL components is not within the scope of this thesis.

4.3 ISO-15926 Semantic Modeling of the Gas Turbine

As discussed in section 4.1, the ISO 15926 standard has been suggested as a way of including the gas turbine semantics. ISO 15926 is an international standard that provides an ontology that is aimed to represent process plant lifecycle data, including facilities of oil and gas production [31]. ISO 15926-2 [32] specifies a stable generic data model with around 200 concepts that enables the exchange and integration of process plant data. Another relevant part of the standard, namely ISO 15926-4 [33], defines a large set of expandable common reference data elements known as the Reference Data Library (RDL) [34].

The ISO 15926 data model was originally specified by means of the EXPRESS language (ISO 10303-11). ISO 15926 Part 8 formalizes a description of the model based on the more common OWL ontology language [35].

Table 4.1 Physical Quantities and Units with CIM and ISO 15926

Class	Unit	Source Information Model
PU		CIM
Boolean		CIM
Float		CIM
Seconds		CIM
RotationSpeed	rad/s	CIM
Area	m2	CIM
Absolute Pressure	Pa	ISO
Absolute Temperature	K	ISO
Density	kg/m3	ISO
Mass Flow Rate	kg/s	ISO
Heat Capacity	J/K	ISO
Specific Enthalpy	J/kg	ISO
Mass Fraction Of B	kg/kg	ISO

A practical approach was adopted in this thesis and consisted of a mapping of the standard concepts to UML classes. This process began with the specification of the data types associated with the class attributes.

Table 4.1 shows how some of the physical quantities defining data types in the Domain package of CIM, can be reutilized in the semantic representation of the gas turbine. However, as can also be seen in the figure, new classes were created to describe quantities that were not covered by CIM. The name of each of the new classes was taken from the equivalent concepts of the ISO 15926-4 *Property* reference data set. A freely available version of the ISO 15926-4 reference data items can be found in the POSC Caesar Association website (see URL: https://www.posccaesar.org/wiki/ISO15926). To be more precise, the data sets can be browsed on the web pages or downloaded as Microsoft Excel worksheets. The units for each of the quantities can be described by class attributes and enumerations, just as it is done in CIM. This means that the resulting ISO-based quantity classes will have at least the attributes *multiplier*, *unit* and *value*. The *multiplier* attribute has as data type the CIM existing or a new *UnitMultiplier* enumeration. The same applies to the *unit* attribute with the *UnitSymbol* enumeration, while *value* has always a *Float* data type.

Table 4.2 Created classes to semantically model the gas turbine

Package	Classes
Basics	Fluid, Flange, Compound, FlangeKind*, FluidKind*
Core*	Physical Object, Tmachinery_Resource*, Matrix*, Matrix_Data*
Heat Transfer	Combustion Chamber, Chamber
Rotating Equipment	Compressor, Axial Compressor, Dynamic Compressor, Expander, Turbo Expander, Turbine, Gas Turbine, Power Generation Gas Turbine
Topology*	TMConnectivityNode*, TMPressureDrop*

* package/class name does not come from ISO 15926.

A total of six packages were created to classify the new classes to be used in the gas turbine semantic model. One of them is Domain which can be viewed as an extension of the same package in CIM. The remaining are shown in Table 4.2.

Physical Object is defined by the ISO 15926-2 standard as a "distribution of matter and/or energy in time and space" [32]. This is the super-class of all the physical concepts borrowed from the standard, from *Fluid* to *Power Generation Gas Turbine*. All the classes in the inheritance relationship path from *Physical Object* were also included. For instance, *Axial Compressor* has an indirectly inherits from *Physical Object*, so that explains why the two intermediate classes (i.e. *Compressor* and *Dynamic Compressor*) were also selected.

Instead of following the formal rules from the standard to express component connectivity, the *Terminal-ConnectivityNode* association relationship from CIM was adopted. However, in the context of the gas turbine *Terminal* is replaced by *Flange* while *ConnectivityNode* is replaced by a new class called *TMConnectivityNode*. Since flanges can be of different types, such as inlet shaft or outlet gas flow, a new enumerator called *FlangeKind* was defined. In addition, the *Tmachinery_Resource* class was created to restrict the relationship between a gas turbine element and the flanges. In this way, the classes *Fluid* and *Compound* are isolated from other classes that inherit from *Physical Object*.

The object naming and identification was implemented by forcing an inheritance relationship between the *Physical Object* class and the CIM *IdentifiedObject* class. CIM also served as a reference for the conception of *Matrix* and *Matrix_Data* classes. This classes are aimed to represent the Look-up tables of the component characteristic maps, by following the same modeling principle of *Curve* and *CurveData* in CIM.

The class attributes were based on the *ThermoPower* components. Both class attributes and relations can be seen in the class diagrams of Figures in Appendix C.

Chapter V

5. STUDIES, SIMULATION AND RESULTS

In this chapter, the simulations and studies that were applied on the equation-based models will be described. An emphasis has been placed on the comparative analysis between the multi-domain model and the model that uses the standard GGOV1 power system domain turbine-governor group. In addition, special attention has been paid to the performance of network models when the load has stochastic behavior. The results of the tests and studies will be shown and discussed in the later sections of this chapter.

5.1 Study and Simulation Cases

This section begins with a description of the identification process of the GGOV1-based turbine model that fits the response of the *ThermoPower* detailed model. The results are necessary to carry out the different comparative studies on the resulting SMIB network models, that are also described in this section.

5.1.1 GGOV1-Turbine Model Identification

The first step in the analysis to be done on the SMIB network models is the identification of the GGOV1 turbine model that is equivalent to the *ThermoPower* model. An open-loop test has been applied for that purpose to the Multi-domain network model as can be observed in Figure 5.1.

The governor has been removed from the SMIB network model and then, a step change on the fuel mass flow rate has been applied on the gas turbine model. Table 3.1 was employed to find the fuel mass flow rate values that give an output mechanical power change from 5 to 8 MW.

Subsequently, a simulation was carried out in *Dymola* with a duration of 100 seconds, where the step change occurred after 30 seconds. The results were saved to proceed with the identification of the GGOV1-based turbine model.

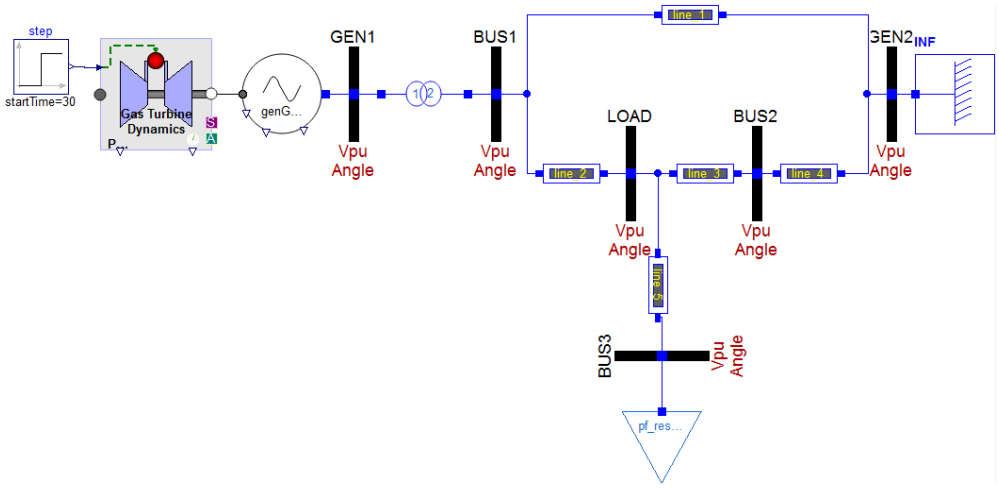


Figure 5.1 Open-Loop test on the Multi-Domain SMIB model

Simulation output data has been imported in MATLAB as a MAT file. Then the system identification `ident` tool has been used to fit a GGOV1-turbine model that suits the reference model. The values of K_{turb} and W_{fnl} were set to 1.5 and 0.15, respectively, as explained in Section 3.2.2. Additionally, the damping factor D_m was set to the typical value of 0. The decision to not consider this parameter is also based on the argument of [21] that states the following: "A speed damping factor can be modeled to influence the temperature limit as a rather gross approximation of the speed dependence of the turbine rating. This is, however, not very accurate". Thus, it has only been required to obtain the values of the parameters of the lead-lag transfer function T_b and T_c, together with the delay transport time T_{eng}.

Another approach that could have been followed involves the use of the tool *RaPId*. For that to be done, the generation of an FMU with the SMIB network model that uses the GGOV1-turbine is required.

From now on, the SMIB network model that contains the physical model of the turbine will be referred to as ***Multi-Domain model***. On the other hand, the SMIB network model using the GGOV1-based turbine model will be referred to as ***Power System-only model***.

5.1.2 Gas Turbine Models Frequency Analysis

A first inspection of the differences between the expected response of the multi-domain model and the power system-only model can be done through a poles and zeros analysis. This study requires a linearization of the physical turbine model around a given operating point, something that can be done automatically from *Dymola* using the Modelica Linear Systems 2 library [36].

To understand the impact of the eigenvalues on the response of each model, the contribution of the poles in the states obtained after linearization has been identified. The results will be displayed in Section 5.2.2.

5.1.3 Load Change Event Simulations

A next step in the comparative study that began in the previous section is to verify the time-domain response of the models under a load change event.

A simulation of 100 seconds was performed on both the multi-domain and power-system models with the same governor model. The active power of the load was increased by 0.2 pu after 30 seconds of simulation, and was set back again to the original value after 20 seconds.

The simulation mentioned above was applied to two scenarios, first with a deterministic load model and later with a stochastic load model.

In the first case, successive simulations were carried out with the objective of evaluating the performance of the models at different operating points. More specifically, the load active power as well as the dispatched power from the generator were increased from 5 to 9.8 MW, in steps of 0.1 MW.

The parameter sweep of the first scenario required the computation of 49 power flow solutions for the initialization of the network models. The solution sets were supplied in the form of records to conform with the description of Section 3.3.4. Appendix D shows part of the Python scripts/code used to automatically generate the Modelica records. The scripts consist of a modified and short version of the toolset used to get the Nordic 44-bus system simulation results published in [11] and [37].

In order to better quantify the time domain response differences of the two models, the settling times were computed. This was only done for the first simulation scenario, when the load model did not include the noise. The respective results are presented in Section 5.2.3.1 and discussed in Section 5.3.

In the second simulation scenario, the load model contained a sinusoidal variation of 0.02 Hz and 0.05 pu of amplitude. Noise has also been injected to ensure a full stochastic model. The sample period of the noise has been set to 0.02 seconds, while the expectation value and the standard deviation were set to 0 and 0.0005, respectively. The resulting plots from these simulations are shown in Section 5.2.3.2 and then discussed in Section 5.3.

5.1.4 Simulations as FMUs on External Tools

In Chapter 1, two master simulation tools (MST) were introduced as the tool chains to fulfill the objectives of the ITEA3 OpenCPS project. Blocks B and C of the MST workflows diagram in Figure 1.2, referred to a stage that involved the import of FMUs, their simulation and the results display. Although the simulation of such FMUs was originally intended for OpenModelica and Papyrus, a test was conducted in three other tools.

A simulation of the multi-domain model with stochastic load behavior was made in Dymola, Modelon FMI Toolbox and FMI Kit for Simulink. The multi-domain model was implemented from three FMUs. For this purpose, the original model was split into three blocks, from which FMUs were generated for Model Exchange in Dymola. These blocks corresponded to the governor, the detailed turbine model and the SMIB network model.

A user guide was prepared to explain the series of steps in the process of generation, import and simulation of FMUs. This contribution has been made available in this report as chapter 6. On the other hand, the results of the test described above can be found in Section 5.2.4.

5.2 Results

The results of the studies and simulations performed on the models are presented in this section.

5.2.1 Model Identification

A GGOV1-based turbine model with one pole and one zero with no time delay was identified. The resulting transfer function is:

$$g_4(s) = K_{turb}\frac{1+T_c s}{1+T_b s} = 1.5 \cdot \frac{1+0.11514s}{1+0.14101s} \tag{5.1}$$

The same step change on the fuel mass flow rate was applied on both the reference Multi-domain model and the Power System-only model without the governor. Figure 5.2 shows the output mechanical power plots from the turbine components of the models.

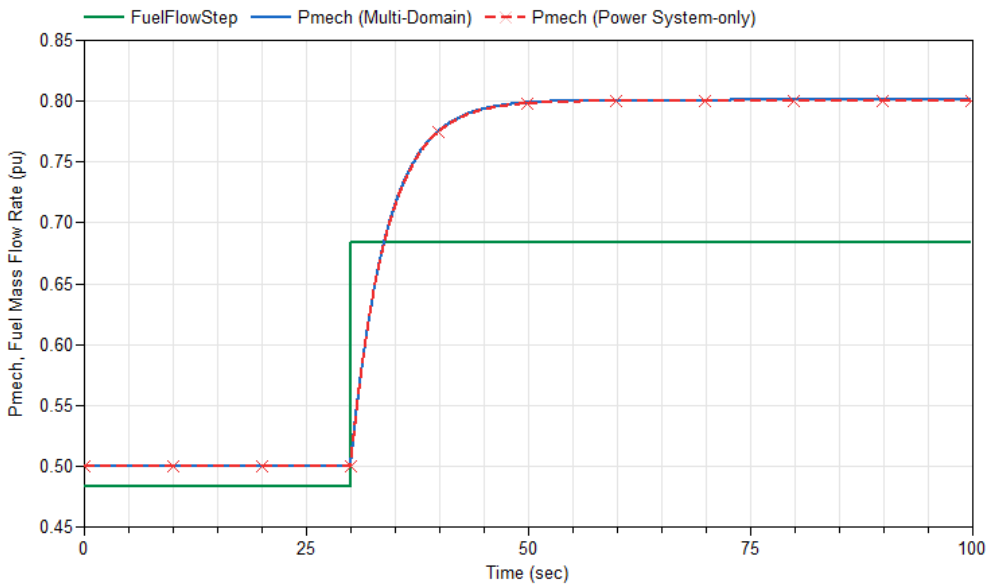


Figure 5.2 Open-Loop test to verify fidelity of the identified model

5.2.2 Frequency-Domain Analysis

The gas turbine systems can be described in state space representation as:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{5.2}$$

In the case of the GGOV1-based turbine system of the Power System-only model, the system vectors and matrices are defined as:

$$u: valve_{position} \qquad y: P_{mech} \qquad x = \begin{pmatrix} g_4(s).x \\ gasFlowActuator.y \end{pmatrix}$$

$$\boldsymbol{A} = \begin{pmatrix} -7.0917 & 11.1226 \\ 0 & -0.25 \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} 0 \\ 0.25 \end{pmatrix}^{-1} \qquad \boldsymbol{C} = \begin{pmatrix} 1.83462x10^6 \\ 12.8066x10^6 \end{pmatrix} \qquad \boldsymbol{D} = 0$$

As can be easily found from the system state space equation there are only real eigenvalues. They are shown on Table 5.1 together with their contribution on the identified systems states.

Table 5.1 Real eigenvalues of gas turbine in the power system-only model

Eigenvalue	T(s)	Contribution to states	
		State	Contribution (%)
$p_1 = -7.0917$	0.1410	$g_4(s).x$	100
$p_2 = -0.25$	4.0	$g_4(s).x$	61.9
		$gasFlowActuator.y$	38.1

The system has only a zero $z_1 = -8.6851$ with $T(s) = 0.1151$

A linearization has been performed on the detailed gas turbine system of the Multi-Domain model at initial time. The state vector is as follows:

$$x = (CC.fluegas.p, \quad CC.fluegas.T, \quad CC.fluegas.X_1, \quad CC.fluegas.X_2, \quad CC.fluegas.X_3, \quad CC.fluegas.X_4, \quad CC.fluegas.X_5, \quad CC.T_m, \quad gasFlowActuator.y, \quad speedSource.\phi)^{-1}$$

System vectors B and C are now defined as follows:

$$\boldsymbol{B} = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 0)^{-1}$$

$$\boldsymbol{C} = (78.8602, \quad 20707.3, \quad 3.93903x10^7, \quad 2.91764x10^7, \quad 7.29113x10^7, \quad 3.16295x10^7, \quad 4.45559x10^7, \quad 0, \quad 0, \quad 0)$$

Matrix A is given by:

$\boldsymbol{A} =$

-1248.12	-397838	-8.62257e+008	-6.90678e+008	-1.53154e+009	-6.26937e+008	-9.84928e+008	0.305572	2.6117e +008	0
-0.3563	-1110.34	-318948	-255481	-566516	-231903	-364324	0.00048293	392099	0
-5.18801e-006	1.53877e-011	-963.175	2.194e-011	4.86509e-011	1.99152e-011	3.12871e-011	0	-39.0958	0
-6.33981e-009	1.35169e-013	6.15453e-012	-963.175	1.09317e-011	4.47489e-012	7.03012e-012	0	-0.0477758	0
2.73763e-006	4.93413e-014	-1.64883e-011	-1.32073e-011	-963.175	-1.19885e-011	-1.88341e-011	0	20.6302	0
3.38233e-006	-2.58546e-012	-1.52838e-012	-1.22425e-012	-2.71472e-012	-963.175	-1.74582e-012	0	25.4885	0
-9.25613e-007	-3.56957e-011	-1.55238e-011	-1.24348e-011	-2.75735e-011	-1.12872e-011	-963.175	0	-6.97527	0
0	0.05	0	0	0	0	0	-0.05	0	0
0	0	0	0	0	0	0	0	-0.25	0
0	0	0	0	0	0	0	0	0	0

The system has also only real eigenvalues as it is shown in Table 5.2.

Table 5.2 Real eigenvalues of gas turbine in the multi-domain model

Eigenvalue	T(s)	Relevant contribution to states	
		State	Contribution (%)
$p_1 = -1.5608\text{x}10^3$	0.0006	$CC.fluegas.p$	99.9
$p_2 = -9.6318\text{x}10^2$	0.0010	$CC.fluegas.T$	97.2
$p_3 = -9.6318\text{x}10^2$	0.0010	$CC.fluegas.T$	75
		$CC.fluegas.X_5$	12.2
$p_4 = -9.6318\text{x}10^2$	0.0010	$CC.fluegas.T$	94.8
$p_5 = -9.6318\text{x}10^2$	0.0010	$CC.fluegas.T$	86.4
		$CC.fluegas.X_4$	5.5
$p_6 = -9.6318\text{x}10^2$	0.0010	$CC.fluegas.T$	99.4
$p_7 = -7.9769\text{x}10^2$	0.0013	$CC.fluegas.p$	99.9
$p_8 = -0.25$	4.0000	$CC.fluegas.p$	99.6
$p_9 = -0.05$	20.0000	$CC.T_m$	100
$p_{10} = 0$	---	$speedSource.\phi$	100

Finally, the gas turbine of the multi-domain model has the following zeros:

Table 5.3 Zeros of the gas turbine in the multi-domain model

Zero	Amount	T(s)
$z_1 = -9.6436x10^2$	1	0.0010
$z_i = -9.6318x10^2$	4	0.0010
$z_6 = -6.4415x10^2$	1	0.0016
$z_7 = -0.05$	1	20.0000
$z_8 = -6.651x10^{-14}$	1	$1.503\,x10^{13}$

The poles and zeros plots of the gas turbine systems from both models can be observed in Figures 5.3 and 5.4.

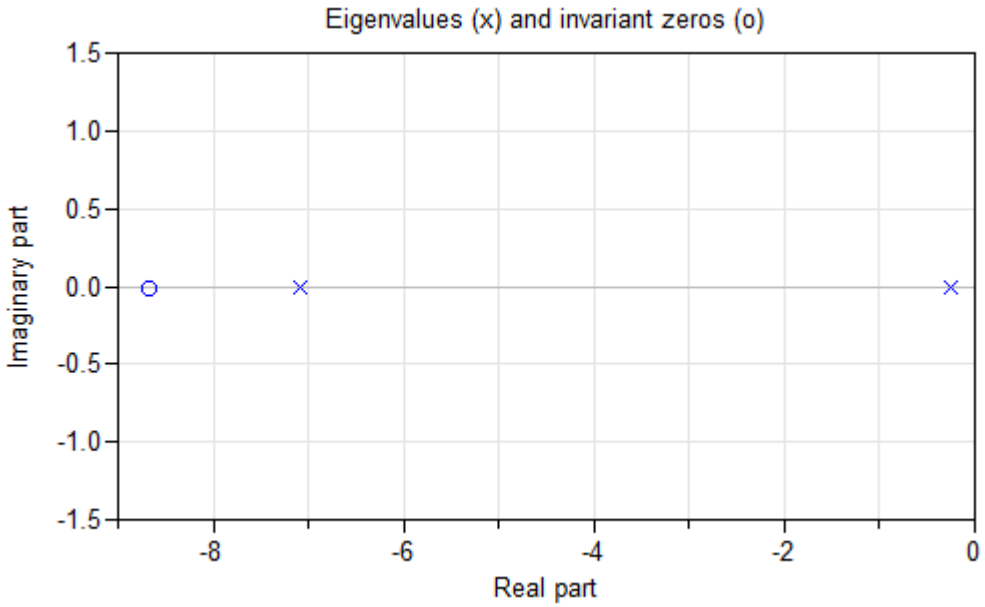


Figure 5.3 Poles and zeros of gas turbine in the power system-only model

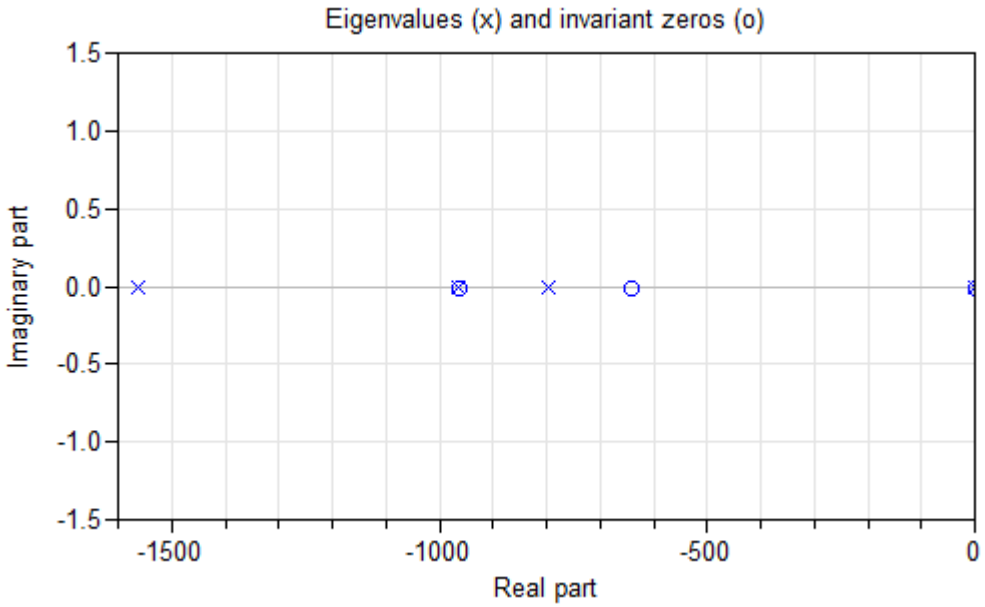


Figure 5.4 Poles and zeros of gas turbine in the multi-domain model

5.2.3 Time Response to Load Change

The governor was added to the Multi-Domain and Power System-only models to evaluate their time response to a load change. The applied test was presented in Section 5.1.3 and the governor parameters can be found on Appendix A. It is worth to mention that the limiters were not included as part of the control logic. This section shows the results of the time response simulations.

5.2.3.1 Response without Stochastic Load

Figure 5.5 shows a plot of the mechanical power delivered by the gas turbine components, when both models under comparison have no stochastic load. A zoomed section was included on the figure to better estimate the influence of the load modeling in the mechanical power response.

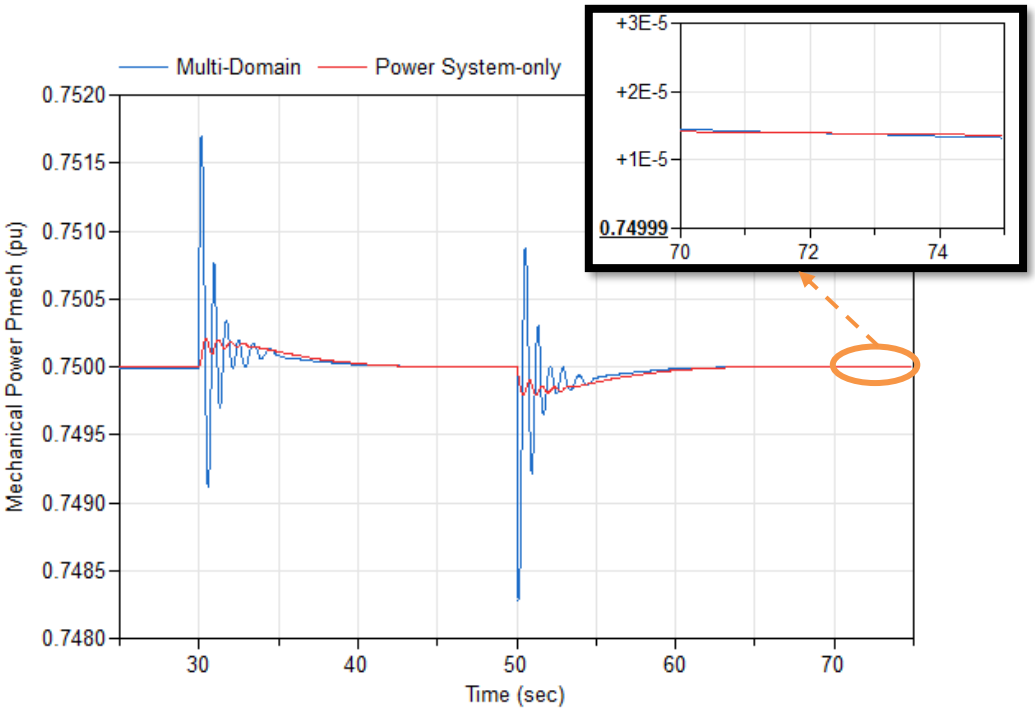


Figure 5.5 Mechanical Power Response Comparison when the models have no stochastic load

Special attention was also given to the response of the system frequency and the electrical power of the generator. The corresponding plots can be appreciated in Figures 5.6 and 5.7, respectively.

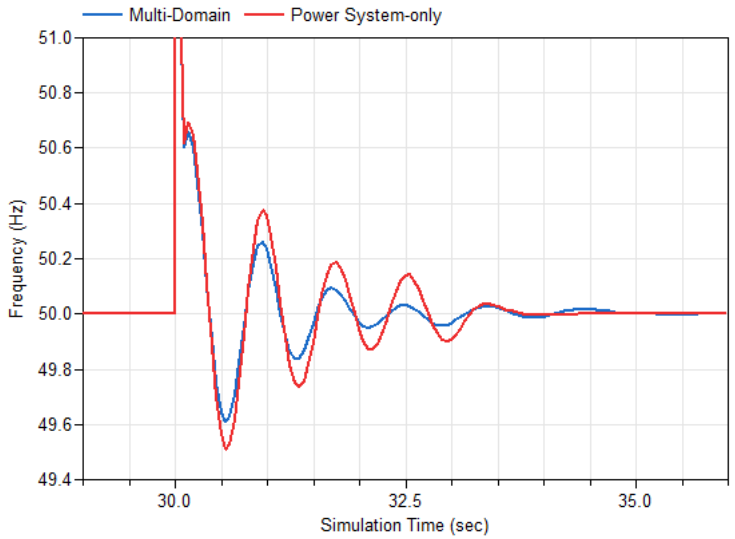


Figure 5.6 Frequency Response Comparison when the models have no stochastic load

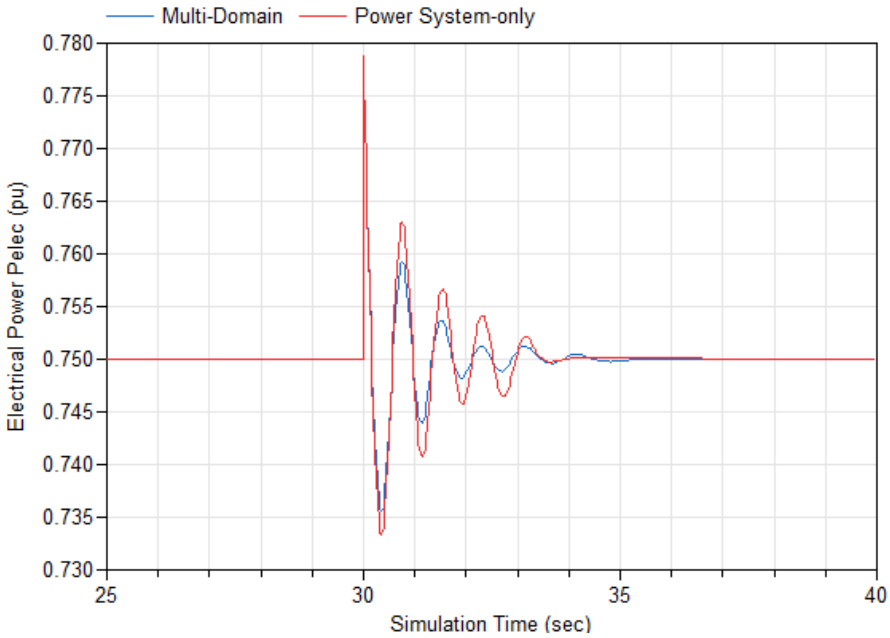


Figure 5.7 Electrical Power Response Comparison when the models have no stochastic load

Then the simulation was repeated several times to measure the settling times, at different operating points of the generator. Figure 5.8, first shows a plot of the calculated settling times in both multi-domain and power system-only models as a function of the load/generator active power. The plot at the bottom of the figure shows the difference between the settling times obtained for the power system-only model and the settling times of the multi-domain model.

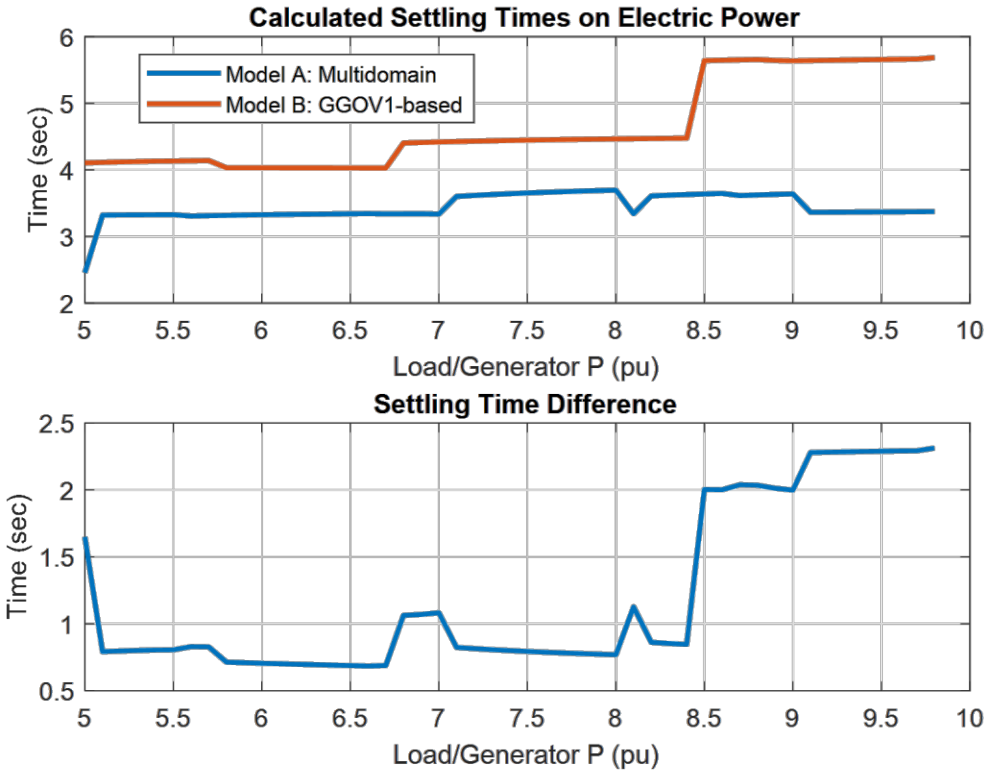


Figure 5.8 Calculated settling times in the electrical power response when the models have no stochastic load

The simulations were performed with the variable step DASSL solver and a tolerance of $1x10^{-4}$.

5.2.3.2 Response with Stochastic Load

The load change test was then applied to the models with the stochastic model of the load described in Section 5.1.3. Figures 5.9, 5.10 and 5.11 show the time

responses of the turbine mechanical power, the system frequency and the generator electrical power, respectively.

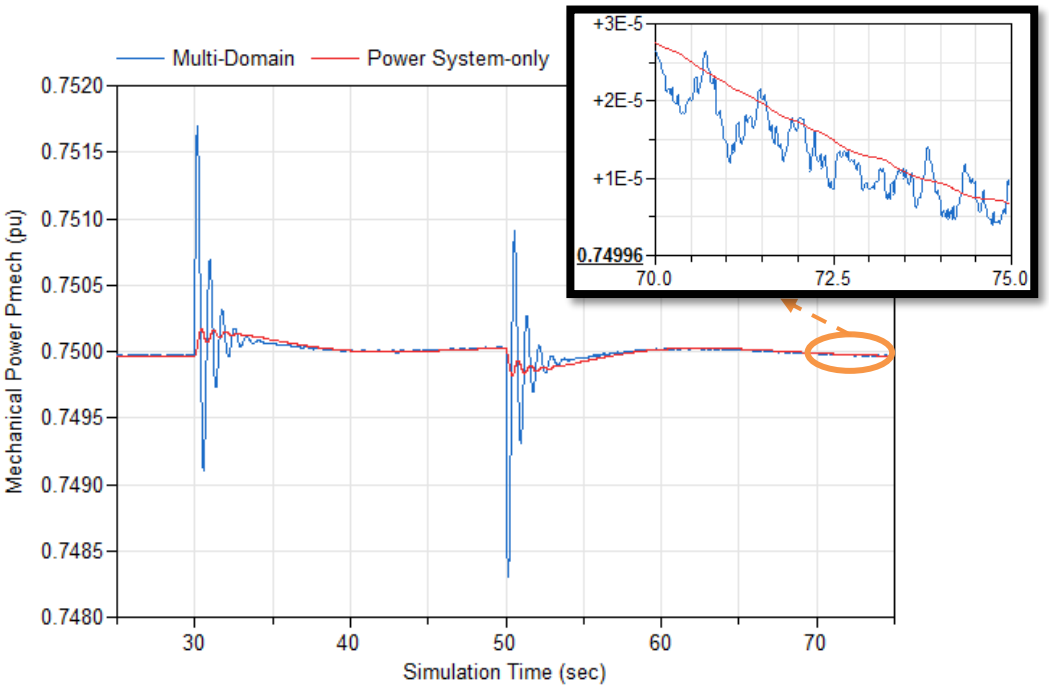


Figure 5.9 Mechanical Power Response Comparison when the models have stochastic load

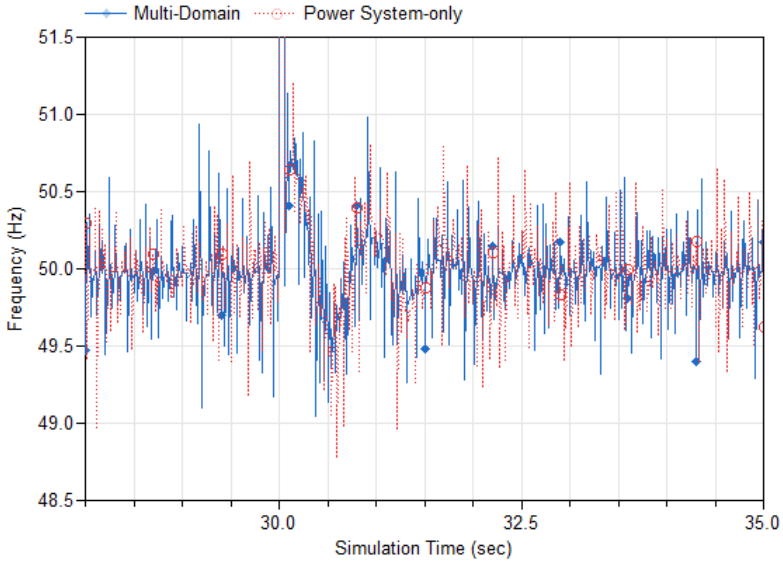


Figure 5.10 Frequency Response Comparison when the models have stochastic load

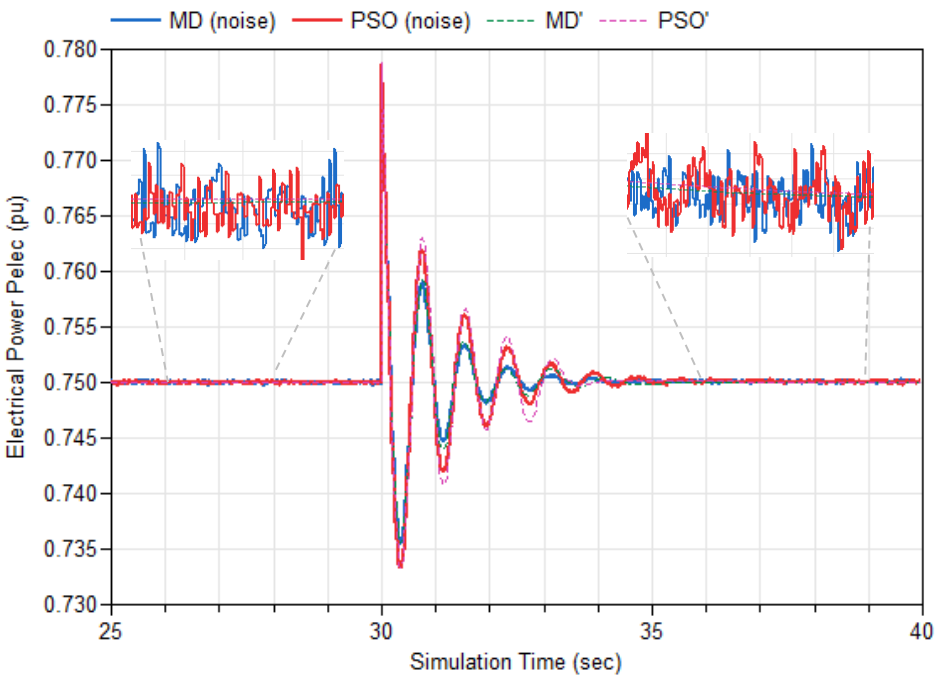


Figure 5.11 Electrical Power Response Comparison when the models have stochastic load

It is recommended to use fixed step solvers with the same sample period of the noise and the appropriate tolerance adjustment. However, none of the available fixed step solvers in Dymola worked. The DASSL solver was used instead with the interval length set to the sample period of the noise.

It is also worth to mention that the frequency measurement (see plots of Figures 5.6 and 5.10) was possible thanks to a pseudo-PMU component from the OpenIPSL library.

5.2.4 Simulation Results from FMU Models

The SMIB network model with stochastic load, the governor and the physical gas turbine model were exported from Dymola as FMUs. Then they were imported and simulated in Dymola, and Simulink as described in section 5.1.4. Figures 5.12 shows the electric power response difference at the beginning of the simulation

and during the load change event, respectively. The used solvers are displayed in the legend of the plots.

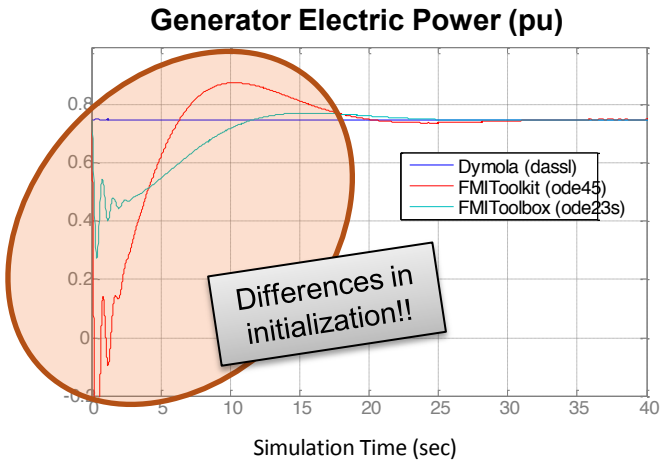


Figure 5.12 FMU simulation initialization differences

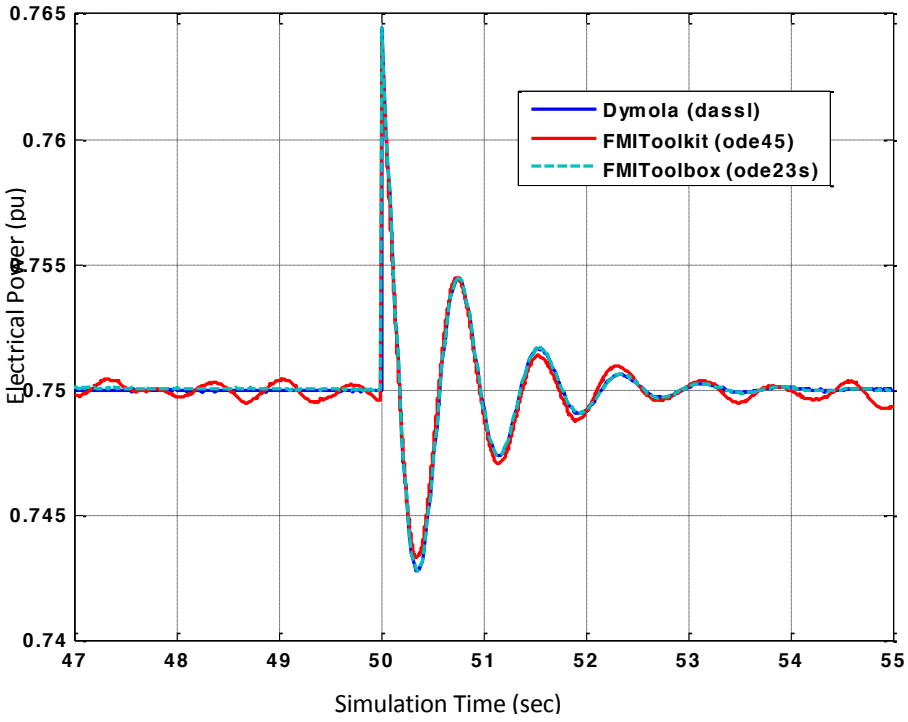


Figure 5.13 Electrical Power response from FMU simulation in several tools

Several issues were however faced during the FMU simulation and will be reported in the discussion section.

OpenModelica was also tested in the procedure that involved the import and simulation of FMUs but, it was not possible to compile the FMUs once they were imported. A simulation of the original multi-domain model without noise was anyway performed to verify that OpenModelica was compliant with both OpenIPSL and ThermoPower components. The resulting plots are available in Appendix E.

5.3 Discussion

The first evidence of the differences between the GGOV1-based turbine model and the physical turbine model can be found in the frequency analysis results from section 5.2. The higher number of poles and states identified in the linearized model of the ThermoPower gas turbine model is a proof of more complexity.

The information reported in Table 5.2 is particularly useful when it comes to giving a better physical explanation to the behavior of the model. First, the list of states allows to appreciate the relevance of the heat addition process to the gas turbine dynamics. Seven out of nine states were related to thermodynamic properties in the boundaries of the combustion chamber (denoted as CC in the state vector). However, since six poles are getting cancelled with zeros, there are only three poles that deserve special attention.

One of those poles is the one that comes from the fuel system actuator transfer function, which is also present in the simpler GGOV1-based model. As it can be seen in Table 5.2, the other two poles have a high contribution in the pressure of the flue gases at the exhaust of the combustion chamber. From the theory, it is possible to remember that the thermodynamic state of a gas is determined by two properties such as pressure and temperature, in addition to the molecular composition. The pressure of the flue gases (i.e. the relevant state in the detailed

gas turbine model) is directly linked to the Brayton Cycle and to the component operation characteristics.

Figures 5.3 and 5.4 together with Tables 5.1 and 5.2 lead to another significant finding. In general, the explicit model from ThermoPower provides a higher bandwidth resolution in behavior modeling that is not present in the GGOV1-based model. Therefore, this shows how a physical model will be more suitable for transient stability studies (e.g. fault analysis).

The effects on the time response of the models can be first examined in figures 5.5 to 5.7. The load change event influence on the system frequency, which is measured closed to the load bus, can be seen in Figure 5.6. Even if it is for a short time (around 2 sec), the frequency experiences a maximum deviation of up to 1 Hz.

The GGOV1-based turbine model is not dependent on the shaft speed and therefore, the changes on the mechanical power of Figure 5.3 are due to the governor response. This is however not the case of the ThermoPower turbine model. That explains why the model produces an additional oscillatory behavior on the mechanical power that cannot be observed in the GGOV1-based turbine model response.

The electrical power can be used to examine the impact of the gas turbine model response on the generator (see Figure 5.7). Nevertheless, it is important to keep in mind that the electrical power is also directly influenced by the speed. The settling times of this variable response were calculated for different values of the load/generator power and then plotted in Figure 5.8. The results show an over-oscillation in the variable response when the GGOV1-based turbine model is employed. Although the settling times difference between the two models' response keeps originally constant, an increase is obtained for active power values greater or equal than 0.85 pu. It has been found that the cause of this performance is the saturation of the fuel actuator limiter in the GGOV1-based turbine model.

As can be appreciated in Figure 5.11, the inclusion of the stochastic behavior of the load yields to more realistic results. This is especially convenient in phasor time simulations performed under the context of cyber physical systems. It is important to keep in mind the need to produce simulation results that better resemble the system's measurements.

The comments on the previous responses remain valid when the load is replaced with the stochastic model. However, the zoomed area in Figure 5.9 provides additional useful information that is worth to highlight. The correct propagation of the stochastic behavior generated by the load model requires suitable models for the different components of the power system. This is clearly not the case of the GGOV1-based gas turbine model, as it can be realized when the mechanical power response from Figure 5.9 is compared to that of Figure 5.5.

Finally, the simulation with FMUs seems to be still a challenge. Several difficulties were found in the different tools that were tried. Although it is always possible to generate, import and simulate FMUs from Dymola, this is not the case of the remaining tools.

In OpenModelica, for instance, it has not been possible to compile the models with the FMUs. The log shows a compilation error when the default `gcc` compiler is used. An attempt to change the compiler to the Microsoft one was not successful, since the minimalistic GNU sub-system used by Open Modelica cannot see the environment variables required by the compiler.

MATLAB/Simulink often crashed when the FMU toolboxes were used to simulate the multi-domain system. In many cases the simulation was stopped due to an error that was always by-passed by the solvers in Dymola. This initialization problem relies on the ThermoPower model and is reported as a "*logarithm of a negative number*" error.

Chapter VI

6. MULTI-DOMAIN MODELLING AND SIMULATION WITH FMUS

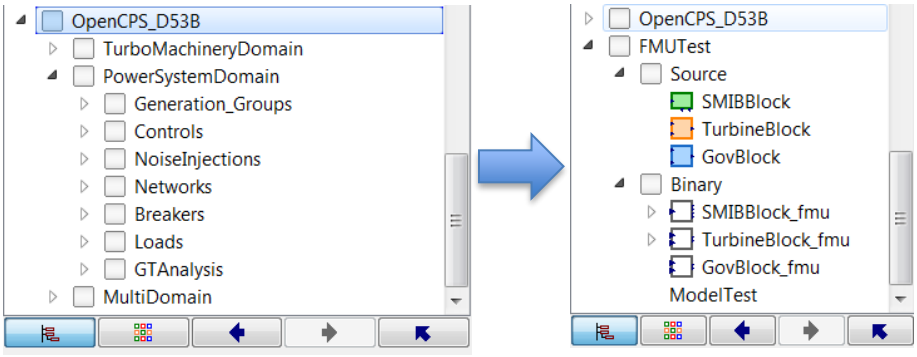
The following pages are dedicated to the description of the steps necessary to: a) export the FMUs of the multi-domain model from Dymola, b) import them into different tools such as Simulink toolboxes, and c) setup and simulate the FMU-based models.

6.1 Requirements

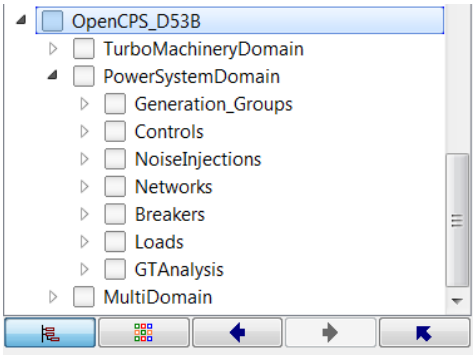
- Dymola version 2016 FD01 or above
- Visual Studio 2008-2013 / Visual C++ 2008-2013 Express Edition (9.0-12.0)
- FMI Toolbox 2.3.1 or above
  MATLAB/Simulink (check for compatibility with FMI Toolbox at http://www.modelon.com/products/fmi-tools/ … fmit-toolbox-for-matlabsimulink-release-information/)
- OpenIPSL library version 1.0.0 (available at https://github.com/SmarTS-Lab/OpenIPSL)
- ThermoPower library version 3.1(available at https://github.com/casella/ThermoPower)
- Modelica_Noise library version 1.0 Beta 1. Only required if Dymola version 2016 is used. It can be downloaded from https://github.com/DLR-SR/Noise/tree/master/ … Modelica_Noise%201.0%20Beta.1
- Distributed package with OpenCPS D5.3b models.

6.2 FMU Export from Dymola

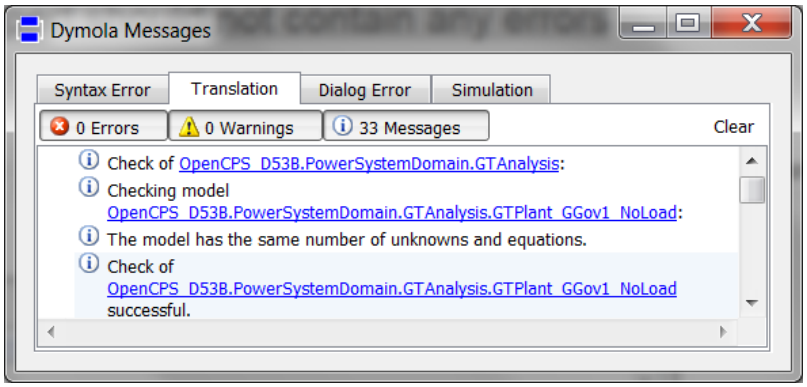
The following steps will guide you in the process of generating FMUs from Dymola. To this end, you will create a new package. Starting from the OpenCPS_D53B.mo library, the goal is to help you get from the screenshot of the library on the left, to the screenshot on the right. The latter includes the required FMUs in a separate package that uses models from the OpenCPS_D53B.mo library.



1. Open the OpenCPS D5.3b package in Dymola. Once the file is loaded, the package explorer should look as follows:



2. Make sure that the package models meet all the dependencies. An easy way to do this is to perform a normal syntax check on the root package. The translation log of the Dymola Messages window should not contain any errors:



3. Create a new package called *FMUTest* with sub-packages *Source* and *Binary*.

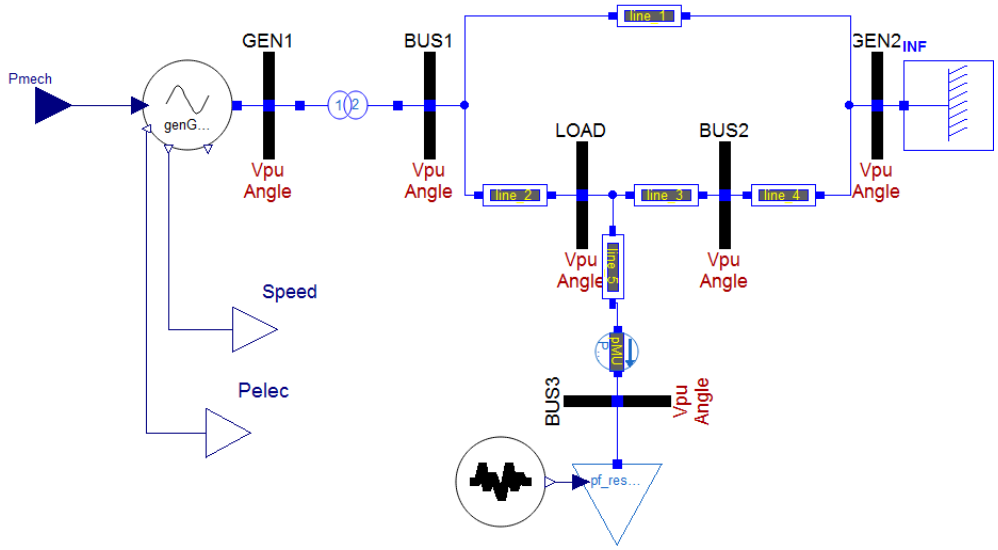
4. Go to *OpenCPS* \ *PowerSystemDomain* \ *Generation_Groups* \ *SMIB*. In the *Records* package, adjust the generator parameters. Also modify the parameters of the AVR and PSS, if you want to include the controls. The parameters have been specified using the *MachinePars1*, *AVRPars* and *PSSPars* records. The default values can be freely modified but it is recommended to duplicate the original records. The *Generator* model implements a generator without controls, while the *Generator_AVR* and *Generator_AVR_PSS* models incorporate the exciter and the stabilizer, respectively. Be sure to refer to the records of your preference by using the *Change Class* option in the context menu of the records icons.

5. Go to *OpenCPS* \ *PowerSystemDomain* \ *Networks* \ *SMIB* \ *NoiseOnLoad* and duplicate the SMIB class inside the Source package. Assign the new class the name of *SMIBBlock*. Notice that in this case the network model contains a generator with no controls. If you want to add the exciter, use the *SMIB_AVR* instead. To include the stabilizer the model to be duplicated should be *SMIB_AVR_PSS*.

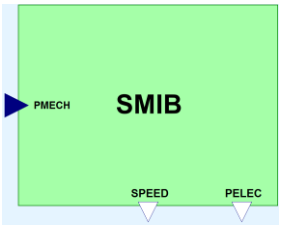
6. Modify the value of the network components' parameters, as needed. The variable *pf_records* must point to a record containing the correct initialization values for power [P0, Q0] and voltage [V0, A0] of each system bus. These values can be obtained through an external tool that can solve power flows (e.g. PSS/E, PSAT, PowerFactory, etc.). Multiple record examples with different power flow solutions have been included in *OpenCPS* \ *PowerSystemDomain* \ *Networks* \ *SMIB* \ *Records*.

   Do not forget to also adjust the parameters of the noise model!

7. The network block interfaces must now be created. To do this, first remove the connection between *Pm0* and *Pmech* that initially exists in the generator. Then add a *RealInput* interface for the mechanical power of the generator (Pmech) and two *RealOutput* interfaces, one for speed and one for electrical power (Pelec). The model should look like the figure below:

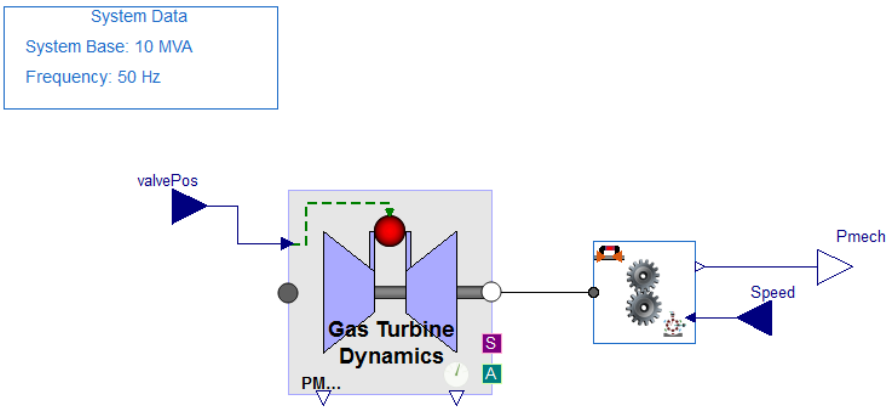


Optionally the block can be edited in the icon view to give it an appearance like the following figure:

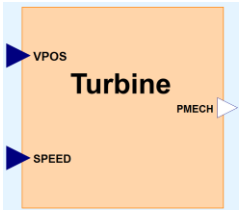


8. Insert two new models into the *Source* package. One of them which represents the turbine can be called *TurbineBlock*, while the other could be called the *GovernorBlock* and will contain the control logic. Drag into *TurbineBlock* your gas turbine model of choice. The ThermoPower model, for instance, is called *ThPowerSSGT* and is located in *OpenCPS* \ *TurboMachineryDomain* \ *GTModels*. This specific model does not directly calculate the mechanical power of the turbine, and therefore requires integration with the TM2EPConverter block to be found in *OpenCPS* \ *MultiDomain* \ *Common*.

9. If the ThermoPower gas turbine model is used, connect its shaft flange to the *Flange_a* interface of the TM2EPConverter. Then insert two *RealInput* interfaces, one for the valve position (*valvePos*) and another one for the speed. Add also one *RealOutput* interface for the mechanical power (*Pmech*). Finally, make a copy of the System Data block from the *SMIBBlock* model into *TurbineBlock*. The model should look like the figure below:

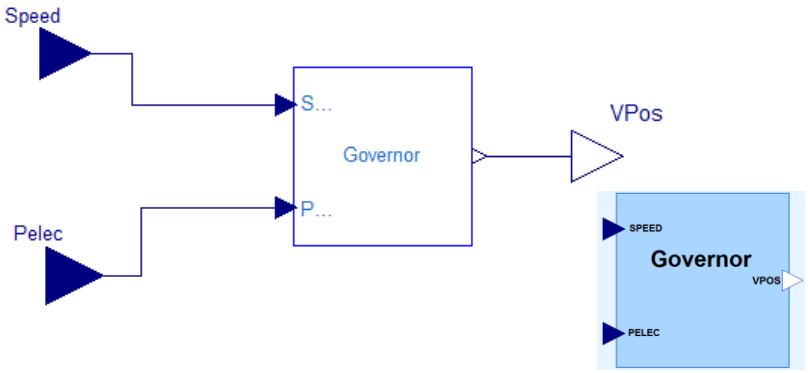


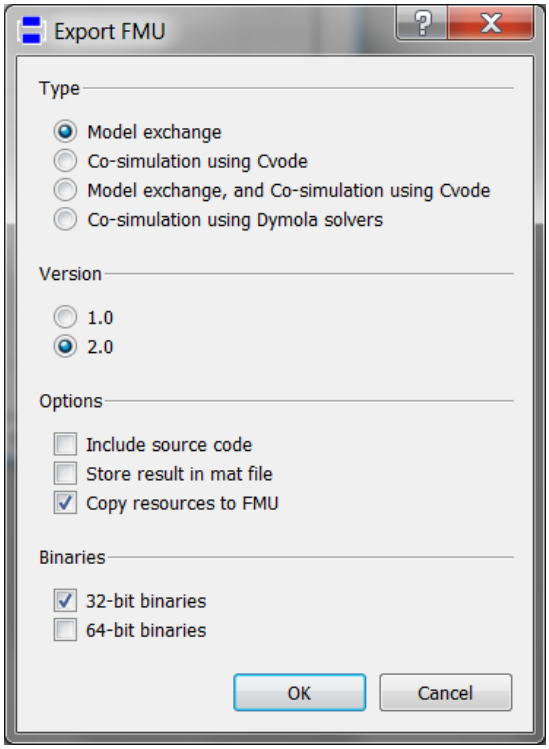
This block can also be edited in the icon view to give it the following look:



10. Fill the *GovernorBlock* with an appropriate control logic and provide the values of its parameters. In the OpenCPS package a complete implementation of the GGOV1 turbine governor model was included under the same name in *OpenCPS* \ *PowerSystemDomain* \ *Controls* \ *GGOV1* \ *Original*. If you want to use this model, note that you must first remove the turbine block to leave only the control logic. In addition, you need to create a new *RealOutput* interface for the exhaust temperature in your turbine model. In this example, a simplified governor model is used that does not include temperature and acceleration limiters. This model is called *PIDGovernor2B* and can be found in *OpenCPS* \ *PowerSystemDomain* \ *Controls* \ *GGOV1* \ *Simplified*.

11. Create the input and output interfaces that are needed to connect this block to the remaining two. Use the following image as a reference of the final appearance of the block.





12. Export each of the blocks as an FMU by clicking on the Translate / FMU option. Choose the model exchange type and keep only the 32-bit binaries option ticked. Alternatively, use the following script:

```
translateModelFMU("FMUTest.Source.SMIBBlock", false, "", "2", "me", fal
se);
```

13. If the translation is successful the command window should look like follows:

```
translateModelFMU("FMUTest.Source.SMIBBlock", false, "", "2", "me", false);
 = "FMUTest_Source_SMIBBlock"
```

```
translateModelFMU("FMUTest.Source.TurbineBlock", false, "", "2", "me", false)
;
 = "FMUTest_Source_TurbineBlock"
translateModelFMU("FMUTest.Source.GovBlock", false, "", "2", "me", false);
        = "FMUTest_Source_GovBlock"
```

The files will be created in the working folder of Dymola. Change their names to *SMIBBlock.fmu*, *TurbineBlock.fmu* and *GovernorBlock.fmu*.

6.3 FMU Import in Different Tools

14. If you have not already done so, include the *$DymolaDir \ Mfiles \ FMIKit_for_Simulink* folder with all its sub-folders in the MATLAB path. Initialize the FMI Kit for Simulink Toolbox by invoking the following script in the command window:

    ```
    FMIKit.initialize()
    ```

    Unless you append this line to the startup script, you should type this command every time you start MATLAB. In case of errors consult the toolbox user 's manual.

15. Similarly, add the *$ProgramFiles \ Modelon \ FMI Toolbox 2.3.1* folder and its sub-folders to the MATLAB path.

16. Create a new Simulink model. In the Simulink Library Browser look for the FMI Kit library if you want to work with *FMIKit*. Otherwise, go to the FMI library.

17. In OpenModelica, create a new package called *FMUTest*. Create a model inside this package (you can call it *ModelTest*).

18. Create a model ModelTest inside of the *FMUTest \ TestBinary* package.

19. Insert 3 instances of the appropriate FMI block into the Simulink model. FMI Kit provides one common block regardless of whether the FMU is CS or ME. In the case of the Modelon FMI Toolbox, the FMU ME block should be used.

20. Save the Simulink model and ***load the files*** that corresponds to each of the blocks.

    With a ***FMI Kit*** block, double click on the block and then press the button Load.

    With a ***Modelon FMI Toolbox*** block, the button to be clicked is Load FMU…

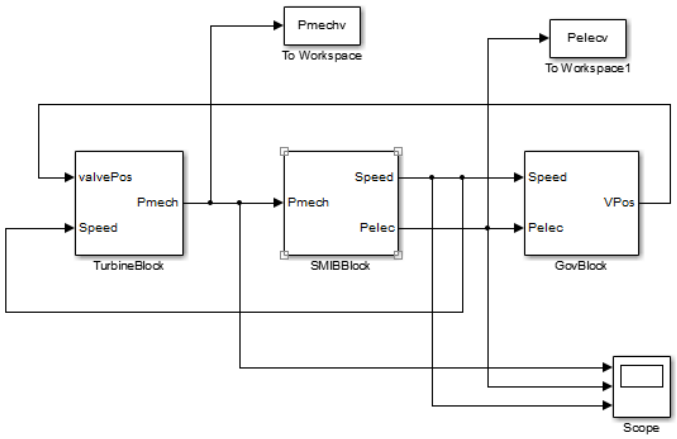
21. Repeat step 20 with Dymola / OpenModelica:

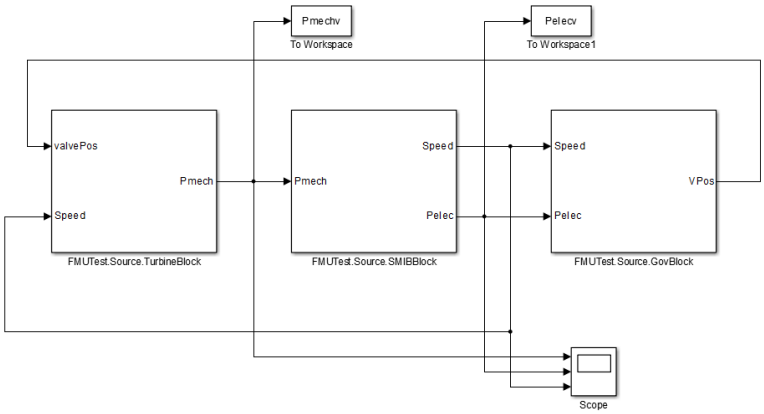
    In ***Dymola***, go to File ► Import ► FMU… Set the preferred type to Model Exchange and then browse to find the FMU file. In the package browser, drag the loaded FMU components to the Binary package.

    In ***OpenModelica***, go to FMI ► Import FMU. Tick the Generate input / output connector pins checkboxes.

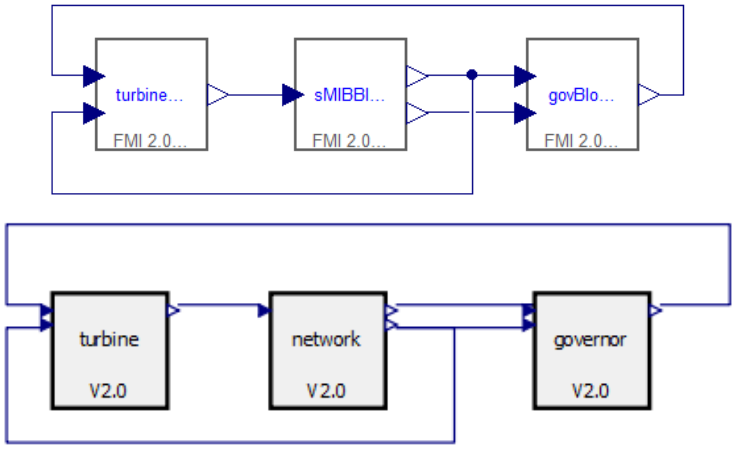
    Drag & drop the FMU blocks inside the *ModelTest* model. In OpenModelica, this is however not possible. Instead, you must duplicate the class and specify the path as the *FMUTest* package.

22. If necessary, adjust the parameters and/or outputs of the block in the configuration window that appears when double-clicking on the block.
23. Add *Scope* and/or *ToWorkspace* blocks to your Simulink model.
24. Connect the blocks in your models. The following pictures provide an example of how the models could look like with FMIKit and Modelon FMI Toolbox blocks, respectively.





In Dymola and OpenModelica, the models could look like follows:



6.4 FMU Model Simulation

25. Setup the simulation.

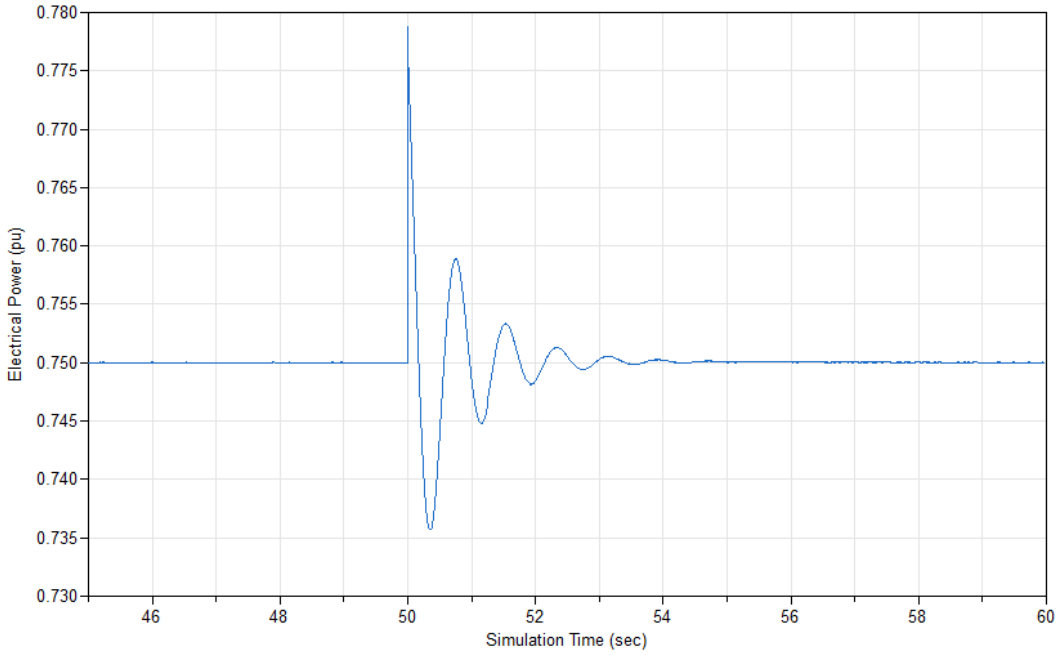
    In ***Simulink***, press the button "Model Configuration Parameters". Set the start time and stop time. Try stiff solvers like ode23s or fixed-step solvers with a fundamental sample time equal to the noise sample time from the Modelica source model. With the variable-step solvers adjust the tolerance value so that convergence is possible. For additional information on the MATLAB solvers selection criteria, please refer to: https://se.mathworks.com/help/matlab/math/choose-an-ode-solver.html. It is also recommended to give a try to the diagnostic options which are available in the *Configuration Parameters* window.

    In ***OpenModelica***, press the button "Simulation Setup". Set the start time and stop time. Then, specify the interval or the number of intervals,

considering that the general step size without control equals $\frac{stop\_time - start\_time}{num\_of\_intervals}$. Bear in mind that the only two integration methods with step size control in OpenModelica are *ida* and *dassl* (the recommended solver to start with). It is also recommended to start with a less ambitious tolerance value than the default one of 1e-6.

In ***Dymola***, go to the "Simulation Setup" window by pressing the button  . Set the simulation interval and an interval length equal to the noise sample time from the Modelica source model. Try stiff solvers such as *Esdirk23a* or the faster fixed-step solvers with a step size also equal to the noise sample time. For more details on the choice of the solver, refer to the section "Selecting the integration method" of the "Dynamic Model Simulator" chapter in the Dymola User Manual.

26. Run the simulation. Repeat step 25 with different settings in case the model was not simulated properly. The following picture shows the result of a simulation of the multi-domain model with FMUs in Dymola (*Esdirk23a* solver):



Chapter VII

7. CLOSURE

7.1 Summary

In this thesis, a multi-domain equation-based and semantic model has been developed. The model comes from a combination of a typical SMIB power system representation and a somehow detailed generic gas turbine model. The turbine model requires a good understanding of certain concepts of the Turbomachinery domain and for this reason, chapter two of this report has been devoted to this purpose.

The implementation and testing of this multi-domain model has two drivers: **a)** to serve as a test bench for the European ITEA OpenCPS project and, **b)** to suggest a more accurate model for gas turbines as an alternative to the ones currently used by the power system simulation tools.

The OpenCPS project aims to provide interoperability between UML and FMI standards, verified code generation and improved (co-)simulation speed. The multi-domain model is a valid cyber-physical scenario case, since it involves a tightly integrated modelling of both physical and control systems. But, for the model to be used as a test case, it must follow the workflows proposed by OpenCPS. One example of these workflows is the import and simulation of FMUs, so a general procedure was described in Chapter 6. The FMU model compilation and simulation was also tested and results were briefly discussed at the end of Chapter 5.

The models were implemented in Dymola, a development tool for the object-oriented equation-based language Modelica. Chapter 3 was dedicated to the detailed description of the multi-domain model assembly, which required the use

of components from OpenIPSL and ThermoPower libraries. A more realistic modeling of the load based on the noise library was also considered.

The multi-domain model simulation should also meet European standardization requirements for information exchange through the IEC-CIM UML-based Common Grid Modelling Exchange Standard (CGMES). Therefore, a multi-domain semantic model based on a mixture of CIM and ISO 15926 was proposed in Chapter 4.

A comparative study between the multi-domain model and a model that was based only on standard components of the electrical industry was performed in Chapter 5. The analysis of the results sought to emphasize differences in accuracy to promote the need for more detailed models.

7.2 General Conclusions and Recommendations

The following conclusions and recommendations can be drawn from the present thesis:

- A multi-domain semantic and equation-based model has been derived to allow simulations of detailed representations of gas turbines and the electric power grid. Although the models have been restricted due to the lack of available information and time, the modeling methodology provides a framework for future studies with multi-domain models in power systems.
- Differences in the simple turbine model (GGOV1) and the multi-domain explicit turbine model have been shown. A relevant source of that difference is the *representation of the speed influence* on the *gas turbine dynamics*. The study was however limited by the lack of measurements that could have served as a reference for the model's tuning and validation. It would also be interesting to analyze the differences between the models in other network variables and not only in the generator response.

- It is possible to exchange the models in the form of FMUs. This opens an opportunity of getting detailed models of the gas turbines from the manufacturers while protecting their intellectual property. The right choice of the *solver* and the *process noise modeling* (load and/or generation uncertainties) is still a challenge.

7.3 Future Work

A better model that includes among other things the *valves dynamics* is desired. This will be useful in the study of *forced oscillations phenomena* in power systems coming from gas power plants. This model can be extended to cover *Combined-cycle power plants*. Several possible future studies using multi-domain models with other power generation technologies such as hydro plants or geothermal plants could also be carried out. Both *ThermoPower* and *ThermoSysPro* libraries contain a complete model of combined cycle plants that can be used as a starting point.

Further work needs to be done to better apply the ISO 15926 standard in multi-domain semantic modeling applications with CIM. It is highly recommended to keep the ISO 15926 semantic model in OWL notation. Thus, the coupling between semantics from CIM and ISO 15926 could be done at the syntax level layer. More specifically, a single RDF/XML file could be used but with two different namespaces.

BIBLIOGRAPHY

[1] Carnegie Mellon University, "Managing Variable Energy Resources to Increase Renewable Electricity's Contribution to the Grid," 2013.

[2] IEA, "Energy, Climate Change & Environment - 2016 Insights," 2016.

[3] F. P. De Mello and D. J. Ahner, "Dynamic models for combined cycle plants in power system studies," *IEEE Trans. Power Syst.*, vol. 9, no. 3, 1994.

[4] S. K. Yee, J. V Milanovic, and F. M. Hughes, "Overview and comparative analysis of gas turbine models for system stability studies," *IEEE Trans. power Syst.*, vol. 23, no. 1, pp. 108–118, 2008.

[5] CEN-CENELEC-ETSI Smart Grid Coordination Group, "Smart Grid Reference Architecture," no. November, pp. 1–46, 2012.

[6] "European Commission Third Energy Legislative Package." [Online]. Available: https://ec.europa.eu/energy/en/topics/markets-and-consumers/market-legislation.

[7] "Regulation (EC) No 714/2009 of The European Parliament and of the Council of 13 July 2009," 2009. [Online]. Available: https://www.energy-community.org/pls/portal/docs/1164178.PDF.

[8] L. Vanfretti, W. Li, T. Bogodorova, and P. Panciatici, "Unambiguous power system dynamic modeling and simulation using modelica tools," in *Power and Energy Society General Meeting (PES), 2013 IEEE*, 2013, pp. 1–5.

[9] F. J. Gómez, L. Vanfretti, and S. H. Olsen, "Binding cim and modelica for consistent power system dynamic model exchange and simulation," in *Power & Energy Society General Meeting, 2015 IEEE*, 2015, pp. 1–5.

[10] ITEA3, "OpenCPS - Open Cyber-Physical System Model-Drive Certified Development," 2017. [Online]. Available: https://itea3.org/project/opencps.html. [Accessed: 19-Jun-2017].

[11] L. Vanfretti, T. Rabuzin, M. Baudette, and M. Murad, "iTesla Power Systems Library (iPSL): A Modelica library for phasor time-domain simulations," *SoftwareX*, vol. 5, pp. 84–88, 2016.

[12] P. Jansohn, *Modern gas turbine systems: High efficiency, low emission, fuel flexible power generation*. Elsevier, 2013.

[13] A. M. Y. Razak, *Industrial gas turbines: performance and operability*. Elsevier, 2007.

[14] M. P. Boyce, *Gas turbine engineering handbook*. Elsevier, 2012.

[15] E. Thirunavukarasu, "Modeling and Simulation Study of a Dynamic Gas Turbine System in a Virtual Test Bed," 2013.

[16] C. Soares, *Gas Turbines A Handbook of Air, Land and Sea Applications*. 2014.

[17] P. P. Walsh and P. Fletcher, *Gas turbine performance*. John Wiley & Sons, 2004.

[18] S. E. Turie, "Gas Turbine Plant Modeling for Dynamic Simulation," p. 58, 2011.
[19] F. Casella, "ThermoPower Gas Library," 2009. [Online]. Available: http://home.deib.polimi.it/casella/thermopower/help/ThermoPower_Gas.html#ThermoPower.Gas.
[20] E. Larsson, "Diagnosis and Supervision of Industrial Gas Turbines," Link{ö}ping University Electronic Press, 2012.
[21] P. Pourbeik, "Dynamic models for turbine-governors in power system studies," *IEEE Task Force Turbine-Governor Model.*, 2013.
[22] F. G. Dent, "Microgovernor – A Replacement of Existing Large Steam Turbine Governing Controls," *IEEE Conf. Refurb. Power Stn. Electr. Plant*, pp. 128–132, 1988.
[23] P. Fritzson and P. Bunus, "Modelica-a general object-oriented language for continuous and discrete-event system modeling and simulation," in *Simulation Symposium, 2002. Proceedings. 35th Annual*, 2002, pp. 365–380.
[24] F. Casella and A. Leva, "Modelica open library for power plant simulation: design and experimental validation," in *Proceeding of the 2003 Modelica conference, Linkoping, Sweden*, 2003.
[25] B. El-Hefni, D. Bouskela, and G. Lebreton, "Dynamic Modelling of a Combined Cycle Power Plant with ThermoSysPro," *Proc. 9th Model. Conf.*, pp. 365–375, 2011.
[26] A. Idebrant *et al.*, "Gas Turbine Applications using ThermoFluid," *Proc. 3rd Int. Model. Conf.*, 2003.
[27] F. Casella, "Object-oriented modelling of power plants: a structured approach," *IFAC Proc. Vol.*, vol. 42, no. 9, pp. 249–254, 2009.
[28] C. Roberts, E. M. Stewart, and F. Milano, "Validation of the Ornstein-Uhlenbeck process for load modeling based on ??PMU measurements," *19th Power Syst. Comput. Conf. PSCC 2016*, 2016.
[29] D. Becker and T. L. Saxton, "The Missing Piece in Achieving Interoperability – a Common Information Model (CIM)-Based Semantic Model Grid," *Grid-Interop Forum 2007*, no. Cim, pp. 125–1--125–6, 2007.
[30] UN, "UN/CEFACT ebXML Core Components Technical Specification (CCTS)," 2002. [Online]. Available: https://www.unece.org/cefact/codesfortrade/ccts_index.html. [Accessed: 22-Jun-2017].
[31] B. Kim and H. Teijgeler, "A Representation and Implementation of Process Plant Models using OWL and ISO 15926," no. July 2015, 2008.
[32] ISO, "15926: Industrial automation systems and integration--Integration of life-cycle data for process plants including oil and gas production facilities - Part 2 Data model," 2003.
[33] ISO, "15926: Industrial automation systems and integration--Integration of life-cycle data for process plants including oil and gas production

facilities - Part 4 Initial reference data," 2007.
[34] R. Batres, M. West, D. Leal, D. Price, and Y. Naka, "An upper ontology based on ISO 15926," *Comput. Aided Chem. Eng.*, vol. 20, no. C, pp. 1543–1548, 2005.
[35] ISO, "15926: Industrial automation systems and integration--Integration of life-cycle data for process plants including oil and gas production facilities - Part 8 Implementation methods for the integration of distributed systems : Web Ontology Language (OWL)," 2011.
[36] M. Baur, M. Otter, and B. Thiele, "Modelica Libraries for Linear Control Systems LinearSystems library," *Proc. 7th Int. Model. Conf.*, pp. 20–22, 2009.
[37] L. Vanfretti *et al.*, "An open data repository and a data processing software toolset of an equivalent Nordic grid model matched to historical electricity market data," *Data Br.*, vol. 11, pp. 349–357, 2017.

APPENDIX A

SMIB System data:

Branch Data ($V_b = 13.8$ kV)			
From bus	To bus	R (pu)	X (pu)
GEN1	BUS1	0.00000	0.15000
BUS1	GEN2	0.00001	0.2000
BUS1	LOAD	0.00003	0.06000
LOAD	BUS2	0.00035	0.07000
BUS2	GEN2	0.00035	0.07000
LOAD	BUS3	0.00000	0.00001

Generator parameters	Gen-1
T'_{d0}	5.0000
T''_{d0}	0.0500
T'_{q0}	0.7000
T''_{q0}	0.1000
Inertia H	4.0000
Damping D	0.0000
X_d	1.4100
X_q	1.3500
X'_d	0.3000
X'_q	0.6000
$X''_d = X''_q$	0.2000
X_l	0.1200
$S_{1.0}$	0.1000
$S_{1.2}$	0.5000
R_a	0.0000

*Generator capacity M_b in Gen-1 is 10 MVA, while in Gen-2 is 1000 MVA.

IEEET1	
$T_R(sec)$	1.0000
K_A	40.000
$T_A(sec)$	0.0400
V_{RMAX}	7.3000
V_{RMIN}	-7.3000
K_E	1.0000
$T_E(sec)$	0.8000
K_F	0.0300
$T_F(sec)$	1.0000
E_1	2.4000
$SE(E_1)$	0.0300
E_2	5.0000
$SE(E_2)$	0.5000

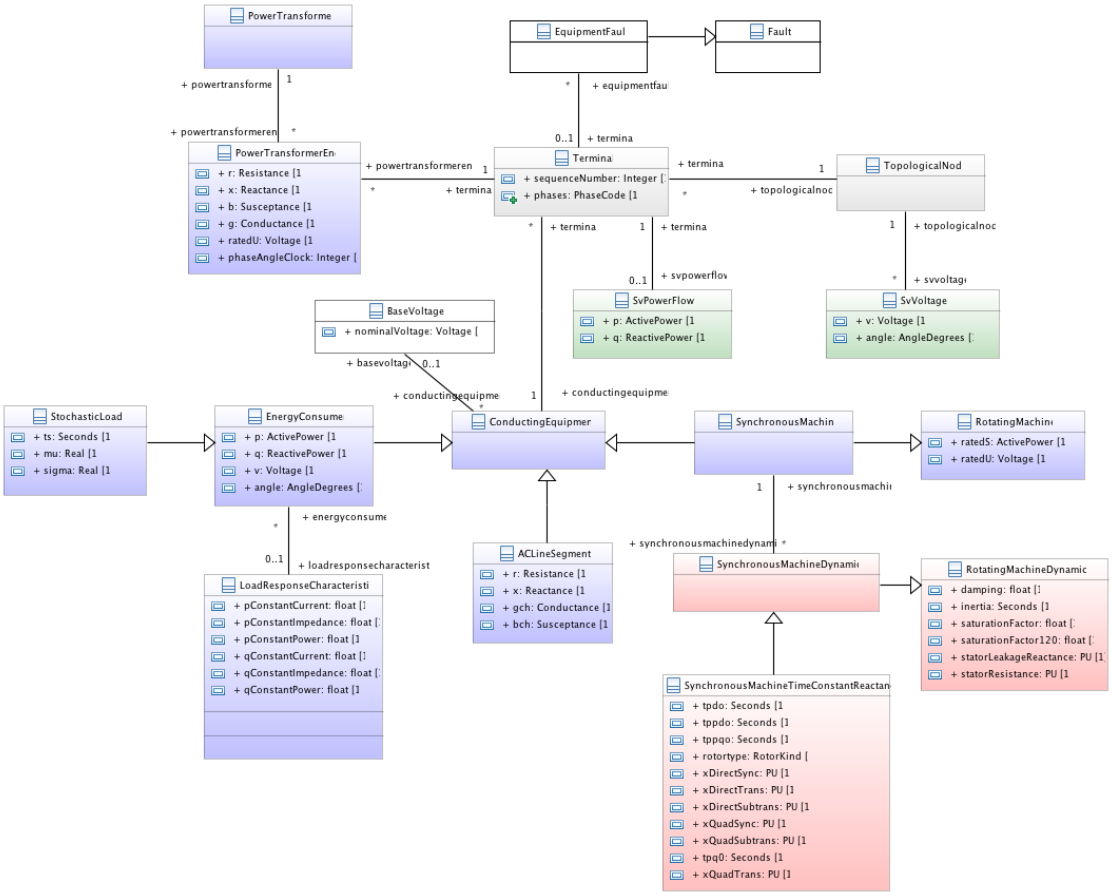
PSS2A	
$T_{W1}(sec)$	10.0000
$T_{W2}(sec)$	10.0000
$T_{W3}(sec)$	10.0000
$T_{W4}(sec)$	1x10e-9
$T_1(sec)$	0.1500
$T_2(sec)$	0.0250
$T_3(sec)$	0.1500
$T_4(sec)$	0.0250
$T_8(sec)$	0.5000
$T_9(sec)$	0.1000
$K_{S1}(pu)$	20.0000
$K_{S2}(pu)$	0.9900
$K_{S3}(pu)$	1.0000
$V_{STMAX}(pu)$	0.1000
$V_{STMIN}(pu)$	-0.1000
M	0
N	0

GGOV1*	
R (pu)	0.04
T_{pelec} (sec)	1.00
max_{err} (pu)	0.05
min_{err} (pu)	-0.05
K_{pgov} (pu)	10.00
K_{igov} (pu)	5.00
K_{dgov} (pu)	0.00
T_{dgov} (sec)	1.00
K_{turb} (pu)	1.50
D_m (pu)	0.00
K_{imw} (pu)	0.00
db (pu)	0.00
V_{max} (pu)	1.00
V_{min} (pu)	0.10
W_{fnl} (pu)	0.15
FLAG	0

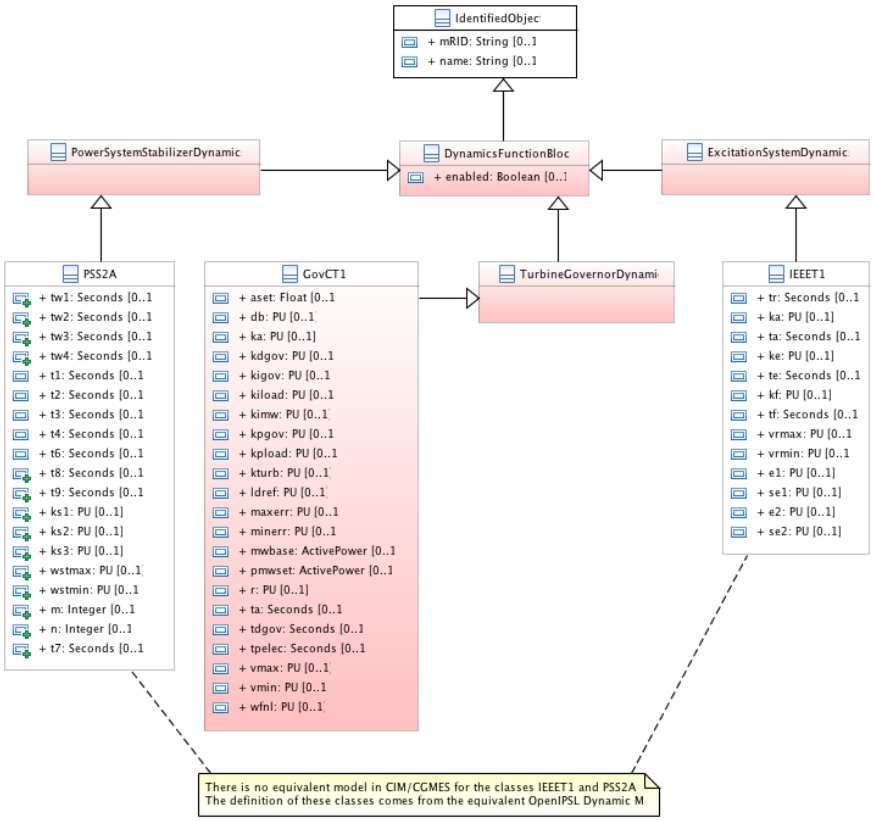
*Governor only

APPENDIX B

CIM Class Diagrams



SMIB Network Components

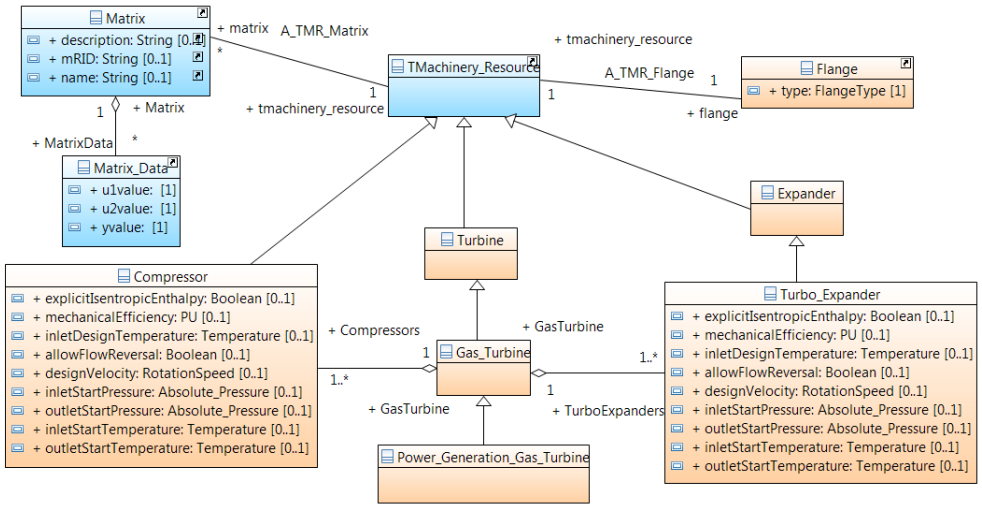


Generator and Turbine Controls

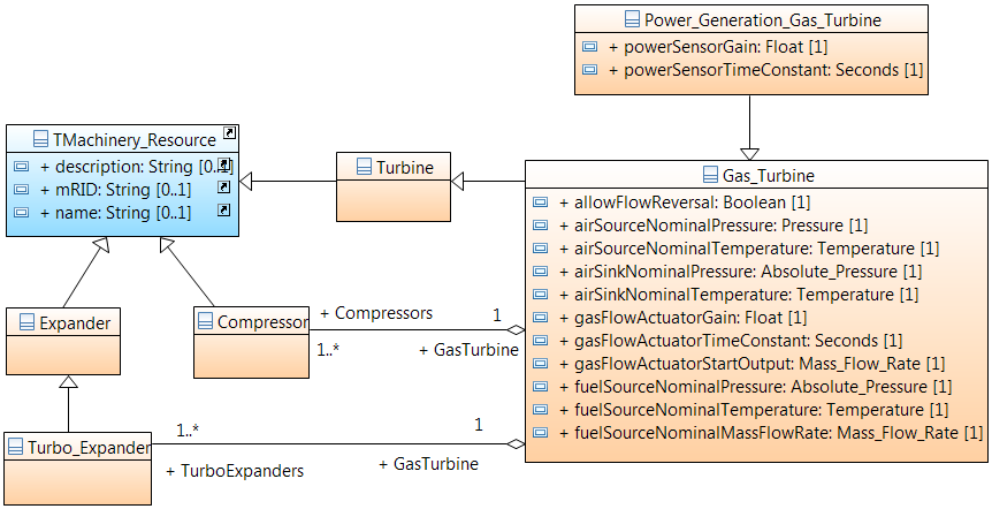
Color Convention: Purple for EQ profile, gray for TP, green for SV, red for DY

APPENDIX C

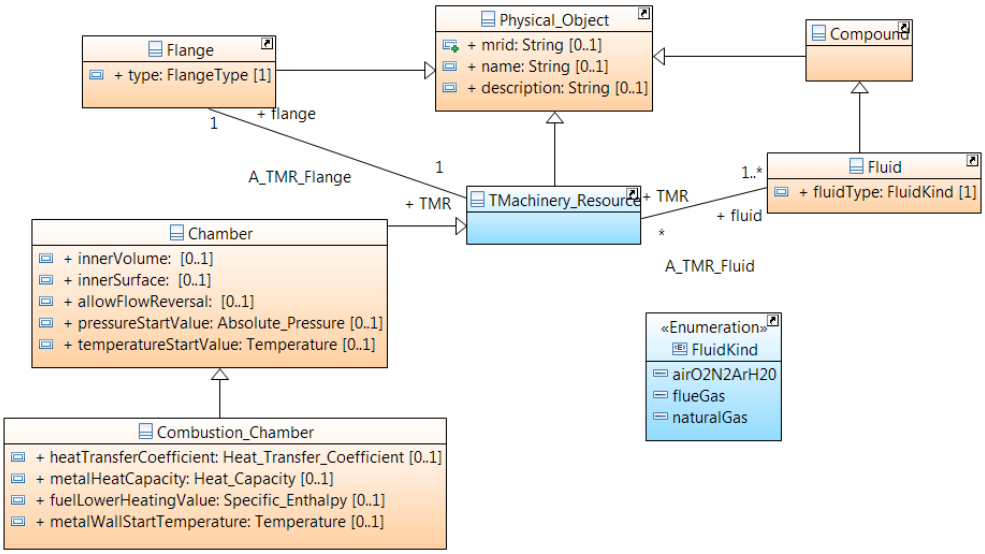
ISO Class Diagrams



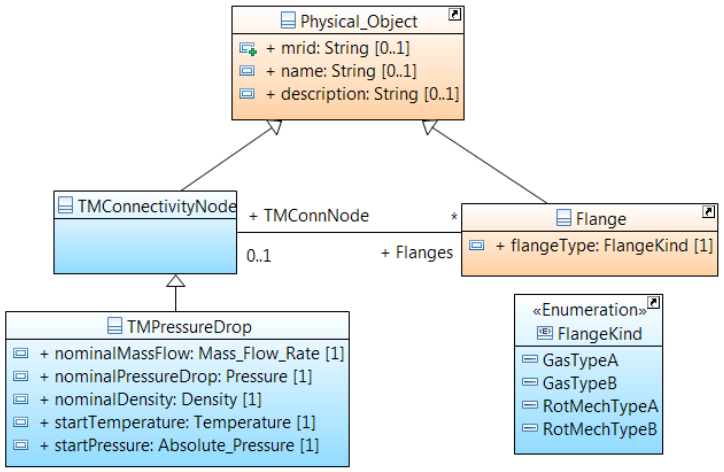
Rotating Equipment



Power Generation Gas Turbine Components



Heat Transfer



Connectivity

Color Convention: Orange for ISO-inspired classes, light blue for new classes

APPENDIX D

Pieces of Python Scripts

smib.py (modified from Nordic44 scripts)

```
"""Functions for updating the PSS/E data sets."""
import os
import sys
import math
import warnings
import psspy
import redirect
import numpy as np

class SMIB(object):
    """Class to create raw files for different SMIB power
flow cases
    """
    def __init__(self, basecase=None):
        if not basecase:
...
    def update_raw_files(self, out_dir=None):
        """Function for updating the psse case file.
        """
        if not out_dir:
            out_dir = os.getcwd()

        redirect.psse2py()
...
        for p in np.arange(5.0, 9.8, 0.1):
            print('Changing load and generator data...')
            psspy.load_data_3(23, str(1), realar1=p,
realar2=p/5)
            psspy.machine_data_2(1, str(1), realar1=p)
...
        psspy.close_powerflow()
```

utilities.py (modified from Nordic44 scripts)

```
"""Run several power flow and store the results as Modelica
datasets"""
import os
import shutil

def pfdata_generator(out_dir=None, records=False, psse=True):
    """The main function."""

    cwd = os.getcwd()
    if not out_dir:
        out_dir = cwd

    if psse:
        from multidomain.smib import SMIB
        raw_dir = os.path.join(out_dir, "PSSE_Resources")
        os.mkdir(raw_dir)
        if records:
            from multidomain.readraw import Reader
            from multidomain.torecord import Record
            record_dir = os.path.join(out_dir, "Records")
            os.mkdir(record_dir)

        dir_str = "SMIB_Jun03"
        tmp_raw = os.path.join(raw_dir, dir_str)
        os.mkdir(tmp_raw)

        if psse:
            smib = SMIB()
            smib.update_raw_files(out_dir=tmp_raw)
            dir_befraw = "more"
            tmp_raw_bfpf = os.path.join(record_dir,
dir_befraw)
            os.mkdir(tmp_raw_bfpf)
            files = os.listdir(tmp_raw)
            for f in files:
                if 'bef' in f.lower():

shutil.move(os.path.join(tmp_raw,f),tmp_raw_bfpf)

            if records:
                reader = Reader(tmp_raw)
                lista = reader.get_list_of_raw_files()
                tmp_record_dir = os.path.join(record_dir,
dir_str)
```

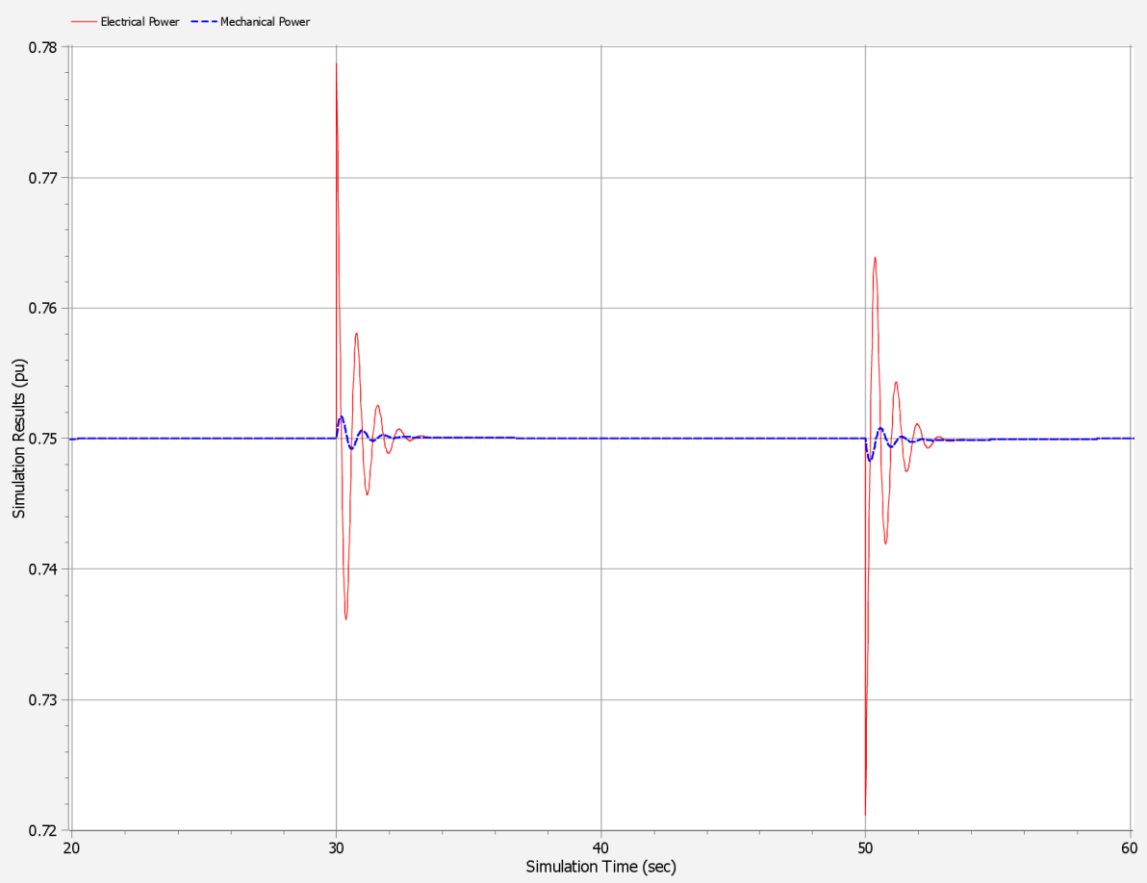
```
                if not os.access(tmp_record_dir, os.F_OK):
                    os.mkdir(tmp_record_dir)

                record_file = open(tmp_record_dir +
'\Records.mo', 'w+')
                record_file.write('package Records' + '\n
//Power flow solution' +
                                  ' records \n  extends
Modelica.Icons.Package;')
                record_file.close()
                for raw in lista:
                    reader.open_raw(raw)
                    reader.read_raw()
                    record = Record(tmp_record_dir +
'\Records.mo',
                                    reader.case_name,
                                    reader.buses,
reader.machines,
                                    reader.loads,
reader.trafos)
                    record.write_voltages()
                    record.write_machines()
                    record.write_loads()
                    record.write_trafos()
                    record.close_record()

                record_file = open(tmp_record_dir +
'\Records.mo', 'a')
                record_file.write('\n \n')
                record_file.write('end Records;')
                record_file.close()
```

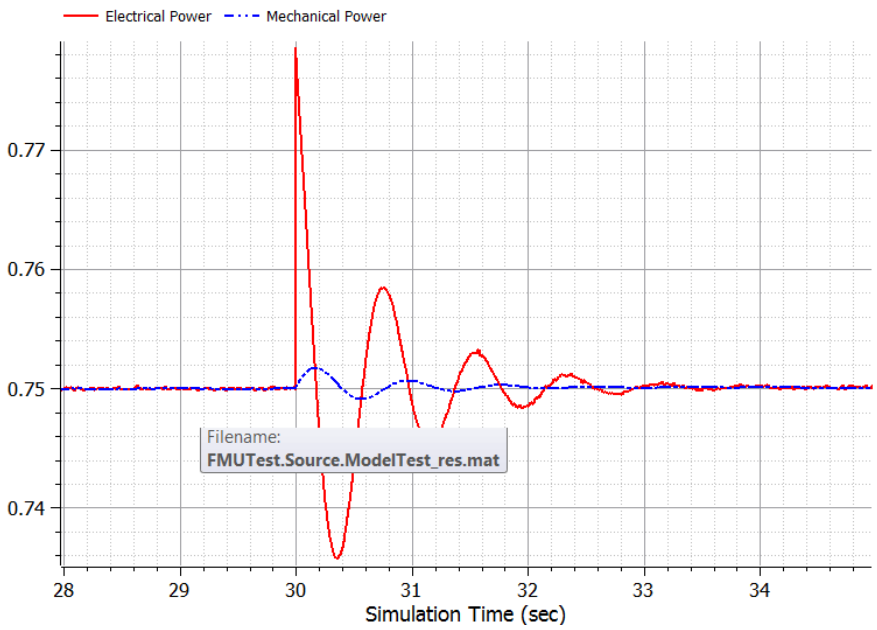
APPENDIX E

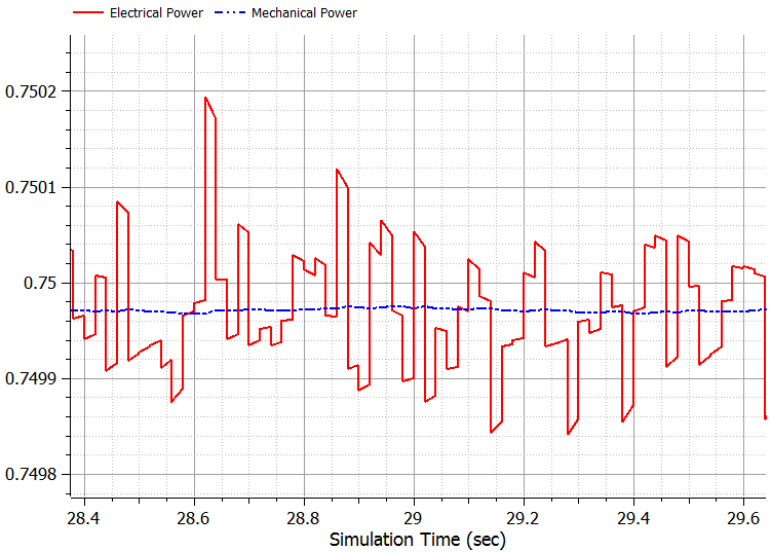
Multi-Domain Model Simulation in OpenModelica (no noise)



Simulation done with DASSL solver, tolerance equal to 1e-6 and 5000 intervals

Multi-Domain Model Simulation in OpenModelica (noise)

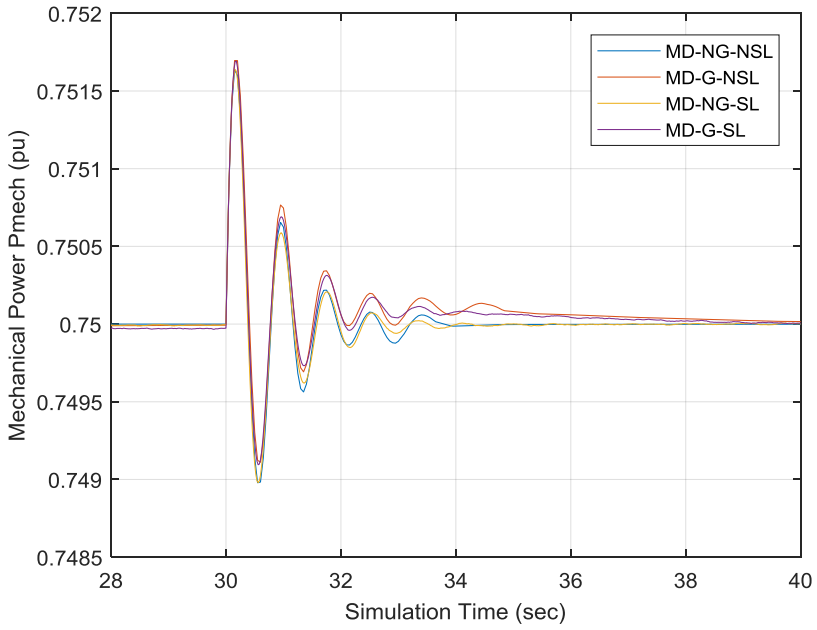


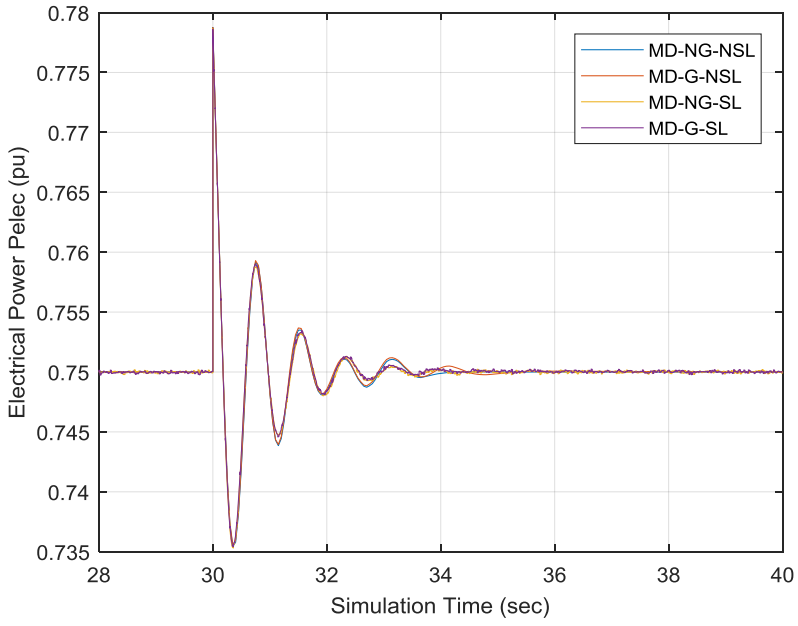


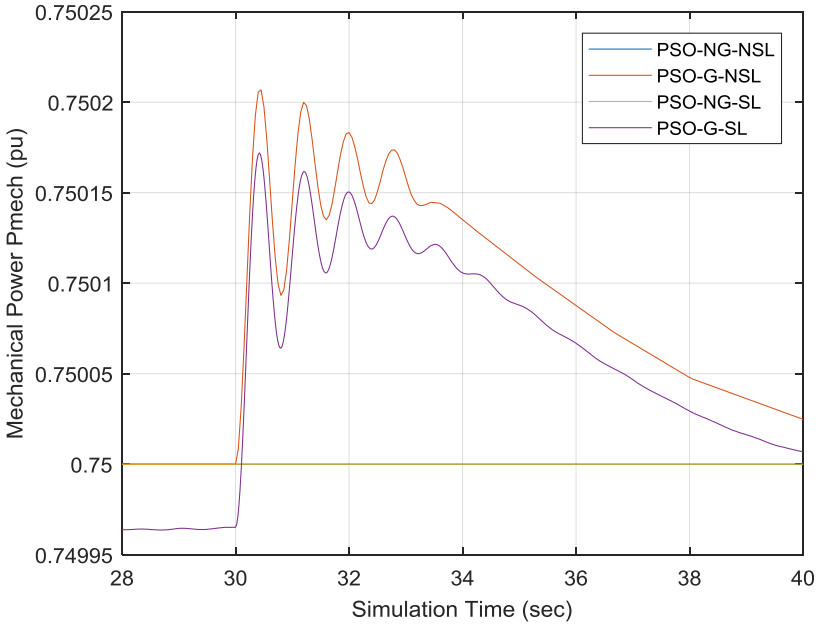
Simulation done with DASSL solver, default tolerance and interval of 0.02 sec.

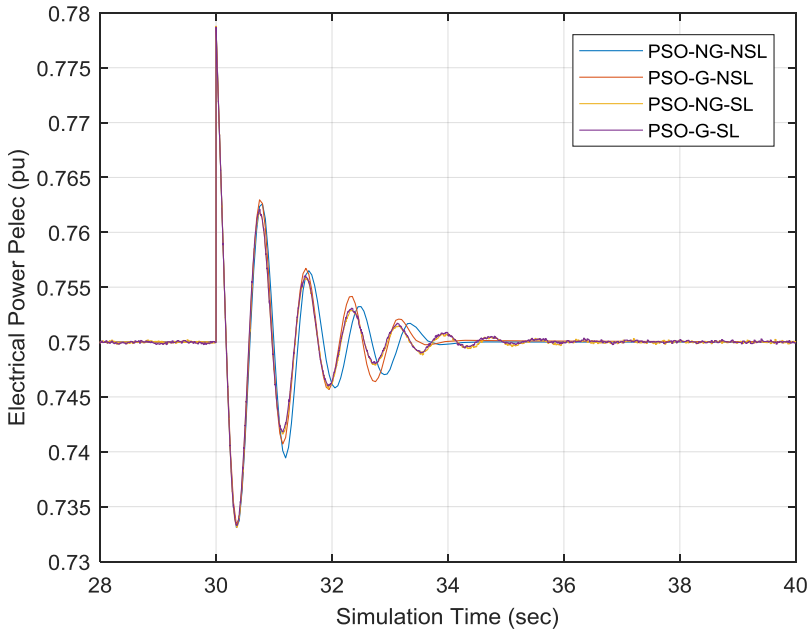
APPENDIX F

Simulation Results With / Without Governor









* MD: Multi-Domain, PSO: Power-System-Only; NG: Without governor, G: With governor; NSL: Non-stochastic load, SL: Stochastic load

TRITA TRITA-EE 2017:097
ISSN 1653-5146