

# Parameter Identification and Information Modeling of Gas Turbine Models for Power System Simulations

Fanny Radesjö

**Abstract**— In the future renewable generation sources will have to co-exist with traditional generation sources that can overcome their limitations in providing secure power supply to society, such as gas turbines.

In this project we proposed to substitute a simple model of a gas turbine by a detailed model to simulate the turbine's behavior in a real system. Parameter identification of a simple gas turbine model was performed based on measurements from a detailed model. By this, the project contributes to a library of electric grid component models. The library is meant to allow operators and engineers to predict the system behavior by simulations. This can facilitate to anticipate possible instability in the grid and thereby contribute to the transition to the Smart Grid.

Further, another gas turbine model was in the Modelica language was represented using the Unified Modeling Language (UML). This is a standard form of modelling that simplifies exchange of model information between project groups.

## I. INTRODUCTION

TACKLING the climate change is one of the greatest challenges the world is facing today. A big source of the problem lies in our production of electricity. Combustion of fossil fuels for electric power generation releases greenhouse gases to the atmosphere that contributes to a warmer climate on earth. Even though there is plenty of clean and renewable energy sources available they are often intermitted and not controllable. The unreliable behavior in terms of not enough generation that occur from wind and solar power generation is compensated by conventional power plants that are often powered by fossil fuels. To minimize the generation part of conventional power plants, smart algorithms on the monitoring and high level control can be developed and implemented, contributing to the Smart Grid development. [1]

A detailed, multi domain model of a gas turbine have been developed by a research group, called SmarTS lab as part of an international project named OpenCPS. The goal of OpenCPS is to benefit the transition to Smart Grids by researching close integration between computational, networking and physical processes. [2]

In this paper the parameters of a simple turbine model was calibrated to give the same output as the multi domain model. This was done using the Rapid Parameter Identification (RaPI) software. RaPI tries to tune parameters to create an output curve that fits the output of the simulation. [3]

Further, a simpler model of a turbine and governor from the OpenIPSL library was presented using the Unified Modeling Language (UML) and the Common Information Model (CIM) standard to facilitate information sharing. This is convenient for future projects to be able to use the model's information even if they work with other programs. In UML, the model with all its components, equations, parameters and connections is described in a way that is simple and easy to comprehend. [2]

## II. NOMENCLATURE

SYMBOL	QUANTITY	UNIT
$W$	Work	J
$\dot{m}$	Mass flow	kg/s
$m$	Mass	kg
$h$	Specific enthalpy	J/kg
$\tau$	Torque	J/rad
$\omega$	Angle	rad
$\eta$	Efficiency	-
$PR$	Pressure Ratio	-
$P$	Pressure	bar
$\dot{M}$	Molecular mass flow	kmol/s
$u$	Molar internal energy	J/kmol
$LHV$	Lower heating value	J/kg
$Q$	Heat transferred	J
$\gamma$	Stoichiometric ratio	-
$A$	Area	m <sup>2</sup>
$T$	Temperature	K
$V$	Volume	m <sup>3</sup>
$\rho$	Density	kg/m <sup>3</sup>
$t$	Time	S
$n$	Amount of substance	kmol
$K_{turb}$	Gain for model GGOV1	-
$T_c$	Lead time constant for GGOV1	s
$T_b$	Lag time constant for GGOV1	s
$T_{eng}$	Delay time for model GGOV1	s
$P_{MECH}$	Mechanical power GGOV1	pu
$WF$	Fuel flow GGOV1	kg/s
$W_{fnl}$	Fuel flow at maximum speed GGOV1	kg/s
$R$	Droop for model TGtypeII	pu
$p_{max0}$	Maximum turbine output TGtypeII	pu
$p_{min0}$	Minimum turbine output TGtypeII	pu
$T_s$	Governor Time constant TGtypeII	s
$T_3$	Transient gain time constant TGtypeII	s
$S_n$	System base power TGtypeII	MVA
$S_b$	Nominal power TGtypeII	MVA

### III. GAS TURBINE WORKING PRINCIPLE

A gas turbine consists of at least three main parts. These are the compressor the combustion chamber and the turbine. The operation of a gas turbine can be described by the thermodynamic cycle called the Brayton cycle. (Figure 1) [4]

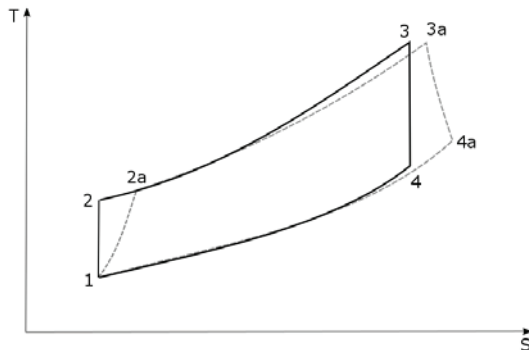


Figure 1 Ideal Brayton cycle (solid) and actual Brayton Cycle (dashed)

Air from the surrounding enters the compressor through an inlet at state 1. (Figure 1) An ideal compressor is said to follow an isentropic process. This means the entropy remains constant though other properties change. Energy must be added to the process for the compression to take place. Around 50% of the energy generated by the cycle is usually used to drive the compressor. The remaining 50% powers the load. In electricity production applications the load is a generator.[4]

The compressed air continues to enter the combustion chamber where both temperature and entropy are increased by adding heat. The heat is generated by combusting fuel gas mixed with air. This process is isobaric which means it is done under constant pressure. [4]

After the combustion, the flue gases enter the turbine at state 3. An isentropic expansion decreases both pressure and temperature. The difference in energy between state 3 and 4 results in a torque in the shaft which powers both the compressor and load. At the end of the cycle the warm exhausts are released to the surrounding and new air enters the compressor. [4]

In an actual gas power plant the cycle is not ideal. (Figure 1) The expansion and compression processes are not isentropic and there are pressure losses in process 2-3 and 4-1. All losses are usually compiled to an efficiency parameter ( $\eta$ ). [4]

### IV. MULTI DOMAIN GAS TURBINE MODEL

The detailed gas turbine model is built in Modelica using blocks and models from the ThermoPower library included in Open Modelica (The Model is made by Miguel Aguilera). [5]

The model is multi domain, meaning it combines equations describing mechanical, thermodynamic and chemical behaviors. (Figure 2) By giving the model inputs of fuel flow entering the combustion chamber and air mass flow entering the compressor, it simulates the processes in the plant resulting in mechanical power as a rotation of the shaft. Medium packages from the ThermoPower library are used to describe the thermodynamic states of the air and fuel entering the system.

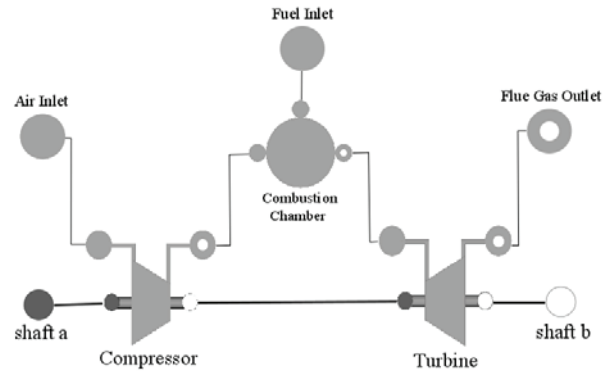


Figure 2 Model structure for gas turbine plant model

#### A. Compressor

Rotation of the shaft and the thermodynamic state of outside air are inputs of the compressor component. The compressor model consists of a combination of equations and look up tables. The equations describe the conservation of energy through the process (Equation 1), the ratio between the pressure before and after the compression (Equation 2) and the compressor efficiency (Equation 3).

$$W_c = \dot{m}_a \cdot (h_2 - h_1) = \tau \cdot \omega \cdot \eta_{mech} \quad (1)$$

$$PR_c = \frac{P_2}{P_1} \quad (2)$$

$$\eta_c = \frac{h'_2 - h_1}{h_2 - h_1} \quad (3)$$

Look up tables are used to describe the two characteristic equations that define the compressor performance map. The first equation relates the flow number ( $phic$ ) to the pressure ratio ( $PR$ ) and referred speed, and the second relates the efficiency ( $eta$ ) to the flow number and referred speed. After introducing one additional variable,  $beta$ , three look up tables can be obtained ( $tablePhic$ ,  $tablePR$  and  $tableEta$ ). The values of referred flow, pressure ratio and efficiency can be determined by inspecting a given  $beta$  value and referred speed. [6]

#### B. Combustion chamber

The thermodynamic processes in the combustion chamber is described by Equation 4 to 7, together with the balanced chemical reaction. Equation 4 describes conservation of mass, Equation 5 conservation of energy and Equation 6 the heat transfer from the chamber walls to the gas. The assumptions are made that there are no meaningful pressure losses, the heat transfer is constant and that the chamber is isolated from the outside. These assumptions influence both Equation 5 and 6.

$$V_{cc} \frac{d\rho}{dt} = \dot{m}_{air} + \dot{m}_{fuel} - \dot{m}_{flue} \quad (4)$$

$$\frac{d(\dot{M}_{cc} u_{cc})}{dt} = \dot{m}_{air} h_{air} + \dot{m}_{fuel} LHV - \dot{m}_{flue} h_{flue} - Q \quad (5)$$

$$Q = \gamma \cdot A \cdot (T_{wall} - T_{gas}) \quad (6)$$

The stoichiometric ratio ( $\gamma$ ) for Equation 6 is defined by Equation 7 and are dependent on the balanced chemical equation of the combustion.

$$\lambda = \frac{m_{air}}{m_{fuel}} = \frac{n_{air}}{n_{fuel}} \quad (7)$$

C. Turbine

The turbine model is similar to the compressor model. It is described by Equation 8 to 10. Equation 8 describes the conservation of energy throughout the process, Equation 9 defines the pressure ratio and Equation 10 defines the turbine efficiency.

$$\dot{W}_T = \dot{m}(h_4 - h_3) \cdot \eta_{mech} = \tau_T \cdot \omega \quad (8)$$

$$PR_T = \frac{P_3}{P_4} \quad (9)$$

$$\eta_T = \frac{h_2 - h_3}{h_2 - h_3'} \quad (10)$$

Same as the compressor, the turbine model uses look up tables representing characteristic equations relating mass flow, pressure ratio, referred speed and efficiency.

V. PARAMETER IDENTIFICATION USING RAPID

Assuming the mechanical power output from the detailed gas turbine model is the same as generated by a real turbine, an accurate gas turbine model for simulations was created. The parameters for a turbine and governor model named GGOV1 was adapted to give it the same output curve as the detailed gas turbine model. A table of mechanical power measurements in time domain was created. A step function was used for fuel flow input during the measuring. The same step function was set as the input for GGOV1 in Modelica before the it was exported as a FMI-file (Functional Mock-up Interface). By giving this model to the parameter identification software called RaPID it is possible to access both the FMI-file and the measurements. Using RaPID, values of defined parameters were optimized to give GGOV1 the same output as the detailed model for the same input. [3]

A. GGOV1

GGOV1 is a general turbine-governor model. In this project, only the turbine part is of interest since it is supposed to be able to replace the multi domain model in simulations. [7]

The model output is mechanical power (*PMECH*) and input fuel flow (*WF*). In the first step of the block diagram the fuel flow is compared to the fuel flow for maximum speed, expressed as a constant. Four parameters affect the model behavior, the delay time (*Teng*), the gain (*Kturb*) and the lead and lag time constants for the lead-lag-compensator (*Tc*, *Tb*). *Tc* and *Tb* are named T1 and T2 in Figure 3. [7]

In steady-state the mechanical power are given by Equation 11. *Wfnl* are equal to the constant fuel flow for maximum speed. [7]

$$P_{MECH} = K_{turb} * (WF - Wfnl) \quad (11)$$

The other blocks in Figure 3 models how the turbine reacts to changes in the input signal. The parameter *Teng* is usually set to zero for gas turbine models and the lead-lag function used to represent the lags for the gas turbine to react to changes. This block can be only a lead function by setting *Tb* to zero or only a lag function by setting *Tc* to zero. [7]

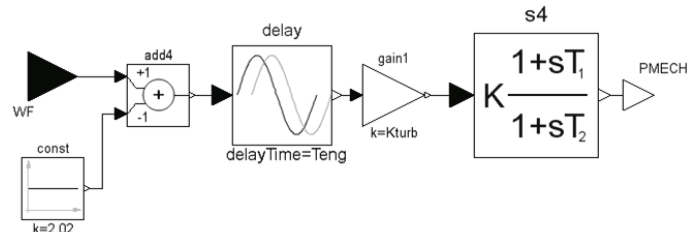


Figure 3 Block diagram of GGOV1 turbine model

A lead function has short rise time and more overshoot than a lag function that tends to give a slower response and generate a constant error. (Figure 4) [8]

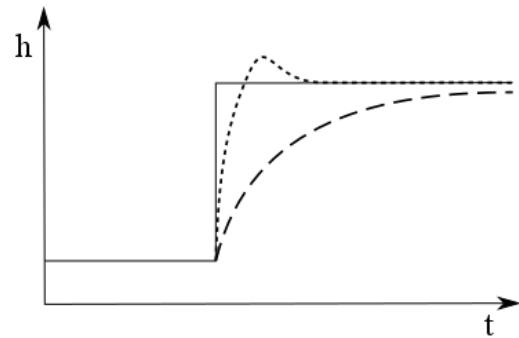


Figure 4 Input step (solid), lag function response (long dashes), lead response (short dashes)

B. Experiment set-up

GGOV1 has four parameters, *Kturb*, *Tb*, *Tc* and *Teng*. Observing the detailed model output curve, (Figure 6) it can be predicted that *Teng* and *Tb* should be set to zero. The mechanical power begins to increase at the time of the step without delay which indicated *Teng* should be zero and the shape of the curve is more similar to a lag response than a lead response which makes *Tc* equal to zero a qualified guess. (Figure 5) This means there are only two parameters to identify, *Kturb* and *Tb*. The span for *Kturb* was set as 1 to 2 and the start value 1.5. The span for *Tb* was set to 0.01 to 10 and start value 0.1. The spans were decided based on data given by power producers. If there is lack of fitness between the outputs of the two models, the spans can be changed to see if the result would improve.

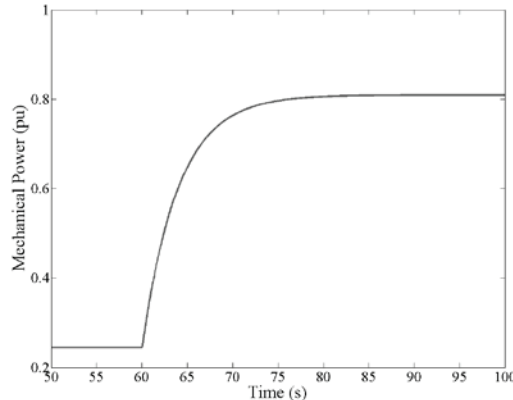


Figure 5 Output curve for the detailed gas turbine model

RaPID was used to identify the parameters given the names, spans and start values of the parameters to identify, FMI-file of GGOV1, and measurements from the detailed model. The RaPID

## K2. GAS TURBINE MODELLING

toolbox includes different methods for optimization. The choice of method is important since it affects both the identified parameter precision and computation time. To prevent the computation time from getting long when the algorithm diverges a maximum number of iterations was defined.[9] A decision was made to set this value to 1000.

Nevertheless, the methods included in RaPid are of different kind, each serves the same purpose. The program strives to adjust the output curve to minimize ta fitness value. This value is calculated by Equation 12.[9]

$$fitness = \frac{1}{M} \sum_i^M \left( \frac{y_i - y_{detail}}{y_i} \right)^2 \quad (12)$$

Four different methods were used for the adjustment. The methods were particle swarm optimization (pso), Nelder-Mead simplex method (nm), particle filter (pf) and naive. [10-13]

Particle swarm optimization (pso) starts by releasing a number of “particles” in random locations. The location of a particle represents values of the parameters and is thereby connected to a specific fitness value. Thereafter all particles are given a random velocity. In every iteration, the particles move according to their velocity. If the location of the particle after the transportation has a better fitness than before, the particle remembers its velocity for next iteration. The iterations continue until all particles are stationary in local optimums. The optimal particle location is then chosen from these. [10]

Like the pso method, the particle filter method uses random located particles. In this method particles are given a weight according to their fitness value. A particle with better fitness has a higher weight than a particle with worse. In the end of each iteration the particles are resampled with a higher density around the location of particles with high weight. This continues until all particles are gathered around the same location.[13]

The Nelder-Mead method generates n+1 points corresponding to a possible combination of parameter values, where n is the number of parameters to be identified. If n=2 the points together form a triangle. The point with the worst fitness is identified and substituted with another point. For every iteration, a new triangle with another shape and smaller area is formed. This eventually results in an optimal solution. [11] The Naive method is more straightforward but lack efficiency compared to the others. A longer runtime is expected for this method.[12]

Further some of the methods were combined. This means that after the program had ran using one method the start values were changed to the result generated before the program was restarted using a new method. Some methods are more dependent on start values than others and it can thereby be beneficial to let an efficient method generate good starting values before running these algorithms.

### C. Results

The values for the parameters, the fitness and runtime for each method and combination of methods are given by Table 1.

Table 1 Results using different solving methods

Method	Solution (K <sub>turb</sub> : T <sub>b</sub> )	Fitness	Time
Nm	1.1742 : 0.1084	0.0319	8.4930
pso	1.4120 : 6.1927	0.0039	2.8287
Pso+nm	1.5916 : 4.7299	0.0085	3.6431
	1.4124 : 3.9653	6.3386e-7	19.8735
pf	1.4981 : 6.6128	0.0036	2.6317
Pf+nm	1.4981 : 6.6128	0.0036	2.6317
	1.4124 : 3.9276	6.3386e-7	16.6718
Naive	1.4072 : 3.9276	4.3995e-6	100.1424
Pso+naive	1.4120 : 6.1927	0.0039	2.8287
	1.4140 : 3.9759	9.6690e-7	101.7683

Plots for some of the adjusted curves are visible in Figure 6 to 9.

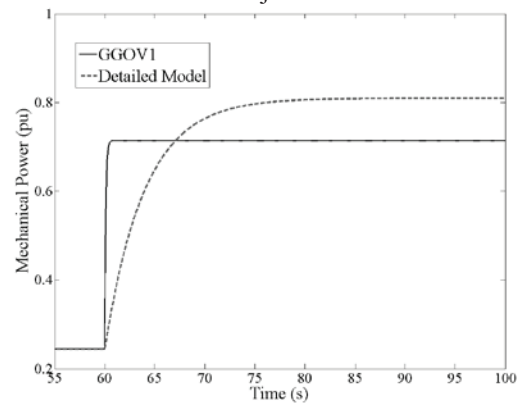


Figure 6 Nelder-Meads method

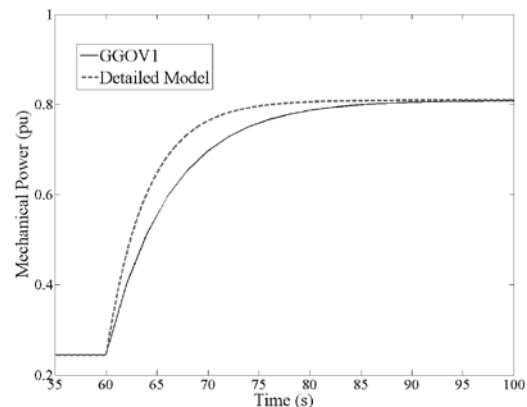


Figure 7 Particle swarming optimization

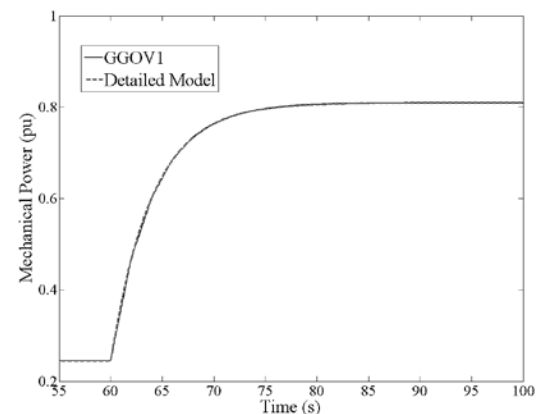


Figure 8 Particle swarm optimization and Nelder-Mead combined

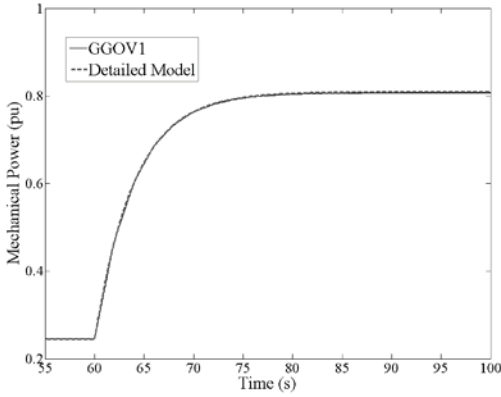


Figure 9 Naive method

The two parameter identification methods  $pso+nm$  and  $pf+nm$  gave the same results and equal fitness.  $Nm$  by itself results in a poor fitness of 0.0319. This indicates that the method relies heavily on the start value. By using another method to generate start values with better fitness a good result could be found.

The fitness achieved by the best methods are close to zero. This means the error received by using the adjusted GGOV1 model instead of the detailed model most likely is insignificant compared to error between the detailed model and reality.

### VI. CIM/UML FOR MODEL EXCHANGE

In this part of the project reverse engineering was used to express a Modelica model of a simple gas turbine, TGtypeII, connected to a generator and grid in Unified model language (UML) using the Common Information Model (CIM). A UML defines datatypes and names of components and attributes together with relationships between components. The CIM semantics then describe the characteristics of electrical and non-electrical equipment that can be found in electrical networks according to a standard. However, the CIM only models parameters for the equipment. The implementation of the mathematical behavior is specific for the model we want to describe, therefore a mapping between CIM and Modelica is required to complete the description of the models.

In this work TGtypeII and a generator model, OrderVI, have been studied and implemented, with the Modelica language, according the CIM semantics specifications.[2] (Figure 10) [14]

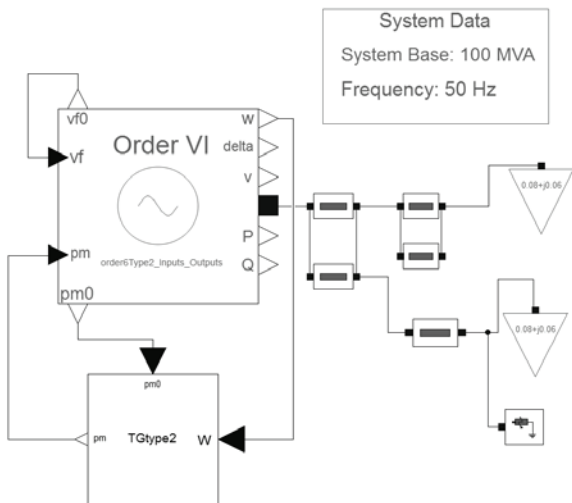


Figure 10 TGtypeII connected to generator OrderVI and grid

### A. TGtypeII

TGtypeII is the simplest gas turbine and governor model in the OpenIPSL library. [14] Figure 11 illustrates the composition of the model as a block diagram.

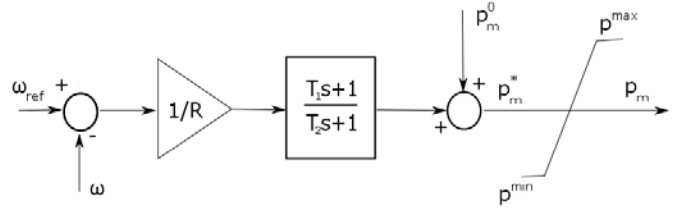


Figure 11 Block diagram of TGtypeII

The output mechanical power is calculated by a transfer function and an upper and lower limit. The inputs given are speed ( $\omega$ ) referred speed ( $\omega_{ref}$ ) and start mechanical power ( $p_m^0$ ). (Equation 13,14,15,16,17,18) [14]

$$Ro = R * \frac{S\_b}{Sn} \tag{13}$$

$$pmax = pmax0 * \frac{Sn}{S\_b} \tag{14}$$

$$pmin = pmin0 * \frac{Sn}{S\_b} \tag{15}$$

$$\dot{x}_g = \left( \frac{1}{Ro} \left( 1 - \frac{T_1}{T_2} \right) (\omega_{ref} - \omega) - x_g \right) / T_2 \tag{16}$$

$$p_m^* = x_g + \frac{1}{Ro} \frac{T_1}{T_2} (\omega_{ref} - \omega) + p_m^0 \tag{17}$$

$$p_m = \begin{cases} p_m^* & \text{if } p^{min} \leq p_m^* \leq p^{max} \\ p^{max} & \text{if } p_m^* > p^{max} \\ p^{min} & \text{if } p_m^* < p^{min} \end{cases} \tag{18}$$

This model contains 7 parameters that are described in Table 2.

Table 2 Parameters for TGtypeII

Parameter name	Description
R	Droop (pu)
pmax0	Maximum turbine output (pu)
pmin0	Minimum turbine output (pu)
Ts	Governor Time constant (s)
T3	Transient gain time constant (s)
Sn	System base power (MVA)
S_b	Nominal power (MVA)

### B. Method

All components showed in Figure 11 were not completely defined within the CIM, thus, a revision of the implementation of those models from OpenIPSL library and the CIM naming convention was the first step for designing the UML classes for those components. The parameters for the models were compared to the parameters in standard CIM models of similar components to find representatives with the same function. It was not necessary to do this identification process for the lines, loads and fault components because CIM standard models for these components already exist. Table 1 shows the original parameters of the TGtypeII and their equivalent CIM parameters. Table 2 shows the same for the OrderVI.

## K2. GAS TURBINE MODELLING

Table 3 Parameters and corresponding standard for TGtypeII

Parameter name	Initial value	Standard model	Standard parameter name
R	0.2	GovGAST	r
pmax0	1	GovGAST	vmax
pmin0	0	GovGAST	vmin
Ts	0.1	GovGAST	t <sub>1</sub>
T3	-0.1	GovGAST	t <sub>3</sub>
Sn	20	GovGAST	mwbase
S b	100	BasePower	basepower

Table 4 Parameters and corresponding standard for OrderVI

Parameter name	Initial value	Standard model	Standard parameter name
xd	1.9	SynchronousMachineTimeConstantReactance	xDirectSync
xq	1.7	SynchronousMachineTimeConstantReactance	xQuadSync
xq1	0.5	SynchronousMachineTimeConstantReactance	xQuadTrans
xd2	0.204	SynchronousMachineTimeConstantReactance	xDirectSubtrans
xq2	0.3	SynchronousMachineTimeConstantReactance	xQuadSubtrans
Td10	8	SynchronousMachineTimeConstantReactance	tpd0
Tq10	0.04	SynchronousMachineTimeConstantReactance	tpq0
Td20	0.04	SynchronousMachineTimeConstantReactance	tpd0
Td20	0.02	SynchronousMachineTimeConstantReactance	tpq0
Taa	2e-3	RotatingMachineDynamics	statorLeakageReactance

This information was then converted to map files that link the information model from the CIM definition of the components with their mathematical model implemented from the OpenIPSL models. This solution solves the problem of initialization of the Modelica models, resulting in complete models for the power system components studied in this work.

The last step was to model the relationships between the components in UML. This was done using the Papyrus UML modelling tool.

### C. Results

The complete UML model is visible in Appendix A and the map files combining the CIM semantics and Modelica models can be found in Appendix B. Together these give all information needed to recreate the model. The mathematical equations are given by the Modelica model, the parameter names and initial values by CIM and connections between components by the UML model.

Using a standard to describe a model is convenient when comparing it to other models. By giving parameters with the same function in both models the same name and same initial value they become more comparable. The benefits of using UML are that it is simple and easy to comprehend. This facilitates for other engineers to use the model.

## VII. CONCLUSION AND FUTURE WORK

An accurate gas turbine model for dynamic simulations was created to be part of a component model library. The purpose of the library is to allow operators and engineers to predict system behaviors by simulation and thereby facilitate the development of the electrical grid towards a Smart Grid. The model was created by adjusting the parameters of a simple general turbine model to give the same output as a detailed, multi domain gas turbine model. Several different methods were used for this optimization. The best result was achieved by combining particle swarm optimization or particle filtering with Nelder-Meads method which results in a fitness of 6.3386 e-7. The values of the parameters in this case were Teng = 0, Tc = 0, Tb = 3,9653 or 3,9276, and Kturb = 1,4124.

An interesting project for the future could be to investigate if these parameters gives a good fitness to measured values if another input signal is used. It could be tested if GGOV1 with the adjusted values for Tb and Kturb is an acceptable replacement for the detailed gas turbine model if the input is e.g. a ramp.

In the second part of the project a UML/CIM representation of a simple gas turbine model was created. The reason was for the model to be expressed in a standard language which makes it easy to comprehend and use in future projects. When working with problems concerning many parties, as global warming and our future energy supply, it is important to be able to share results to avoid for several people to do the exact same work. The UML representation and the CIM-Modelica map files are visible in Appendix A and Appendix B.

## VIII. ACKNOWLEDGEMENTS

The author would sincerely like to thank Tetiana Bogodorova for contributing with support and guidance during this project. Without you this would not have been possible. Further I would also like to thank Luigi Vanfretti, Francisco Gomez, Miguel Aguilera and the rest of the SmarTS lab group for all your help and support.

## IX. REFERENCES

- [1] J. Ekanayake, *et al.*, *The Smart Grid*, John Wiley & Sons, United Kingdom, pp. 1-15: 2017.
- [2] *CIM/UML and Modelica Electrical Power System Models and Tooling for MST Testing and Demonstration*, ITEA3, 2016.
- [3] L. Vanfretti, M. Baudette, A. Amazouz *et al.*, "RaPID: A modular and extensible toolbox for parameter estimation of Modelica and FMI compliant models," *SoftwareX*, vol. 5, pp. 144-149, 2016.
- [4] A. M. Y. Razak, "2 - Thermodynamics of gas turbine cycles," *Industrial Gas Turbines*, pp. 13-59: Woodhead Publishing, 2007.
- [5] (2017, may). ThermoPower [online]. Available: <https://build.openmodelica.org/Documentation/ThermoPower.html>.
- [6] P. Walsh, *Gas Turbine Performance*, 2nd ed., Hoboken: Wiley, 2008.
- [7] P. Pourbeik, "Dynamic models for turbine-governors in power system studies," *IEEE Task Force on Turbine-Governor Modeling*, 2013.
- [8] T. Glad, and L. Ljung, *Reglerteknik*, 4th ed. Lund, Sweden Studentlitteratur, pp. 93 - 116, 2015
- [9] T. Bogodorova, L. Vanfretti, and K. Turitsyn, "Bayesian Parameter Estimation of Power System Primary Frequency Controls under Modeling Uncertainties," in *IFAC-PapersOnLine* : pp. 461-465, 2015
- [10] X. Deng, "System Identification Based on Particle Swarm Optimization Algorithm," in *Computational Intelligence and Security*, Beijing, pp. 259-263, 2009
- [11] J. C. Lagarias, J. A. Reeds, M. H. Wright *et al.*, "Convergence Properties of the Nelder-- Mead Simplex Method in Low

- Dimensions,” *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112-147, 1998.
- [12] (2017, april). Naive algorithm [online]. available: [http://wcipeg.com/wiki/Naive\\_algorithm](http://wcipeg.com/wiki/Naive_algorithm).
- [13] Z. Chen, “Bayesian filtering: From Kalman filters to particle filters, and beyond,” McMaster University, Canada pp. 1-69, 2003.
- [14] F. Milano, “PSAT”, Universidad de Castilla, Spain, pp. 183, 2013.